sat-nms Software User Manual

ACU IDU User Manual

Version 3.80.6-a -- 2025-12-03

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

satnms-support@satservicegmbh.de

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

Table of Contents

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

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

These subsets are:

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

Support and Assistance

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

satnms-support@satservicegmbh.de

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

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

1 Overview

Abstract

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

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

The main sections of the user manual are:

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

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

1.1 Using The Online Help

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

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

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

Invoking The Online Help

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

The Navigation Buttons

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

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

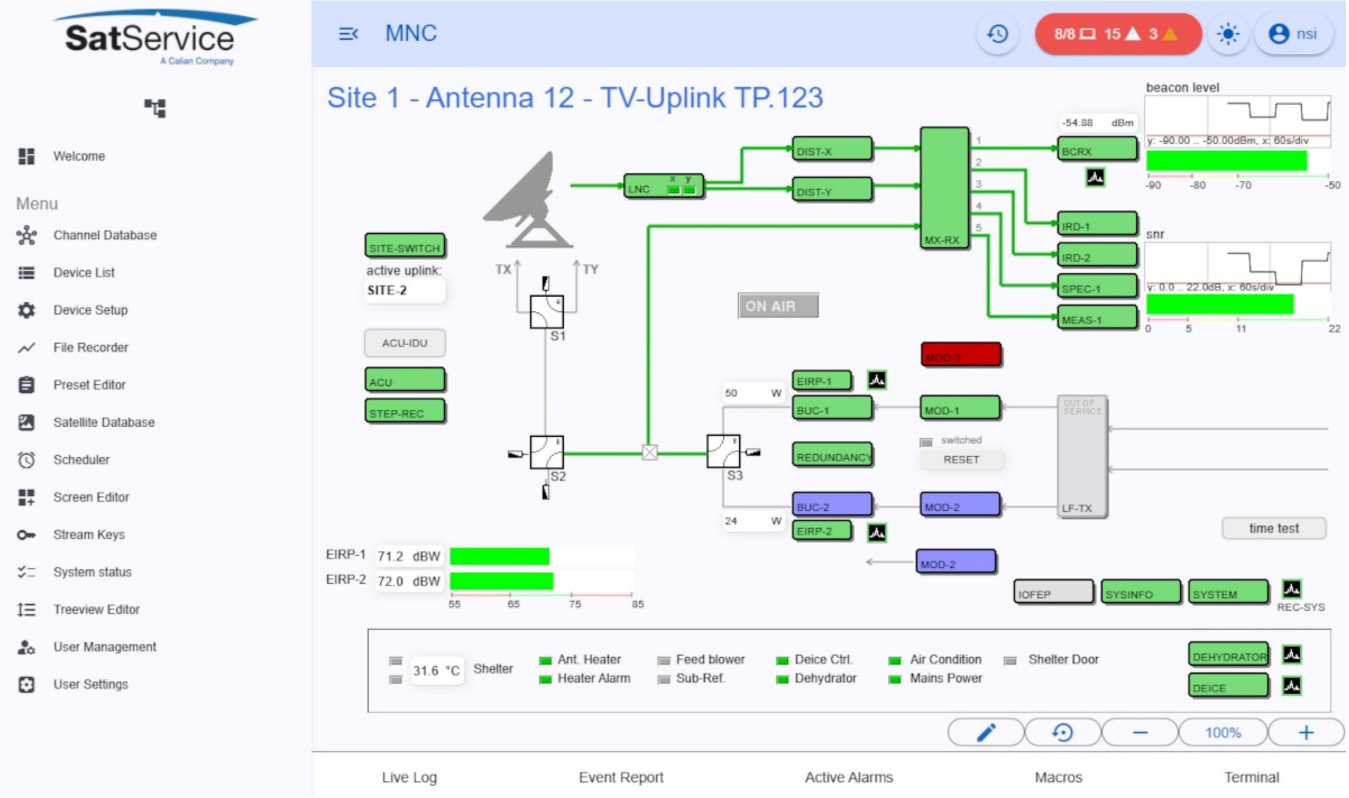
1.2 Architecture

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

TODO

2 M&C web-client

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

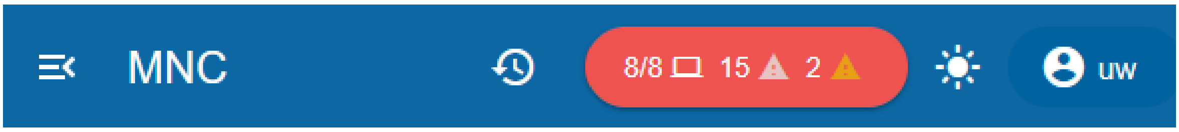


Quick access links to the main sections of the document:

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

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

2.1 Top toolbar



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

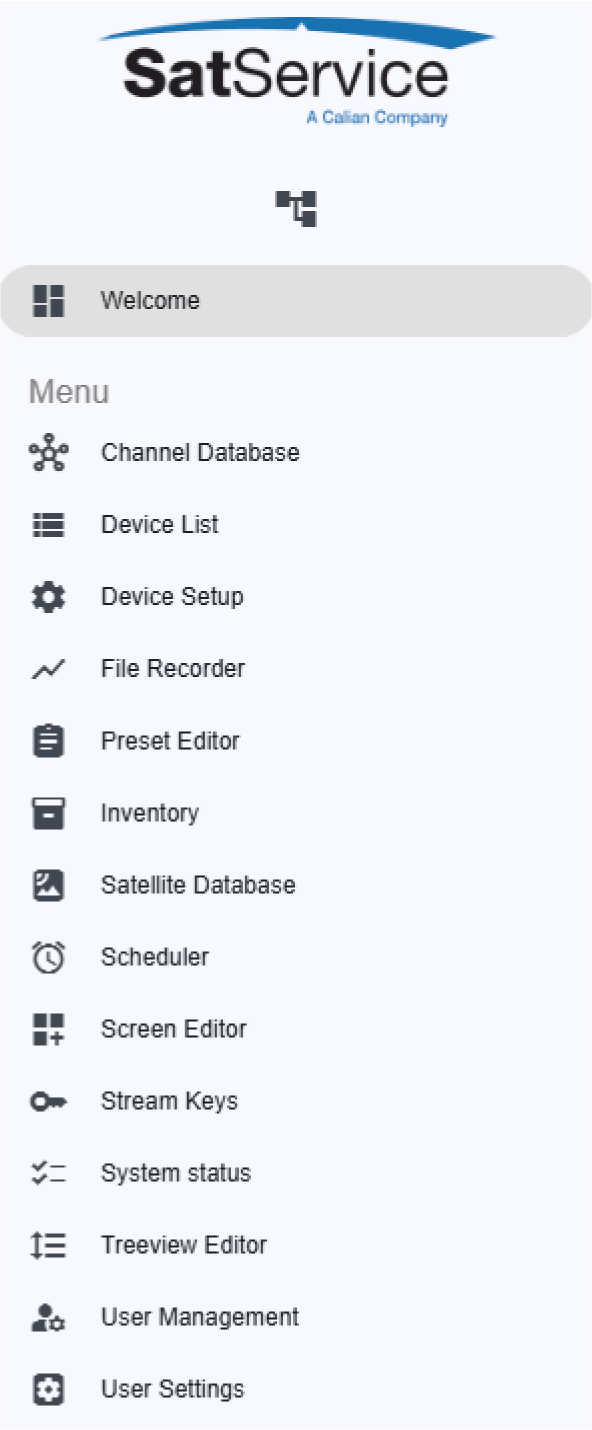
The following functions are available by clicking on the elements:

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

2.1.1 Sidebar / main menu

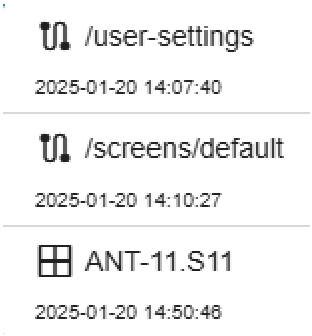
The side bar provide shortcuts to the different functions of the***sat-nms*** MNC system. It a kind of menu menu which gives you quick access to key areas and can be find on the left hand side. All menus are explained in the separates chapters.

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



2.1.2 History

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



/user-settings
2025-01-20 14:07:40

/screens/default
2025-01-20 14:10:27

ANT-11.S11
2025-01-20 14:50:46

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

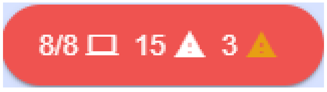
icon	description
⇅	jump to a page or user screen (URL)
⊞	opens specified device window

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

2.1.3 Notification area

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

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



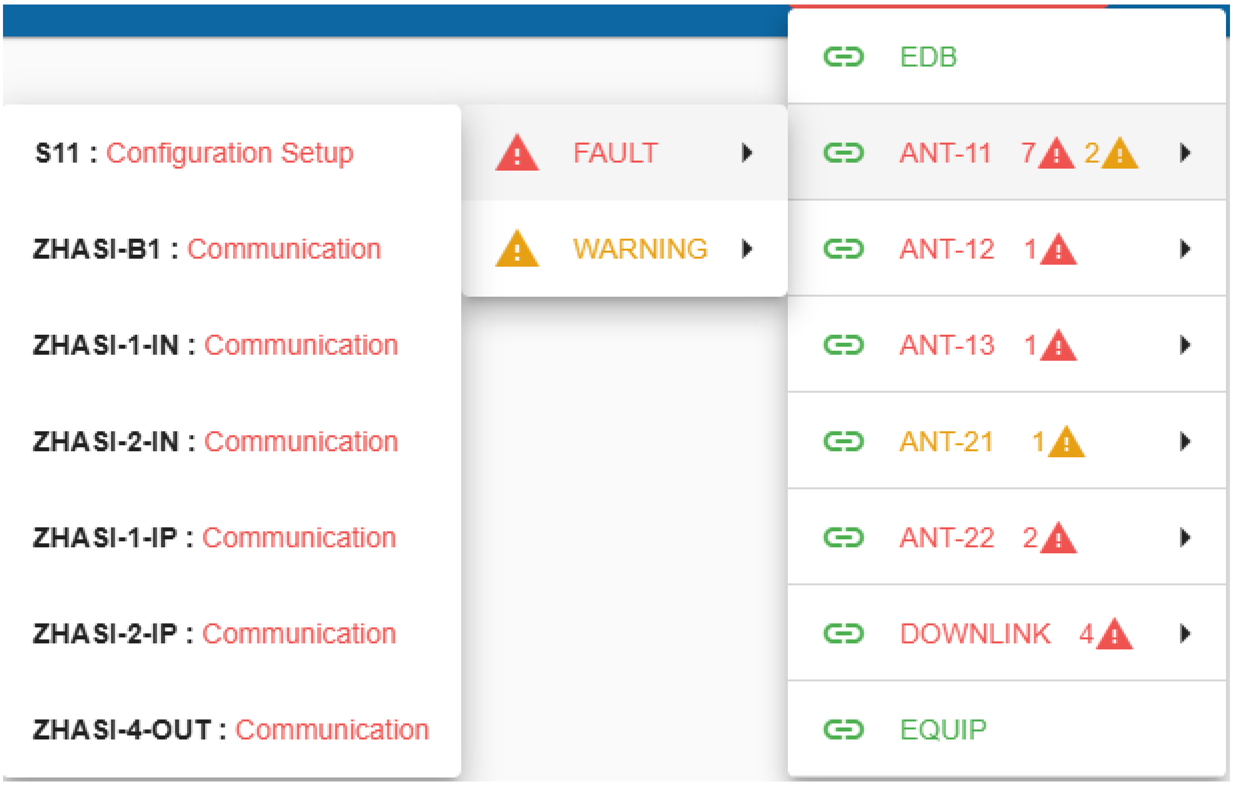
8/8 □ 15 ⚠ 3 ⚠

The number inside the notification represents the following information:

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

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

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


S11 : Configuration Setup
ZHASI-B1 : Communication
ZHASI-1-IN : Communication
ZHASI-2-IN : Communication
ZHASI-1-IP : Communication
ZHASI-2-IP : Communication
ZHASI-4-OUT : Communication
FAULT
WARNING
EDB
ANT-11 7 2
ANT-12 1
ANT-13 1
ANT-21 1
ANT-22 2
DOWNLINK 4
EQUIP

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

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

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

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

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

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

2.1.4 Theme

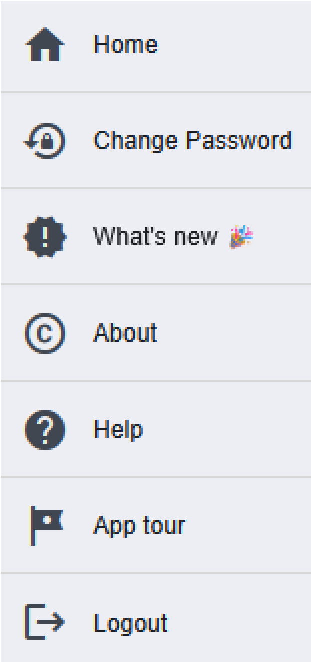
This icon shows current theme of this app. On clicking shows the following options:

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

2.1.5 User menu

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

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



2.2 Main Content Window

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

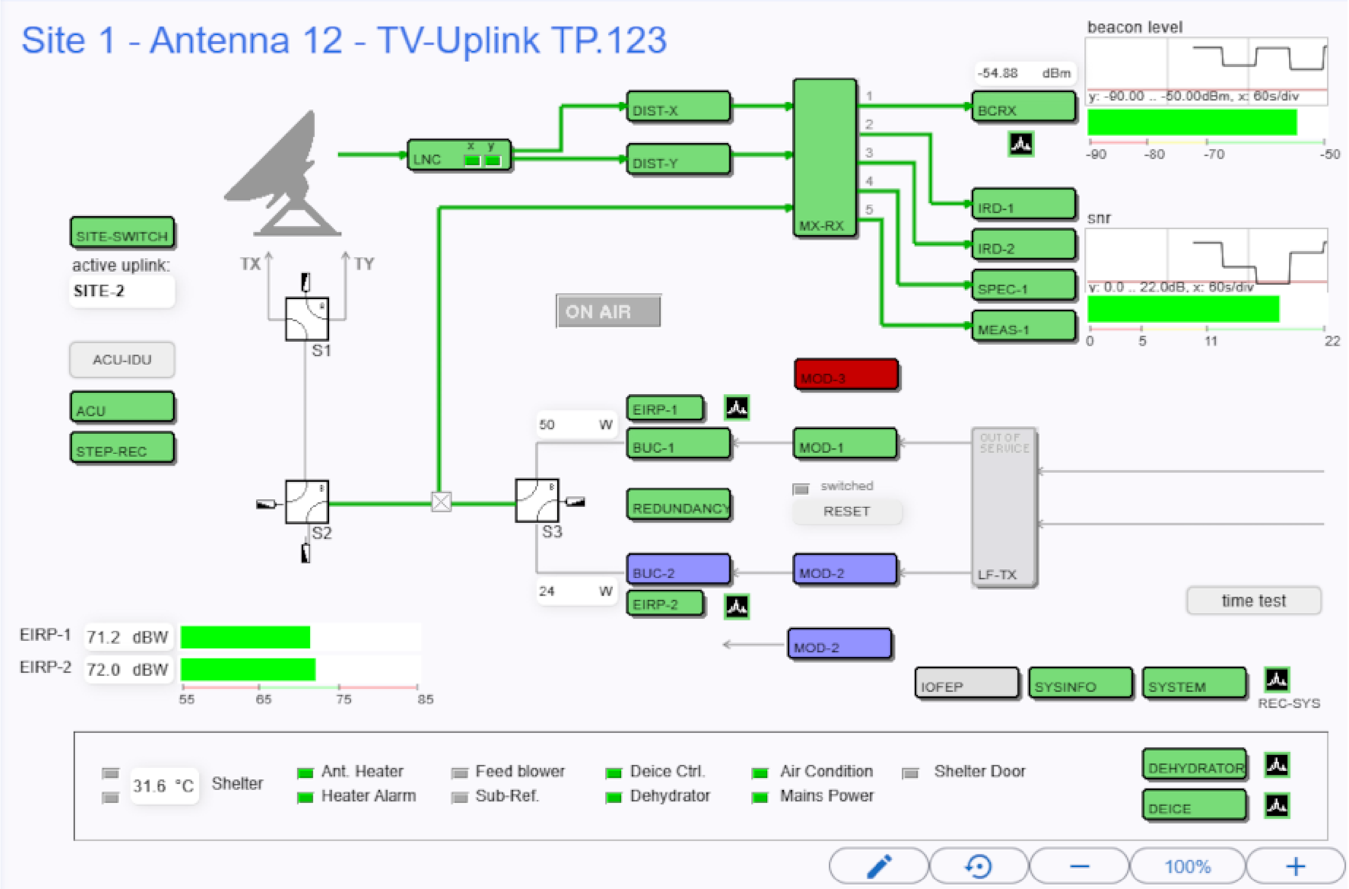
- The Main Window adapts dynamically based on the selected page or action.
- It works together with the top toolbar, the sidebar and the bottom toolbar to ensure a seamless user experience.
- Users can navigate between different sections using the sidebar menu or other navigation controls, and the content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

Following content types are shown in the Main Window:

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

2.2.1 User screens

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



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

Function	Icon	Description
edit screen	✎	open the screen editor with the active user screen
Reset position	⟲	resets the position of the user screen
Zoom out	−	zoom out, scale down the elements inside of the page
Current zoom	100%	display the current zoom of the user screen. Clicking on it works like a button and resets the zoom to the default setting.
Zoom in	+	zoom in, scale up the elements inside of the page

User screens shortcuts

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

2.2.2 Display elements

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

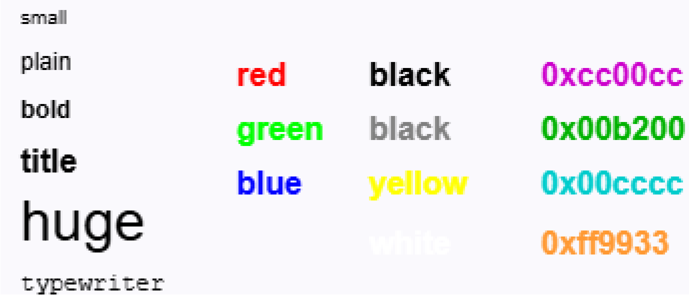
Category	Element	Readonly	Interaction	Dynamic update	Update data
Static	Text	□	□	□	□
	Frame	□	□	□	□
Dynamic non-interactive	Display (Readonly)	□	□	□	□
	Icon	□	□	□	□

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

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

2.2.2.1 Text element

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



small

plain

bold

title

huge

typewriter

red black 0xcc00cc

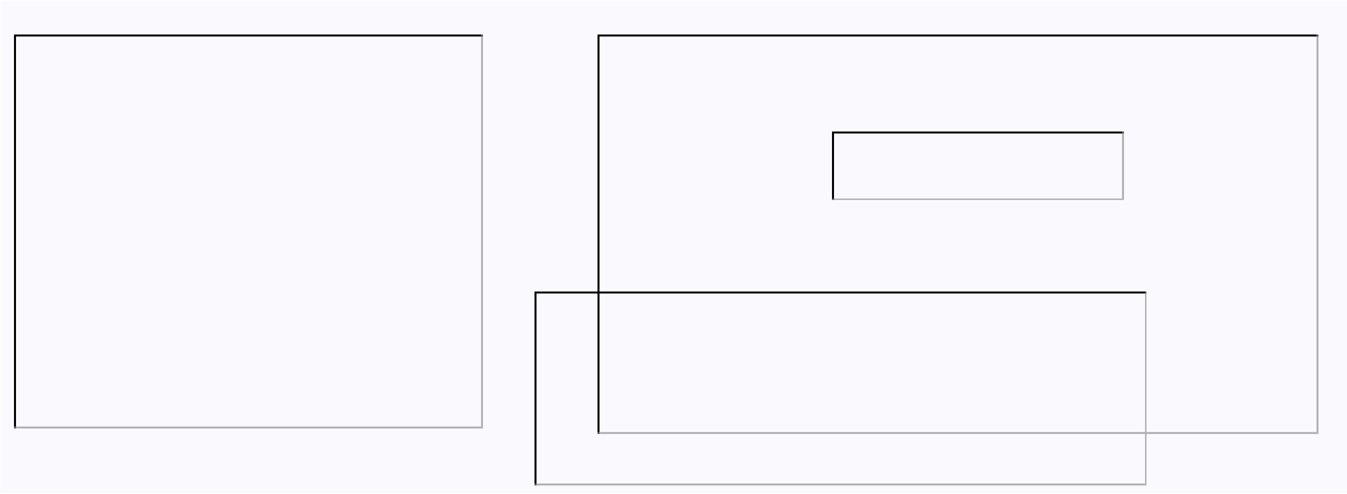
green black 0x00b200

blue yellow 0x00cccc

white 0xff9933

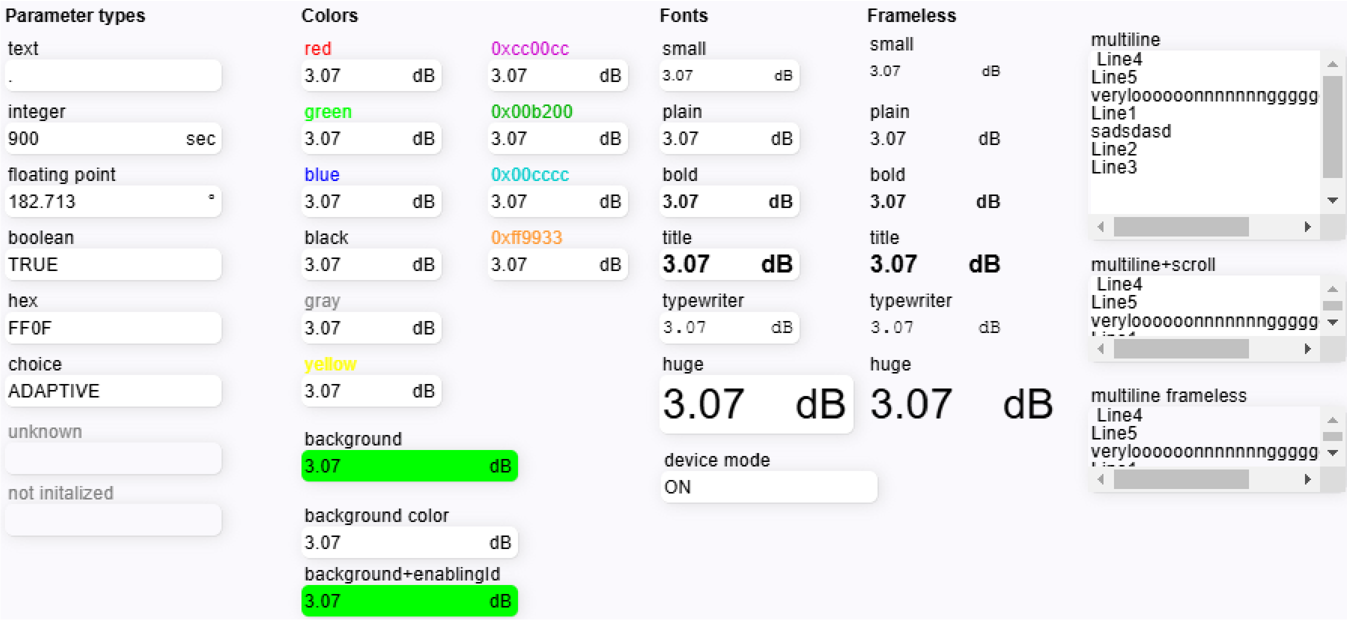
2.2.2.2 Frame element

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



2.2.2.3 Display element

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



Parameter types

text
.

integer
900 sec

floating point
182.713 °

boolean
TRUE

hex
FF0F

choice
ADAPTIVE

unknown

not initalized

Colors

red
3.07 dB

green
3.07 dB

blue
3.07 dB

black
3.07 dB

gray
3.07 dB

yellow
3.07 dB

background
3.07 dB

background color
3.07 dB

background+enablingId
3.07 dB

0xcc00cc
3.07 dB

0x00b200
3.07 dB

0x00cccc
3.07 dB

0xff9933
3.07 dB

Fonts

small
3.07 dB

plain
3.07 dB

bold
3.07 dB

title
3.07 dB

typewriter
3.07 dB

huge
3.07 dB

device mode
ON

Frameless

small
3.07 dB

plain
3.07 dB

bold
3.07 dB

title
3.07 dB

typewriter
3.07 dB

huge
3.07 dB

multiline
Line4
Line5
veryloooooonnnnnnnggggg
Line1
sadsdasd
Line2
Line3

multiline+scroll
Line4
Line5
veryloooooonnnnnnnggggg

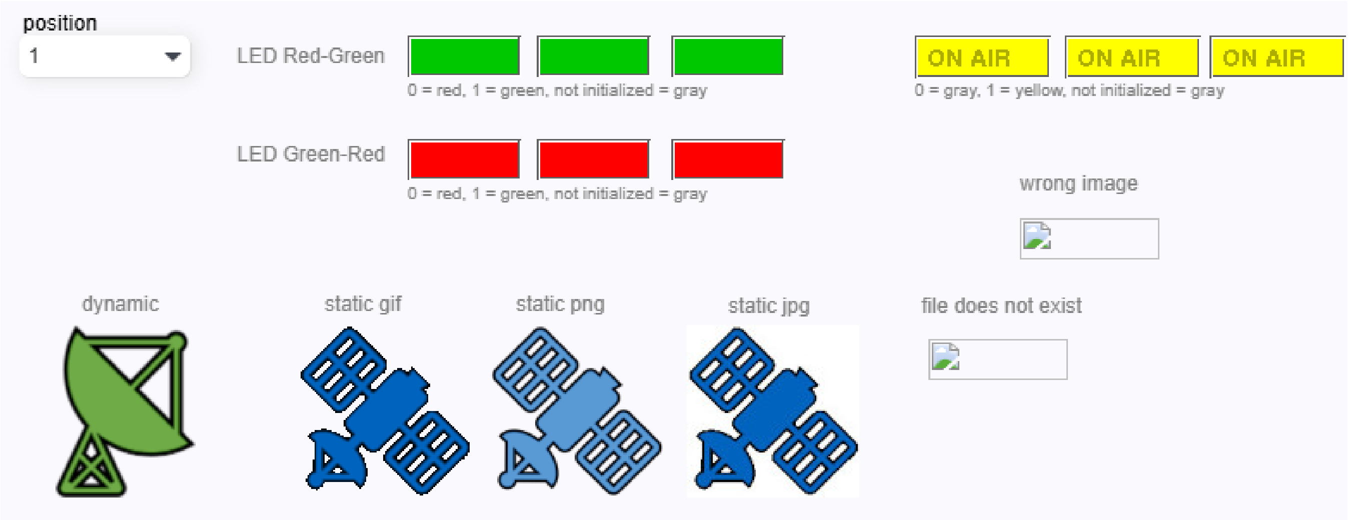
multiline frameless
Line4
Line5
veryloooooonnnnnnnggggg

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

2.2.2.4 Icon element

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

The images below shows static and dynamic icons.



position
1

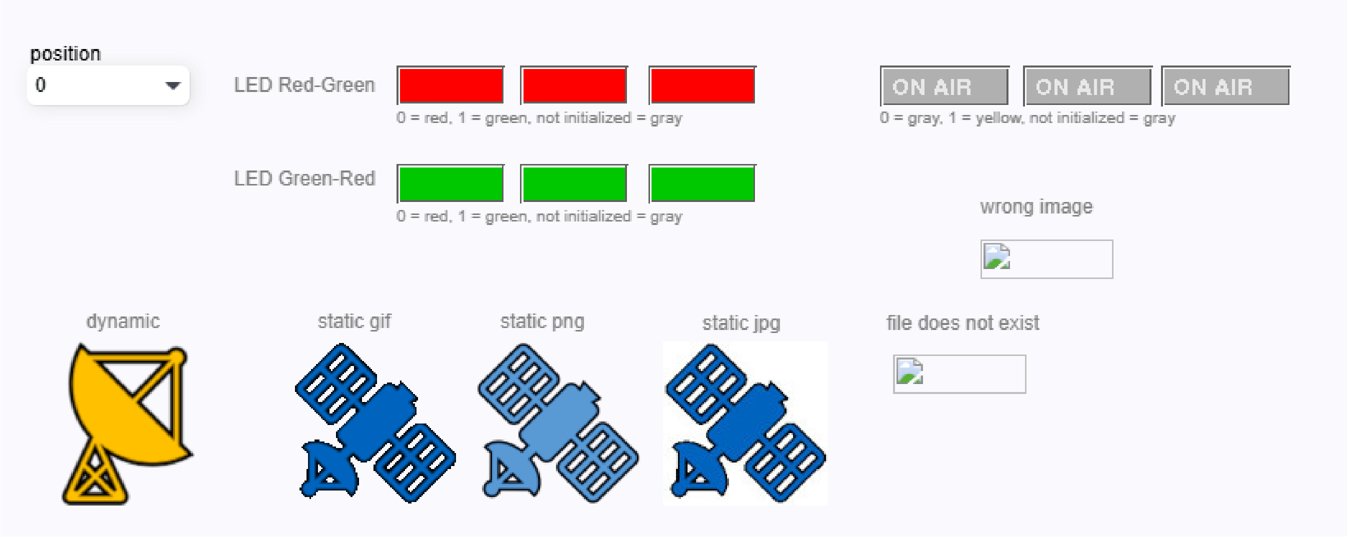
LED Red-Green
0 = red, 1 = green, not initialized = gray

LED Green-Red
0 = red, 1 = green, not initialized = gray

0 = gray, 1 = yellow, not initialized = gray

wrong image

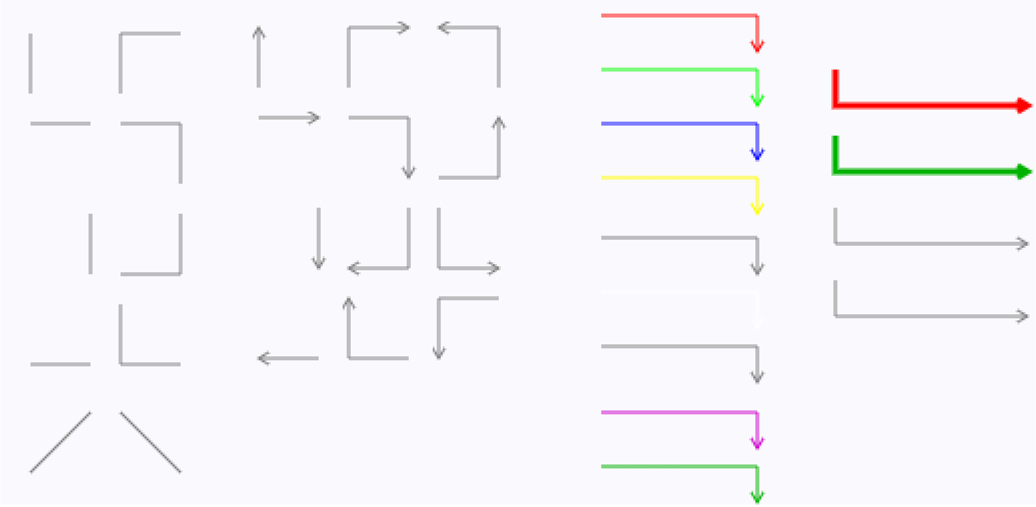
Dynamic icons are bind with position parameter element.


position
0
LED Red-Green
0 = red, 1 = green, not initialized = gray
LED Green-Red
0 = red, 1 = green, not initialized = gray
ON AIR
ON AIR
ON AIR
0 = gray, 1 = yellow, not initialized = gray
wrong image
dynamic
static gif
static png
static jpg
file does not exist

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

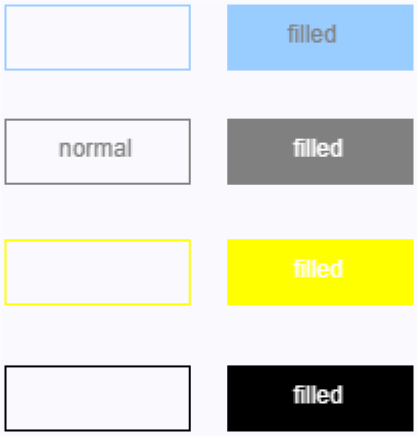
2.2.2.5 Arrow element

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



2.2.2.6 Rect element

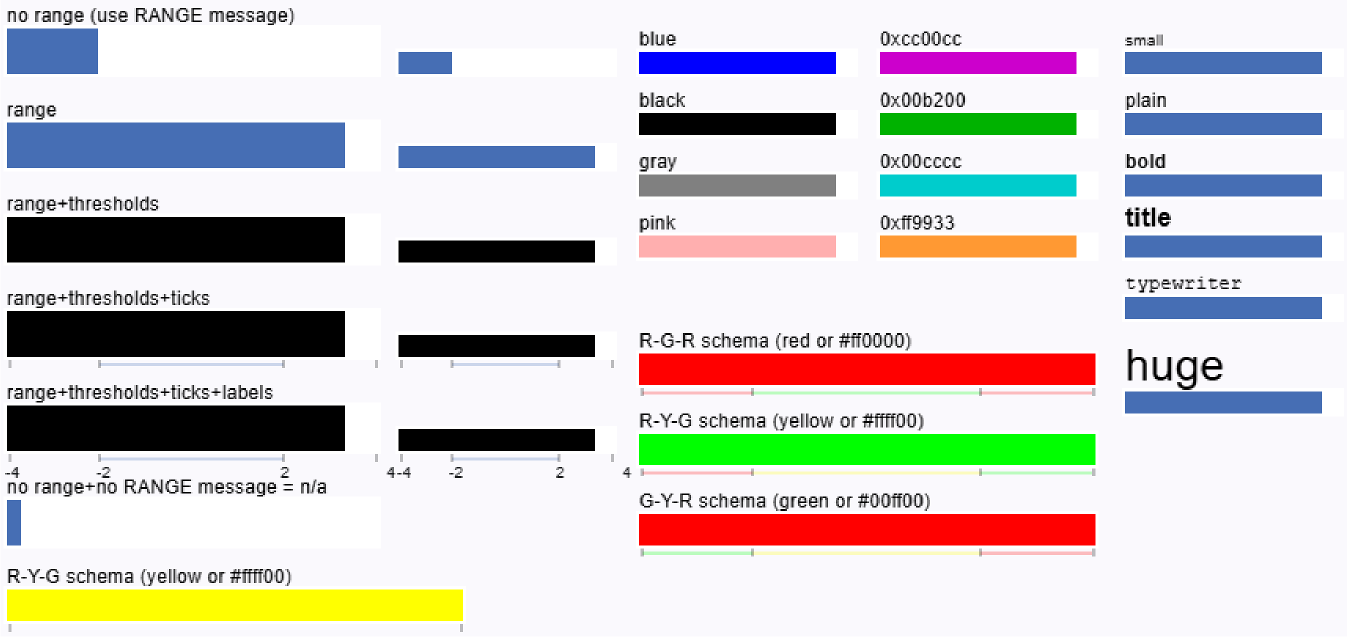
The Arrow element work as a dynamic non-interactive element and also as a static element. It draws a rectangle with a selectable color. The element may listen to a parameter value and change the color of the rectangle according to the parameter value.



2.2.2.7 Gauge element

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

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

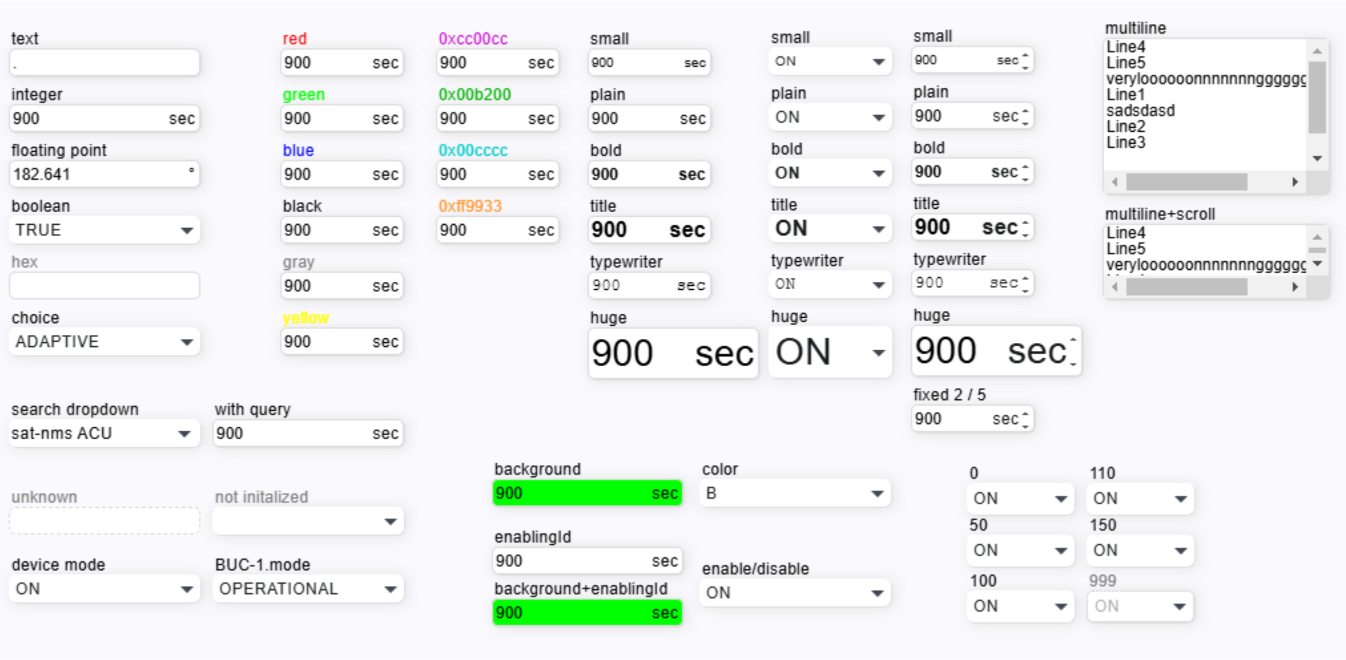

no range (use RANGE message)
range
range+thresholds
range+thresholds+ticks
range+thresholds+ticks+labels
-4
-2
2
4-4
-2
2
4
no range+no RANGE message = n/a
R-Y-G schema (yellow or #ffff00)
blue
black
gray
pink
0xcc00cc
0x00b200
0x00cccc
0xff9933
R-G-R schema (red or #ff0000)
R-Y-G schema (yellow or #ffff00)
G-Y-R schema (green or #00ff00)
small
plain
bold
title
typewriter
huge

2.2.2.8 Thumbnail element

2.2.2.9 Parameter element

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

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



text .
integer 900 sec
floating point 182.641
boolean TRUE
hex
choice ADAPTIVE
red 900 sec
green 900 sec
blue 900 sec
black 900 sec
gray 900 sec
yellow 900 sec
0xcc00cc 900 sec
0x00b200 900 sec
0x00cccc 900 sec
0xff9933 900 sec
small 900 sec
plain 900 sec
bold 900 sec
title 900 sec
typewriter 900 sec
huge 900 sec
small ON
plain ON
bold ON
title ON
typewriter ON
huge ON
small 900 sec
plain 900 sec
bold 900 sec
title 900 sec
typewriter 900 sec
huge 900 sec
fixed 2 / 5 900 sec
multiline Line4 Line5 veryloooooonnnnnnngggggg Line1 sadsdasd Line2 Line3
multiline+scroll Line4 Line5 veryloooooonnnnnnngggggg
search dropdown sat-nms ACU
with query 900 sec
unknown
not initalized
device mode ON
BUC-1.mode OPERATIONAL
background 900 sec
enablingId 900 sec
background+enablingId 900 sec
color B
enable/disable ON
0 ON
50 ON
100 ON
110 ON
150 ON
999 ON

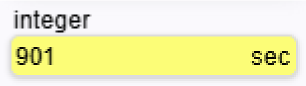
Widget Variants: Depending on the type of parameter and on the*height* parameter there are different widget types associated with the ParameterElement. As the parameter type is not known at the time the screen definition is read, the decision what widget type shall be used must be done after the parameter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

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

Spin Buttons: With *useSpinButtons* set 'true', the ParameterElement shows spin buttons with editable numeric values. The spin buttons allow to increment/decrement the value by clicking on them. By default, a spin button click increments or decrements the lowest significant digit shown in the entry field. With the shift key hold down, the effective increment is x10. The above applies if the *spinSmallIncrement* / *spinLargeIncrement* properties are empty. If set, these values override the defaults.

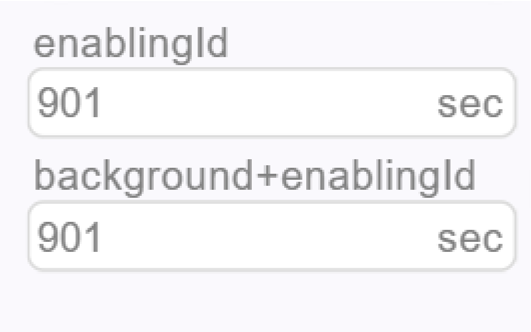
Unsaved changes

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



integer
901 sec

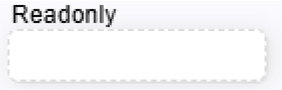
Element disabled



enablingId
901 sec
background+enablingId
901 sec

Element readonly

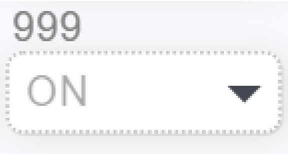
Readonly elements is displayed with dashed border with grey text color.



Readonly

Element disabled because of low privilege level

Elements is displayed with dotted border with grey text color.



999
ON

2.2.2.10 Parameter button element

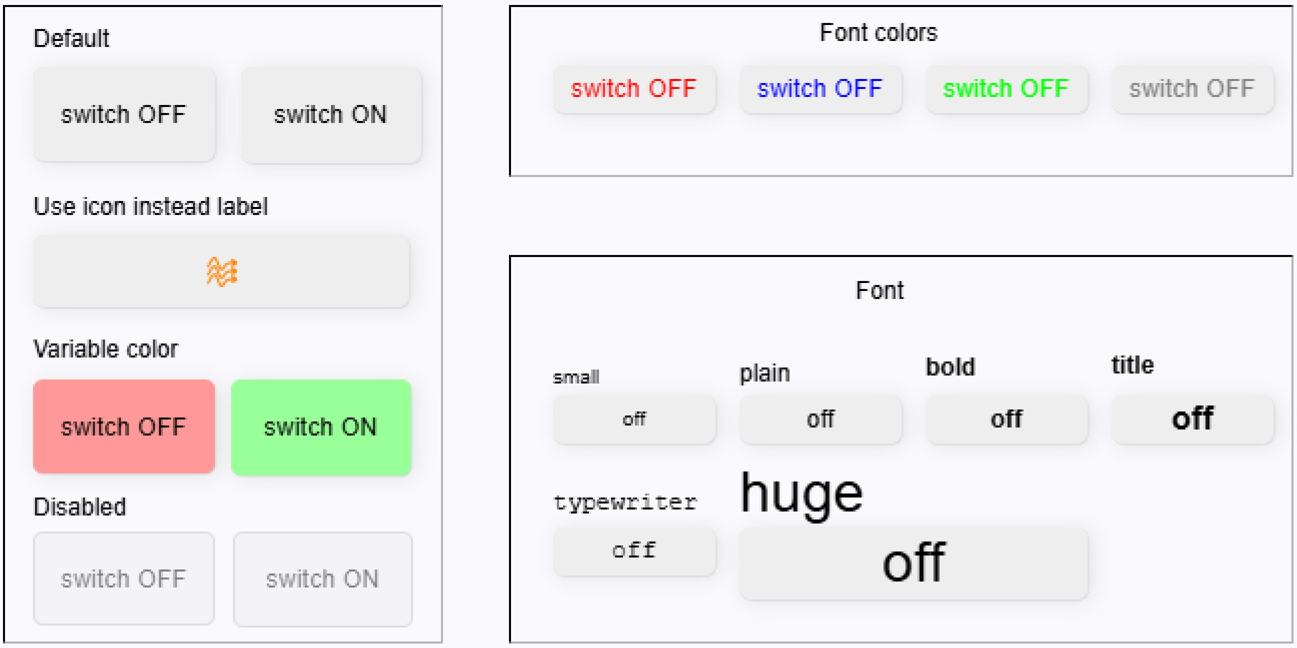
The parameter button element is an interactive element and a button which sends a parameter value if pressed. A frequently used application for the parameter button is a RF-OFF button which sends a "tx.on=OFF" to a certain device. Beside this, a parameter button also may be programmed to play a parameter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

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

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

Macro Buttons

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



2.2.2.11 Latching button element

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

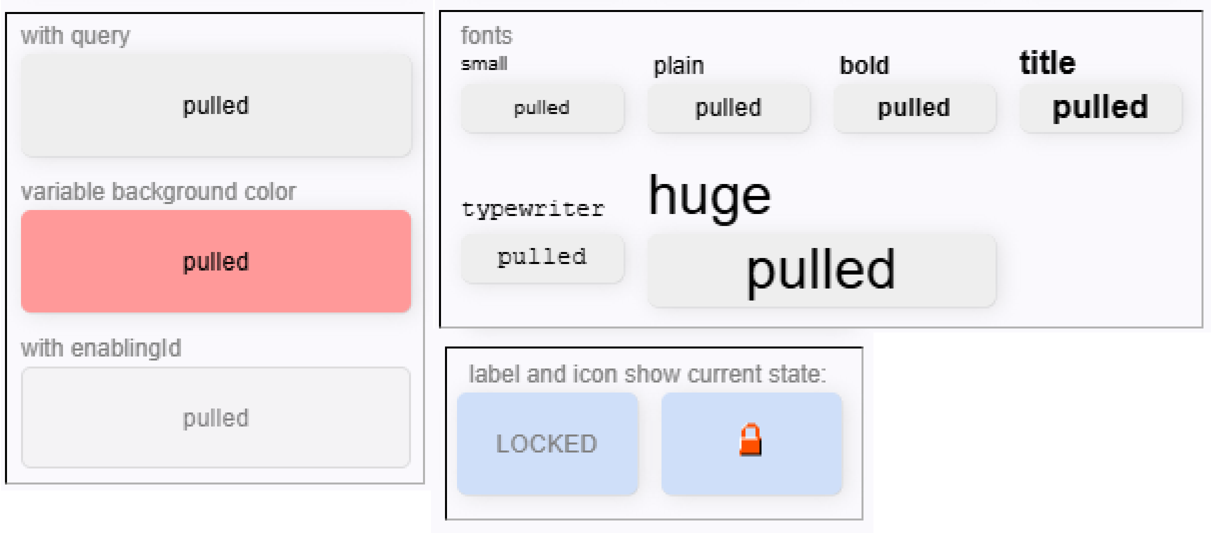
When the operator changes the state of the button by clicking it once, the latching button sends the 'other' parameter value to the device. On the other hand, if some other instance in the system changes the parameter state, the latching button recognizes this and changes the up/down state of the button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

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

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

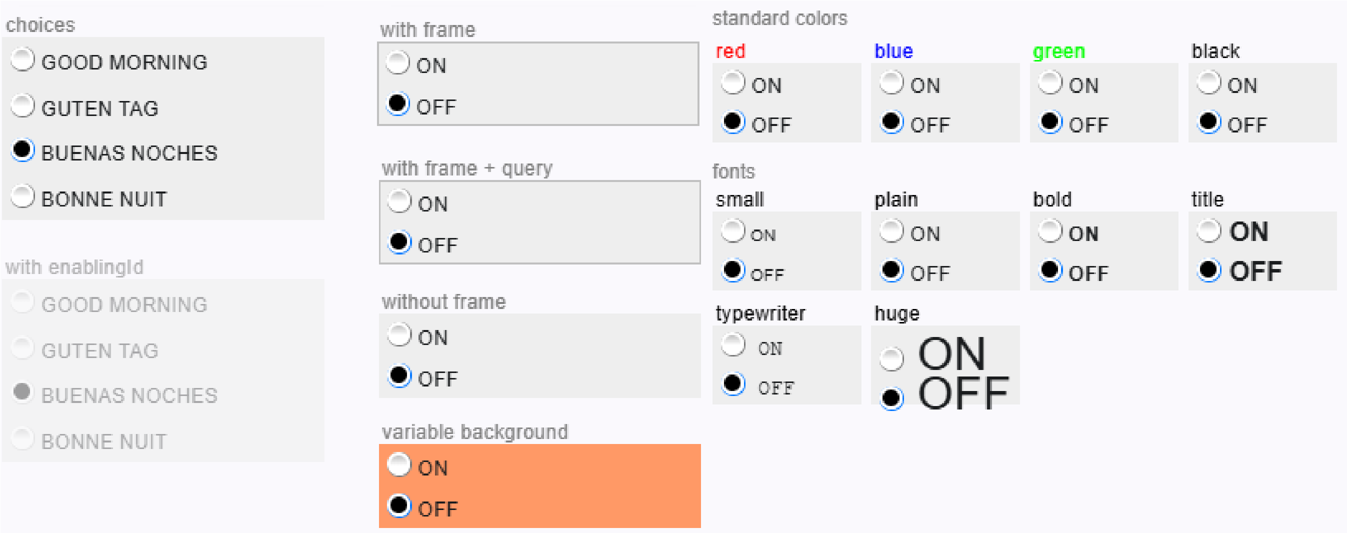
The valueUp and valueDown values are not only sent when the button is pressed or released.

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



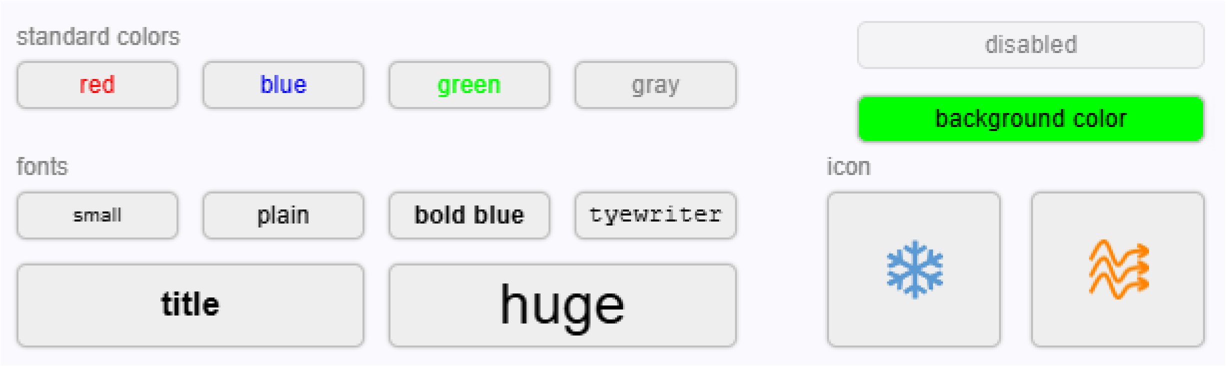
2.2.2.12 Radio button element

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



2.2.2.13 Button element

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



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

CHILD-SCREEN

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

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

REPLACE-SCREEN

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

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

LOAD-PRESET

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

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

FREC-VIEW

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

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

SPECTR-VIEW

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

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

BROWSER-VIEW

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

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

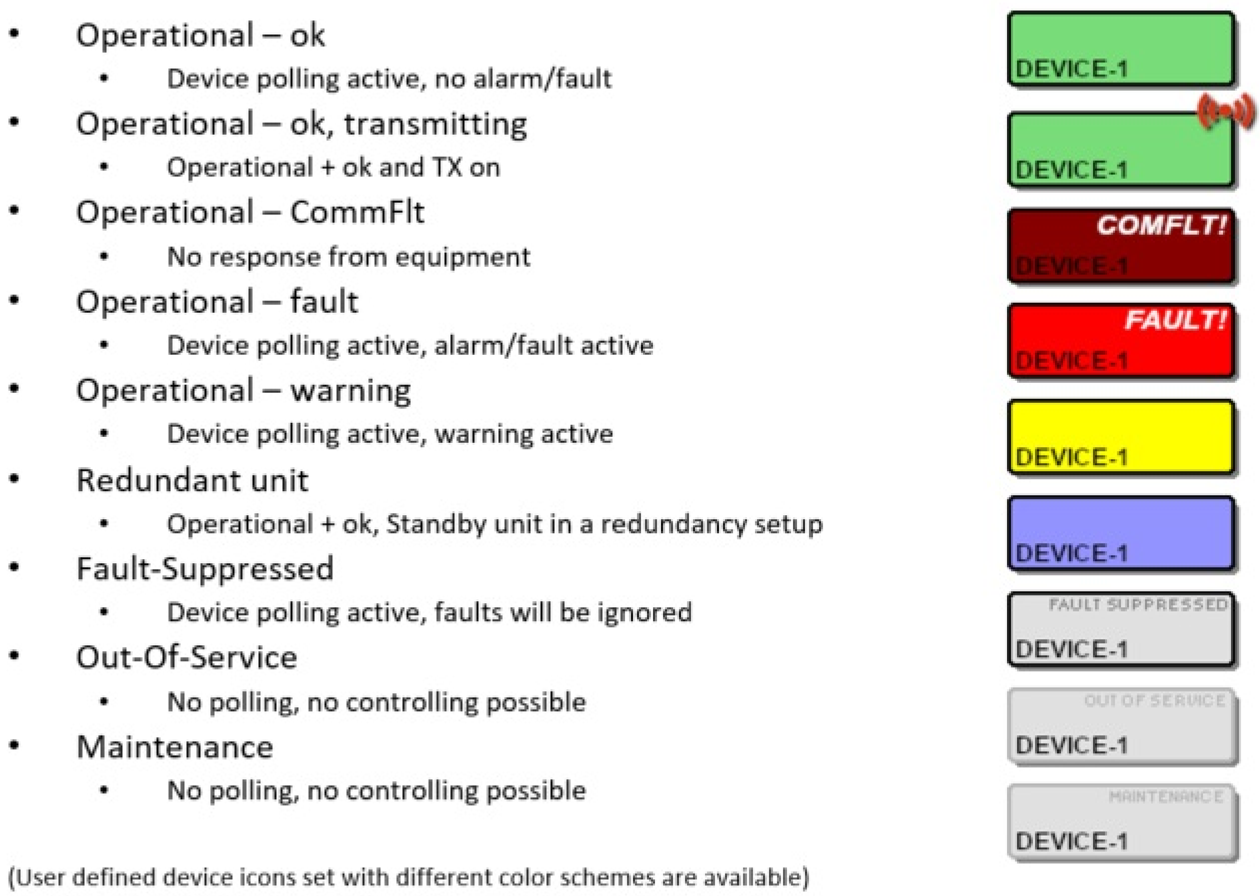
TREE-NAVIGATE

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

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

2.2.2.14 Device element

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

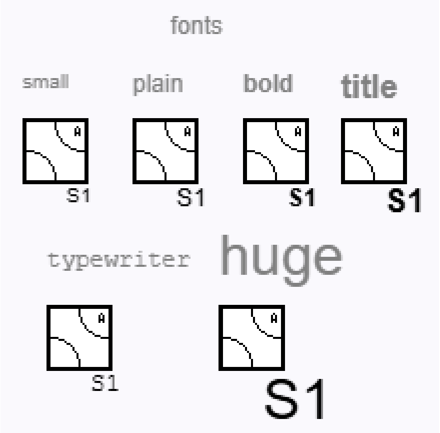


- Operational – ok
  - Device polling active, no alarm/fault
- Operational – ok, transmitting
  - Operational + ok and TX on
- Operational – CommFlt
  - No response from equipment
- Operational – fault
  - Device polling active, alarm/fault active
- Operational – warning
  - Device polling active, warning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

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

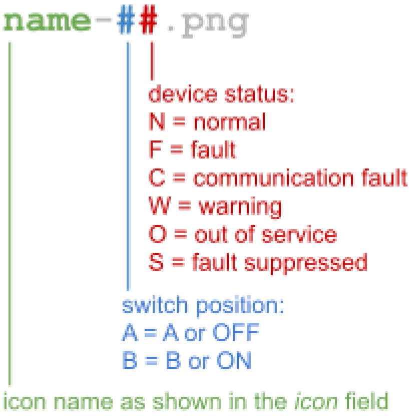
2.2.2.15 Switch element

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



Icon Selection

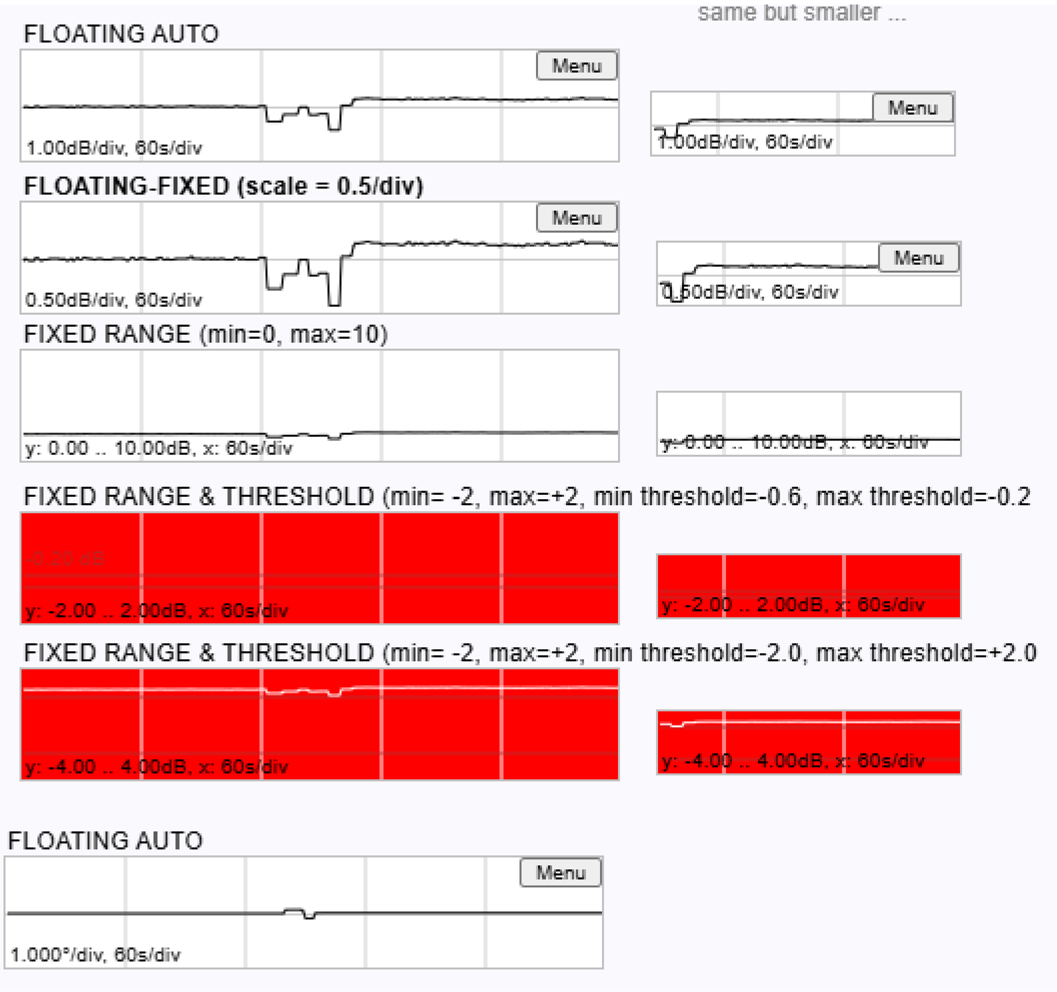
The switch element displays one icon of a given set depending on the switch position and the state of the switch device. For this the SwitchElement appends a '-' character, a 2-character status code and the suffix '.png' to the icon name given in *icon* in order to ge the file name to load / display:



2.2.2.16 Chart element

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

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


same but smaller ...
FLOATING AUTO
Menu
1.00dB/div, 60s/div
Menu
1.00dB/div, 60s/div
FLOATING-FIXED (scale = 0.5/div)
Menu
0.50dB/div, 60s/div
Menu
0.50dB/div, 60s/div
FIXED RANGE (min=0, max=10)
y: 0.00 .. 10.00dB, x: 60s/div
y: 0.00 .. 10.00dB, x: 60s/div
FIXED RANGE & THRESHOLD (min= -2, max=+2, min threshold=-0.6, max threshold=-0.2
y: -2.00 .. 2.00dB, x: 60s/div
y: -2.00 .. 2.00dB, x: 60s/div
FIXED RANGE & THRESHOLD (min= -2, max=+2, min threshold=-2.0, max threshold=+2.0
y: -4.00 .. 4.00dB, x: 60s/div
y: -4.00 .. 4.00dB, x: 60s/div
FLOATING AUTO
Menu
1.000°/div, 60s/div

Display Modes

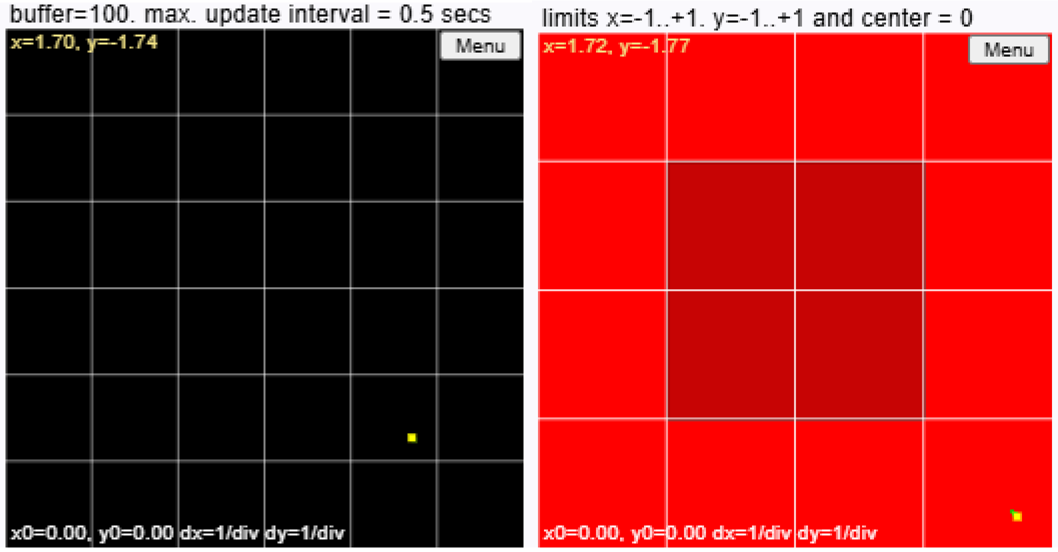
The ChartElement uses different display and scaling modes, depending on the display mode defined in *mode*. The table below lists the defined display modes an their behavior.

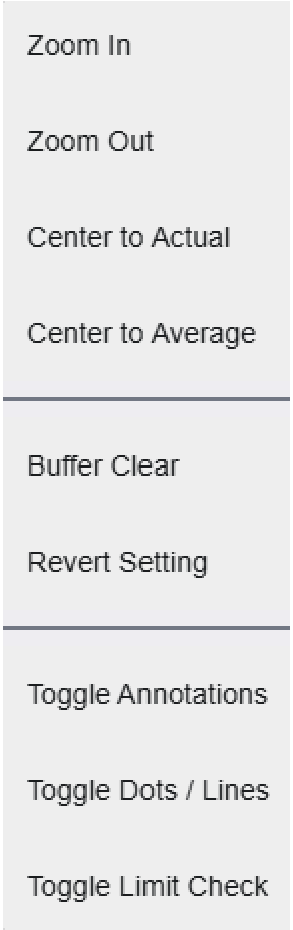
mode	**Description**
FLOATING-AUTO	Does a full autoscale. The y-offset of the strip chart is set that the newest value displayed appers at the middle of the y axis. The chart y-scale is evaluated in a 1-2-5 raster to the finest scale that allows all points in the history to be displayed in the chart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

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

2.2.2.17 XY chart element

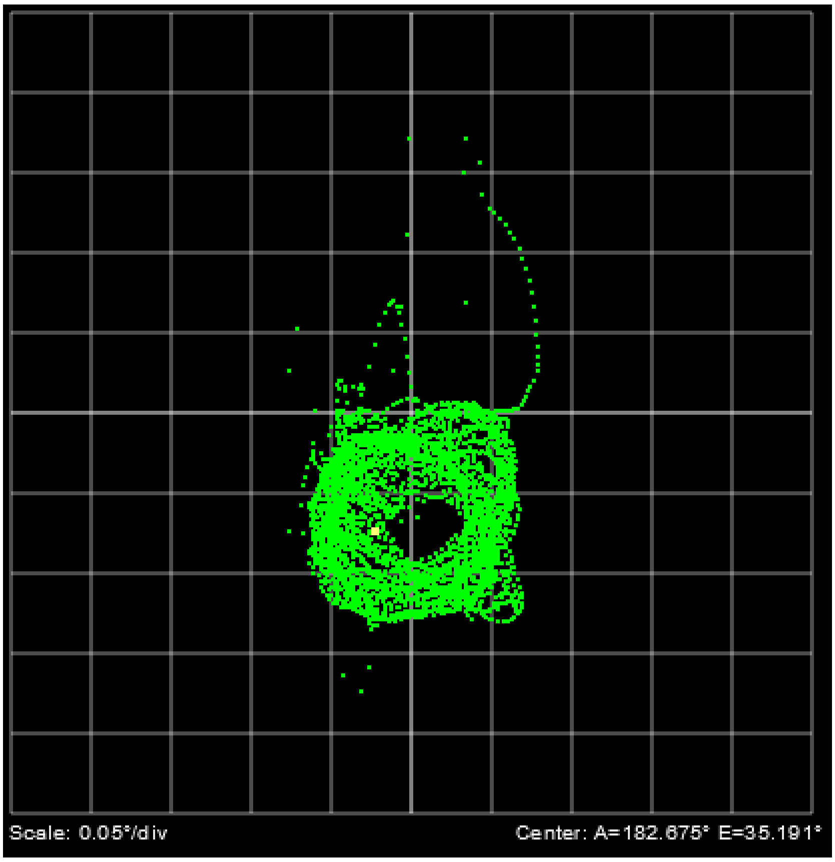
This element shows the relation of two numeric variables in an X/Y diagram, featuring a 'trace' which shows the recent history of the values with a configurable depth. The update rate, the diagram scaling and much more is configurable with this screen element.

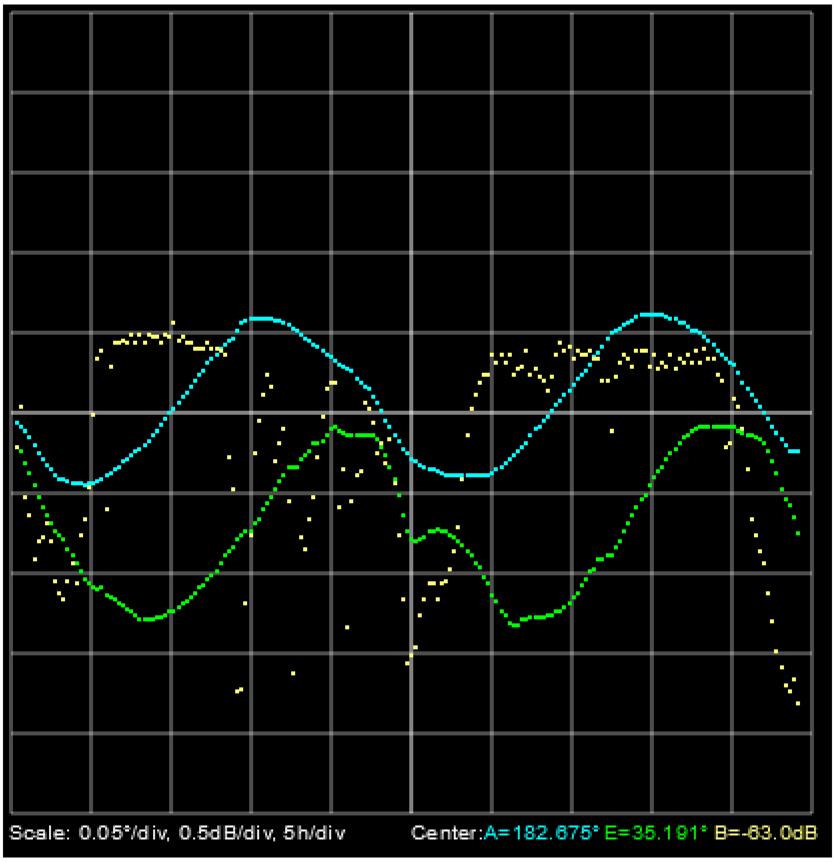

buffer=100. max. update interval = 0.5 secs
x=1.70, y=-1.74
Menu
x0=0.00, y0=0.00 dx=1/div dy=1/div
limits x=-1..+1. y=-1..+1 and center = 0
x=1.72, y=-1.77
Menu
x0=0.00, y0=0.00 dx=1/div dy=1/div



2.2.2.18 AzEl element

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

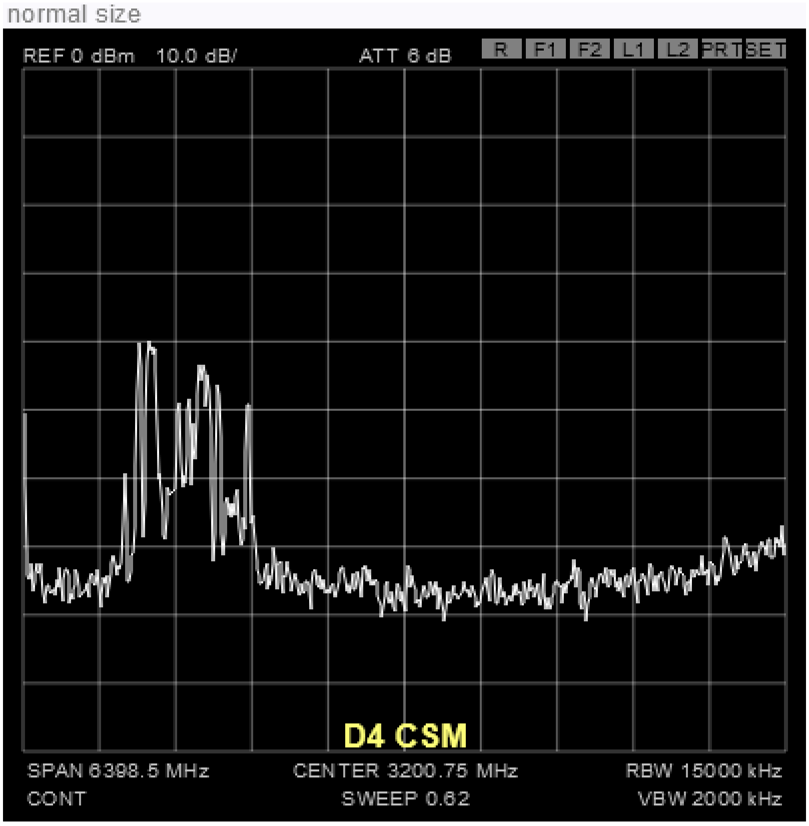

Scale: 0.05°/div
Center: A=182.675° E=35.191°


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

2.2.2.19 Spectrum element

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

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



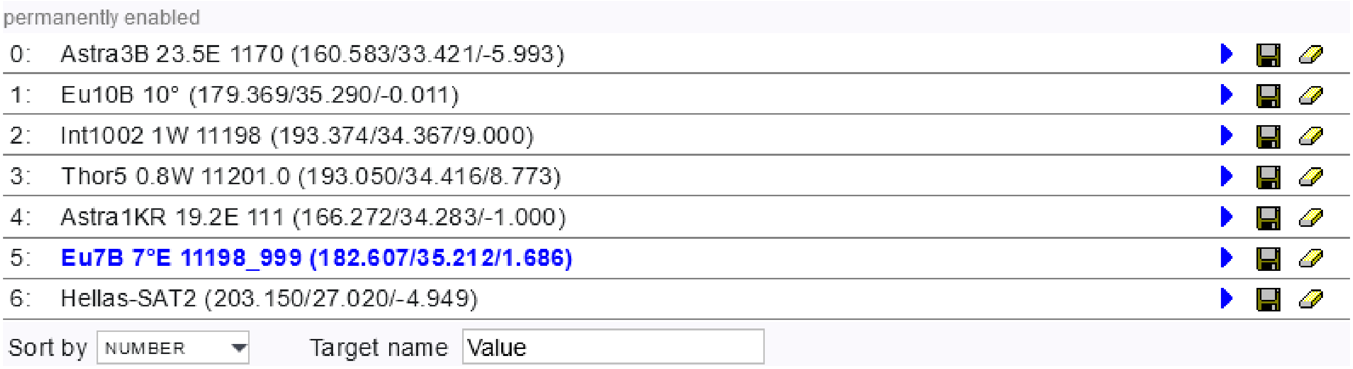
REF 0 dBm 10.0 dB/ ATT 6 dB

SPAN 6398.5 MHz CENTER 3200.75 MHz RBW 15000 kHz

CONT SWEEP 0.62 VBW 2000 kHz

2.2.2.20 Target list element

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



permanently enabled

0: Astra3B 23.5E 1170 (160.583/33.421/-5.993)

1: Eu10B 10° (179.369/35.290/-0.011)

2: Int1002 1W 11198 (193.374/34.367/9.000)

3: Thor5 0.8W 11201.0 (193.050/34.416/8.773)

4: Astra1KR 19.2E 111 (166.272/34.283/-1.000)

5: **Eu7B 7°E 11198_999 (182.607/35.212/1.686)**

6: Hellas-SAT2 (203.150/27.020/-4.949)

Sort by NUMBER ▾ Target name Value

Device Name Indirection

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

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

Target List Parsing

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

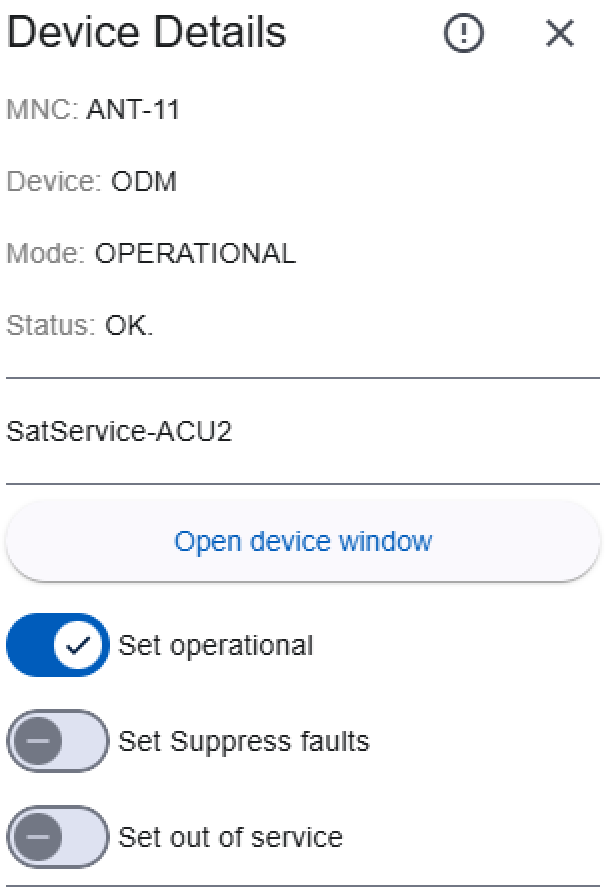
Each target definition contains the target name and the azimuth / elevation / polarization angles. The target number is defined implicitly from the position of the target in the complete string. The format of each target definition is as follows:

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

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

2.2.3 Device details panel

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



Device Details

MNC: ANT-11

Device: ODM

Mode: OPERATIONAL

Status: OK.

SatService-ACU2

This panel will open if:

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

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

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

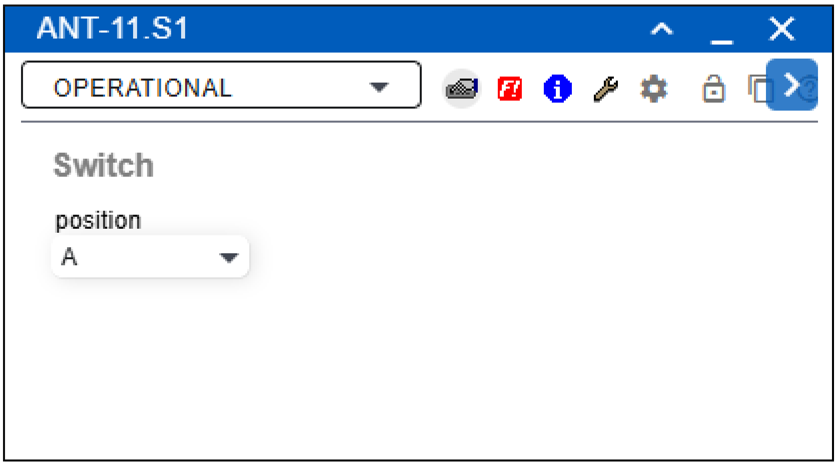
Details

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

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

2.3 Device window

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



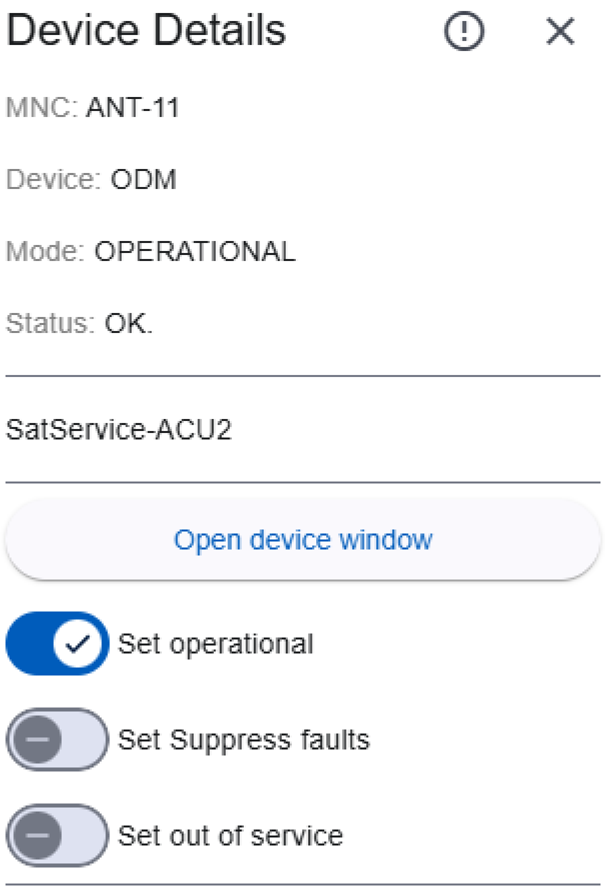
Functionality

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

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

2.3.1 Device details panel

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



Device Details

MNC: ANT-11

Device: ODM

Mode: OPERATIONAL

Status: OK.

SatService-ACU2

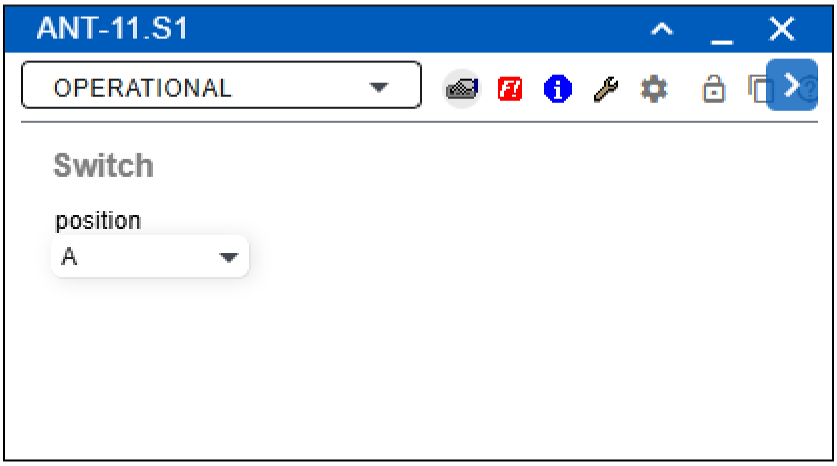
Top

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

Details

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

2.3.2 Device window



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

Header

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

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

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

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

Device locking

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

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

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

Content

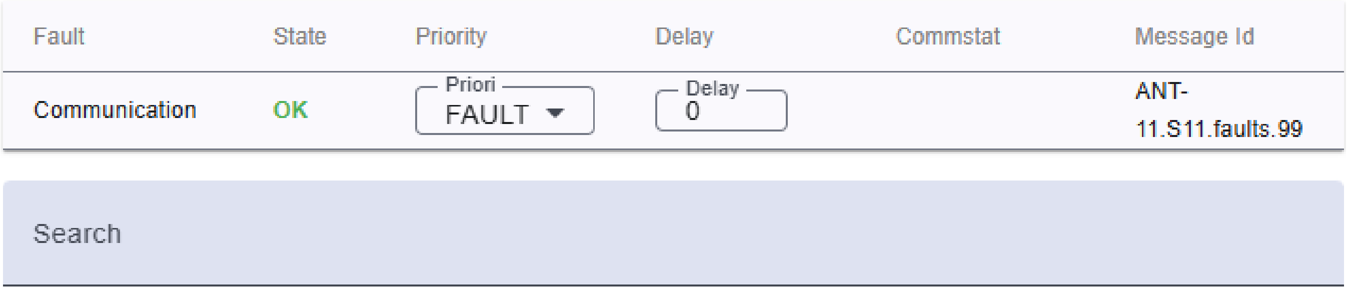
Writable parameters are shown below the toolbar, read only parameters appear with the standard gray background of the window. Choice parameters are changed immediately when you alter the selection. If you edit a text parameter, the field is shown with a yellow background. Once you press ENTER or you leave the field, the changed value gets transferred to the device. To cancel a parameter setting press ESC. The old value gets restored in the field and the background color returns to light gray. No parameter value is sent to the device in this case.

2.3.2.1 Device window pages

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

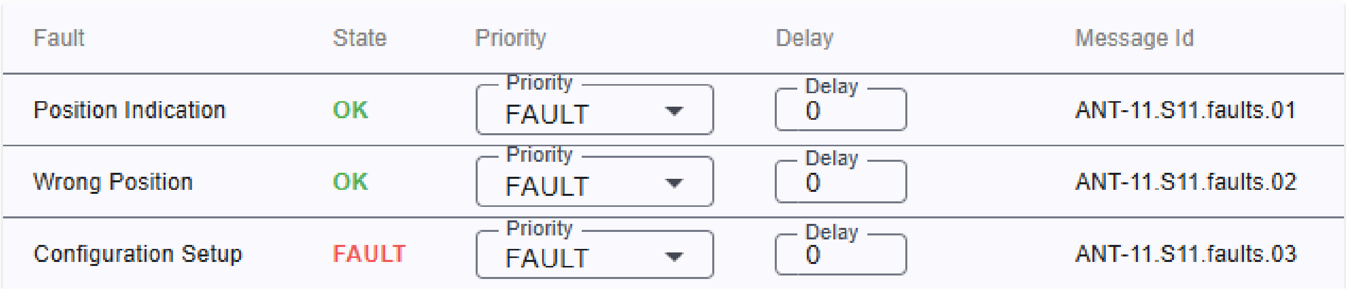
- Fault page
- Info page
- Maintenance page
- Config page

2.3.2.1.1 Fault page



Fault	State	Priority	Delay	Commstat	Message Id
Communication	OK	Priori FAULT	Delay 0		ANT-11.S11.faults.99

Search



Fault	State	Priority	Delay	Message Id
Position Indication	OK	Priority FAULT	Delay 0	ANT-11.S11.faults.01
Wrong Position	OK	Priority FAULT	Delay 0	ANT-11.S11.faults.02
Configuration Setup	FAULT	Priority FAULT	Delay 0	ANT-11.S11.faults.03

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

Header row

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

Table columns

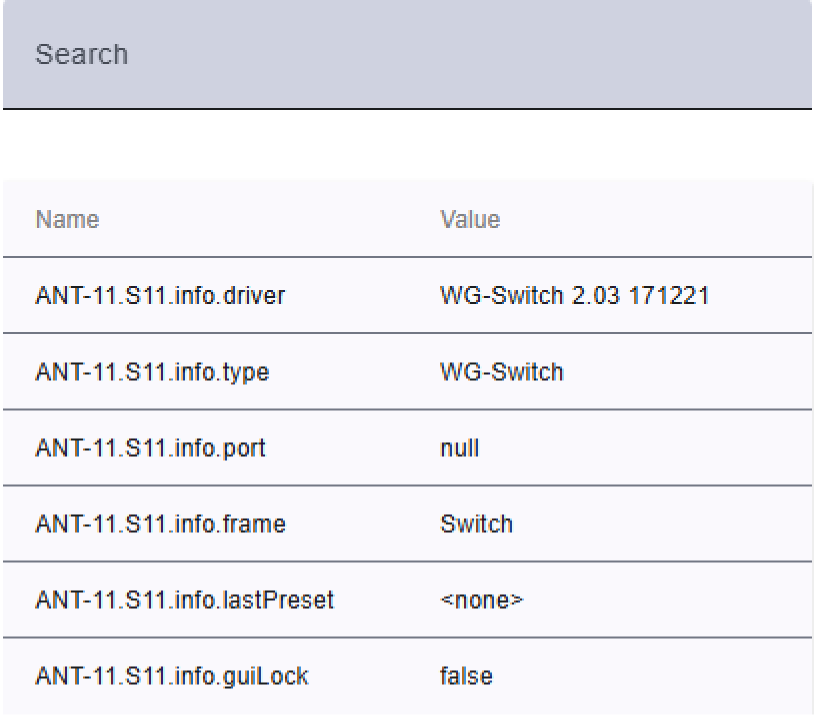
- **Fault**: The name of the fault.
- **State**: State or fault of device.
- **Priority**: Priority of the fault.
  - **OFF**: This flag is completely suppressed.

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

Search bar: Allows users to filter events by keywords.

2.3.2.1.2 Info page

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



Search

Name	Value
ANT-11.S11.info.driver	WG-Switch 2.03 171221
ANT-11.S11.info.type	WG-Switch
ANT-11.S11.info.port	null
ANT-11.S11.info.frame	Switch
ANT-11.S11.info.lastPreset	<none>
ANT-11.S11.info.guiLock	false

Search bar: Allows users to filter events by keywords.

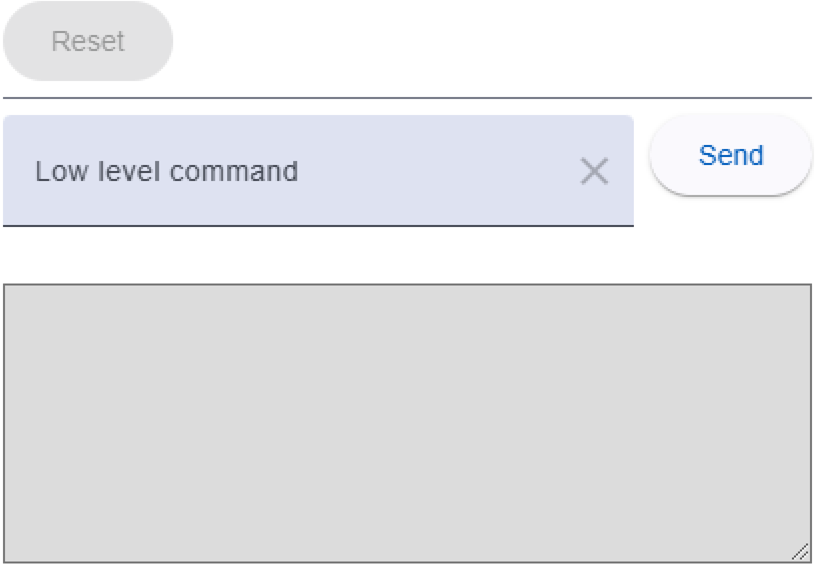
Header row

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

Table columns

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

2.3.2.1.3 Maintenance page



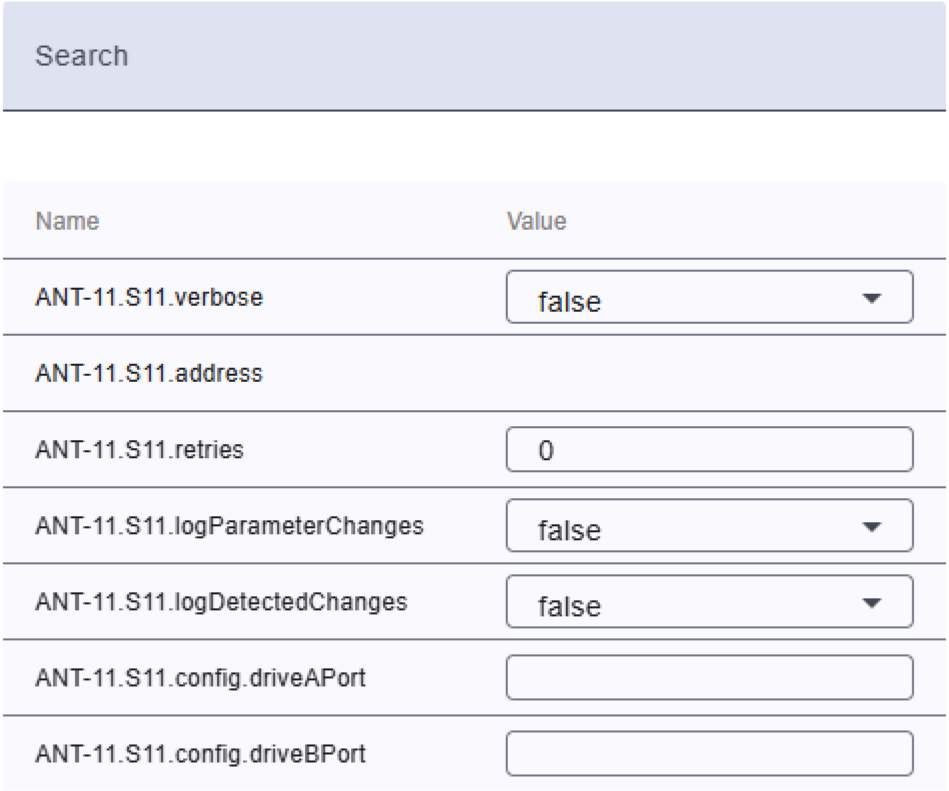
Reset: Resets the device.

Low level command: Command to check check the value.

Send: Send the low level command.

Text area: Shows the returned value of command.

2.3.2.1.4 Config page



Name	Value
ANT-11.S11.verbose	false
ANT-11.S11.address	
ANT-11.S11.retries	0
ANT-11.S11.logParameterChanges	false
ANT-11.S11.logDetectedChanges	false
ANT-11.S11.config.driveAPort	
ANT-11.S11.config.driveBPort	

Search bar: Allows users to filter events by keywords.

Header row

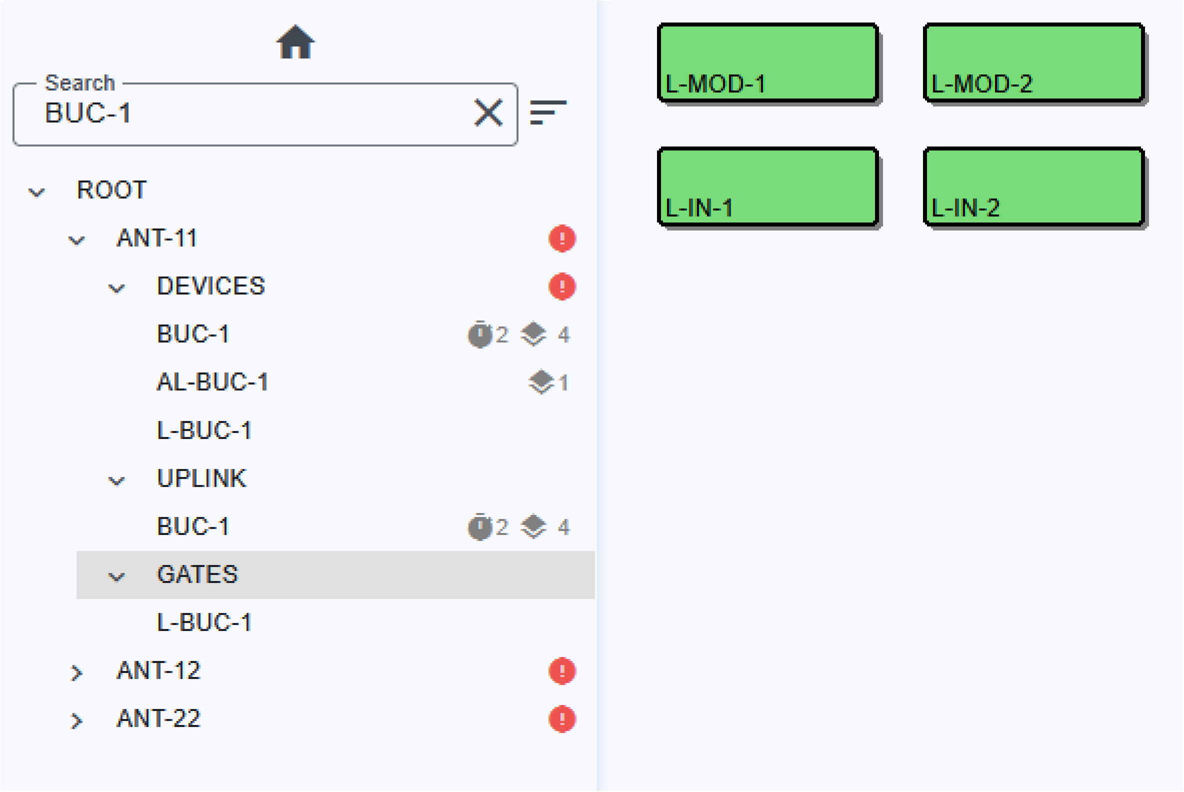
- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.

Table columns

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

2.4 Treeview

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



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

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

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

Each device node is marked with its fault state with icons

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

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

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

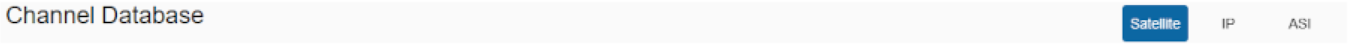
Functionality

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

2.5 Channel database

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

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



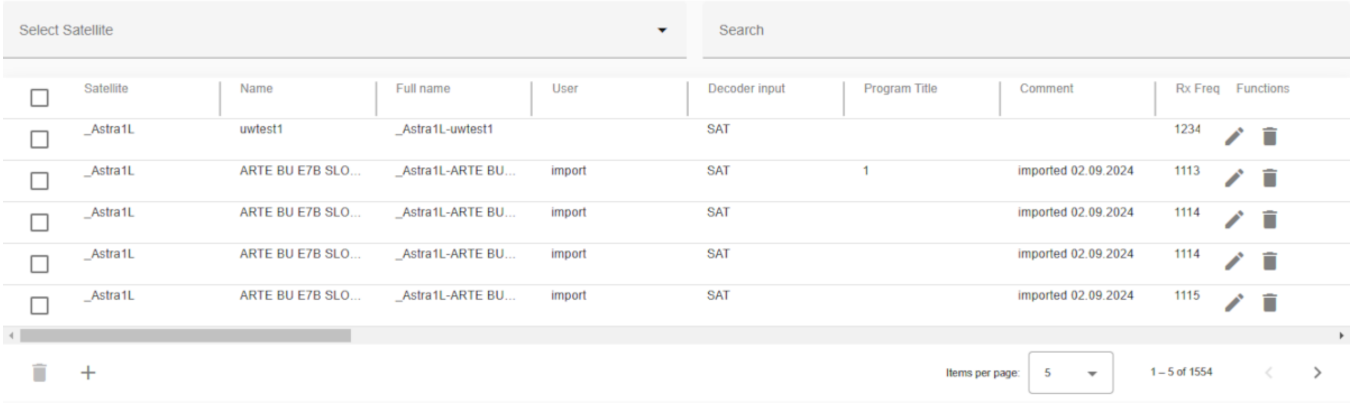
- Satellite channels (default)
  - satellite channels include parameters like frequency, modulation type
  - transmit and receiving via satellite is supported
  - basic audio/video settings are also part of these channels
- IP/Streaming channels
  - IP channels support various network streaming protocols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

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

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

2.5.1 Satellite channels

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



□	Satellite	Name	Full name	User	Decoder input	Program Title	Comment	Rx Freq	Functions
□	_Astra1L	uwtest1	_Astra1L-uwtest1		SAT			1234	
□	_Astra1L	ARTE BU E7B SLO...	_Astra1L-ARTE BU...	import	SAT	1	imported 02.09.2024	1113	
□	_Astra1L	ARTE BU E7B SLO...	_Astra1L-ARTE BU...	import	SAT		imported 02.09.2024	1114	
□	_Astra1L	ARTE BU E7B SLO...	_Astra1L-ARTE BU...	import	SAT		imported 02.09.2024	1114	
□	_Astra1L	ARTE BU E7B SLO...	_Astra1L-ARTE BU...	import	SAT		imported 02.09.2024	1115	

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

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

For receive applications the following logical devices are available:

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

For transmit applications the following logical devices are available:

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

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

Search bars

- Located at the top of the table.

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

Header row

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

Table columns

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

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

The following function(s) are available:

Function	Icon	Description
delete	🗑	delete the selected channels from the database

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

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

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

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

Function	Icon	Description
Add	+	add a new channel

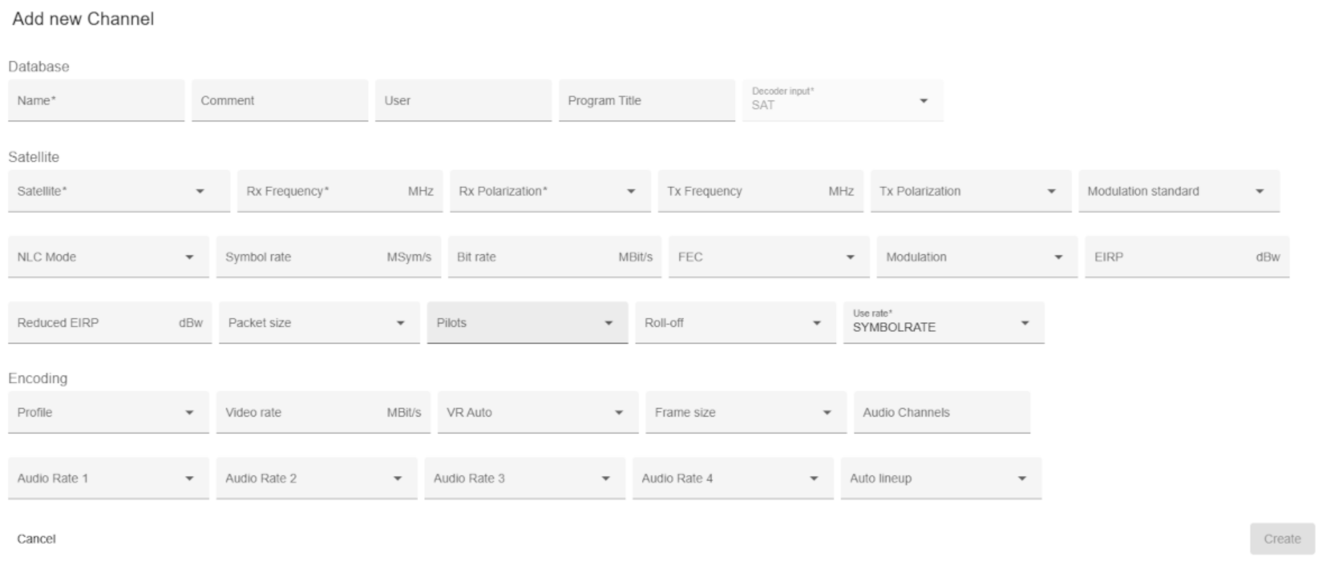
Function	Icon	Description
Delete	🗑	to delete all selected the channels
Page	Pagination controls	set the number of items displayed per page (e.g., 5, 10, or 25) and navigate between pages

The satellite channel table contains the following columns:

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

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

2.5.1.1 Add/Edit satellite channel dialog


Add new Channel

Decoder input* SAT

Use rate* SYMBOLRATE

The Add or Edit satellite channel dialog shows up on

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

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

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

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

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

The following actions are available :

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

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

2.5.1.2 On delete confirmation dialog

The Delete confirmation dialog shows up on

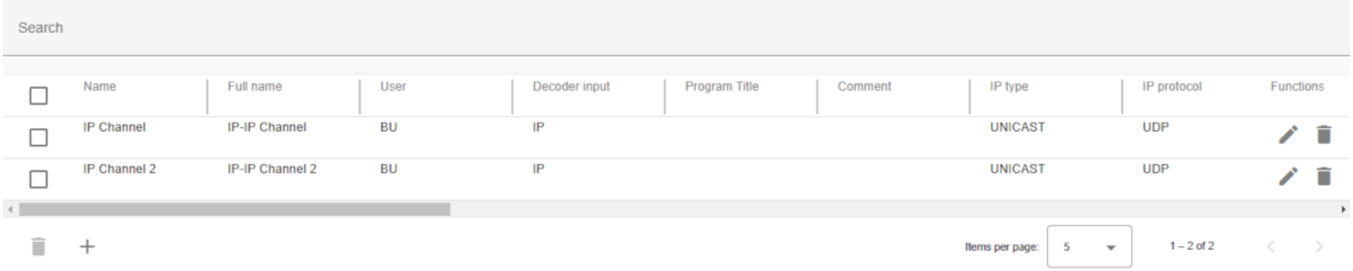
- click on 🗑 in the table row on the bottom toolbar

The following actions are `available`:

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

2.5.2 IP channels

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



Search

☐	Name	Full name	User	Decoder input	Program Title	Comment	IP type	IP protocol	Functions
☐	IP Channel	IP-IP Channel	BU	IP			UNICAST	UDP	✎ 🗑
☐	IP Channel 2	IP-IP Channel 2	BU	IP			UNICAST	UDP	✎ 🗑

🗑 +   Items per page: 5   1 – 2 of 2

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

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

For receive applications the following logical devices are available:

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

For transmit applications the following logical devices are available:

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

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

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

Search bars

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

Header row

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

Table columns

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

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

The following function(s) are available:

Function	Icon	Description
delete	🗑	delete the selected channels from the database

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

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

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

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

Function	Icon	Description

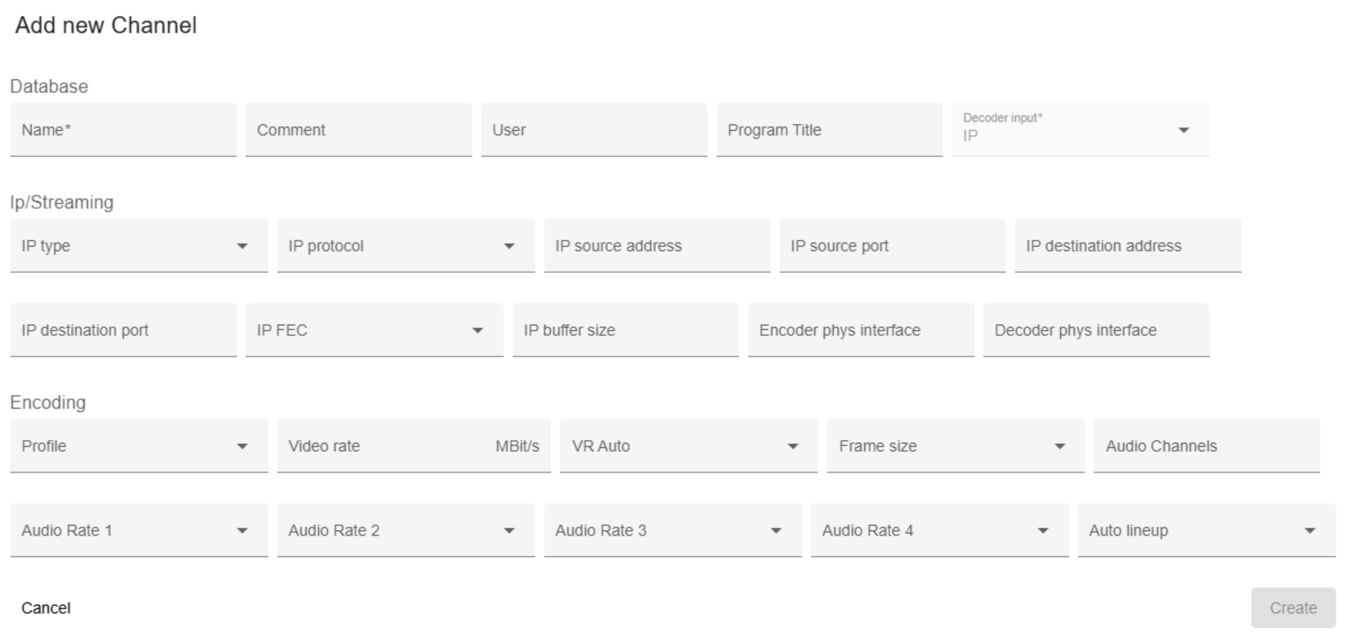
Function	Icon	Description
Add	+	add a new channel
Delete	🗑	to delete all selected the channels
Page	Pagination controls	set the number of items displayed per page (e.g., 5, 10, or 25) and navigate between pages

The IP channel table contains the following columns:

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

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

2.5.2.1 Add/Edit IP channel dialog**


Add new Channel

Database

Name*, Comment, User, Program Title, Decoder input* IP

Ip/Streaming

IP type, IP protocol, IP source address, IP source port, IP destination address

IP destination port, IP FEC, IP buffer size, Encoder phys interface, Decoder phys interface

Encoding

Profile, Video rate MBit/s, VR Auto, Frame size, Audio Channels

Audio Rate 1, Audio Rate 2, Audio Rate 3, Audio Rate 4, Auto lineup

The Add or Edit IP channel dialog shows up on

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

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

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

Field	Description

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

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

The following actions are `available`:

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

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

2.5.2.2 On delete confirmation dialog

The Delete confirmation dialog shows up on

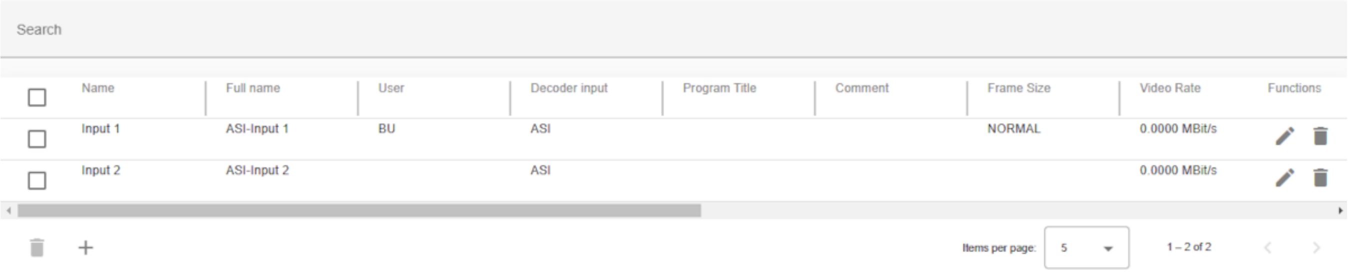
- click on 🗑 in the table row on the bottom toolbar

The following actions are `available`:

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

2.5.3 ASI channels

The list of all ASI channel records in the database. The table can be scrolled horizontally. Name and functions row are sticky.



Search

□	Name	Full name	User	Decoder input	Program Title	Comment	Frame Size	Video Rate	Functions
□	Input 1	ASI-Input 1	BU	ASI			NORMAL	0.0000 MBit/s	
□	Input 2	ASI-Input 2		ASI				0.0000 MBit/s	

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

For receive applications the following logical devices are available:

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

For transmit applications the following logical devices are available:

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

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

Search bars

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

Header row

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

Table columns

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

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

Function	Icon	Description
delete	🗑	delete the selected channels from the database

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

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

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

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

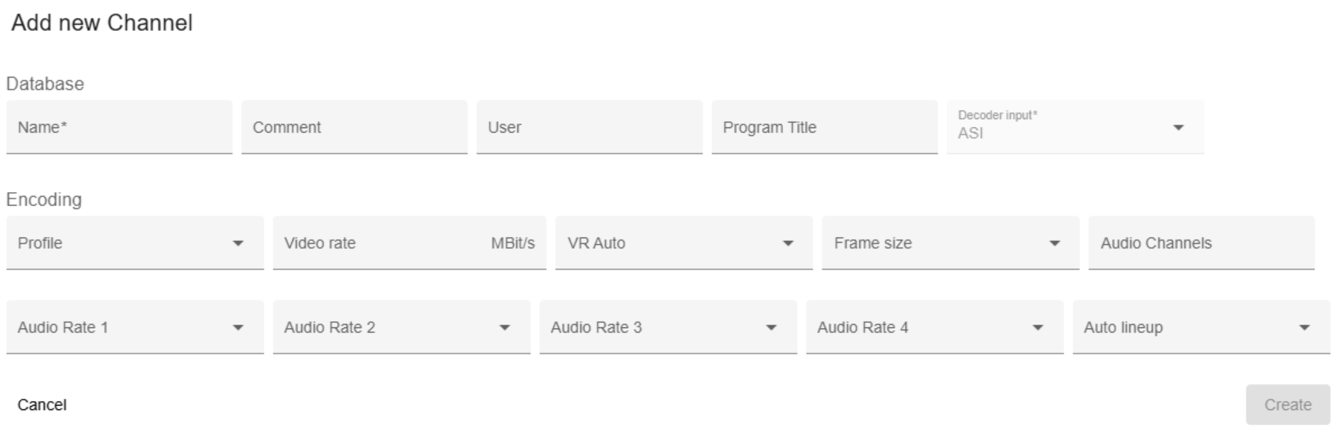
Function	Icon	Description
Add	+	add a new channel
Delete	🗑	to delete all selected the channels
Page	Pagination controls	set the number of items displayed per page (e.g., 5, 10, or 25) and navigate between pages

The ASI channel table contains the following columns:

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

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

2.5.3.1 Add/edit ASI channel dialog**



The Add or Edit ASI channel dialog shows up on

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

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

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

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

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

The following actions are `available` :

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

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

2.5.3.2 On delete confirmation dialog

The Delete confirmation dialog shows up on

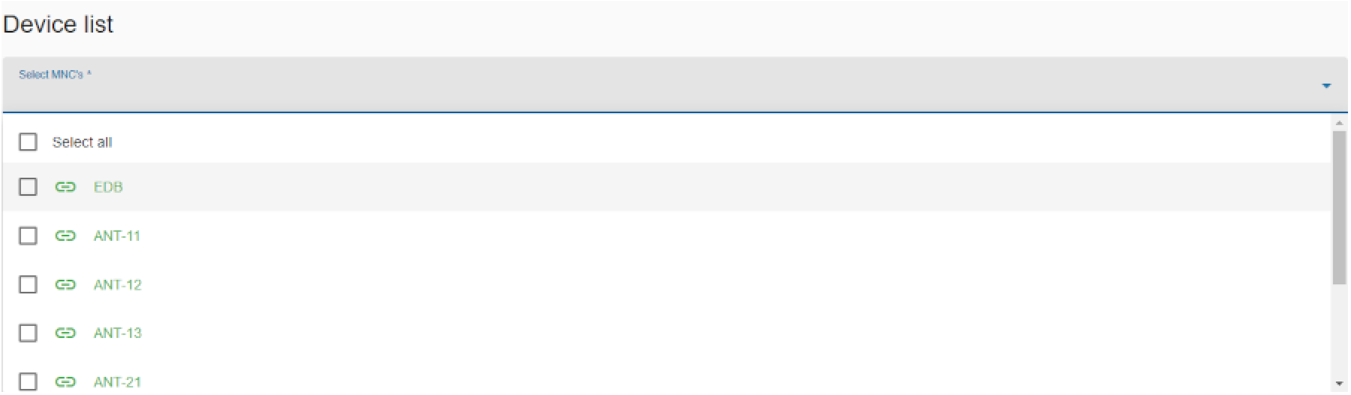
- click on 🗑 in the table row on the bottom toolbar

The following actions are `available` :

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

2.6 Device list

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

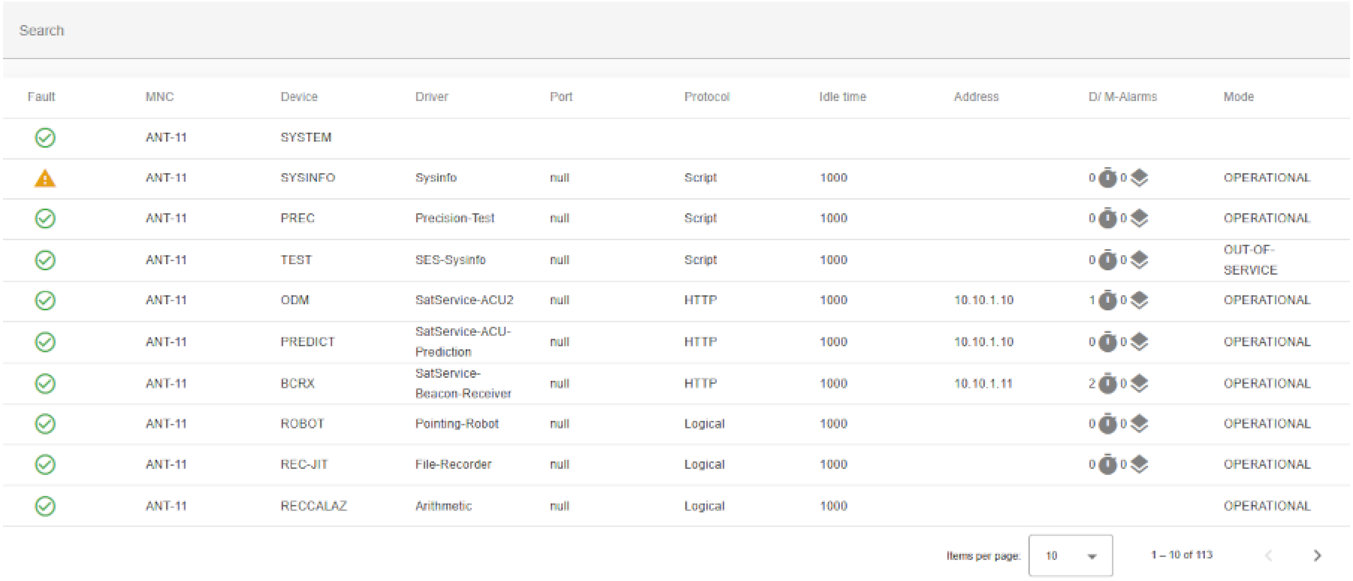


Device list

Select MNC's *

□ Select all
□ EDB
□ ANT-11
□ ANT-12
□ ANT-13
□ ANT-21

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



Search

Fault	MNC	Device	Driver	Port	Protocol	Idle time	Address	D/ M-Alarms	Mode
✓	ANT-11	SYSTEM							
⚠	ANT-11	SYSINFO	Sysinfo	null	Script	1000		0 ⏱ 0 ◆	OPERATIONAL
✓	ANT-11	PREC	Precision-Test	null	Script	1000		0 ⏱ 0 ◆	OPERATIONAL
✓	ANT-11	TEST	SES-Sysinfo	null	Script	1000		0 ⏱ 0 ◆	OUT-OF-SERVICE
✓	ANT-11	ODM	SatService-ACU2	null	HTTP	1000	10.10.1.10	1 ⏱ 0 ◆	OPERATIONAL
✓	ANT-11	PREDICT	SatService-ACU-Prediction	null	HTTP	1000	10.10.1.10	0 ⏱ 0 ◆	OPERATIONAL
✓	ANT-11	BCRX	SatService-Beacon-Receiver	null	HTTP	1000	10.10.1.11	2 ⏱ 0 ◆	OPERATIONAL
✓	ANT-11	ROBOT	Pointing-Robot	null	Logical	1000		0 ⏱ 0 ◆	OPERATIONAL
✓	ANT-11	REC-JIT	File-Recorder	null	Logical	1000		0 ⏱ 0 ◆	OPERATIONAL
✓	ANT-11	RECCALAZ	Arithmetic	null	Logical	1000			OPERATIONAL

Items per page: 10 1 – 10 of 113

Search bar

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

Header row

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

Table columns

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

Bottom toolbar

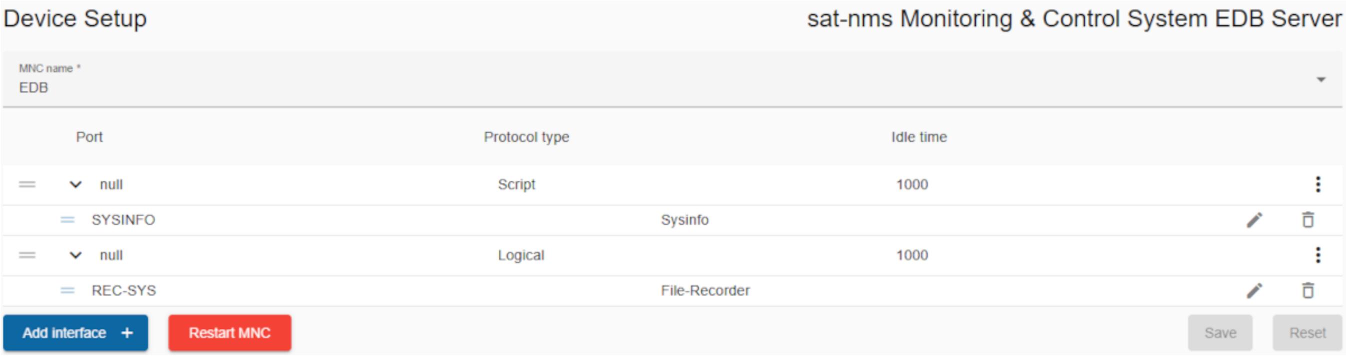
- **Pagination controls**: Allows users to set the number of items displayed per page and navigate between pages.

Functionality

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

2.7 Device setup

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



Device Setup

sat-nms Monitoring & Control System EDB Server

MNC name *
EDB

	Port	Protocol type		Idle time
	null	Script		1000
	SYSINFO		Sysinfo	
	null	Logical		1000
	REC-SYS		File-Recorder	

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

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

To configure device first select the MNC:

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

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

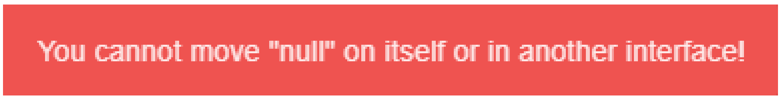
The first level of this tree represents the interface.

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

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

One or more devices can be attached to one interface.

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



You cannot move "null" on itself or in another interface!

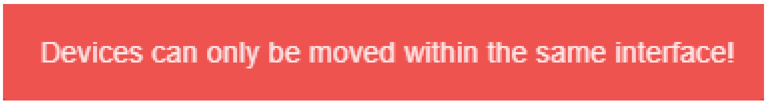
Expandable/Collapsible Rows = Interfaces

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

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

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

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

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

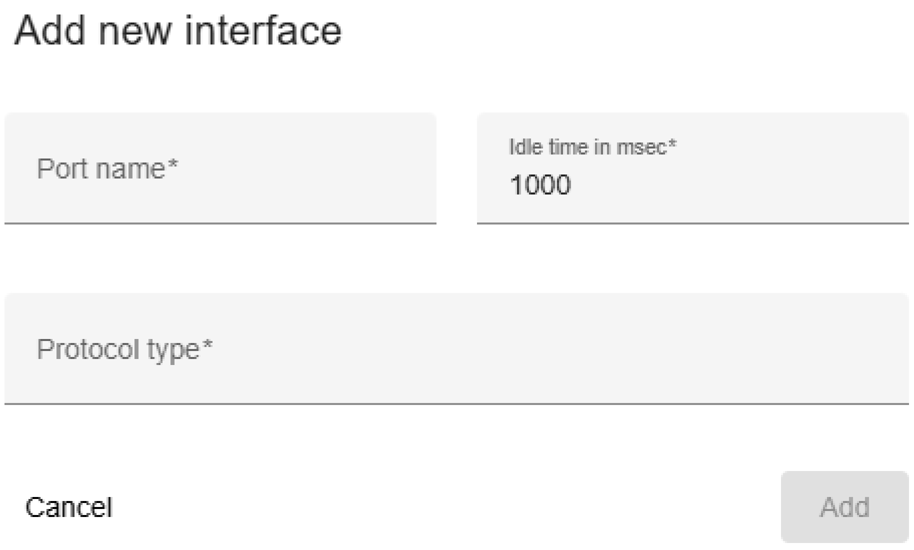
Functions

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

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

2.7.1 Dialogs

Add/edit interface


Add new interface

Port name*

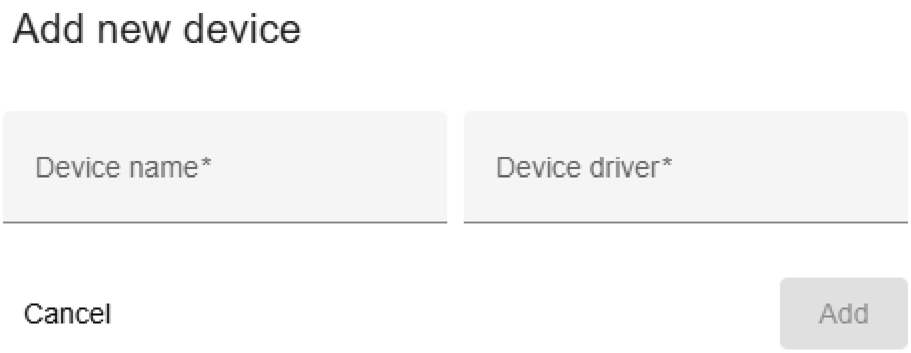
Idle time in msec*
1000

Protocol type*

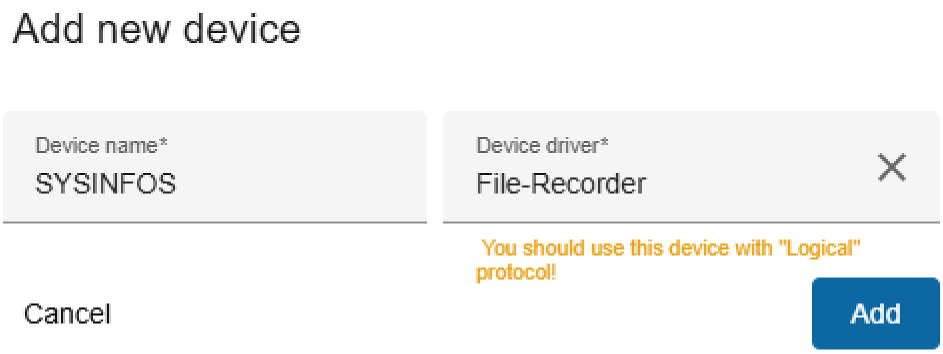
- **Trigger**:
  - when you click on the "Add interface +" button on the bottom to create new interface or
  - an on *pencil icon* to edit from context menu or
  - on *down arrow Add interface below*to add interface from context menu
  - The dialog title clearly indicates whether you are creating a new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

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

Add/edit device



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



Device name*
SYSINFOS

Device driver*
File-Recorder

You should use this device with "Logical" protocol!

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

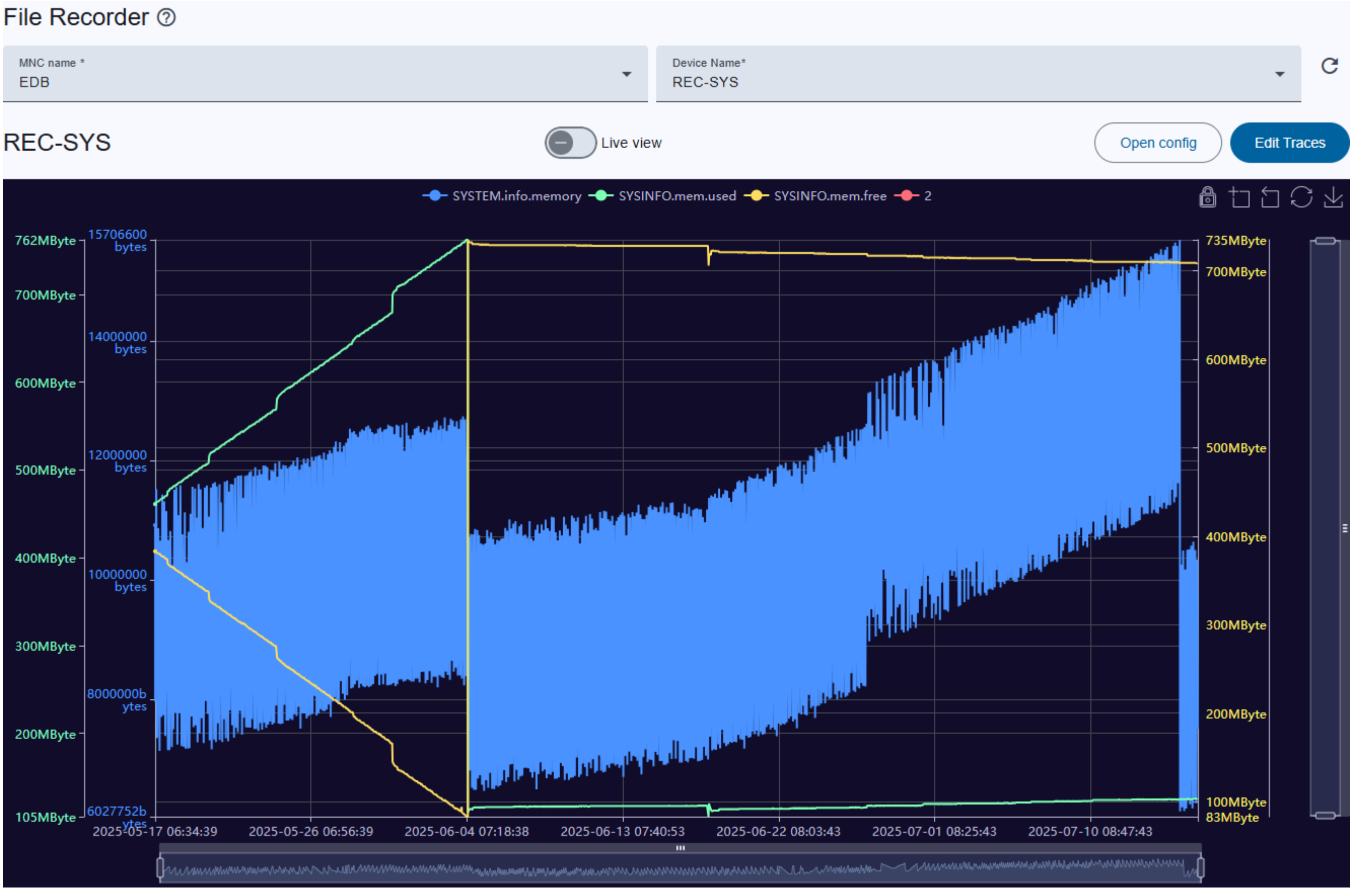
        - For new device, this creates the device with the provided details.
        - For existing device, this saves modifications.
        - If an error occurs in the input fields, the button is deactivated.
    - **Cancel**: Closes the dialog without saving.

Confirmation dialog

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

2.8 File recorder

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



File Recorder

MNC name * EDB

Device Name* REC-SYS

REC-SYS

Top toolbar

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

Title, live view, and configuration buttons

- **Title**: Displays the currently selected device name (e.g., "REC-SYS").

- **Live view toggle**: Beside for viewing stored log files, the File recorder Viewer can be used to display the logged data as it is acquired with an automated display update. By pressing the "live view" toggle button, the viewer changes to live view mode. In this mode the viewer shows up to 400 recent data points, automatically updated with an interval that is configurable separately from the standard interval at the device window setup (open config button).

- **Configuration buttons**:

  - **Open config**: Opens a device window allowing users to view or modify settings.
  - **Edit traces**: Opens a dialog for customizing or editing trace parameters, such as specific trace name and their units.

Graph

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

2.8.1 Functionality

Graph interaction

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

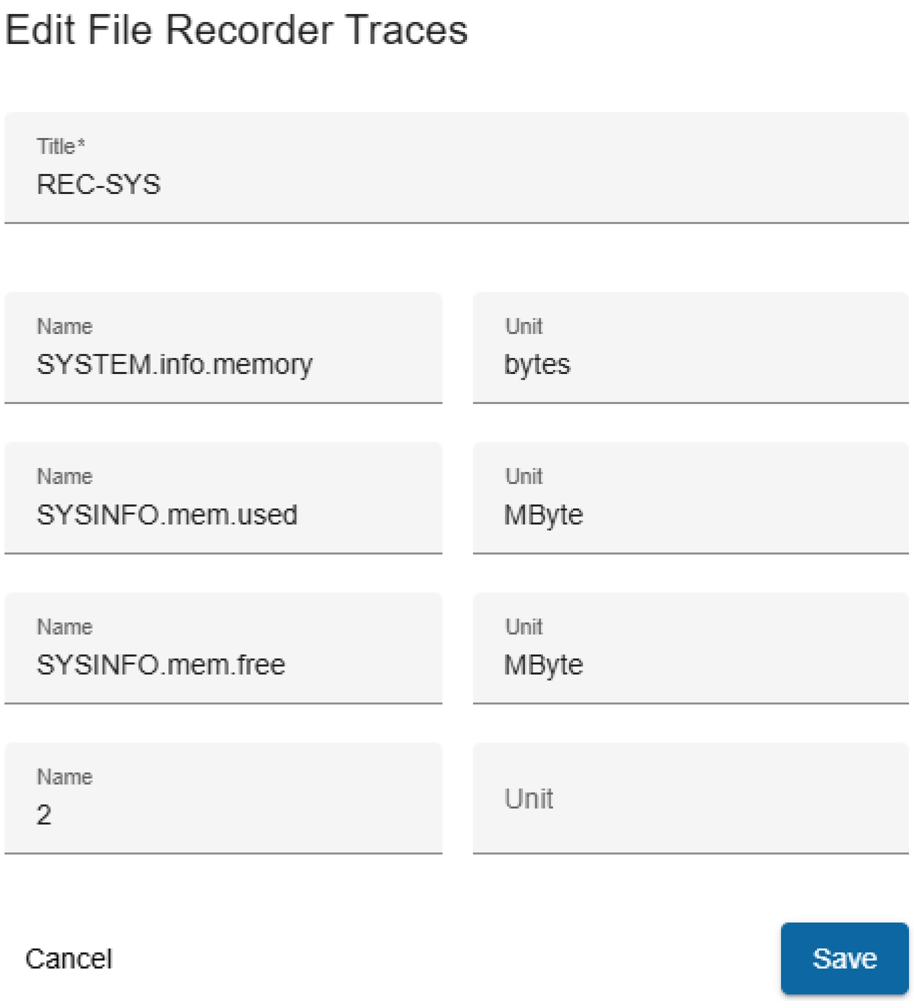
Tooltip: Displaying data reference to line and axis value under mouse pointer.

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

2.8.2 Dialogs

Open config: Opens a device window config page.

Configure traces



Edit File Recorder Traces

Title*
REC-SYS

Name
SYSTEM.info.memory

Unit
bytes

Name
SYSINFO.mem.used

Unit
MByte

Name
SYSINFO.mem.free

Unit
MByte

Name
2

Unit

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

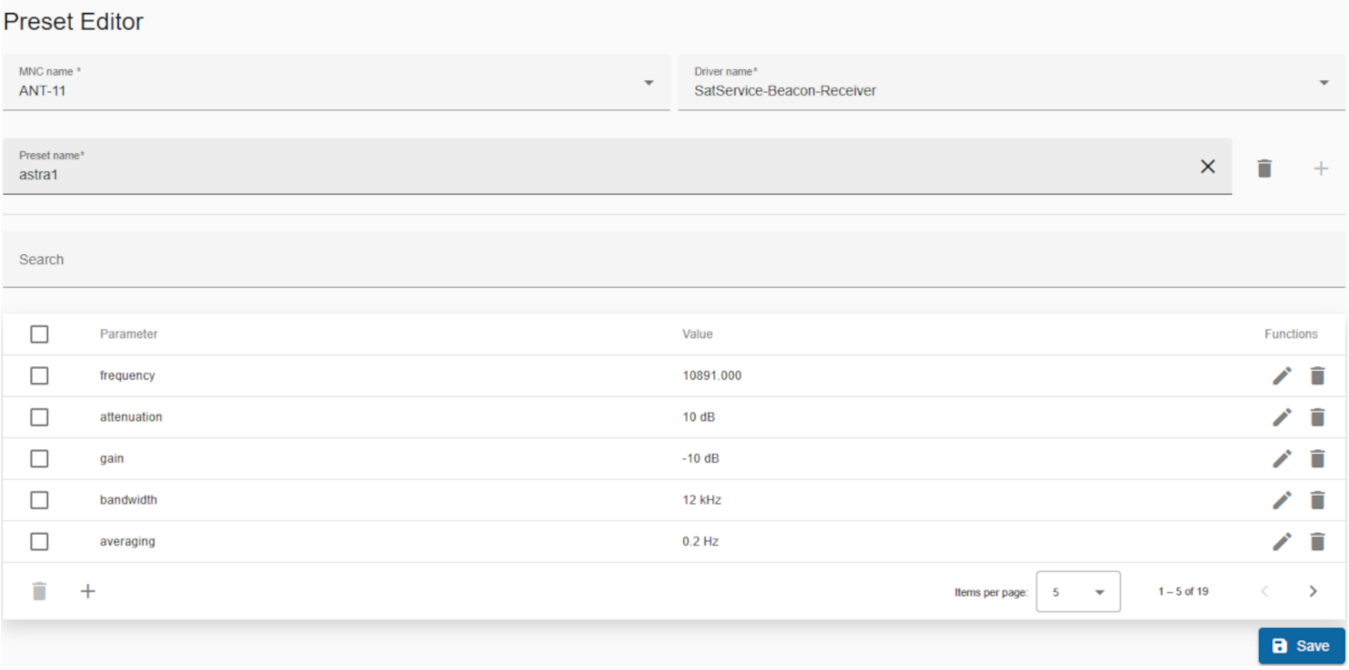
- **Cancel**: Closes the dialog without saving.

2.9 Preset editor

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

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

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



Preset Editor

MNC name *: ANT-11

Driver name*: SatService-Beacon-Receiver

Preset name*: astra1

Search

☐	Parameter	Value	Functions
☐	frequency	10891.000	✎ 🗑
☐	attenuation	10 dB	✎ 🗑
☐	gain	-10 dB	✎ 🗑
☐	bandwidth	12 kHz	✎ 🗑
☐	averaging	0.2 Hz	✎ 🗑

Items per page: 5

1 – 5 of 19

Title: Title of this page.

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

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

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

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

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

Search bar

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

Header row

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

Table columns

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

Bottom toolbar

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

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

2.9.1 Functionality

Create preset

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

Create parameter

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

Edit parameter

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

Delete preset

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

Delete parameter

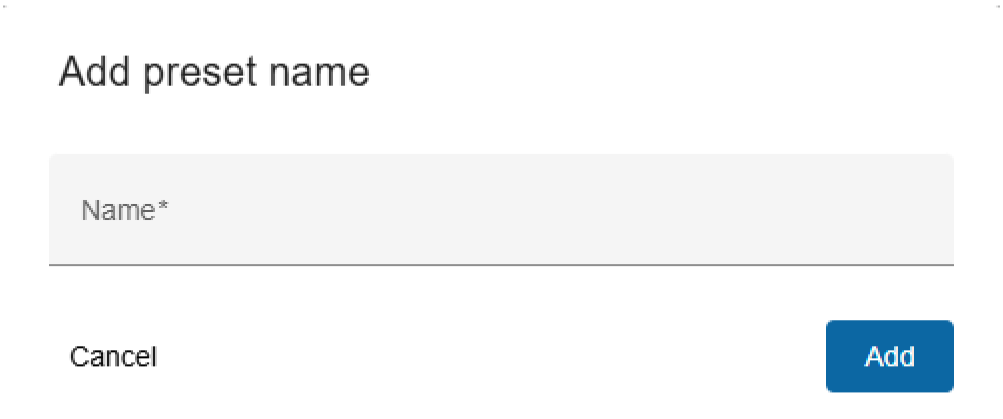
- Click the *trash icon* in the Functions column to delete an parameter.
- Confirm the action in the pop-up dialog to delete the parameter.

Pagination

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

2.9.2 Dialogs

Add preset dialog



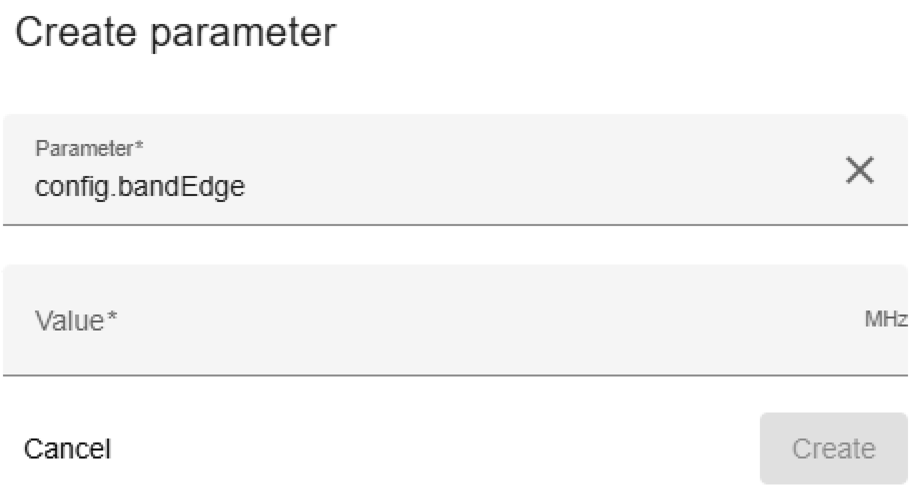
Add preset name

Name*

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

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

Add/edit parameter



Create parameter

Parameter*
config.bandEdge

Value*

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

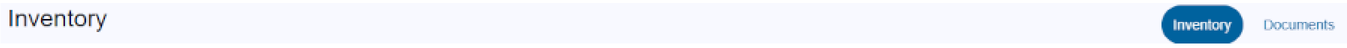
Confirmation dialog

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

2.10 Inventory database

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

The title bar provides access to the two different datasets:


Inventory

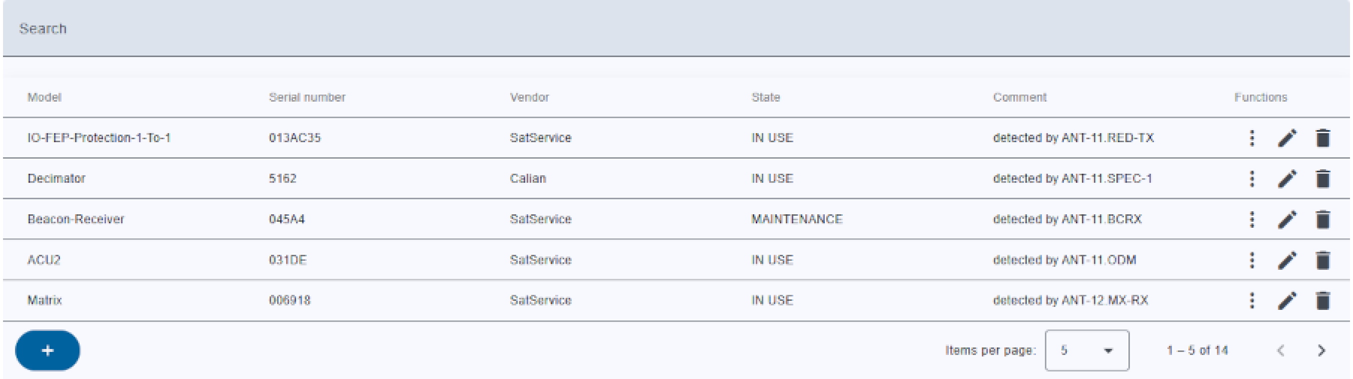
- Inventory (default)
  - displays the list of inventory drivers
- Documents
  - upload document
  - list of uploaded documents

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

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

2.10.1 Inventory

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


Search

Model	Serial number	Vendor	State	Comment	Functions
IO-FEP-Protection-1-To-1	013AC35	SatService	IN USE	detected by ANT-11.RED-TX	⋮ ✎ 🗑
Decimator	5162	Calian	IN USE	detected by ANT-11.SPEC-1	⋮ ✎ 🗑
Beacon-Receiver	045A4	SatService	MAINTENANCE	detected by ANT-11.BCRX	⋮ ✎ 🗑
ACU2	031DE	SatService	IN USE	detected by ANT-11.ODM	⋮ ✎ 🗑
Matrix	006918	SatService	IN USE	detected by ANT-12.MX-RX	⋮ ✎ 🗑

Items per page: 5

1 – 5 of 14

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

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

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

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

The Inventory list also contains devices and logs.

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

Search bar

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

Header row

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

Table columns

The inventory table contains the following columns:

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

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

The following functions for inventory are `available` :

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

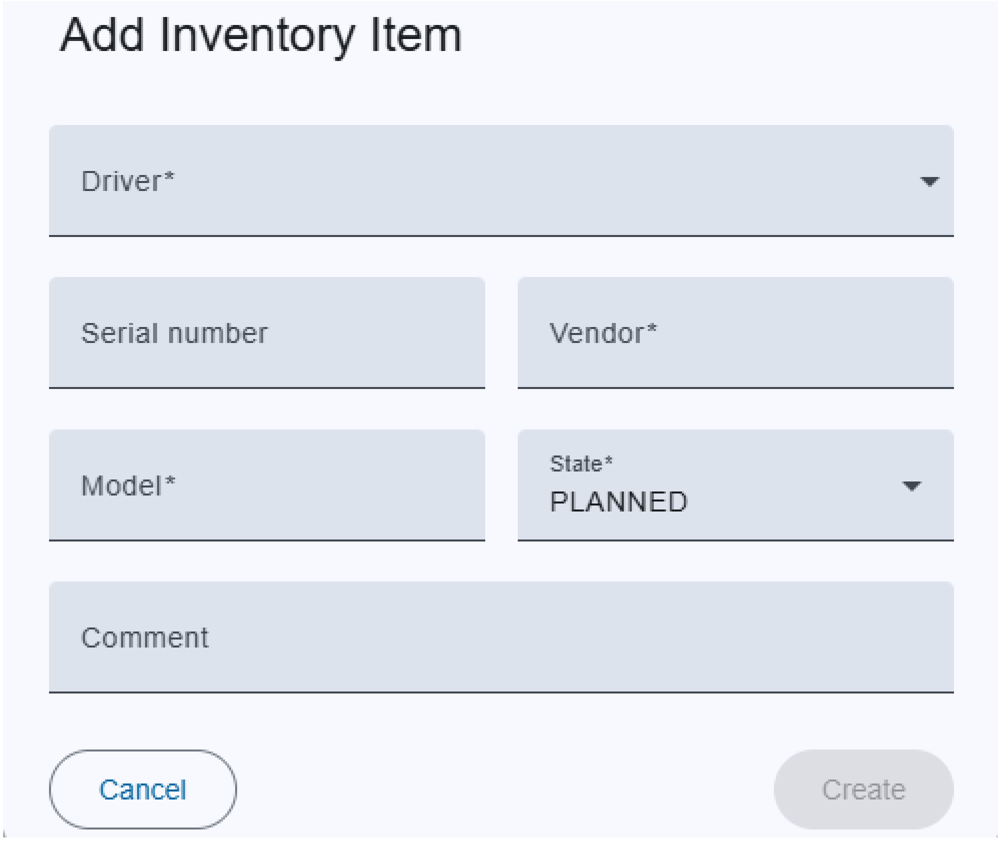
Clicking on a inventory item row shows other relatives tables:

- Devices
- Logs

Bottom toolbar

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

2.10.1.1 Add/edit inventory dialog



Add Inventory Item

Driver*

Serial number

Vendor*

Model*

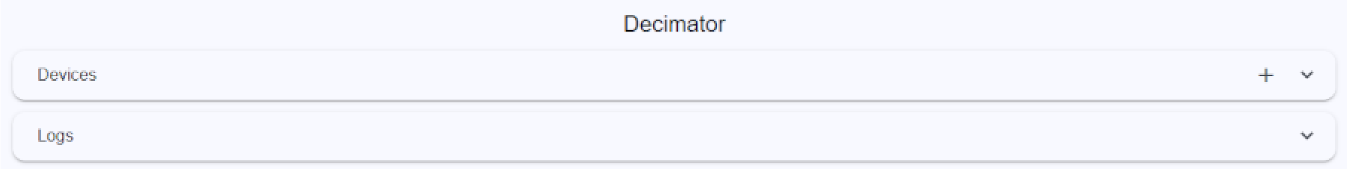
State*
PLANNED

Comment

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

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

2.10.2 Inventory related tables

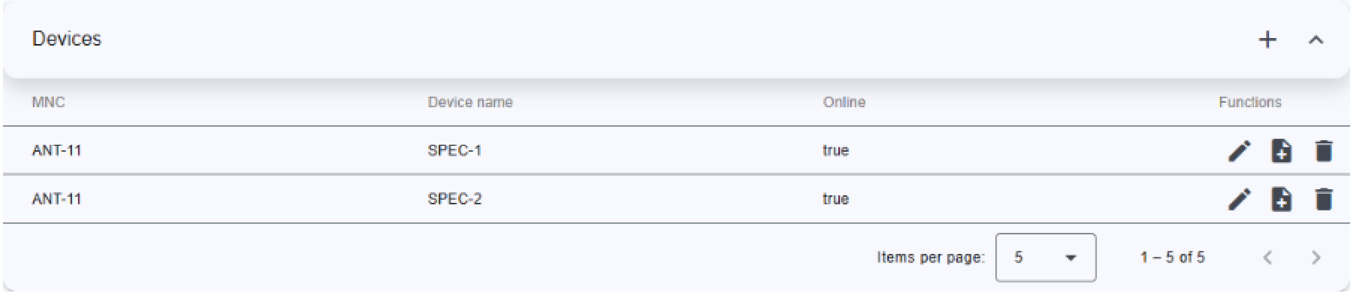


Toolbar (top)

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

2.10.2.1 Devices table

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



MNC	Device name	Online	Functions
ANT-11	SPEC-1	true	✎ 🗋 🗑
ANT-11	SPEC-2	true	✎ 🗋 🗑

Top

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

Header row

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

near the column name.

Table columns

The devices table contains the following columns:

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

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

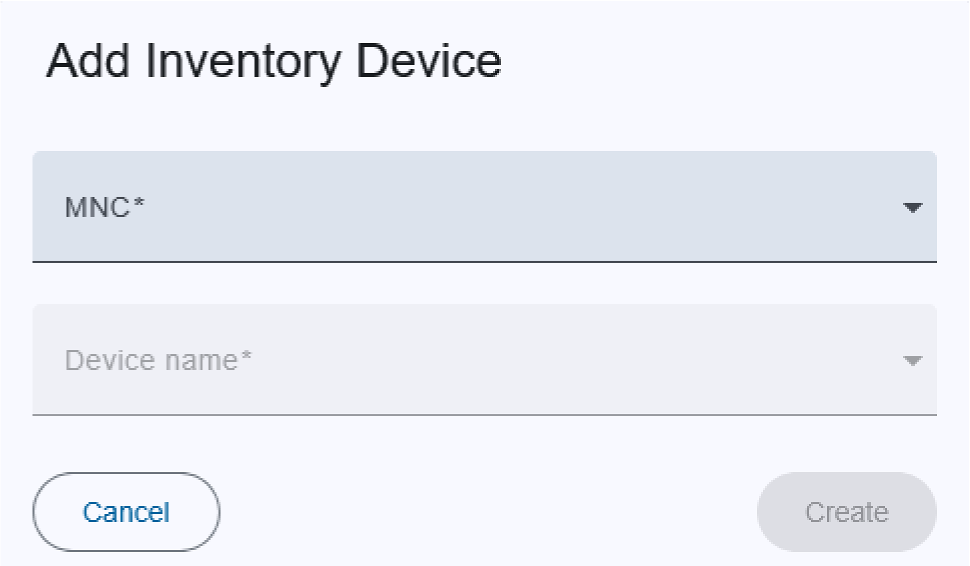
The following functions for device are `available` :

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

Bottom toolbar

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

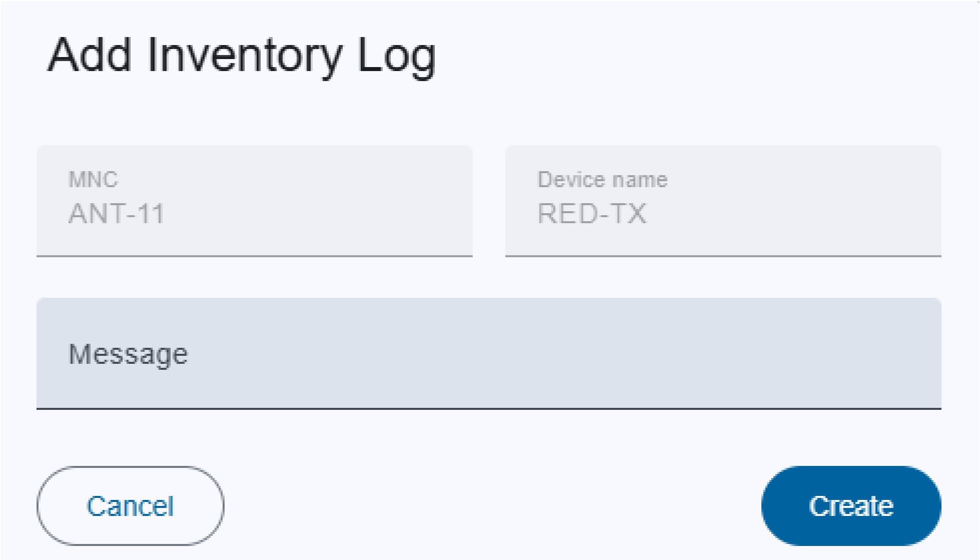
2.10.2.2 Add/edit device dialog



Add Inventory Device

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

2.10.2.3 Add log dialog



Add Inventory Log

MNC
ANT-11

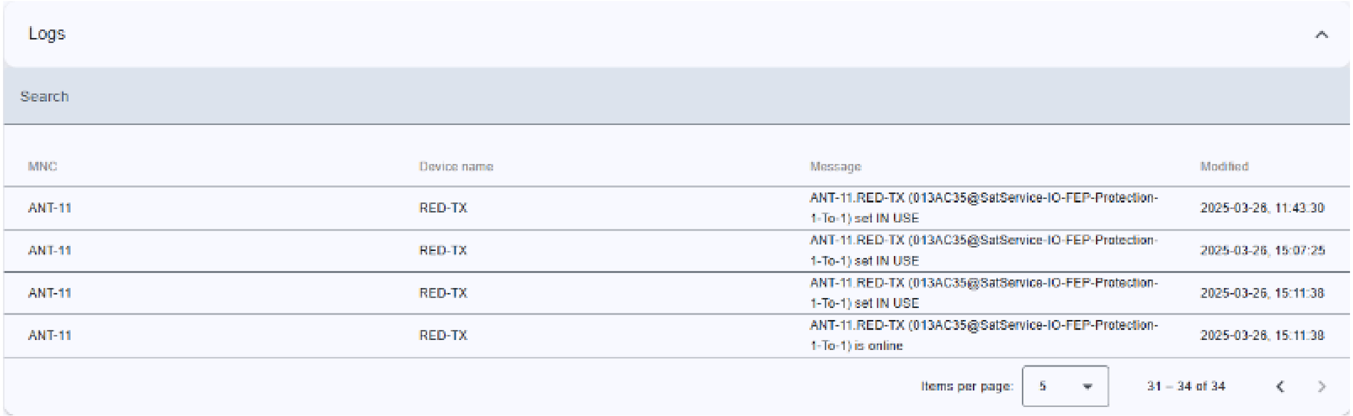
Device name
RED-TX

Message

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

2.10.2.4 Logs table

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



Logs

Search

MNC	Device name	Message	Modified
ANT-11	RED-TX	ANT-11.RED-TX (013AC35@SatService-IO-FEP-Protection-1-To-1) set IN USE	2025-03-26, 11:43:30
ANT-11	RED-TX	ANT-11.RED-TX (013AC35@SatService-IO-FEP-Protection-1-To-1) set IN USE	2025-03-26, 15:07:25
ANT-11	RED-TX	ANT-11.RED-TX (013AC35@SatService-IO-FEP-Protection-1-To-1) set IN USE	2025-03-26, 15:11:38
ANT-11	RED-TX	ANT-11.RED-TX (013AC35@SatService-IO-FEP-Protection-1-To-1) is online	2025-03-26, 15:11:38

Items per page: 5 31 – 34 of 34

Top

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

Search bar

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

Header row

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

Table columns

The Logs table contains the following columns:

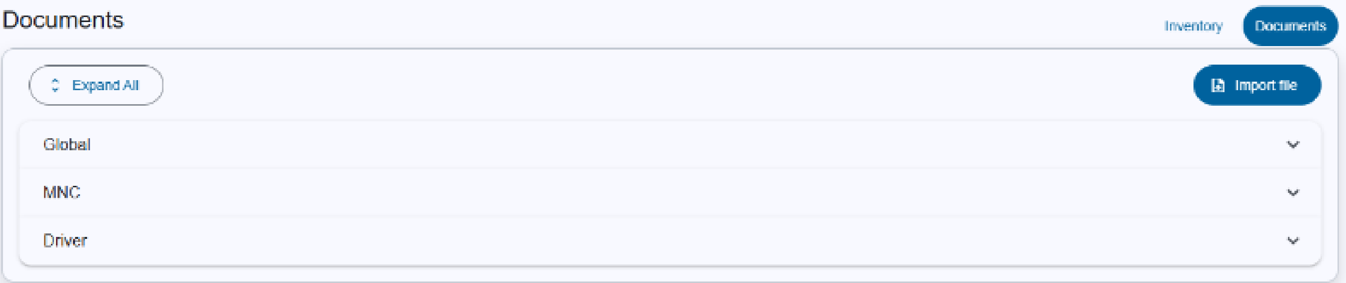
Column name	Description
MNC	Name of M&C.
Device name	Device name.
Message	Purpose of this log.
Modified	Date of added log.

Bottom toolbar

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

2.11 Documents

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



Expand All: Expand all category tables.

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

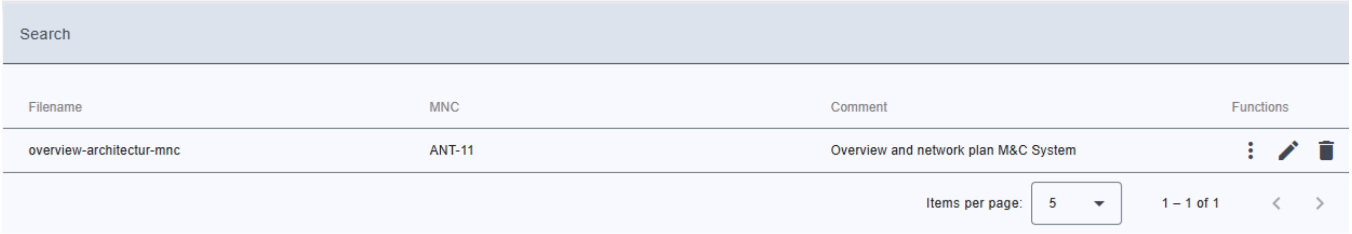
Global : If the document is uploaded without M&C or driver name, it belongs to Global category.

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

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

Tables

The tables for Global, MNC and Driver are identical.



Filename	MNC	Comment	Functions
overview-architectur-mnc	ANT-11	Overview and network plan M&C System	⋮ ✎ 🗑

Search bars

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

Header row

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

Table columns

The document table contains the following columns:

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

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

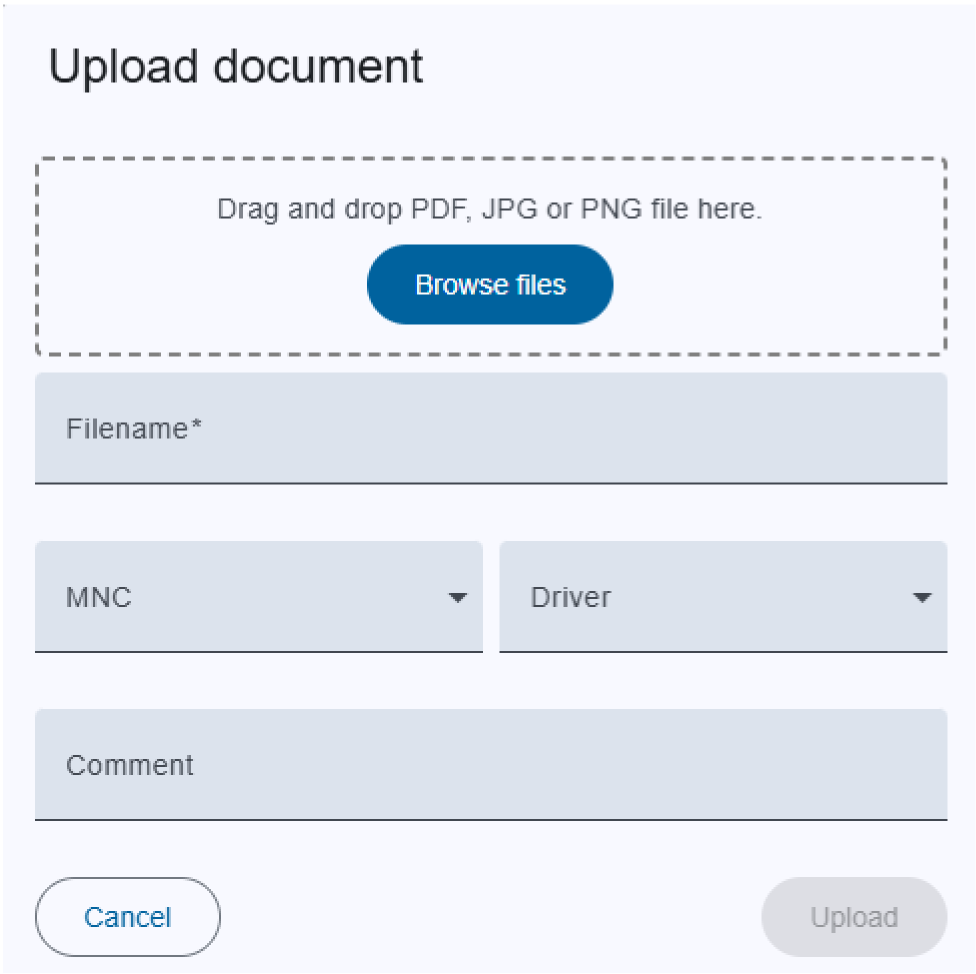
The following functions for inventory are `available` :

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

Bottom toolbar

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

2.11.1 Upload/edit document



Drag and drop PDF, JPG or PNG file here.

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

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

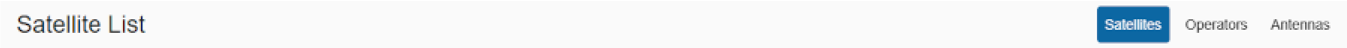
Edit document: If you want to change the document properties but do not want to upload the document, you do not need to drop or browse the "File" section.

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

2.12 Satellite database

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

The title bar provides access to the three different datasets:



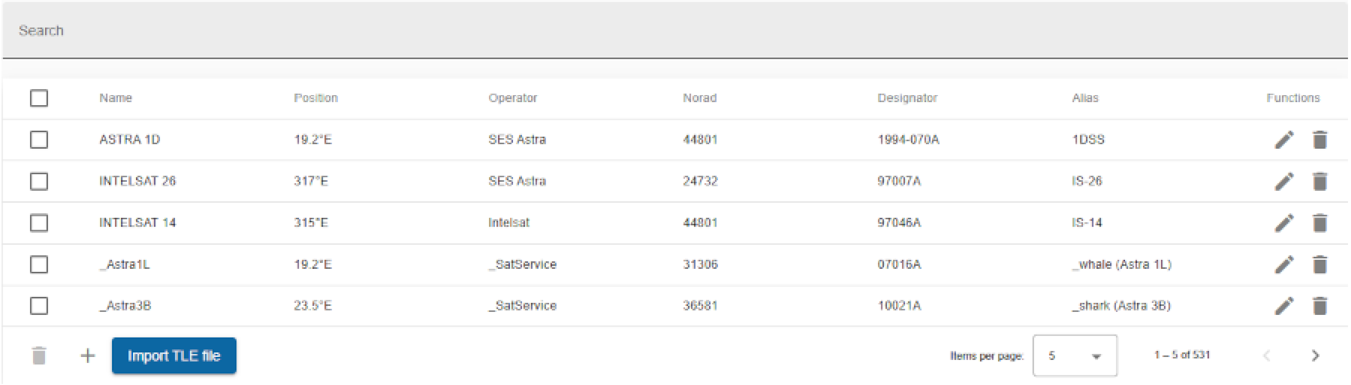
- Satellites (default)
  - displays the list of satellites stored in the database
- Satellite operators
  - each satellite belongs to an satellite operator
  - here you can store for example contact information for line-ups
- Antennas
  - list of configured antennas (and the related devices) which should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

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

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

2.12.1 Satellite list

This shows the list of all satellites in the database.



Search

☐	Name	Position	Operator	Norad	Designator	Alias	Functions
☐	ASTRA 1D	19.2°E	SES Astra	44801	1994-070A	1DSS	✎ 🗑
☐	INTELSAT 26	317°E	SES Astra	24732	97007A	IS-26	✎ 🗑
☐	INTELSAT 14	315°E	Intelsat	44801	97046A	IS-14	✎ 🗑
☐	_Astra1L	19.2°E	_SatService	31306	07016A	_whale (Astra 1L)	✎ 🗑
☐	_Astra3B	23.5°E	_SatService	36581	10021A	_shark (Astra 3B)	✎ 🗑

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

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

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

- Antenna-Management

Search bars

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

Header row

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

Table columns

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

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

The following function(s) are available:

Function	Icon	Description

Function	Icon	Description
delete	🗑	delete the selected satellite from the database

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

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

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

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

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

The satellite table contains the following columns:

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

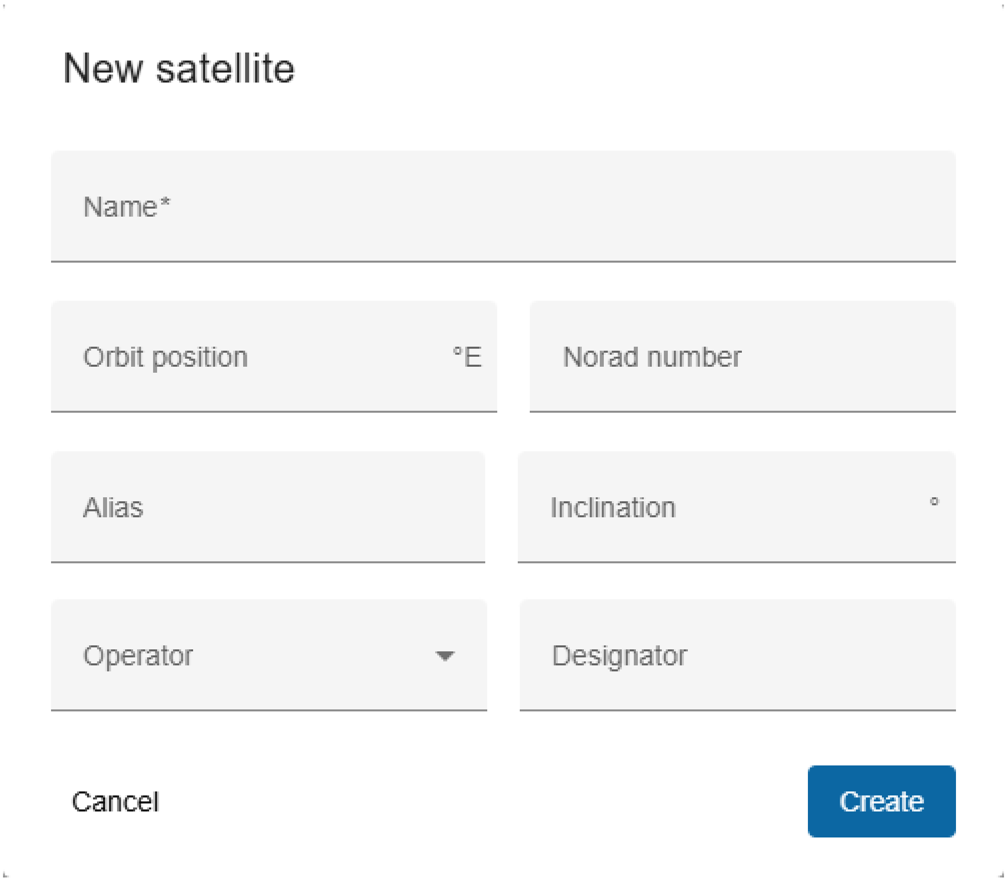
Clicking on a satellite row shows other relatives tables:

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

Bottom toolbar

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

2.12.2 Add/edit satellite dialog


New satellite

Name*

Orbit position °E

Norad number

Alias

Inclination °

Operator

Designator

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

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

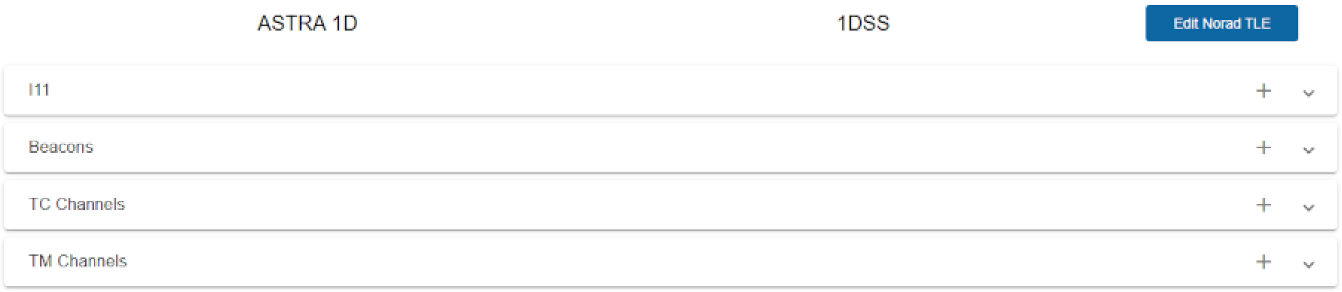
- **Actions**:

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

2.12.2.1 On delete confirmation dialog

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

2.12.3 Satellite related tables



Toolbar (top)

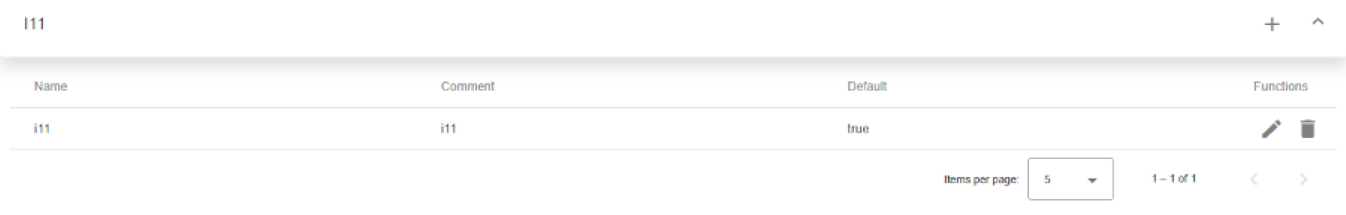
- **Title**:
    - Satellite name
    - Satellite alias
- **Edit NORAD TLE**: Trigger dialog to modify NORAD TLE.

Tables:

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

2.12.3.1 I11 table

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



I11

Name	Comment	Default	Functions
i11	i11	true	

Top

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

Header row

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

Table columns

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

Bottom toolbar

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

2.12.3.1.1 Functionality

Create i11

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

Edit i11

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

Delete i11

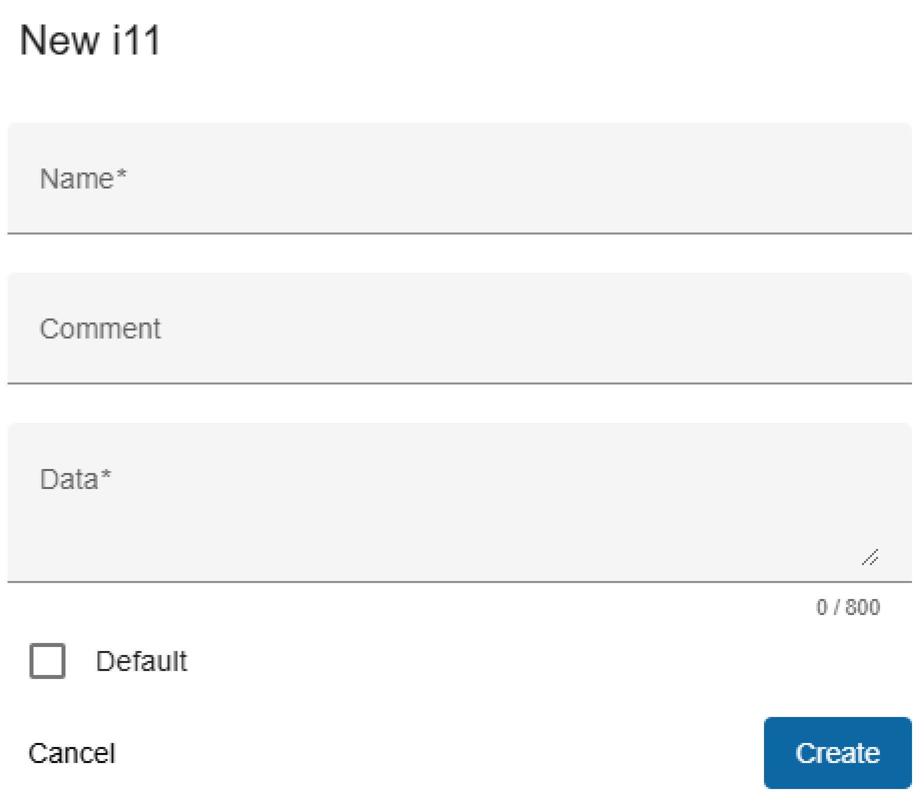
- Click the *trash icon* in the Functions column to delete an i11.
- Confirm the action in the pop-up dialog to delete the i11.

Pagination

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

2.12.3.1.2 Dialogs

Add/edit i11 dialog


New i11

Name*

Comment

Data*

0 / 800

□ Default

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

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

- **Actions**:

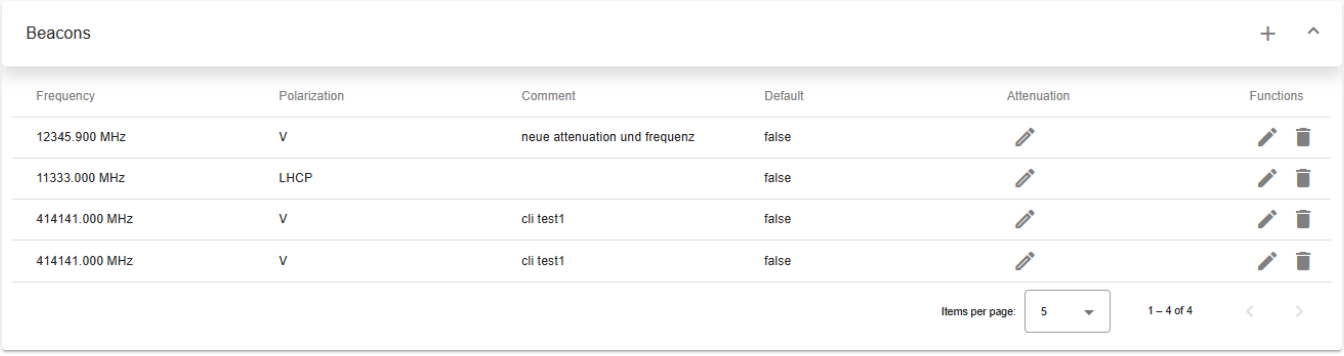
  - **Create/Save**:
    - For new i11, this creates the satellite with the provided details.
    - For existing i11, this saves modifications.
    - If an error occurs in the input fields, the button is deactivated.
  - **Cancel**: Closes the dialog without saving.

On delete confirmation dialog

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

2.12.3.2 Beacons table

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



Beacons

Frequency	Polarization	Comment	Default	Attenuation	Functions
12345.900 MHz	V	neue attenuation und frequenz	false	✎	✎ 🗑
11333.000 MHz	LHCP		false	✎	✎ 🗑
414141.000 MHz	V	cli test1	false	✎	✎ 🗑
414141.000 MHz	V	cli test1	false	✎	✎ 🗑

Items per page: 5 1 – 4 of 4

Top

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

Header row

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

Table columns

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

Bottom toolbar

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

2.12.3.2.1 Functionality

Create beacon

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

Edit beacon

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

Attenuation

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

Delete beacon

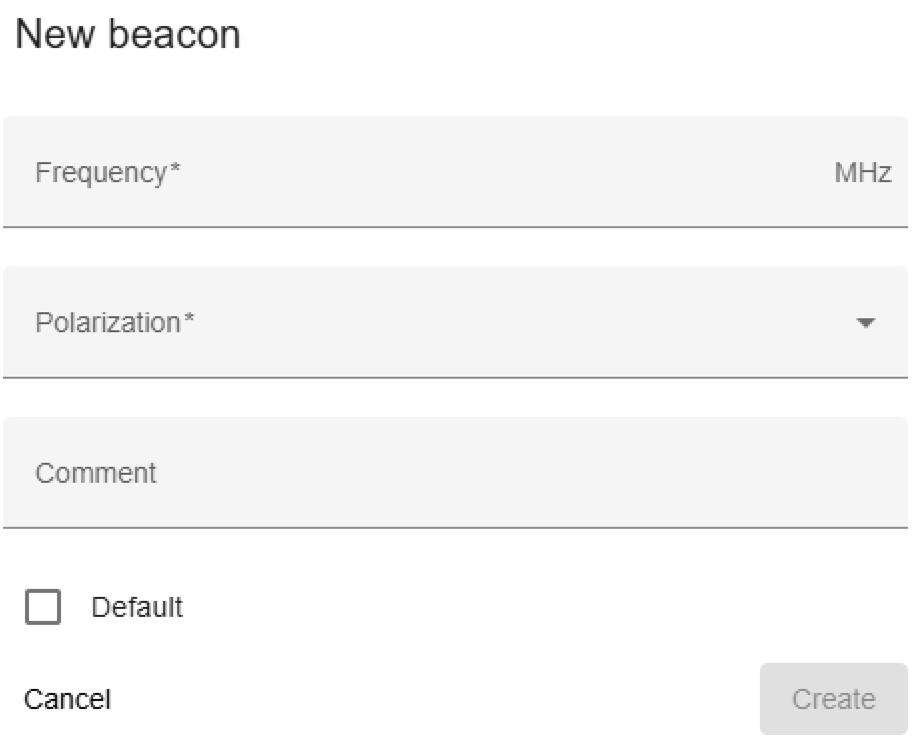
- Click the *trash icon* in the Functions column to delete an beacon.
- Confirm the action in the pop-up dialog to delete the beacon.

Pagination

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

2.12.3.2.2 Dialogs

Add/edit beacon dialog



New beacon

Frequency* MHz

Polarization*

Comment

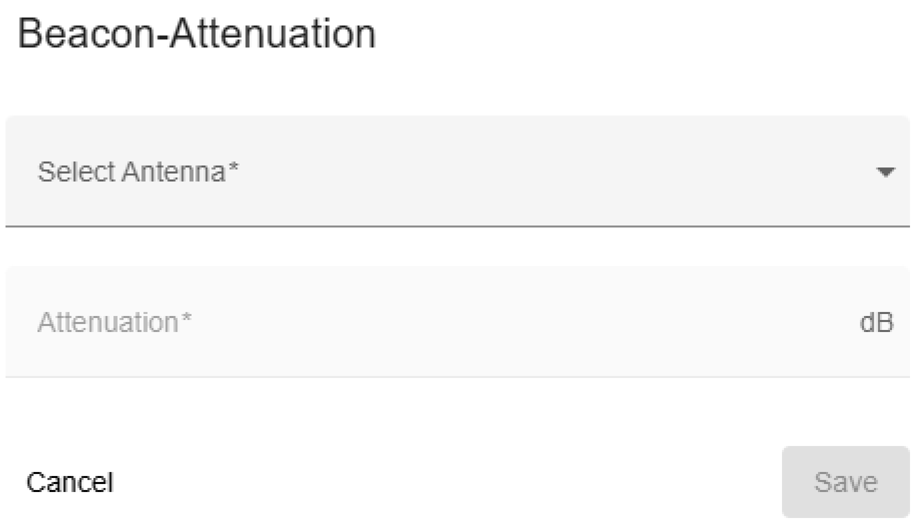
□ Default

Cancel

Create

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

Modify beacon attenuation



Beacon-Attenuation

Select Antenna*

Attenuation* dB

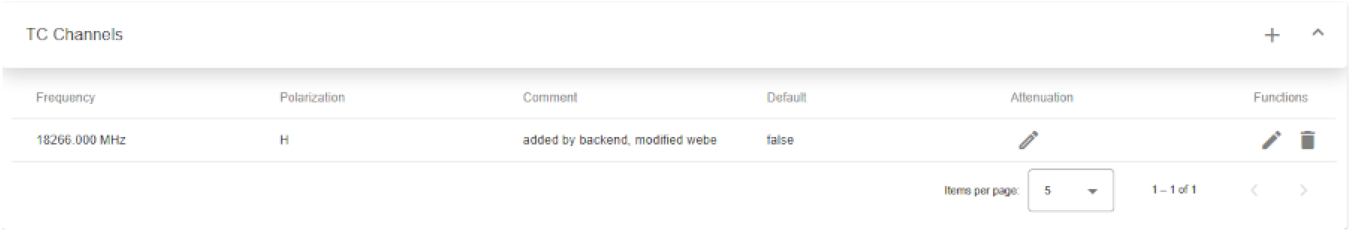
- **Trigger**: Click on the *pencil outlined* button in the row under *attenuation*
- **Fields**:
    - **Select antenna**: Drop-down to select antenna.
    - **Attenuation**: Receiver attenuation, floating point, dB.
- **Actions**:
    - **Save**:
        - Saves modifications.
        - If an error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

On delete confirmation dialog

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

2.12.3.3 TC channel table

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



TC Channels

Frequency	Polarization	Comment	Default	Attenuation	Functions
18266.000 MHz	H	added by backend, modified webe	false	✎	✎ 🗑

Items per page: 5 1 – 1 of 1

Top

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

Header row

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

Table columns

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

Bottom toolbar

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

2.12.3.3.1 Functionality

Create TC channel

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

Edit TC channel

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

Attenuation

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

Delete TC channel

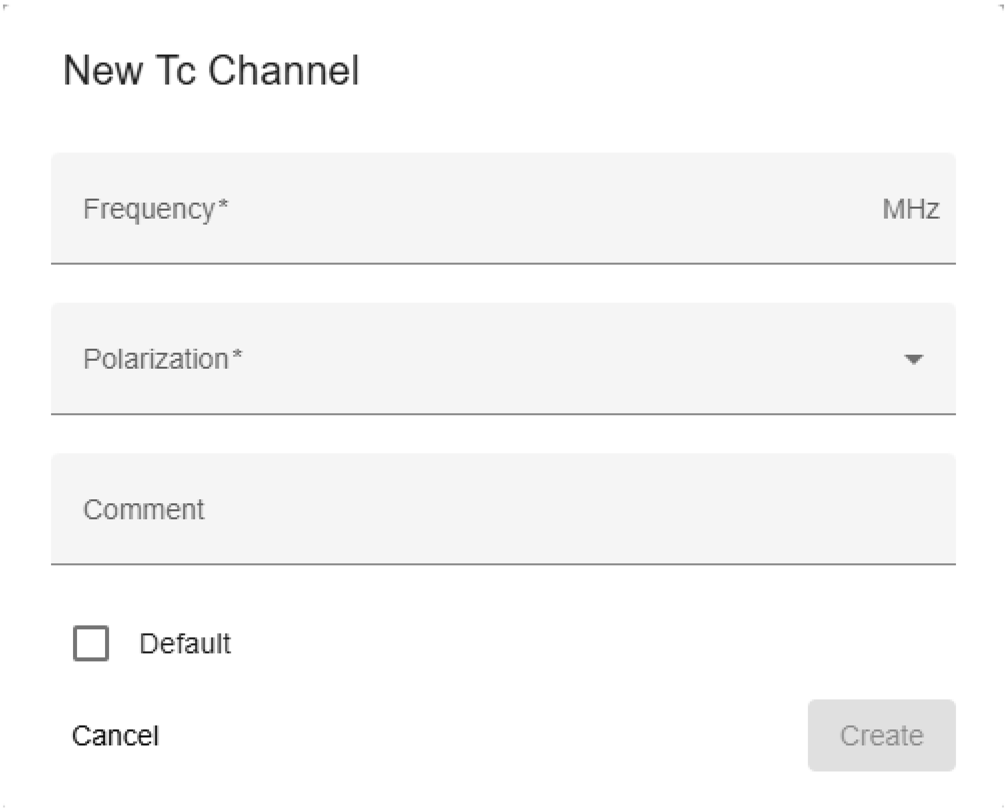
- Click the *trash icon* in the Functions column to delete an TC channel.
- Confirm the action in the pop-up dialog to delete the TC channel.

Pagination

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

2.12.3.3.2 Dialogs

Add/edit TC channel dialog



New Tc Channel

Frequency* MHz

Polarization*

Comment

□ Default

Cancel

Create

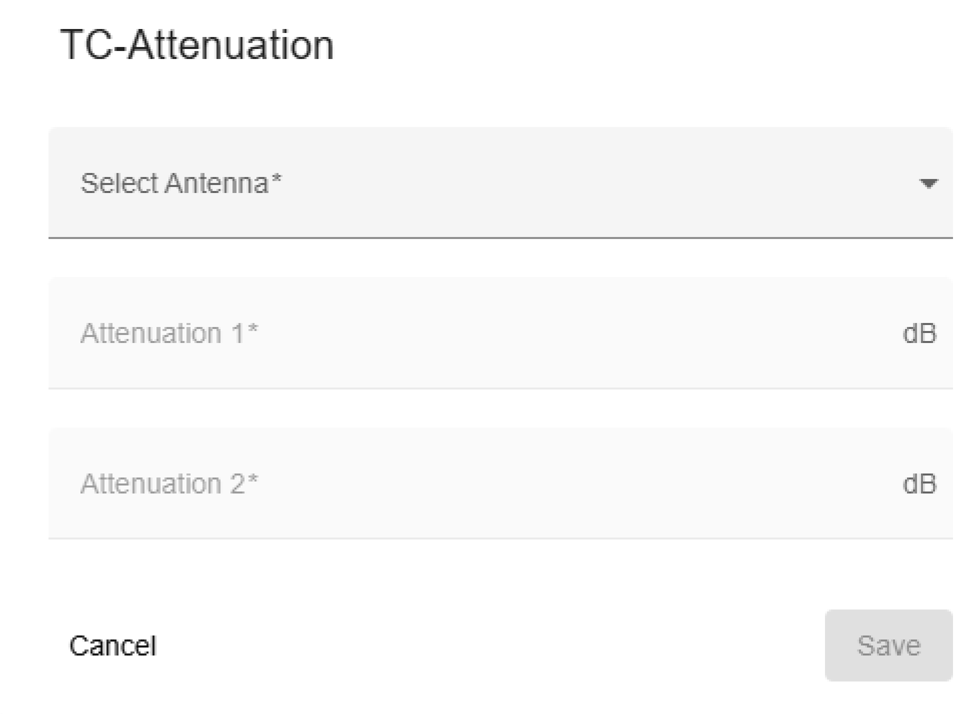
- **Trigger**:
    - when you click on the "+" button on the top toolbar to create new or
    - an on *pencil icon* to edit TC channel in the menu column
    - The dialog title clearly indicates whether you are creating a new, editing an existing one, ensuring clarity in the action being performed.
-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

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

- **Actions**:

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

Modify TC channel attenuation


TC-Attenuation

- **Trigger**: Click on the *pencil outlined* button in the row under *attenuation*

- **Fields**:

  - **Select antenna**: Drop-down to select antenna.
  - **Attenuation 1**: Transmit attenuation1, floating point, dB.
  - **Attenuation 2**: Transmit attenuation2, floating point, dB.

- **Actions**:

  - **Save**:
    - Saves modifications.
    - If an error occurs in the input fields, the button is deactivated.

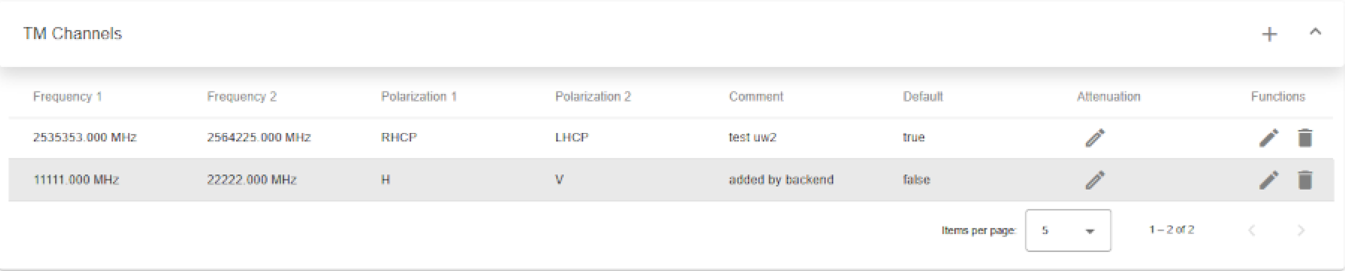
  - **Cancel**: Closes the dialog without saving.

On delete confirmation dialog

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

2.12.3.4 TM channel table

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



TM Channels

Frequency 1	Frequency 2	Polarization 1	Polarization 2	Comment	Default	Attenuation	Functions
2535353.000 MHz	2564225.000 MHz	RHCP	LHCP	test uw2	true	✎	✎ 🗑
11111.000 MHz	22222.000 MHz	H	V	added by backend	false	✎	✎ 🗑

Top

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

Header row

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

Table columns

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

Bottom toolbar

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

2.12.3.4.1 Functionality

Create TM channel

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

Edit TC channel

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

Attenuation

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

Delete TC channel

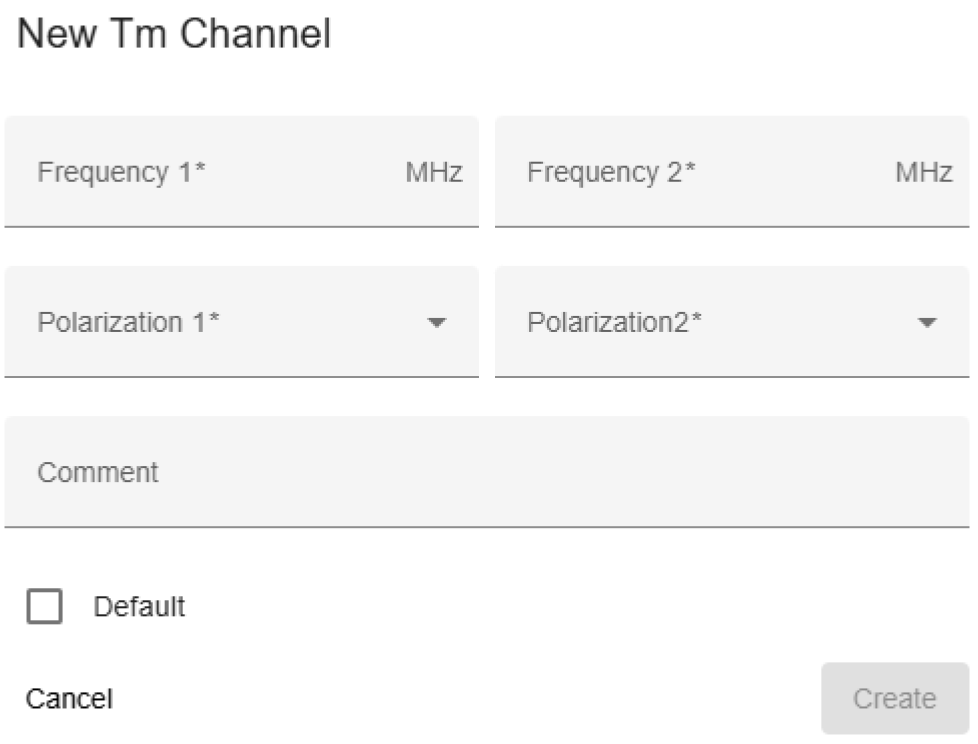
- Click the *trash icon* in the Functions column to delete an TC channel.
- Confirm the action in the pop-up dialog to delete the TC channel.

Pagination

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

2.12.3.4.2 Dialogs

Add/edit TC channel dialog



New Tm Channel

Frequency 1* MHz

Frequency 2* MHz

Polarization 1*

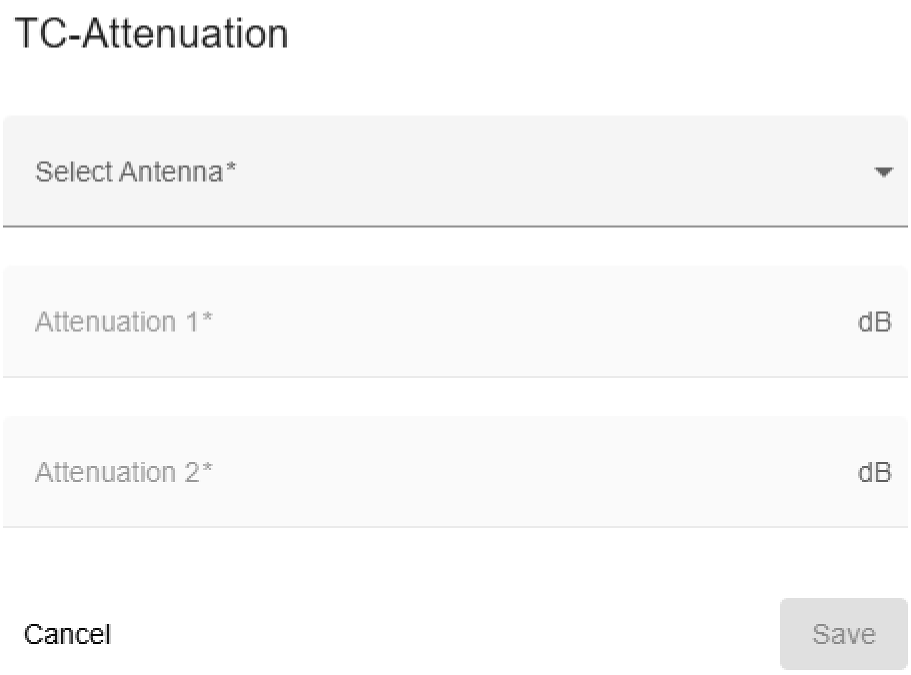
Polarization2*

Comment

□ Default

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

Modify TM channel attenuation



TC-Attenuation

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

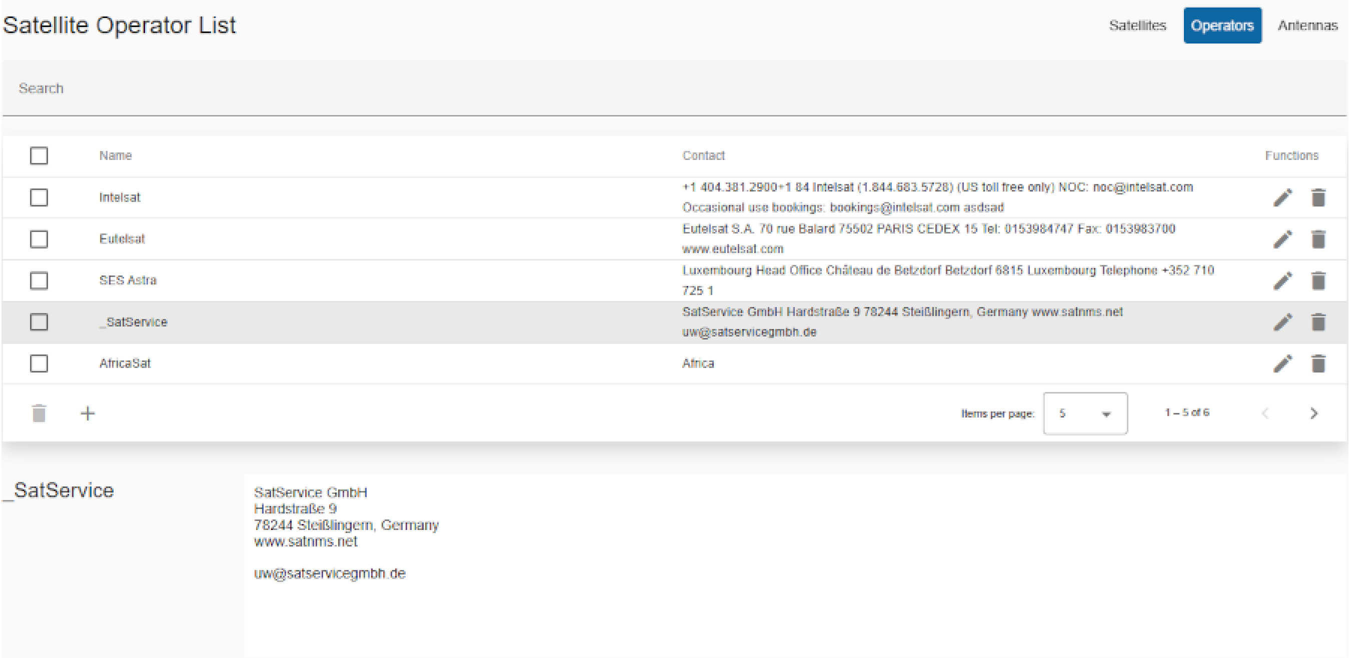
On delete confirmation dialog

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

2.13 Satellite operator list

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

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



□	Name	Contact	Functions
□	Intelsat	+1 404.381.2900+1 84 Intelsat (1.844.683.5728) (US toll free only) NOC: noc@intelsat.com Occasional use bookings: bookings@intelsat.com asdsad	
□	Eutelsat	Eutelsat S.A. 70 rue Balard 75502 PARIS CEDEX 15 Tel: 0153984747 Fax: 0153983700 www.eutelsat.com	
□	SES Astra	Luxembourg Head Office Château de Betzdorf Betzdorf 6815 Luxembourg Telephone +352 710 725 1	
□	_SatService	SatService GmbH Hardstraße 9 78244 Steißlingern, Germany www.satnms.net uw@satservicegmbh.de	
□	AfricaSat	Africa	

_SatService

SatService GmbH
Hardstraße 9
78244 Steißlingern, Germany
www.satnms.net

uw@satservicegmbh.de

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

Menu options:

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

Search bar

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

Header row

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

Table columns

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

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

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

Bottom toolbar

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

Details of selected operator:

- Name of operator
- Contact details of operator

2.13.1 Functionality

Create operator

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

Edit operator

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

Delete operator

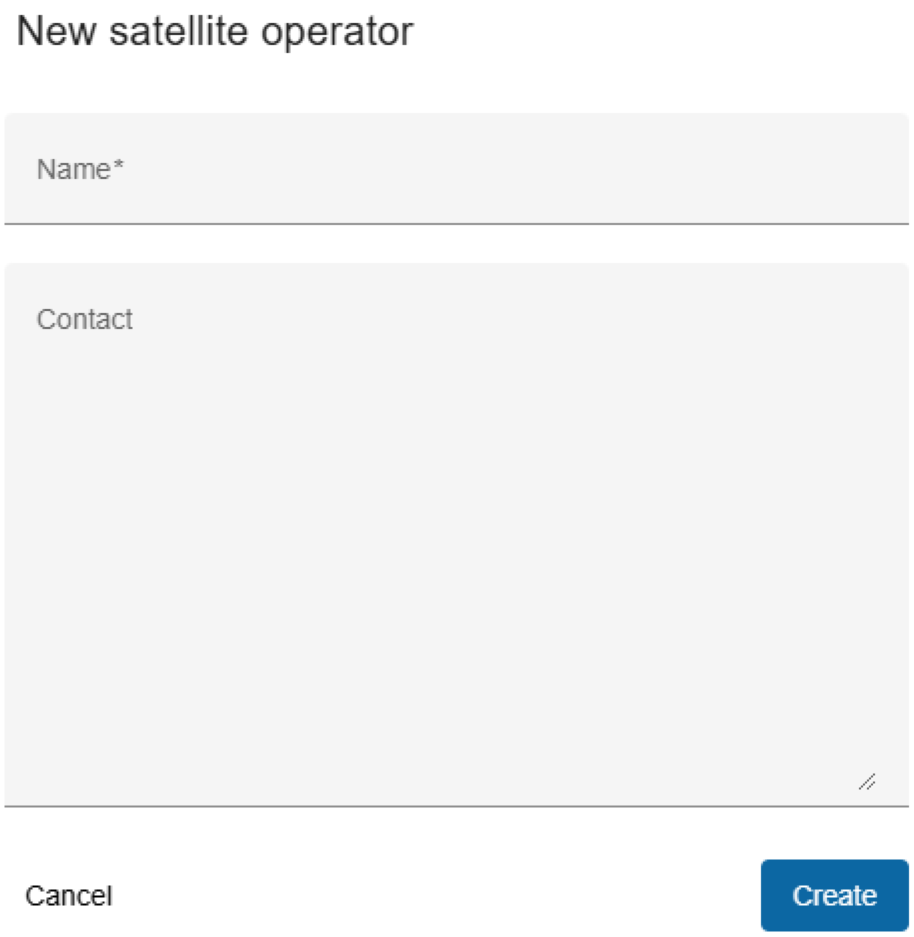
- Click the *trash icon* in the Functions column to delete an operator.
- Confirm the action in the pop-up dialog to delete the operator.

Pagination

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

2.13.2 Dialogs

Add/edit satellite dialog



New satellite operator

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

On delete confirmation dialog

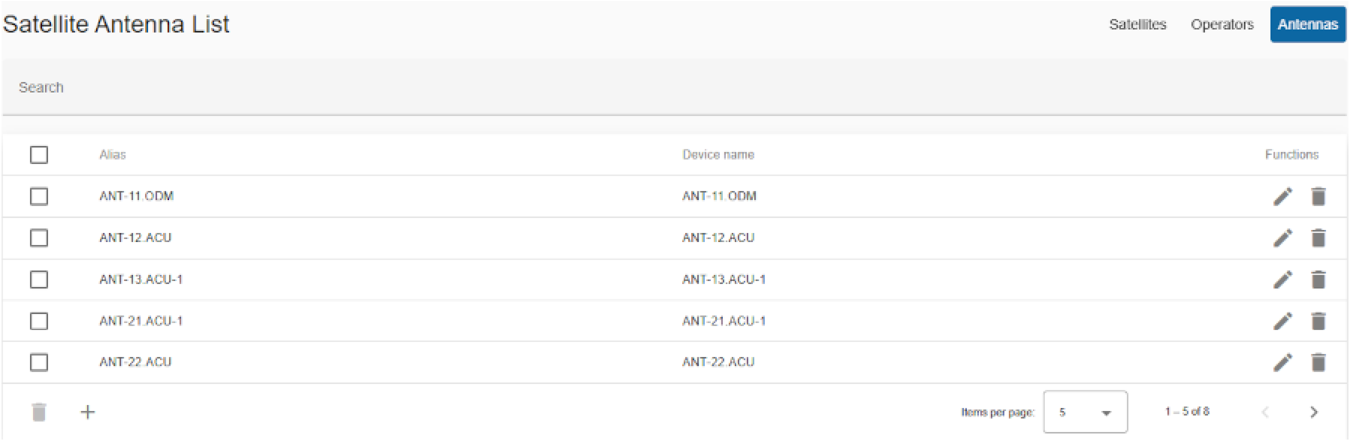
- **Trigger**: Opens when you click the trash icon for an operator from menu column or trash icon from bottom toolbar .
- **Content**: Asks the user to confirm whether they want to delete the selected operator(s).
- **Actions**:
  - **Yes**: Deletes the operator(s) permanently.
  - **Cancel**: Cancels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

2.14 Antenna list

All saved antenna in database.

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

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



Satellite Antenna List

	Alias	Device name	Functions
☐	ANT-11.ODM	ANT-11.ODM	✎ 🗑
☐	ANT-12.ACU	ANT-12.ACU	✎ 🗑
☐	ANT-13.ACU-1	ANT-13.ACU-1	✎ 🗑
☐	ANT-21.ACU-1	ANT-21.ACU-1	✎ 🗑
☐	ANT-22.ACU	ANT-22.ACU	✎ 🗑

Items per page: 5 1 – 5 of 8

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

Menu options:

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

Search bar

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

Header row

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

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

Table columns

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

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

Bottom toolbar

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

2.14.1 Functionality

Create antenna

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

Edit antenna

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

Delete antenna

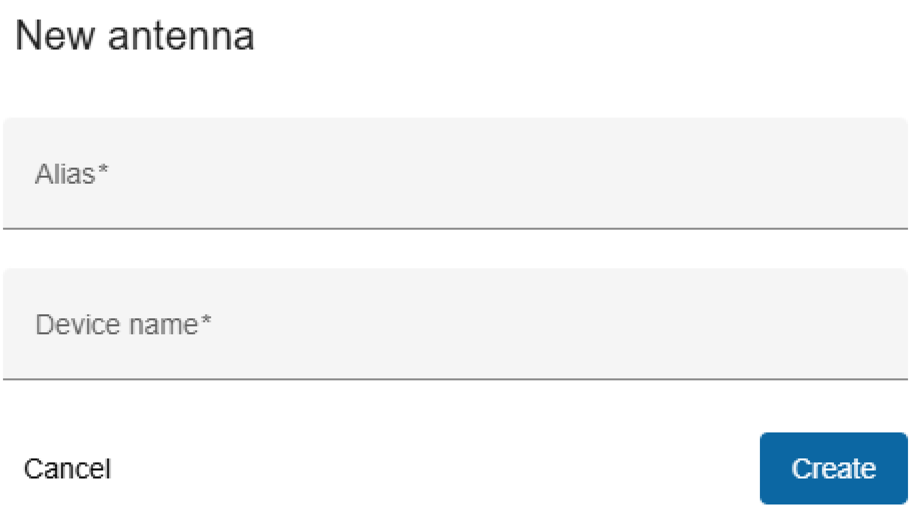
- Click the *trash icon* in the Functions column to delete an antenna.
- Confirm the action in the pop-up dialog to delete the antenna.

Pagination

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

2.14.2 Dialogs

Add/edit satellite dialog



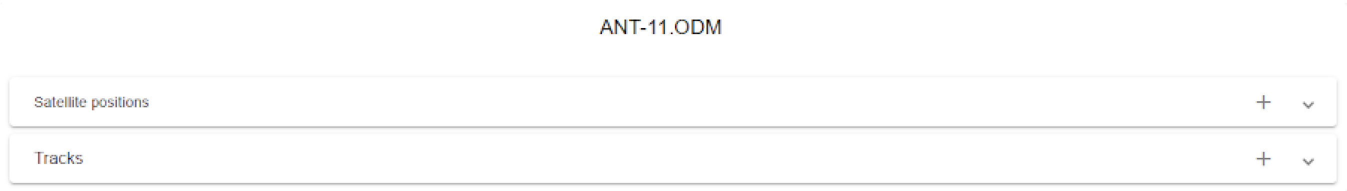
- **Trigger**:
  - when you click on the "+" button on the bottom toolbar to create new antenna or
  - an on *pencil icon* to edit in the menu column
  - The dialog title clearly indicates whether you are creating a new antenna, editing an existing one, ensuring clarity in the action being performed.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

On delete confirmation dialog

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

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

2.14.3 Antenna related tables



ANT-11.ODM

Satellite positions + ⌄

Tracks + ⌄

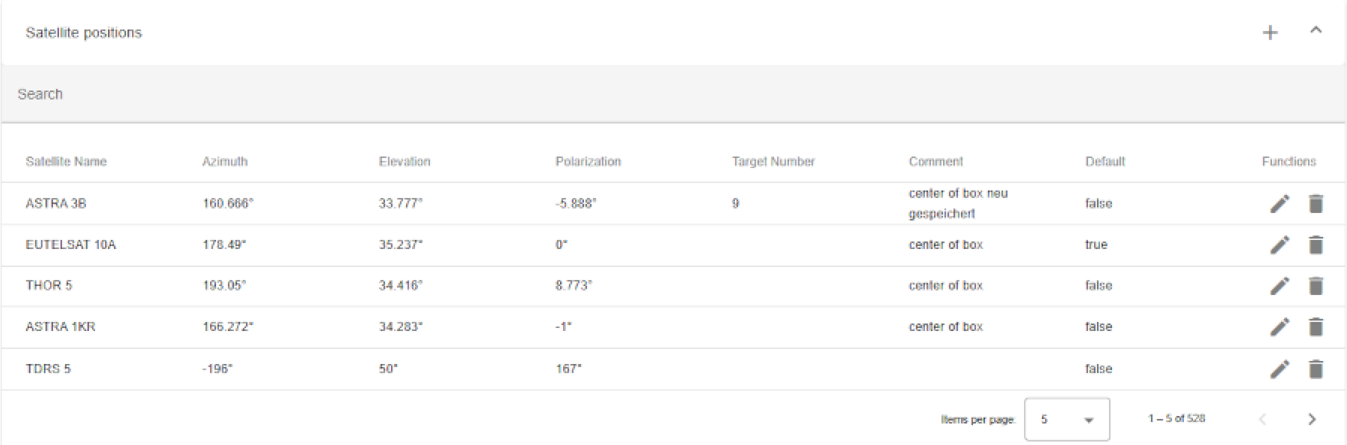
Title: Antenna name

Tables:

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

2.14.3.1 Satellite position table

The list of all satellite positions for the given antenna.



Satellite positions + ^

Search

Satellite Name	Azimuth	Elevation	Polarization	Target Number	Comment	Default	Functions
ASTRA 3B	160.666°	33.777°	-5.888°	9	center of box neu gespeichert	false	✎ 🗑
EUTELSAT 10A	178.49°	35.237°	0°		center of box	true	✎ 🗑
THOR 5	193.05°	34.416°	8.773°		center of box	false	✎ 🗑
ASTRA 1KR	166.272°	34.283°	-1°		center of box	false	✎ 🗑
TDRS 5	-196°	50°	167°			false	✎ 🗑

Items per page: 5 ▾ 1 – 5 of 528 ‹ ›

Top

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

Search bar

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

Header row

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

Table columns

Accessing the table from satellites page:

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

Accessing the table from antennas page:

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

Common columns:

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

Bottom toolbar

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

2.14.3.1.1 Functionality

Create satellite position

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

Edit satellite position

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

Delete satellite position

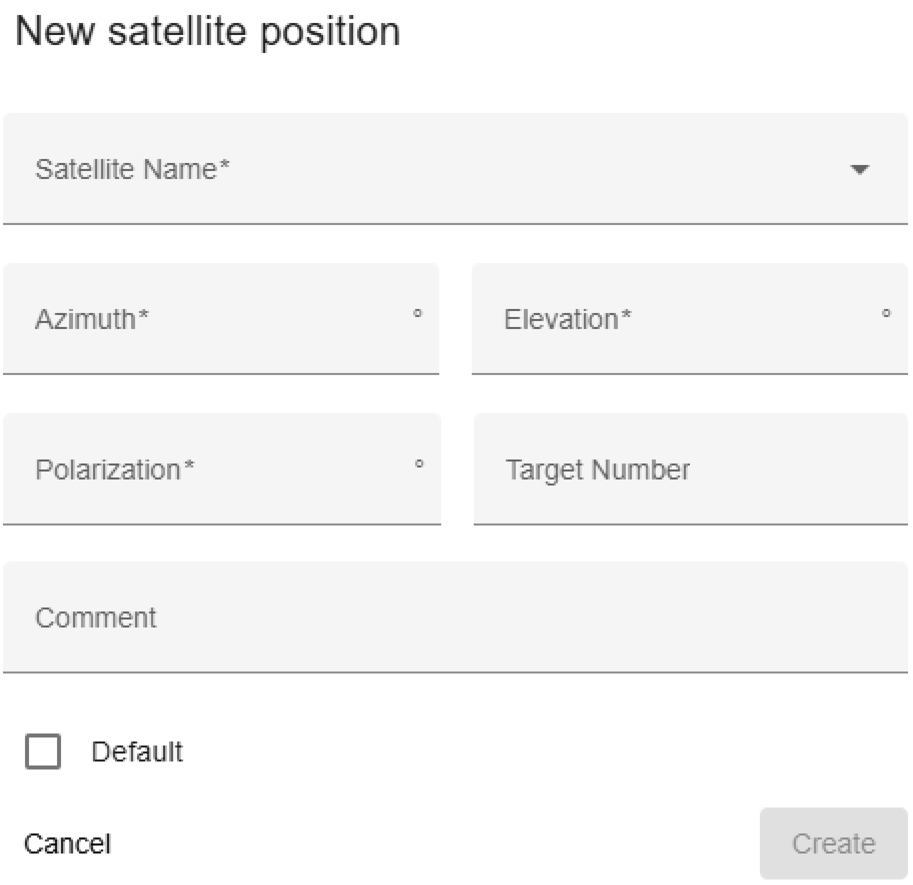
- Click the *trash icon* in the Functions column to delete an satellite position.
- Confirm the action in the pop-up dialog to delete the satellite position.

Pagination

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

2.14.3.1.2 Dialogs

Add/edit satellite position dialog



New satellite position

Satellite Name*

Azimuth* °

Elevation* °

Polarization* °

Target Number

Comment

□ Default

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

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

- **Actions**:

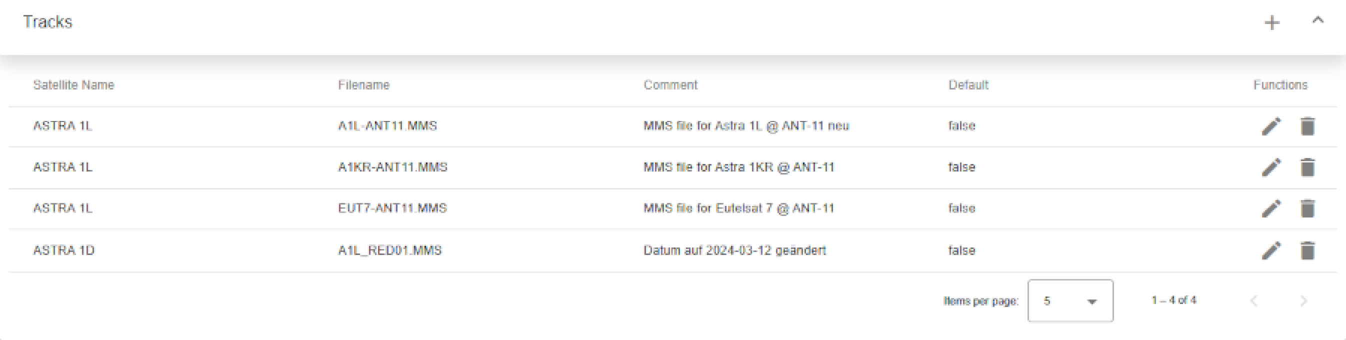
    - **Create/Save**:
        - For new position, this creates the satellite position with the provided details.
        - For existing position, this saves modifications.
        - If an error occurs in the input fields, the button is deactivated.
    - **Cancel**: Closes the dialog without saving.

On delete confirmation dialog

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

2.14.3.2 Tracks table

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



Tracks

Satellite Name	Filename	Comment	Default	Functions
ASTRA 1L	A1L-ANT11.MMS	MMS file for Astra 1L @ ANT-11 neu	false	✎ 🗑
ASTRA 1L	A1KR-ANT11.MMS	MMS file for Astra 1KR @ ANT-11	false	✎ 🗑
ASTRA 1L	EUT7-ANT11.MMS	MMS file for Eutelsat 7 @ ANT-11	false	✎ 🗑
ASTRA 1D	A1L_RED01.MMS	Datum auf 2024-03-12 geändert	false	✎ 🗑

Top

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

Header row

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

near the column name.

Table columns

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

Bottom toolbar

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

2.14.3.2.1 Functionality

Create track

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

Edit track

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

Delete track

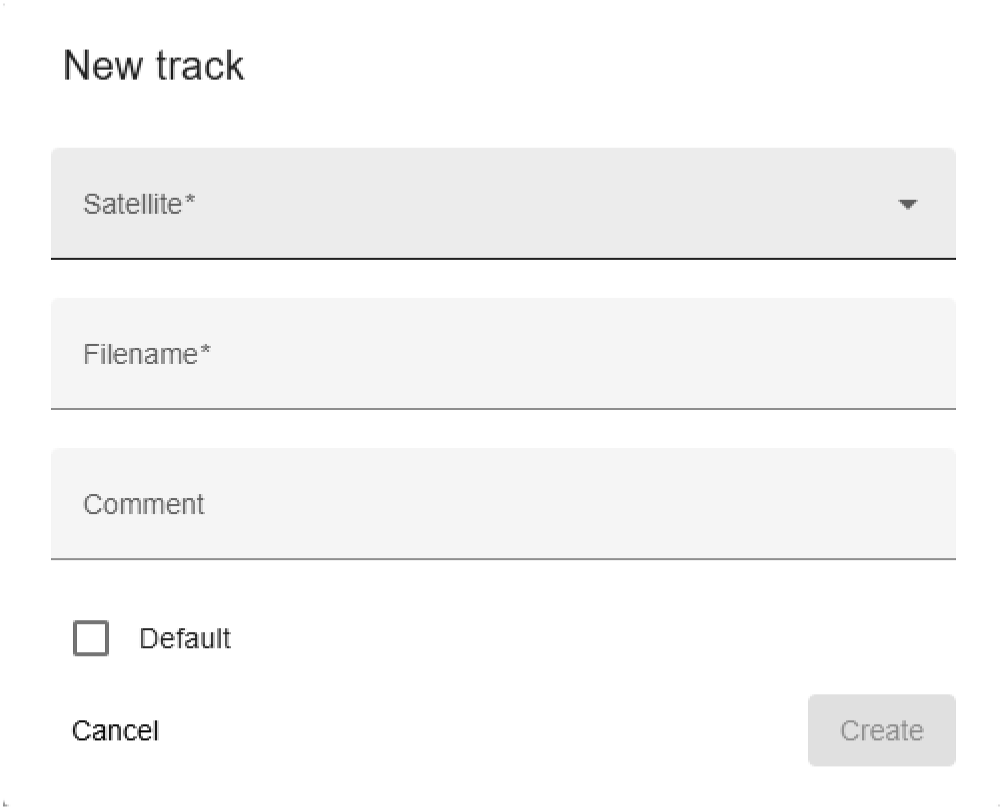
- Click the *trash icon* in the Functions column to delete an track.
- Confirm the action in the pop-up dialog to delete the track.

Pagination

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

2.14.3.2.2 Dialogs

Add/edit track dialog



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

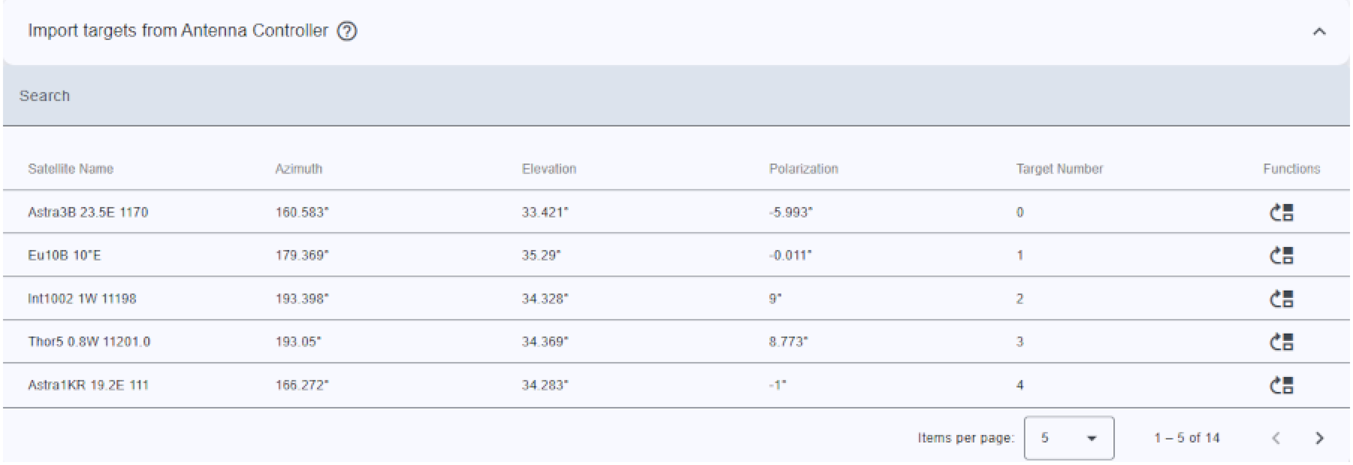
On delete confirmation dialog

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

- Cancel: Cancels the deletion action.

2.14.3.3 Imports table

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



Import targets from Antenna Controller ?

Search

Satellite Name	Azimuth	Elevation	Polarization	Target Number	Functions
Astra3B 23.5E 1170	160.583°	33.421°	-5.993°	0	
Eu10B 10°E	179.369°	35.29°	-0.011°	1	
Int1002 1W 11198	193.398°	34.328°	9°	2	
Thor5 0.8W 11201.0	193.05°	34.369°	8.773°	3	
Astra1KR 19.2E 111	166.272°	34.283°	-1°	4	

Items per page: 5

1 – 5 of 14

Top

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

Search bar

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

Header row

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

Table columns

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

Bottom toolbar

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

2.14.3.3.1 Functionality

Import satellite position

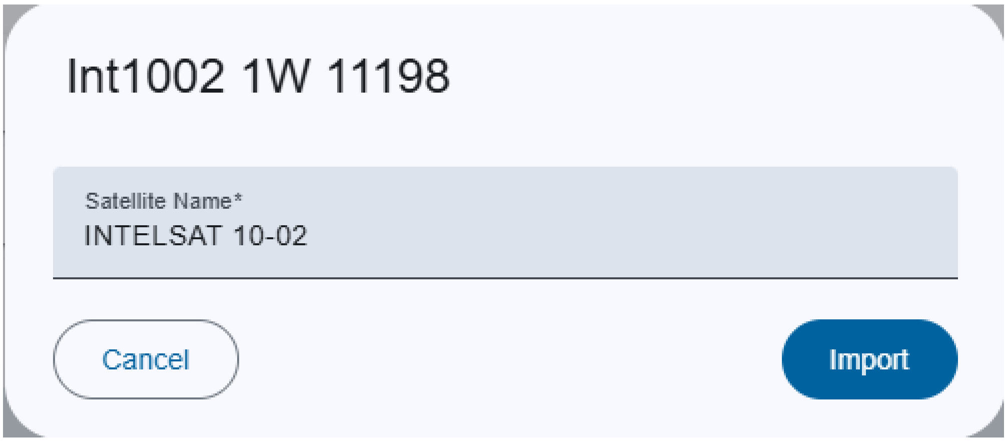
- Click the ⇄ icon opens a dialog to import values.

Pagination

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

2.14.3.3.2 Dialogs

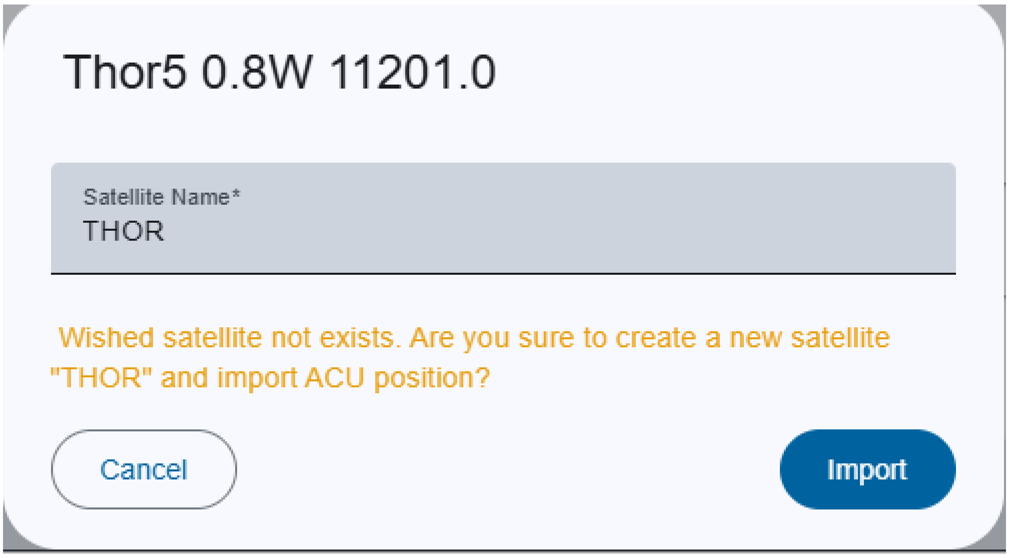
Import satellite position



Int1002 1W 11198

Satellite Name*
INTELSAT 10-02

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

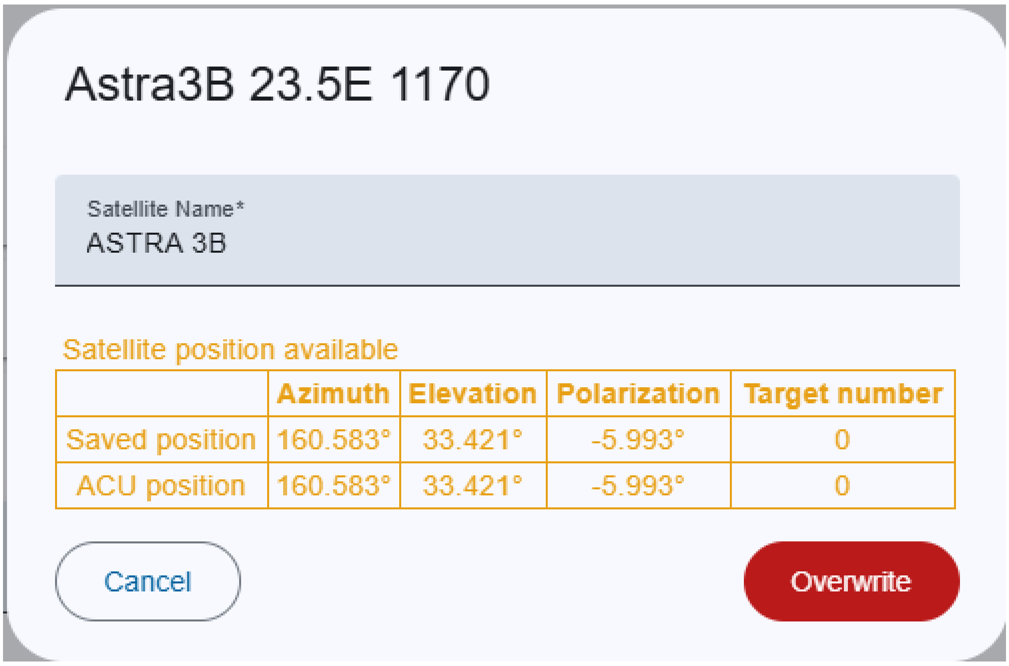


Thor5 0.8W 11201.0

Satellite Name*
THOR

Wished satellite not exists. Are you sure to create a new satellite "THOR" and import ACU position?

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



Astra3B 23.5E 1170

Satellite Name*
ASTRA 3B

Satellite position available

	Azimuth	Elevation	Polarization	Target number
Saved position	160.583°	33.421°	-5.993°	0
ACU position	160.583°	33.421°	-5.993°	0

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

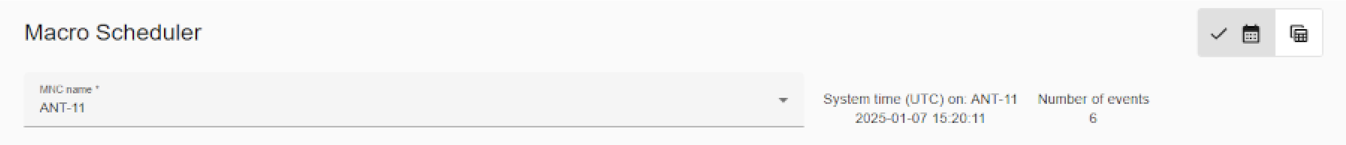
2.15 The Macro Scheduler

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

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

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

2.15.1 Scheduler



Macro Scheduler

MNC name *
ANT-11

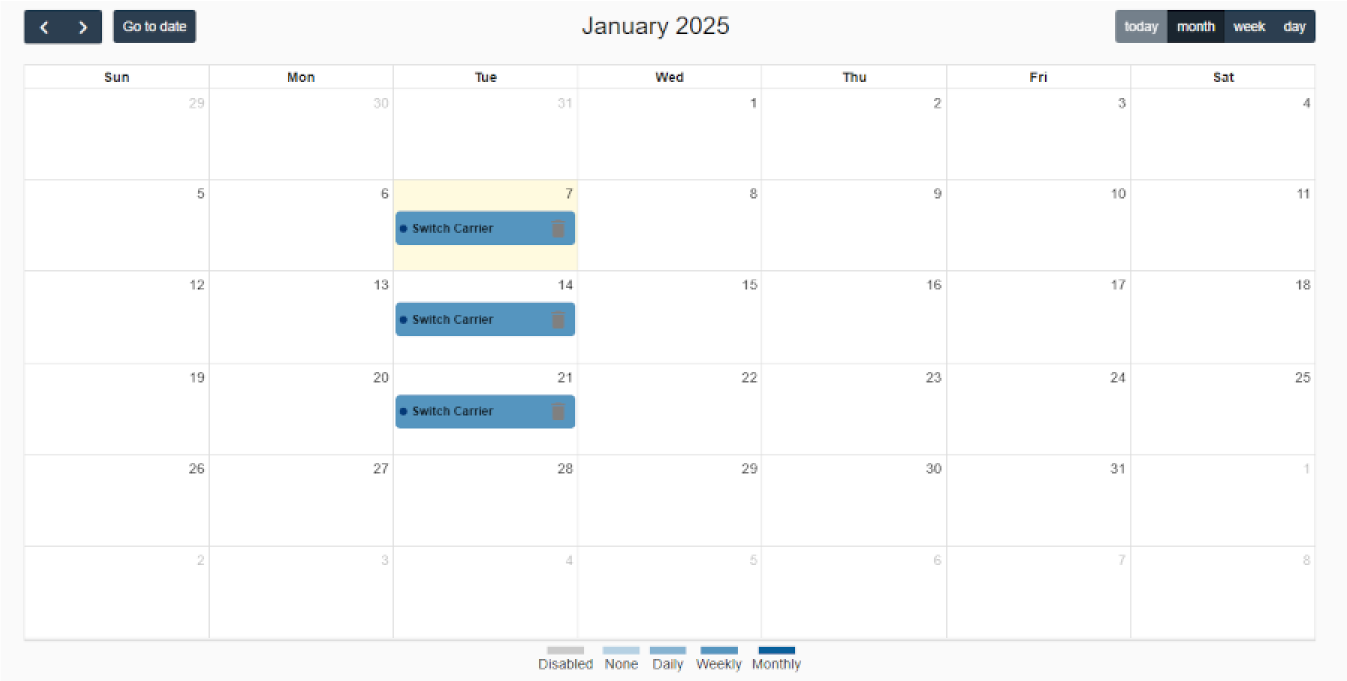
System time (UTC) on: ANT-11
2025-01-07 15:20:11

Number of events
6

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

2.15.2 Calender view

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



Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	1	2	3	4
5	6	7 Switch Carrier	8	9	10	11
12	13	14 Switch Carrier	15	16	17	18
19	20	21 Switch Carrier	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Navigation bar (top)

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

Current date

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

View selection buttons

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

Calender

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

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

Color-coded legend (bottom)

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

2.15.2.1 Features

Create event

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

Edit event

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

Delete event

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

Drag and drop rescheduling

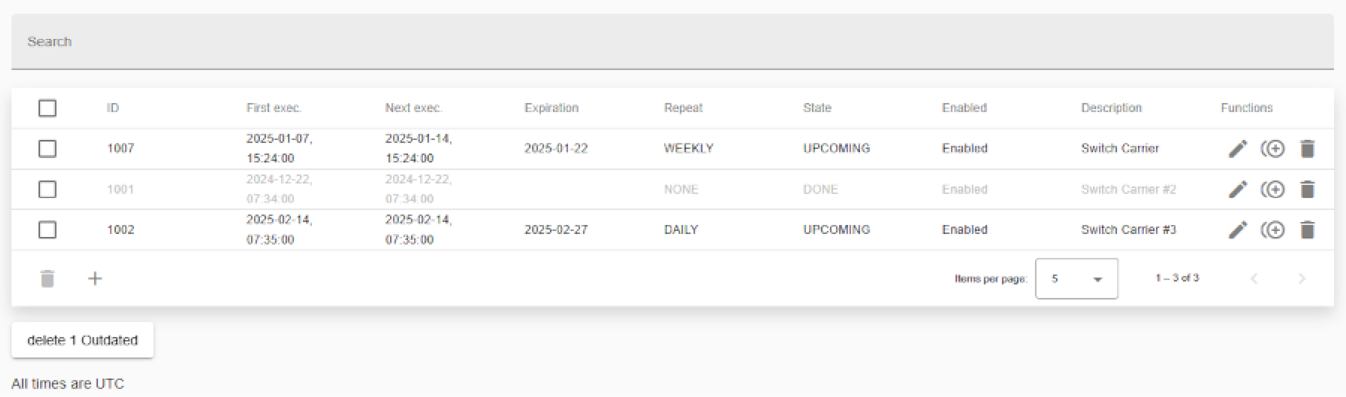
- Drag an event from one date to another to reschedule it.
- A confirmation dialog will appear to validate the changes.

Time frame navigation

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

2.15.3 Table view

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



Search

□	ID	First exec.	Next exec.	Expiration	Repeat	State	Enabled	Description	Functions
□	1007	2025-01-07, 15:24:00	2025-01-14, 15:24:00	2025-01-22	WEEKLY	UPCOMING	Enabled	Switch Carrier	
□	1001	2024-12-22, 07:34:00	2024-12-22, 07:34:00		NONE	DONE	Enabled	Switch Carrier #2	
□	1002	2025-02-14, 07:35:00	2025-02-14, 07:35:00	2025-02-27	DAILY	UPCOMING	Enabled	Switch Carrier #3	

All times are UTC

Search bar

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

Header row

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

Table columns

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

Bottom toolbar

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

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

2.15.3.1 Features

Create event

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

Edit event

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

Duplicate event

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

Delete event

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

Delete outdated events

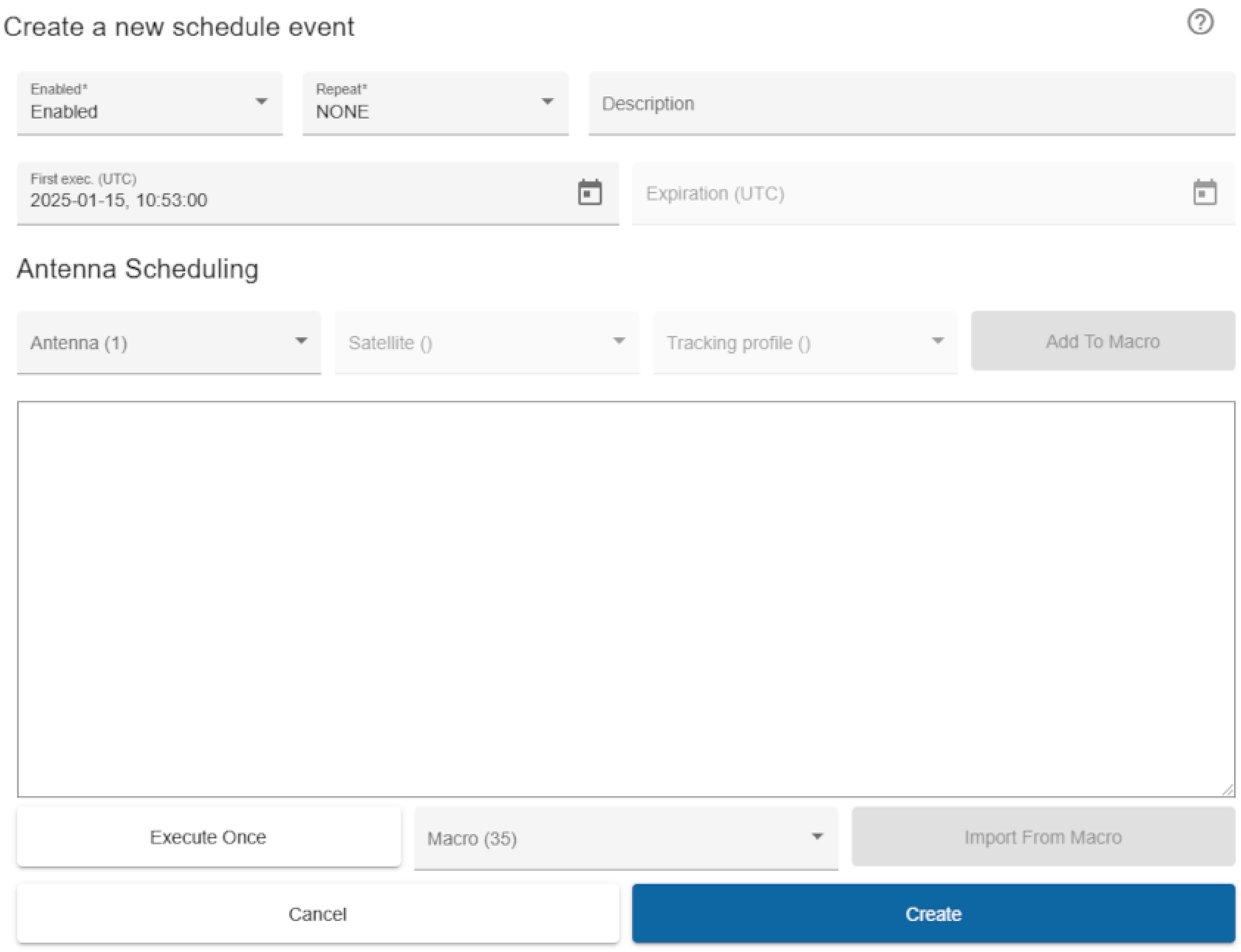
- Use the *Delete Outdated* button to remove all events that are no longer relevant or expired.
- Number shows that how much outdated events are there.
- A confirmation dialog will appear to with outdated event(s) details to perform action.

Pagination

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

2.15.4 Unified dialogs

Schedule-event dialog


Create a new schedule event

Enabled*: Enabled

Repeat*: NONE

Description

First exec. (UTC): 2025-01-15, 10:53:00

Expiration (UTC)

Antenna Scheduling

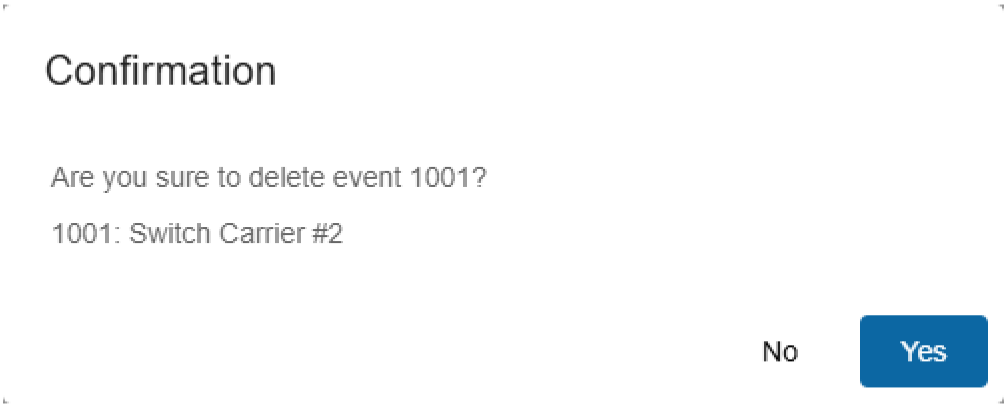
- **Trigger**:
  - when you click on an empty date in the Calendar View or the "+" button in the Table View (to create a new event) or
  - an existing event (to edit) in either the Calendar View on click on an event or in the Table View on *pencil icon* (to edit) or *circular arrow* (to duplicate) in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

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

- **Actions**:

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

On delete event confirmation dialog



Confirmation

Are you sure to delete event 1001?

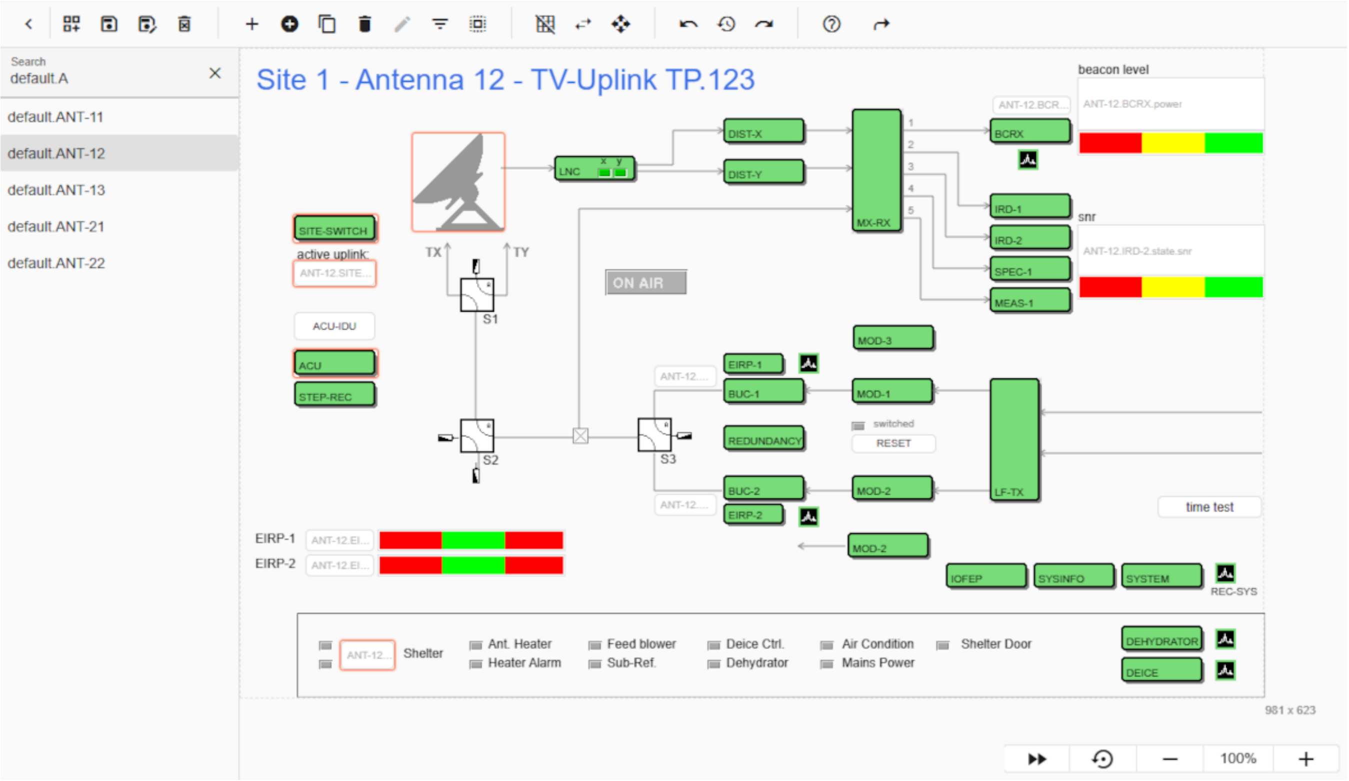
1001: Switch Carrier #2

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

2.16 Screen editor

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

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

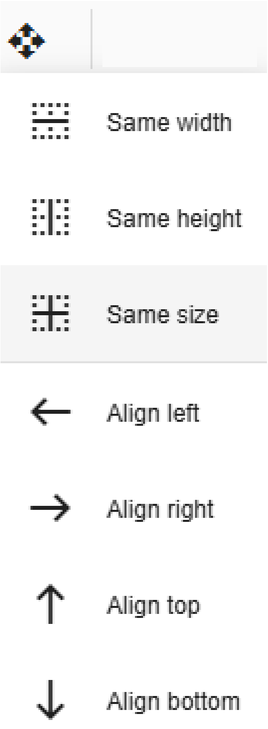


Toolbar

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

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

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

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

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

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

Screen List

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

Editing area

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

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

2.16.1 Functionality

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

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

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

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

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

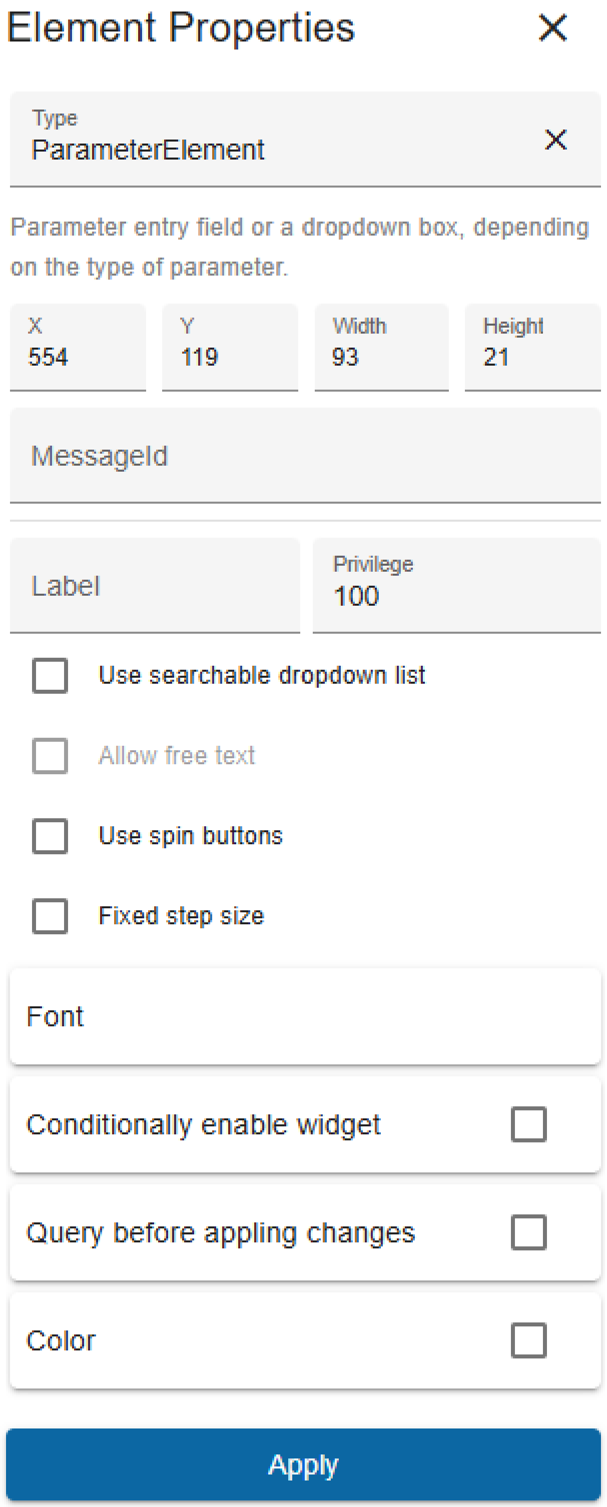
- **Key combinations**
  - **Zoom in**: By hold down ctrl key and pressing + key.
  - **Zoom out**: By hold down ctrl key and pressing - key.
  - **Default zoom (100%)**: By hold down ctrl key and pressing 0 key.
- **Mouse and key combination**: Simply hold down ctrl key and scroll with mouse wheel

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

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

2.16.2 Element properties

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



Element Properties ✕

Type
ParameterElement ✕

Parameter entry field or a dropdown box, depending on the type of parameter.

X	Y	Width	Height
554	119	93	21

MessageId

Label

Privilege
100

□ Use searchable dropdown list

□ Allow free text

□ Use spin buttons

□ Fixed step size

Font

Conditionally enable widget □

Query before appling changes □

Color □

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

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

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

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

2.16.2.1 Display element types and their properties

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

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

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

2.16.2.1.1 Button element

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

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

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

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

- **Conditionally enable widget**: Parameter fields may be locked to read only state unless another parameter matches a given value. For instance, changing the position of a wave guide switch may be inhibited while the carrier is switched on. Check this to activate this feature.
    - **Enable ID**: This field defines the ID for the enabling parameter.
    - **Enable value**: The value belonging to the parameter above. The actual value of the parameter
- **Color**: Checking this mark enables the variable background color feature for this screen element. Depending on the actual value of the variable addressed by *color id*, the element's background is set using the translation table. If the actual value of the color id variable does not match any of the table entries or if no color id is specified, the first color in the list is used to draw the element's background.The buttons Add/Set/Delete along with the value and color fields are used to edit the color translation table.
    - **Color ID**: The message ID of the parameter which controls the background color of this element.
    - **Color**: The color specification of the actually selected table entry. Clicking to this opens a color chooser dialog which lets you select/compose the the color to your needs.
    - **Value**: The parameter value of the actually selected table entry.
    - **Bold**: Draws the text bold if the parameter value matches.
    - **Add**: To add a new value / color pair to the list, fill the value and color fields.
    - **Set**: To change the color assigned to a given parameter value, first select the table entry of interest, change the color value.
    - **Delete**: To delete a value / color pair from the list, first select the table entry then click delete.

2.16.2.1.2 Parameter element

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

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

- **Message ID**: The message ID addresses the parameter the element shall display or edit. The message ID consists of the MNC name, device name followed by a colon and the name of the parameter.ANT.11.XMIT-1.tx.on for example addresses the variable tx.on at a device named XMIT-1 at MNC.
- **Label**: The label text is drawn above the element.
- **Privilege level**: The privilege level necessary to change this parameter. The predefined screens of the software use 100. In your own screens you may set certain parameters to

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

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

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

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

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

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

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

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

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

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

the parameter

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

NOTE

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

2.16.2.1.3 Display element

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

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

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

2.16.2.1.4 Radio button element

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

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

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

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

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

- **Draw frame**: Checking this checkbox makes the element show a rectangular frame around the radio buttons.

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

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

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

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

the parameter

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

2.16.2.1.5 Text element

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

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

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

2.16.2.1.6 Frame element

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

NOTE

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

2.16.2.1.7 Rect element

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

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

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

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

2.16.2.1.8 Arrow element

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

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

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

2.16.2.1.9 Icon element

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

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

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

Using an icon to display a parameter

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

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

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

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

Using an icon to display a device summary fault

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

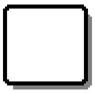
To display a device summary fault as an color changing icon, you have to provide three image files, for example:

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

2.16.2.1.10 Device icon element

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

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

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

Using customized images

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

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

Using a customized 'transmitting' emblem

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

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

2.16.2.1.11 Parameter button element

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

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

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

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

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

- **Start macro**: Selects if the button shall start a macro (checked) or send a certain parameter (unchecked).

- **Use icon instead of label**: Check this checkbox in order to make the button show an icon instead of a label text.

    - **Icon**: Provides a list of all available icons / images on the sat-nms server.

- **Specify values to send**: Checking this checkbox let you define the parameter value to be sent with a button click. Without this option the button sends the label text as a parameter value:

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

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

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

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

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

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

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

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

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

click delete.

2.16.2.1.12 Latching button element

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

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

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

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

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

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

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

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

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

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

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

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

2.16.2.1.13 Gauge element

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

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

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

show the scale labels without overlap.

2.16.2.1.14 Chart element

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

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

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

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

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

automatic reference level alignment with this mode.

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

- **Fixed thresholds**: It is like the above mode, but additionally there are min/max thresholds defined. If the actual value of the monitored parameter, the display background changes to red.
    - **min value**: The minimum value to display for the fixed range display modes.
    - **max value**: The maximum value to display for the fixed range display modes.
    - **min threshold**: The lower threshold value for the "fixed range & thresholds" mode.
    - **max threshold**: The upper threshold value for the "fixed range & thresholds" mode.

2.16.2.1.15 Spectrum element

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

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

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

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

2.16.2.1.16 Target list element

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

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

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

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

2.16.2.1.17 XY chart element

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

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

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

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

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

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

2.16.2.1.18 Thumbnail icon element

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

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

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

REMARKS

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

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

2.16.2.1.19 Switch element

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

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

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

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

Remarks

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

User defined switch icons

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

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

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

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

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

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

Replacing the predefined icons

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

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

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

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

2.16.2.1.20 AzEl element

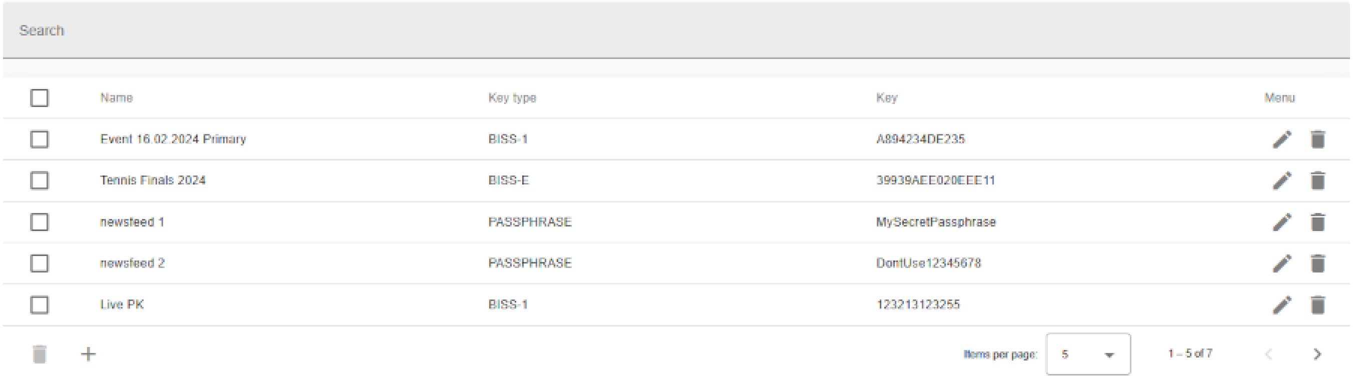
The AzElElement shows the tracking history of an antenna controller as a cloud of points in a elevation over azimuth coordinate system. Actually only SatService antenna controllers provide the tracking history data for this screen element.

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

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

2.17 Stream keys

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



Search

☐	Name	Key type	Key	Menu
☐	Event 16.02.2024 Primary	BISS-1	A894234DE235	✎ 🗑
☐	Tennis Finals 2024	BISS-E	39939AEE020EEE11	✎ 🗑
☐	newsfeed 1	PASSPHRASE	MySecretPassphrase	✎ 🗑
☐	newsfeed 2	PASSPHRASE	DontUse12345678	✎ 🗑
☐	Live PK	BISS-1	123213123255	✎ 🗑

Search bar

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

Header row

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

Table columns

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

Bottom toolbar

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

navigate between pages.

2.17.1 Features

Create key

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

Edit key

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

Delete event

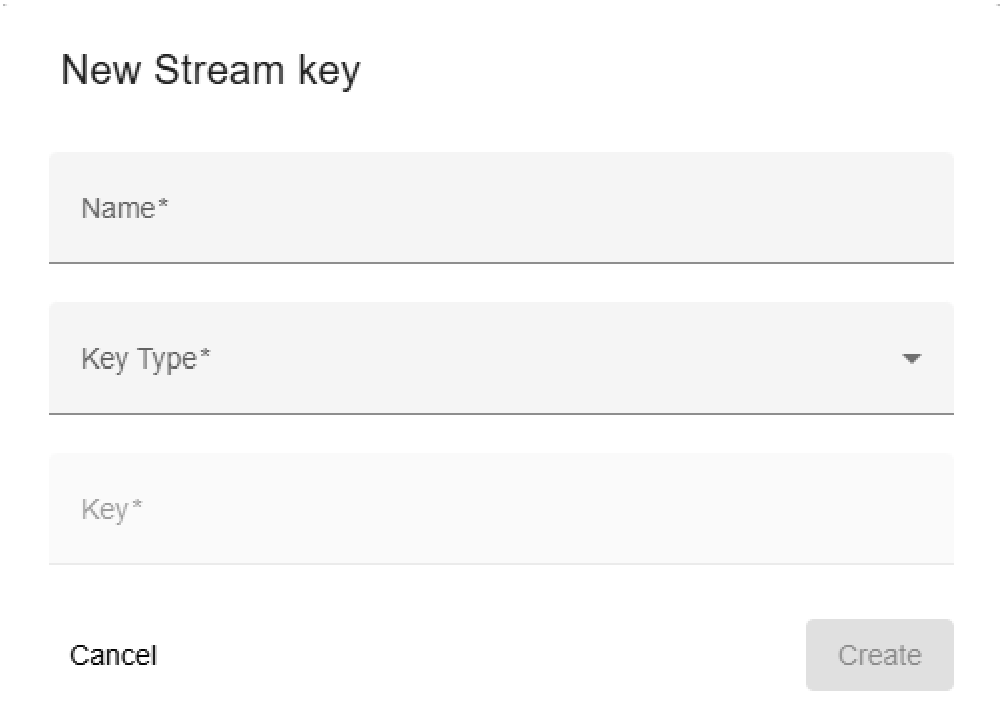
- Click the *trash icon* in the Functions column to delete an key.
- Confirm the action in the pop-up dialog to delete the key.

Pagination

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

2.17.2 Dialogs

Stream key add/edit dialog



New Stream key

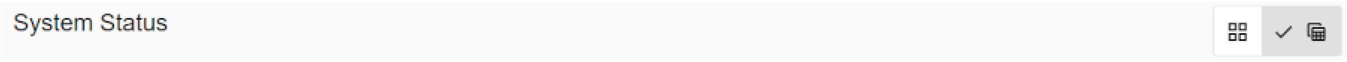
- **Trigger**:
  - when you click on the "+" button on the bottom toolbar to create new key or
  - an on *pencil icon* to edit in the menu column
  - The dialog title clearly indicates whether you are creating a new stream key, editing an existing one, ensuring clarity in the action being performed.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

On delete kex confirmation dialog

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

2.18 System status

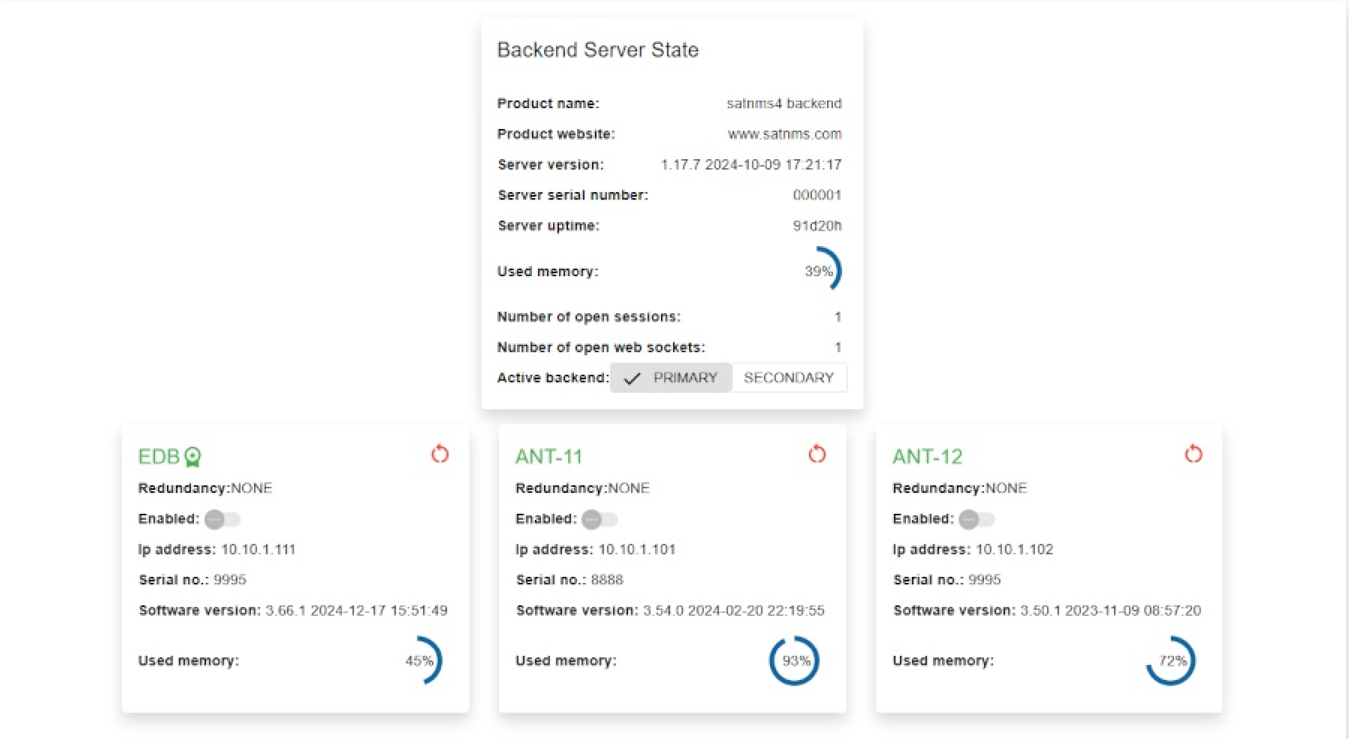
This page gives you status information about the backend server.



System Status

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

2.18.1 Grid view



Backend Server State

Product name:	satnms4 backend
Product website:	www.satnms.com
Server version:	1.17.7 2024-10-09 17:21:17
Server serial number:	000001
Server uptime:	91d20h
Used memory:	39%
Number of open sessions:	1
Number of open web sockets:	1

Active backend: ✓ PRIMARY | SECONDARY

EDB

Redundancy:NONE

Enabled: ○

Ip address: 10.10.1.111

Serial no.: 9995

Software version: 3.66.1 2024-12-17 15:51:49

Used memory: 45%

ANT-11

Redundancy:NONE

Enabled: ○

Ip address: 10.10.1.101

Serial no.: 8888

Software version: 3.54.0 2024-02-20 22:19:55

Used memory: 93%

ANT-12

Redundancy:NONE

Enabled: ○

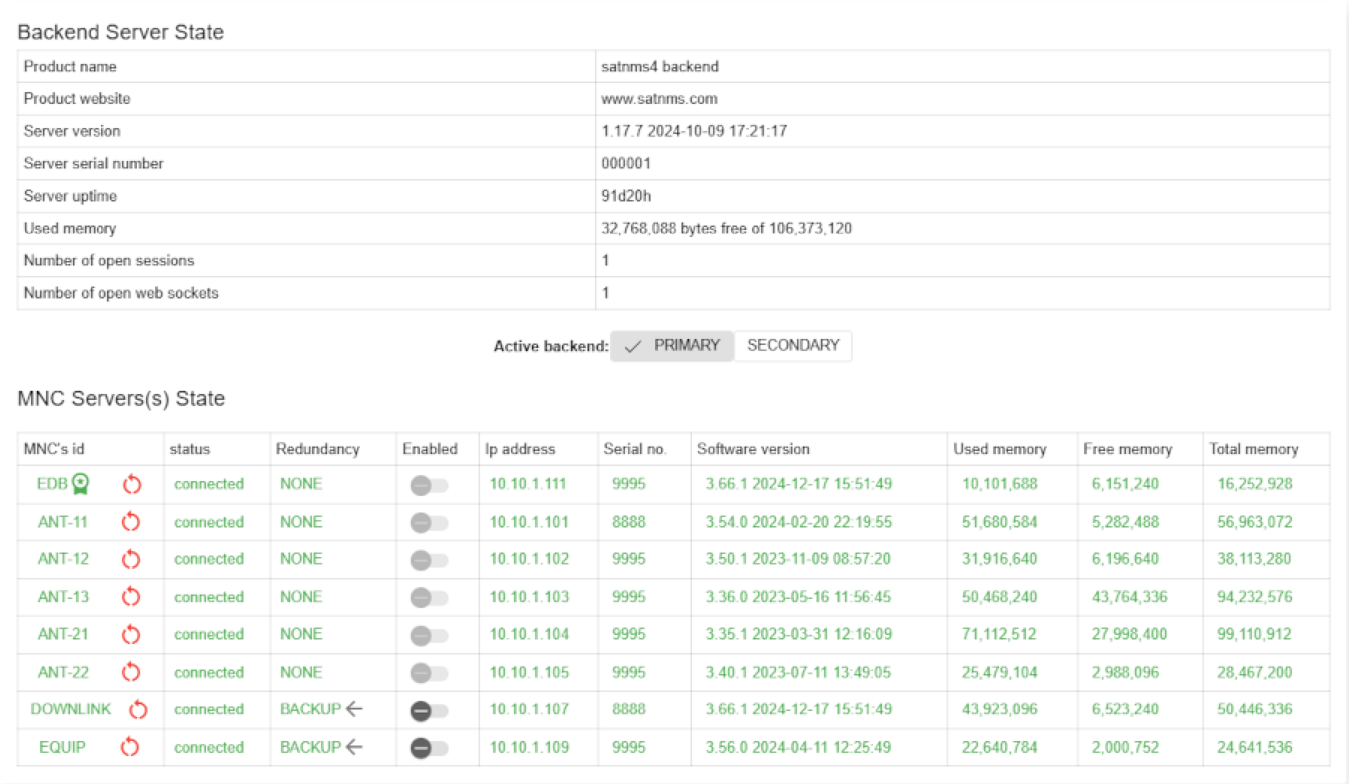
Ip address: 10.10.1.102

Serial no.: 9995

Software version: 3.50.1 2023-11-09 08:57:20

Used memory: 72%

2.18.2 Table view



Backend Server State

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

Active backend: ✓ PRIMARY | SECONDARY

MNC Servers(s) State

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

2.18.3 Common functions and details

Backend server state

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

MNC information

Below you will find information about MNC.

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

2.18.4 Features

Restart MNC

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

Active backend

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

Switch to main/backup

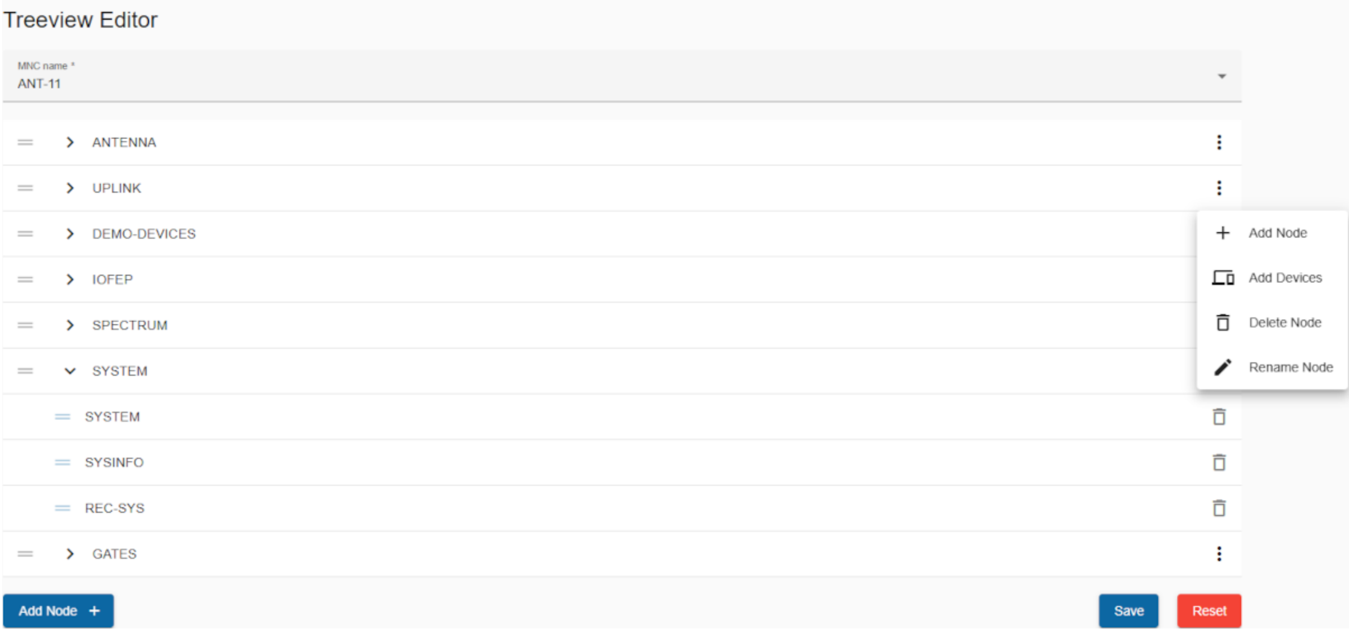
- Click on the arrow buttons (left or right) next to*Redundancy* to change MNC from *Backup to Main or Main to Backup*.

Toggle redundancy

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

2.19 Treeview editor

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



Title: Title of page.

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

Tree structure

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

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

Bottom toolbar actions

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

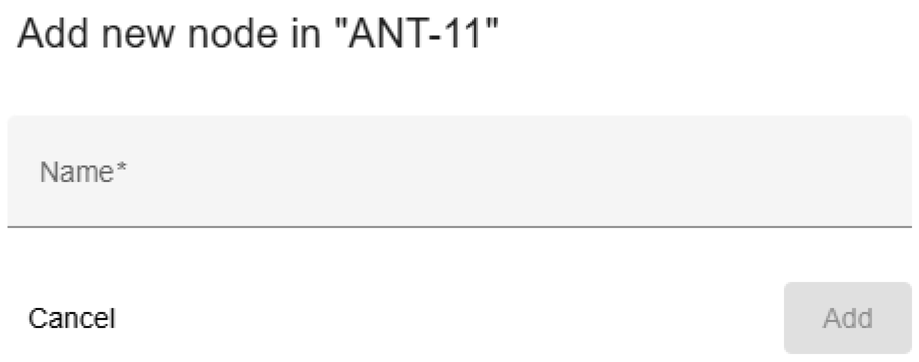
- **Add node**: Use the *Add Node +* button at the bottom of the editor opens a dialog to quickly add a new subsystem/parent node to the tree.
- **Save**: Saves all changes made in the editor. If you have not saved the changes and accidentally leave the editor, a confirmation dialog will appear in which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

2.19.1 Functionality

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

2.19.2 Dialogs

Add/rename node/subsystem

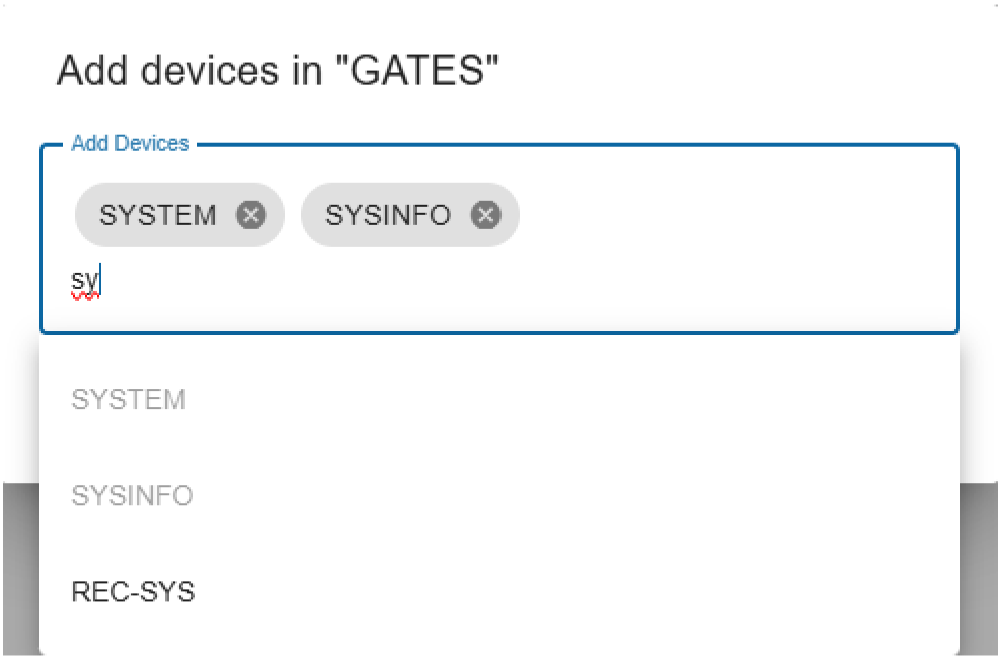


Add new node in "ANT-11"

Name*

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

Add devices



Add devices in "GATES"

Add Devices

SYSTEM ⊗ SYSINFO ⊗

sy

SYSTEM

SYSINFO

REC-SYS

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

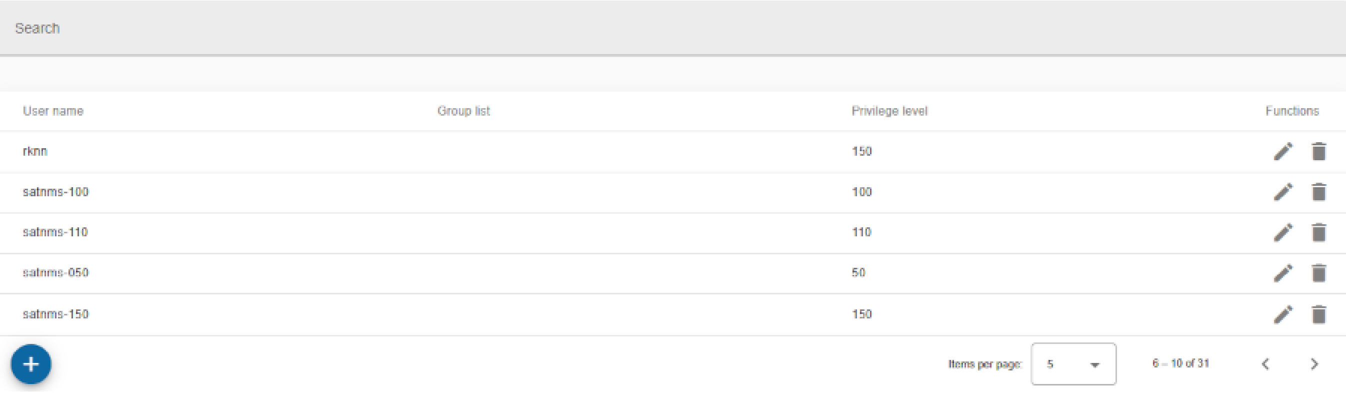
Confirmation dialog

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

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

2.20 User management

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



User name	Group list	Privilege level	Functions
rknn		150	✎ 🗑
satnms-100		100	✎ 🗑
satnms-110		110	✎ 🗑
satnms-050		50	✎ 🗑
satnms-150		150	✎ 🗑

Search bar

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

Header row

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

Table columns

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

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

Bottom toolbar

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

2.20.1 Functionality

Create user

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

Edit user

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

Delete user

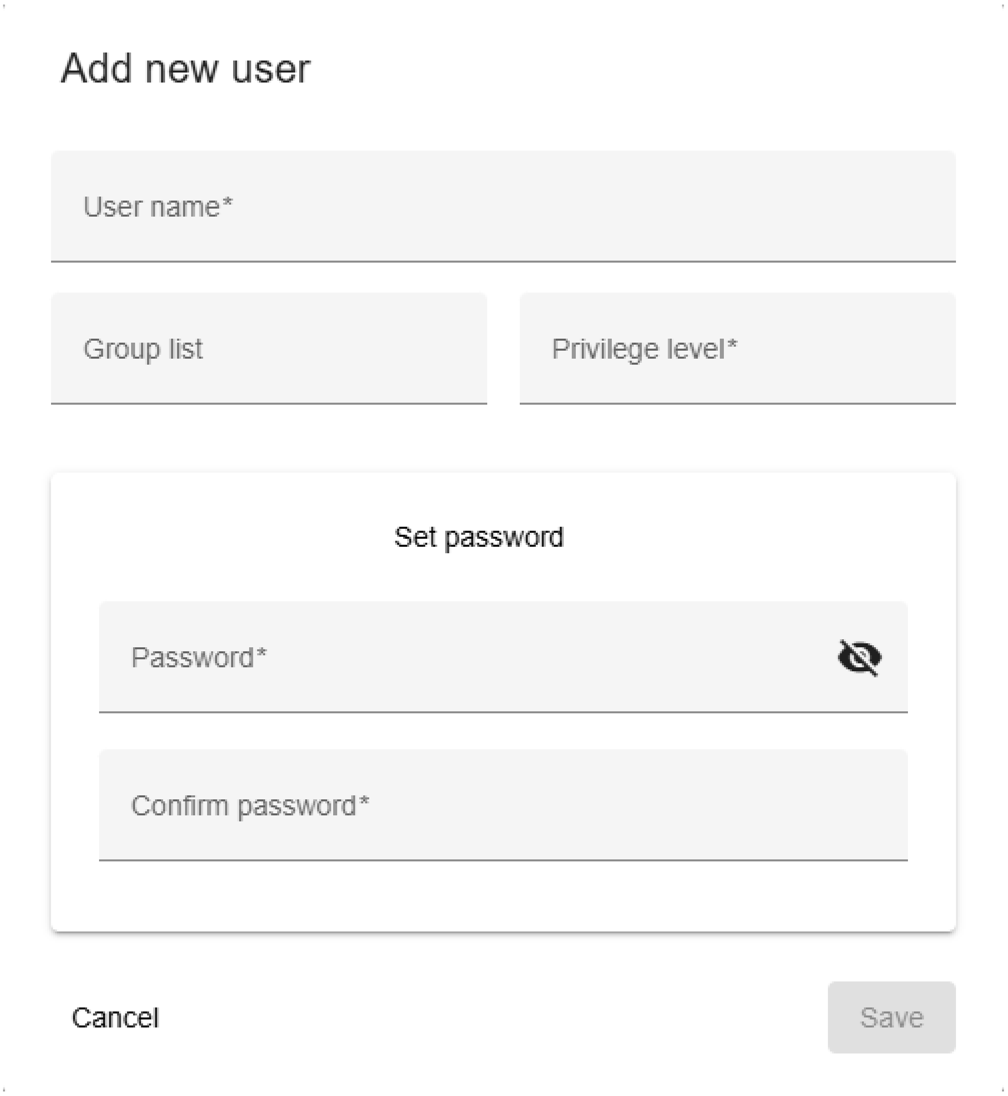
- Click the *trash icon* in the Functions column to delete an user.
- Confirm the action in the pop-up dialog to delete the user.

Pagination

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

2.20.2 Dialogs

Add/edit user dialog


Add new user

User name*

Group list

Privilege level*

Set password

Password*

Confirm password*

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

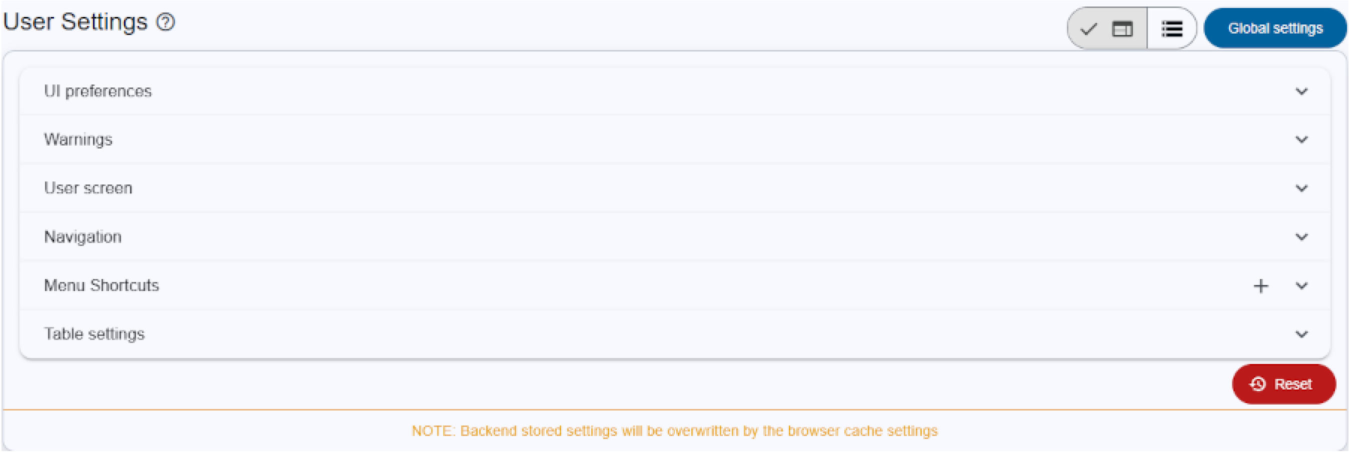
- **Actions**:
    - **Save**:
        - This saves modifications.
        - If an error occurs in the input fields, the button is deactivated.
    - **Cancel**: Closes the dialog without saving.

On delete confirmation dialog

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

2.21 User settings

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



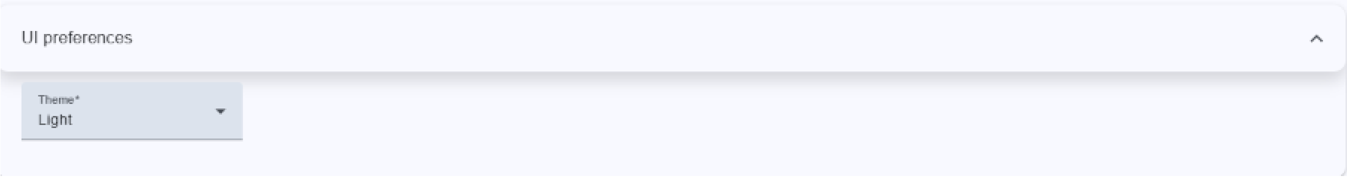
NOTE: Backend stored settings will be overwritten by the browser cache settings

Title: Title of page.

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

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

2.21.1 UI preferences

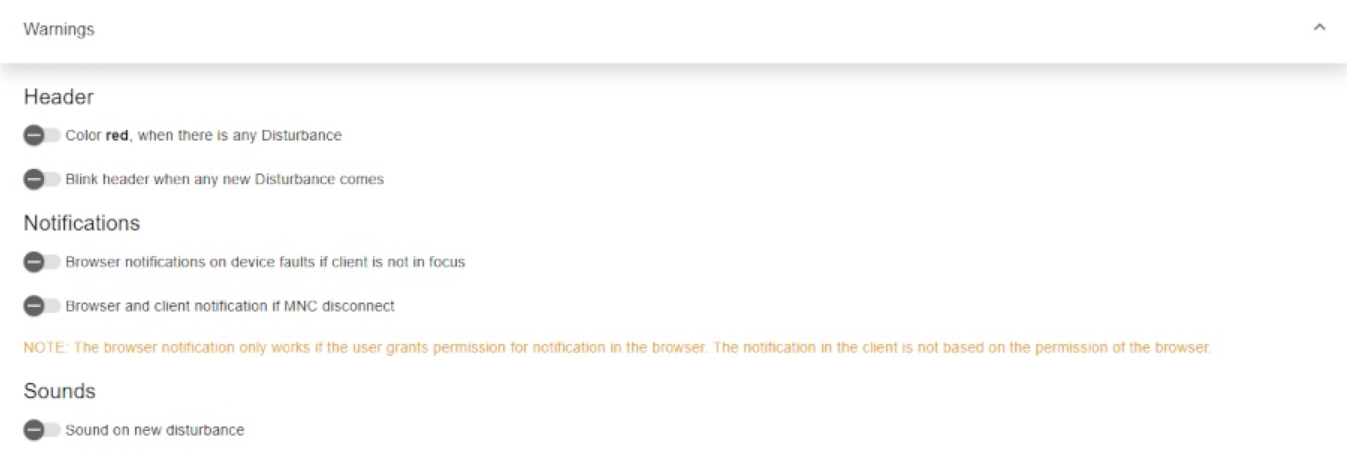


UI preferences

Theme*
Light

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

2.21.2 Warnings



Warnings

Header

Color **red**, when there is any Disturbance

Blink header when any new Disturbance comes

Notifications

Browser notifications on device faults if client is not in focus

Browser and client notification if MNC disconnect

NOTE: The browser notification only works if the user grants permission for notification in the browser. The notification in the client is not based on the permission of the browser.

Sounds

Sound on new disturbance

Top

- Section title
- Icon to expand or collapse section

Header

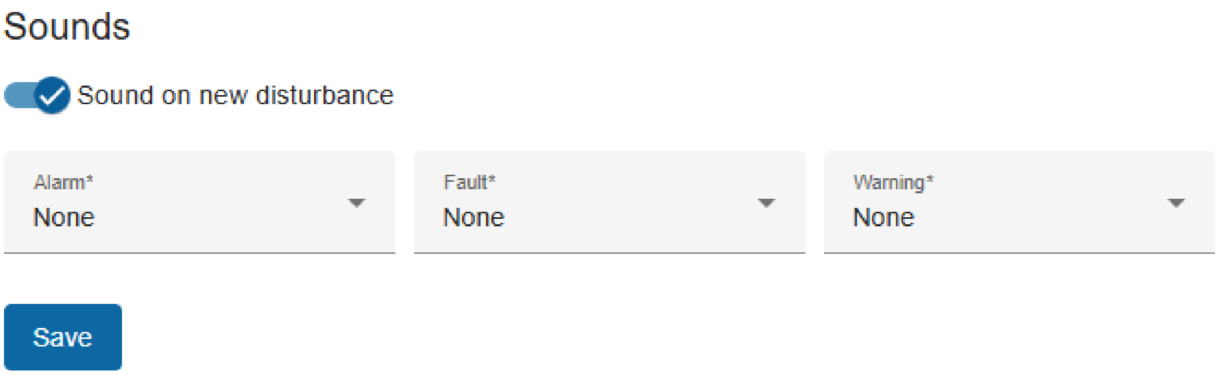
- Color red, when there is any Disturbance
- Blink header when any new Disturbance comes

Notifications

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

Sounds

- Sound on new disturbance



Sounds

☑ Sound on new disturbance

Alarm*: None

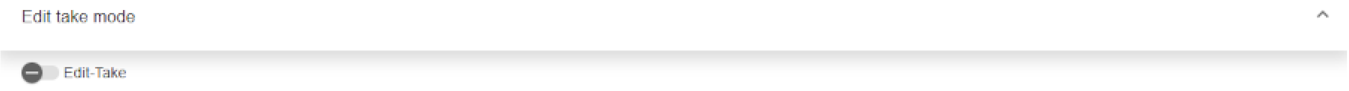
Fault*: None

Warning*: None

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

2.21.3 Edit take mode

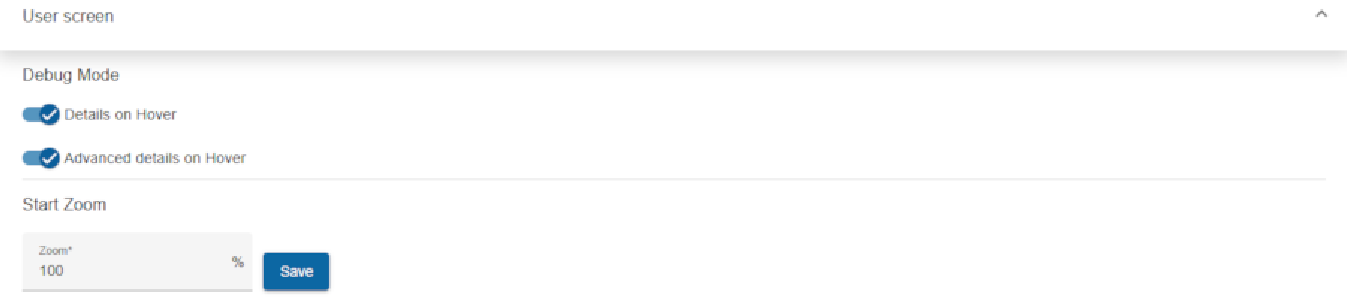
Configure the edit take mode for device windows. If turns on, device window parameters are disabled. You can click on edit button to start editing parameters on device window and clicking on take button will save the edited values.



Edit take mode

○ Edit-Take

2.21.4 User screen



User screen

Debug Mode

☑ Details on Hover

☑ Advanced details on Hover

Start Zoom

Zoom*: 100 %

Debug mode

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

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

Start zoom

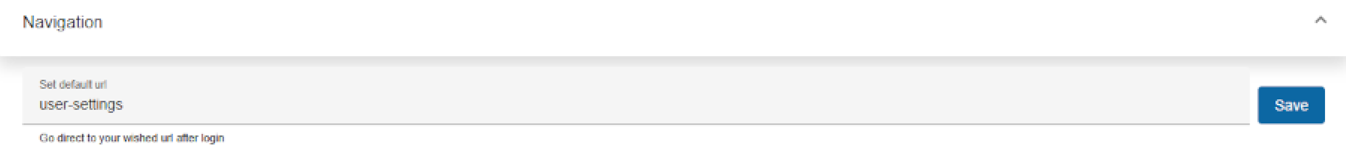
Users can define a custom zoom level (in %) for the user screens. Once set, the user screen will always display at the specified zoom level.

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

Save: Will save the zoom level.

2.21.5 Navigation

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



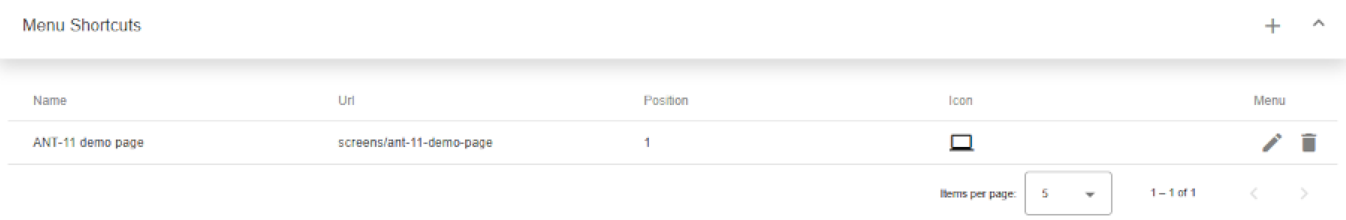
Navigation

Set default url
user-settings
Go direct to your wished url after login

Set default url: Just type your whished url.

Save: Will save the url

2.21.6 Menu shortcuts

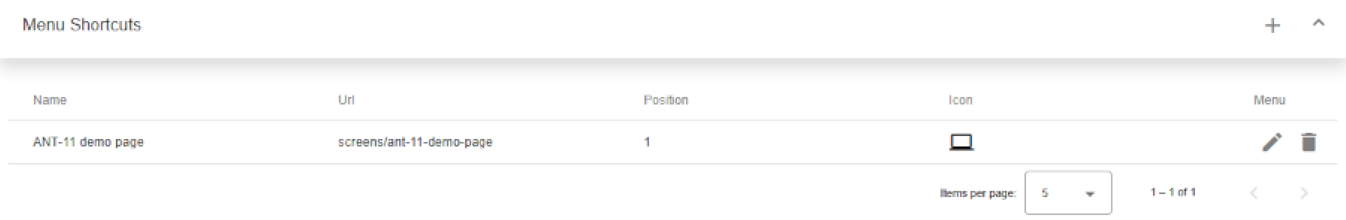


Menu Shortcuts

Name	Url	Position	Icon	Menu
ANT-11 demo page	screens/ant-11-demo-page	1	💻	✎ 🗑

Items per page: 5 1 – 1 of 1

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



Menu Shortcuts

Name	Url	Position	Icon	Menu
ANT-11 demo page	screens/ant-11-demo-page	1	💻	✎ 🗑

Items per page: 5 1 – 1 of 1

Top

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

Table columns

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

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

Bottom toolbar

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

2.21.6.1 Functionality

Create shortcut

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

Edit shortcut

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

Delete shortcut

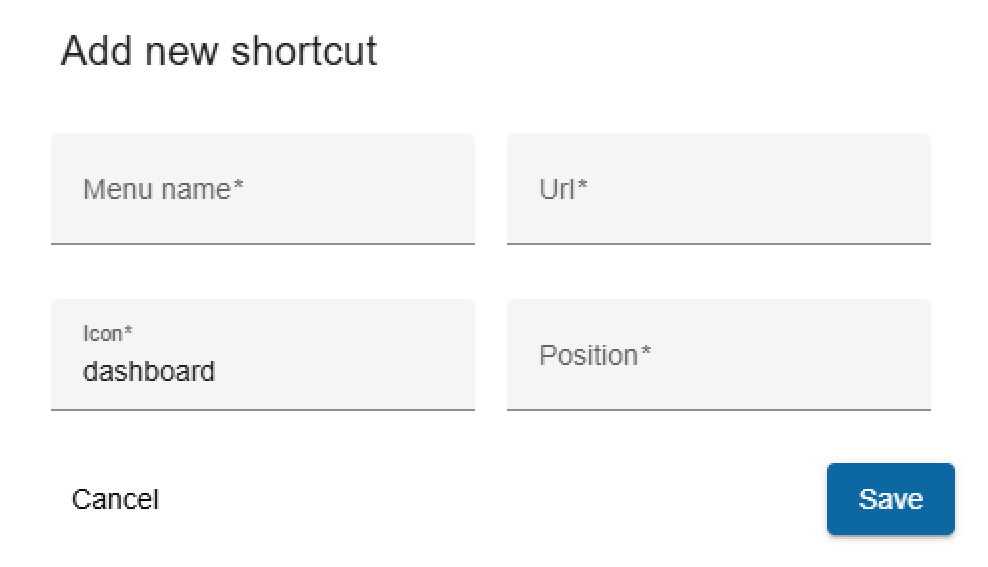
- Click the *trash icon* in the Functions column to delete an shortcut.
- Confirm the action in the pop-up dialog to delete the shortcut.

Pagination

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

2.21.6.2 Dialogs

Add/edit shortcut dialog



Add new shortcut

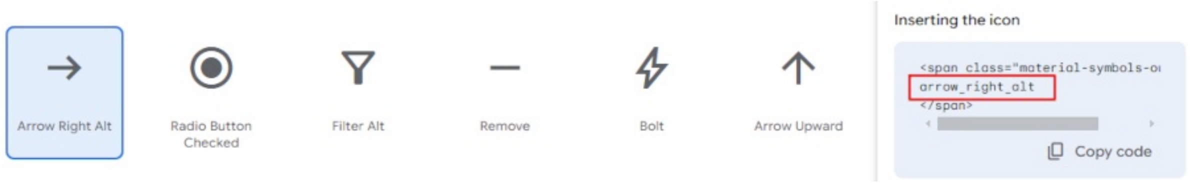
Menu name*

Url*

Icon*
dashboard

Position*

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



```
<span class="material-symbols-o
arrow_right_alt
</span>
```

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

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

On delete confirmation dialog

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

2.21.7 Table settings

Provides the option to set settings on tables of the app.

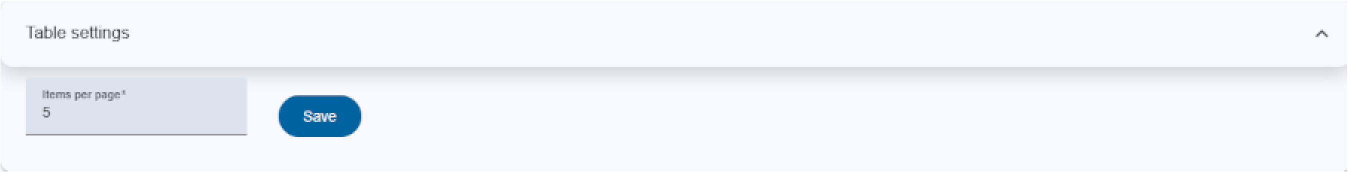


Table settings

Items per page*: 5

Items per page: Set number items that table should show.

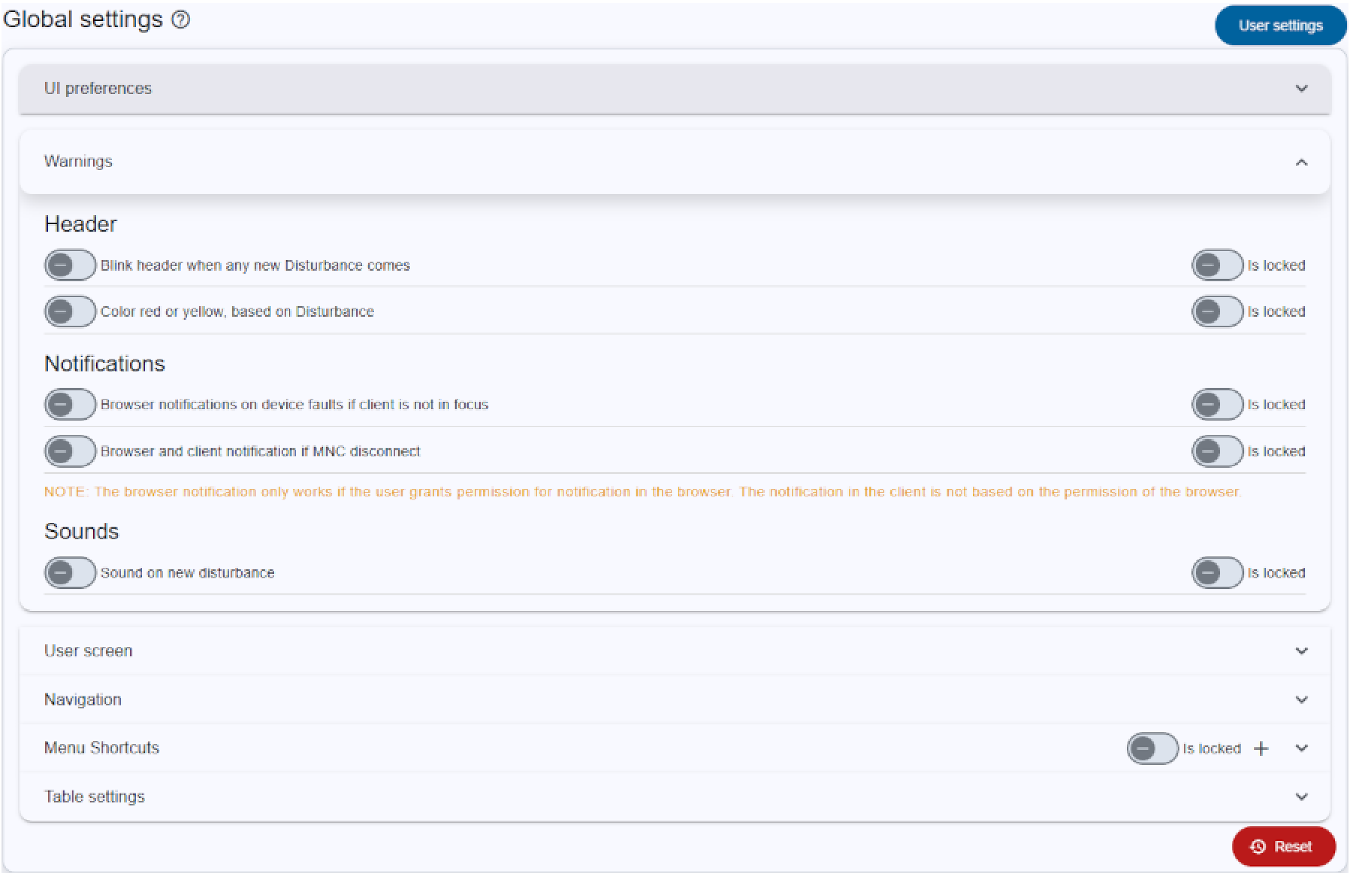
Save: Will save the setting.

2.21.8 Reset

2.22 Global settings

The Global Settings Page allows administrators to define and enforce default configurations across all users in the application. It shares the same structure and layout as the User settings, providing a familiar interface with the added ability to lock specific settings. Only users with administrative privileges can access this page.

Settings configured here act as defaults for users who haven’t explicitly saved their own preferences. Each setting includes a lock toggle. When a setting is locked, users cannot modify it in their personal settings, and the locked value is automatically applied by the application.



Global settings ⓘ

User settings

UI preferences

Warnings

Header

Blink header when any new Disturbance comes — Is locked

Color red or yellow, based on Disturbance — Is locked

Notifications

Browser notifications on device faults if client is not in focus — Is locked

Browser and client notification if MNC disconnect — Is locked

NOTE: The browser notification only works if the user grants permission for notification in the browser. The notification in the client is not based on the permission of the browser.

Sounds

Sound on new disturbance — Is locked

User screen

Navigation

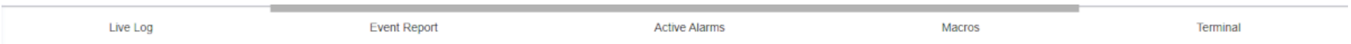
Menu Shortcuts — Is locked +

Table settings

Reset

2.23 Bottom panel

The bottom section of the webpage, referred to as the Bottom Toolbar, offers quick access to essential features and logs for system monitoring and interaction. The Bottom Toolbar is available throughout the application, ensuring users can quickly interact with features at all times. The bottom toolbar is expandable with selected tab.



It contains the following tabs:

- **Live log**: displays real-time logs and M&C activity.
- **Event report**: Shows a detailed report of recent events in the system.
- **Active Alarms**: Lists all currently active alarms or notifications.
- **Macros**: Provides access to automated scripts or macros.
- **Terminal**: Offers a direct terminal interface for advanced system commands.

2.23.1 Interaction features

Persistent accessibility

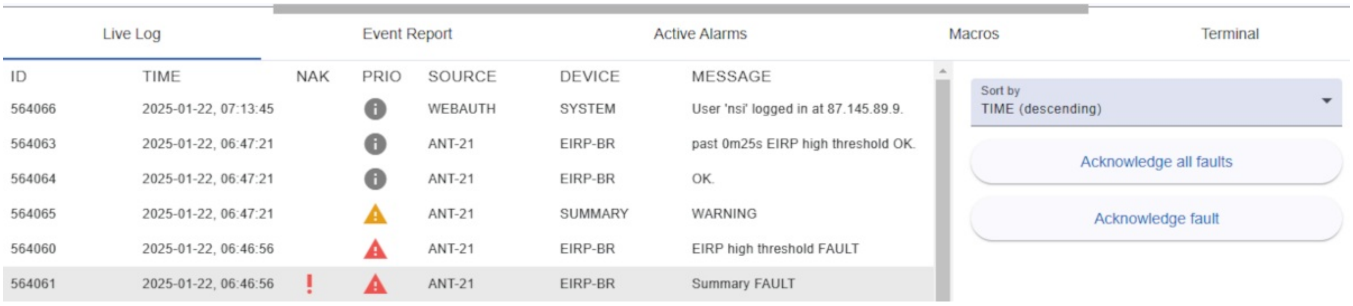
- The bottom toolbar is always present and can be accessed from any page in the app.
- Simply hover on the bottom toolbar to reveal the toolbar's resize handle.

Resize handle

- **Drag to resize**: Once the handle is visible, click and drag it upwards to expand the toolbar or downwards to minimize it.
- **Double-Click**: Double-clicking the resize handle will instantly expand the toolbar to full screen and by full screen double-clicking the resize handle minimize the toolbar instantly.

2.23.2 Live log

The Live Event Log window shows the recent event messages as they are received from one of the M&Cs. The Live Event Log window shows the messages in the order of their arrival at the M&C. The window can be resized, scrolled and always shows the last 25 messages.



ID	TIME	NAK	PRIO	SOURCE	DEVICE	MESSAGE
564066	2025-01-22, 07:13:45		ⓘ	WEBAUTH	SYSTEM	User 'nsi' logged in at 87.145.89.9.
564063	2025-01-22, 06:47:21		ⓘ	ANT-21	EIRP-BR	past 0m25s EIRP high threshold OK.
564064	2025-01-22, 06:47:21		ⓘ	ANT-21	EIRP-BR	OK.
564065	2025-01-22, 06:47:21		⚠	ANT-21	SUMMARY	WARNING
564060	2025-01-22, 06:46:56		⚠	ANT-21	EIRP-BR	EIRP high threshold FAULT
564061	2025-01-22, 06:46:56	!	⚠	ANT-21	EIRP-BR	Summary FAULT

Sort by
TIME (descending)

The sort order of the event list (newest first / oldest first) may be toggled using the*Sort by* dropdown on the right hand side. The default sort order is determined from a parameter set in the "client.properties" file located at the M&C server. Setting the key "gui.eventSortOrder" to "true" or "false" defines the default behavior of the live event log window.

Table columns

- **ID**: The event ID. This is a unique number the software assigns to each event which arrives at the M&C.
- **TIME**: The time stamp when the event was originated
- **NAK**: The 'need acknowledgment' mark. Messages showing a red exclamation mark in this column are fault messages which need to be acknowledged by the operator.
- **PRIO**: The message priority. The message priority decides on the urgency, the message gets delivered from the M&C to the client. It also is a search criteria in the Event Report window. The following priority symbols may appear:
    - ⓘ : This is an informational message (lowest priority)
    - ⚠ : This is a fault message (medium priority)
    - ⚠ : This is an alarm message (highest priority, the MNC tries to deliver this immediately, regardless of the dial timing settings)
- **SOURCE**: The primary message source. This usually is the ID of the MNC which issued the message. Messages marked with 'USER' or 'MNC' in this column.
- **DEVICE**: The secondary message source. This gives a more detailed information about the source of the message. It may be the name of the device in a MNC which caused the event.

- **MESSAGE**: The message text.

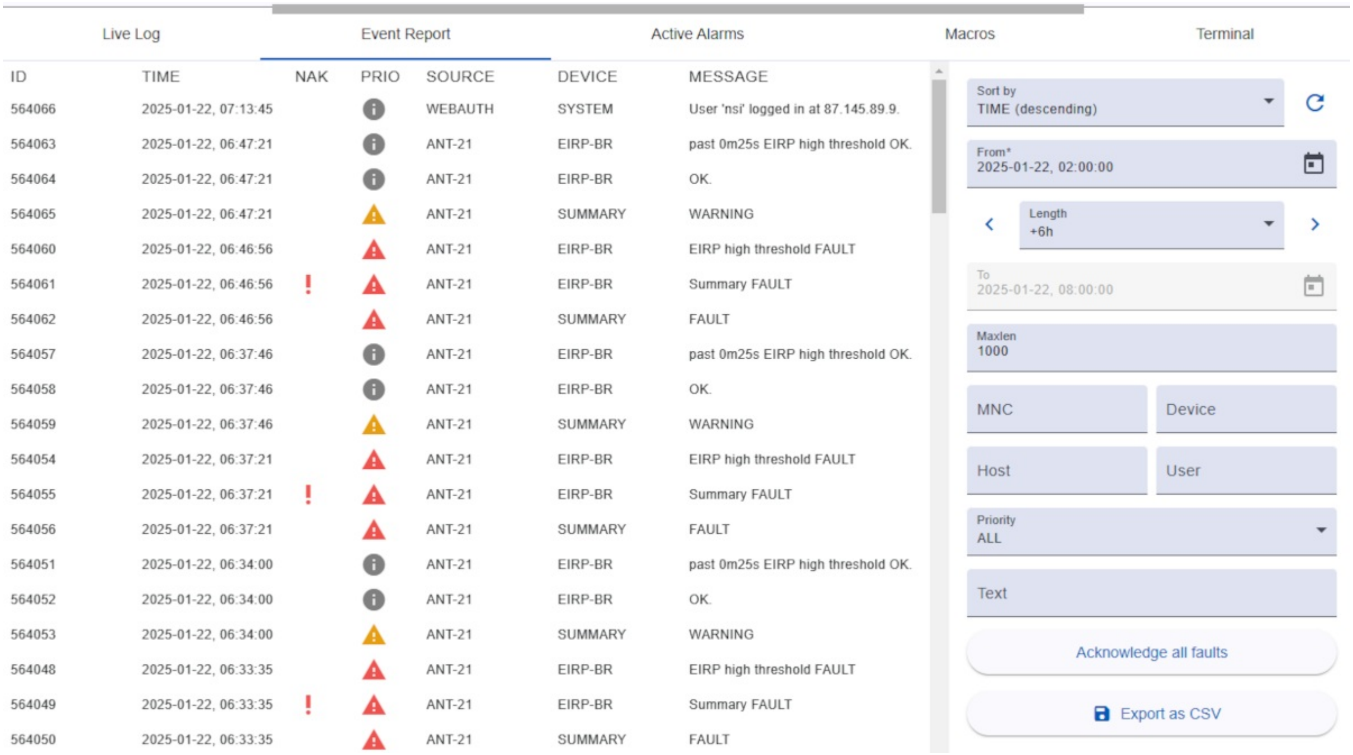
Right side toolbar

- **Sort table**: The table content can sort by descending and ascending.
- **Acknowledge all faults**: Acknowledge all faults marked with the red exclamation mark.
- **Acknowledge fault**: This button may appear if you want to acknowledge messages marked with the red exclamation mark. Simply select the row with red exclamation mark with mouse click. The exclamation mark will then disappear and a confirmation message will be added at the end of the log.

2.23.3 Event report

The Event Report window lets you generate filtered reports from the event database maintained by the NMS. This database contains all event (fault) messages issued by the MNCs during the last months (the exact depth of event storage is configurable during system installation).

The event report shown in the table gets automatically updated when you change the time range or apply other filter to display or the sort order. This lets you combine various filter specifications without being interrupted by the generation of intermediate reports.



ID	TIME	NAK	PRIO	SOURCE	DEVICE	MESSAGE
564066	2025-01-22, 07:13:45		ℹ	WEBAUTH	SYSTEM	User 'nsi' logged in at 87.145.89.9.
564063	2025-01-22, 06:47:21		ℹ	ANT-21	EIRP-BR	past 0m25s EIRP high threshold OK.
564064	2025-01-22, 06:47:21		ℹ	ANT-21	EIRP-BR	OK.
564065	2025-01-22, 06:47:21		⚠	ANT-21	SUMMARY	WARNING
564060	2025-01-22, 06:46:56		⚠	ANT-21	EIRP-BR	EIRP high threshold FAULT
564061	2025-01-22, 06:46:56	!	⚠	ANT-21	EIRP-BR	Summary FAULT
564062	2025-01-22, 06:46:56		⚠	ANT-21	SUMMARY	FAULT
564057	2025-01-22, 06:37:46		ℹ	ANT-21	EIRP-BR	past 0m25s EIRP high threshold OK.
564058	2025-01-22, 06:37:46		ℹ	ANT-21	EIRP-BR	OK.
564059	2025-01-22, 06:37:46		⚠	ANT-21	SUMMARY	WARNING
564054	2025-01-22, 06:37:21		⚠	ANT-21	EIRP-BR	EIRP high threshold FAULT
564055	2025-01-22, 06:37:21	!	⚠	ANT-21	EIRP-BR	Summary FAULT
564056	2025-01-22, 06:37:21		⚠	ANT-21	SUMMARY	FAULT
564051	2025-01-22, 06:34:00		ℹ	ANT-21	EIRP-BR	past 0m25s EIRP high threshold OK.
564052	2025-01-22, 06:34:00		ℹ	ANT-21	EIRP-BR	OK.
564053	2025-01-22, 06:34:00		⚠	ANT-21	SUMMARY	WARNING
564048	2025-01-22, 06:33:35		⚠	ANT-21	EIRP-BR	EIRP high threshold FAULT
564049	2025-01-22, 06:33:35	!	⚠	ANT-21	EIRP-BR	Summary FAULT
564050	2025-01-22, 06:33:35		⚠	ANT-21	SUMMARY	FAULT

Sort by: TIME (descending)
From*: 2025-01-22, 02:00:00
Length: +6h
To: 2025-01-22, 08:00:00
Maxlen: 1000
MNC
Device
Host
User
Priority: ALL
Text

Table columns

- **ID**: The event ID. This is a unique number the software assigns to each event which arrives at the M&C.
- **TIME**: The time stamp when the event was originated

- **NAK**: The 'need acknowledgment' mark. Messages showing a red exclamation mark in this column are fault messages which need to be acknowledged by the operator.
- **PRIO**: The message priority. The message priority decides on the urgency, the message gets delivered from the M&C to the client. It also is a search criteria in the Event Report window. The following priority symbols may appear:
  - ⓘ : This is an informational message (lowest priority)
  - ⚠ : This is a fault message (medium priority)
  - ⚠ : This is an alarm message (highest priority, the MNC tries to deliver this immediately, regardless of the dial timing settings)
- **SOURCE**: The primary message source. This usually is the ID of the MNC which issued the message. Messages marked with 'USER' or 'MNC' in this column.
- **DEVICE**: The secondary message source. This gives a more detailed information about the source of the message. It may be the name of the device in a MNC which caused the event.
- **MESSAGE**: The message text.

Filter functions

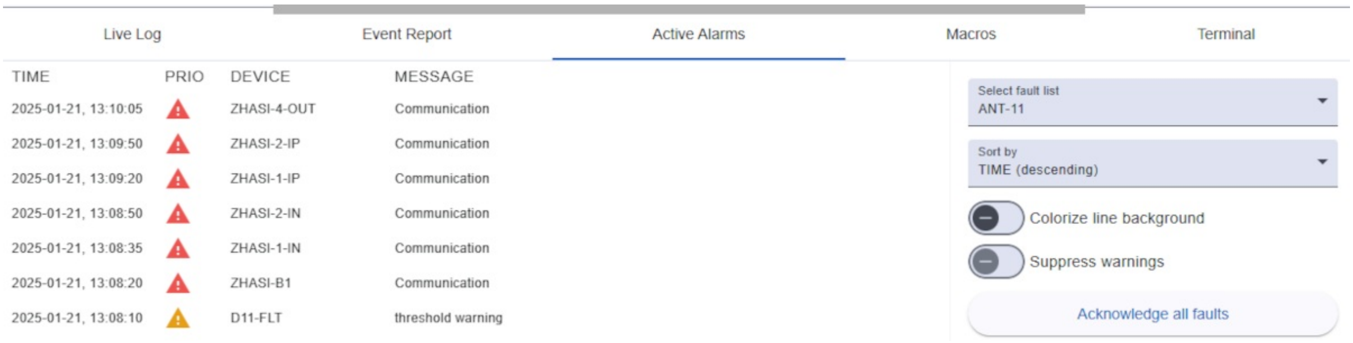
Located on the right side of event report tab. All filter can be combined with each other.

- **Sort by**: The table content can sort by descending and ascending of it's time.
- **Reload button**: Reload the table content to get latest data.
- **From**: Define the start time. If you click on the calendar icon on the right-hand side, a calendar view for selecting the date is displayed. The time must be entered manually.
- **Length**: The length is a drop-down menu that allows you to select duration. If you click on the arrow button on the right or left, the hours are added (on right arrow) to or subtracted (on left arrow) from the start time.
- **To**: Define the end time. Normally this input field is disabled to activate, please select 'select end time' from length dropdown.
- **Maxlen**: Define the maximum number of events that a report may contain. There is a default value for this (1000 in most cases), which represents a good compromise between performance and the number of events displayed simultaneously. You can change this limit according to your needs, but remember: reports with millions of events can take some time.
- **MNC**: This filter lets you filter with the M&C name.
- **Device**: This filter lets you filter messages along their device (secondary source) which caused the message.
- **Host**: The host name filter lets you restrict the report to messages originated from a certain client PC. Login messages and messages logging the active change of a parameter are stored with the origin (hostname or IP4 address) of the event.
- **User**: Lets you restrict the report to messages originated by a certain user. Login messages and messages logging the active change of a parameter are stored with the name of the user who is responsible for this event.
- **Priority**: The priority selector defines the priority (or type) that an event must have in order to be listed in the report. The selection *ALL* shows all events. *FAULT* contains events that have at least fault priority. *ALARM* only shows alarm messages (see the description

of alarm priorities above). The NOT ACK selection shows all messages that (still) need to be acknowledged.
- **Text**: Filter with the message text.
- **Acknowledge all faults**: Acknowledge all faults marked with the red exclamation mark.
- **Acknowledge fault**: This button may appear if you want to acknowledge messages marked with the red exclamation mark. Simply select the row with red exclamation mark with mouse click. The exclamation mark will then disappear and a confirmation message will be added at the end of the log.
- **Export as CSV**: This button downloads the current event report, as displayed in the table, as a CSV file.

2.23.4 Active alarms

The Active alarms tab lists all actually pending faults and warnings in the M&C. The Table automatically updates its content as the faults appear or disappear.



TIME	PRIO	DEVICE	MESSAGE
2025-01-21, 13:10:05	⚠	ZHASI-4-OUT	Communication
2025-01-21, 13:09:50	⚠	ZHASI-2-IP	Communication
2025-01-21, 13:09:20	⚠	ZHASI-1-IP	Communication
2025-01-21, 13:08:50	⚠	ZHASI-2-IN	Communication
2025-01-21, 13:08:35	⚠	ZHASI-1-IN	Communication
2025-01-21, 13:08:20	⚠	ZHASI-B1	Communication
2025-01-21, 13:08:10	⚠	D11-FLT	threshold warning

Select fault list: ANT-11

Sort by: TIME (descending)

Colorize line background

Suppress warnings

Acknowledge all faults

Table columns

- **TIME**: The time stamp when the event was originated
- **PRIO**: The message priority. The message priority decides on the urgency, the message gets delivered from the M&C to the client. It also is a search criteria in the Event Report window. The following priority symbols may appear:
  - ⓘ : This is an informational message (lowest priority)
  - ⚠ : This is a fault message (medium priority)
  - ⚠ : This is an alarm message (highest priority, the MNC tries to deliver this immediately, regardless of the dial timing settings)
- **DEVICE**: The secondary message source. This gives a more detailed information about the source of the message. It may be the name of the device in a MNC which caused the event.
- **MESSAGE**: The message text.

Right toolbar

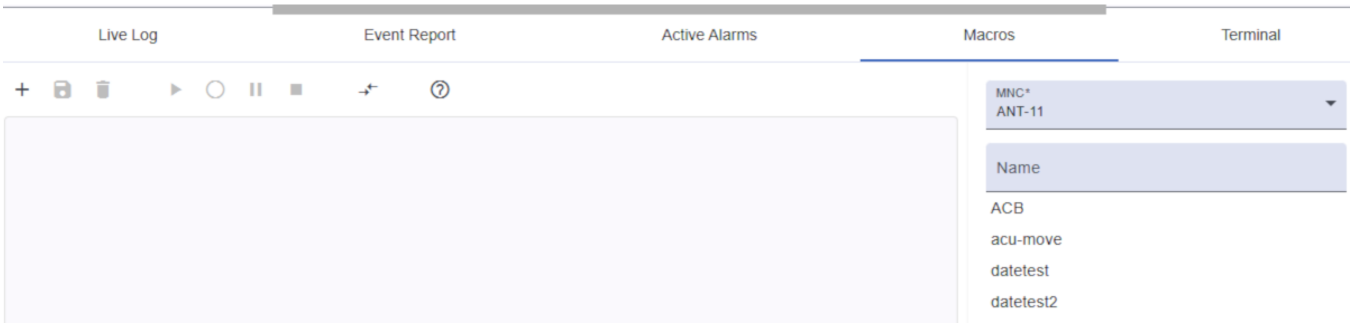
The toolbar buttons of the window let you select how the fault list displays the active faults:

- **Select fault list**: Select an MNC name from the drop-down list for which you want to check faults. Primary M&C is selected as default.

- **Sort by**: The table content can sort by descending and ascending of it's time.
- **Colorized the background**: If checked, the background of each line gets background color as the icon red for faults, yellow for warnings.
- **Suppress warnings**: If checked, the table content suppresses all warnings and show only faults only. Otherwise faults and warnings are displayed.
- **Acknowledge all faults**: Acknowledge all faults marked with the red exclamation mark.

2.23.5 Macros

The sat-nms software includes a powerful macro recording and playback feature which lets you automate complex equipment settings to one mouse click. The following chapters describe the user interface which give access to this feature.



Top toolbar

You must select an M&C from the right side toolbar, so that this toolbar can be displayed.

- **Add macro**: On click on + button you can create the new macro.
- **Save macro**: If you made some modifications you can save by clicking on this button.
- **Delete macro**: Selected macro can be deleted.
- **Play macro**: Play the selected macro.
- **Record macro**: You can start recording the macro. On toolbar of app the record icon will be displayed.
- **Pause/resume recording**: You can pause or resume recording.
- **Stop recording**: Stop the macro recording.
- **Compare macros**: To compare macros, please select macros from the macro list on the right-hand side. Both macros are displayed next to each other. The “Compare” button is displayed at the bottom of the window. It triggers a dialog that displays a better comparison. Clicking again on button will disable compare mode.

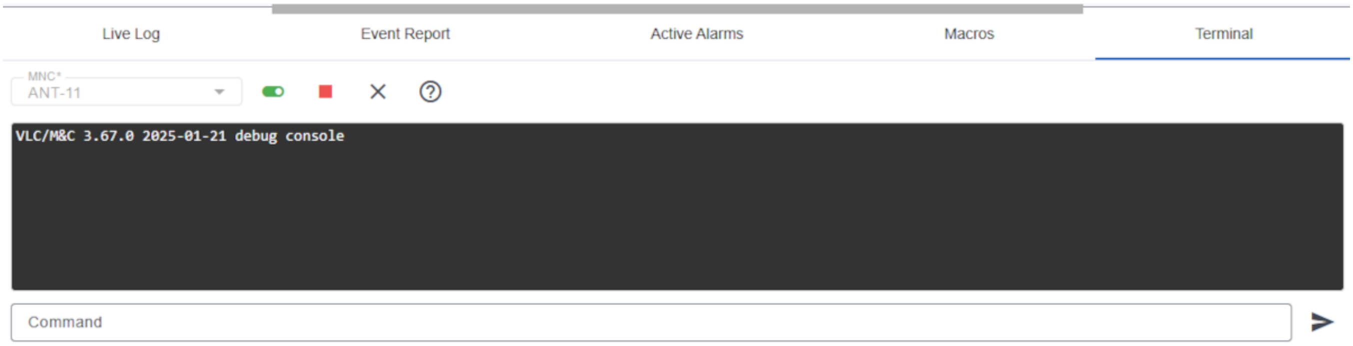
Right side toolbar

- **MNC**: Select an M&C to load all stored macros or perform other actions.
- **Name**: On creating new macro name should be entered, otherwise it works as a filter also. You can type the name of the macro to find the saved macro.
- **Macro list**: Once you selected an M&C, a macro list will be displayed under the name field. Simply select macro to perform actions described in above toolbar section.

Editing area: The macro editing area is located under the top toolbar. You can also edit the saved macro by simply typing as you would in a text editor. This area is also used to compare macros.

2.23.6 Terminal

The terminal tab gives you access to the debug console of a M&C system. This is a valuable tool to inspect the low level communication to devices which are controlled by the M&C.



```
VLC/M&C 3.67.0 2025-01-21 debug console
```

The terminal window consists of a display area which shows the debug output of the M&C and a command entry field at the bottom of the window. The tool bar at the top gives you access to the following functions:

1. **MNC**: Select an M&C to which you want to start the terminal connection.
2. **Switch**: After selecting an M&C you have to toggle this switch to start connect or disconnect the terminal connection. Green means terminal is connected.
3. **Stop/play icon**: It pauses or resumes the terminal session.
4. **Close icon**: It's clear the terminal display area.

After the terminal display area there are 'Command' input and 'send' icon. You can just type in command input and press 'enter' key to send command.

Debug commands

At the entry line at the bottom of the window you can enter a number of debugging commands which may help to test e.g. new device drivers. The following commands may be entered:

- get <msg-id> --- Get the current value of an parameter with the ID <msg-id>
- set <msg-id> <v> --- Set parameter with the ID <msg-id> to a value as it would entered by the user
- loop <msg-id> <v> --- Set parameter with the ID <msg-id> to a value as it would read from the device
- touch <msg-id> --- Force the parameter with <msg-id> to be read from the device with the next driver cycle
- threads --- Show a comprehensive list of device threads
- thread <n> --- Show details of thread number <n>
- peers --- Show peers / list of connected clients
- peer <n> (view|verbose|silent|kill) --- Details and functions of peer number <n>
- restart --- Restart MNC service (same as restart button in client)
- shutdown --- Stops MNC service. Attention: service has to be restarted on the command

line

- limit / nolimit --- make the 'debug.log' file unlimited or limited in size. The MNC always starts with the 'debug.log' file limited in size.
- csm-debug --- toggles the debug output of**all** CSM spectrum analyzers ON/OFF. The debug output appears in the `.panic.log` file. Remarks:
  - Be careful to switch csm-debug off again after using it. It remains on when the debug terminal window is closed unless it has been switched off before. This may cause a large amount data in the `.panic.log` file.
  - Using csm-debug only gives meaningful information if there is only one spectrum analyzer running in the system. If you have multiple spectrum analyzers configured, be sure to set all but the one you want to debug to OUT-OF-SERVICE.

3 Administration

This section of the manual gives some information how to administer and configure the various components of the satnms M&C software. It describes how to invoke the programs used by the software and which command line parameters these programs understand.

Chapter Configuration Files contains a reference to all configuration parameters which can be defined in varipus configuration files of the software.

Chapter Event Database Configuration and Event Notification Add-Onsdescribes how to configure the event database and opetional add-ons which may be configured to trigger some actions on certain events.

Finally, the chapter Remote Control describes the features built into the satnms M&C software for being remotely monitored and controlled by an external software.

3.1 Program Invocation

The satnms M&C software consists of a number of programs running independently on the same or on separate machines. The following chapters describes how to invoke these programs and which command line parameters they accept.

3.1.1 M&C Server

The M&C server is run as a system service and is automatically started when the operating system comes up. Normally there is no need to change the command line parameters which are passed to the program by the the startup scripts. Most parameters of the M&C server may be configured by editing the vlc.propertis configuration file.

If the M&C server must be stopped or started explicitly, the follwing commands sould be used:

```
$ systemctl stop satnms
```

and

```
$ systemctl start satnms
```

respectively. To start or stop the M&C one must be logged in as 'root' on the machine hosting the M&C server.

3.1.2 Backend Server

The backend server is run as a system service and is automatically started when the operating system comes up. There are no command line parameters for the server program itself, the behavior of the backend is entirely controlled by the backend.properties file.

If the backend server must be stopped or started explicitly, the follwing commands sould be used:

```
$ systemctl stop backend
```

and

```
$ systemctl start backend
```

respectively. To start or stop the backend one must be logged in as 'root' on the machine hosting the backend server.

3.1.3 Client Programs

The following paragraphs list the command line invocation syntax and options for the sat-nms client software. Being pure Java applications, the sat-nms clients may be run on severals operating systems. The explanations given in the following are mostly independent from the operating system you are using. For all examples given below, we assume that the file "client.jar" is located in your current working directory.

M&C Client

The M&C client program is invoked by a command line looking like this:

java -cp client.jar satnms3.gui.MCFrame [options ...] <IP address of the server>

The IP address of the M&C server may be replaced by a host name if your network installation provides name resolution for this address.

The following command line options may be used with the program:

- -u <user-name> --- Specifies the user name for autologin. Must be used together with -p
- -p <password> --- Specifies the password for autologin. Must be used together with -u
- -l <look-and-feel> --- Lets you set the "look and feel" of the user interface to one of several definitions. Accepted values are: Aero, Fast, Aluminium, McWin, Luna, Pulsar, Noire, Mint, Graphite, Smart, Bernstein, Acryl and HiFi.
- -m x0,y0,x1,x2,xc,yc --- Specifies the (overall) bounds of a multi monitor setup. When operated on a single monitor PC, the client automatically determines the screen size and ensures that automatically placed windows remain within the bounds of the screen. The -m option is not necessary in this case. With multiple monitors, the abolute bounds of the rectangle enclosing all monitors must be specified with the -m option. -m must be followed by a space character and 6 numbers separated by comma characters: x0 = x coordinate of the top left corner y0 = y coordinate of the top left corner x1 = x coordinate of the bottom right corner y1 = y coordinate of the bottom right corner xc = x coordinate of center position yc = y coordinate of center position Example: 0,0,3840,1080,2880,540 describes a layout with two full HD monitors located side by side. The center position is defined as the center of the right monitor (the splash screen and some other windows are

shown centered at this point.

- -f 0 -f 1 --- Starts the client in full screen mode. The main window covers the complete screen, no window decorations are shown. *-f 1* shows the satnms toolbar (to launch the event log for example) *-f 0* suppresses the tool bar.The full screen mode is intended to be used with touch screen panels. There are limited capabilities to handle child windows in *-f 0* mode, depending on the operating system of the client pc.

- -t --- Starts the client without the toolbar.

- -n 1 -n 2 -n 3 --- Tells the client how to treat "new faults". Without the -n option, new faults are not reported separately. The fault indicator in the main window's toolbar ia active while the server's summary fault is active.

    - **n1** --- The program treats new faults separately. A fault is new until an operator acknowledges it by clicking into the fault indicator or by acknowledging the fault in the event log or event report window. The new fault is displayed as a red mark labeled "NEW FAULT". If the fault disappears without being acknowledged, a gray "NEW FAULT" mark is shown.
    - **n2** --- Behaves like **-n 1** , but makes the program ring the keyboard bell while a new fault is pending.
    - **n3** --- Works like the previous, but suppresses the very first "NEW FAULT" after the start of the client. New faults which occurred while the client program did not run are not signalled, nor are gray "NEW FAULT" conditions. Only new faults recognized while the client program runs are shown.

- -s <screen-name> --- Tells the program to load the given screen definition file instead of the "default" screen on startup. Screen definition files are expected in the "screens" subdirectory of the server. The -s also suppresses the display of the applications's splash screen on startup.

- -v --- Suppresses the program version match check on startup

- -d <r,g,b> --- Sets the foreground color of disabled texts / elements in the GUI. r,g,b are numbers in the range 0..255, comma separated, specifying the RGB components of the color.

- -V --- Tells the program to be verbose on the command line. This option is for debugging only, **never** set it for operational use.

- -y <layout-name> --- Tells the program to load the given window layout instead of the layour "default".

TreeView Client

The TreeView version of the M&C client program is invoked by a command line looking like this:

java -cp client.jar satnms3.gui.TreeView [options ...] <IP address of the server>

The IP address of the M&C server may be replaced by a host name if your network installation provides name resolution for this address.

The following command line options may be used with the program:

- -u <user-name> --- Specifies the user name for autologin. Must be used together with -p
- -p <password> --- Specifies the password for autologin. Must be used together with -u
- -l <look-and-feel> --- Lets you set the "look and feel" of the user interface to one of several definitions. Accepted values are: Aero, Fast, Aluminium, McWin, Luna, Pulsar, Noire, Mint, Graphite, Smart, Bernstein, Acryl and HiFi.
- -m x0,y0,x1,x2,xc,yc --- Specifies the (overall) bounds of a multi monitor setup. When operated on a single monitor PC, the client automatically determines the screen size and ensures that automatically placed windows remain within the bounds of the screen. The -m option is not necessary in this case. With multiple monitors, the abolute bounds of the rectangle enclosing all monitors must be specified with the -m option. -m must be followed by a space character and 6 numbers separated by comma characters: x0 = x coordinate of the top left corner y0 = y coordinate of the top left corner x1 = x coordinate of the bottom right corner y1 = y coordinate of the bottom right corner xc = x coordinate of center position yc = y coordinate of center position Example: 0,0,3840,1080,2880,540 describes a layout with two full HD monitors located side by side. The center position is defined as the center of the right monitor (the splash screen and some other windows are shown centered at this point.
- -f 0 -f 1 --- Starts the client in full screen mode. The main window covers the complete screen, no window decorations are shown. *-f 1* shows the sat-nms toolbar (to launch the event log for example) *-f 0* suppresses the tool bar. The full screen mode is intended to be used with touch screen panels. There are limited capabilities to handle child windows in *-f 0* mode, depending on the operating system of the client pc.
- -t --- Starts the client without the toolbar.
- -o --- Enables automatic pop ups of the fault list window with every new fault.
- -n 1 -n 2 -n 3 --- Tells the client how to treat "new faults". Without the -n option, new faults are not reported separately. The fault indicator in the main window's toolbar ia active while the server's summary fault is active.
    - **n1** --- The program treats new faults separately. A fault is new until an operator acknowledges it by clicking into the fault indicator or by acknowledging the fault in the event log or event report window. The new fault is displayed as a red mark labeled "NEW FAULT". If the fault disappears without being acknowledged, a gray "NEW FAULT" mark is shown.
    - **n2** --- Behaves like **-n 1** , but makes the program ring the keyboard bell while a new fault is pending.
    - **n3** --- Works like the previous, but suppresses the very first "NEW FAULT" after the start of the client. New faults which occurred while the client program did not run

are not signalled, nor are gray "NEW FAULT" conditions. Only new faults recognized while the client program runs are shown.

- -s <screen-name> --- Tells the program to load the given screen definition file instead of the "default" screen on startup. Screen definition files are expected in the "screens" subdirectory of the server. The -s also suppresses the display of the applications's splash screen on startup.

- -v --- Suppresses the program version match check on startup

- -d <r,g,b> --- Sets the foreground color of disabled texts / elements in the GUI. r,g,b are numbers in the range 0..255, comma separated, specifying the RGB components of the color.

- -V --- Tells the program to be verbose on the command line. This option is for debugging only, **never** set it for operational use.

Tabbed View Client

The Tabbed View NMS client program is invoked by a command line looking like this:

java -cp client.jar satnms3.gui.TabbedView [options ...] <IP address of the server>

The IP address of the NMS server may be replaced by a host name if your network installation provides name resolution for this address.

The following command line options may be used with the program:

- -u <user-name> --- Specifies the user name for autologin. Must be used together with -p
- -p <password> --- Specifies the password for autologin. Must be used together with -u
- -l <look-and-feel> --- Lets you set the "look and feel" of the user interface to one of several definitions. Accepted values are: Aero, Fast, Aluminium, McWin, Luna, Pulsar, Noire, Mint, Graphite, Smart, Bernstein, Acryl and HiFi.
- -m x0,y0,x1,x2,xc,yc --- Specifies the (overall) bounds of a multi monitor setup. When operated on a single monitor PC, the client automatically determines the screen size and ensures that automatically placed windows remain within the bounds of the screen. The -m option is not necessary in this case. With multiple monitors, the abolute bounds of the rectangle enclosing all monitors must be specified with the -m option. -m must be followed by a space character and 6 numbers separated by comma characters: x0 = x coordinate of the top left corner y0 = y coordinate of the top left corner x1 = x coordinate of the bottom right corner y1 = y coordinate of the bottom right corner xc = x coordinate of center position yc = y coordinate of center position Example: 0,0,3840,1080,2880,540 describes a layout with two full HD monitors located side by side. The center position is defined as the center of the right monitor (the splash screen and some other windows are shown centered at this point.
- -s --- Suppresses the beeper ("silent").
- -t <tab-definition> --- Tells the program to load the given tab definition file instead of the "default" definition on startup. Tab definition files are expected in the "tviews" subdirectory of the NMS server.

- -d <r,g,b> --- Sets the foreground color of disabled texts / elements in the GUI. r,g,b are numbers in the range 0..255, comma separated, specifying the RGB components of the color.
- -v --- Suppresses the program version match check on startup
- -V --- Tells the program to be verbose on the command line. This option is for debugging only, **never** set it for operational use.

Service Client

The ServiceClient client program is invoked by a command line looking like this:

java -cp client.jar satnms3.sclient.ServiceClient [options ...] <IP address of the server>

The IP address of the M&C server may be replaced by a host name if your network installation provides name resolution for this address.

The following command line options may be used with the program:

- -u <user-name> --- Specifies the user name for autologin. Must be used together with -p

- -p <password> --- Specifies the password for autologin. Must be used together with -u

- -l <look-and-feel> --- Lets you set the "look and feel" of the user interface to one of several definitions. Accepted values are: Aero, Fast, Aluminium, McWin, Luna, Pulsar, Noire, Mint, Graphite, Smart, Bernstein, Acryl and HiFi.

- -n 1 -n 2 -n 3 --- Tells the client how to treat "new faults". Without the -n option, new faults are not reported separately. The fault indicator in the main window's toolbar ia active while the server's summary fault is active.

  - **n1** --- The program treats new faults separately. A fault is new until an operator acknowledges it by clicking into the fault indicator or by acknowledging the fault in the event log or event report window. The new fault is displayed as a red mark labeled "NEW FAULT". If the fault disappears without being acknowledged, a gray "NEW FAULT" mark is shown.
  - **n2** --- Behaves like **-n 1** , but makes the program ring the keyboard bell while a new fault is pending.
  - **n3** --- Works like the previous, but suppresses the very first "NEW FAULT" after the start of the client. New faults which occurred while the client program did not run are not signalled, nor are gray "NEW FAULT" conditions. Only new faults recognized while the client program runs are shown.

- -s <service-definition> --- Tells the client to use another service definition file than 'default.xml'. The service definition file must be placed in the ~/sclient directory at the M&C server

- -o --- Enables automatic pop ups of the fault list window with every new fault.

- -r <seconds> --- Enables the automatic reconnect feature. Should not be used directly, the MasterClient application makes use of this switch.

- -d <r,g,b> --- Sets the foreground color of disabled texts / elements in the GUI. r,g,b are numbers in the range 0..255, comma separated, specifying the RGB components of the color.
- -v --- Suppresses the program version match check on startup
- -V --- Tells the program to be verbose on the command line. This option is for debugging only, **never** set it for operational use.

VioIP Client

The VioIPClient client program is invoked by a command line looking like this:

java -cp client.jar satnms3.vioip.VioIPClient [options ...] <IP address of the server>

The IP address of the M&C server may be replaced by a host name if your network installation provides name resolution for this address.

The following command line options may be used with the program:

- -u <user-name> --- Specifies the user name for autologin. Must be used together with -p
- -p <password> --- Specifies the password for autologin. Must be used together with -u
- -g <group-ip> --- Defines the group to be displayed by this client instance. Expects the group ID (not the name) as a parameter. This parameter is mandatory to make the program work, the group ID cannot be set or changed at the GUI.
- -l <look-and-feel> --- Lets you set the "look and feel" of the user interface to one of several definitions. Accepted values are: Aero, Fast, Aluminium, McWin, Luna, Pulsar, Noire, Mint, Graphite, Smart, Bernstein, Acryl and HiFi.
- -n 1 -n 2 -n 3 --- Tells the client how to treat "new faults". Without the -n option, new faults are not reported separately. The fault indicator in the main window's toolbar ia active while the server's summary fault is active.
    - **n1** --- The program treats new faults separately. A fault is new until an operator acknowledges it by clicking into the fault indicator or by acknowledging the fault in the event log or event report window. The new fault is displayed as a red mark labeled "NEW FAULT". If the fault disappears without being acknowledged, a gray "NEW FAULT" mark is shown.
    - **n2** --- Behaves like **-n 1** , but makes the program ring the keyboard bell while a new fault is pending.
    - **n3** --- Works like the previous, but suppresses the very first "NEW FAULT" after the start of the client. New faults which occurred while the client program did not run are not signalled, nor are gray "NEW FAULT" conditions. Only new faults recognized while the client program runs are shown.
- -q --- Suppresses some alerts like 'connection broken' and others.

- -r <seconds> --- Enables the automatic reconnect feature. Should not be used directly, the MasterClient application makes use of this switch.
- -d <r,g,b> --- Sets the foreground color of disabled texts / elements in the GUI. r,g,b are numbers in the range 0..255, comma separated, specifying the RGB components of the color.
- -v --- Suppresses the program version match check on startup
- -V --- Tells the program to be verbose on the command line. This option is for debugging only, **never** set it for operational use.

VioIP Configurator

The VioIPConfigurator program usually is invoked with a button click from the VioIPClient program. Beside this, it also may be invoked as a stand alone program by a command line looking like this:

java -cp client.jar satnms3.vioip.VioIPConfigurator [options ...] <IP address of the server>

The IP address of the M&C server may be replaced by a host name if your network installation provides name resolution for this address.

The following command line options may be used with the program:

- -u <user-name> --- Specifies the user name for autologin. Must be used together with -p
- -p <password> --- Specifies the password for autologin. Must be used together with -u
- -s --- Suppresses the splash screen.
- -l <look-and-feel> --- Lets you set the "look and feel" of the user interface to one of several definitions. Accepted values are: Aero, Fast, Aluminium, McWin, Luna, Pulsar, Noire, Mint, Graphite, Smart, Bernstein, Acryl and HiFi.
- -q --- Suppresses some alerts like 'connection broken' and others.
- -d <r,g,b> --- Sets the foreground color of disabled texts / elements in the GUI. r,g,b are numbers in the range 0..255, comma separated, specifying the RGB components of the color.
- -v --- Suppresses the program version match check on startup
- -V --- Tells the program to be verbose on the command line. This option is for debugging only, **never** set it for operational use.

CDB Table Editor

The CDB Table Editor program usually is invoked with a button click from the M&C Client program. Beside this, it also may be invoked as a stand alone program by a command line looking like this:

java -cp client.jar satnms3.gui.CDBTableEditor [options ...] <IP address of the server>

The IP address of the M&C server may be replaced by a host name if your network installation provides name resolution for this address.

The following command line options may be used with the program:

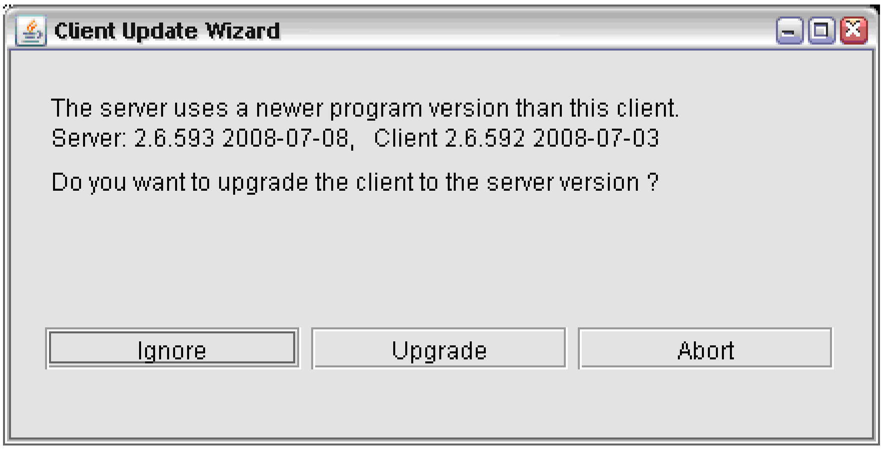
- -u <user-name> --- Specifies the user name for autologin. Must be used together with -p
- -p <password> --- Specifies the password for autologin. Must be used together with -u
- -s --- Suppresses the database selector widget in the toolbar.
- -l <look-and-feel> --- Lets you set the "look and feel" of the user interface to one of several definitions. Accepted values are: Aero, Fast, Aluminium, McWin, Luna, Pulsar, Noire, Mint, Graphite, Smart, Bernstein, Acryl and HiFi.
- -db <db-name> --- Selects the database to be edited. Default is 'channels'.

3.1.4 Automatic Software Update Of The Client

The sat-nms Client Software (version 2.6.591 and above) includes an automatic check for updates. After startup the client connects to the NMS server and compares the software versions. Client and server should always run the same version of the sat-nms software. The software update feature also supports a downgrade of the client software, e.g. if the server has been restored from a backup with an older software version.

Upgrade

If the NMS server is running a newer version of sat-nms you can upgrade the local client to this version.



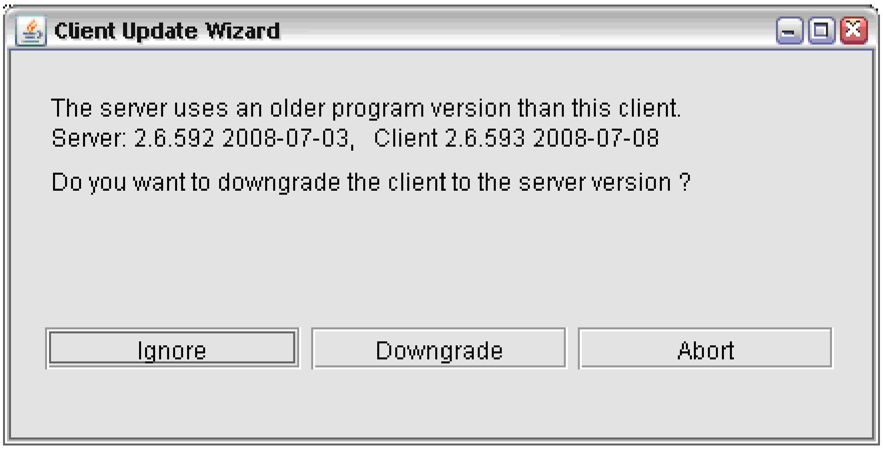
The server uses a newer program version than this client.
Server: 2.6.593 2008-07-08,   Client 2.6.592 2008-07-03

Do you want to upgrade the client to the server version ?

- Ignore: --- This ignores the version discrepancy and you will working with the already installed client program.
- Upgrade: --- The newer client program will be installed on the local machine
- Abort: --- Aborts the client program, cancels the program download

Downgrade

If the NMS server is running an older version of sat-nms you can downgrade the local client to this version.



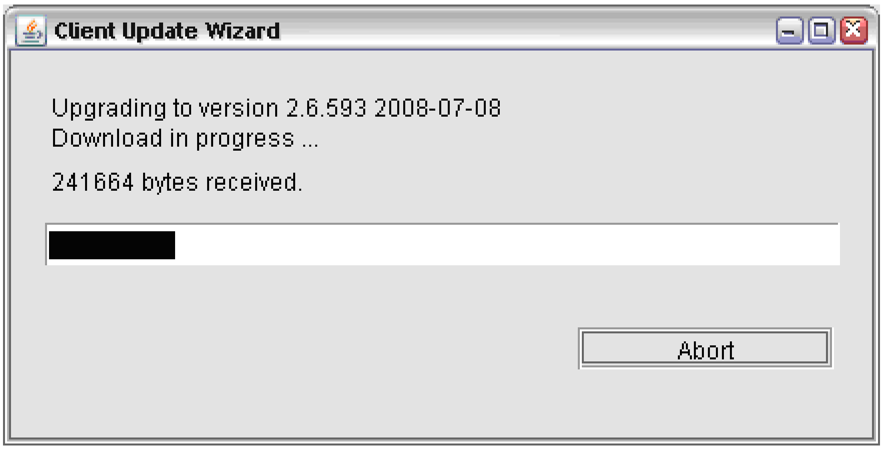
The server uses an older program version than this client.
Server: 2.6.592 2008-07-03, Client 2.6.593 2008-07-08

Do you want to downgrade the client to the server version ?

- Ignore: --- This ignores the version discrepancy and you will working with the already installed client program.
- Downgrade: --- The older client program will be installed on the local machine
- Abort: --- Aborts the client program, cancels the program download

Download process

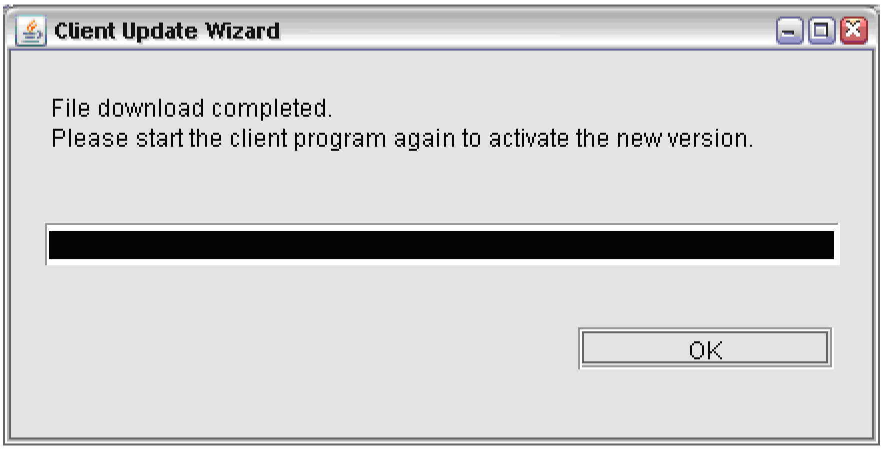
After pressing the upgrade or downgrade button, the software starts the download of the client program from the server to the local machine. With a LAN connection this takes only a few seconds, if you are connected through a dial up connection may take somewhat longer.



Upgrading to version 2.6.593 2008-07-08
Download in progress ...

241664 bytes received.

You may cancel the download at any time by clicking the 'Abort' button. After successfully downloading the software displays a message and after selecting OK the client program will be closed.



File download completed.
Please start the client program again to activate the new version.

Now simply restart the client again as described in chapter 'start client' and you will work with the upgraded or downgraded version.

3.2 Configuration Files

The satnms M&C software provides a number of configuration files which control wide aspects of the functionality of the software files all use an easy to read and mostly self explaining format. The files explained in the following chapters are:

- backend.properties: This file is located in the base directory of the satnms backend installation. It controls the behavior of the backend server.
- user-ip-restrictions.json: This file is also located in the base directory of the satnms backend installation. It controls which user may login to the backend from which machine.
- vlc.properties: This file is located in the base directory of the satnms M&C installation. It controls all important parameters of the M&C server.
- client.properties: This file also is located in the base directory of the software. It is read by every legacy Java client connecting to the server. It controls important aspects of the client software's behavior.
- satnms.rc: Controls the startup of the satnms software and programs which have to be initialized before the M&C server itself can be started.
- system.name: Defines the name of the M&C system
- traprec.json: Controls the behavior of the SNMP trap receiver built into the software.

3.2.1 backend.properties

The backend.properties file controls the configuration of the backend server. How many and which M&C systems are controlled by the backend, how to authenticate at the backend and some debugging options are defined in this file. The backend.properties file is read once by the software on startup, so any changes to this file become effective with the next start of the backend. The backend installation package contains a backend.properties.default file which can be used as a template to setup a new properties file from scatch

Controlled M&C systems

For every M&C system controlled by the backend, at least two parameters must be set: The M&C system's IP address (`backend.mnc.*.ip`) and it's name (`backend.mnc.*.name`). Example for the first M&C:

```
backend.mnc.1.ip=127.0.0.1
backend.mnc.1.name=MNC
```

M&Cs are numbered 1,2 ... n, number 1 is called the primary M&C. It must always exist, the backend uses it to get screens, driver definitions and to authenticate users during the login. The name of each M&C should match the name which is set at the M&C server itself. This ensures that the M&C name presented at the UI frontend matches the M&C name appearing in event log messages.

The backend supports redundant M&C pairs, monitors and control the switchover. To define a redundant M&C pair, some more parameters are required. They are the IP address of the backup M&C and some timing parameter controlling the switchover process.

```
backend.mnc.2.ip=192.168.2.229
backend.mnc.2.name=MNC-2
backend.mnc.2.backup.ip=192.168.2.227
backend.mnc.2.backup.retries=3
backend.mnc.2.backup.delay=15
```

As soon as `backend.mnc.*.backup.ip` is defined for a M&C, this is considered to be a redundant pair. The `backend.mnc.*.backup.retries` parameter defined how many reconnect attempts to the primary M&C shall be made until a switch over is done. `backend.mnc.*.backup.delay` specifies the wait time between connection attempts.

EventDB server

The backend needs to know where the event database is hosted. The IP address is specified with:

```
eventdb.server=127.0.0.1
```

For small installations this is typically the same machine as the backend

SQL database access

The backend stores information about satellites, satellite channels and antenna pointings in SQL a database. The setings below specify how to access this database. Again in a small installation the database will run on the same host as the backend itself.

```
db.satlist.host=127.0.0.1
db.satlist.user=satnms
db.satlist.password=satnms
db.satlist.usessl=false
```

The backend may use a redundant database configuration where the primary database server is configured r/w and the backup server is r/o. For such a configuration the `db.satlist.host` parameter defines the primary and backup server addresses separated by a comma. The first address denotes the primary database server.

```
db.satlist.host=192.168.0.100,192.168.0.101
db.satlist.keep=300
```

The `db.satlist.keep` parameter defines the time (secs) after which the backend cuts a connetion to the backup server in order to try if the primary one is available again.

Inventory database

The inventory database is part of the SQL database which also stores satellite channels etc. The inventory feature must be enabled explicitly by setting the parameter below to true:

```
db.inventory.enabled=false
db.inventory.debug=false
```

The `db.satlist.*` parameters shown in the section above this must also be setup properly to enable the inventory database. Setting the debug flag *true* lets the software print out all SQL traffic regarding the inventory into the .panic.log file. This causes huge log files, therefore shouls be used only investigating problems, not during normal operation.

OAuth2 authentication

The backend uses OAuth2 to authenticate API calls. it provides a local authority which issues tokens for local logins. The parameters below configure the behavior of this feature

```
oauth2.access.token.valid.time=3600
```

Specifies the time a token is valid (seconds). default is 3600

```
oauth2.refresh.token.valid.time=7200
```

Specifies the time the refresh token is valid (seconds). default is 7200

```
max.sessions.total=0
```

Specifieds the max. number of session which totally may be open at the same time (default 0, means don't care)

```
max.sessions.user=0
```

Specified the max. number of session which may be open at the same time for the same user (default 0, means don't care)

The backend accepts JWT tokens from an external authentication authority if the following parameters are set URL to access the JWKS from this authority. default is empty.

```
oauth2.jwk.provider=
```

The parameter below secifies IP address of an HTTP(s) proxy if the JWKS provider shall be accessed thru this proxy. default is empty (no proxy)

```
oauth2.jwk.proxy=
```

The parameter below specifies the AD group name to privilege level look up list. It consists of group-name:privilege-level pairs, separated by comma characters. default is an empty list.

```
oauth2.groups=SATNMS-ADM:150,SATNMS-OP:110
```

Beside from the "Authentication" header the backend may read the JWT token from a cookie coming with the API call. default is mpty (don't accept JWT from a cookie)

```
oauth2.cookie.name=
```

LDAP local login

For local logins the backend may look up the credentials of a user at an LDAP directory server instead of parsing the `.users` file from the primary M&C. The settings below are used to specify how to access the LDAP server and how to interprets the data read from there:

```
ldap.enable=yes
ldap.url=ldaps://ad.mydomain:636
ldap.default.domain=ad.mydomain
ldap.search.user=searchsatnms
ldap.search.password=searchsatnms
ldap.groups=SATNMS-ADM:150,SATNMS-OP:110
ldap.keystore.name=ldap.p12
ldap.keystore.password=satnms
```

Debug settings

The backend offers a number of parameters which enable extensive logging of certain events, internal messages of parameter updates for debugging purposes. For normal operation you should keep the disabled all because log files can become very large with debuggin enabled. Debugging also may reduce the performance of the backend.

The parameter/status communication to a specific M&C may be logged by setting this parameter true (replace '*' with the index of the M&C to switch to verbose.

```
backend.mnc.*.verbose=true
```

Setting this true logs every message sent over a websocket. Creates a huge amount of log data, logs are better to interpret if only one single frontend with one window is connected to the backend

```
websocket.verbose=false
```

Setting the parameter blow to true logs m&c connect/disconnect events.

```
websocket.logConnected=false
```

Setting the parameter below to true logs the oauth2 authentication process for every API call in detail.

```
oauth2.verbose=false
```

The parameter below controls logging for LDAP logins. Please note that this setting expects 'yes' to be activated.

```
ldap.verbose=no
```

3.2.1.1 Example configurations

MNC and Backend services on same host (all-in-one installation)

This is a very simple configuration running backend and MNC service on the same host without any additional features:

```
# MNC server
backend.mnc.1.ip=127.0.0.1
backend.mnc.1.name=MNC

# EventDB server
eventdb.server=127.0.0.1
```

Two MNC systems and a backend server on different host

This setup includes 3 different hosts:

- One central server running the backend service, event database SatDB datbase.
- Two hosts running the MNC service.

The backend service connects via network to the MNC services on both host. And the MNC services deliver their events to the central server and using the SatDB on the central server.

```
# MNC servers
backend.mnc.1.ip=10.16.2.1
backend.mnc.1.name=UPLINK-TP1
backend.mnc.2.ip=10.16.2.2
backend.mnc.2.name=UPLINK-TP2

# EventDB server
eventdb.server=10.16.2.21

# SatDB server
db.satlist.host=10.16.2.22
db.satlist.user=postgres
db.satlist.password=mysecretpassword
db.satlist.usessl=true
```

backend server controlling redundant MNC systems

This setup includes also 3 different hosts:

- One central server running the backend service, event database SatDB datbase.
- A redundant pair of MNC services (Main and Backup)

The backend service connects via network to both hosts and control which one is the currently active MNC service. The backend services monitors the availability and stops & starts the MNC services on the hosts as required.

The active MNC service delivers its events to the cental EventDB server

```
# MNC servers
backend.mnc.1.ip=10.16.2.1
backend.mnc.1.name=UPLINK-TP1
backend.mnc.1.backup.ip=10.16.2.2
backend.mnc.1.backup.retries=3
backend.mnc.1.backup.delay=30

# EventDB server
eventdb.server=10.16.2.21
```

3.2.2 user-ip-restrictions.json

If the backend finds this file at startup, it restricts the access to the backend following the rules defined in this file. The rules in the file describe which user may login to the backend from which client machine. This applies to WebUI logins, not to operation system level (ssh) logins.

IP based login restrictions be defined in two ways:

1. From a given client IP address (or subnet) only a defined set of users may log in.
2. A given user may login only from a defined set of IP addresses or subnets.

While these restrictions at the first glance sound quite similar, the consequences of these restrictions may be completly different. A type 1 rule is used to exclude all users who are not listed in the rule, a type 2 rule exludes all non listed locations for a particular user.

The user-ip-restrictions.json file uses JSON to describe the rules to be applied. Example:

```
{
   "comment": "Example rules file",
   "ip_sources": [
      {
         "source_address": "10.10.21.0/24",
         "allowed_users": ["peter","paul","mary"]
      },
      {
         "source_address": "192.168.0.144",
         "allowed_users": ["guest"]
      }
   ],
   "users": [
      {
         "username": "john",
         "allowed_addresses": [ "10.10.1.0/24", "192.168.2.0/24" ]
      }
   ],
   "default_behavior": "grant"
}
```

The array "ip_source" contains the list of type 1 rules. Each rule defines an IP address or subnet and the list of users who are permitted to login from this network location. In the example above, from the subnet "10.10.21.0/24" only the users "peter","paul" and "mary" are permitted to login. all other users are rejected when trying to login from this subnet, even if they entered valid credentials. The second rule in this list limits the login from the IP "192.168.0.144" to the user "guest".

The array "users" contains the list of type 2 rules. In this example there is only one rule define which limits the login for user "john" to the subnets "10.10.1.0/24" and "192.168.2.0/24". Login attempts for "john" will be rejected unless the IP address of his computer is in one of these subnets.

Finally, the definition "default_behavior" tells how to deal with users who are not covered by any rule above. Valid options are "grant" or "deny".

The file may contain type 1 and type 2 rules at the same time or only one of the arrays "ip_source" or "users". For example only one client machine may be limited to a certain user:

```
{
   "comment": "Example rules file",
   "ip_source": [
      {
         "source_address": "192.168.0.144",
         "allowed_users": ["guest"]
      }
   ],
   "default_behavior": "grant"
}
```

This file only limits the Browser at "192.168.0.144", only user "guest" may login from here. Other users are rejected when logging in from here, but they may login from any other source.

If the user-ip-restrictions.json file does not exist, not IP based login restrictions are applies. user-ip-restrictions.json uses strict JSON syntax, any comments in the file must be coded as definitions which are ignored by the parser (e.g. "comment":"...").

3.2.3 vlc.properties

Defines some general configuration settings of the VLC/M&C Server, some of them may be defined on the command line, too.

Timezone settings

By default the sat-nms software displays all timestamps in UTC. you may change this with the timezone parameter. IF CHANGED, THE PARAMETER MUST BE CHANGED IN BOTH, vlc.properties AND client.properties TO THE SAME VALUE !!

With a NMS system, the timezone for NMS, all VLCs and all clients MUST be UTC, the timezone parameter must not be defined !!

```
## Examples for non-UTC time zones:
## timezone=GMT+03
## timezone=GMT-10
```

SNMP exceptions

Print exceptions for SNMP communication usefull for debugging SNMP. Default is `no`. Exceptions will be logged to `.panic.log`

```
showSNMPExceptions=no
```

Catch Exceptions

Print more/all exceptions for SNMP communication usefull for debugging SNMP. Default is `no`. Exceptions will be logged to `.panic.log`

```
showCatchedExceptions=no
```

Startup debug

`yes` makes the software to log in detail what it does during startup Default is `no` . Messages will be logged to `.panic.log`

```
## startup.debug=yes
startup.debug=no
```

Procedure debug

Setting to `true` adds additional information to the output of verbose mode. It shows at the beginning of a driver cycle which procedures are marked as "to be excecuted" (dirty) with a timestamp and it show when these procedures are executed.

```
## procedures.debug=true
procedures.debug=false
```

Remember to the set verbose mode of the device to `true` if you want to see the additional information.

Example:

```
                              GET procedures 9 12 marked dirty 14:25:35.263
                                      running procedure 9 14:25:35.264
tx(BCRX-1): /read?levl=?
rx(BCRX-1): levl=-77.45
 BCRX-1.power=-77.45
                                     running procedure 12 14:25:35.493
tx(BCRX-1): /rmt?sact=?
rx(BCRX-1): sact=0
 BCRX-1.ssrch.active=0
```

Startup Delay

Adds a delay between the initialization of two device threads. Value in msecs, default 0.

```
startup.delay=100
```

Communication timeout *deprecated*

The communication timeout to be used for tty over ip connections in msecs, Default 2000. This setting is *deprecated* because the timeout can be configured via `Device-Window > Maintenance Page > Serial > Timeout` individually for each device.

```
tty.ip.timeout=2000
```

TCP/IP interface definitions *deprecated*

Defines interfaces to control devices via tcp/ip. This is deprecated because it can be easily defined via client `Device Setup Window`. There is no need anymore to define `ttyNx` pseudo-interfaces.

```
```

Local event database

If set to 'yes' the server starts it's own event database. This is for stand alone M&C systems.

VLC's integrated to a NMS don't use a local event database, they send the messages to the central event database on the NMS server instead. Also MNCs working with a common Event Database, don't use a local event database.

By default, the command line switch `-e` in `/home/satnms/satnms-start` will already enable a local event database if `SATNMS_TYPE="mnc"` in /etc/satnms.rc is set to `mnc`.

```
```

Duplicate events

If set to `yes` on a MNC system sending it's event messages primarily to a central event database, the message are also duplicated to a local event database. Default is `no`.

To make this work, `local.edb` must be set to `yes` as well.

Never set `duplicate.edb=yes` for a stand alone MNC system. All messages will appear twice in the event log!

```
duplicate.edb=yes
```

EventToSerial AddOn

AddOn for fault messages reports to external systems:

- satnms.addon.EventToSerial: sends text messages to serial interface
- satnms.addon.EventToMail: sends text messages to mail recipients
- satnms.addon.EventToScript: calls a user defined script on selected events

Both needs additional configuration in separate configuration files.

```
server.addon=EventToSerial
server.addon=EventToMail
server.addon=EventToScripts
```

Serial Control Port

To control the VLC/MNC remotely via a serial interface set this option to a valid port name. Don't use this port to control a device at the same time.

```
server.rcontrol.port=/dev/ttyS1
server.rcontrol.baudrate=9600
```

Network Control Port

The VLC/MNC server provides a SSL remote control port (read-write) at port 2015, a TCP remote control port at port 2016 (read-write) and a second one at port 2017 (read-only). To switch off either of these ports comment out the appropriate line.

```
server.rcontrol.ssl.read-write=yes
server.rcontrol.tcp.read-write=yes
server.rcontrol.tcp.read-only=yes
```

Build-in WebServer

Setting this true makes the built in web server on TCP port 2002 require the user to login with a valid user/password combination in order to see / modify the parameter tree.

```
http.requiresAuthentication=true
```

TimeSync with NMS server

The following entry specifies the behavior of the server when it detects that the VLC clock is more than 30 seconds off the NMS time. Three settings are possible:

- FAULT : A fault of the SYSTEM device is generated (default)
- LOG : The time difference is logged as a INFO message to the event log
- IGNORE : The time difference is ignored completely

Only required in NMS environment where no NTP service is available.

```
server.nms.clockdiff=IGNORE
```

Default Help Page

To configure the main help page set this option to the name of the html-page without any directory or file extension. The main help page is the page that appears when clicking the help button in the main screen. default: 2210

```
help.topic=4520
```

Channel database

The CDB client in the software is preconfigured to work in a VLC/NMS configuration. The CDB server is located at the NMS in this configuration, the connection to the CDB is *online* while the

VLC *online* from the NMS' point of view.

Beside this, two other configurations are possible: A MNC can run it's own CDB server as a self contained solution or multiple MNC servers can share a CDB running on one of them.

A MNC running a CDB server must set this to `true` . `false` disables the local CDB server

```
## cdb.localOnly=true
cdb.localOnly=false
```

A MNC which wants to use the CDB running on another MNC must uncomment this line with the IP address modified to the address of the M&C running the CDB server.

```
 cdb.externServerAddress=192.168.2.214
```

Setting this true, enables some magic when editing channel names in the RX-LineSettings-CDB-KSC device: Channel names automatically get a signature containing the DVB-mode, the symbol rate and the RX pol appended. In the channel editor page this signature is not visible.

```
## cdb.LineSettings.magicname=true
cdb.LineSettings.magicname=false
```

If present, these entries make the channel / biss-key databases sync the local backup to the central database in the NMS in regular intervals. The time interval is defined in hours (floating point).

It is disabled if cdb.localOnly=true

```
## cdb.channels.syncInterval=24.0
## cdb.bisskey12.syncInterval=24.0
## cdb.bisskey16.syncInterval=24.0
```

settings for CSM spectrum analyzers

Switches the preamp for a LPTech LPT3000RX4 spectrum analyzer ON/OFF affects all LPT3000RX4 controlled by the M&C, default is preamp ON

```
 csm.lpt3000rx4.preamp=ON
```

This setting is deprecated, it refers to the outdated CSM-Spectrum-Analyzer device. Actual spectrum analyzer drivers handle this internally

Universal SNMP agent

The following entries refer to the universal table based snmp agent listening at UDP port 2261.

Defaults are:

```
snmp.agent.enable=no
snmp.agent.readCommunity=public
snmp.agent.writeCommunity=private
snmp.agent.trapTargets=
```

`trapTargets` is a space separated list of IPv4 addresses, each optionally followed by a colon and a port number. for entries without port number port 162 is assumed.

Example:

```
snmp.agent.enable=yes
snmp.agent.readCommunity=public
snmp.agent.writeCommunity=private
snmp.agent.trapTargets=192.168.0.89:2162 127.0.0.1:2162
```

User configurable SNMP agent

The following entries refer to the user configurable snmp agent listening at UDP port 2161.

```
snmp.max.oid=1000
```

The agent gets initialized if a snmp.setup file exists. The agent limit the number of OIDs it creates from the snmp.setup file. The default limit is 1000 ans may be changed uing the parameter shown above.

UDP Heartbeat

These entries control the UDP heartbeat sent by a VLC to make the NMS recognize that the VLC is reachable in the network. To activate the heartbeat, set `udp.heartbeat.enabled=true`

The interval generally should not be changed. It is given in msecs, it should be set to one third of the timeout value set at the NMS.

```
udp.heartbeat.enabled=false
udp.heartbeat.enabled=true
udp.heartbeat.interval=5000
```

logParameterChanges

If set true, event log messages indicating parameter changes (logParameterChanges set true for a device) also contain the name and the IP address of the initiator of the parameter change. default is false for backward compatibility

```
## log.initiator=false
log.initiator=true
```

Prologix debug

Enable debug output for ProLogix GPIB to Ethernet converter. Disabled by default.

```
prologix.debug=false
```

Macro Scheduler

If set true, the macro scheduler automatically deletes outdated events. if false, these events remain in the schedule and can be viewed in the GUI. Default is true.

```
## scheduler.autodelete=true
scheduler.autodelete=false
```

Client Queue Size

This entry configures the client queue size (number of messages). Default value is 10000, the value which was set before as fixed size.

```
clientpeer.queueSize=50000
```

Port binding

By adding a line `bind.xxxx=a.b.c.d` the server socket at port `xxxx` can be bound to a specific address / interface

works fo the following port numbers

Port	Service
2000	Debug terminal
2001	Server-Client message transfer
2002	HTTP webserver (deprecated)
2003	M&C server control
2004	EventDB feed
2005	Live event log
2006	EventDB query
2007	JFTP data transfer

3.2.4 client.properties

Several aspects of the client behavior can be configured in the text based configuration file `/home/satnms/client.properties` . The following chapter descripes the different option and related functionalities.

All chances in this file we be applied for every client which connects to the server after this configuration file was written.

Addressing users and hosts

All keys in this file may be prepended by a user name (the user name to log into the sat-nms GUI) or a host name. If the value for a key 'some.key' is searched, the software first tries 'user.some.key' where 'user' is replaced by the current user's login name. If this key was not found, it tries 'hostname.some.key' where hostname ist the client's host name as reported by the operating system. If this also fails, the as the last fall back the plain key is searched.

This search method gives the capability to overwrite some client properties for specific users or client hosts.

Examples:

```
## disable edit/take mode for client with hostname "pc1":
pc1.gui.useEditTake=false

## disable edit/take mode for sat-nms user "user1" on any client pc:
user1.gui.useEditTake=false
```

Time zone settings

By default the sat-nms software displays all timestamps in UTC. you may change this with the timezone parameter. IF CHANGED, THE PARAMETER MUST BE CHANGED IN BOTH, vlc.properties AND client.properties TO THE SAME VALUE !!

Examples for non-UTC time zones:

```
timezone=GMT+03
timezone=GMT-10
```

Exclusive login

Setting this true enables _exclusive login*, meaning that only client may be logged in at a time.

```
login.exclusive=false
login.exclusive=true
```

When trying to login on a system with enabled _exclusive login* and there is already another user logged in, the login dialog will report this with the message_ `another client is logged in` _and gives you the option to *FORCE_ your own login and kick out the connected user. Forcing login is only possible with an administrator account (privilege level 150)

Primary client

Setting this to an IP address of a certain client computer gives this client instance an exclusive priority: no other client may force the log out of the primary client, but the primary client may log out any other client.

This option is meaningful only if login.exclusive=true

A typical application scenario is, that a client installed at the antenna site should be able to block a remote connected users.

```
login.primary=192.168.0.101
```

Automatically timed logout

Setting this to a value greater 0 enables automatically logout of users sat-nms client users after the given number of minutes of (user-) inactivity.

To make special setting for a given user or host add username or hostname in front of autologout parameters. By default this automatically logout is disabled (0).

Example disable autologout for user _operator* but set it to 5 minutes for all others:

```
## other users
 operator.autologout=0
 autologout=5
```

Device list windows access level

The privilege level necessary to open the device list in the toolbar in the VLCFrame / MCFrame windows. Default required privilege level to open the device list windows is 100 (operator).

```
devicelist.privilege=110
```

Config parameter access level

The privilege level necessary to change device setup parameters at the maintenance / setup window.

Default is 150, select a lower number will allow different users than normal administrators to change these settings. Recommendation is to define users with level 140.

```
configParameters.privilege=140
```

Macro scheduler access level

This defines the privilege level necessary to edit the schedule of the system macro scheduler.

```
scheduler.privilege=110
```

Screen element background color

True makes the configured background color of disabled screen elements override the system 'disabled' background. Works only with entry fields, read-only fields and with the searchable drop down list field.

```
gui.overrideBackground=true
```

Look and feel of read-only elements

True lets read-only elements look "enabled" even if the operator is not logged in. This is the way, satnms2 did show read-only fields. Default is *false*

```
gui.keepReadOnlyEnabled=true
```

Edit/Take mode

True makes the user interface use the edit/take mode for all device screens. This option change the behavior of device screens. With enabled edit/take mode commands will be not send directly if you enter a new value or select a dropdown entry. But all device windows will show 2 additional buttons : EDIT and TAKE.

The EDIT button enables all read/write parameter fields to change the values and the TAKE button send all changed parameters to the device.

By default this mode is disabled.

```
gui.useEditTake=true
```

**Event sort order*

True makes the event report and event log windows sort recent fault messages first (on top). Default is *false*.

```
gui.eventSortRecentFirst=true
```

Fault list window background color

If set *true*, in the fault list window warnings are shown with a yellow background, faults with a red background (may be changed by a toolbar button of the window)

```
gui.faultListColored=true
```

Fault list window line color

If set *true*, the fault list window sorts the lines fist by device, then by time (may be changed by a toolbar button of the window)

```
gui.faultListByDevice=true
```

Suppress warnings fault list window

If set true, the fault list window suppresses all warnings (may be changed by a toolbar button of the window)

```
gui.faultListSuppressWarnings=true
```

Suppress signal on emblem

True prevents device icons showing the 'signal on' emblem, even if the device state is recognized as 'signal on' and an image file for the emblem has been found.

```
gui.suppressSignalOnEmblem=true
```

New fault popup

If set true, each new fault makes the fault list window pop up (or come to front). applies only to the M&C client. The -o command line switch of satnms3.gui.MCFrame has the same effect.

```
gui.faultListPopUpOnNewFault=true
```

Show tree view button

True makes the M&C client window show a toolbar button to launch a tree view window, false suppresses this button. This feature is enabled by default, the tree view button is shown if this definition is missing

```
gui.showTreeViewButton=true
```

Tree view default device icon

The treeview client/window uses this device icon to build dynamically generated device screens.

```
gui.treeview.dynamicDeviceIcon=device-rectangle.gif
```

Alternative Switch icon set

Define different switch icons for the screen editor

```
gui.editor.predefinedSwitch.1=cwgswitch
gui.editor.predefinedSwitch.2=cwgswitch-mirrored
gui.editor.predefinedSwitch.3=ccxswitch
gui.editor.predefinedSwitch.4=ccxswitch-mirrored
gui.editor.predefinedSwitch.5=clswitch
gui.editor.predefinedSwitch.6=crswitch
```

VLC player integration

The full path of the VideoLAN media player. Required to run the VLC player on Windows client machines from a "VideoLAN" frame button.

Backslashes must be doubled, the whole path must be enclosed in double quotes if it contains spaces. You can also defined this setting by host to target client PCs with different operating systems.

```
videolan.exe="vlc.exe"
videolan.exe="C:\\Program Files (x86)\\VideoLAN\\VLC\\vlc.exe"
```

Event DB

Defines the IP Address of the EventDB Server used by the client to show the Eventlog. This is usually *not* configured: The Client will use the IP address of the MNC server.

In case the EventDB is not available locally on the MNC server but on any other machine you can configure here the IP address of the "external" EventDB Server.

```
eventdb.server=10.10.1.111
```

3.2.5 satnms.rc

working mode

Select between operation as MNC server (mnc) or together with an NMS Server (vlc)

```
SATNMS_TYPE="mnc"
```

serial ports

Moxa PCI Express cards did not have any jumpers to select RS232/422/485 and termination. This has to be set with `muestty` on startup.

Servers with OS Debian-satnms-4 and later supports 1 card with up to 8 ports. Define here the existing ports and interface types:

```
## add 2 lines for each port and replace "x" with the number 1..8
SATNMS_SERIALx_INT = [RS232,RS422,RS4852W,RS4854W]
SATNMS_SERIALx_TERM = [NONTERM,120TERM]
## (only relevant for 422/485)

## example Port 1 to RS422 and no termination
SATNMS_SERIAL_INT_1=RS422
SATNMS_SERIAL_TERM_1=NONTERM
```

Please refer to Moxa documentation for details. Check current settings with

```
muestty -g /dev/ttyS11
```

Card comes up with RS232, therefore it is not necessary to configure these cards. Just comment out lines with RS232 to prevent error messages.

3.2.6 system.name

Add one line (text) which should be shown in the window title of the sat-nms MNC Java Client. Default is `sat-nms MNC Monitoring & Control System`

Examples:

- Antenna 1 M&C
- MNC Teleport Steisslingen

3.2.7 traprec.json

This file configures the trap receiver built into the M&C server. The trap receiver is used to call `PROC ASYNC TRAP` procedures in selected devices / device drivers on every received SNMP trap. The trap receiver is started if this file is present and can be parsed successfully.

File Syntax

The file uses JSON syntax to define the traprec configuration in a structured way. The elements of the JSON document are explained best by example:

```
{
   "comment": "sat-nms SNMP trap receiver configuration",
   "general": {
      "listenPort": 2162,
      "logToStdOut": true,
      "logToEdb": false,
      "trapCommunity": "public"
   },
   "snmpv1": {
      "enabled": true
   },
   "snmpv2c": {
      "enabled": true
   },
   "snmpv3": {
      "enabled": true,
      "users": [
         {
            "name": "secretname",
            "authpass": "AuthPassword",
            "authentication": "MD5",
            "privacy": "NONE",
            "privpass": "PrivPassword"
         }
      ]
   }
}
```

general

The *general* section defines some general properties of the trap receiver. The definitions in this section are:

name	type	default	description
listenPort	integer	2162	The port number to listen at. The valid range for this is 1024 to 65535. If you require the trap receiver listen at the standard port 162, you should setup a port forwarding to 2162 for this.
logToStdOut	bool	false	If enabled, every received trap gets logged to stdout (to the .panic.log file). The complete trap information as available in the PROC ASYNC TRAP procedures ist logged. This is useful for debugging which traps are sent by a device.

name	type	default	description
logToEdb	bool	false	If enabled, every received trap gets logged in the M&C event log.
trapCommunity	string	null	If set, the trap receiver accepts only traps which match the given trap community (SNMPv1 and SNMPv2c only)

snmpv1

The *snmpv1* section contains a flag to enable or disable SNMPv1 at the trap receiver.

name	type	default	description
enable	bool	false	*true* enables SNMPv1

If disabled, the trapreceiver ignores all SNMPv1 traps it receives.

snmpv2c

The *snmpv2c* section contains a flag to enable or disable SNMPv2c at the trap receiver.

name	type	default	description
enable	bool	false	*true* enables SNMPv2c

If disabled, the trapreceiver ignores all SNMPv2c traps or inform messages it receives. SNMPv2c inform messages are not acknowledged by the trap receiver if SNMPV2c is disabled.

snmpv3

The *snmpv3* section contains a flag to enable or disable SNMPv3 at the trap receiver and the list of authenticaed users the trap receiver accepts.

name	type	default	description
enable	bool	false	*true* enables SNMPv3
users	array	-	a list of user definitions (see below for details)

If disabled, the trapreceiver ignores all SNMPv3 traps or inform messages it receives. SNMPv3 inform messages are not acknowledged by the trap receiver if SNMPV3 is disabled.

users

If SNMPv3 is enabled, the *snmpv3* section must contain a *users* array with at least one element. SNMPv3 does not work without a user definition, even in noAuthNoPriv mode. The definitions for one user are:

name	type	default	description
name	string	-	The secret name of the user.
authpass	string	-	The user's pass phrase, this is required if *authentication* is not "NONE"
authentication	string	"NONE"	The authentication mode, one of "NONE", MD5" or "SHA". "NONE" implies noAuthNoPriv mode, *privacy* is ignored in this case.
privacy	string	"NONE"	The privacy mode, one of "NONE", "AES" or "DES".
privpass	string	-	The privacy pass phrase, this is required if*privacy* is not "NONE"

The *name* definition is mandatory for a user as well as the*authentication* and *privacy* definitions. The pass phrase definitions are only required if *authentication* and/or *privacy* are enabled.

3.3 Event Database Configuration and Event Notification Add-Ons

Basic EDB settings

The Event Database basic configuration (e.g. if the EDB should be started on a MNC system) is defined in `/home/satnms/vlc.properties` . Please refer to the chaptervlc.properties for details.

EDB settings in the SYSTEM device

There are also settings regarding the Event Database (e.g. IP address of the EventDB Server) which needs to be configured in the SYSTEM device

EDB.properties

Event Database properties are definesin the `/home/satnms/edb/EDB.properties` file. Page EDB.properties describes this feature in detail.

Event Notification Add-Ons

There are two add-ons which may be configured with the M&C software to notify operators if new event messages have been issued by the system. Notification may be through a RS232 serial interface of the M&C server or by E-mail. Event notification add-ons are available for sat-nms M&C systems only, not for the NMS server.

Event To Serial Notification

The event to serial notification add-on sends event messages through a serial RS232 interface.

This interface may be connected to a separate unit which sends SMS messages or teletype messages from this information. Page Event To Serial Notification describes this feature in detail.

Event To Mail Notification

The event to mail notification add-on sends event messages as e-mail notifications to one or more recipients. Page Event To Mail Notification describes this feature in detail.

Add-On Configuration

A notification add-on is activated by setting the key "server.addon" in the file "vlc.properties" either to "EventToSerial", "EventToMail" or to "EventToScript". All particular settings for the selected notification add-on are defined in a separate configuration file. A file called "rules" has to be created with the filter rules which messages are to be forwarded and which not.

The "rules" File

All event notification add-ons provide a filter function for the event messages to forward. Using filter rules defined in the file "rules", the software forwards only messages which match one of the rules specified in the "rules" file.

This file is a plain text file containing one rule per line. Empty lines and comment lines (starting with a '#'-character) are ignored. Each rule has the format <device-name>:<text-pattern>, where both parts may be replaced by "*" as a general wildcard. A rule matches, if the event source matches the given device name and if the message text contains the given text pattern.

Here some example rules:

```
HPA-1:FAULT           matches messages from HPA-1 containing
                  "FAULT" in the event message text.

HPA-2:*             matches all messages from device HPA-2

*:Lock              matches messages from all devices which
                  contain "Lock" in the message text

*:*                matches all messages.
```

If the "rules" file is missing or empty, the software assumes a "*:*" rule. You may have to create a new "rules" file on your, system, there is no default file installed.

3.3.1 Event To Serial Notification

The event to serial notification add-on sends event messages through a serial RS232 interface. This interface may be connected to a separate unit which sends SMS messages or teletype messages from this information.

To use this feature, you need an unused RS232 port at the M&C server. Configuring / installing this feature is a four step procedure:

1. Create a "rules" file. See page Event Notification Add-Ons for details about this file.
2. Edit the "e2s.properties" file for your needs, some information about the settings in this file are given below.
3. Edit the "vlc.properties" file and set "server.addon=EventToSerial" This line probably is already presented in the file but deactivated by a '#' character in front of it.
4. restart the M&C server

The "e2s.properties" file contains the parameters which configure the behavior of the event to serial notification. The file follows the Java properties file syntax and may contain thefollowing keys / values:

hostName specifies the hostname or IP address of the machine which runs the M&C server. In most cases you will leave this at 'localhost'.

ruleFileName specifies the name of the rule file used by the notification add-on. By default this is 'rules', you may change this if you like to keep multiple rules files which you want to use alternatively.

portName specifies the name of the serial port to use. The default value is '/dev/ttyS0', every serial port available at the machine which is not in use for another purpose may be used.

baudRate specifies the baudrate to be used with the serial port. Default is '9600'.

verbose is for software tests only. For normal operation this always must be 'false'.

The file is read once at system start, hence any changes to this file become effective with the next restart of the M&C server.

3.3.2 Event To Mail Notification

The event to mail notification add-on sends event messages as e-mail notifications to one or more recipients.

To use this feature, the M&C server must be able to reach an SMTP mail server in the network. This may be a private mail server in the company LAN or a public mail server. For the latter case, the M&C server requires internet access.

Configuring / installing this feature is a four step procedure:

1. Create a "rules" file. See page Event Notification Add-Ons for details about this file.
2. Edit the "e2m.properties" file for your needs, some information about the settings in this file are given below.
3. Edit the "vlc.properties" file and set "server.addon=EventToMail" This line probably is already presented in the file but deactivated by a '#' character in front of it.
4. restart the M&C server

The "e2m.properties" file contains the parameters which configure the behavior of the event to mail notification. The file follows the Java properties file syntax and may contain thefollowing keys / values:

hostName specifies the hostname or IP address of the machine which runs the M&C server. In most cases you will leave this at 'localhost'.

ruleFileName specifies the name of the rule file used by the notification add-on. By default this is 'rules', you may change this if you like to keep multiple rules files which you want to use alternatively.

originate specifies the name or some identification of the M&C system. The text given here will be referenced as the message source in the text body.

minSendInterval specifies the minimal time (minutes) between two messages sent by the notification add-on. For instance, setting this to '60' will configure the software to send mails not more often than once an hour. Events happening in this time will be accumulated and included to the next mail as a list.

summary specifies the position of the message summary in the mail body. It may be one of 'none', 'top' or 'bottom'. If no summary is defined, 'top' is assumed as the default.

smtpHost specifies the name or IP address of the SMTP server which shall be used to send the mails.

smtpPort specifies the port number of the SMTP server which shall be used to send the mails. If not specified, the program assumes port 25 (SMTP).

smtpUser specifies the username to login at the SMTP server. if left empty or commented out, the software uses no SMTP authentication.

smtpPassword specifies the password to login at the SMTP server.

smtpSSL may be one of yes/no. Specifies if SMTPS (SMTP via SSL/TLS) shall be used as message transport. If not specified, the program uses no encryption.

smtpFrom specifies the from address to be set in the messages sent. Please note, the many public SNMP servers require this address to be a valid address because of their spam protection policy.

smtpTo specifies the list of recipients for the mail. Multiple e-mail addresses may be separated by comma characters, they all have to appear in the same line as the 'snmpTo' key.

verbose is for software tests only. For normal operation this always must be 'false'.

The file is read once at system start, hence any changes to this file become effective with the next restart of the M&C server.

3.3.3 Event To Script Notification

The event to script notification add-on calls a user defined script on each event passing the 'rules' filter.

Configuring / installing this feature is a four step procedure:

1. Create a "rules" file. See page Event Notification Add-Ons for details about this file.
2. Edit the "e2c.properties" file for your needs, some information about the settings in this file are given below.
3. Edit the "vlc.properties" file and set "server.addon=EventToScript" This line probably is already presented in the file but deactivated by a '#' character in front of it.
4. restart the M&C server

The "e2c.properties" file contains the parameters which configure the behavior of the event to script notification. The file follows the Java properties file syntax and may contain thefollowing keys / values:

scriptName - specifies the name of the (bash-) script to be executed. The file*must* be located in the ./scripts/ directory of the ***sat-nms*** installation. The file must be set readable and executable for the user that shall execute it.

ruleFileName - specifies the name of the rule file used by the notification add-on. By default this is 'rules', you may change this if you like to keep multiple rules files which you want to use alternatively.

user - if this key is defined, the software does execute the the script with `sudo -u <user> <scriptName> ...` instead of calling the script directly. You probably have to set the password property as well to make this work.

password - specifies the login password of the user set with the 'user' property.

verbose is for software tests only. For normal operation this always must be 'false'.

The file is read once at system start, hence any changes to this file become effective with the next restart of the M&C server.

3.3.4 EDB.properties

Storage depth

Depth defines the number of days the event database records in it's circular storage on disc. Typicall value is 1 year (365 days)

```
depth=365
```

Maximum Report Size

MaxReportSize defines the maximum number of events which are included in a database report. If more events match the search criteria, the reports gets truncated and the operator receives a message about this. This value is just the default used by the client event report window. There you can enter a new one (even greater) for a single request.

```
maxReportSize=1000
```

Event sort order

SortEvents controls if the filtered events are to be sorted by creation time before they are reported to the client. This is necessary in an environment where the events stored in another order than they were created (e.g. in the NMS or if multiple MNC systems using the same event database).

Therefore sortEvents is assumed to `true` if not stated otherwise. In a MNC environment reporting a large number of events can be speed up significantly by settings this to `false`

```
sortEvents=false
## sortEvents=true
```

Event DB debugging

Set to `true` for activation of additional debug output. It is disabled by default`false` and should be not enabled in production environments.

```
```

3.4 Remote Control

The satnms M&C software support several remote control interface which let an superordinate NMS control the M&C. In this section the direct remote interfaces of the M&C server are discussed. Both require direct IP access to the M&C server, accepting any possible security riscs arising from that.

The preferred method of monitoring or controlling an M&C from an external program is the HTTPS based API of the M&C backend. USing this, all M&C systems controlled by the backend may be monitored thru one single interface.

The direct remote interfaces of the M&C server areSNMP and a text based terminal protocol using TCP/SSL Ports for remote contol.

The SNMP Agent built into the M&C server comes in two different flavors: One is table based and standardized, offers a limited number of parameters in a MIB which is the same for all satnms M&C systems. The second one is application specific, for every satnms M&C installation a unique MIB can be generated which contains exactly the parameter which shal be monitored or controlled.

3.4.1 SNMP Control

The sat-nms software contains two SNMP agents which may be used to control the software from an external NMS.

Universal SNMP Agent

The universal SNMP agent publishes a limited number of device parameters in a fixed, table oriented MIB. This SNMP agent permits to monitor and partially control any sat-nms M&C using the same MIB - regardless of the device configuration of the particular M&C. The universal SNMP agent listens at port 2261, it is based on SNMPv2.

The MIB provides one table which contains all devices which are configured in the sat-nms M&C. No special mapping is necessary for this table as there are only very common parameters like the device name, its fault state or its administrative state in this table.

Beside this, the MIB contains a couple of tables, each representing all devices belonging to a so called 'device class'. For instance, there is one table containing all HPAs in the M&C and one table containing all waveguide switches. These class specific tables contain some parameters which are common to the devices of the particular class. The HPA table contains parameters for tx-on, for attenuation and forthe measured output power for each device, the waveguide switch table provides a parameter for the switch position.

As device drivers sometimes use different parameter names for the same or similar function, an individual mapping from the 'universal' parameter in the MIB to the real parameter in the device is necessary for every single sat-nms device driver. Device drivers which do not contain any information about the device class this devices belongs to and how the parameters are to be mapped into the MIB only appear in the in the general device table of the MIB.

A sat-nms device (physical or logical) may be represented by multiple device classes in the MIB or by multiple instances of the same class. For example a combined upconverter and HPA may appera as a HPA in this table (with its HPA specific parameters) and a second time in the upconverter table with its frequency parameter.

Also devices consisting of multiple modules can be treated this way in the MIB, there are quad block upconverter on the market which provide four independent converter modules in one device. Such a device can be mapped in way, that it appears as four BUC devices in that table.

By default the universal SNMP agent is disabled. Use the following keys in the vlc.properties file to enable the agent, set the read/write community strings and an optional list of trap/notification destinations. Each IP address in this list will receive a notification message for each device which changes its fault state. Example:

```
snmp.agent.enable=yes
snmp.agent.readCommunity=public
snmp.agent.writeCommunity=private
snmp.agent.trapTargets=192.168.0.89:2162 127.0.0.1:162
```

'snmp.agent.trapTargets' defines a list of zero or more IP addresses in dotted quad notation. The addresses may contain a port number like in the example above, if no port number is specified, the standard SNMP trap port 162 is assumed. There is no fixed limitation in in the software for the number of trap destinations, you should however be aware, that a large number of destinations (greater than 10) may cause performance problems, depending on network speed and response time of the targets.

The agent sends a SNMPv2 notification (mncDeviceFaultChanged from the MIB) every time the fault state of a device changes. The notification message has the name of the device which caused the notification and its new fault state as varbinds attached. At program start, one initial notification message for each device is sent.

Application Specific SNMP Agent

The application specific agent provided by the sat-nms software is based on SNMPv1, it accepts requests at UDP port 2161. It permits read and write access from the 'public' community. Beside that, the sat-nms M&C may be configured to send SNMP traps on changes of certain parameters, e.g. fault flags.

Each M&C system is configured individually for the device setup it controls. Therefore also the SNMP agent configuration and the specific MIB for a M&C must be unique for each installed system. With the sat-nms M&C / VLC, the SNMP agent gets configured by a (text based) configuration file which defines the M&C parameters which shall be visible through SNMP.

The typical procedure to prepare the sat-nms M&C for SNMP access is as follows:

1. Create a template SNMP configuration file using the 'create-snmp.template' utility program. This template file contains all parameters available in this particular M&C installation.
2. From this template file edit the SNMP configuration setup or simply copy the template to the configuration file if you want to export all parameters via SNMP.
3. Restart the M&C server to make it read the SNMP configuration file.
4. Invoke the 'create-satnms.mib' utility program to create a MIB definition file reflecting the actual setup.

3.4.1.1 SNMP Parameter Translations

SNMP treats parameters (objects) in a different way as the sat-nms software does. The SNMP agent translates the internal representation of M&C parameters to the SNMP representation and vice versa. The default translation rules are listed in the table below:

Default Translation

- sat-nms *data type* --- *SNMP data type**comment*
- INTEGER --- INTEGER : No translation necessary.
- FLOAT --- INTEGERSNMPv1 : cannot handle floating point numbers. FLOAT variables therefore are translated to integer values when exported as a SNMP object. The value is scaled by 10^ *n* where *n* is the number of precision digits set for this variable in the device driver definition file. Example: The azimuth value of an antenna controller appears with three precision digits as 180.000 in the sat-nms user interface. The SNMP representation of this angle will be 180000 (floating point value multiplied by 10^3). When the SNMP agent receives values to set at the M&C, the translation happens the other way round.
- TEXT --- OCTET-STRING : Text type variables translate as OCTET-STRING SNMP objects. In the MIB, text variables should be declared as 'DisplayString' (defined in RFC1213-MIB). This ensures, that the strings are shown as expected by the MIB browser. (Some browsers assume OCTET-STRINGs to be binary and display a hex-dump instead of the textual contents.)

- CHOICE --- INTEGER : Internally the sat-nms software treats CHOICE parameters as text. SNMP uses enumerated integers for this. The value of a sat-nms CHOICE parameter translates to the position of the value in the list of choices as defined in device driver file. If - for some reason - the value does not match any of the selections given by the device driver, it translates to 0. The MIB should define memnotics for all selections.
- BOOLEAN --- INTEGER : A boolean value translates to an enumerated integer with the values 0=FALSE and 1=TRUE.

The SNMP agent knows from the sat-nms variable definition if a parameter may be accessed read-write or read-only. Each individual parameter may be limited to read-only access when exported to SNMP (see below).

'force to text' Translation

The SNMP agent may be configured (individually for each parameter) to force a text-translation to OCTET-STRING, regardless of the type of the parameter. In this case, INTEGER or FLOAT values translate to a readable text, formatted with the precision defined in the device driver file and the unit string appended. BOOLEAN value translate to the keywords TRUE or FALSE and any other parameter translate the text the sat-nms user interface shows. The MIB should contain a 'DisplayString' entry for each parameter which is configured this way. If the agent receives a SET request for a parameter which ist translated in 'force to text' mode, it parses the text and sets the value as the user interface would do.

The 'force to text' option specially is valuable together with the 'force to readonly' option described below. It provides a simple method to create easy to read status reports which can be displayed in a MIB browser.

'force to readonly' Translation

The 'force to readonly' option permits to limit the access to read-only for each individual parameter. This may be used to prevent SNMP users from changing sensible settings in the M&C.

'suppress range limits' Translation

The 'suppress range limits' option suppresses the output of the min/max range of a numeric parameter. This may be useful if this range dynamically changes and shall not be fixed to the limits valid at the time when the MIB is created.

Limitations

Not all M&C parameters may be translated to SNMP objects in a way that makes sense. First of all, there are a couple of M&C variables which carry complex information, binary coded as Java objects. Translating these objects to SNMP is not possible as the information contained in these objects is for internal use in the sat-nms software only.

Another problem of translating values between their sat-nms representation and SNMP objects arises if the M&C dynamically changes the definition of this variable during runtime. This is a special feature of the sat-nms software which makes it's internal message concept superior to SNMP for this application.

Example: A satellite modem is capable to encode with FEC rates 1/2, 3/4 and 7/8 with BPSK modulation. If switched to QPSK, also the rates 4/5 and 5/6 are available. The device driver will change the definition of the 'fec' parameter as the 'modulation-type' parameter changes.

It is easy to understand that the translation of the 'fec' along the enumeration list given in the device driver will give wrong results after the device driver changed the parameter definition. It is strongly recommended to configure such parameter as 'force to text' and 'force to readonly'. This way at least a correct display via SNMP can be achieved.

A similar situation arises if a device dynamically changes the limits for a numeric parameter. An example for this are the pointing angles of an antenna: The soft limits set by the operator will be used as range limits for this parameter. If they are changed after the MIB has been created, they will no longer match the limits stated in the mib. Such parameters should be marked with the 'suppress range limits' translation option in the configuration file.

3.4.1.2 Configuring the SNMP Agent

The application specific SNMP agent gets configured by a the file 'snmp.setup'. If this file exists in the VLC's home directory while the M&C server starts, the SNMP agent is started. The sat-nms software contains a utility program called 'create-snmp.template' to create a template of this file including all parameters of a particular M&C / VLC. The file is a text file using the following format:

- Empty lines and lines starting with '#' are ignored.
- At the beginning of the file, the global settings 'systemname', 'systemid' and 'traphost' must be defined (see example below).
- After this file header, each line must contain exactly one sat-nms parameter identifier, consisting of the device name followed by a dot and the variable identifier.
- Each parameter identifier may be followed by zero or more parameter attributes, separated by space characters.

Global settings

The file 'snmp.setup' must define the global settings 'systemname', 'systemid' and 'traphost' before the first parameter identifier appears in the file. The order of appearance does not matter.

- **traphost<aaa.bbb.ccc.ddd>** --- Defines the IP address of the host which shall receive any traps from this M&C. Traps always are sent to the standard port number 162.
- **systemid <n>** --- Defines the base node number of the MIB tree created for this M&C. The created MIB tree is a branch of *iso.org.dod.internet.private.enterprises.satservice* , the 'systemid' parameter defines the number of the child node below *satservice* . Numbers 1 .. 99 are reserved for special applications. The 'create-snmp.template' utility program uses the M&C serial number as a default for this parameter to ensure a unique value for each M&C.
- **systemname <name>** --- Defines the memnotics for the 'systemid' parameter described above. Following the conventions for MIB definition, this should be a single word, starting with a letter and consisting entirely of letters and digits. The 'create-

snmp.template' utility program prepends 'vlc' to the serial number in order to create a unique value for this parameter.

Parameter attributes

The software accepts the following parameter attribute keywords following a parameter identifier. The activate the corresponding translation option for this parameter.

- **RO** --- Forces this parameter to be treated as read-only, even if it has read-write access within the sat-nms environment. It is not necessary to mark parameters with **RO** if they already are marked as read-only by the sat-nms device driver.
- **TEXT** --- Forces this parameter to be translated into a readable text (OCTET-STRING) when exported to the SNMP world, regardless of it's native data type. See chapter SNMP Parameter Translations for more information about the software converts native data types to SNMP types an vice versa.
- **NORANGE** --- Suppresses the output of a mi/max range for numeric parameters. This is useful for parameter which dynamically change their limits. See chapter SNMP Parameter Translations for more information about the software converts native data types to SNMP types an vice versa.
- **TRAP** --- Tells the software to send an SNMP trap if this parameter changes. Although mostly used for fault flags, this feature is not limited to this application. Any M&C parameter may be monitored by sending TRAPS.

MIB tree creation

When reading the 'snmp.setup' file, the software dynamically allocates MIB nodes for each parameter. The program splits up parameter identifiers at the dots and builds a MIB tree breaking down the M&C parameter structure along devices and subsystems. Node numbers are assigned automatically starting at 1, parent nodes are created as needed. This means, the order of the parameters in the 'snmp.setup' file determines the numbering of OIDs.

As a consequence of the automatic MIB tree creation, existing MIBs usually become invalid after a configuration change in a M&C, due to the renumbering of the OIDs.

However, with the knowledge about the OID assignment algorithm used by the software, minor configuration changes can be worked into the SNMP configuration while keeping the existing OID numbering unchanged. There are simply two rules to follow:

1. Do not remove existing definitions. Neither single parameters nor complete devices. Deleting definitions would cause a renumbering (decrement) of the following nodes.
2. Add additional parameter definitions always at the very end of the 'snmp.setup' file. Although for human readers the order of parameters in the file will look somewhat 'messy', the software will assign the OIDs in a way that the numbering of existing parameters will not change.

If you follow the rules above, you still can use the previous MIB of a M&C after an extension of the configuration. You however will not be able to access the added parameters as long you use the old MIB.

'snmp.setup' Example

The example below shows the contents of a short 'snmp.setup' file.

```
### snmp.setup example
### created on Thu Jun 22 19:11:18 GMT 2006

traphost 127.0.0.1
systemname vlc2345
systemid 2345

SUMMARY.fault                  TRAP
E1.fault                   TRAP
E1.mode                     RO
SW.fault                   TRAP
SW.mode                      RO
```

Fault messages

If the software encounters problems while it reads the 'snmp.setup' file, it issues fault messages to the sat-nms event log. Hence, after a change of the SNMP configuration, you should inspect the event log after the server restart. The following error messages are possible:

Unknown variable in line n of 'snmp.setup': The snmp.setup file references a sat-nms variable which does not exist. The reason may be a misspelled parameter name or you changed the sat-nms device setup without adapting the SNMP configuration.

Unknown attribute ABC in line n of 'snmp.setup': You specified a parameter attribute which is neither 'TRAP', 'TEXT' nor 'RO'.

Warning: 'snmp.setup' has 0 entries: There is a 'snmp.setup' file, but it does not specify any parameters

Failed to read 'snmp.setup': A general error occurred. Delete and recreate 'snmp.setup'.

3.4.1.3 The 'create-snmp.template' Utility Program

To assist you in setting up the SNMP configuration file, the sat-nms software comes with a utility program called 'create-snmp.template'. This program creates a template configuration file which contains all (or the most common) M&C parameters of the actual system.

To create a template configuration file, perform the following steps:

1. Open a telnet window to the M&C server
2. Log in as user 'satnms', password 'satnms'.
3. Type 'create-snmp.template <ENTER>' to invoke the program

This creates a file called 'snmp.template', a template for the 'snmp.setup' file. It defines all M&C parameters of the system except of the device setup variables to be accessible via SNMP.

The M&C variables appear in alphabetical order, all fault flags are marked to send SNMP traps in case of a change of the value. To use the template file, rename it to 'snmp.setup' and restart the software. To make certain parameters invisible for the SNMP NMS, remove the corresponding lines from the file or put comment characters ('#') in front of them.

The 'create-snmp.template' utility program accepts a number of command line options which let you modify the default behavior of the program. These options are:

- **--include-config** --- Includes the device setup variables into the template file. Normally the program skips these variables.
- **--view-only** --- Forces all parameter to be read-only and translated to text (parameter attributes 'TEXT RO'). This is a fail safe mode, making it impossible for the SNMP user to change anything at the M&C system.
- **--comment** --- This option places a comment character in front of line of the file. This is valuable if you intend to export only a small number of parameters to the SNMP world. By default all parameters are disabled, you may enable the SNMP export of particular parameter by removing the comment character in front ot the parameter name.

For example, typing 'create-snmp.template --include-config --comment <ENTER>' in the telnet window creates a template file which also includes the device setup variables, but with all parameter definitions commented out.

3.4.1.4 The 'create-satnms.mib' Utility Program

Once the SNMP agent of the VLC / M&C has been configured and the server has been restarted, the agent is active and listens on UDP port 2161 for SNMP requests. At this point, calling the 'create-satnms.mib' utility program causes the server to write a MIB file which exactly matches the agent's internal MIB.

Using the the 'create-satnms.mib' utility program is as follows:

1. Open a telnet window to the M&C server
2. Log in as user 'satnms', password 'satnms'.
3. Type 'create-satnms.mib <ENTER>' to invoke the program

This creates the MIB file (file name 'satnms.mib') in the server's home directory. The file also appears in the 'public' share as a symbolic link, making it readable to other computers in the network.

3.4.2 TCP/SSL Remote Control

sat-nms M&C systems can be controlled remotely using a simple text based IP protocol. A remote application may use this protocol to query state information or device settings and to control the settings of each individual device. The M&C provides three flavours of this protocol on three different IP ports:

- Port 2015 accepts SSL connections giving monitor and control access to all parameters / devices in the M&C
- Port 2016 accepts plain TCP connections giving monitor and control access to all

parameters / devices in the M&C
- Port 2017 accepts plain TCP connections like port 2016, but grants monitoring access only.

All three ports are disabled by default. In order to use them, they have to be enabled in the vlc.properties file:

```
    server.rcontrol.ssl.read-write=yes
    server.rcontrol.tcp.read-write=yes
    server.rcontrol.tcp.read-only=yes
```

To make changes in the vlc.properties file effective, you must restart the M&C server. The protocol itself is very simple, there are only three commands:

```
? *parameter-id*
```

Queries the actual state of a parameter.*parameter-id* is a textual string as used in the configuration of logical devices, e.g. BCRX-1.power identifies the power reading of a SatService beacon receiver named BCRX-1. The parameter may be any device setting or a status reading. If there is a value known for the given parameter-id, the parameter-id followed by a space character and the actual value is replied, numeric values are formatted in the same way as for presenting them at the sat-nms GUI. If the parameter-id is not known or if the device driver could read a value from the device since start of the program, nothing is replied.

```
@ *parameter-id*
```

Subscribes for updates on every change of a parameter. If there is a value known for the given parameter-id, the parameter-id followed by a space character and the actual value is replied, numeric values are formatted in the same way as for presenting them at the sat-nms GUI. If the parameter-id is not known or if the device driver could read a value from the device since start of the program, nothing is replied. Every time the value of this parameter changes, the M&C will send a line with the parameter-id followed by a space character and the actual value of the parameter.

```
! *parameter-id new-value*
```

Sets a parameter to the given value. *new-value* is parsed as if entered to a field in the sat-nms GUI. There is no reply to this command. If you want to check if the new value has been accepted, use the '?' or '@' commands. At the monitoring only port 2017 set commands are silently ignored.

Remarks

- Subscriptions remain active until the connection is closed, there is no 'unsubscribe' command.
- If you are using subscriptions, it is always possible that you receive a change message for a changed parameter before you receive the expected reply to a subscribe or query

command. You always must be capable to receive messages in any sequence.

4 Installation

This section of the manual provides installation instruction for the different modules of the***sat-nms*** Monitoring & Control software.

> ***Remark:*** On a preconfigured system delivered by SatService these step as usually **not** required. So this manual is aimed especially at people who want to deploy the system in virtual machines or customer provided bare metal server or for deployments in private cloud environment. And it is intended for a better understanding of your installation.

Quick access links to the main sections of the document:

- Base Debian Linux setup
  - packages: satnms4-base, satnms4-utils
- MNC service
  - packages: satnms4-mnc, satnms4-images, satnms4-documentation, satnms4-jars
- EventDB service
  - packages: satnms4-mnc, satnms4-images, satnms4-documentation, satnms4-jars
- Backend service
  - packages: satnms4-backend
- WebClient and nginx
  - packages: satnms4-webclient, satnms4-images, satnms4-documentation
- Database SatDB
  - packages: satnms4-satdb

There are different way where to installed all these components. The simplest solution is to install all components on one host (bare metal server or virtual machine). On the other hand all components can be run on different hosts for several reasons like redundancy or load distribution.

Please refer to the chapter Architecture for the different options. TODO!!!

The following chapters consider all components as a stand-alone installation. Therefore some steps may be repeated and can be skipped in different scenarios. But the installation of all services (MNC, backend, EventDB, WebClient and SatDB) expect the base Debian Linux installation. So whatever you want to install, make sure you followed first the instruction in the chapter Base Debian Linux setup

4.1 Installation Base Debian Linux

This section provides installation instructions for the ***sat-nms*** base and utils packages for the ***sat-nms*** Monitoring & Control System.

These packages are used to add settings and tools to a standard minimal Debian 12 installation. The following chapter descripes the installation on Debian Linux 12 but should be similar for other Debian-based distributions.

4.1.1 Requirements

- Minimal Debian 12 Installation and console or SSH access
- headless installation is sufficient
- root permission in Linux either via `su` or `sudo`
  - all examples in this manual assume that you get root shell with the `su` command
  - for installation with no direct `root` login, you have to add `sudo` in front of all commands which requires `root` permission
- access to Debian 12 repositories either via Internet or from a local Debian archive mirror
  - check with `apt-get update` access to the configured repository is available.
- for base setup: `satnms4-base_[version].deb` package (in the following referred to as `satnms4-base`)
- for utilities setup: `satnms4-utils_[version].deb` package (in the following referred to as `satnms4-utils`)

4.1.2 For the impatient

This section provides a quick installation guide for the impatient and experienced Debian Linux users. Detailed information are in the following chapters.

Install Debian Linux 12 on your host with your favorite installation method.

We recommend the following settings:

- Language: `English`
- Disk: 20.0 GB, ext4 filesystem
- TaskSel: not a full installation, just select
  - SSH Server
  - Standard system utilities
- Time: `UTC`
- Default user: `satnms`

Install ***sat-nms*** base components

```
apt-get update
apt-get upgrade
apt install /home/public/satnms4-base_[version].deb
apt install /home/public/satnms4-utils_[version].deb
reboot
```

4.1.3 Installation

For the following instructions we assume that you are using Debian 12. Similar Linux distributions like Ubuntu may also work but are not covered by this manual.

Debian Linux 12

Please follow the standard Debian 12 instruction for a minimal Debian installation. We recommend the following setting for the Debian installer:

- Language: `English`
- locale: `en_GB.UTF-8`
- Network: fixed IP address
- Clock: use NTP
- Root user: set password and allow login
- Normal user: `satnms`
- Timezone: `UTC`
- Disk: 20GB partion for all with ext4
- TaskSel: only `SSH Server` and `Standard system utilities`

Additional packages

For convenience we recommend the following additional packages:

- curl
- debsums
- dialog
- htop
- mc
- rsync
- snmp
- sudo
- tcpdump
- tnftp
- vim

```
root@satnms7-dev:~# apt-get install curl debsums dialog htop mc rsync snmp sudo tcpdump
```

satnms4-base package

The `satnms4-base` package contains files and directories required by the other satnms4 components. It should be installed on each host which run ***sat-nms*** MNC components.

```
root@satnms7-dev:~# apt-get install ./satnms4-base_1.2.2-1_all.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'satnms4-base' instead of './satnms4-base_1.2.2-1_all.deb'
The following NEW packages will be installed:
  satnms4-base
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/462 kB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 /root/satnms4-base_1.2.2-1_all.deb satnms4-base all 1.2.2-1 [462 kB]
Selecting previously unselected package satnms4-base.
(Reading database ... 36810 files and directories currently installed.)
Preparing to unpack .../satnms4-base_1.2.2-1_all.deb ...
Unpacking satnms4-base (1.2.2-1) ...
Setting up satnms4-base (1.2.2-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/satnms.service → /etc/systemd/
Processing triggers for mailcap (3.70+nmu1) ...
```

Beside installing different files it will also setup the following:

- create a user `satnms` if not already exists
- create a directory `/home/public` which should be used later to place satnms4 packages and backups
- enable `sudo` for a list of commands
- add additional MOTD (message of the day) scripts to show information on login
- and some more

> ***Remark:*** All satnms4 packages must be installed from local source. All following instructions assume that you have these packages in the folder `/home/public` on the host where you want to install them.

If the user `satnms` does not exists before installing this package, a new user will be created. This user will be used to run the different ***sat-nms*** services. So it is required. By default it is not possible to login to this user, because there is no password set. Login as `root` and change the password for the user `satnms` :

```
root@satnms7-dev:~# passwd satnms
New password:
Retype new password:
passwd: password updated successfully
```

satnms4-utils package

The `satnms4-utils` package contains a collection of utilities for the***sat-nms*** Monitoring & Control System like backup and restore scripts.

```
root@satnms7-dev:/home/public# apt install ./satnms4-utils_1.0.6-1_all.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'satnms4-utils' instead of './satnms4-utils_1.0.6-1_all.deb'
The following NEW packages will be installed:
  satnms4-utils
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/30.7 kB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 /home/public/satnms4-utils_1.0.6-1_all.deb satnms4-utils all 1.0.6-1 [30.7 kB]
Selecting previously unselected package satnms4-utils.
(Reading database ... 36932 files and directories currently installed.)
Preparing to unpack .../satnms4-utils_1.0.6-1_all.deb ...
Unpacking satnms4-utils (1.0.6-1) ...
Setting up satnms4-utils (1.0.6-1) ...
Processing triggers for mailcap (3.70+nmu1) ...
```

Common Linux settings

In preinstalled systems delivered by SatService a dedicated file will display the version of the ***sat-nms*** Linux Operation System installation: `/etc/satnms_version` .

Its format is defined as: `[linux-distribution]-satnms-[version] [date]` . For customer build OS version it is required to create this file by your own.

- `linux-distribution` is usually `debian` , but other names are allowed like `ubuntu`
- the second word needs to be `-satnms-`
  - required because several tools parsing this file to get the OS version
  - the Sysinfo device uses this information to display the matching server information
- version format is `[major-version].[minor-version][suffix] [date]` e.g. `7.01`
  - major-number must be 7
  - minor-number will be increased if SatService creates new OS images
  - for customer builds minor numbers needs to be managed by customer
  - optional suffix can be added to mark special editions: e.g. `7.01idu` shows that it is a special OS image for the ***sat-nms*** ACU Indoor Unit
  - the date should be in format `YYYY-MM-DD`

```
satnms@satnms7-dev:~$ cat /etc/satnms_version
debian-satnms-7.01 2023-10-05
```

The follwing additional settings are recommended for convenience.

- `/home/satnms/.bashrc` : uncomment line `force_color_prompt=yes` to enable colored prompt
- `/root/.bashrc` : configure prompt colors, add the following line:

```
PS1='${debian_chroot:+($debian_chroot)}\[\033[01;31m\]\u@\[\033[01;33m\]\h\[\033[00m\]:
```

4.2 Installation MNC Service

This section provides installation instructions for the MNC service used by the***sat-nms*** Monitoring & Control System.

The MNC service runs device drivers and logical device and is the interface to the equipment to be monitored and controlled.

The following topics will be covered by the Administration Manual

- Update ***sat-nms*** MNC service TODO LINK !!!
- Backup and restore ***sat-nms*** MNC service settings TODO LINK !!!

4.2.1 Requirements

- Base Debian 12 Installation with installed packages
  - `satnms4-base`
  - `satnms4-utils`
- console or SSH access
- headless installation is sufficient
- root permission in Linux either via `su` or `sudo`
  - all examples in this manual assume that you get root shell with the `su` command
  - for installation with no direct `root` login, you have to add `sudo` in front of all commands which requires `root` permission
- for MNC installation the following packages:
  - `satnms4-documentation_[version].deb` package (in the following referred to as `satnms4-documentation`)
  - `satnms4-images_[version].deb` package (in the following referred to as `satnms4-images`)
  - `satnms4-jars_[version].deb` package (in the following referred to as `satnms4-jars`)
  - `satnms4-mnc_[version].deb` package (in the following referred to as `satnms4-mnc`)

4.2.2 For the impatient

This section provides a quick installation guide for the impatient and experienced Debian Linux users. Detailed information are in the following chapters.

```
apt-get update
apt-get upgrade
apt-get install openjdk-17-jre-headless
apt install /home/public/satnms4-documentation_[version].deb
apt install /home/public/satnms4-images_[version].deb
apt install /home/public/satnms4-jars_[version].deb
apt install /home/public/satnms4-mnc_[version].deb
systemctl enable satnms
systemctl start satnms
```

4.2.3 Installation

One external tools must be installed:

1. OpenJDK Java Runtime Environment (JRE)
2. JQ Pretty print of JSON data on CLI (optional)

Debian provided installation packages (`.deb`) for OpenJDK 17.

- login as `root`
- update all installed Debian packages: `apt-get update; apt-get upgrade`
- install JRE from Debian repository: `apt-get install openjdk-17-jre-headless`

example output

```
satnms@satnms7-dev:~$ su -
Password:
root@satnms7-dev:~# apt-get install openjdk-17-jre-headless
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ... ca-certificates-java java-common ...
Suggested packages:
  ...
The following NEW packages will be installed:
  ... ca-certificates-java java-common openjdk-17-jre-headless ...
0 upgraded, 21 newly installed, 0 to remove and 1 not upgraded.
Need to get 50.3 MB of archives.
After this operation, 210 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
... downloading and installing packages ...
```

To test if Java Runtime was installed successfully, run `java -version` as user `satnms` :

example output

```
satnms@satnms7-dev:/home/public$ java -version
openjdk version "17.0.14" 2025-01-21
OpenJDK Runtime Environment (build 17.0.14+7-Debian-1deb12u1)
OpenJDK 64-Bit Server VM (build 17.0.14+7-Debian-1deb12u1, mixed mode, sharing)
```

Now install the ***sat-nms*** backend software package:

- upload `satnms4-documentation` , `satnms4-images` , `satnms4-jars` and `satnms4-mnc` package to \`/home/public on the target system
- login as `root`
- install the local satnms4-* packages
  - `apt install /home/public/satnms4-documentation_[version].deb`
  - `apt install /home/public/satnms4-images_[version].deb`
  - `apt install /home/public/satnms4-jars_[version].deb`
  - `apt install /home/public/satnms4-mnc_[version].deb`

example output for package satnms4-mnc

```
root@satnms7-dev:/home/public# apt-get install ./satnms4-mnc_3.68.9-1_all.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'satnms4-mnc' instead of './satnms4-mnc_3.68.9-1_all.deb'
The following packages will be upgraded:
  satnms4-mnc
1 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/10.2 MB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 /home/public/satnms4-mnc_3.68.9-1_all.deb satnms4-mnc all 3.68.9-1 [10.2 MB]
Reading changelogs... Done
(Reading database ... 48696 files and directories currently installed.)
Preparing to unpack .../satnms4-mnc_3.68.9-1_all.deb ...
Unpacking satnms4-mnc (3.68.9-1) over (3.68.2-1) ...
Setting up satnms4-mnc (3.68.9-1) ...
```

On success you will find a multiple new files and directories `/home/satnms/` like `satnms.jar` and directories `drivers` , `protocols` etc.

4.2.4 Configuration

This section describes only the intital configuration step for the***sat-nms*** MNC service. Because the configuration of the M&C itself can be done via Web interface. Please refere to the following chapters:

- web client: Device-Setup
- web client: Screen-Editor

The system ships a default configuration which is "ready-to-run" but you can modify some configuration to adapt the system to your environment. For a all-in-one MNC System no additional configuration is required.

Several other options require your attention only for complex setups which includes server redundancy. A detailed description of all options from is available in the Administration Section of this Manual.

To ensure the MNC service will automatically start on reboot, login as root and enable this service.

- login as `root`
- enable autostart with `systemctl enable satnms"

Now you can start the backend service as user `satnms` and check if its running.

- login as `satnms`
- run the `satnms-watch`

```
┌──────────────────────┤ sat-nms state ├──────────────────────┐
│                                  │
│  hostname / MNC name : satnms7-dev / MNC                │
│  satnms OS version   : debian-satnms-7.01 2023-10-05     │
│  ─────────────────────────────────────────────────── │
│                                  │
│  MNC service         : STOPPED                   │
│  MNC service version : not installed               │
│                                  │
 ...
│                                  │
│  last updated        : 2025-14-02 16:48:48 UTC          │
│  ─────────────────────────────────────────────────── │
│                                  │
│       refresh      refresh service states         │
│       satnms-start start MNC Service              │
│       backend-stop stop Backend Service             │
│       nginx-start  start Nginx Webserver            │
│                                  │
│       <Select service>       <Exit>           │
└─────────────────────────────────────────────────────────────────┘
```

The output depends on the installed services. In the example above, only the***sat-nms*** MNC service is installed and running.

You can also control the MNC service directly with the follwing commands:

- start MNC service `sudo systemctl start satnms`
- stop MNC service `sudo systemctl stop satnms`

Check if the MNC service start successfully:

- status of MNC service `systemctl status satnms`

```
satnms@satnms7-dev:~/backend$ systemctl status backend
TODO
```

You can test if the MNC started successfully by checking the output of the log file `/home/satnms/debug.log` . The first line shows an early stage of start process and is an indication that the MNC service begins to start. The second line shows that all sub processes are now initialized and the MNC service is ready to serve client requests. Starting from this the backend service or clients are able to connect to the MNC service.

```
satnms@satnms7-dev:~$ tail /home/satnms/debug.log
...
000000 2025-03-05 12:32:18 2025-03-05 12:32:18   I SYSTEM Loading setup data (sat-nms MI
...
000000 2025-03-05 12:32:18 2025-03-05 12:32:18   I SYSTEM VLC 3.68.9 2025-03-05 07:22:0
...
```

Some more details and extended error messages if something failed are available in `/home/satnms/.panic.log`

A typical output of the default installation should looks like this:

```
satnms@satnms:~$ cat .panic.log
satnms version 3.68.9 2025-03-05 07:22:02
TRAPREC: File traprcv.json not found
TRAPREC: Cannot parse traprcv.json
TRAPREC: Trap receiver not started.
Universal SNMP agent listening at port 2261
TCP remote control server (r/w) listening at port 2016
TCP remote control server (r/o) listening at port 2017
SSL remote control server (r/w) listening at port 2015
```

A detailed description how to run the ***sat-nms*** MNC service is available in the administration manual.

Troubleshooting

If you have trouble to bring up the MNC service check first if the MNC service is still running with:

- `systemctl status satnms` or
- `/home/satnms/satnms-stat`

Additionally you can lookup the process ID (PID) of the backend service process:

```
satnms@satnms7-dev:~$  ps aux |grep satnms.jar
satnms   8204 ?   Sl   25:18 java -cp satnms.jar:/home/satnms/jars/* -mx512m -Djava.security.
```

Now check if the MNC process with this PID (here 8204 from the second column) is associated to a process listening on TCP ports from 2000 and above. The real number of listing ports depends on the actual configuration, but at least 2000-2007 should be visible.

```
satnms@satnms7-dev:~$ ss -tanp |grep 8204
LISTEN 0      50                 *:2000              *:*     users:(("java",pid=8204,fd=11))
LISTEN 0      50                 *:2001              *:*     users:(("java",pid=8204,fd=35))
LISTEN 0      50                 *:2002              *:*     users:(("java",pid=8204,fd=42))
LISTEN 0      50                 *:2003              *:*     users:(("java",pid=8204,fd=40))
LISTEN 0      50                 *:2004              *:*     users:(("java",pid=8204,fd=12))
LISTEN 0      50                 *:2005              *:*     users:(("java",pid=8204,fd=13))
LISTEN 0      50                 *:2006              *:*     users:(("java",pid=8204,fd=19))
LISTEN 0      50                 *:2007              *:*     users:(("java",pid=8204,fd=17))
...
```

!!!! The backend internal web server runs on Port 8443 and serves only HTTPS requests. ???? You should that this process is `LISTEN` on all interfaces (`*:8443`)

4.3 Installation Backend Service

This section provides installation instructions for the backend service used by the***sat-nms*** Monitoring & Control System.

The backend provides an unified REST-API to control all parts of an sat-nms Monitoring & Control System. It is used by the sat-nms WebClient but also available for third party applications. Please refere to the [API documentation](API Reference) for details.

The following topics will be covered by the Administration Manual

- Update ***sat-nms*** backend TODO LINK !!!
- Backup and restore ***sat-nms*** backend settings TODO LINK !!!

4.3.1 Requirements

- Base Debian 12 Installation with installed packages
    - `satnms4-base`
    - `satnms4-utils`
- console or SSH access
- headless installation is sufficient
- root permission in Linux either via `su` or `sudo`
    - all examples in this manual assume that you get root shell with the `su` command
    - for installation with no direct `root` login, you have to add `sudo` in front of all commands which requires `root` permission

- for backend installation: `satnms4-backend_[version].deb` package (in the following referred to as `satnms4-backend`)

4.3.2 For the impatient

This section provides a quick installation guide for the impatient and experienced Debian Linux users. Detailed information are in the following chapters.

```
apt-get update
apt-get upgrade
apt-get install openjdk-17-jre-headless
apt install /home/public/satnms4-backend_[version].deb
cp /home/satnms/backend/backend.properties.default /home/satnms/backend/backend.proper
systemctl enable backend
systemctl start backend
```

4.3.3 Installation

One external tools must be installed:

1. OpenJDK Java Runtime Environment (JRE)
2. JQ Pretty print of JSON data on CLI (optional)

Debian provided installation packages (`.deb`) for OpenJDK 17.

- login as `root`
- update all installed Debian packages: `apt-get update; apt-get upgrade`
- install JRE from Debian repository: `apt-get install openjdk-17-jre-headless`

example output

```
satnms@satnms7-dev:~$ su -
Password:
root@satnms7-dev:~# apt-get install openjdk-17-jre-headless
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ... ca-certificates-java java-common ...
Suggested packages:
  ...
The following NEW packages will be installed:
  ... ca-certificates-java java-common openjdk-17-jre-headless ...
0 upgraded, 21 newly installed, 0 to remove and 1 not upgraded.
Need to get 50.3 MB of archives.
After this operation, 210 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
... downloading and installing packages ...
```

To test if Java Runtime was installed successfully, run `java -version` as user `satnms` :

example output

```
satnms@satnms7-dev:/home/public$ java -version
openjdk version "17.0.14" 2025-01-21
OpenJDK Runtime Environment (build 17.0.14+7-Debian-1deb12u1)
OpenJDK 64-Bit Server VM (build 17.0.14+7-Debian-1deb12u1, mixed mode, sharing)
```

Optionally install JQ an lightweight and flexible command-line JSON processor for test the backend API.

- login as `root`
- update all installed Debian packages: `apt-get update; apt-get upgrade`
- install JRE from Debian repository: `apt-get install openjdk-17-jre-headless`

example output

```
root@satnms7-dev:/home/satnms# apt-get install jq
...
The following NEW packages will be installed:
  jq libjq1 libonig5
...
```

Now install the ***sat-nms*** backend software package:

- upload `satnms4-backend` package to `/home/public on the target system
- login as `root`
- install the local satnms4-backend package

`apt install /home/public/satnms4-backend_[version].deb` .

example output

```
root@satnms7-dev:/home/public# apt install ./satnms4-backend_1.17.10-1_all.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'satnms4-backend' instead of './satnms4-backend_1.17.10-1_all.deb'
The following NEW packages will be installed:
  satnms4-backend
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 0 B/24.7 MB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 /home/public/satnms4-backend_1.17.10-1_all.deb satnms4-backend all 1.17.10-1 [24.7
Selecting previously unselected package satnms4-backend.
(Reading database ... 38210 files and directories currently installed.)
Preparing to unpack .../satnms4-backend_1.17.10-1_all.deb ...
Unpacking satnms4-backend (1.17.10-1) ...
Setting up satnms4-backend (1.17.10-1) ...
```

On success you will find a new directory `/home/satnms/backend/` which contains the software and configuration files.

4.3.4 Configuration

A ***sat-nms*** backend service provides a gate to all ***sat-nms*** MNC servers in your overall system. This could be just a single MNC service running on the same host as the backend (all-in-one server) or any number of MNC service distributed over your network.

There are three steps to configure the ***sat-nms*** backend service:

1. create configuration file `/home/satnms/backend/backend.properties`
2. modify configuration file
3. enable autostart of backend service

By default the `satnms4-backend` package delivers a sample configuration for an all-in-one server installation. This template can be found in `/home/satnms/backend/backend.properties.default` . You can use this template by copy it to real backend configuration file: `/home/satnms/backend/backend.properties` and then editing it regarding you requirements.

To enable a minimal backend setup with a MNC service running on localhost just use the provide template:

```
satnms@satnms7-dev:~/backend$ cp backend.properties.default backend.properties
```

A detailed description of all options from `backend.properties` is available in the administration manual.

To ensure the backend service will automatically start on reboot, login as root and enable this service.

- login as `root`
- enable autostart with `systemctl enable backend"

Now you can start the backend service as user `satnms` and check if its running.

- login as `satnms`
- run the `satnms-watch`

```
┌──────────────────────┤ sat-nms state ├─────────────────────────┐
│                                    │
│   hostname / MNC name : satnms7-dev / MNC                │
│   satnms OS version   : debian-satnms-7.01 2023-10-05       │
│   ────────────────────────────────────────────────── │
│                                    │
 ...
│                                    │
│   backend service     : RUNNING                    │
│   backend version     : 1.17.10                   │

...
│                                    │
│   last updated       : 2025-14-02 16:48:48 UTC            │
│   ────────────────────────────────────────────────── │
│                                    │
│          refresh      refresh service states          │
│          satnms-start start MNC Service               │
│          backend-stop stop Backend Service              │
│          nginx-start  start Nginx Webserver             │
│                                    │
│          <Select service>         <Exit>          │
└────────────────────────────────────────────────────────────────┘
```

The output depends on the installed services. In the example above, only the***sat-nms*** backend service is installed and running.

You can also control the backend service directly with the follwing commands:

- start backend service `sudo systemctl start backend`
- stop backend service `sudo systemctl stop backend`

Check if the backend service start successfully:

- status of backend service `systemctl status backend`

```
satnms@satnms7-dev:~/backend$ systemctl status backend
● backend.service - sat-nms MNC Backend
     Loaded: loaded (/etc/systemd/system/backend.service; disabled; preset: enabled)
     Active: active (running) since Fri 2025-02-14 16:06:10 UTC; 5min ago
   Main PID: 18136 (java)
      Tasks: 22 (limit: 2307)
     Memory: 136.9M
        CPU: 8.788s
     CGroup: /system.slice/backend.service
             └─18136 java -Djava.security.egd=file:/dev/./urandom -jar backend.jar
```

You can test if the backend is ready to reply on REST-API requests use the heathcheck API-Endpoint which does not require authentication. It requires `jq` and `curl` installed. Run: `curl -k https://localhost:8443/healthcheck | jq` and if should report some information about the running backend software.

```
satnms@satnms7-dev:~$ curl -k -s  https://localhost:8443/healthcheck | jq
{
  "dict": [
    {
      "key": "PRODUCT",
      "val": "satnms4 backend"
    },
    {
      "key": "WEBSITE",
      "val": "www.satnms.com"
    },
    {
      "key": "VERSION",
      "val": "1.17.10 2024-12-17 12:22:54"
    },
    {
      "key": "SERIALNO",
      "val": "000001"
    }
  ]
}
```

Logfiles are available:

- `cat /home/satnms/backend/backend.log`
- `journalctl -u backend`

A detailed description how to run the ***sat-nms*** backend service is available in the administration manual.

Troubleshooting

If you have trouble to connect to the backend server's API endpoints check first if the backend service is still running with:

- `systemctl status backend` or
- `/home/satnms/backend-stat`

Additionally you can lookup the process ID (PID) of the backend service process:

```
satnms@satnms7-dev:~$ ps axu | grep backend.jar
satnms   18136  0.3  8.4 2606224 170268 ?   Ssl  16:06   0:25 java -Djava.security.egd=file:/d
```

Now check if the backend process with this PID (here 18136 from the second column) is associated to a process listening on the TCP port 8443:

```
satnms@satnms7-dev:~$ ss -taunp |grep 8443
tcp   LISTEN 0      100    *:8443     *:*    users:(("java",pid=18136,fd=10))
```

The backend internal web server runs on Port 8443 and serves only HTTPS requests. You should that this process is `LISTEN` on all interfaces (`*:8443`)

4.4 Installation SatDB database

This section provides installation instructions for the PostgreSQL database used by the***sat-nms*** Monitoring & Control System.

PostgreSQL is a free and open-source relational database management system. Installation packages are available for different operating systems and Linux distributions. The following chapter descripes the installation on Debian Linux 12 with the PostgreSQL version 15.

Details about PostgreSQL: https://www.postgresql.org/

If you have your own PostgreSQL installation or PostgreSQL cluster running on a different host or as a cloud service, you can skip the chapter Installation and move directly to chapter Configuration.

To update the database schema, we are using the Golang-Migrate tool. Details can be found here: https://github.com/golang-migrate/migrate.

Quick access links to the main sections of this installation guide:

- Database Installation
- Add database user
- Create SatDB database

The following topics will be covered by the Administration Manual

- Update SatDB database TODO LINK !!!
- Backup and restore SatDB database TODO LINK !!!

4.4.1 Requirements

- Minimal Debian 12 Installation and console or SSH access
- headless installation is sufficient
- root permission in Linux either via `su` or `sudo`
  - all examples in this manual assume that you get root shell with the `su` command
  - for installation with no direct `root` login, you have to add `sudo` in front of all commands which requires `root` permission
- access to Debian 12 repositories either via Internet or from a local Debian archive mirror
  - check with `apt-get update` access to the configured repository is available.
- for SatDB database setup: `satnms4-satdb_[version].deb` package (in the following referred to as `satnms4-satdb`)

4.4.2 For the impatient

This section provides a quick installation guide for the impatient and experienced Debian Linux users. Detailed information are in the following chapters.

```
apt-get update
apt-get upgrade
apt-get install postgresql
apt install /home/public/migrate.linux-amd64.deb
apt install /home/public/satnms4-satdb_[version].deb
```

4.4.3 Installation

Two external tools must be installed:

1. PostgreSQL
2. Golang-Migrate

Debian provided installation packages (`.deb`) for PostgreSQL.

- login as `root`
- update all installed Debian packages: `apt-get update; apt-get upgrade`
- install PostgreSQL from Debian repository: `apt-get install postgresql`

example output

```
satnms@satnms7-dev:~$ su -
Password:
root@satnms7-dev:~# apt-get update
Hit:1 http://security.debian.org/debian-security bookworm-security InRelease
Hit:2 http://deb.debian.org/debian bookworm InRelease
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease
Reading package lists... Done

root@satnms7-dev:~# apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

root@satnms7-dev:~# apt-get install postgresql
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ... postgresql-client-15 postgresql-client-common postgresql-common ...
The following NEW packages will be installed:
  ... postgresql-client-15 postgresql-client-common postgresql-common ...
0 upgraded, 17 newly installed, 0 to remove and 0 not upgraded.
Need to get 49.1 MB/49.1 MB of archives.
After this operation, 200 MB of additional disk space will be used.
Do you want to continue? [Y/n]
... downloading and installing packages ...
```

By default Linux user `postgres` has access to the database from `localhost` only without password.

To test if the PostgreSQL database was installed and started successfully, you can connect with command line tool `psql` (in short CLI):

- login as `root`
- switch to the PostgreSQL user: `su - postgres`
- run the PostgreSQL interactive terminal: `psql`
  - installed version will be show
  - an interactive prompt is available
- leave PostgreSQL terminal with `\q`
- switch back to user root: `exit`

example output

```
root@satnms7-dev:~# su - postgres
postgres@satnms7-dev:~$ psql
psql (15.10 (Debian 15.10-0+deb12u1))
Type "help" for help.

postgres=# \q
postgres@satnms7-dev:~$
```

To install Golang-migrate you need to manually download the Debian AMD64 package

- download debian package `migrate.linux-amd64.deb` from https://github.com/golang-migrate/migrate/releases/
- upload to target system in `/home/public`
- install the package with `apt-get install /home/public/migrate.linux-amd64.deb`

example output

```
root@satnms7-dev:~# apt-get install /home/public/migrate.linux-amd64.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'migrate' instead of '/home/public/migrate.linux-amd64.deb'
The following NEW packages will be installed:
  migrate
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/16.2 MB of archives.
After this operation, 53.2 MB of additional disk space will be used.
Get:1 /home/public/migrate.linux-amd64.deb migrate amd64 4.18.1 [16.2 MB]
Selecting previously unselected package migrate.
(Reading database ... 55809 files and directories currently installed.)
Preparing to unpack .../public/migrate.linux-amd64.deb ...
Unpacking migrate (4.18.1) ...
Setting up migrate (4.18.1) ...
```

You can test the successfull installation with `migrate --version` which should display the installed version number, e.g. 4.18.1

4.4.4 Configuration

If PostgreSQL is installed you need to setup an user and a database for the***sat-nms*** MNC system. These setups will be automatically performed if you install the `satnms4-satdb` package . Installation instructions for system where you are not able to install our Debian package, manual instructions are included in the following chapters.

The ***sat-nms*** MNC system will use the user `satnms` to connect to the database. This user will also be the owner of the SatDB. This prevents the application from requiring administrative rights for the entire database and the MNC system can use the same PostgreSQL database engine together with other software.

The ***sat-nms*** MNC system will use a database with the name `satnms`. If you are using you own PostgreSQL installation, make sure that this database name is not already used by any other application.

- upload the `satnms4-satdb` to /home/public on the target system running PostgreSQL.
- login as `root`
- install the local satnms4-satdb package `apt install /home/public/satnms4-satdb_[version].deb`. During installation you will be prompted for a password. The default password in preinstalled systems delivered by SatService is `satnms`. Please change it to secure your system.

example output

```
root@satnms7-dev:~# apt install /home/public/satnms4-satdb_0.2.0-1_all.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'satnms4-satdb' instead of '/home/public/satnms4-satdb_0.2.0-1_all.deb'
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'apt autoremove' to remove them.
The following NEW packages will be installed:
  satnms4-satdb
0 upgraded, 1 newly installed, 0 to remove and 2 not upgraded.
Need to get 0 B/5,304 B of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 /home/public/satnms4-satdb_0.2.0-1_all.deb satnms4-satdb all 0.2.0-1 [5,304 B]
Selecting previously unselected package satnms4-satdb.
(Reading database ... 159218 files and directories currently installed.)
Preparing to unpack .../satnms4-satdb_0.2.0-1_all.deb ...
checking user 'satnms' ...
checking postgresql ...
/var/run/postgresql:5432 - accepting connections
checking migrate ...
check database user 'satnms' ...
create PostgreSQL user 'satnms' ...
Enter password for new role:
Enter it again:
check database 'satnms' ...
Unpacking satnms4-satdb (0.2.0-1) ...
Setting up satnms4-satdb (0.2.0-1) ...
installing SatDB database schema ...
1/u init_satdb (26.36938ms)
2/u init_satdb (29.867137ms)
```

Remark: the default password in preinstalled systems delivered by SatService is `satnms`. Please change it to secure your system.

4.4.5 PostgreSQL network configuration

By default, access to the PostgreSQL database engine is restricted to the local system (the system where the database is installed) and no remote access via network is possible. If the ***sat-nms*** MNC system is a all-in-one installation this default configuration is sufficient and you can skip this chapter.

For a distributed MNC installation network access from a different host must be possible. Usually the database runs on the central server and the connected MNC servers needs to access the database via network.

- login as `root`
- edit the file `/etc/postgresql/15/main/pg_hba.conf`
    - if a different version of PostgreSQL is installed replace `15` with the version number
    - add a line at the end of the file with `host all all [local-net] md5`
        - where [local-net] is the network from where you want to connect e.g `192.168.2.0/24`
- edit file `/etc/postgresql/15/main/postgresql.conf`
    - uncomment and change line to: `listen_addresses = '*'`
- restart PostgreSQL: `systemctl restart postgresql`

example from pg_hba.conf

```
# allow access for two additional networks
host    all             all             172.17.1.0/16           md5
host    all             all             192.168.1.0/24           md5
```

example from postgresql.conf

```
# - Connection Settings -
listen_addresses = '*'    # what IP address(es) to listen on;
port = 5432               # (change requires restart)
```

You can test the network configuration with from any other system with an installed `psql` tool, or even on the local system if you specify the IP address of your PostgreSQL server:

- login as user `satnms`
- test connection: `psql --host [server-ip-address]`

example output

```
satnms@satnms7-dev:/etc/postgresql/15/main$ psql --host 192.168.2.231
Password for user satnms:
psql (15.10 (Debian 15.10-0+deb12u1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)
Type "help" for help.
satnms=>
```

4.4.5.1 Change users password

To avoid that the clear text password appears in the CLI history, you can use the `\password` command in `psql` .

- login as `root`
- switch to user the PostgreSQL user: `su - postgres`
- run `psql`
- enter command `\password satnms` which prompts you for a new password

example output

```
postgres@satnms7-dev:~$ psql
psql (15.10 (Debian 15.10-0+deb12u1))
Type "help" for help.

postgres=# \password satnms
Enter new password for user "satnms":
Enter it again:
postgres=# \q
```

4.4.6 Manual configuration

You can skip this section if you successfully installed the `satnms4-satdb` package.

4.4.6.1 Manual add database user

New users can be created via CLI or with SQL statements. This manual uses the CLI tool. There is no difference between database actions or CLI command, `createuser` is just a wrapper around the SQL command `CREATE ROLE` .

- Roles subsumes users and groups, so to be precises we are create a new role and not a user

- CLI https://www.postgresql.org/docs/current/app-createuser.html

- login as `root`

- switch to user the PostgreSQL user: `su - postgres`

- run `createuser --interactive --pwprompt` to create a new user

  - name: `satnms`
  - password: [set your own password here]
  - be superuser: `no`
  - allow to create databases: `no`
  - allow to create more new roles : `no`

example output

```
root@satnms7-dev:~# su - postgres
postgres@satnms7-dev:~$ pwd
/var/lib/postgresql
postgres@satnms7-dev:~$ createuser --interactive --pwprompt
Enter name of role to add: satnms
Enter password for new role:
Enter it again:
Shall the new role be a superuser? (y/n) n
Shall the new role be allowed to create databases? (y/n) n
Shall the new role be allowed to create more new roles? (y/n) n
```

4.4.6.2 Create SatDB database

New databases can be created via CLI or with SQL statements. This manual uses the CLI tool. There is no difference between database actions or CLI command, createdb is just a wrapper around CREATE DATABASE

- Database must be named `satnms` and the owner of this database must be the role `satnms` as created above
- before you try to to create the database, make sure there exist no database with the same name.
- CLI: https://www.postgresql.org/docs/current/app-createdb.html
- create database: `createdb satnms "sat-nms MNC database"`

example output

```
TODO
```

4.4.6.3 Load SatDB database schema

To initialize the newly create database with all the required object definitions (schema) the installed `migrate` tool will be used. This tool needs the schema definition files `*.sql`) which are installed by the `satnms4-satdb` package to `/home/satnms/satdb/` . Usually this package automatically runs the database migration steps, but if this fails you will find below how to run the step manually.

You can check the current database version with:

```
migrate -database 'postgres://satnms:satnms@localhost:5432/satnms?sslmode=disable' -path
```

- copy migration files to `/home/satnms/satdb/`
  - for each migration (aka database version) two files exists
    - `[number]_init_satdb.up.sql`
    - `[number]_init_satdb.dow.sql`
- run migration script to apply the changes.
  - `up` command migrates to the latests version
  - execute this as user `satnms`, its not necessary to switch to user `postgres` because the database and role `satnms` are already created in the previous steps.
  - `migrate -database 'postgres://satnms:satnms@localhost:5432/satnms?sslmode=dis`

Remark: If the database is already initialized it will only run the migration for the new version(s). E.g. if already database version 2 is installed and there are migration files for version 3 and 4 it will upgrade the database to version 4.

example output

```
satnms@satnms7-dev:~$ migrate -database 'postgres://satnms:satnms@localhost:5432/satnr
1/u init_satdb (81.381471ms)
2/u init_satdb (96.040331ms)
satnms@satnms7-dev:~$
```

4.4.6.4 Force database version

In case a migration step fails, migrate will not let you run other migrations on the same database and will mark the database as dirty. You have to fix the migration and then force the corresponding version.

- run `migrate -database 'postgres://satnms:satnms@localhost:5432/satnms?sslmode=disable'`
- replace [number] by the database version you want to force. e.g. `1`

4.4.6.5 Delete database user

If you created a user by mistake you can remove it. For details refer to the PostgreSQL manual:

- CLI: https://www.postgresql.org/docs/current/app-dropuser.html
- `dropuser` is a wrapper around `DROP ROLE`

example output

```
postgres@satnms7-dev:~$ dropuser --interactive satnms
Role "satnms" will be permanently removed.
Are you sure? (y/n) y
```

4.4.6.6 Remove database

If you created a database by mistake you can remove it. For details refer to the PostgreSQL manual:

- CLI https://www.postgresql.org/docs/current/app-dropdb.html
- `dropdb` is a wrapper around `DROP DATABASE`

5 API Reference

This document describes the sat-nms WebGUI backend server API which is used by the Web front end of the software and also may be used to control a sat-nms M&C in a programmatic way.

Quick access links to the main sections of the document:

- HTTPS Communication and OAuth2 Authentication
- API End Points
- WebSocket Communication
- Data Models
- Satellite Database

5.1 HTTPS Communication and OAuth2 Authentication

The sat-nms WebGUI backend server uses HTTPS for all communication. Unsecured HTTP calls are not accepted. In most cases a non-standard port number is used by the server, details about the port and also about the certificate provided by the server may vary for every installation of the software.

The backend server uses OAuth2 for authorization. It may be configured to accept JWTs issued by an external provider. In parallel the backend server provides a service for obtaining and renewing access tokens as well.

5.2 API End Points

This section describes the API end points provided by the sat-nms WebGUI backend server. The end point descriptions appear in an order which groups related functions. An alphabetic list of all end points is shown below.

- /api/v1/activebackend
- /api/v1/acutargets
- /api/v1/antennas
- /api/v1/applypreset
- /api/v1/backendinfo
- /api/v1/channels
- /api/v1/collectpreset
- /api/v1/dbstate
- /api/v1/dbversion
- /api/v1/debug
- /api/v1/devicesetup
- /api/v1/dframes
- /api/v1/documents
- /api/v1/drivers
- /api/v1/dscreens

- /api/v1/eventack
- /api/v1/eventreport
- /api/v1/filerecordings
- /api/v1/filerecordersettings
- /api/v1/globalSettings
- /api/v1/healthcheck
- /api/v1/inventory
- /api/v1/logout
- /api/v1/macros
- /api/v1/makewebsocket
- /api/v1/mnclist
- /api/v1/peek
- /api/v1/playmacro
- /api/v1/poke
- /api/v1/preferredprotocols
- /api/v1/presets
- /api/v1/presetvars
- /api/v1/devicevars
- /api/v1/protocols
- /api/v1/redundancy
- /api/v1/restart
- /api/v1/satdetails
- /api/v1/satellites
- /api/v1/satoperators
- /api/v1/schedule
- /api/v1/streamkeys
- /api/v1/subscribe
- /api/v1/tleimport
- /api/v1/thumbnail
- /api/v1/trackmem
- /api/v1/treeview
- /api/v1/uscreens
- /api/v1/user
- /api/v1/users
- /api/v1/userSettings
- /healthcheck
- /token

5.2.1 /token

The /token API call requests an access token from the server which can be used to authorize subsequent API calls.

All API calls except /token require a valid access token (JWT) to be passed with the header of the request as a "Authorization: Bearer ..." header or the server will abort with a 401 NOT AUTHENTICATED error.

Supported HTTP methods: POST

POST /token

Requests an access token for the user specified with the parameters added to this call. The parameter to this request must be passed as HTML form style parameters (application/x-www-form-urlencoded). The same end point is used to get a new and to renew an existing token.

Parameters:

Name	Value
username	The name of the user to log in. Must be a valid user name known to the sat-nms M&C application.
password	The password supplied by the user for login (clear text).
grant_type	One of 'password' or 'refresh_token'
refresh_token	A valid refresh token, making the backend to reply with a new version of the existing access token linked to this refresh token.

The *grant_type* parameter is mandatory, if it is set to 'password' the parameters*password* and *username* are requires as well. If *grant_type* is set to 'refresh_token', the parameter *refresh_token* is mandatory as well.

On success the API call returns a TokenReply document containing beside other information the access token and a refresh token to renew the access token.

Return Codes:

Code	Description
200	OK, login was successful. A TokenReply document is replied.
400	*grant_type* is neiter 'password' nor 'refresh_token'
401	Login failed. In this case a ApiError document is returned, describing details of the error. This happens either with *grant_type*='password' and invalid credentials or with *grant_type*='refresh_token' and an invalid refresh token.

5.2.2 /api/v1/logout

Logs out the currently logged in user.

Supported HTTP methods: POST

POST /api/v1/logout

Logs out the currently logged in user. Requires a valid access token to be passed with the request header to identify the session to be terminated. In fact, this call does not logout all open sessions of a given user, it terminates exactly one session which is referenced by the token.

Parameters: none

Return Codes:

Code	Description
204	OK, logout was successful. No payload data follows.

5.2.3 /healthcheck

This API call can be used to check if the backend server responds to API calls. It works without being logged in.

Supported HTTP methods: GET,HEAD,OPTIONS

GET /healthcheck

Returns an Dictionary document listing some basic status information about the backend server. As this API call is with public access, it returns only a limited set of information.

The Dictionary document returned contains the following keys with its values:

Key	Description
PRODUCT	The name of this product
WEBSITE	The SatService website URL
VERSION	The software version actually running
SERIALNO	The software serial number

Parameters: none

Return Codes:

Code	Description
200	OK, the Dictionary with the status information is returned as payload.

5.2.4 /api/v1/user

Delivers some information about the user currently logged in. Returns aUserProperties document describing the user. This document contains -- beside other information -- the *privilege level* of the user which must be known to decide which user interface elements must be shown in a read-only way because the user is not authorized to change these parameter (but he may read them if logged in).

This API end point also may be used with PUT to change the password of the actually logged in user.

Supported HTTP methods: GET, PUT, HEAD, OPTIONS

GET /api/v1/user

Returns a UserProperties document describing the currently logged in user.

Expected Payload: none

Replied Payload: a UserProperties document with information about the user who issued this call.

Parameters: none

Return Codes:

Code	Description
200	OK, returns a UserProperties document describing the currently logged in user. If no information about the currently logged in user is available, a dummy Document describing a user with the name 'unknown' and the privilege level '0' is returned.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

PUT /api/v1/user

Changes the password of the user issuing this call. Returns the (updated)UserProperties for the currently logged in user.

Expected Payload: a UserProperties document containing the new password for the actual user. All other fields of the the document are ignored.

Replied Payload: a UserProperties document with the updated user information.

Parameters: none

Return Codes:

Code	Description
200	OK, returns a UserProperties with the updated data for the actually logged in user.

Code	Description
401	not logged in. In this case a ApiError document is returned, describing details of the error.

5.2.5 /api/v1/users

Reads or writes the list of users known to the backend. When editing the user list, the list must be read with a GET and later sent back with a PUT with all fields in each UserProperties record.

Supported HTTP methods: GET, PUT, HEAD, OPTIONS

GET /api/v1/users

Gets a UserList document containing the list of all known users.

Expected Payload: none

Replied Payload: a UserList document containing the list of all known users.

Parameters: none

Return Codes:

Code	Description
200	OK, returns a UserList containing the list of all known users.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

PUT /api/v1/users

Send a UserList document containing the list of all known users to the backend. The UserList replaces the previous one. Already logged in users remain logged in, even if they are deleted from the list or if their password was changed.

Expected Payload: The UserList document to write.

Replied Payload: The UserList document echoed back.

Parameters: none

Return Codes:

Code	Description
200	OK, returns the written UserList echoed back.

Code	Description
400	cannot change the password. Probably the user logged in by LDAP. In this case a ApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

5.2.6 /api/v1/mnclist

Delivers a list containing some information about the M&C systems managed by the backend. Returns a MncList document containing this list.

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/v1/mnclist

Returns a MncList document containing the list of managed M&C systems and their properties.

Parameters: none

Return Codes:

Code	Description
200	OK, returns a MncList document containing the list of managed M&C systems.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

5.2.7 /api/v1/activebackend

Delivers an ActiveBackend document showing the name of the backend which is actually considered to be active for M&C redundancy switching. Also permits to change this using a PUT call.

Supported HTTP methods: GET, PUT, HEAD, OPTIONS

GET /api/v1/activebackend

Returns an ActiveBackend document showing the name of the backend which is actually considered to be active for M&C redundancy switching.

Parameters: none

Expected Payload: none.

Replied Payload: Returns an ActiveBackend document showing the name of the backend which is actually considered to be active for M&C redundancy switching.

Return Codes:

Code	Description
200	OK, returns the expected document
401	not logged in. In this case a ApiError document is returned, describing details of the error.

PUT /api/v1/activebackend

Sets the name of the of the backend which is actually considered to be active for M&C redundancy switching.

Parameters: none

Expected Payload: An ActiveBackend document containing the name of the backend which shall be considered to be active for M&C redundancy switching.

Replied Payload: Returns an ActiveBackend document showing the name of the backend which is actually considered to be active for M&C redundancy switching.

Return Codes:

Code	Description
200	OK, the name of the active backend has been updated. Returns the expected document
401	not logged in. In this case a ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

5.2.8 /api/v1/peek

Reads ('peeks') the Parameter or Range definition addressed by {messageId} from the backend's message cache. If no such message is present in the cache, the M&C server is asked to deliver the message. The API call waits up to 3 seconds for the M&C server to answer, then a 404 error is returned. Peeking parameters is a very inefficient way to get a parameter value. It only should be used if a parameter value is required only once (without subsequent updates) or for testing purposes. Section WebSocket Communication describes the subscription model using STOMP/websockets which is the preferred way to read/update parameters in the UI.

The {messageId} supplied to /api/v1/peek must be a complete message identifier consisting of M&C-, device- and parameter-ID.

The document returned on success is either a Parameter or a Range, depending on the {messageId} supplied. For details about messages and message identifiers, see section

Message later in this document.

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/v1/peek/{messageId}

Returns a Message document describing the requested parameter.

Parameters:

Parameter	Description
messageId	The requested messageId. Must be a complete message identifier consisting of M&C-, device- and parameter-ID.

Return Codes:

Code	Description
200	OK, returns a Message document describing the requested parameter.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no parameter with the given messageId exists. In this case aApiError document is returned, describing details of the error.

5.2.9 /api/v1/poke

Writes ('pokes') a Parameter or Range message to the M&C server. The message to be written must be supplied as POST data.

Supported HTTP methods: POST, HEAD, OPTIONS

POST /api/v1/poke

Expects exactly one Parameter document as POST data. This creates and distributes the message on the M&C server addressed by the M&C identifier contained in the supplied message. This does not check if the specified destination exists. Messages sent to non-existing destinations are silently ignored, also result in a 201 response.

Parameters: none

Return Codes:

Code	Description
201	Success, echoes the message created.

Code	Description
401	not logged in. In this case a ApiError document is returned, describing details of the error.

5.2.10 /api/v1/makewebsocket

Creates a new websocket to connect to. A unique ID is generated for this websocket and returned as a WebsocketId document in the call's response body.

Supported HTTP methods: POST, HEAD, OPTIONS

POST /api/v1/makewebsocket

Creates a new websocket to connect to. A unique ID is generated for this websocket and returned as a WebsocketId document in the call's response body.

The WebsocketId is required to open the socket itself and for all subscribe / unsubscribe operations on this websocket. See section Websocket Communication for details about how to open a websocket.

Parameters: none

Return Codes:

Code	Description
201	Success, delivers the ID of the created websocket in the response body (data model WebsocketId).
401	not logged in. In this case a ApiError document is returned, describing details of the error.

5.2.11 /api/v1/subscribe

Subscribes or unsubscribes the delivery of one or more message IDs over the websocket connection. Whether to subscribe or unsubscribe is defined by the HTTP method.

Supported HTTP methods: POST, DELETE, HEAD, OPTIONS

POST /api/v1/subscribe/{websocketId}/{messageId}

Subscribes for the reception of messages with the given messageId over the websocket identified by websocketId. The requested message and all subsequent updates are sent over the websocket connection.

Returns a SubscribeList with the response body containing the message IDs which have been successfully subscribed (in this case of a single subscription the list contains either one or no message IDs).

The API call does not check if the remote M&C system can deliver a message with the given ID. Messages may become available later, this is a common situation.

Parameters:

Parameter	Description
websocketID	Identifies the websocket to send the message and subsequent updates to. See section Websocket Communication how to obtain a websocketID and how to open a websocket.
messageId	Identifies the message / parameter to subscribe. Must be a complete sat-nms message identifier including the M&C ID.

Return Codes:

Code	Description
201	Success, a SubscribeList is returned with the response body containing the (single) subscribed message ID.
401	Not logged in. In this case a ApiError document is returned, describing details of the error.
403	Invalid websocketId. The websocketId referenced in the URL either does not exist or the session which sent the request does not own this websocket. An empty SubscribeList is returned with the response body.

POST /api/v1/subscribe/{websocketId}

Subscribes for the reception of messages of multiple message IDs over a websocket. This variant of the API call expects a SubscribeList data object in the request body. This data object contains the message IDs to subscribe for.

The subscription requests are forwarded and are checked if successful one by one. The API call returns a SubscribeList in the response body containing the message IDs which have been successfully subscribed.

The API call does not check if the remote M&C system can deliver messages with the given IDs. Messages may become available later, this is a common situation.

Parameters:

Parameter	Description
websocketID	Identifies the websocket to send the message and subsequent updates to. See section Websocket Communication how to obtain a websocketID and how to open a websocket.

Return Codes:

Code	Description
201	Success, a SubscribeList is returned with the response body containing the subscribed message IDs.
401	Not logged in. In this case a ApiError document is returned, describing details of the error.
403	Invalid websocketId. The websocketId referenced in the URL either does not exist or the session which sent the request does not own this websocket. An empty SubscribeList is returned with the response body.

DELETE /api/v1/subscribe/{websocketId}/{messageId}

Unsubscribes the reception of messages with the given messageId over the websocket identified by websocketId.

Returns a SubscribeList with the response body containing the message IDs which have been successfully unsubscribed (in this case the list contains either one or no message IDs).

The API call does not check if there is a subscription pending for the given parameters. If a subscription is canceled which never has been done or which has been unsubscribed before, this is silently ignored.

Parameters:

Parameter	Description
websocketID	Identifies the websocket to unsubscribe. See section Websocket Communication how to obtain a websocketID and how to open a websocket.
messageId	Identifies the message / parameter to unsubscribe. Must be a complete sat-nms message identifier including the M&C ID.

Return Codes:

Code	Description
200	Success, a SubscribeList is returned with the response body containing the (single) unsubscribed message ID.
401	Not logged in. In this case a ApiError document is returned, describing details of the error.

Code	Description
403	Invalid websocketId. The websocketId referenced in the URL either does not exist or the session which sent the request does not own this websocket. An empty SubscribeList is returned with the response body.

DELETE /api/v1/subscribe/{websocketId}

Unsubscribes the reception of multiple message IDs over a websocket. This variant of the API call expects a SubscribeList data object in the request body. This data object contains the message IDs to unsubscribe.

The subscription requests are forwarded and are checked if successful one by one. The API call returns a SubscribeList in the response body containing the message IDs which have been successfully subscribed.

The API call does not check if there are subscriptions pending for the given message IDs. If a subscription is canceled which never has been done or which has been unsubscribed before, this is silently ignored.

Parameters:

Parameter	Description
websocketID	Identifies the websocket to unsubscribe. See section Websocket Communication how to obtain a websocketID and how to open a websocket.

Return Codes:

Code	Description
201	Success, a SubscribeList is returned with the response body containing the unsubscribed message IDs.
401	Not logged in. In this case a ApiError document is returned, describing details of the error.
403	Invalid websocketId. The websocketId referenced in the URL either does not exist or the session which sent the request does not own this websocket. An empty SubscribeList is returned with the response body.

5.2.12 /api/v1/uscreens

Gives access to the user defined screen definitions stored on the primary M&C. Screen definitions may be listed, read or written with this call.

Supported HTTP methods: GET, POST, HEAD, OPTIONS

GET /api/v1/uscreens

Returns a list of all available uscreens.

Parameters: none.

Expected Payload: none.

Replied Payload: Returns an ItemList containing the names of all available user screens. The list is read from the primary M&C.

Return Codes:

Code	Description
200	OK, returns an ItemList containing the names of all available user screens.
400	the primary M&C is offline, the list cannot be read. In this case aApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

GET /api/v1/uscreens/{name}

Returns a ScreenDefinition document describing the requested screen. The screen definition is read from the primary M&C.

Parameters:

Parameter	Description
name	The name of the screen to load. Please note, that screen names are case sensitive, the name must match exactly to be loaded.

Expected Payload: none.

Replied Payload: Returns a ScreenDefinition document describing the requested screen.

Return Codes:

Code	Description
200	OK, returns a ScreenDefinition document describing the requested screen.
400	the primary M&C is offline, the screen cannot be read. In this case aApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

Code	Description
404	no screen definition with the given name exists. In this case aApiError document is returned, describing details of the error.

POST /api/v1/uscreens/{name}

Stores a ScreenDefinition at the primary M&C. If the screen already exists, it gets overwritten.

Parameters:

Parameter	Description
name	The name of the screen to store. Screen definition names may consist of the characters A-Z, a-z, 0-9 and the hyphen character.

Expected Payload: The ScreenDefinition to store.

Replied Payload: Echoes back the written ScreenDefinition.

Return Codes:

Code	Description
200	OK, echoes back the written ScreenDefinition.
400	the primary M&C is offline, the screen cannot be written. In this case aApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

DELETE /api/v1/uscreens/{name}

Deletes a ScreenDefinition at the primary M&C.

Parameters:

Parameter	Description
name	The name of the screen to delete. Screen definition names may consist of the characters A-Z, a-z, 0-9 and the hyphen character.

Expected Payload: none.

Replied Payload: none.

Return Codes:

Code	Description
204	OK, the screen has been deleted.
400	the primary M&C is offline, the screen cannot be deleted. In this case aApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	not found. No screen of this name exists. In this case aApiError document is returned, describing details of the error.

5.2.13 /api/v1/dscreens

Gives access to the device screen definitions stored on the primary M&C. Screen definitions may be listed, read or written with this call.

Supported HTTP methods: GET, POST, HEAD, OPTIONS

GET /api/v1/dscreens

Returns a list of all available dscreens. The list is read from the primary M&C.

Parameters: none.

Expected Payload: none.

Replied Payload: Returns an ItemList containing the names of all available device screens.

Return Codes:

Code	Description
200	OK, returns an ItemList containing the names of all available device screens.
400	the primary M&C is offline, the list cannot be read. In this case aApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

GET /api/v1/dscreens/{name}

Returns a ScreenDefinition document describing the requested screen. The screen is read from the primary M&C.

Parameters:

Parameter	Description

Parameter	Description
name	The name of the screen to load. Please note, that screen names are case sensitive, the name must match exactly to be loaded.

Expected Payload: none.

Replied Payload: Returns a ScreenDefinition document describing the requested screen.

Return Codes:

Code	Description
200	OK, returns a ScreenDefinition document describing the requested screen.
400	the primary M&C is offline, the screen cannot be read. In this case aApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no screen definition with the given name exists. In this case aApiError document is returned, describing details of the error.

POST /api/v1/dscreens/{name}

Stores a ScreenDefinition at the primary M&C. If the screen already exists, it gets overwritten.

Parameters:

Parameter	Description
name	The name of the screen to store. Screen definition names may consist of the characters A-Z, a-z, 0-9 and the hyphen character.

Expected Payload: The ScreenDefinition to store.

Replied Payload: Echoes back the written ScreenDefinition.

Return Codes:

Code	Description
200	OK, echoes back the written ScreenDefinition.
400	the primary M&C is offline, the screen cannot be written. In this case aApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

DELETE /api/v1/dscreens/{name}

Deletes a ScreenDefinition at the primary M&C.

Parameters:

Parameter	Description
name	The name of the screen to delete. Screen definition names may consist of the characters A-Z, a-z, 0-9 and the hyphen character.

Expected Payload: none.

Replied Payload: none.

Return Codes:

Code	Description
204	OK, the screen has been deleted.
400	the primary M&C is offline, the screen cannot be deleted. In this case aApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	not found. No screen of this name exists. In this case aApiError document is returned, describing details of the error.

5.2.14 /api/v1/dframes

Gives access to the device frame definitions stored on the primary M&C. Device frame definitions may be listed, read or written with this call.

Supported HTTP methods: GET, POST, HEAD, OPTIONS

GET /api/v1/dframes

Returns a list of all available device frame definitions. The list is read from the primary M&C.

Parameters: none.

Expected Payload: none.

Replied Payload: Returns an ItemList containing the names of all available device frame definitions.

Return Codes:

Code	Description
200	OK, returns an ItemList containing the names of all available device frame definitions.
400	the primary M&C is offline, the list cannot be read. In this case aApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

GET /api/v1/dframes/{name}

Returns a DeviceFrameDefinition document describing the requested frame. The frame definition is read from the primary M&C.

Parameters:

Parameter	Description
name	The name of the frame definition to load. Please note, that frame names are case sensitive, the name must match exactly to be loaded.

Expected Payload: none.

Replied Payload: Returns a DeviceFrameDefinition document describing the requested frame.

Return Codes:

Code	Description
200	OK, returns a DeviceFrameDefinition document describing the requested frame.
400	the primary M&C is offline, the frame definition cannot be read. In this case a ApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no frame definition with the given name exists. In this case aApiError document is returned, describing details of the error.

POST /api/v1/dframes/{name}

Stores a DeviceFrameDefinition at the primary M&C. If the frame definition already exists, it gets overwritten.

Parameters:

Parameter	Description
name	The name of the frame to store. DeviceFrame definition names may consist of the characters A-Z, a-z, 0-9 and the hyphen character.

Expected Payload: The DeviceFrameDefinition to store.

Replied Payload: Echoes back the written DeviceFrameDefinition.

Return Codes:

Code	Description
200	OK, echoes back the written DeviceFrameDefinition.
400	the primary M&C is offline, the frame definition cannot be written. In this case a ApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

DELETE /api/v1/dframe/{name}

Deletes a DeviceFrameDefinition at the primary M&C.

Parameters:

Parameter	Description
name	The name of the frame definition do delete. DeviceFrame definition names may consist of the characters A-Z, a-z, 0-9 and the hyphen character.

Expected Payload: none.

Replied Payload: none.

Return Codes:

Code	Description
204	OK, the device frame has been deleted.
400	the primary M&C is offline, the frame definition cannot be deleted. In this case a ApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

Code	Description
404	not found. No device frame of this name exists. In this case aApiError document is returned, describing details of the error.

5.2.15 /api/v1/macros

Permits to load and save sat-nms macro definitions as well as to read the list of macros defined on a given M&C.

Supported HTTP methods: GET, PUT, DELETE, HEAD, OPTIONS

GET /api/v1/macros/{mncName}

Returns an ItemList document containing the list of macros defined on the given M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C from which the macro list shall be loaded.

Return Codes:

Code	Description
200	OK, returns an ItemList containing the requested macro list.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no M&C with the given name exists. In this case a ApiError document is returned, describing details of the error.

GET /api/v1/macros/{mncName}/{macroName}

Returns a Macro document containing the requested macro.

Parameters:

Parameter	Description
mncName	The name of the M&C from which the macro shall be loaded.
macroName	The name of the macro to load.

Return Codes:

Code	Description
200	OK, returns a Macro containing the requested macro.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	either M&C with the given name exists or at the known M&C there exists no macro with the requested name. In this case a ApiError document is returned, describing details of the error.

PUT /api/v1/macros/{mncName}/{macroName}

Saves a macro on the given M&C. Expects Macro document containing the macro text to save as request body. The *name* field in the Macro document will be ignored, the macro will be saved under the name given in the URL. As every PUT request, this will either overwrite an existing macro with this name or create a new macro definition at the M&C if no macro with this name exists.

Parameters:

Parameter	Description
mncName	The name of the M&C on which the macro shall be saved.
macroName	The name of the macro to save.

Return Codes:

Code	Description
200	OK, returns the Macro containing the saved macro. This will have the *name* field set the name under which the macro actually has been saved.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no M&C with the given name exists. In this case a ApiError document is returned, describing details of the error.

DELETE /api/v1/macros/{mncName}/{macroName}

Deletes a macro at the given M&C.

Parameters:

Parameter	Description

Parameter	Description
mncName	The name of the M&C on which the macro shall be deleted.
macroName	The name of the macro to delete.

Return Codes:

Code	Description
200	OK, returns the Macro containing the deleted macro.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	either M&C with the given name exists or at the known M&C there exists no macro with the requested name. In this case a ApiError document is returned, describing details of the error.

5.2.16 /api/v1/playmacro

Executes a macro on a given M&C.

Supported HTTP methods: POST, HEAD, OPTIONS

POST /api/v1/playmacro/{mncName}

Executes the referenced macro. Expects the macro to play as aMacro body, delegates the request to the M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C at which the macro shall be loaded and executed.

Return Codes:

Code	Description
200	OK, returns the Macro containing the executed macro.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	the M&C with the given name does nit exist. In this case aApiError document is returned, describing details of the error.

POST /api/v1/playmacro/{mncName}/{macroName}

Executes the referenced macro. Expects no request body, simply delegates the request to the M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C at which the macro shall be loaded and executed.
macroName	The name of the macro to execute.

Return Codes:

Code	Description
200	OK, returns the Macro containing the executed macro.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	either M&C with the given name exists or at the known M&C there exists no macro with the requested name. In this case a ApiError document is returned, describing details of the error.

5.2.17 /api/v1/presets

Permits to load and save sat-nms device preset definitions as well as to read the list of presets defined on a given M&C.

Supported HTTP methods: GET, PUT, HEAD, OPTIONS

GET /api/v1/presets/{mncName}

Returns an ItemList document containing the list of preset directories found on the given M&C. This is the list of device drivers which have at least one preset saved on this M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C from which the preset directory list shall be loaded.

Return Codes:

Code	Description

Code	Description
200	OK, returns an ItemList containing the requested preset directory list.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no M&C with the given name exists. In this case a ApiError document is returned, describing details of the error.

If the given M&C name exists, but there are no presets stored at all, the request returns 200 OK and an empty list.

GET /api/v1/presets/{mncName}/{driverName}

Returns an ItemList document containing the list of presets defined this device driver on the given M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C from which the preset list shall be loaded.
driverName	The name of the device driver for which the preset list shall be loaded.

Return Codes:

Code	Description
200	OK, returns an ItemList containing the requested preset list.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no M&C with the given name exists. In this case a ApiError document is returned, describing details of the error.

If the given M&C name exists, but there are no presets stored for the driver type, the request returns 200 OK and an empty list.

GET /api/v1/presets/{mncName}/{driverName}/{presetName}

Returns a DevicePreset document containing the requested preset.

Parameters:

Parameter	Description

Parameter	Description
mncName	The name of the M&C from which the preset shall be loaded.
driverName	The name of the device driver for which the preset shall be loaded.
presetName	The name of the preset to load.

Return Codes:

Code	Description
200	OK, returns a DevicePreset containing the requested preset.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	either M&C with the given name exists or at the known M&C there exists no preset with the requested driver/name combination. In this case a ApiError document is returned, describing details of the error.

PUT /api/v1/presets/{mncName}/{driverName}/{presetName}

Saves a preset on the given M&C. Expects DevicePreset document containing the preset text to save as request body. The *name* and *driver* fields in the DevicePreset document will be ignored, the preset will be saved under the driver / name given in the URL. As every PUT request, this will either overwrite an existing preset with this name or create a new preset definition at the M&C if no preset with this name exists.

Parameters:

Parameter	Description
mncName	The name of the M&C on which the preset shall be saved.
driverName	The name of the device driver for which the preset shall be saved.
presetName	The name of the preset to save.

Return Codes:

Code	Description
200	OK, returns the DevicePreset containing the saved preset. This will have the *name* and *driver* fields set the values under which the preset actually has been saved.

Code	Description
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no M&C with the given name exists. In this case a ApiError document is returned, describing details of the error.

DELETE /api/v1/presets/{mncName}/{driverName}/{presetName}

Deletes a preset from the given M&C. Expects no request body.

Parameters:

Parameter	Description
mncName	The name of the M&C from which the preset shall be deleted.
driverName	The name of the device driver for which the preset shall be deleted.
presetName	The name of the preset to delete.

Return Codes:

Code	Description
200	OK, returns the DevicePreset containing the deleted preset.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no M&C with the given name exists or no device preset with that name / device driver could be found. In this case a ApiError document is returned, describing details of the error.

5.2.18 /api/v1/applypreset

Applies a preset on a given M&C.

Supported HTTP methods: POST, HEAD, OPTIONS

POST /api/v1/applypreset/{mncName}/{deviceName}

Applies a preset to the given device. Expects a DevicePreset as request body, then delegates the request to the M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C at which the preset shall be applied.
deviceName	The name of the device to apply the preset to.

Return Codes:

Code	Description
200	OK, returns the DevicePreset containing the applied preset.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no M&C with the given name exists. In this case a ApiError document is returned, describing details of the error.

POST /api/v1/applypreset/{mncName}/{driverName}/{deviceName}/{presetName}

Applies the referenced preset to the given device. Expects no request body, simply delegates the request to the M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C at which the preset shall be applied.
driverName	The name of the device driver for which the preset shall be applied.
deviceName	The name of the device to apply the preset to.
presetName	The name of the preset to apply.

Return Codes:

Code	Description
200	OK, returns the DevicePreset containing the applied preset.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	either M&C with the given name exists or at the known M&C there exists no driver/device/preset with the requested name. In this case a ApiError document is returned, describing details of the error.

5.2.19 /api/v1/collectpreset

Collects a preset on a given M&C. This tells the referenced device to collect its settings into a preset and to store this preset in the M&C with the given name.

Supported HTTP methods: POST, HEAD, OPTIONS

POST /api/v1/collectpreset/{mncName}/{driverName}/{deviceName}/{presetName}

Collects a preset on a given M&C. This tells the referenced device to collect its settings into a preset and to store this preset in the M&C with the given name. Expects no request body, simply delegates the request to the M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C at which the preset shall be collected.
driverName	The name of the device driver for which the preset shall be collected.
deviceName	The name of the device to collect the preset from.
presetName	The name of the preset to store.

Return Codes:

Code	Description
200	OK, returns the DevicePreset containing the collected preset.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	either M&C with the given name exists or at the known M&C there exists no device with the requested name. In this case a ApiError document is returned, describing details of the error.

5.2.20 /api/v1/presetvars

Delivers the list of driver variable which may be stored in a preset for a given device driver.

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/vi/presetvars/{mncName}/{driverName}

Returns a PresetVars document containing the requested list of preset-storable variables. The backend reads the device driver *driverName.device* from the given M&C, compiles the driver text and selects the preset storable variables from the compiled list. The returned PresetVars document contains the Range definitions for all variables of the driver which may appear in a device preset. The *messageId* field in each Range object is set to the name of the variable.

Parameters:

Parameter	Description
mncName	The name of the M&C from which the variable list shall be loaded.
driverName	The name of the device driver for which the variable list shall be evaluated.

Return Codes:

Code	Description
200	OK, returns a PresetVars document containing the requested variable list.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	either no M&C with the given name exists, or at the known M&C there exists no device driver with the given name. In this case a ApiError document is returned, describing details of the error.

5.2.21 /api/v1/devicevars

Delivers the list of all variables a device defines

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/vi/devicevars/{mncName}/{deviceName}

Returns a DeviceVars document containing the requested list of variables. The backend evaluates the driver for the given device, compiled the device driver and gets the variable list from this. The returned DeviceVars document contains the Range definitions for all variables of the driver which may appear in a device preset. The *messageId* field in each Range object is set to the name of the variable.

The API call returns the variable list as it exists at compile time at system startup. Changes made dynamically to the list of variables or to some variable ranges are not contained in the list

Parameters:

Parameter	Description
mncName	The name of the M&C from which the variable list shall be loaded.

Parameter	Description
deviceName	The name of the device for which the variable list shall be evaluated.

Return Codes:

Code	Description
200	OK, returns a PresetVars document containing the requested variable list.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	either no M&C with the given name exists, or at the known M&C there exists no device driver with the given name. In this case a ApiError document is returned, describing details of the error.

5.2.22 /api/v1/schedule

Permits to load and save the macro scheduler's schedule at a given M&C.

When subscribing for a '.SYSTEM.schedule' parameter of a M&C, the front end receives a message which indicates 'an updated schedule is available' instead of sending the bulky schedule itself over the websocket connection.

After the front end received this message, it can use the /api/v1/schedule API call to retrieve the schedule as a Schedule data object.

Supported HTTP methods: GET, PUT, HEAD, OPTIONS

GET /api/v1/schedule/{mncName}

Returns a Schedule document containing the actual system schedule from this M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C from which the schedule shall be loaded.

Return Codes:

Code	Description
200	OK, returns a Schedule document containing the requested schedule.

Code	Description
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no M&C with the given name exists. In this case a ApiError document is returned, describing details of the error.

PUT /api/v1/schedule/{mncName}

Overwrites the system schedule at the given M&C. Expects a Schedule document containing new schedule to set.

Parameters:

Parameter	Description
mncName	The name of the M&C on which the preset shall be saved.

Return Codes:

Code	Description
200	OK, returns the Schedule containing the saved schedule.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no M&C with the given name exists. In this case a ApiError document is returned, describing details of the error.

5.2.23 /api/v1/healthcheck

This API call delivers some status information about the running backend server. Unlike the version with public access, this version of the health check contains the complete set of information.

Supported HTTP methods: GET,HEAD,OPTIONS

GET /api/v1/healthcheck

Returns an Dictionary document listing some status information about the backend server.

The Dictionary document returned contains the following keys with its values:

Key	Description
PRODUCT	The name of this product

Key	Description
WEBSITE	The SatService website URL
VERSION	The software version actually running
SERIALNO	The software serial number
UPTIME	The server uptime
MEMORY	The memory usage of the backend
SESSIONS	The number of open sessions
WEBSOCKS	The number of open (message) websocket connections

Parameters: none

Return Codes:

Code	Description
200	OK, the Dictionary with the status information is returned as payload.

5.2.24 /api/v1/dbstate

This API call delivers some status information about the backend's connection to the database server. It may be used by a frontend to evaluate if the backend actually is connected to the (read-only) backeup database and therefore no changes to the database content are allowed

Supported HTTP methods: GET,HEAD,OPTIONS

GET /api/v1/dbstate

Returns a SatDbState document listing some status information about the database connection.

Parameters: none

Return Codes:

Code	Description
200	OK, the SatDbState document with the status information is returned as payload.

5.2.25 /api/v1/dbversion

This API call delivers the version information about the database connected to the backend.

Supported HTTP methods: GET,HEAD,OPTIONS

GET /api/v1/dbversion

Returns a SatDbVersion document showing the actual version of the database and the version required by the backend.

Parameters: none

Return Codes:

Code	Description
200	OK, the SatDbState document with the status information is returned as payload.

5.2.26 /api/v1/satellites

This API endpoint gives access to various functions of the sat-nms satellite database, permitting to read, add, change or delete entries at the database

Supported HTTP methods: GET, POST, PATCH, DELETE, HEAD, OPTIONS**GET /api/v1/satellites**

Returns the list of stored satellites as an array ofSatDbSatellite data objects.

Parameters: none

Expected Payload: none

Replied Payload: The list of stored satellites as an array ofSatDbSatellite data objects.

Return Codes:

Code	Description
200	OK, returns a JSON document with all satellites and their attributes.
401	not logged in. In this case anApiError document is returned,describing details of the error.

GET /api/v1/satellites/{satelliteId}

Returns the basic properties of the satellite addressed by {satelliteId} as aSatDbSatellite data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: none

Replied Payload: The requested SatDbSatellite data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exists. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/satellites

Creates a new satellite with the given attributes . The unique *satelliteId* is generated by the database backend and returned in the response document.

Parameters: none

Expected Payload: The SatDbSatellite data object to store. Any *satelliteId* definition in the source is ignored as this will be set by the database when the new entry is created. If other keys of the data object are missing, default values will be assigned these keys.

Replied Payload: The SatDbSatellite data object added to the database. The replied data object contains the *satelliteId* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created SatDbSatellite data object is replied as payload
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/satellites/{satelliteId}

Updates an existing satellite with the given *satelliteId* from the attached SatDbSatellite data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: The SatDbSatellite data object containing the fields to be updated in the database. If the data object contains a *satelliteId* key, the value is ignored. The database record is solely identified by *satelliteId* set in the URL, the *satelliteId* of the record is immutable. Keys not contained in the supplied data object are retained unchanged.

Replied Payload: The SatDbSatellite data object read from the database after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exists. In this case anApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}

Remove a satellite from the database. This deletes the satellite's record from the satellite table and all records from other tables like beacons, TLEs etc which unambiguously depend on the deleted satellite.

Parameters:

Parameter	Description
satelliteId	Id of the satellite to delete

Expected Payload: none

Replied Payload: The SatDbSatellite data object which has been deleted.

Return Codes:

Code	Description
200	OK, the satellite has been deleted.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exist. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/satellites/{satelliteId}/positions

Returns the list of satellite positions for a given satellite as an array ofSatDbPosition data objects.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: none

Replied Payload: The list of satellite positions as an array ofSatDbPosition data objects. This array may be empty if there are no positions defined for this satellite.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exists. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/satellites/{satelliteId}/beacons

Returns the list of satellite beacons for a given satellite as an array ofSatDbBeacon data objects.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: none

Replied Payload: The list of satellite beacons as an array ofSatDbBeacon data objects. This array may be empty if there are no beacons defined for this satellite.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exists. In this case anApiError document is returned, describing details of the error.

GET /api/v1/satellites/{satelliteId}/beacons/{beaconId}

Returns the information stored for a particular satellite beacon addressed by {satelliteId} / {beaconId} as a SatDbBeacon data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.
beaconId	Id of the beacon.

Expected Payload: none

Replied Payload: The requested SatDbBeacon data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
401	not logged in. In this case anApiError document is returned, describing details of the error.

Code	Description
404	no satellite satellite beacon exists with the given *satelliteId* / *beaconId*. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/satellites/{satelliteId}/beacons

Creates a new satellite beacon with the given attributes. {satelliteId} specifies the satellite for which the beacon shall be created. The unique *beaconId* is generated by the database backend and returned in the response document.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: The SatDbBeacon data object to store. Any *beaconId* definition in the source is ignored as this will be set by the database when the new entry is created. The same applies to *satellite_Id*, the newly created SatDbBeacon data object will be linked to the satellite specified in the URL. If other keys of the data object are missing, default values will be assigned these keys.

Replied Payload: The SatDbBeacon data object added to the datbase. The replied data object contains the *beaconId* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created SatDbBeacon data object is replied as payload
401	not logged in. In this case anApiError document is returned,describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

PATCH /api/v1/satellites/{satelliteId}/beacons/{beaconId}

Updates an existing satellite beacon with the given *satelliteId* / *beaconId* from the attached SatDbBeacon data object.

Parameters:

Parameter	Description

Parameter	Description
satelliteId	Id of the satellite.
beaconId	Id of the beacon.

Expected Payload: The SatDbBeacon data object containing the fields to be updated in the database. If the data object contains values for *beaconId* and / or *satelliteId*, these values are ignored. *beaconId* and *satelliteId* are immutable once a record has been created. Keys not contained in the supplied data object are retained unchanged.

Replied Payload: The SatDbBeacon data object read from the datbase after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no satellite satellite beacon exists with the given *satelliteId* / *beaconId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/beacons/{beaconId}

Remove a satellite beacon from the database.

Parameters:

Parameter	Description
satelliteId	Id of the satellite the beacon belongs to.
beaconId	Id of the beacon to delete.

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description
204	NO CONTENT, the beacon has been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	no satellite satellite beacon exists with the given *satelliteId* / *beaconId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/beacons

Remove all satellite beacons which belong to the given *satelliteId*.

Parameters:

Parameter	Description
satelliteId	Id of the satellite

Return Codes:

Code	Description
204	NO CONTENT, all beacons for the selected satellite have been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	satellite with the given *satelliteId* does not exist. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/satellites/{satelliteId}/tcchans

Returns the list of satellite TC channels for a given satellite as an array ofSatDbTc data objects.

Parameters:

Parameter	Description

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: none

Replied Payload: The list of satellite TC channels as an array ofSatDbTc data objects. This array may be empty if there are no TC channels defined for this satellite.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exists. In this case anApiError document is returned, describing details of the error.

GET /api/v1/satellites/{satelliteId}/tcchans/{tcId}

Returns the information stored for a particular satellite TC channel addressed by {satelliteId} / {tcId} as a SatDbTc data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.
tcId	Id of the TC channel.

Expected Payload: none

Replied Payload: The requested SatDbTc data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite channel.
401	not logged in. In this case anApiError document is returned, describing details of the error.

Code	Description
404	no satellite TC channel exists with the given *satelliteId* / *tcId*. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/satellites/{satelliteId}/tcchans

Creates a new satellite TC channel with the given attributes. {satelliteId} specifies the satellite for which the TC channel shall be created. The unique *tcId* is generated by the database backend and returned in the response document.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: The SatDbTc data object to store. Any *tcId* definition in the source is ignored as this will be set by the database when the new entry is created. The same applies to *satelliteId*, the newly created SatDbTc data object will be linked to the satellite specified in the URL. If other keys of the data object are missing, default values will be assigned these keys.

Replied Payload: The SatDbTc data object added to the datbase. The replied data object contains the *tcId* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created SatDbTc data object is replied as payload
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

PATCH /api/v1/satellites/{satelliteId}/tcchans/{tcId}

Updates an existing satellite TC channel with the given *satelliteId* / *tcId* from the attached SatDbTc data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Parameter	Description
tcId	Id of the TC channel.

Expected Payload: The SatDbTc data object containing the fields to be updated in the database. If the data object contains values for *tcId* and / or *satelliteId*, these values are ignored. *tcId* and *satelliteId* are immutable once a record has been created. Keys not contained in the supplied data object are retained unchanged.

Replied Payload: The SatDbTc data object read from the datbase after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	no satellite satellite TC channel exists with the given*satelliteId* / *tcId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/tcchans/{tcId}

Remove a satellite TC channel from the database.

Parameters:

Parameter	Description
satelliteId	Id of the satellite the TC channel belongs to.
tcId	Id of the TC channel to delete.

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description

Code	Description
204	NO CONTENT, the TC channel has been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	no satellite satellite TC channel exists with the given *satelliteId* / *tcId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/tcchans

Remove all satellite TC channels which belong to the given *satelliteId*.

Parameters:

Parameter	Description
satelliteId	Id of the satellite

Return Codes:

Code	Description
204	NO CONTENT, all TC channels for the selected satellite have been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	satellite with the given *satelliteId* does not exist. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/satellites/{satelliteId}/tmchans

Returns the list of satellite TM channels for a given satellite as an array ofSatDbTm data objects.

Parameters:

Parameter	Description

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: none

Replied Payload: The list of satellite TM channels as an array ofSatDbTm data objects. This array may be empty if there are no tm channels defined for this satellite.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exists. In this case anApiError document is returned, describing details of the error.

GET /api/v1/satellites/{satelliteId}/tmchans/{tmId}

Returns the information stored for a particular satellite TM channel addressed by {satelliteId} / {tmId} as a SatDbTm data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.
tmId	Id of the TM channel.

Expected Payload: none

Replied Payload: The requested SatDbTm data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite channel.
401	not logged in. In this case anApiError document is returned, describing details of the error.

Code	Description
404	no satellite TM channel exists with the given *satelliteId* / *tmId*. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/satellites/{satelliteId}/tmchans

Creates a new satellite TM channel with the given attributes. {satelliteId} specifies the satellite for which the tm channel shall be created. The unique *tmId* is generated by the database backend and returned in the response document.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: The SatDbTm data object to store. Any *tmId* definition in the source is ignored as this will be set by the database when the new entry is created. The same applies to *satelliteId*, the newly created SatDbTm data object will be linked to the satellite specified in the URL. If other keys of the data object are missing, default values will be assigned these keys.

Replied Payload: The SatDbTm data object added to the datbase. The replied data object contains the *tmId* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created SatDbTm data object is replied as payload
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

PATCH /api/v1/satellites/{satelliteId}/tmchans/{tmId}

Updates an existing satellite TM channel with the given *satelliteId* / *tmId* from the attached SatDbTm data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Parameter	Description
tmId	Id of the TM channel.

Expected Payload: The SatDbTm data object containing the fields to be updated in the database. If the data object contains values for *tmId* and / or *satelliteId*, these values are ignored. *tmId* and *satelliteId* are immutable once a record has been created. Keys not contained in the supplied data object are retained unchanged.

Replied Payload: The SatDbTm data object read from the datbase after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	no satellite satellite TM channel exists with the given *satelliteId* / *tmId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/tmchans/{tmId}

Remove a satellite TM channel from the database.

Parameters:

Parameter	Description
satelliteId	Id of the satellite the TM channel belongs to.
tmId	Id of the TM channel to delete.

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description

Code	Description
204	NO CONTENT, the TM channel has been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	no satellite satellite TM channel exists with the given *satelliteId* / *tmId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/tmchans

Remove all satellite TM channels which belong to the given *satelliteId*.

Parameters:

Parameter	Description
satelliteId	Id of the satellite

Return Codes:

Code	Description
204	NO CONTENT, all TM channels for the selected satellite have been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	satellite with the given *satelliteId* does not exist. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/satellites/{satelliteId}/i11params

Returns the list of I11 parameter sets for a given satellite as an array ofSatDbI11Data data objects.

Parameters:

Parameter	Description

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: none

Replied Payload: The list of I11 parameter sets as an array ofSatDbI11Data data objects. This array may be empty if there are no I11 parameter sets defined for this satellite.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exists. In this case anApiError document is returned, describing details of the error.

GET /api/v1/satellites/{satelliteId}/i11params/{i11Id}

Returns the information stored for a particular I11 parameter set addressed by {satelliteId} / {i11Id} as a SatDbI11Data data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.
i11Id	Id of the I11 parameter set.

Expected Payload: none

Replied Payload: The requested SatDbI11Data data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
401	not logged in. In this case anApiError document is returned, describing details of the error.

Code	Description
404	no satellite I11 parameter set exists with the given *satelliteId* / *i11Id*. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/satellites/{satelliteId}/i11params

Creates a new I11 parameter set with the given attributes. {satelliteId} specifies the satellite for which the I11 parameter set shall be created. The unique *i11Id* is generated by the database backend and returned in the response document.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: The SatDbI11Data data object to store. Any *i11Id* definition in the source is ignored as this will be set by the database when the new entry is created. The same applies to *satellite_Id*, the newly created SatDbI11Data data object will be linked to the satellite specified in the URL. If other keys of the data object are missing, default values will be assigned these keys.

Replied Payload: The SatDbI11Data data object added to the datbase. The replied data object contains the *i11Id* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created SatDbI11Data data object is replied as payload
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/satellites/{satelliteId}/i11params/{i11Id}

Updates an existing I11 parameter set with the given *satelliteId* / *i11Id* from the attached SatDbI11Data data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Parameter	Description
i11Id	Id of the I11 parameter set.

Expected Payload: The SatDbI11Data data object containing the fields to be updated in the database. If the data object contains values for *i11Id* and / or *satelliteId*, these values are ignored. *i11Id* and *satelliteId* are immutable once a record has been created. Keys not contained in the supplied data object are retained unchanged.

Replied Payload: The SatDbI11Data data object read from the datbase after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no satellite I11 parameter set exists with the given *satelliteId* / *i11Id*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/i11params/{i11Id}

Remove a I11 parameter set from the database.

Parameters:

Parameter	Description
satelliteId	Id of the satellite the I11 parameter set belongs to.
i11Id	Id of the I11 parameter set to delete.

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description

Code	Description
204	NO CONTENT, the I11 parameter set has been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	no satellite I11 parameter set exists with the given *satelliteId* / *i11Id*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/i11params

Remove all I11 parameter sets which belong to the given *satelliteId*.

Parameters:

Parameter	Description
satelliteId	Id of the satellite

Return Codes:

Code	Description
204	NO CONTENT, all I11 parameter sets for the selected satellite have been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	satellite with the given *satelliteId* does not exist. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/satellites/{satelliteId}/tleparams

Returns the list of TLE parameter sets for a given satellite as an array of SatDbTLEData data objects.

Parameters:

Parameter	Description

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: none

Replied Payload: The list of TLE parameter sets as an array ofSatDbTLEData data objects. This array may be empty if there are no TLE parameter sets defined for this satellite. As there may be actually only one TLE parameter set for each satellite, the array may contain only zero or one elements.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exists. In this case anApiError document is returned, describing details of the error.

GET /api/v1/satellites/{satelliteId}/tleparams/{noradNumber}

Returns the information stored for a particular TLE parameter set addressed by {satelliteId} / {noradNumber} as a SatDbTLEData data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.
noradNumber	Id / norad number of the TLE parameter set.

Expected Payload: none

Replied Payload: The requested SatDbTLEData data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.

Code	Description
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no satellite TLE parameter set exists with the given *satelliteId* / *noradNumber*. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/satellites/{satelliteId}/tleparams

Creates a new TLE parameter set with the given attributes. {satelliteId} specifies the satellite for which the TLE parameter set shall be created. The unique *noradNumber* is taken from the *noradNumber* field in the satellite's master data and returned in the response document. Hence it is important that the *noradNumber* field in the satellite's master data has been set before to the correct value.

As there may be only TLE parameter set assigned to a satellite, this POST method behaves in a special way: If there already exists a TLE parameter set for this satellite, it will be silently overwritten by the POST call. The call proceeds, stores the new data and returns a 201 return code. This behavior ensures, that not more than one TLE parameter set can be assigned to a satellite.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: The SatDbTLEData data object to store. Any *noradNumber* definition in the source is ignored as this will be taken from the *noradNumber* field in the satellite's master data when the new entry is created. The same applies to *satellite_Id*, the newly created SatDbTLEData data object will be linked to the satellite specified in the URL. If other keys of the data object are missing, default values will be assigned these keys.

Replied Payload: The SatDbTLEData data object added to the datbase. The replied data object contains the *noradNumber* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created SatDbTLEData data object is replied as payload
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

PATCH /api/v1/satellites/{satelliteId}/tleparams/{noradNumber}

Updates an existing TLE parameter set with the given *satelliteId* / *noradNumber* from the attached SatDbTLEData data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.
noradNumber	Id of the TLE parameter set.

Expected Payload: The SatDbTLEData data object containing the fields to be updated in the database. If the data object contains values for *noradNumber* and / or *satelliteId*, these values are ignored. *noradNumber* and *satelliteId* are immutable once a record has been created. Keys not contained in the supplied data object are retained unchanged.

Replied Payload: The SatDbTLEData data object read from the datbase after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no satellite TLE parameter set exists with the given *satelliteId* / *noradNumber*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/tleparams/{noradNumber}

Remove a TLE parameter set from the database.

Parameters:

Parameter	Description
satelliteId	Id of the satellite the TLE parameter set belongs to.
noradNumber	Id of the TLE parameter set to delete.

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description
204	NO CONTENT, the TLE parameter set has been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	no satellite TLE parameter set exists with the given *satelliteId* / *noradNumber*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/tleparams

Remove all TLE parameter sets which belong to the given *satelliteId*.

Parameters:

Parameter	Description
satelliteId	Id of the satellite

Return Codes:

Code	Description
204	NO CONTENT, all TLE parameter sets for the selected satellite have been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	satellite with the given *satelliteId* does not exist. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/ttracks

In general, the table track ('ttrack') API calls are located at /api/v1/antennas. The only call located at /api/v1/satellites is to temove all table track data sets which belong to the given *satelliteId* (for all antennas).

Parameters:

Parameter	Description
satelliteId	Id of the satellite

Return Codes:

Code	Description
204	NO CONTENT, all table track data sets for the selected satellite have been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	satellite with the given *satelliteId* does not exist. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

5.2.27 /api/v1/satoperators

This API endpoint gives access to various functions of the sat-nms satoperator database table, permitting to read, add, change or delete entries at the database

Supported HTTP methods: GET, POST, PATCH, DELETE, HEAD, OPTIONS

The API generally does not permit PATCH or DELETE operations on entire tables, this to protect the database from being accidentally be deleted. Even POST operations only work with single data objects.

GET /api/v1/satoperators

Returns the list of stored satoperators as an array ofSatDbSatOperator data objects.

Parameters: none

Expected Payload: none

Replied Payload: The list of stored satoperators as an array ofSatDbSatOperator data objects.

Return Codes:

Code	Description
200	OK, returns a JSON document with all satoperators and their attributes.

Code	Description
401	not logged in. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/satoperators/{operatorId}

Returns the basic properties of the satoperator addressed by {operatorId} as a SatDbSatOperator data object.

Parameters:

Parameter	Description
operatorId	Id of the satoperator.

Expected Payload: none

Replied Payload: The requested SatDbSatOperator data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satoperator.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	satoperator with the given *operatorId* does not exists. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/satoperators

Creates a new satoperator with the given attributes. The unique *operatorId* is generated by the database backend and returned in the response document.

Parameters: none

Expected Payload: The SatDbSatOperator data object to store. Any *operatorId* definition in the source is ignored as this will be set by the database when the new entry is created. If other keys of the data object are missing, default values will be assigned these keys.

Replied Payload: The SatDbSatOperator data object added to the datbase. The replied data object contains the *operatorId* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created SatDbSatOperator data object is replied as payload
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

PATCH /api/v1/satoperators/{operatorId}

Updates an existing satellite operator with the given *operatorId* from the attached SatDbSatOperator data object.

Parameters:

Parameter	Description
operatorId	Id of the satoperator.

Expected Payload: The SatDbSatOperator data object containing the fields to be updated in the database. If the data object contains an *operatorId* key, the value is ignored. The database record is solely identified by *operatorId* set in the URL, the *operatorId* of the record is immutable. Keys not contained in the supplied data object are retained unchanged.

Replied Payload: The SatDbSatOperator data object read from the datbase after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	the satellite operator with the given *operatorId* does not exists. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/satoperators/{operatorId}

Remove a satoperator from the database. This deletes the satoperator's record from the satoperator table and removes all references to the deleted satoperator in the *satellites* table.

Parameters:

Parameter	Description
operatorId	Id of the satoperator to delete

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description
204	NO CONTENT, the satellite has been deleted.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	the satellite operator with the given *operatorId* does not exists. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

5.2.28 /api/v1/channels

This API endpoint gives access to various functions of the*channels* database table, permitting to read, add, change or delete entries at the database. This database table holds all satellite channels known in the satnms database.

Supported HTTP methods: GET, POST, PATCH, DELETE, HEAD, OPTIONS

The API generally does not permit PATCH or DELETE operations on entire tables, this to protect the database from being accidentally be deleted. Even POST operations only work with single data objects.

GET /api/v1/channels/

Returns the list of all satellite channel records as an array ofChannelData data objects.

Parameters: none

Expected Payload: none

Replied Payload: The list of satellite channel records as an array ofChannelData data objects.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/channels/ip

Returns the list of all 'IP' type satellite channel records as an array ofChannelData data objects.

Parameters: none

Expected Payload: none

Replied Payload: The list of satellite channel records as an array ofChannelData data objects. This array may be empty if there are no such channel records in the database..

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/channels/asi

Returns the list of all 'ASI' type satellite channel records as an array ofChannelData data objects.

Parameters: none

Expected Payload: none

Replied Payload: The list of satellite channel records as an array ofChannelData data objects. This array may be empty if there are no such channel records in the database.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/channels/sat

Returns the list of all 'SAT' type satellite channel records as an array ofChannelData data objects.

Parameters: none

Expected Payload: none

Replied Payload: The list of satellite channel records as an array ofChannelData data objects. This array may be empty if there are no such channel records in the database.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case anApiError document is returned, describing details of the error.

GET /api/v1/channels/sat/{satelliteId}

Returns the list of 'SAT' type satellite channel records for a given satellite as an array of ChannelData data objects.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: none

Replied Payload: The list of satellite channel records as an array ofChannelData data objects. This array may be empty if there are no channel records defined for this satellite.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exists. In this case anApiError document is returned, describing details of the error.

GET /api/v1/channels/{channelId}

Returns the information stored for a particular channel record addressed by {channelId} as a ChannelData data object.

Parameters:

Parameter	Description
channelId	Id of the channel record.

Expected Payload: none

Replied Payload: The requested ChannelData data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no satellite channel record exists with the given *channelId*. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/channels

Creates a new channel record with the given attributes.

Parameters: none

Expected Payload: The ChannelData data object to store. The channel definition must conform the following conditions:

- The *decoderInput* field must be set to one of 'SAT', 'IP' or 'ASI' to define the type of channel.
- The *name* field must be set to a value which is unique for the given satellite or the API call will decline to create the database record.
- 'SAT' type records must contain a valid *satelliteId*
- Any *channelId* definition in the source is ignored as this will be set by the database when the new entry is created.

Replied Payload: The ChannelData data object added to the datbase. The replied data object contains the *channelId* assigned to the newly created record.

Return Codes:

Code	Description

Code	Description
201	CREATED, the created ChannelData data object is replied as payload
400	BAD_REQUEST if the supplied ChannelData does not conform the conditions listed above.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/channels/{channelId}

Updates an existing channel record with the given *channelId* from the attached ChannelData data object.

Parameters:

Parameter	Description
channelId	Id of the channel record.

Expected Payload: The ChannelData data object containing the fields to be updated in the database. Fields not contained in the supplied data object are retained unchanged. The channel definition must conform the following conditions:

- If it changes the *name* of the record, this must be set to a value which is unique for the given satellite or the API call will decline to create the database record.
- 'SAT' type records must contain a valid *satelliteId*
- Any *channelId* definition in the source is ignored as the *channelId* is immutable once a record has been created.

Replied Payload: The ChannelData data object read from the database after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
400	BAD_REQUEST if the supplied ChannelData does not conform the conditions listed above.
401	not logged in. In this case an ApiError document is returned, describing details of the error.

Code	Description
404	no satellite channel record exists with the given *channelId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/channels/{channelId}

Remove a channel record from the database.

Parameters:

Parameter	Description
channelId	Id of the channel record to delete.

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description
204	NO CONTENT, the channel record has been deleted.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no satellite channel record exists with the given *channelId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/channels/sat/{satelliteId}

Remove all channel records which belong to the given *satelliteId*.

Parameters:

Parameter	Description
satelliteId	Id of the satellite

Return Codes:

Code	Description
204	NO CONTENT, all channel records for the selected satellite have been deleted.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exist. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

5.2.29 /api/v1/streamkeys

This API endpoint gives access to various functions of the*streamkeys* database table, permitting to read, add, change or delete entries at the database. This database table holds all BISS-1/BISS-E keys and SRT passphrases known in the sat-nms database.

Supported HTTP methods: GET, POST, PATCH, DELETE, HEAD, OPTIONS

The API generally does not permit PATCH or DELETE operations on entire tables, this to protect the database from being accidentally be deleted. Even POST operations only work with single data objects.

GET /api/v1/streamkeys

Returns the list of stored stream keys as an array ofStreamKeyData data objects.

Parameters: none

Expected Payload: none

Replied Payload: The list of stored stream keys as an array ofStreamKeyData data objects.

Return Codes:

Code	Description
200	OK, returns a JSON document with all streamkeys and their attributes.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/streamkeys/{keyId}

Returns the basic properties of the stream key addressed by {keyId} as aStreamKeyData data object.

Parameters:

Parameter	Description
keyId	Id of the stream key.

Expected Payload: none

Replied Payload: The requested StreamKeyData data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested stream key.
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	stream key with the given *keyId* does not exists. In this case anApiError document is returned, describing details of the error.

POST /api/v1/streamkeys

Creates a new stream key with the given attributes. The unique*keyId* is generated by the database backend and returned in the response document.

Parameters: none

Expected Payload: The StreamKeyData data object to store. Any *keyId* definition in the source is ignored as this will be set by the database when the new entry is created. The *name* field in the supplied payload must be unique over the entire table or the API call will decline to create the database record.

Replied Payload: The StreamKeyData data object added to the database. The replied data object contains the *keyId* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created StreamKeyData data object is replied as payload
400	BAD_REQUEST. Either the *name* supplied with the request is not unique or the*key* does not conform to the format restrictions for the given *keyType*

Code	Description
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/streamkeys/{keyId}

Updates an existing stream key with the given *keyId* from the attached StreamKeyData data object.

Parameters:

Parameter	Description
keyId	Id of the stream key.

Expected Payload: The StreamKeyData data object containing the fields to be updated in the database. If the data object contains an *keyId* key, the value is ignored. The database record is solely identified by *keyId* set in the URL, the *keyId* of the record is immutable. Keys not contained in the supplied data object are retained unchanged. The *name* field in the supplied payload must be unique over the entire table or the API call will decline to modify the database record.

Replied Payload: The StreamKeyData data object read from the database after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
400	BAD_REQUEST. Either the *name* supplied with the request is not unique or the *key* does not conform to the format restrictions for the given *keyType*
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	the stream key with the given *keyId* does not exists. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/streamkeys/{keyId}

Remove a stream key from the database.

Parameters:

Parameter	Description
keyId	Id of the stream key to delete

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description
204	NO CONTENT, the satellite has been deleted.
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	the stream key with the given *keyId* does not exists. In this case anApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

5.2.30 /api/v1/antennas

This API endpoint gives access to various functions of the sat-nms antenna database table, permitting to read, add, change or delete entries at the database

Supported HTTP methods: GET, POST, PATCH, DELETE, HEAD, OPTIONS

The API generally does not permit PATCH or DELETE operations on entire tables, this to protect the database from being accidentally be deleted. Even POST operations only work with single data objects.

GET /api/v1/antennas

Returns the list of stored antennas as an array ofSatDbAntenna data objects.

Parameters: none

Expected Payload: none

Replied Payload: The list of stored antennas as an array ofSatDbAntenna data objects.

Return Codes:

Code	Description
200	OK, returns a JSON document with all antennas and their attributes.
401	not logged in. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/antennas/{antennaId}

Returns the basic properties of the antenna addressed by {antennaId} as aSatDbAntenna data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.

Expected Payload: none

Replied Payload: The requested SatDbAntenna data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested antenna.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	antenna with the given *antennaId* does not exists. In this case anApiError document is returned, describing details of the error.

POST /api/v1/antennas

Creates a new antenna with the given attributes. The unique *antennaId* is generated by the database backend and returned in the response document.

Parameters: none

Expected Payload: The SatDbAntenna data object to store. Any *antennaId* definition in the source is ignored as this will be set by the database when the new entry is created. If other keys of the data object are missing, default values will be assigned these keys.

Replied Payload: The SatDbAntenna data object added to the database. The replied data object contains the *antennaId* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created SatDbAntenna data object is replied as payload
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

PATCH /api/v1/antennas/{antennaId}

Updates an existing antenna with the given *antennaId* from the attached SatDbAntenna data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.

Expected Payload: The SatDbAntenna data object containing the fields to be updated in the database. If the data object contains an *antennaId* key, the value is ignored. The database record is solely identified by *antennaId* set in the URL, the *antennaId* of the record is immutable. Keys not contained in the supplied data object are retained unchanged.

Replied Payload: The SatDbAntenna data object read from the database after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	antenna with the given *antennaId* does not exists. In this case anApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/antennas/{antennaId}

Remove a antenna from the database. This deletes the antenna's record from the antenna table and all records from the *satpositions* table which refer to the deleted antenna.

Parameters:

Parameter	Description
antennaId	Id of the antenna to delete

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description
204	NO CONTENT, the satellite has been deleted.
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	antenna with the given *antennaId* does not exists. In this case anApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

GET /api/v1/antennas/{antennaId}/positions

Returns the list of satellite positions for a given antenna as an array ofSatDbPosition data objects.

Parameters:

Parameter	Description
antennaId	Id of the antenna.

Expected Payload: none

Replied Payload: The list of satellite positions as an array ofSatDbPosition data objects. This array may be empty if there are no satellite positions defined for this antenna.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information

Code	Description
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	the antenna with the given *antennaId* does not exists. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/antennas/{antennaId}/positions/{positionId}

Returns the information stored for a particular satellite position addressed by {antennaId} / {positionId} as a SatDbPosition data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
positionId	Id of the satellite position.

Expected Payload: none

Replied Payload: The requested SatDbPosition data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested antenna.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no antenna satellite position exists with the given *antennaId* / *positionId*. In this case an ApiError document is returned, describing details of the error.

ST /api/v1/antennas/{antennaId}/positions

Creates a new satellite position with the given attributes. {antennaId} specifies the antenna for which the satellite position shall be created. The unique *positionId* is generated by the database backend and returned in the response document.

Parameters:

Parameter	Description

Parameter	Description
antennaId	Id of the antenna.

Expected Payload: The SatDbPosition data object to store. Any *positionId* definition in the source is ignored as this will be set by the database when the new entry is created. The same applies to *antennaId*, the newly created SatDbPosition data object will be linked to the antenna specified in the URL. If other keys of the data object are missing, default values will be assigned these keys.

Replied Payload: The SatDbPosition data object added to the database. The replied data object contains the *positionId* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created SatDbPosition data object is replied as payload
401	not logged in. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/antennas/{antennaId}/positions/{positionId}

Updates an existing satellite position with the given *antennaId / positionId* from the attached SatDbPosition data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
positionId	Id of the satellite position.

Expected Payload: The SatDbPosition data object containing the fields to be updated in the database. If the data object contains values for *positionId* and / or *antennaId*, these values are ignored. *positionId* and *antennaId* are immutable once a record has been created. Keys not contained in the supplied data object are retained unchanged.

Replied Payload: The SatDbPosition data object read from the database after the requested changes have been applied.

Return Codes:

Code	Description

Code	Description
200	OK, changes have been applied.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no antenna satellite position exists with the given *antennaId* / *positionId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/antennas/{antennaId}/positions/{positionId}

Remove a satellite position from the database.

Parameters:

Parameter	Description
antennaId	Id of the antenna the satellite position belongs to.
positionId	Id of the satellite position to delete.

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description
204	NO CONTENT, the satellite position has been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	no antenna satellite position exists with the given *antennaId* / *positionId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/antennas/{antennaId}/positions

Remove all satellite positions which belong to the given *antennaId*.

Parameters:

Parameter	Description
antennaId	Id of the antenna

Return Codes:

Code	Description
204	NO CONTENT, all satellite positions for the selected antenna have been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	antenna with the given *antennaId* does not exist. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/antennas/{antennaId}/ttracks

Returns the list of table track data sets for a given antenna as an array ofSatDbTableTracking data objects.

Parameters:

Parameter	Description
antennaId	Id of the antenna.

Expected Payload: none

Replied Payload: The list of table track data sets as an array ofSatDbTableTracking data objects. This array may be empty if there are no table track data sets defined for this antenna.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	antenna with the given *antennaId* does not exists. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/antennas/{antennaId}/{satelliteId}/ttracks

Returns the list of table track data sets for a given satellite/antenna combination as an array of SatDbTableTracking data objects.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
satelliteId	Id of the satellite.

Expected Payload: none

Replied Payload: The list of table track data sets as an array ofSatDbTableTracking data objects. This array may be empty if there are no table track data sets defined for this satellite.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exists. In this case anApiError document is returned, describing details of the error.

GET /api/v1/antennas/{antennaId}/{satelliteId}/ttracks/{ttId}

Returns the information stored for a particular table track data set addressed by {antennaId} / {satelliteId} / {ttId} as a SatDbTableTracking data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
satelliteId	Id of the satellite.
ttId	Id of the table track data set.

Expected Payload: none

Replied Payload: The requested SatDbTableTracking data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no satellite table track data set exists with the given *satelliteId* / *ttId*. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/antennas/{antennaId}/{satelliteId}/ttracks

Creates a new table track data set with the given attributes. {antennaId} / {satelliteId} specifies the satellite / antenna combination for which the table track data set shall be created. The unique *ttId* is generated by the database backend and returned in the response document.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
satelliteId	Id of the satellite.

Expected Payload: The SatDbTableTracking data object to store. Any *ttId* definition in the source is ignored as this will be set by the database when the new entry is created. The same applies to *satellite_Id*, the newly created SatDbTableTracking data object will be linked to the satellite specified in the URL. If other keys of the data object are missing, default values will be assigned these keys.

Replied Payload: The SatDbTableTracking data object added to the datbase. The replied data object contains the *ttId* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created SatDbTableTracking data object is replied as payload
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

PATCH /api/v1/antennas/{antennaId}/{satelliteId}/ttracks/{ttId}

Updates an existing table tracking data set with the given*antennaId* / *satelliteId* / *ttId* from the attached SatDbTableTracking data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
satelliteId	Id of the satellite.
ttId	Id of the table track data set.

Expected Payload: The SatDbTableTracking data object containing the fields to be updated in the database. If the data object contains values for any of *ttId*, *satelliteId* or *antennaId*, these values are ignored. *ttId*, *satelliteId* and *antennaId* are immutable once a record has been created. Keys not contained in the supplied data object are retained unchanged.

Replied Payload: The SatDbTableTracking data object read from the datbase after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	no satellite table track data set exists with the given*satelliteId* / *ttId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/antennas/{antennaId}/{satelliteId}/ttracks/{ttId}

Remove a table track data set from the database.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
satelliteId	Id of the satellite the table track data set belongs to.

Parameter	Description
ttId	Id of the table track data set to delete.

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description
204	NO CONTENT, the table track data set has been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	no satellite table track data set exists with the given *satelliteId* / *ttId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/antennas/{antennaId}/{satelliteId}/ttracks

Remove all table track data sets which belong to the given *satelliteId* / *antennaId* combination.

Parameters:

Parameter	Description
satelliteId	Id of the satellite
antennaId	Id of the antenna.

Return Codes:

Code	Description
204	NO CONTENT, all table track data sets for the selected satellite have been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	satellite with the given *satelliteId* does not exist. In this case an ApiError document is returned, describing details of the error.

Code	Description
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/antennas/{antennaId}/ttracks

Remove all table track data sets which belong to the given antenna.

Parameters:

Parameter	Description
antennaId	Id of the antenna

Return Codes:

Code	Description
204	NO CONTENT, all table track data sets for the selected satellite have been deleted.
401	not logged in. In this case anApiError document is returned,describing details of the error.
404	satellite with the given *satelliteId* does not exist. In this case anApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

GET /api/v1/antennas/{antennaId}/beacon-atten/{beaconId}

Returns the attenuation value addressed by {antennaId} / {beaconId} as a SatDbBeaconAttenuation data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
beaconId	Id of the beacon.

Expected Payload: none

Replied Payload: The requested SatDbBeaconAttenuation data object. If no attenuation has been stored for this antennaId/beaconId combination, an empty SatDbBeaconAttenuation data object with the attenuation value set to 0.0 is returned.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
400	BAD_REQUEST, antennaId or beaconId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.

PUT /api/v1/antennas/{antennaId}/beacon-atten/{beaconId}

Sets the attenuation value for an {antennaId} / {beaconId} combination. Returns the attenuation set as a SatDbBeaconAttenuation data object. If no attenuation record exists for this antenna / beacon combination a new one is created. If there is already an attenuation record for this combination, the existing one will be overwritten.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
beaconId	Id of the beacon.

Expected Payload: The attenuation to set as a SatDbBeaconAttenuation data object. Only the attenuation is read from this data object, any IDs passed to this API function are ignored. The IDs are defined solely by the URL parameters.

Replied Payload: The SatDbBeaconAttenuation data object with the updated attenuation .

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
400	BAD_REQUEST, antennaId or beaconId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.

Code	Description
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

PATCH /api/v1/antennas/{antennaId}/beacon-atten/{beaconId}

Sets the attenuation value for an {antennaId} / {beaconId} combination. Returns the attenuation set as a SatDbBeaconAttenuation data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
beaconId	Id of the beacon.

Expected Payload: The attenuation to set as a SatDbBeaconAttenuation data object. Only the attenuation is read from this data object, any IDs passed to this API function are ignored. The IDs are defined solely by the URL parameters.

Replied Payload: The SatDbBeaconAttenuation data object with the updated attenuation .

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
400	BAD_REQUEST, antennaId or beaconId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no attenuation record exists with the given *satelliteId* / *beaconId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/antennas/{antennaId}/beacon-atten/{beaconId}

Deletes the attenuation value addressed by {antennaId} / {beaconId}.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
beaconId	Id of the beacon.

Expected Payload: none

Replied Payload: none.

Return Codes:

Code	Description
204	OK, the record has been deleted
400	BAD_REQUEST, antennaId or beaconId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	no attenuation record exists with the given *satelliteId* / *beaconId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

GET /api/v1/antennas/{antennaId}/tc-atten/{tcId}

Returns the attenuation values addressed by {antennaId} / {tcId} as aSatDbTcAttenuation data object. If no attenuation has been stored for this antennaId/tcId combination, an empty SatDbTcAttenuation data object with the attenuation values set to 0.0 is returned.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
tcId	Id of the tc channel.

Expected Payload: none

Replied Payload: The requested SatDbTcAttenuation data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
400	BAD_REQUEST, antennaId or tcId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.

PUT /api/v1/antennas/{antennaId}/tc-atten/{tcId}

Sets the attenuation values for an {antennaId} / {tcId} combination. Returns the attenuation set as a SatDbTcAttenuation data object. If no attenuation record exists for this antenna / tc combination a new one is created. If there is already an attenuation record for this combination, the existing one will be overwritten.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
tcId	Id of the tc channel.

Expected Payload: The attenuation to set as a SatDbTcAttenuation data object. Only the attenuation values are read from this data object, any IDs passed to this API function are ignored. The IDs are defined solely by the URL parameters.

Replied Payload: The SatDbTcAttenuation data object with the updated attenuation .

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
400	BAD_REQUEST, antennaId or tcId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/antennas/{antennaId}/tc-atten/{tcId}

Modifies the attenuation values for an {antennaId} / {tcId} combination. Returns the attenuation set as a SatDbTcAttenuation data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
tcId	Id of the tc channel.

Expected Payload: One or both attenuation values to set in aSatDbTcAttenuation data object. Only the attenuation values are read from this data object, any IDs passed to this API function are ignored. The IDs are defined solely by the URL parameters.

Replied Payload: The SatDbTcAttenuation data object with the updated attenuation.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
400	BAD_REQUEST, antennaId or tcId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	no attenuation record exists with the given *satelliteId* / *tcId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/antennas/{antennaId}/tc-atten/{tcId}

Deletes the attenuation values addressed by {antennaId} / {tcId}.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
tcId	Id of the tc channel.

Expected Payload: none

Replied Payload: none.

Return Codes:

Code	Description
204	OK, the record has been deleted
400	BAD_REQUEST, antennaId or tcId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	no attenuation record exists with the given *satelliteId* / *tcId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

GET /api/v1/antennas/{antennaId}/tm-atten/{tmId}

Returns the attenuation values addressed by {antennaId} / {tmId} as aSatDbTmAttenuation data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
tmId	Id of the tm channel.

Expected Payload: none

Replied Payload: The requested SatDbTmAttenuation data object. If no attenuation has been stored for this antennaId/tcId combination, an empty SatDbTmAttenuation data object with the attenuation values set to 0.0 is returned.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
400	BAD_REQUEST, antennaId or tmId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.

Code	Description
401	not logged in. In this case an ApiError document is returned, describing details of the error.

PUT /api/v1/antennas/{antennaId}/tm-atten/{tmId}

Sets the attenuation values for an {antennaId} / {tmId} combination. Returns the attenuation values set as a SatDbTmAttenuation data object. If no attenuation record exists for this antenna / tm combination a new one is created. If there is already an attenuation record for this combination, the existing one will be overwritten.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
tmId	Id of the tm channel.

Expected Payload: The attenuation to set as a SatDbTmAttenuation data object. Only the attenuation values are read from this data object, any IDs passed to this API function are ignored. The IDs are defined solely by the URL parameters.

Replied Payload: The SatDbTmAttenuation data object with the updated attenuation .

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
400	BAD_REQUEST, antennaId or tmId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/antennas/{antennaId}/tm-atten/{tmId}

Modifies the attenuation values for an {antennaId} / {tmId} combination. Returns the attenuation set as a SatDbTmAttenuation data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
tmId	Id of the tm channel.

Expected Payload: One or both attenuation values to set in aSatDbTmAttenuation data object. Only the attenuation values are read from this data object, any IDs passed to this API function are ignored. The IDs are defined solely by the URL parameters.

Replied Payload: The SatDbTmAttenuation data object with the updated attenuation.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
400	BAD_REQUEST, antennaId or tmId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	no attenuation record exists with the given*satelliteId / tmId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/antennas/{antennaId}/tm-atten/{tmId}

Deletes the attenuation values addressed by {antennaId} / {tmId}.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
tmId	Id of the tm channel.

Expected Payload: none

Replied Payload: none.

Return Codes:

Code	Description
204	OK, the record has been deleted
400	BAD_REQUEST, antennaId or tmId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	no attenuation record exists with the given*satelliteId* / *tmId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

5.2.31 /api/v1/satdetails

This API endpoint reports all relevant data for a satellite in a compact form. It has been added for convenience, unburdening the front end programmer from the need to gather this information from different database tables using multiple API calls.

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/v1/satdetails/{satelliteId}

Returns detailed properties of the satellite addressed by {satelliteId} as aSatDbSatDetails data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: none

Replied Payload: The requested SatDbSatDetails data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
401	not logged in. In this case anApiError document is returned, describing details of the error.

Code	Description
404	satellite with the given *satelliteId* does not exists. In this case anApiError document is returned, describing details of the error.

5.2.32 /api/v1/tleimport

This API endpoint lets you upload and import a list of TLE parameter sets from a data file. The expected file format is compatible to the well known 'geo.txt' file from celestrak.com. The file must contain three lines of text for each satellite, the firs line contains the satellite name, followed by two lines in TLE format. The norad number contained in both lines of the TLE format is used to identify for which satellite each record shall be used. The satellite name is treated as a comment only.

Supported HTTP methods: POST, HEAD, OPTIONS

POST /api/v1/tleimport

Expects a SatDbTleImport document containing the file to import along with some additional parameters.

Parameters: none

Expected Payload: The SatDbTleImport document containing the file to import and the import mode to use.

Replied Payload: none, except for errors.

Return Codes:

Code	Description
204	OK, the file has been imported to the database. Do data is returned.
400	The file could not be parsed. In this case anApiError document is returned, describing details of the error.
401	not logged in. In this case anApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

5.2.33 /api/v1/eventreport

This API endpoint lets you create reports from the M&C system event database. The M&C systems managed by the backend share a common event database which store all faults, warnings and informational messages occurring in the particular M&C systems.

A report from the event database is created by sending anEventReportRequest document with a POST call to this API endpoint. The EventReportRequest defines the time span to read and some filter settings which allow to read a set of messages with a specific content.

The API call responds with a EventReport document containing the requested event report.

Supported HTTP methods: POST, HEAD, OPTIONS

POST /api/v1/eventreport

Expects an EventReportRequest document containing the time span to read and the report filter settings.

Parameters: none

Expected Payload: The EventReportRequest document containing the time span to read and the report filter settings.

Replied Payload: The EventReport document containing the requested report.

Return Codes:

Code	Description
200	OK, the request has been processed, the report generated is replied.
500	The event database does not reply (internal error). In this case anApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.

5.2.34 /api/v1/eventack

The sat-nms event database provides a "not yet acknowledged" flag with each event in the database. This flag is initially set for all faults. This API endpoint gives access to the "acknowledge" function which clears this flag for a single event or for all events which have the flag set.

To acknowledge on or all events, the front end has to send anEventAck document with a POST call to this API end point. The EventAck document defines if all events or a particular one shall be acknowledged.

Supported HTTP methods: POST, HEAD, OPTIONS

POST /api/v1/eventack

Expects an EventAck document which defines if all events or a particular one shall be acknowledged.

Parameters: none

Expected Payload: The EventAck document which defines if all events or a particular one shall be acknowledged.

Replied Payload: none, except for errors.

Return Codes:

Code	Description
204	OK, the request has been processed. Do data is returned. There is no error checking, acknowledging a non-existing event or an event which does not require acknowledging is not treated as a fault.
500	The event database does not reply (internal error). In this case anApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.

5.2.35 /api/v1/trackmem

When subscribing for a '.state.mode4' parameter of a SatService ACU-ODM, the front end receives a message which indicates 'a new tracking history for this antenna is available' instead of sending the bulky tracking history itself over the websocket connection.

After the front end received this message, it can use the /api/v1/trackmem API call to retrieve the full tracking history as a TrackingHistory data object.

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/v1/trackmem/{odmName}

Parameters:

Parameter	Description
odmName	The device name of the ODM for which the tracking history shall be read. This must be a fully qualified name including name of the M&C where the ODM is managed.

Expected Payload: none

Replied Payload: A TrackingHistory document containing the tracking history of the referenced antenna controller.

Return Codes:

Code	Description

Code	Description
200	OK, the requested TrackingHistory document is returned.
400	Either the requested device does not exist or the device has not yet sent a tracking history. This may happen if the front end did no subscribe for the 'state.mode4' of the ODM before or if the ODM is not operational. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.

5.2.36 /api/v1/treeview

This API end point lets you retrieve the treeview data structure as a whole or partially by requesting only a branch of the tree. The API also provides a PUT method for this end point which lets you replace the user defined part of the tree for a given M&C name.

Supported HTTP methods: GET, PUT, HEAD, OPTIONS

GET /api/v1/treeview

Gets the complete tree definition for all M&C systems configured in the backend. Includes the auto-generated and the user defined parts of the tree definition

Parameters: none

Expected Payload: none

Replied Payload: A TreeViewNode document containing the complete tree view tree, starting with the ROOT node of the tree.

Return Codes:

Code	Description
200	OK, the requested TreeViewNode document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/treeview/{mncName}

Gets the tree definition for a given M&C system. Includes the user defined part of the tree definition as well as the device list for this M&C system.

Parameters:

Parameter	Description

Parameter	Description
mncName	The name of the M&C for which the tree definition shall be read.

Expected Payload: none

Replied Payload: A TreeViewNode document containing the tree branch for the given M&C, starting with the MNCNODE node for this M&C.

Return Codes:

Code	Description
200	OK, the requested TreeViewNode document is returned.
404	The requested M&C system does not exist or is actually offline. In this case an ApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/treeview/{mncName}/{subsystemName}

Gets the tree definition for a given M&C subsystem.

Parameters:

Parameter	Description
mncName	The name of the M&C for which the tree definition shall be read.
subsystemName	The name of the subsystem to retrieve. This subsystem name also may address nested subsystems, e.g. `SUBSYS-1.SUB_SYS-2` denotes a subsystem `SUSBSYS-2` contained in `SUBSYS-1`

Expected Payload: none

Replied Payload: A TreeViewNode document containing the tree branch for the given M&C subsystem, starting with the SUBSYSTEM node for this M&C subsystem.

Return Codes:

Code	Description
200	OK, the requested TreeViewNode document is returned.

Code	Description
404	Either the requested M&C system does not exist or is actually offline or no subsystem with the given name exists in this M&C system. In this case an ApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

PUT /api/v1/treeview/{mncName}

Replaces the tree definition for a given M&C system with a new version supplied by the client. Echoes the tree definition for this M&C system after the request has been processed.

Parameters:

Parameter	Description
mncName	The name of the M&C for which the tree definition shall be replaced.

Expected Payload: A TreeViewNode document containing the user defined part tree branch for the given M&C, starting with the MNCNODE node for this M&C. Only the SUBSYSTEM children of this node are processed, if the TReeViewNode has a DEVICELIST child, this is ignored.

Replied Payload: A TreeViewNode document containing the tree branch for the given M&C, starting with the MNCNODE node for this M&C. This is the tree view branch after the request has been processed, it includes the DEVICELIST for this M&C.

Return Codes:

Code	Description
200	OK, the requested TreeViewNode document is returned.
404	The requested M&C system does not exist or is actually offline. In this case an ApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

5.2.37 /api/v1/debug

This API end point lets you send debug commands to the (already opened) debug terminal of a M&C.

Supported HTTP methods: POST, HEAD, OPTIONS

POST /api/v1/debug/{mncName}

Sends one line containing a debug command to the debug terminal of the given M&C. The websocket connection for the debug terminal must already be open to make the command work. Echoes the original command after the request has been processed.

Parameters:

Parameter	Description
mncName	The name of the M&C to send the debug command to.

Expected Payload: A DebugMessage document containing the line with the command to send.

Replied Payload: A copy of the DebugMessage which has been sent.

Return Codes:

Code	Description
200	OK, the echoed DebugMessage document is returned.
404	The requested M&C system does not exist, is actually offline or another user is already connected to the debug terminal at this M&C. In this case an ApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

5.2.38 /api/v1/redundancy

This API end point lets you control or interrogate the redundancy properties for redundant M&Cs.

Supported HTTP methods: GET, PUT, HEAD, OPTIONS

GET /api/v1/redundancy/{mncName}

Queries the state of the redundancy control logic for the given M&C

Parameters:

Parameter	Description
mncName	The name of the M&C to query.

Expected Payload: none.

Replied Payload: A RedundancyState object reporting the redundancy properties of the M&C.

Return Codes:

Code	Description
200	OK, a RedundancyState document is returned.
400	The requested M&C system is not configured as a redundant pair. In this case an ApiError document is returned, describing details of the error.
404	The requested M&C system does not exist. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/redundancy

Queries the state of the redundancy control logic for all redundant M&Cs.

Parameters: none.

Expected Payload: none.

Replied Payload: A RedundancySummary object reporting the redundancy properties of all redundant M&Cs.

Return Codes:

Code	Description
200	OK, a RedundancySummary document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

PUT /api/v1/redundancy/{mncName}

Sends a redundancy command to either switch a redundant M&C pair or to enable/disable the M&C redundancy.

Parameters:

Parameter	Description
mncName	The name of the M&C to send the redundancy command to.

Expected Payload: A RedundancyCmd document containing the command to send.

Replied Payload: A copy of the RedundancyCmd which has been sent.

Return Codes:

Code	Description
200	OK, the echoed RedundancyCmd document is returned.
404	The requested M&C system does not exist. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

5.2.39 /api/v1/restart

This API end point lets you send a restart command to a given M&C.

Supported HTTP methods: PUT, HEAD, OPTIONS

PUT /api/v1/restart/{mncName}

Sends a restart command to the given M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C to send the redundancy command to.

Expected Payload: A RestartCmd document with *cmd* set to 'RESTART'. With other values of *cmd* the restart command is ignored.

Replied Payload: A copy of the RestartCmd which has been sent.

Return Codes:

Code	Description
200	OK, the echoed RestartCmd document is returned.
400	Either the RestartCmd passed to the call did not contain cmd=RESTART or the requested M&C system is offline and cannot be restarted. In this case an ApiError document is returned, describing details of the error.

Code	Description
404	The requested M&C system does not exist. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

5.2.40 /api/v1/devicesetup

This API end point lets read and write the device setup for a specific M&C.

Writing a device setup as a whole to the M&C requires a M&C restart to make the M&C read and interpret the changed setup. Hence this is to be considered a legacy operation which will be replaced by API calls to modify the device configuration of the M&C 'on the fly' without restarting it, as soon as these API calls are available.

Supported HTTP methods: GET, PUT, HEAD, OPTIONS

GET /api/v1/devicesetup/{mncName}

Get the device setup for the given M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C to read the device setup from.

Expected Payload: none.

Replied Payload: A DeviceSetup document describing the device configuration of the given M&C.

Return Codes:

Code	Description
200	OK, a DeviceSetup document is returned.
404	The requested M&C system does not exist. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

PUT /api/v1/devicesetup/{mncName}

Sends a new device configuration to the given M&C. The configuration will be saved as 'vlc.setup' on this M&C, replacing the existing configuration. The M&C will use this configuration after the next restart (see /api/v1/restart).

Parameters:

Parameter	Description
mncName	The name of the M&C to send the device setup to.

Expected Payload: The DeviceSetup document containing the new / modified device configuration.

Replied Payload: Echoes back the DeviceSetup which has been sent.

Return Codes:

Code	Description
200	OK, the echoed DeviceSetup document is returned.
404	The requested M&C system does not exist. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

5.2.41 /api/v1/drivers

This API end point returns a list of the available device drivers for a specific M&C.

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/v1/drivers

Get the list of the available device drivers. The list is fetched from the primary M&C.

Parameters: none

Expected Payload: none.

Replied Payload: An ItemList document containing the names of all device drivers available.

Return Codes:

Code	Description
200	OK, an ItemList document is returned.

Code	Description
400	The primary M&C is offline, cannot be queried. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

5.2.42 /api/v1/protocols

This API end point returns a list of the available device communication protocols. The list is fetched from the primary M&C.

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/v1/protocols

Get the list of all available device communication protocols. The list is fetched from the primary M&C.

Parameters: none.

Expected Payload: none.

Replied Payload: An ItemList document containing the names of all device communication protocols available.

Return Codes:

Code	Description
200	OK, an ItemList document is returned.
400	The primary M&C is offline, cannot be queried. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

5.2.43 /api/v1/preferredprotocols

This API end point returns a list of the preferred device communication protocols a particular device driver uses. The list is fetched from the primary M&C.

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/v1/preferredprotocols/{driverName}

Get the list of the preferred device communication protocols a particular device driver uses.

Actually a ***sat-nms*** devices driver defines exactly one preferred protocol, hence this list will always have exactly one element. Future ***sat-nms*** versions may permit to define multiple preferred protocols for a driver.

Parameters:

Parameter	Description
driverName	The name of the device driver to check.

Expected Payload: none.

Replied Payload: An ItemList document containing the names of the preferred device communication protocols of the given driver.

Return Codes:

Code	Description
200	OK, an ItemList document is returned.
400	The primary M&C is offline, cannot be queried. In this case anApiError document is returned, describing details of the error.
404	The device driver this call refers to does not exist. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

5.2.44 /api/v1/filerecordings

This API end point returns the recorded data for a File-Recorder device

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/v1/filerecordings/{mncName}/{deviceName}

Reads the recorded data from the given File-Recorder device

Parameters:

Parameter	Description
mncName	The name of the M&C to read the data from
deviceName	The name of the File-Recorder device to read the data from

Expected Payload: none.

Replied Payload: The recorded data in CSV format (text/plain). Columns are separated by comma characters, the first column contains the time stamp in 'YYYY-MM-DD HH:MM:SS' format (UTC). All other columns contain the recorded data. Example:

```
2023-02-16 09:12:38,28.3,,,
2023-02-16 09:13:39,28.4,,,
2023-02-16 09:14:39,28.5,,,
2023-02-16 09:15:38,28.4,,,
2023-02-16 09:16:38,28.5,,,
2023-02-16 09:17:38,28.3,,,
2023-02-16 09:18:38,28.4,,,
2023-02-16 09:19:38,28.4,,,
```

Return Codes:

Code	Description
200	OK, the requested file recorder data is returned.
400	The M&C is offline, cannot be queried. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	Either the referred M&C or the device does not exists or the device is not a File-Recorder device. In this case an ApiError document is returned, describing details of the error.

5.2.45 /api/v1/filerecordersettings

Reads and stores the diagram settings for a given file recorder device.

Supported HTTP methods: GET, PUT, HEAD, OPTIONS

GET /api/v1/filerecordersettings/{mncName}/{deviceName}

Reads the diagram settings for a given file recorder device.

Parameters:

Parameter	Description
mncName	The name of the M&C to read the data from
deviceName	The name of the File-Recorder device to read the data from

Expected Payload: none.

Replied Payload: The diagram settings as a FRViewProperties document.

Return Codes:

Code	Description
200	OK, the requested file recorder data is returned.
400	The M&C is offline, cannot be queried. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	Either the referred M&C or the device does not exists or the device is not a File-Recorder device. In this case an ApiError document is returned, describing details of the error.

PUT /api/v1/filerecordersettings/{mncName}/{deviceName}

Stores the diagram settings for a given file recorder device.

Parameters:

Parameter	Description
mncName	The name of the M&C to store the data to
deviceName	The name of the File-Recorder device to store the data for

Expected Payload: The diagram settings to store as a FRViewProperties document.

Replied Payload: Echoes back the stored data (after storing it at the destination M&C and reading it back from there).

Return Codes:

Code	Description
200	OK, the echoed back diagram settings document is returned.
400	The M&C is offline, data cannot be stored. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

Code	Description
404	Either the referred M&C or the device does not exists or the device is not a File-Recorder device. In this case an ApiError document is returned, describing details of the error.

5.2.46 /api/v1/acutargets

Reads the targets / satellites stored on a given ACU.

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/v1/acutargets/{mncName}/{deviceName}

Reads the targets / satellites stored on a given ACU.

Parameters:

Parameter	Description
mncName	The name of the M&C to read the data from
deviceName	The name of the device to read the data from. This may be either the name of the ACU device itself or the name of the Antenna-Management device controlling the ACU

Expected Payload: none.

Replied Payload: The target list of the queries ACU as a AcuTargetList document.

Return Codes:

Code	Description
200	OK, the requested file recorder data is returned.
400	The M&C is offline, cannot be queried. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	Either the referred M&C or the device does not exists or the device does not provide a target list. In this case an ApiError document is returned, describing details of the error.

5.2.47 /api/v1/inventory

This API endpoint is used to inspect or modify the backend's inventory of items / devices.

Supported HTTP methods: GET, POST, PATCH, HEAD, OPTIONS

GET /api/v1/inventory/{itemId}

Gets the inventory item with the given ID.

Parameters:

Parameter	Description
itemId	The itemId of the item to get

Expected Payload: none.

Replied Payload: An InventoryItem record with the given ID

Return Codes:

Code	Description
200	OK, the requested InventoryItem document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No InventoryItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/inventory/{itemId}/devices

Gets the list of InventoryDevice objects mapped to the given inventory item.

Parameters:

Parameter	Description
itemId	The itemId of the item to get

Expected Payload: none.

Replied Payload: An array of InventoryDevice records mapped to the item with the given ID

Return Codes:

Code	Description
200	OK, the requested InventoryDevice Array is returned.

Code	Description
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No InventoryItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/inventory/{itemId}/devices/{deviceId}

Gets the InventoryDevice objejct with the given ID combination.

Parameters:

Parameter	Description
itemId	The itemId of the item to get
deviceId	The deviceId of the device to get

Expected Payload: none.

Replied Payload: An InventoryDevice record with the given ID combination.

Return Codes:

Code	Description
200	OK, the requested InventoryDevice document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No InventoryItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/inventory

Creates a new InventoryItem entry in the database.

Parameters: none.

Expected Payload:An InventoryItem record containing the fields of the record to create. The following rules apply:

- Fields which are null or missing are set to 'null' in the record.
- The field 'itemId' is ignored. The POST command creates a new record, the database

assigns a unique itemId to the created record.
- The 'modified' field is set to the time when the command is executed, a 'modified' time specified in the supplied data is ignored.

Replied Payload: The created InventoryItem record

Return Codes:

Code	Description
200	OK, the modified InventoryItem document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No InventoryItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

POST /api/v1/inventory/{itemId}/devices

Creates a new InventoryDevice entry in the database and attaches this to the given item.

Parameters:

Parameter	Description
itemId	The itemId of the item to add the new device to

Expected Payload:An InventoryDevice record containing the fields of the record to create. The following rules apply:

- Fields which are null or missing are set to 'null' in the record.
- The field 'deviceId' is ignored. The POST command creates a new record, the database assigns a unique deviceId to the created record.
- The field 'itemId' is ignored. The POST command always takes the the 'itemId' from the URL and sets this in the created record.
- The 'modified' field is set to the time when the command is executed, a 'modified' time specified in the supplied data is ignored.

Replied Payload: The created InventoryItem record

Return Codes:

Code	Description

Code	Description
200	OK, the modified InventoryDevice document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No InventoryItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

PATCH /api/v1/inventory/{itemId}

Modifies the inventory item with the given ID.

Parameters:

Parameter	Description
itemId	The itemId of the item to modify

Expected Payload:An InventoryItem record containing the fields to modify. The following rules apply:

- Fields which are null or missing are not changed in the record.
- To set a field to 'null' set it to an empty string.
- The field 'itemId' is ignored. The PATCH command modifies the record specified in the URI, the 'itemId' ofthis record never changes.
- The 'modified' field is set to the time when the command is executed, a 'modified' time specified in the supplied data is ignored.

Replied Payload: The InventoryItem record after modification

Return Codes:

Code	Description
200	OK, the modified InventoryItem document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No InventoryItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.

Code	Description
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

PATCH /api/v1/inventory/{itemId}/devices/{deviceId}

Modifies the InventoryDevice object with the given ID combination.

Parameters:

Parameter	Description
itemId	The itemId of the item to modify
deviceId	The deviceId of the device to modify

Expected Payload:An InventoryDevice record containing the fields to modify. The following rules apply:

- Fields which are null or missing are not changed in the record.
- To set a field to 'null' set it to an empty string.
- The field 'deviceId' is ignored. The PATCH command modifies the record specified in the URI, the 'deviceId' of this record never changes.
- The 'modified' field is set to the time when the command is executed, a 'modified' time specified in the supplied data is ignored.

Replied Payload: The InventoryDevice record after modification

Return Codes:

Code	Description
200	OK, the modified InventoryDevice document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No InventoryItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/inventory/{itemId}

Deletes the inventory item with the given ID. As a side effect, allInventoryDevice and InventoryLogEntry objects referring to this inventory item are deleted as well.

Parameters:

Parameter	Description
itemId	The itemId of the item to delete

Expected Payload: none.

Replied Payload: none.

Return Codes:

Code	Description
204	OK, the addressed InventoryItem has been deleted.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No InventoryItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/inventory/{itemId}/devices/{deviceId}

Deletes the InventoryDevice obkect with the given ID combination.

Parameters:

Parameter	Description
itemId	The itemId of the item to delete
deviceId	The deviceId of the device to delete

Expected Payload: none.

Replied Payload: none.

Return Codes:

Code	Description
204	OK, the addressed InventoryDevice has been deleted.

Code	Description
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No InventoryItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

POST /api/v1/inventory/list

Gets a list of inventory items matching the filter information sent with the request.

Parameters: none

Expected Payload: An InventoryItem document containing the filter to be applied. Any fields in the document which are missing or null are intepreted as a wildcard. Specified fields must *all* match to include the item in the returned list. If a 'modified' timestamp is given in the filter description, it is interpreted as 'must be modified at this time or later'.

Replied Payload: An array ofInventoryItem records fulfilling the filter spec. The list is sorted by modification time, oldest first. If no records match the filter, an empty list is returned.

Return Codes:

Code	Description
200	OK, the requested InventoryItem array is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	The backend does not support the inventory database. In this case anApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

GET /api/v1/inventory/log/{entryId}

Gets the inventory log entry with the given ID.

Parameters:

Parameter	Description

Parameter	Description
entryId	The entryId of the log entry to get

Expected Payload: none.

Replied Payload: An InventoryLogEntry record with the given ID

Return Codes:

Code	Description
200	OK, the requested InventoryLogEntry document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No InventoryLogEntry with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/inventory/log

Adds a new InventoryLogEntry to the database.

Parameters: none.

Expected Payload:An InventoryLogEntry record containing the fields of the record to create. The following rules apply:

- log entries cannot be modified / rewritten. All fields must be supplied with their desired values.
- The field 'entryId' is ignored. The POST command creates a new record, the database assigns a unique itemId to the created record.
- The 'created' field is set to the time when the command is executed, a 'created' time specified in the supplied data is ignored.

Replied Payload: The created InventoryLogEntry record

Return Codes:

Code	Description
200	OK, the modified InventoryLogEntry document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

Code	Description
404	No InventoryLogEntry with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

POST /api/v1/inventory/log/list

Gets a list of inventory log entries matching the filter information sent with the request.

Parameters: none

Expected Payload: An InventoryLogEntry document containing the filter to be applied. Any fields in the document which are missing or null are intepreted as a wildcard. Specified fields must *all* match to include the entry in the returned list. If a 'modified' timestamp is given in the filter description, it is interpreted as 'must be created at this time or later'.

Replied Payload: An array ofInventoryLogEntry records fulfilling the filter spec. The list is sorted by creation time, oldest first.

Return Codes:

Code	Description
200	OK, the requested InventoryLogEntry array is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	The backend does not support the inventory database. In this case anApiError document is returned, describing details of the error.

5.2.48 /api/v1/documents

This API endpoint is used to inspect or modify the backend's list of documents.

Supported HTTP methods: GET, POST, PATCH, HEAD, OPTIONS

GET /api/v1/documents/{documentId}

Gets the documents item with the given ID.

Parameters:

Parameter	Description

Parameter	Description
documentId	The documentId of the item to get

Expected Payload: none.

Replied Payload: An DocumentItem record with the given ID. The DocumentItem does not contain its 'fileContent' field, use the '/api/v1/documents/content/' API call to retrieve the document content.

Return Codes:

Code	Description
200	OK, the requested DocumentItem document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No DocumentItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/documents

Creates a new DocumentItem entry in the database.

Parameters: none.

Expected Payload:An DocumentItem record containing the fields of the record to create. The following rules apply:

- Fields which are null or missing are set to 'null' in the record.
- The field 'documentId' is ignored. The POST command creates a new record, the database assigns a unique documentId to the created record.
- The 'modified' field is set to the time when the command is executed, a 'modified' time specified in the supplied data is ignored.

Replied Payload: The created DocumentItem record. The DocumentItem does not contain its 'fileContent' field, use the '/api/v1/documents/content/' API call to retrieve the document content.

Return Codes:

Code	Description
200	OK, the modified DocumentItem document is returned.

Code	Description
400	The file already exists. In this case the file is not written and anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No DocumentItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

PATCH /api/v1/documents/{documentId}

Modifies the documents item with the given ID.

Parameters:

Parameter	Description
documentId	The documentId of the item to modify

Expected Payload:An DocumentItem record containing the fields to modify. The following rules apply:

- Fields which are null or missing are not changed in the record.
- To set a field to 'null' set it to an empty string.
- The field 'documentId' is ignored. The PATCH command modifies the record specified in the URI, the 'documentId' of this record never changes.
- The 'modified' field is set to the time when the command is executed, a 'modified' time specified in the supplied data is ignored.
- If the field 'fileContent' is contained, the stored file gets rewritten with this content.
- If 'fileName' and 'fileContent' are changed at the same time, the old file is deleted before writing the new one.

Replied Payload: The DocumentItem record after modification. The DocumentItem does not contain its 'fileContent' field, use the '/api/v1/documents/content/' API call to retrieve the document content.

Return Codes:

Code	Description
200	OK, the modified DocumentItem document is returned.

Code	Description
400	A file with the new name already exists. In this case the file is not written and an ApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No DocumentItem with this ID exists or the backend does not support the documents database. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/documents/{documentId}

Deletes the documents item with the given ID. This also deletes the file this record refers to.

Parameters:

Parameter	Description
documentId	The documentId of the item to delete

Expected Payload: none.

Replied Payload: none.

Return Codes:

Code	Description
204	OK, the addressed DocumentItem has been deleted.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No DocumentItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

GET /api/v1/documents/content/{documentId}

Gets the file content of the document item with the given ID.

Parameters:

Parameter	Description
documentId	The documentId of the item to get

Expected Payload: none.

Replied Payload: The file content of the data file this document record refers to.

Return Codes:

Code	Description
200	OK, the requested file is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No DocumentItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/documents/list

Gets a list of documents items matching the filter information sent with the request.

Parameters: none

Expected Payload: An DocumentItem document containing the filter to be applied. Any fields in the document which are missing or null are intepreted as a wildcard. Specified fields must *all* match to include the item in the returned list. If a 'modified' timestamp is given in the filter description, it is interpreted as 'must be modified at this time or later'. The 'fileContent' field is ignored as a filter.

Replied Payload: An array of DocumentItem records fulfilling the filter spec. The list is sorted by file name, 'a' to 'z'. The DocumentItem records do not contain their 'fileContent' field, this API call is intended to list the items, not to to execute a bulk read of the documents.

Return Codes:

Code	Description
200	OK, the requested DocumentItem array is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

Code	Description
404	The backend does not support the inventory database. In this case anApiError document is returned, describing details of the error.

5.2.49 /api/v1/thumbnail

This API call fetches a thumnail image from the backend. Thumbnail images are cached in the backend. If you subscribe for a thumbnail image via the websocket, the frontend only receives a messages which tells that a new thumbnail picture is available. The image itself then must be fetched with the `/api/v1/thumbnail` API call.

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/v1/thumbnail/{messageId}

Gets the thumbnail image for the given message ID (e.g. 'MNC-1.IRD-3.thumbnail') Please note, that the messageId must be subscribed or the API call will return a 404 Not Found because the backens has not subscribes for the thumbnail message at the M&C

Parameters:

Parameter	Description
messageId	The full message ID of the thumbnail parameter of the device

Expected Payload: none.

Replied Payload: An Thumbnail document containing the thumbnail image.

Return Codes:

Code	Description
200	OK, the requested Thumbnail document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	Not found. In this case an ApiError document is returned, describing details of the error.

5.2.50 /api/v1/backendinfo

This API endpoint gives the backend information for example name and role.

Supported HTTP methods: GET

GET /api/v1/backendinfo

Returns the BackendInfo data objects.

Parameters: none

Expected Payload: none

Replied Payload: BackendInfo data objects

Return Codes:

Code	Description
200	OK, returns a JSON document.
401	not logged in. In this case an ApiError document is returned,describing details of the error.

5.2.51 /api/v1/globalSettings

This API endpoint is used to get and or save global settings for the users.

Supported HTTP methods: GET, PATCH, HEAD, OPTIONS

GET /api/v1/globalSettings

Gets the saved settings if not Backend should retrun default values with setting keys.

Parameters: none

Expected Payload: none

Replied Payload: The list of stored and default settings as an array ofUISettings data objects.

Return Codes:

Code	Description
200	OK, returns a JSON document.
401	not logged in. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/globalSettings

Save setting in the Database. If *setting key* already exists it will overwrite the value.

Parameters: none

Expected Payload: The UISettings data object to store.

Replied Payload: The list of stored and default settings as an array ofUISettings data objects.

Return Codes:

Code	Description
201	OK, returns a JSON document.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/globalSettings/{id}

Deletes the global settings with the given id.

Parameters:

Parameter	Description
id	The id of the settings to delete

Expected Payload: none.

Replied Payload: none.

Return Codes:

Code	Description
202	ACCEPTED, Return Object of UISettings.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No user setting with this username exists . In this case anApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

5.2.52 /api/v1/userSettings

This API endpoint is used to save or get user settings for the users.

Supported HTTP methods: GET, PATCH, HEAD, OPTIONS

GET /api/v1/userSettings

Gets the saved settings in data base if not Backend should retrun default values with setting keys.

Parameters: none

Expected Payload: none.

Replied Payload: The list of stored and default settings as an array ofUISettings data objects.

Return Codes:

Code	Description
200	OK, returns a JSON document.
401	not logged in. In this case anApiError document is returned,describing details of the error.

PATCH /api/v1/userSettings

Save user setting in the Database.

Parameters: NONE

Expected Payload: The UISettings data object to store.

Replied Payload: The list of stored and default settings as an array ofUISettings data objects.

Return Codes:

Code	Description
201	OK, returns a JSON document with all user settings.
401	not logged in. In this case anApiError document is returned,describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/userSettings/{id}

Deletes the user settings with the given id.

Parameters:

Parameter	Description

Parameter	Description
id	The id of the settings to delete

Expected Payload: none.

Replied Payload: none.

Return Codes:

Code	Description
202	ACCEPTED, Return Object of UISettings.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No user setting with this username exists . In this case anApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/userSettings

Deletes the user settings with username. Username is take from token by the backend.

Parameters: none

Expected Payload: none.

Replied Payload: none.

Return Codes:

Code	Description
202	ACCEPTED, Return list of UISettings array list.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No user setting with this username exists . In this case anApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

5.3 WebSocket Communication

The API provides a websocket at /msgs to update volatile data asynchronously. The data to be sent over the websocket is controlled with a subscription model. With the sat-nms Software, everything is about device parameters: Every device parameter has a unique ID, changes of this parameters are transported as messages having the id and the value of this parameter as properties. The websocket is used one-way: There are only messages sent from the backend server to the front end. The /api/v1/subscribe API call controls which messages / parameters shall be send over the websocket.

5.3.1 Creating WebSockets

Creating a websocket id a two-step procedure: First a websocket ID must be obtained. This is done using the /api/v1/makewebsocket API call. This call prepares a websocket which can be connected to. It returns an ID for the newly prepared websocket. The client must remember this ID during the session, it is required for all subsequent operations on this websocket.

The second step is to connect to this websocket. The URL for these websockets is**/msgs**, all connections share this. To identify which websocket to open, a query parameter defining the websocket ID must added to the HTTP URI when opening the websocket:

```
/msgs&satnmsWebsocketId=XXXXX
```

XXXXX has to be replaced by the 5-character websocket ID obtained with the /api/v1/makewebsocket API call.

The now connected websocket does not receive any data.

5.3.2 Subscribing For Messages

To receive data over the websocket, the data must be subscribed for. With the sat-nms Software, each device parameter and each type/range description has its unique message ID. For each message ID an explicit subscription must be done to get this parameter and its updates. The /api/v1/subscribe API call does this, either for a single message ID or for a list of IDs. Subscriptions are made for a certain websocket instance, hence the ID of the websocket must be supplied with the subscribe call.

Subscribed messages are sent asynchronously, each time when a change of the parameter is detected. It is not guaranteed, that a subscribe results immediately in a message sent over the websocket. Parameters may be in the state "unknown", specially in the first minutes after starting a M&C. Such a parameter will be sent when the first value becomes available.

The /api/v1/subscribe API call is also is used to unsubscribe messages using the DELETE HTTP method. It is recommended to unsubscribe messages for which there are no more updates needed.

Closing the websocket automatically unsubscribes all pending subscriptions. There is no need to do this explicitly before closing the session.

5.3.3 Interpreting WebSocket Data

The websocket transports text messages containing single JSON data objects. These objects all are Message objects. Parsing and interpreting the Message objects can be done along the*type* field in the message. The value of this field determines the data type and further content of the message:

Token	Message Type
VALUE	Parameter, contains an update for a parameter value
RANGE	Range, contains a data type an valid range definition for a parameter
PONG	This could be an echoed message used by the front end as a heartbeat (see Echo / Heartbeat Function below)

Range messages contain an additional *dataType* field which specifies the data type of the parameter belonging to this range definition. Depending on *dataType*, there will me more fields in the range message, specifying the valid range of a numeric parameter, the available choices for an enumeration and more. The table below list all defined data types:

Token	Data Type
INTEGER	IntegerRange
HEX	HexRange
FLOAT	DoubleRange
BOOLEAN	BooleanRange
ALARM	AlarmFlagRange
CHOICE	EnumRange
OBJECT	ObjectRange

5.3.4 Echo / Heartbeat Function

"Ping" messages on websocket protocol level sent to the websocket are answered with a "Pong" message. This can be used to implement a heartbeat mechanism at protocol level.

At application level this is possible as well. The backend's websocket implementation echoes all text messages received from the front end back to it. It is recommended to use Message data objects for this and setting *type* to a value not used by the backend.

5.3.5 Closing Websockets

When a websocket is closed, the backend server deletes all subscriptions made for this websocket. The websocket cannot be re-opened, the websocket ID obtained for the closed websocket becomes invalid.

When there is no websocket opened in a session for 120 seconds, the backend server deletes the session, this is the way the backend server handles the case when the operator closes the browser window. *(not yet implemented)*

5.4 Debug Terminal Communication

The backend provides a relay mechanism to access the debug terminal at each M&C connected to the backend. While sending commands to the debug terminal is done with an API call, the messages showing in the terminal are sent to the front end through a websocket. The following paragraphs give an overview about this.

5.4.1 Opening A Debug Terminal

To open a debug terminal, the front end opens a websocket at the URI defined for this. It receives every line to be shown in the terminal window as one message on the websocket. The M&C allows only one client to connect to the debug terminal at a time, opening the websocket may fail if either the M&C is down or if another user is already connected to the debug terminal (either from the Web-UI or from a Java client).

The URI to connect to the debug terminal is `/debug&mncName=XXXXX`

Where XXXXX has to be replaced with the name of the M&C to connect to. Once opened, the websocket transports every line from the debug terminal as a DebugMessage data object to the front end.

5.4.2 Closing A Debug Terminal

While the websocket for a debug terminal is opened, this prevents other users from using this feature. Hence every debug terminal websocket should be closed by the front end as soon as it is no longer needed. Closing the websocket also closes the data connection the M&C debug terminal port and by this frees this resource.

5.4.3 Sending Debug Commands

Sending commands to the debug terminal is done with the/api/v1/debug API call. This also takes a DebugMessage data object payload, carrying the line to be entered at the debug terminal. /api/v1/debug API calls require a debug terminal websocket to the referenced M&C being already opened. The will fail if the websocket for this M&C is not open because this means that also the data connection to the M&C is not present.

5.4.4 M&C Redundancy Management

The backend provides a redundancy switching mechanism for M&C systems connected to it. A redundant pair of M&C systems share the M&C name, but the M&C systems have different IP addresses. The redundancy works asymmetrically (there is one primary and one backup M&C) and one-way (redundancy switch will never happen from backup to primary).

If the communication to the primary M&C breaks and cannot be re-established for some time, the backend stops the M&C service on the primary machine and starts it on the backup machine. From now on the backend connects to the IP of backup M&C to control it.

All configuration about this is done in the 'backend.properties' file. The second IP address for the backup M&C and parameters how long the backend shall retry to connect to the primary M&C before it does the switch over to the backup M&C.

The actual state of the redundancy for each M&C (pair) is reported through the/api/v1/mnclist API call. Each MncInfo record in the returned list defines a 'redundancy' field which tells about the actual state.

5.4.4.1 Redundancy Commands

While the general aspects of the redundancy are statically configured in the 'backend.properties' file, the backend provides a /api/v1/redundancy API call to control if the primary or the backup machine of a redundant pair shall be used. There are two commands which may be sent with this API call.

command	description
RESET	tells the backend to switch back to primary M&C. If the backend is already connected to the primary M&C, or if no redundancy is configured for this M&C, the 'RESET' command has no effect. If the backend cannot connect to the primary M&C for some time, it tries to switch over to the backup machine.
FORCE	forces a switch over to the backup M&C - even if the communication with the primary one works fine. Has no effect if no redundancy is configured or if the backend is already connected to the backup machine.

5.5 Data Models

This section describes the data models used as payload with the API calls mentioned above.

The data model descriptions appear in an order which groups related functions. An alphabetic list of all data models is listed below

- ActiveBackend
- AcuTarget
- AcuTargetList
- AlarmFlagRange
- ApiError
- ArrowElement
- AzElElement
- BackendInfo
- StreamKeyData
- BooleanRange

- ButtonElement
- ChannelData
- ChartElement
- ColorDefinition
- DatabaseVersion
- DebugMessage
- DeviceDefinition
- DeviceElement
- DeviceFrameDefinition
- DeviceFrameTab
- DevicePreset
- DeviceSetup
- DeviceThreadDefinition
- Dictionary
- DisplayElement
- DocumentItem
- DocumentList
- DoubleRange
- EditButtonElement
- EnumRange
- EventAck
- EventInfo
- EventLog
- EventReportRequest
- EventReport
- FaultElement
- FaultInfo
- FaultList
- FrameElement
- FRViewProperties
- FRViewTraceDescription
- FRViewPreset
- FRViewTraceScaling
- GaugeElement
- HexRange
- IconElement
- IntegerRange
- InventoryDevice
- InventoryItem
- InventoryLogEntry
- ItemList
- KeyValPair
- LatchingButtonElement
- LockButtonElement
- MCPElement
- Macro
- Message

- MncInfo
- MncList
- ObjectRange
- ParameterButtonElement
- ParameterElement
- Parameter
- PresetValue
- PresetVars
- DeviceVars
- RadioButtonElement
- Range
- RectElement
- RedundancyCmd
- RedundancyState
- RedundancySummary
- RestartCmd
- SatDbAntenna
- SatDbBeacon
- SatDbBeaconAttenuation
- SatDbI11Data
- SatDbPosition
- SatDbSatDetails
- SatDbSatOperator
- SatDbSatellite
- SatDbTLEData
- SatDbTableTracking
- SatDbTc
- SatDbTcAttenuation
- SatDbTleImport
- SatDbTm
- SatDbTmAttenuation
- SatDbState
- ScheduleEvent
- Schedule
- ScreenDefinition
- ScreenElement
- SpectrumElement
- SpectrumTrace
- StringRange
- SubscribeList
- SwitchElement
- TakeButtonElement
- TargetListElement
- TextElement
- Thumbnail
- TokenReply
- TrackingHistory

- TrackingPoint
- TreeViewNode
- UISettings
- UserList
- UserProperties
- WebsocketId
- XYChartElement

5.5.1 TokenReply

The TokenReply data model contains the data returned from a successful login or token refresh (API end point /token). The fields defined in this data model follow the OAuth2 recommendations.

Data Model

Key	Type	Value
access_token	String	The access token (JWT) to be added as 'Authorization: Bearer ...' header field with subsequent API calls.
refresh_token	String	The refresh token (JWT) to be used to refresh an expired access token with the /token API end point and *grant_type* set to 'refresh_token'.
token_type	String	Always 'JWT'
expires	Number	The number of seconds the access_token is valid

5.5.2 UserProperties

The UserProperties document contains some information about the user currently logged in. It is returned as a reply to the GET /api/v1/user call.

Data Model

Key	Type	Value
userName	String	The user (login) name.
fullName	String	The user's full name.
password	String	The user's password (encrypted).
groupList	String	A comma separated list of group identifiers this user is allowed to access.
privilege	Number	The user's privilege level (1..150)

Remarks

- The user's privilege level is very important for the front end as it specifies which user interface elements may be accessed by the user. With the sat-nms WebGUI the privilege level mechanism replaces the commonly used 'roles'. This works the following way:
    - Each active user interface element like an entry field, a drop down box has a privilege level defined. Predefined screens like standard device screens have these level set to standard values, with user defined screens each user interface element may be configured to an individual privilege level.
    - If the privilege level of the user is higher or equal to the level of the user interface element, he may change the parameter controlled by this element.
    - If the privilege level of the user is lower, he gets the actual value of the parameter displayed, but he may not change this value.
- The privilege level has a range from 1 to 150, commonly used values are 100 for standard operator access and 150 for admin privileges.

5.5.3 UserList

The UserList document contains the complete list of users known to the backend. It is returned as a reply to the GET /api/v1/users call.

Data Model

Key	**Type**	**Value**
users	Array	Contains the list of known users as an array ofUserProperties objects

5.5.4 ApiError

The backend server returns an ApiError document if something went wrong, i.e. The HTTP return code is not in the 200 range.

Data Model

Key	**Type**	**Value**
timestamp	String	The time and date when the event happened.
status	Number	The HTTP status code returned.
error	String	The textual description of *status*.
message	String	A message describing the reason of the error.
path	String	The path part of the URL which caused the error.

Remarks

- The *timestamp* has a format like `2020-01-16T09:40:48.079+0000`

- The fields *error* and *message* contain the same text in most cases.

5.5.5 MncList

The MncList is used with the /api/v1/mnclist API call to report the list of M&C systems managed by the backend.

Data Model

Key	Type	Value
mncList	Array	contains an array of MncInfo objects, each describing the properties of one M&C system manged by the backend. The first entry of the list is called the primary M&C, it is used to get the login data and the live event log. This (and only this) M&C has "primary" set true.

5.5.6 MncInfo

The MncInfo data model contains the properties of one entry in a MncList. It is used with the /api/v1/mnclist API call to report the list of M&C systems managed by the backend.

Data Model

Key	Type	Value
id	String	The ID / name of the M&C. This ID is used to identify the M&C in messages. See section Message below for details how the M&C ID is embedded in the message ID.
ipAddress	String	The IP address of the M&C server as defined in the backend configuration file.
connected	Boolean	The actual state of the connection to the M&C.
primary	Boolean	True marks this M&C as the primary one.
redundancy	String	Describes the state of the redundancy control for this M&C. Possible values are 'NONE' = no redundancy control configured for this M&C 'PRIMARY' = the backend connects to the primary IP address for the M&C 'BACKUP' = the backend connects to the backup IP address for the M&C

5.5.7 ActiveBackend

The ActiveBackend data model carries the name of the backend which actively controls M&C redundancy switching in a redundant backend configuration. It is used with the /api/v1/activebackend API call.

Data Model

Key	Type	Value
activeBackendName	String	The name of the backend which is or shall be the active backend

The name of a backend is defined in the 'backend.properties' file with the 'backend.name' key.

5.5.8 SubscribeList

The SubscribeList is used with the /api/v1/subscribe API call to define a list of message identifiers the front end wants to subscribe or to unsubscribe from.

Data Model

Key	Type	Value
subscribeList	Array	contains an array of strings, each string is on message ID to be subscribed / unsubscribed.

5.5.9 WebsocketId

The WebsocketId carries the ID of a newly created websocket instance. It is returned by the /api/v1/makewebsocket API call in the response body.

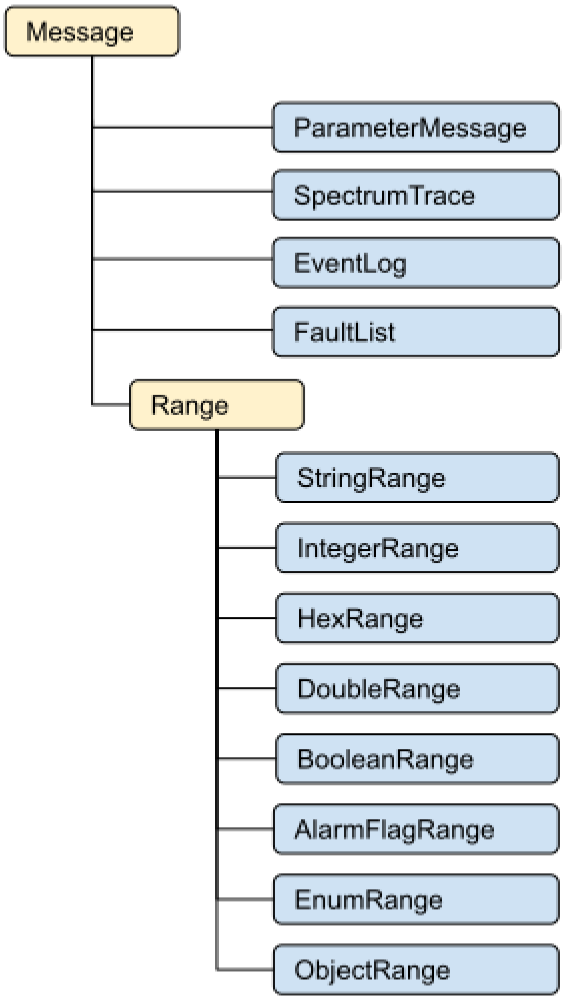
Data Model

Key	Type	Value
id	String	A newly created websocket ID. This is a unique 5-character string identifying the websocket for all further processing.

5.5.10 Message

With the sat-nms software, all volatile data gets exchanged as a Message. A message may carry a parameter like a value read from a device or a setting for this device. Beside this, data type and range information about parameters are exchanged as messages as well.

Because of this there are several flavors of messages defined, based of a hierarchical model derived from object oriented programming. The diagram below shows the known variants of Message as they depend on each other. Data models with yellow background do not exist in reality, they are defined as 'abstract base classes' which define the common parameters of their siblings.



Messages are used either with the /api/v1/peek and /api/v1/poke API calls or with the subscription model over a websocket connection.

The core property of a Message is its message ID. The message ID describes, which parameter of which device connected to which M&C server is addressed with the information of the message. The general format of a message ID is:

```
{M&C-Name}.{Device-Name}.{Parameter-Name}[.R]
```

The message ID is composed of at least three words with dots in between. The first word describes the M&C server to which the device is connected which is addressed by the second word. The third word identifies one parameter of this device, this identifier again may contain dots, if the parameter in the device are organized in a tree-like structure. Finally, the character sequence **.R** may be appended to the ID. This marks the message containing the range information about the parameter instead of the actual parameter value. Example:

```
MNC-1.DEVICE-3.parameter
```

Data Model

Key	Type	Value
id	String	The message ID associated with this message.
type	String	The message type (see below).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ' (UTC)

Defined Message Types

Token	Data Type
VALUE	A Parameter message.
RANGE	A Range message.
SPECTRUM	A SpectrumTrace message. This message may not be sent with a /api/v1/poke call, it is read-only from the front end's point of view.
EVENTLOG	An EventLog message. This message may not be sent with a /api/v1/poke call, it is read-only from the front end's point of view.
FAULTLIST	A FaultList message. This message may not be sent with a /api/v1/poke call, it is read-only from the front end's point of view.
PONG	Echo reply (PONG is only an example, the front end may send any message type other than *VALUE* or *RANGE*, the backend will echo the message as it is received.)

Message IDs of Range messages always end with **.R** - this is the way the underlying ***sat-nms*** M&C software marks these messages.

5.5.11 Parameter

A Parameter message carries a value to or from a device controlled by the M&C server. When sent from the UI to the M&C server, the Parameter message commands a new value to be set at the device. When sent from the M&C server to the UI, the message informs about a change of the parameter. This applies to read-only parameters like state flags or measurement values and to settings-type parameters as well.

Data Model (extends Message)

Key	Type	Value
id	String	The message ID (see Message)

Key	Type	Value
type	String	The message type (always **VALUE**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
val	String	The value of the parameter, coded as a String.

The value contained in a Parameter message (*val*) may be of different types. The Range of the parameter defines this type, strictly spoken the *dataType* field in the Range message. For simplicity, all data types are coded as a string value in the Parameter message. The table below specifies, in which way parameter values are coded for all defined *dataType* flavors:

Range *dataType*	**Coding in the *val* field**
TEXT	UTF8 coded String.
INTEGER	Decimal number, consisting of digits 0..9
HEX	Decimal (!) number, consisting of digits 0..9
FLOAT	Floating point number with decimal point
BOOLEAN	Either **true** or **false** (all lower case)
ALARM	Either **true** or **false** (all lower case)
CHOICE	The selected choice as a UTF8 coded string. Please note, case is significant with choices. The values **ON** and **On** are treated as different ones.
OBJECT	ObjectRange parameter are not supported with this version of the API. Writing an ObjectRange parameter to the M&C is ignored, when queries an ObjectRange parameter always reports an empty string in *val*.

5.5.12 Range

A Range message contains the type definition and an optional specification for a range/limit validation for numeric parameters. The data model definition for Range is abstract, which means it never will be used in practice, only the data models derived from this will be used in the API.

Data Model (extends Message)

Key	Type	Value
id	String	The message ID (see Message)

Key	Type	Value
type	String	The message type (always **RANGE**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
dataType	String	A token defining the data type this Range object describes. Depending on *dataType*, there are more parameters present in the Range definition.
disabled	Bool	True tells the front end that this parameter is currently not in use. No value shall be displayed for this parameter, even if a corresponding Parameter message has been received. Any decorations like frame or descriptive label shall be displayed grayed/faded in order to show this state to the operator.
readonly	Bool	This parameter is for display only, the operator may not change it.

Defined Data Types

Token	Data Type
TEXT	StringRange
INTEGER	IntegerRange
HEX	HexRange
FLOAT	DoubleRange
BOOLEAN	BooleanRange
ALARM	AlarmFlagRange
CHOICE	EnumRange
OBJECT	ObjectRange

Remark: Range objects are also used in the PresetVars list of device driver variables. In this case *messageId* does not contain a fully qualified message ID, but the name of the variable.

5.5.13 StringRange

A StringRange parameter carries a free text.

Data Model (extends Range)

Key	Type	Value
id	String	The message ID (see Message)

Key	Type	Value
type	String	The message data type (always **RANGE**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
dataType	String	Always **TEXT** (see Range)
disabled	Bool	Inherited from Range
readonly	Bool	Inherited from Range

5.5.14 IntegerRange

An IntegerRange parameter carries an integer number. Internally the value is represented by a signed 64-bit number (data type *long*).

Data Model (extends Range)

Key	Type	Value
id	String	The message ID (see Message)
type	String	The message data type (always **RANGE**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
dataType	String	Always **INTEGER** (see Range)
disabled	Bool	Inherited from Range
readonly	Bool	Inherited from Range
checkLimits	Boolean	true = the front end must validate any entry for this parameter to be in the range *min ... max*. Invalid entries must be rejected, no values shall be sent to the server if outside the given range.
min	Number	The minimum value to be accepted with *checkLimits*=true
max	Number	The maximum value to be accepted with *checkLimits*=true
unit	String	A unit string to be displayed right of the parameter value (e.g. "MHz" for a frequency)

5.5.15 HexRange

A HexRange parameter carries an integer number. Internally the value is represented by a signed 64-bit number (data type *long*). HexRange is much like IntegerRange with the only difference, that values are printed at the UI in hexadecimal notation (no prefix like '0x' or '#') and user entries are expected in the same format.

Data Model (extends Range)

Key	Type	Value
id	String	The message ID (see Message)
type	String	The message type (always **RANGE**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
dataType	String	Always **HEX** (see Range)
disabled	Bool	Inherited from Range
readonly	Bool	Inherited from Range
checkLimits	Boolean	true = the front end must validate any entry for this parameter to be in the range *min ... max*. Invalid entries must be rejected, no values shall be sent to the server if outside the given range.
min	Number	The minimum value to be accepted with *checkLimits*=true
max	Number	The maximum value to be accepted with *checkLimits*=true
unit	String	A unit string to be displayed right of the parameter value (e.g. "MHz" for a frequency)

5.5.16 DoubleRange

A DoubleRange parameter carries a floating point value. Internally floating point values are represented by 64-bit floating point values (data type *double*, that gave the name for the DoubleRange).

Data Model (extends Range)

Key	Type	Value
id	String	The message ID (see Message)
type	String	The message data type (always **RANGE**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
dataType	String	Always **FLOAT** (see Range)
disabled	Bool	Inherited from Range
readonly	Bool	Inherited from Range

Key	Type	Value
checkLimits	Boolean	true = the front end must validate any entry for this parameter to be in the range *min ... max*. Invalid entries must be rejected, no values shall be sent to the server if outside the given range.
min	Number	The minimum value to be accepted with *checkLimits*=true
max	Number	The maximum value to be accepted with *checkLimits*=true
prec	Number	The precision (number of digits following the decimal point). Details are given in the paragraph **Number Formatting** below
unit	String	A unit string to be displayed right of the parameter value (e.g. "MHz" for a frequency)

Number Formatting

FLOAT variables are displayed with a fixed precision. The *prec* parameter defines how many digits right of the decimal point shall be displayed. Values of *prec* above 100 define a scientific formatted floating point with *prec - 100* digits precision (e.g. '0.123E-2' for *prec=103*)

5.5.17 BooleanRange

A BooleanRange parameter carries a BOOL value. It may be one of the values*true* or *false*.

Data Model (extends Range)

Key	Type	Value
id	String	The message ID (see Message)
type	String	The message type (always **RANGE**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
dataType	String	Always **BOOL** (see Range)
disabled	Bool	Inherited from Range
readonly	Bool	Inherited from Range

5.5.18 AlarmFlagRange

The AlarmFlagRange describes a sat-nms ALARM parameter. Like the BOOL parameter may be *true* (alarm) or *false* (no alarm). The AlarmFlagRange defines a descriptive text for the alarm flag which can be displayed beside the actual state.

Data Model (extends Range)

Key	Type	Value
id	String	The message ID (see Message)
type	String	The message data type (always **RANGE**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
dataType	String	Always **ALARM** (see Range)
disabled	Bool	Inherited from Range
readonly	Bool	Inherited from Range (always *true* with alarm flags)
description	String	A short description of the ALARM

5.5.19 EnumRange

The EnumRange describes a parameter which' value can be one of a given set of values. An UI typically implements this parameter type as a drop down box or as a list box which allows a single selection.

Data Model (extends Range)

Key	Type	Value
id	String	The message ID (see Message)
type	String	The message type (always **RANGE**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
dataType	String	Always **CHOICE** (see Range)
disabled	Bool	Inherited from Range
readonly	Bool	Inherited from Range
choices	Array	An array of string containing the choices for this parameter

5.5.20 ObjectRange

An ObjectRange describes a parameter carrying binary data, e.g. A spectrum analyzer trace. Special software is required to decode the data contained.

Data Model (extends Range)

Key	Type	Value

Key	Type	Value
id	String	The message ID (see Message)
type	String	The message data type (always **RANGE**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
dataType	String	Always **OBJECT** (see Range)
disabled	Bool	Inherited from Range
readonly	Bool	Inherited from Range

5.5.21 SpectrumTrace

A SpectrumTrace message contains the spectrum trace data distributed by a CSM-Spectrum-Analyzer device. The spectrum trace is much like a read-only parameter (it may not be commanded to the M&C), instead of a single value it carries an array of numeric values.

The array elements represent the spectrum trace, all elements are integer numbers describing the point position in pixels from the top of the display downwards. The length of the trace depends on the particular spectrum analyzer model, common trace lengths are 501 or 601 points. The first array element is the leftmost point (at *centerFrequency-frequencySpan/2*), the last array element is the is the rightmost point (at *centerFrequency+frequencySpan/2*).

The point positions in the array (array index, element value) refer to the native size of the spectrum analyzer display as read out by the driver. This spectrum display size in pixels may be read from the spectrum analyzer device as the 'info.screen' variable. The spectrum analyzer widget on the screen may however re-scale the display to any size requested.

Data Model (extends Message)

Key	Type	Value
id	String	The message ID (see Message)
type	String	The message type (always **SPECTRUM**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
trace	Array of Number	The spectrum trace, as an array of integer values, describing the point position in pixels from the top of the display downwards.

5.5.22 TrackingPoint

A TrackingPoint message contains pointing data of an antenna tracking controller for one tracking step. An array of TrackingPoint elements makes up the TrackingHistory which is described below

Data Model

Key	Type	Value
time	String	A time stamp telling when this tracking point was recorded. The format follows ISO 8601, example: `2021-03-26T08:02:30Z`, time zone is always UTC.
apos	Number	The azimuth pointing of the antenna after the tracking step.
epos	Number	The elevation pointing of the antenna after the tracking step.
blev	Number	The beacon level after the tracking step.
asucc	Boolean	True = the peaking process of the azimuth axis was successful.
apeak	Number	The azimuth peak position evaluated during the tracking step. Only meaningful if *asucc* is true.
esucc	Boolean	True = the peaking process of the elevation axis was successful.
epeak	Number	The elevation peak position evaluated during the tracking step. Only meaningful if *esucc* is true.

5.5.23 TrackingHistory

A TrackingHistory message contains the tracking memory of an antenna tracking controller. Actually, only the SatService ACU provides this data. The TrackingHistory message is read-only, meaning it may not be commanded to the M&C.

Data Model

Key	Type	Value
trace	Array of TrackingPoint	The tracking memory. The tracking points are ordered by time in the array, the first element refers to the oldest tracking point.

5.5.24 EventLog

An EventLog message carries the up to 24 most recent events that occurred in one of the M&C systems managed by the backend. The backend collects the events from the M&C systems and combines them to this list. The backend updates this message whenever a new message is added to the log, but at a maximum rate of one message per second.

To subscribe for the EventLog message, a/api/v1/subscribe API call must be made for the message ID `{Primary-M&C-Name}.SYSTEM.eventLog` where {Primary-M&C-Name} is the name / ID of the M&C reported with the /api/v1/mnclist API call with the "primary" key set true.

Data Model (extends Message)

Key	Type	Value
id	String	The message ID. This is always the message ID described above. (see Message)
type	String	The message type (always **EVENTLOG**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
events	Array	An array of EventInfo objects representing the recent events which occurred. This array has not more than 24 elements, directly after system start the array may be shorter or even empty if no events have been collected yet.

5.5.25 EventReport

An EventReport contains a list of EventInfo objects. The EventReport is the reply of the backend for a /api/v1/eventreport call when sent as a POST.

Data Model

Key	Type	Value
events	Array	An array of EventInfo objects.

5.5.26 EventReportRequest

The EventReportRequest specifies the filters to be applied when generating an EventReport. It has to be sent with an /api/v1/eventreport POST call.

Data Model

Key	Type	Value
from	String	Specifies the start of the time range of the report. Must be formatted following ISO 8601, example: `2021-03-26T08:02:30Z` , time zone is always UTC.
to	String	Specifies the end of the time range of the report. Must be formatted following ISO 8601, example: `2021-03-26T08:02:30Z` , time zone is always UTC. May be *null* if *length* is not zero.
length	Number	The length of the time range of the report in hours. If this value is zero, the report length is determines as the difference *to - from*. If the value is not *null*, the *to* value is ignored and *length* is used as the report length.

Key	Type	Value
mnc	String	The M&C name filter. Only events containing this text in their *source1* (M&C name) field are included in the report. If*mnc* is *null*, all events pass this filter.
device	String	The device name filter. Only events containing this text in their *source2* (device name) field are included in the report. If*device* is *null*, all events pass this filter.
host	String	The host name filter. Only events containing this text in their *origin1* (host name) field are included in the report. If *host* is *null*, all events pass this filter. Please note that only parameter change events have a host name set, other events will never match this filter.
user	String	The user name filter. Only events containing this text in their *origin2* (user name) field are included in the report. If *user* is *null*, all events pass this filter. Please note that only parameter change events have a user name set, other events will never match this filter.
maxlen	Number	The maximum length of the report (number of event messages). If more events than this number match the filter settings, only the oldest *maxlen* events of these will be contained in the report. If maxlen is null, the server side default for the maximum report length is used.
pri	String	The event priority filter. Must contain one of the following: ALL = include all events (no filter) INFO = include the events with priority INFO or higher INFO-ONLY = include the events with exactly priority INFO WARNING = include the events with priority WARNING or higher WARNING-ONLY = include the events with exactly priority WARNING FAULT = include the events with priority FAULT or higher FAULT-ONLY = include the events with exactly priority FAULT
text	String	The message text filter. Only messages which contain the given text are included in the report. If *text* consists of multiple words separated by space characters, the message text must contain all these words in arbitrary sequence. If *text* is *null*, all events pass this filter.

5.5.27 EventAck

The EventAck tells the backend which events in the event log shall be acknowledged. It is sent with a POST call to /api/v1/eventack.

Data Model

Key	Type	Value
ackAll	Boolean	Selects if a single event shall be acknowledged (false) of if all pending events shall be acknowledged (true).
event	EventInfo	The event to be acknowledged. *event* is ignored if *ackAll* is true.

5.5.28 EventInfo

An EventInfo contains the properties of one event within anEventReport or an EventLog message.

Data Model

Key	Type	Value
id	Number	A unique numeric identifier for this record. The event id is required to acknowledge a particular event.
pri	Number	The event priority coded as a one digit number 0 = suppressed, the event shall not be shown 1 = informational 2 = fault 3 = alarm 4 = warning
nak	Boolean	*(needs to be acknowledged)* This reads true for unacknowledged fault events.
msg	String	The event message text.
t1	String	*(time1)* A time stamp defining the time when the event was detected. The format follows ISO 8601, example: `2021-03-26T08:02:30Z` , time zone is always UTC.
t2	String	*(time2)* A time stamp defining the time when the event was stored in the event database. The format follows ISO 8601, example: `2021-03-26T08:02:30Z` , time zone is always UTC.
s1	String	*(source1)* The name / id of the M&C which generated this event.
s2	String	*(source2)* The name of the device which generated this event.
o1	String	*(origin1)* The name of the originator. Used with parameter change events to tell who commanded the change. May be a user name or in case of automatic parameter changes the name of the device which caused the change.
o2	String	*(origin2)* The host name / IP address from where a parameter change was commanded.

5.5.29 FaultList

A FaultList message carries the list of active faults and warnings which are pending at a given M&C. The backend updates this list whenever a fault changes at this M&C but at a maximum rate of one message per second.

To subscribe for the FaultList message, a/api/v1/subscribe API call must be made for the message ID `{M&C-Name}.SYSTEM.faultList` where {M&C-Name} is the name / ID of the M&C of interest.

Data Model (extends Message)

Key	Type	Value
id	String	The message ID. This is always the message ID described above. (see Message)
type	String	The message type (always **FAULTLIST**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
faults	Array	An array of FaultInfo objects representing list of actually pending faults and warnings at the given M&C.

5.5.30 FaultInfo

A FaultInfo contains the properties of one fault of warning reported in aFaultList message.

Data Model

Key	Type	Value
time	String	A time stamp telling when this fault changed its state. The format follows ISO 8601, example: `2021-03-26T08:02:30Z` , time zone is always UTC.
src	String	The name of the device which generated this event.
pri	Number	The event priority coded as a one digit number 0 = suppressed or device out-of-service, the fault shall not be shown 1 = informational 2 = fault 3 = alarm 4 = warning
msg	String	The fault message text.

5.5.31 DeviceFrameDefinition

A DeviceFrameDefinition document defines tabs to be shown in a sat-nms device screen. It resembles the content of a file in the *dframes* directory of the ***sat-nms*** M&C software.

DeviceFrameDefinition documents are returned by a GET /api/v1/dframes/{name} call.

Data Model

Key	Type	Value
name	String	The file name of the dframe definition
allowPresets	Boolean	true = the UI shall show the load/save preset buttons for this dframe
presetPrivilege	Number	the privilege level required to launch the preset load/save dialogs. This only is meaningful with *allowPresets=true*
tabs	Array	Contains a list of DeviceFrameTab objects, each describing one tab of the device frame

5.5.32 DeviceFrameTab

A DeviceFrameTab document defines one tab to be shown in a ***sat-nms*** device screen. It appears as part of the DeviceFrameDefinition data object.

Data Model

Key	Type	Value
icon	String	The name of the icon to be shown in the tab header
dscreen	String	The name of the dscreen to be shown in this tab
description	String	A description of what is shown in this tab, the Java UI shows this as a mouse over tooltip

5.5.33 ScreenDefinition

A ScreenDefinition document defines the layout of a screen / page to display. The document contains some global properties of the screen and an array of ScreenElement objects, which each define one UI widget with its properties.

With the sat-nms software there are two general types of screen definitions: 'user defined screens' and 'device screens'. While the first type may contain widgets which show parameters of any device controlled by the sat-nms M&C are 'device screens' bound to one particular device. Device screens are returned by a GET /api/v1/dscreens/{name} call, user screens by a GET /api/v1/uscreens/{name} call.

Data Model

Key	Type	Value
name	String	The file name of the screen definition
deviceRelative	Boolean	true = contains no device IDs, the front end application must add the device ID from the context of the screen
width	Number	Width of the screen (pixels in the Java client)
height	Number	Height of the screen (pixels in the Java client)
bgImage	String	Name of a background image to draw in the screen. An empty value states that there is no background image.
noEditTake	Boolean	true = exclude this screen from global edit/take if this feature has been set for the UI.
elements	Array	Contains ScreenElement objects, the list of the screen elements contained in this screen.

5.5.34 ScreenElement

A ScreenElement object describes the properties of one UI widget contained in a ScreenDefinition. This comprises the type of widget, its location and size and many more parameters which depend on the type of widget this ScreenElement defines. This way ScreenElement is kind of an abstract definition or 'base class' in terms of object oriented programming. ScreenElement defines the parameters which are common to all types of screen elements.

Data Model

Key	Type	Value
type	String	The type of the screen element. See the list of screen element types below for details.
xpos	Number	X-position of the upper left corner of the element in the screen (pixels in the Java client)
ypos	Number	Y-position of the upper left corner of the element in the screen (pixels in the Java client)
width	Number	Width of the element (pixels in the Java client)
height	Number	Height of the element (pixels in the Java client)

As mentioned above, a ScreenDefinition will never contain a ScreenElement with only the parameters shown in the table above. All ScreenElement objects will be of a certain screen element type and will contain all parameters defines for this element type. The parameters described for ScreenElement will be always included.

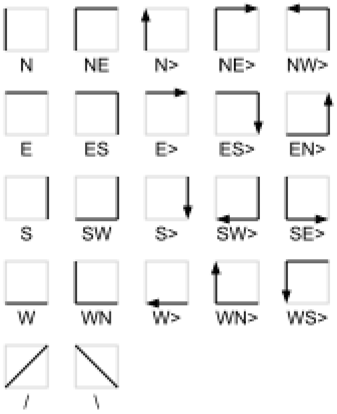
List of screen element types:

Element Type	Description
ArrowElement	Draws a horizontal/vertical line or arrow.
AzElElement	A chart showing the tracking history of an antenna controller as a point cloud in an elevation over azimuth coordinate system.
ButtonElement	Defines a button which launches another screen, e.g. for detail views.
ChartElement	A strip chart element, displaying a numeric parameter as an y/t diagram that automatically advances with 1 pixel / second.
DeviceElement	Places an icon into the screen which represents a device. This icon will display the device's operating/fault state by it's color / shape.
DisplayElement	A parameter display field which does not allow the parameter to be edited.
EditButtonElement	Places an EDIT button into the screen, implicitly forcing the EDIT/TAKE operating method for this screen instead of changing each parameter independently.
FaultElement	A special display element to display a fault flag. It shows the textual description of the fault and it's state in an entry field frame. The right mouse button shows a context menu to change the fault's priority
FrameElement	Draws a sunken 3D frame, may be used to group parameters.
GaugeElement	A gauge element, displaying a numeric parameter as a horizontal bar.
IconElement	Places an icon (PNG, GIF or JPG image) into the screen. Optionally the icon can be programmed to change with a parameter value.
LatchingButtonElement	A button which displays/controls a 2-state parameter using it's pressed state.
LockButtonElement	A button to control the lock state for exclusive operation for a number of devices.
MCPElement	The MCP display element integrates the measurement display of a spectrum analyzer device in multi channel measurement mode in a user screen.

Element Type	Description
ParameterButtonElement	A button which sends a certain parameter value when pressed.
ParameterElement	Either a parameter entry field or a drop down box, depending on the type of parameter.
RadioButtonElement	A parameter entry field specially for CHOICE parameters.
RectElement	Draws a rectangle.
SpectrumElement	The spectrum display element integrates the spectrum display of a spectrum analyzer device in a user screen.
SwitchElement	Like the device icon, but additionally displays the actual position of a switch (Meant to be used for block diagrams showing the true signal path.
TakeButtonElement	Places a TAKE button into the screen.
TargetListElement	The ODM Target List element shows the list of targets of a SatService-ACU-ODM antenna controller device. It permits to recall, save or delete target definitions of this type of antenna controller. It is specialized to this antenna controller, does not support other types.
TextElement	Displays a single line of text.
ThumbnailElement	Displays a thumbnail image fetched directly from a video processing device (encoder, decoder, gateway)
XYChartElement	This element shows the relation of two numeric variables in an X/Y diagram, featuring a trace which shows the recent history of the values with a configurable depth. The update rate, the diagram scaling and much more is configurable with this screen element.

5.5.35 ArrowElement

The ArrowElement draws a horizontal and/or a vertical line and optionally an arrowhead. The line's color is selectable, also may follow a parameter value. Strictly spoken the ArrowElement is a rectangle with only one or two sides drawn. The parameter *arrowCode* defines which sides of the rectangle shall be drawn and if an arrow shall be placed at the end of the line. The figure below shows all arrow codes defined:



Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "ArrowElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The message ID this element listens to. Is only relevant if *variableLineColor* reads *true*.
arrowCode	String	A 1..3 character code defining the line/arrow outline. See the figure above which arrow codes are possible.
color	String	The color used for the line / arrow.
variableLineColor	Boolean	true = set the line color from the value received from the parameter addressed by *id*.
useColorTranslation	Boolean	use the colors[] translation
colors	Array	of ColorDefinition objects. Defines how to translate values of the *id* parameter to color values if *useColorTranslation* reads *true*.

Variable Line Color

With *variableLineColor=true* the ArrowElement will change its line color depending on the value of the parameter addressed by *id*. This way the line can be used as a status indicator, e.g. in a block diagram signalling "the signal goes actually this way". The way the ArrowElement interprets the parameter value depends on the value of *useColorTranslation*.

If *useColorTranslation=false*, the ArrowElement tries to decode the line color directly from the parameter value. First it does the following tests on the parameter value (not case sensitive):

Parameter Value	**Line Color And Thickness**
ends with **BOLD**	The line is drawn thicker than normal
starts with **BLACK**	The line is drawn with this color
starts with **WHITE**	The line is drawn with this color
starts with **RED**	The line is drawn with this color
starts with **BLUE**	The line is drawn with this color
starts with **GREEN**	The line is drawn with this color
starts with **YELLOW**	The line is drawn with this color
starts with **GRAY**	The line is drawn with this color
contains **TRUE**	If the message ID identifies the parameter as an alarm flag (the ID contains 'fault'), the line is drawn **RED-BOLD**, for other parameters it is drawn **GREEN-BOLD**
contains **FAULT**	The line is drawn **RED-BOLD**
starts with **ON**	The line is drawn **GREEN-BOLD**

If none of the above conditions apply, the ArrowElement tries to interpret the parameter value as a hex coded color (#RRGGBB). If this also fails, the standard line color defined in *color* is used.

If *useColorTranslation=true*, the ArrowElement uses the ColorDefinition table in *colors[]* to decode the line's color and thickness.

5.5.36 ButtonElement

The ButtonElement creates a button which launches another user interface screen if pressed.

The screen to be launched may be another user defined screen or a predefined one.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "ParameterButtonElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
privilege	Number	The privilege level required to make a button click send a value
label	String	A label shown on the button
icon	String	An icon shown on the button
enableId	String	The enable ID (may be empty)
enableValue	String	The value which must be matched to enable the field
font	String	The font to be used
color	String	The color used for the label (#RRGGBB)
action	String	The action to perform if the button is clicked (see below)
par1	String	A parameter that applies for the action to perform. The meaning of this field depends on the *action* selected (see below)
par2	String	A second parameter that applies for the action to perform. The meaning of this field depends on the *action* selected (see below)
par3	String	A third parameter that applies for the action to perform. The meaning of this field depends on the *action* selected (see below)
variableBackgroundColor	Boolean	true = use a variable background color
colorId	String	The message ID to listen for the background color
colors	Array	of ColorDefinition objects. Defines how to translate values of the *colorId* parameter to color values.

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

CHILD-SCREEN

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

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

REPLACE-SCREEN

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

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

LOAD-PRESET

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

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

TRACK-VIEW

Launches a tracking view window. This window shows the tracking history of a given antenna controller device. Not all tracking controller devices are capable to report their tracking history.

- *par1:* The name of the antenna controller device for which the tracking history shall be shown. The device name has the name of the M&C prepended where it is controlled. Example: `MYMNC.MYACU` (must be set in the screen editor this way).
- *par2:* null
- *par3:* null

AZEL-VIEW

Launches a Az/El view window. This window shows the tracking history of a given antenna controller device as a cloud of dots in a azimuth over elevation coordinate system. Not all tracking controller devices are capable to report their tracking history.

- *par1:* The name of the antenna controller device for which the tracking history shall be shown. The device name has the name of the M&C prepended where it is controlled. Example: `MYMNC.MYACU` (must be set in the screen editor this way).
- *par2:* null
- *par3:* null

FREC-VIEW

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

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

SPECTR-VIEW

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

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

BROWSER-VIEW

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

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

escapes this behavior and gets replaced by a single '@' in the URL string.

- *par2:* null
- *par3:* null

VIDEO-VIEW

Launches a VideoLAN video player to show the video stream from a surveillance camera controlled by a M&C. The VIDEO-VIEW ButtonElement may only occur in device screens, the IP address of the camera has to be taken from the address_ parameter of the device.

- *par1:* null
- *par2:* null
- *par3:* null

ROBOT-IMPORT

Launches a Robot Import screen. This screen lets the operator select a Pointing-Robot file from the local PC, the file gets converted to sat-nms format an then transferred to the M&C.

- *par1:* null
- *par2:* null
- *par3:* null

TREE-NAVIGATE

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

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

UNKNOWN

All ButtonElement definitions in a screen, which have no WebGUI equivalent for their programmed *action* (like "launching a new Java VM') are translated into an UNKNOWN action. The Frontend may show this e.g. as a disabled button with an "n/a" label.

- *par1:* The complete content of the message-id field in the original button definition.
- *par2:* null
- *par3:* null

5.5.37 ChartElement

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

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

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "ChartElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The message ID this element listens to. This should point to a numeric parameter to make sense.
label	String	The label shown above the strip chart
font	String	The font to be used for the label
color	String	The color used for the label and the strip chart line.
mode	String	The strip chart's display mode. This mainly controls the scaling behavior of the chart. See Display Modes below.
scale	double	The scale 1/div
minValue	double	The value corresponding the bottom line of the widget
maxValue	double	The value corresponding the top line of the widget
minThreshold	double	Values below this threshold cause the widget to display a fault
maxThreshold	double	Values above this threshold cause the widget to display a fault

Display Modes

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

mode	**Description**

mode	**Description**
FLOATING-AUTO	Does a full autoscale. The y-offset of the strip chart is set that the newest value displayed appers at the middle of the y axis. The chart y-scale is evaluated in a 1-2-5 raster to the finest scale that allows all points in the history to be displayed in the chart area. All scaling parameters are ignored.
FLOATING-FIXED	Evaluates the y-offset like in **FLOATING-AUTO**, but applies a fixed y-scale as defined in *scale*. The *scale* value is 1/div and chart height is assumed as 2 divisions. So, the top line of the chart corresponds to y-offset + *scale*, the bottom line to y-offset - *scale*.
FIXED	Sets a fixed y-range from the *minValue* / *maxValue* parameters.
FIXED-THRESHOLD	Like **FIXED**, but also checks every new value against the *minThreshold* / *maxThreshold* limits. If outside the limits, the chart background turns to red.
ANTENNA-AZEL	Makes the element to be a AZ/EL x/y diagram instead of the strip chart. The details about this element type will be defined in a later version of the API.

5.5.38 AzElElement

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

Data Model (extends ScreenElement)

Key	**Type**	**Value**
type	String	Always "AzElElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The device name (prepended by the M&C name) of the antenna controller for which the AzElElement shall be shown.
label	String	The label shown above the chart
font	String	The font to be used for the label

5.5.39 DeviceElement

The DeviceElement represents a device in the M&C user interface. It displays the status of the device by it's color/shape and gives access to the device window for this particular device by a double mouse click. The right mouse button launches a context menu with common operations for the device.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "DeviceElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The device ID this element represents in the UI. The device ID consists of the M&C name followed by a dot and the device name.
privilege	Number	The privilege level required to show details about the device and to change the device's operating state.
icon	String	The (base-) name of the icon to show for the device. See Icon Selection below.
showName	Boolean	true = show the device name als a label on top of the icon
showIsBackup	Boolean	true = use a different icon variant for this device if it is in 'is backup' state.
signalBackupId	String	A message ID to listen for a parameter from which the 'is backup' state is derived. Ignored unless *showIsBackup* is true.
backupValue	String	The parameter value which must be matched for the 'is backup' state.
font	String	The font to be used for the label.
color	String	The color to be used for the label (#RRGGBB)

Based on the *id* parameter, the DeviceElement subscribes for a number of parameters, the message IDs of these are as shown below:

{id}.fault: The device's summary fault state. May be one of**OK.**, **WARNING** or **Summary**

FAULT

{id}.faults.99: (Boolean) The device's communication fault

{id}.mode: The operation mode of the device. This is one of**OPERATIONAL**, **FAULT-SUPPRESSED**, **OUT-OF-SERVICE** and **MAINTENANCE**

{id}.info.signal.on: If the value reported via this messageID is**true** or **ON**, the device is considered to be actually ON-AIR. In this case, there is an additional emblem displayed at the upper right corner of the icon signalling this state.

signalBackupId: If the value reported via this messageID equals the *backupValue*, the DeviceElement considers the device to be a backup for something else.

Icon Selection

The DeviceElement shows the state of the device by switching between several variants of of the icon. The data model defines the base version of the icon which is shown when the device is in operating state without a fault. Other states are displayed by changing the icon to a variant which has a status code appended to the base name of the icon:

Icon Name	**State**	**Example**
basename.gif	Operational and OK	□
basename-**C**.gif	There is no communication to the device	COMFLT!
basename-**F**.gif	The device shows a FAULT	FAULT!
basename-**W**.gif	The device shows a WARNING	🟨
basename-**S**.gif	The device id in fault suppressed mode	FAULT SUPPRESSED
basename-**M**.gif	The device is in maintenance mode	MAINTENANCE
basename-**O**.gif	The device is out of service	OUT OF SERVICE
basename-**B**.gif	The device is OK and used as backup	🟦

In the sat-nms software there exist several sets of these icons, the is extensible by the customer. Supported image formats / file extensions are *.gif, *.jpg and *.png.

The pseudo code sniplet below describes how the DeviceElements is expected to combine the individual status variables to the name of icon to be shown.

```
if summary-alarm
   if communication-alarm
      use -C icon
   else
      use -F icon
else if summary-warning
   use -W icon
else if mode == MAINTENANCE
   use -M icon
else if mode == OUT-OF-SERVICE
   use -O icon
else if mode == FAULT-SUPPRESSED
   use -S icon
else if is-backup
   use -B icon
else
   use standard icon
```

5.5.40 DisplayElement

The DisplayElement is used to display M&C parameters read-only. It looks much alike the ParameterElement element, but never allows to change the parameter it displays.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "DisplayElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The message ID this element listens to
label	String	A label shown above the field / drop down
enableId	String	The enable ID (may be empty)

Key	Type	Value
enableValue	String	The value which must be matched to enable the field
font	String	The font to be used
color	String	The color used for the label (#RRGGBB)
drawFrameless	Boolean	true = display without a frame and with transparent background
variableBackgroundColor	Boolean	true = use a variable background color.
colorId	String	The message ID to listen for the background color
colors	Array	of ColorDefinition objects. Defines how to translate values of the *colorId* parameter to color values.

Remarks

Conditionally Enable: If the *enableId* property is not empty, the ParameterElement subscribes for this additional parameter and disables (disabled means "dimmed" in this case) the widget unless the received parameter value matches the values stated in *enableValue*.

Font Selection: The sat-nms software defines 6 preset fonts named 'small', 'plain', 'bold', 'title', 'huge' and 'typewriter'. The *font* property contains one of these names. The selected font is used for the text in the widget, the label above the entry field always is shown with the 'plain' font, unless *font* reads 'small', in this case the label is shown with the 'small' font as well. For information about fonts and font sizes see section Fonts in the appendix of this document.

Variable Background Color: With the *variableBackgroundColor* parameter set 'true', the DisplayElement listens to the additional parameter stated in *colorId* and sets the background color of the widget according to the value received. The received value is translated to a color (and optional 'bold' printing) through the *colors* array. If the received value does not match any of the values listed in the array, the first entry in the array is used as a default / fallback.

5.5.41 EditButtonElement

The EditButtonElement places an EDIT button to the screen. It appears as a simple button with the label "EDIT" on it. A screen definition containing an EDIT button switches the screen to Edit/Take mode. This means, parameter changes are not sent immediately to the server, but the screen remembers (and marks) all changes until the operator either clicks TAKE to send all changes to the server or clicks EDIT again to revert all changes made. Also see the definition of TakeButtonElement

Data Model (extends ScreenElement)

Key	Type	Value

Key	Type	Value
type	String	Always "EditButtonElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
privilege	Number	The privilege level required to switch to EDIT mode.

5.5.42 FaultElement

The FaultElement shows a fault flag. The faults page of the predefined device windows makes use of this display element.

The software treats device faults in a special way. In the variable list of a device driver the fault flags appear as variables called "faults.00" .. "faults.99". The software automatically produces a variable "config.faults.XX" for each fault flag "faults.XX" the driver defines. This configuration variable controls the priority of the fault.

Actually, FaultElement is not included in the API. If the original screen definition contains instances of this type, they are skipped and not included in the ScreenDefinition reported to the UI.

5.5.43 FrameElement

The FrameElement draws a sunken 3D frame, which is intended to be used to group other elements. The 3D frame's inside area is assumes to be transparent, it does not conceal the screen elements it encloses.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "FrameElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement

5.5.44 GaugeElement

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

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "GaugeElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The message ID this element listens to. This should point to a numeric parameter to make sense.
label	String	The label shown above the strip chart
font	String	The font to be used for the label
color	String	The bar color. If thresholds are defined, this color is used while the value is inside the thresholds.
useMinMax	Boolean	If true, the min/max values for the gauge are defined in *minValue*, maxValue_. If false, the GaugeElement takes the limits from the watches parameter's range definition.
minValue	double	The value corresponding the left edge of the widget
maxValue	double	The value corresponding the right edge of the widget
useThresholds	Boolean	If true, the GaugeElement checks the value against the thresholds defined in *minThreshold*, *maxThreshold* and changes the color of the bar to *belowColor* or *above_Color* accordingly.
minThreshold	double	Values below this threshold cause the widget to display a fault
maxThreshold	double	Values above this threshold cause the widget to display a fault
belowColor	String	The bar color to be used if *useThresholds* is true and the actual value is below *minThreshold*
aboveColor	String	The bar color to be used if *useThresholds* is true and the actual value is above *maxThreshold*

Key	Type	Value
tickMode	String	One of 'NONE', 'TICKS', 'TICKS-LABELS'. 'TICKS' makes the element show ticks below the bar graph to indicate the bar position for the *minValue*, *minThreshold*, *maxThreshold* and *maxValue* values. The ticks are to be drawn outside the rectangle defining the element bounds. 'TICKS-LABELS' adds labels below the tick marks, showing the numeric values of the four limit parameters. The number of precision digits shown is taken from the subscribed value's range definition.

Restrictions for colors with *useThresholds* ###

Actually the colors to be used with *useThresholds* set are not freely definable. This is a restriction of how the sat-nms M&C software stores the screen definition, it uses a fixed set of color combinations when thresholds are enabled. These are the valid combinations:

belowColor	*color*	*aboveColor*	sat-nms mode
#ff0000	#00ff00	#ff0000	red - green - red
#ff0000	#ffff00	#00ff00	red - yellow - green
#00ff00	#ffff00	#ff0000	green - yellow - red

Using other color combinations than those from the table above will cause unpredictable results.

Special Behavior With Level-Adjust/Level-Control Devices

Normally the GaugeElement expects a numeric value to display in the bar graph. When used with the 'gauge' or the 'gauge2' parameter of the Level-Adjust/Level-Control devices, the GaugeElement must implement a special behavior:

The gauge/gauge2 parameters contain a comma separated string with a list of numeric and textual values. The GaugeElement must parse this string and update its display from the values parsed. The content of the gauge/gauge2 comma separated list is defined as follows:

Pos.	Name	Remark
0	val	The gauge value (numeric)
1	minLabel	The label to be shown for the min value below the graph (textual). Overwrites the label derived from the *minValue* field described above.

Pos.	Name	Remark
2	min	The value corresponding the left edge of the widget (numeric). Overwrites the value defined in the *minValue* field described above.
3	maxLabel	The label to be shown for the max value below the graph (textual). Overwrites the label derived from the *maxValue* field described above.
4	max	The value corresponding the right edge of the widget (numeric). Overwrites the value defined in the *maxValue* field described above.
5	minTLabel	The label to be shown for the minThreshold value below the graph (textual). Overwrites the label derived from the *minThreshold* field described above.
6	minThreshold	The value lower threshold value (numeric). Overwrites the value defined in the *minThreshold* field described above.
7	maxTLabel	The label to be shown for the maxThreshold value below the graph (textual). Overwrites the label derived from the *maxThreshold* field described above.
8	maxThreshold	The upper threshold value (numeric). Overwrites the value defined in the *maxThreshold* field described above.

In order to maintain compatibility to future level control devices, the GaugeElement should not use the parameter names or the driver type of the device to decide if this special behavior applies. Instead, the GaugeElement should make this decision from the data message it receives:

Contains the message a 9-element comma separated list?

If yes, the element should behave as with *useMinMax*=true and *useThresholds*=true, but with the settings contained in this list instead of the ones contained in the GaugeElement data.

Contains the message a 5-element comma separated list?

If yes, the element should behave as with *useMinMax*=true and *useThresholds*=false, but with the settings contained in this list instead of the ones contained in the GaugeElement data.

In all other cases the element should try to parse the message value to floating point value and display it with the settings given in the GaugeElement data object.

5.5.45 IconElement

The IconElement shows an arbitrary GIF/JPEG picture. If a message ID is defined with the element properties, the image displayed will change with the parameter value addressed by the message identifier.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "IconElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	the message ID this element listens to
icon	String	the initial icon file name

When displayed the first time, the Iconelement shows the image specified in*icon*. If the message ID *id* is not empty, the IconElement dubscribes for this ID and changes the displayed icon every time a new value is received. To make use of this, the name of the icon should be structured in the following way:

```
base-name.{value-part}.{extension}
```

The value-part gets replaced by the parameter value received. For a boolean value for example, two icons 'red-lamp.false.png' and 'red-lamp.true.png'. The IconElement is configured to show 'red-lamp.false.png' initially. When the parameter addressed by *id* gets updated, the display changes according to the parameter value.

5.5.46 LatchingButtonElement

The LatchingButtonElement works much like the ParameterButtonElement described later in this document, but is specialized to show and control an enumeration parameter which knows exactly two states (e.g. on/off or true/false).

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

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

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "LatchingButtonElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The message ID this element listens for
privilege	Number	The privilege level required to change the button state / parameter value
labelUp	String	The label shown on the button when not pressed
labelDown	String	The label shown on the button when pressed
iconUp	String	The icon shown on the button when not pressed
iconDown	String	The icon shown on the button when pressed
enableId	String	The enable ID (may be empty)
enableValue	String	The value which must be matched to enable the field
font	String	The font to be used for the button label
color	String	The color used for the label
mustQuery	Boolean	Show a query popup before changing the value
queryText	String	the text to show in the query
valueUp	String	The parameter value to send when the button is going up
valueDown	String	The parameter value to send when the button is going down
variableBackgroundColor	Boolean	true = use a variable background color
colorId	String	The message ID to listen for the background color
colors	Array	of ColorDefinition objects. Defines how to translate values of the *colorId* parameter to color values.

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

The *valueUp* and *valueDown* values are not only sent when the button is pressed or released. They also are used to set the button state when a parameter value is received though *id*: If the value matches *valueDown* the button goes to 'pressed' state, in all other cases the button gets released. Labels or icons change an this moment according to the new state.

5.5.47 LockButtonElement

The LockButtonElement defines a button to lock the operation of a number of device devices that no other user can change device settings. The button is in pressed state if at least one of the devices it is assigned to is locked by some user. An operator only may release the button if he set the lock by himself of if he has a privilege level of 150 or more.

Actually, LockButtonElement is not included in the API. If the original screen definition contains instances of this type, they are skipped and not included in the ScreenDefinition reported to the UI.

5.5.48 MCPElement

The MCPElement embeds the measurement display of a spectrum analyzer device in multi channel power measurement mode into the screen.

Actually, MCPElement is not included in the API. If the original screen definition contains instances of this type, they are skipped and not included in the ScreenDefinition reported to the UI.

5.5.49 ParameterButtonElement

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

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "ParameterButtonElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The message ID this element sends

Key	Type	Value
privilege	Number	The privilege level required to make a button click send a value
label	String	A label shown on the button
icon	String	An icon shown on the button
enableId	String	The enable ID (may be empty)
enableValue	String	The value which must be matched to enable the field
font	String	The font to be used
color	String	The color used for the label (#RRGGBB)
mustQuery	Boolean	true = show a query popup before changing the value
queryText	String	The text to show in the query. If empty, a default query text shall be used.
parameterValue	String	The parameter value to send
isMacroButton	Boolean	If true, this button starts a macro instead of sending a parameter
variableBackgroundColor	Boolean	true = use a variable background color
colorId	String	The message ID to listen for the background color
colors	Array	of ColorDefinition objects. Defines how to translate values of the *colorId* parameter to color values.

The ParameterButtonElement never uses a button label and an icon at the same time. Either *icon* or *label* is an empty string.

If no *parameterValue* is set, the ParameterButtonElement sends the button label or the icon name as the parameter value when clicked

Macro Buttons

The MacroButton is a special version of the ParameterButtonElement. A *isMacroButton = true* marks a ParameterButtonElement to be a MacroButton. In this case the 'id' field contains the name of the M&C where to play the macro and the macro name in this case, separated by a period. Example `VLC0001.myMacro`

Instead of calling 'poke' when the button is clicked, a POST /api/v1/playmacro/{mnc-name}/{macro-name} call must be made by the front end (Example

`POST /api/v1/playmacro/VLC0001/myMacro` .

5.5.50 ParameterElement

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

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "ParameterElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The message ID this element listens to
privilege	Number	The privilege level requires to change this parameter
label	String	A label shown above the field / drop down
enableId	String	The enable ID (may be empty)
enableValue	String	The value which must be matched to enable the field
font	String	The font to be used
color	String	The color used for the label (#RRGGBB)
mustQuery	Boolean	true = show a query popup before changing the value
queryText	String	The text to show in the query. If empty, a default query text shall be used.
useSpinButtons	Boolean	true = use spin buttons for numeric values
spinSmallIncrement	String	Small spin increment, empty = use default
spinLargeIncrement	String	Large spin increment, empty = use default
useComboBox	Boolean	true = use a searchable combobox

Key	**Type**	**Value**
editableComboBox	Boolean	true = allow free text edit in the box
variableBackgroundColor	Boolean	true = use a variable background color.
colorId	String	The message ID to listen for the background color
colors	Array	of ColorDefinition objects. Defines how to translate values of the *colorId* parameter to color values.

Remarks

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

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

Conditionally Enable: If the *enableId* property is not empty, the ParameterElement subscribes for this additional parameter and disables (disabled means "dimmed, not changeable") the widget unless the received parameter value matches the values stated in *enableValue*.

Font Selection: The sat-nms software defines 6 preset fonts named 'small', 'plain', 'bold', 'title', 'huge' and 'typewriter'. The *font* property contains one of these names. The selected font is used for the text in the widget, the label above the entry field always is shown with the 'plain' font, unless *font* reads 'small', in this case the label is shown with the 'small' font as well. For information about fonts and font sizes see section Fonts in the appendix of this document.

Operator Query: If the property *mustQuery* reads 'true', the ParameterElement pops up a 'Do you really want...' dialog before a parameter change is sent to the server. *queryText* contains the question to be shown in this case. In the question text, placeholders may used for two values: Any occurrences of the pattern **$P** get replaced by the parameter name (message ID). Any occurrences of the pattern **$V** get replaced by the new value to set. If*queryText* is empty, the ParameterElement uses a standard text instead.

Spin Buttons: With *useSpinButtons* set 'true', the ParameterElement shows spin buttons with editable numberic values. The spin buttons allow to increment/decrement the value by clicking on them. By default, a spin button click increments or decrements the lowest significant digit shown in the entry field. With the shift key hold down, the effective increment is x10. The above applies if the *spinSmallIncrement* / *spinLargeIncrement* properties are empty. If set, these values override the defaults.

ComboBox Variants: For CHOICE parameter the ParameterElement normally uses a simple drop down box. With selection parameter with a handful of choices this easy to operate. There are however situations where the ParameterElement must handle hundreds of choices (e.g. in satellite channel lists). For this the ParameterElement provides variants of the simple drop down box.

With the *useComboBox* parameter set 'true', the ParameterElement uses a special combo box, which implements some filtering on the displayed choices. The operator may enter some text into the edit field of the combo box and only choices which (partially) contain the the entered text are offered for selection.

With the *editableComboBox* parameter set 'true', the ParameterElement permits to send any entered text as the new parameter value to the server, not only one of the valid choices for this parameter.

Variable Background Color: With the *variableBackgroundColor* parameter set 'true', the ParameterElement listens to the additional parameter stated in *colorId* and sets the background color of the widget according to the value received. The received value is translated to a color (and optional 'bold' printing) through the *colors* array. If the received value does not match any of the values listed in the array, the first entry in the array is used as a default / fallback.

5.5.51 RadioButtonElement

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

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "RadioButtonElement".

Key	Type	Value
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The message ID this element listens to
privilege	Number	The privilege level requires to change this parameter
label	String	A label shown above the field / drop down
enableId	String	The enable ID (may be empty)
enableValue	String	The value which must be matched to enable the field
font	String	The font to be used
color	String	The color used for the label (#RRGGBB)
mustQuery	Boolean	true = show a query popup before changing the value
queryText	String	The text to show in the query. If empty, a default query text shall be used.
drawFrame	Boolean	true = draw a gray rectangle around the radio button group.
variableBackgroundColor	Boolean	true = use a variable background color.
colorId	String	The message ID to listen for the background color
colors	Array	of ColorDefinition objects. Defines how to translate values of the *colorId* parameter to color values.

5.5.52 RectElement

The RectElement draws a rectangle with a selectable color. The element may listen to a parameter value and change the color of the rectangle according to the parameter value.

Data Model (extends ScreenElement)

Key	Type	Value

Key	Type	Value
type	String	Always "RectElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The message ID this element listens to. Is only relevant if *variableLineColor* reads *true*.
fill	Boolean	true = draw the rectangle filled, not only the outline.
color	String	The color used for the rectangle.
variableLineColor	Boolean	true = set the line/fill color from the value received from the parameter addressed by *id*.
useColorTranslation	Boolean	use the colors[] translation
colors	Array	of ColorDefinition objects. Defines how to translate values of the *id* parameter to color values if *useColorTranslation* reads *true*.

Variable Line Color

With *variableLineColor=true* the RectElement will change its line/fill color depending on the value of the parameter addressed by *id*. This way the rectangle can be used as a status indicator. The way the RectElement interprets the parameter value depends on the value of *useColorTranslation*.

If *useColorTranslation=false*, the RectElement tries to decode the line color directly from the parameter value. First it does the following tests on the parameter value (not case sensitive):

Parameter Value	Line Color And Thickness
ends with **BOLD**	The line is drawn thicker than normal
starts with **BLACK**	The line is drawn with this color
starts with **WHITE**	The line is drawn with this color

Parameter Value	Line Color And Thickness
starts with **RED**	The line is drawn with this color
starts with **BLUE**	The line is drawn with this color
starts with **GREEN**	The line is drawn with this color
starts with **YELLOW**	The line is drawn with this color
starts with **GRAY**	The line is drawn with this color
contains **TRUE**	If the message ID identifies the parameter as an alarm flag (the ID contains 'fault'), the line is drawn **RED-BOLD**, for other parameters it is drawn **GREEN-BOLD**
contains **FAULT**	The line is drawn **RED-BOLD**
starts with **ON**	The line is drawn **GREEN-BOLD**

If none of the above conditions apply, the RectElement tries to interpret the parameter value as a hex coded color (#RRGGBB). If this also fails, the standard line/fill color defined in *color* is used.

If *useColorTranslation=true*, the RectElement uses the ColorDefinition table in *colors[]* to decode the line's color and thickness.

5.5.53 SpectrumElement

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

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

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "SpectrumElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The device name of the spectrum analyzer which shall be shown.
privilege	Number	The privilege level requires to change settings of the spectrum analyzer.
label	String	A label shown above the element
font	String	The font to be used for the label
color	String	The color used for the label (#RRGGBB)

5.5.54 SwitchElement

The SwitchElement is a special version of the device element which may be used to visualize the position of a switch in a user interface screen designed as a block diagram. The switch icon has all capabilities of a plain device icon display element. The context menu shown with this right mouse button additionally contains an option to toggle the switch position.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "SwitchElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The device ID this element represents in the UI. The device ID consists of the M&C name followed by a dot and the device name.
privilege	Number	The privilege level required to show details about the device and to change the device's operating state.

Key	Type	Value
icon	String	The (base-) name of the icon to show
enableId	String	The enable ID (may be empty)
enableValue	String	The value which must be matched to enable the field
touchScreenMode	Boolean	true = use a different click response, optimized for touch screen
font	String	The font to be used for the label
color	String	The color used for the label (#RRGGBB)
mustQuery	Boolean	true = show a query popup before changing the value
queryText	String	The text to show in the query

Based on the *id* parameter, the SwitchElement subscribes for a number of parameters, the message IDs of these are as shown below:

{id}.fault: The device's summary fault state. May be one of**OK.**, **WARNING** or **Summary FAULT**

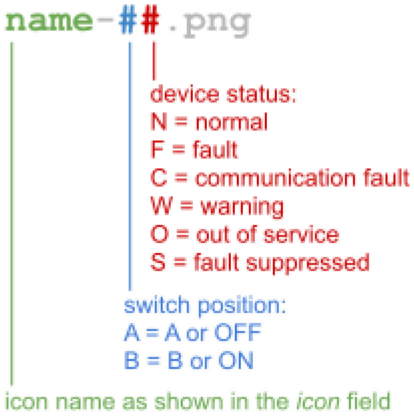
{id}.faults.99: (Boolean) The device's communication fault

{id}.mode: The operation mode of the device. This is one of**OPERATIONAL**, **FAULT-SUPPRESSED**, **OUT-OF-SERVICE** and **MAINTENANCE**

{id}.position: The switch position. May be either 'A' or 'B'. Recognized 'ON'/'OFF' es well and changes the the command sent with a switch toggle accordingly.

Icon Selection

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



5.5.55 TakeButtonElement

The TakeButtonElement places an TAKE button to the screen. It appears as a simple button with the label "TAKE" on it. The TAKE button complements the EDIT button described above in this document. It sends the changed parameter values to their destination when pressed.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "TakeButtonElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement

5.5.56 TargetListElement

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

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "TargetListElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement

Key	Type	Value
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The device name of the SatService-ACU-ODM. In device screens the *id* has to be interpreted differently, see below.
privilege	Number	The privilege level required to show details about the device and to change the device's operating state.
enableId	String	The enable ID (may be empty)
enableValue	String	The value which must be matched to enable the field

Device Name Indirection

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

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

Target List Parsing

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

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

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

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

5.5.57 TextElement

The TextElement shows a label (a one line text string) with a chosable font / color.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "TextElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
label	String	The text to be drawn
font	String	The font to be used
color	String	The color used for the label (#RRGGBB)

5.5.58 ThumbnailElement

The ThumbnailElement displays a thumbnail of the video actually processed by a device like an encoder, decoder or gateway. The displayed icon is a static image, fetched directly from the device and updated every couple of seconds.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "ThumbnailElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The message ID this element listens to

REMARKS

The Thumbnail Icon element itself must fetch the image to display in regular intervals directly from the device. As a consequence of this the following restrictions apply:

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

With *urlType = PLAIN* , the *imageUrl* is used without modification to fetch the image. The image received from the device must be scaled to fit the screen element size.

With *urlType = ADVANCED* the front end appends some parameters to the URL as shown in the example below. In this case no scaling is done as the software queries the image already in the correct size from the device.

```
http://1.2.3.4/somepath?t=1693050862003&w=320&h=180
```

The example above starts from a *imageUrl = http://1.2.3.4/somepath* definition. With *urlType = ADVANCED* the front end must add the parameters *t*, *w*, and *h* as shown. The value of *t* is the actual time in milliseconds since Jan 1st 1970, 00:00 UTC, *w* and *h* define the dimension of the image in pixels. They usually math the *width* and *height* fields of the data model.

5.5.59 XYChartElement

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

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "TextElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
idx	String	The message ID of the parameter to show at the x-axis
idy	String	The message ID of the parameter to show at the y-axis
label	String	The label to be drawn above the diagram
font	String	The font to be used
color	String	The color used for the label (#RRGGBB)
divisions	Number	The number of divisions shown in the diagram for both directions. Typical values are 2 ('hair cross'), 4 or 10.
interval	Number	The update time interval for the display in seconds. 0.1 means to add every 100msec a new value to the display and remove the oldest value from the buffer at the same time.

Key	Type	Value
bufsize	Number	The display maintains a "first in first out" buffer of a size defined with this parameter. The buffer provides a short time memory the display shows as a trace of past values. Typical values are in the range 100-300.
xorigin	Number	The value to be shown at the center of the x axis
yorigin	Number	The value to be shown at the center of the y axis
xscale	Number	The x axis scale in 1/division
yscale	Number	The y axis scale in 1/division
limits	Boolean	If true, the display monitors the actual X/Y values to be within the limits defined below. The limit values are shown as a dark red rectangle. When the actual X/Y values exceed the limits, the diagram background becomes red.
minx	Number	The lower x limit to check. Only meaningful if *limits* is true.
maxx	Number	The upper x limit to check. Only meaningful if *limits* is true.
miny	Number	The lower y limit to check. Only meaningful if *limits* is true.
maxy	Number	The upper y limit to check. Only meaningful if *limits* is true.

5.5.60 ColorDefinition

A ColorDefinition data object defines an entry of a value to color translation table which appears as the *colors* array with several ScreenElement definitions.

Data Model

Key	Type	Value
value	String	The parameter value for which the color shall be used
color	String	The color value (#RRGGBB)
bold	Boolean	true draws the text bold if the value matches

5.5.61 ItemList

The ItemList data object is used by the API at several places to return a list of named items to the front end, e.g. when a list of existing macros or device presets is required.

Data Model

Key	Type	Value
items	Array of String	The item list

5.5.62 Macro

A Macro data object contains a macro of the M&C software, a multi-line text in the sat-nms macro language containing settings / commands to be executed.

Data Model

Key	Type	Value
name	String	The name of this macro
macro	String	The macro itself (multi-line text, lines separated by '\n' character sequences).

5.5.63 PresetValue

Contains one parameter / value definition in a DevicePreset.

Key	Type	Value
name	String	The name of the driver variable this refers to. The driver variable name is the message ID of a parameter with the M&C name and the device name stripped off.
value	String	The value to be assigned to *varName* when the preset get applied to a device.

Values appear in the same way here as in the device preset file. This is formatted along the Range definition for this driver variable:

- FLOAT parameters are formatted with the precision defined in the range, but without a trailing unit string.
- INTEGER parameters appear as simple integer values, also without unit string.
- HEX parameters appear as hexadecimal values without unit string, letters A-F are upper case.
- BOOLEAN parameters are written as TRUE/FALSE.
- CHOICE parameters show the selected choice string.
- TEXT parameters are written without quotes, this means trailing blancs are not stored (this is a general limitation of the sat-nms M&C software)

5.5.64 DevicePreset

A DevicePreset data object contains a device preset of the M&C software.

Data Model

Key	Type	Value
name	String	The name of this preset
driver	String	The name of the device driver this preset is intended to be used with.
comment	String	The comment line appearing as the first line of a device preset file in the sat-nms M&C. Does not include the leading comment character ('#') appearing in the sat-nms preset file.
preset	Array of PresetValue	The value definitions from the device preset and an array of PresetValue definitions.

5.5.65 PresetVars

A PresetVars data object contains a description of an M&C device driver with its preset-storable variable definitions.

Data Model

Key	Type	Value
driver	String	The name of the device driver.
comment	String	The content of the COMMENT statement in the driver (includes the driver version).
vars	Array of Range	The Range definitions of all driver variable which may be stored in a preset. This includes all writable variables which have neither the SETUP nor the NOPRESET property set. The list contains a Range object for each variable with the data type, limits, choices, unit and precision as defined in the device driver file. The *messageId* field of each Range object contains the local name of the variable, not a fully qualified message ID

5.5.66 DeviceVars

A DeviceVars data object contains a description of an M&C device driver with all its variable definitions.

Data Model

Key	Type	Value
driver	String	The name of the device driver.

Key	Type	Value
comment	String	The content of the COMMENT statement in the driver (includes the driver version).
vars	Array of Range	The Range definitions of all driver variables. The list contains a Range object for each variable with the data type, limits, choices, unit and precision as defined in the device driver file. The *messageId* field of each Range object contains the local name of the variable, not a fully qualified message ID

5.5.67 KeyValPair

A KeyValPair defines a key / value pair as used with the Dictionary data model.

Data Model

Key	Type	Value
key	String	The name of the key
val	String	The value assigned to *key*

5.5.68 Dictionary

The Dictionary data model contains a list of key / value pairs.

Data Model

Key	Type	Value
dict	Array	An array of KeyValPair data objects

5.5.69 SatDbSatellite

The SatDbSatellite data record contains the basic properties for one satellite in the *satellites* database table. The fields in the SatDbSatellite data record are 1:1 replicas of the SQL database record. Please refer to the database model description at section Satellite Database how relations between tables are maintained using IDs.

Data Model

Key	Type	Value
satelliteId	Number	unique, primary key, auto generated

Key	Type	Value
name	String	free text, typically the name as given by satellite operator
operatorId	Number	relation to table *satellite-operators*
noradNumber	Number	unique name (world wide standardized), also relation to table *norad_tle*
intDesignator	String	unique name (world wide standardized)
orbitPosition	Number	satellite orbit position in degrees
inclination	Number	satellite inclination in degrees
alias	String	free text, can be used by customer to add own name
modified	String	timestamp, generated by database at creation / modification time

5.5.70 SatDbBeaconAttenuation

The SatDbBeaconAttenuation data record contains the attenuation value stored in the database for one particular pair of antennaId and beaconId. This is the attenuation to be applied for the given beacon signal when receivbed by this antenn.

Data Model

Key	Type	Value
attenuationId	Number	unique, primary key, auto generated
beaconId	Number	ID of the beacon this data record belongs to
antennaId	Number	ID of the antenna this beacon belongs to
attenuation	Number	receiver attenuation, floating point, dB
modified	String	timestamp, generated by database at creation / modification time

5.5.71 SatDbBeacon

The SatDbBeacon data record contains the basic properties for one satellite in the *satellite_beacons* database table. The fields in the SatDbBeacon data record are 1:1 replicas of the SQL database record. Please refer to the database model description at section Satellite Database how relations between tables are maintained using IDs.

There may be multiple beacon definitions assigned to one satellite. These beacons share the same *satelliteId*. One of these beacons is the default beacon, this and only this beacon has the *defaultBeacon* flag set *true*.

Data Model

Key	Type	Value
beaconId	Number	unique, primary key, auto generated
satelliteId	Number	ID of the satellite this beacon belongs to
frequency	Number	beacon frequency, floating point, MHz
polarization	String	this is an enumeration, may be one of "H", "V", "RHCP" or "LHCP"
defaultBeacon	Boolean	true means "this is the default beacon for this satellite"
comment	String	free text describing the purpose of this beacon
modified	String	timestamp, generated by database at creation / modification time

5.5.72 SatDbTcAttenuation

The SatDbTcAttenuation data record contains the attenuation values stored in the database for one particular pair of antennaId and tcId. This is the attenuation to be applied for the given TC signal when transmitted by this antenna. The two attenuation values are applied to the same signal at different stages of the transmit chain, e.g. at the upconverter and the HPA.

Data Model

Key	Type	Value
attenuationId	Number	unique, primary key, auto generated
tcId	Number	ID of the TC signal this data record belongs to
antennaId	Number	ID of the antenna this beacon belongs to
attenuation1	Number	transmit attenuation1, floating point, dB
attenuation2	Number	transmit attenuation2, floating point, dB
modified	String	timestamp, generated by database at creation / modification time

5.5.73 SatDbTc

The SatDbTc data record contains the basic properties for one satellite in the*satellite_tc* database table. The fields in the SatDbTc data record are 1:1 replicas of the SQL database record. Please refer to the database model description at section Satellite Database how relations between tables are maintained using IDs.

There may be multiple telecommand channel definitions assigned to one satellite. These channels share the same *satelliteId*. One of these channels is the default TC channel, this and only this channels has the *defaultTc* flag set *true*.

Data Model

Key	Type	Value
tcId	Number	unique, primary key, auto generated
satelliteId	Number	ID of the satellite this TC channel belongs to
frequency	Number	transmit frequency, floating point, MHz
polarization	String	this is an enumeration, may be one of "H", "V", "RHCP" or "LHCP"
defaultTc	Boolean	true means "this is the default TC channel for this satellite"
comment	String	free text describing the purpose of this channel
modified	String	timestamp, generated by database at creation / modification time

5.5.74 SatDbTmAttenuation

The SatDbTmAttenuation data record contains the attenuation values stored in the database for one particular pair of antennaId and tmId. These are the attenuation values to be applied for the given primary and secondary TM signals when received by this antenna.

Data Model

Key	Type	Value
attenuationId	Number	unique, primary key, auto generated
tmId	Number	ID of the TM signal this data record belongs to
antennaId	Number	ID of the antenna this beacon belongs to
attenuation1	Number	transmit attenuation1, floating point, dB
attenuation2	Number	transmit attenuation2, floating point, dB
modified	String	timestamp, generated by database at creation / modification time

5.5.75 SatDbTm

The SatDbTm data record contains the basic properties for one satellite in the*satellite_tm* database table. The fields in the SatDbBeacon data record are 1:1 replicas of the SQL database record. Please refer to the database model description at section Satellite Database how relations between tables are maintained using IDs.

There may be multiple telemetry channel definitions assigned to one satellite. These channels share the same *satelliteId*. One of these TM channels is the default channel, this and only this channel has the *defaultTm* flag set *true*.

Data Model

Key	Type	Value
tmId	Number	unique, primary key, auto generated
satelliteId	Number	ID of the satellite this beacon belongs to
frequency1	Number	primary receive frequency, floating point, MHz
frequency2	Number	secondary receive frequency, floating point, MHz
polarization1	String	this is an enumeration, may be one of "H", "V", "RHCP" or "LHCP"
polarization2	String	this is an enumeration, may be one of "H", "V", "RHCP" or "LHCP"
defaultTm	Boolean	true means "this is the default telemetry channel for this satellite"
comment	String	free text describing the purpose of this beacon
modified	String	timestamp, generated by database at creation / modification time

5.5.76 SatDbSatOperator

The SatDbSatOperator data record contains the basic properties for one satellite in the *satellite_operators* database table. The fields in the SatDbSatOperator data record are 1:1 replicas of the SQL database record. Please refer to the database model description at section Satellite Database how relations between tables are maintained using IDs.

There is a limited list of satellite operators, each being assigned to multiple satellites. The relation is made by the *operatorId* which is stored with each satellite.

Data Model

Key	Type	Value

Key	Type	Value
operatorId	Number	unique, primary key, auto generated
name	String	free text, company name of satellite operator
contact	String	free text, multiline, line breaks are coded as '\n' character sequences.
modified	String	timestamp, generated by database at creation / modification time

5.5.77 SatDbPosition

The SatDbPosition data record contains the antenna pointing information for one satellite and one antenna in the *satellite_positions* database table. The fields in the SatDbPosition data record are 1:1 replicas of the SQL database record. Please refer to the database model description at section Satellite Database how relations between tables are maintained using IDs.

The *Satellite-Positions* table typically contains *number-of-satellites* x *number-of-antennas* records, as each satellite requires different antenna pointing angles for each particular antenna.

Data Model

Key	Type	Value
positionId	Number	unique, primary key, auto generated
satelliteId	Number	the ID of the satellite this SatDbPointing record refers to (relates to the *satellites* table).
azimuth	Number	antenna azimuth pointing angle in degrees (floting point)
elevation	Number	antenna elevation pointing angle in degrees (floting point)
polarization	Number	antenna polarization pointing angle in degrees (floting point)
satelliteColocation	Array of Number	contains the list of satellites (referenced by their satelliteId) which are seen at the same pointing angles by this particular antenna. See remark below
targetNumber	Number	a target number to be set at some antenna controllers (integer)
antennaId	Number	the ID of the antenna this SatDbPointing record refers to (relates to the *antennas* table).

Key	**Type**	**Value**
defaultPosition	Boolean	true means "this is the default position for this antenna"
comment	String	a free text describing the purpose of this database entry
modified	String	timestamp, generated by database at creation / modification time

Remark: Depending on the dish size of an antenna, the antenna "sees" not only the satellite it is pointed to but also other satellites which are at positions in the orbit very near to the first satellite. The *satelliteColocation* list contains the satelliteIds for these satellites.

Actually this list is not maintined by the the backend, the *satelliteColocation* will always read as an empty array.

5.5.78 SatDbAntenna

The SatDbAntenna data record describes one antenna in *antennas* database table. The fields in the SatDbPosition data record are 1:1 replicas of the SQL database record. Please refer to the database model description at section Satellite Database how relations between tables are maintained using IDs.

Data Model

Key	**Type**	**Value**
antennaId	Number	the ID of the antenna this SatDbAntenna record refers to.
devicename	String	the name of the sat-nms device. controlling this antenna. The name has then name of the M&C system prepended, e.c. "VLC0001.ANTENNA-1"
alias	String	a free text describing the antenna.
modified	String	timestamp, generated by database at creation / modification time

5.5.79 SatDbTLEData

The SatDbTLEData data record contains the TLE ephemeris data for one satellite. The fields in the SatDbTLEData data record are 1:1 replicas of the SQL database record from the *norad_tle* table. Please refer to the database model description at section Satellite Database how relations between tables are maintained using IDs.

Data Model

Key	Type	Value
noradNumber	Number	the norad number of the satellite this SatDbTLEData record refers to. As the norad number is unique, this field is used as the primary key in the database table.
tleName	String	the name of the satellite as stated in the original TLE parameter set.
tleLine1	String	the first line of TLE data
tleLine2	String	the second line of TLE data
modified	String	timestamp, generated by database at creation / modification time

Remark: The TLE (aka Nasa Two Line Elements) format identifies single parameters by their exact position in the lines 1/2. This requires that the *tleLine1* and *tleLine2* fields are processed exactly as they are, there must no whitespace be added or removed from the start or end of the value, multiple blanc characters must not be compressed to one. This is essential to keep the values readable.

5.5.80 SatDbI11Data

The SatDbI11Data data record contains the Intelsat 11 parameter ephemeris data for one satellite. The fields in the SatDbI11Data data record are 1:1 replicas of the SQL database record from the *intelsat_11parameters* table. Please refer to the database model description at section Satellite Database how relations between tables are maintained using IDs.

Data Model

Key	Type	Value
i11Id	Number	a unique ID of this SatDbI11Data record (auto generated).
satelliteId	Number	the ID of the satellite this SatDbI11Data record refers to.
i11Name	String	the name of the satellite as stated in the original I11 parameter set.
i11Data	String	the 11 ephemeris parameters, floating point, separated by semicolon characters
defaultI11	Boolean	true means "this is the default I11 dataset for this satellite"
comment	String	free text describing the purpose of this dataset
modified	String	timestamp, generated by database at creation / modification time

5.5.81 SatDbTableTracking

The SatDbTableTracking data record contains the table tracking information for one satellite. The fields in the SatDbTableTracking data record are 1:1 replicas of the SQL database record from the *table_tracking* table. Please refer to the database model description at section Satellite Database how relations between tables are maintained using IDs.

Data Model

Key	Type	Value
ttId	Number	a unique ID of this SatDbTableTracking record (auto generated).
satelliteId	Number	the ID of the satellite this SatDbTableTracking record refers to.
antennaId	Number	the ID of the antenna this SatDbTableTracking record refers to.
filename	String	the name of the data file containing the position data of the satellite. Should be treated as a free text entry field in the UI.
defaultTt	Boolean	true means "this is the default table tracking dataset for this satellite"
comment	String	free text describing the purpose of this dataset
modified	String	timestamp, generated by database at creation / modification time

5.5.82 SatDbSatDetails

The SatDbSatDetails data record contains detailed information for one satellite. It is returned by a GET performed on the /api/v1/satdetails endpoint, collecting all information known about the satellite in one single document.

Data Model

Key	Type	Value
satellite	SatDbSatellite	the basic properties of the satellite
operator	SatDbSatOperator	the satellite operator assigned to this satellite
beacons	Array of SatDbBeacon	a list of all beacons defined for this satellite, may contain 0 or more elements.
tcChannels	Array of SatDbTc	a list of all telecommand channels defined for this satellite, may contain 0 or more elements.
tmChannels	Array of SatDbTm	a list of all telemetry channels defined for this satellite, may contain 0 or more elements.

Key	Type	Value
tleData	Array of SatDbTLEData	the TLE ephemeris data defined for this satellite, may contain 0 or 1 elements.
i11Data	Array of SatDbI11Data	the I11 ephemeris data defined for this satellite, may contain 0 or more elements.
tableData	Array of SatDbTableTracking	the table tracking information for this satellite, may contain 0 or more elements.

The SatDbSatDetails data record does not contain any antenna pointing information as this also depends on the antenna used to see this satellite.

5.5.83 SatDbTleImport

The SatDbTleUpload data record is used to upload a file containing a list of TLE definitions to the satellite database. It is used with the /api/v1/tleimport API endpoint.

Data Model

Key	Type	Value
fileName	String	the name of the file which is to be imported. This is used in the backend for logging etc.
replaceAll	Boolean	Controls the behavior of the import process, FALSE merges the database and file content together, overwiting data sets with the same norad number. TRUE ensures that after the import only the TLE datasets from imported file are in the database.
fileContent	String	The content of the file to be imported, BASE64 coded.

5.5.84 SatDbState

The SatDbState data record reports the state of the connection of the backend to the database server. The state may be queried by a GET /api/v1/dbstate API call.

Data Model

Key	Type	Value
connected	Boolean	true if the backend is connected to a DB server
primary	Boolean	true if the backend is connected to the primary DB server
serverAddress	String	the IP address of the DB server the backend is actually connected to

5.5.85 ChannelData

The ChannelData data record holds the information about one satellite channel stored in the satellite channel database. It is used by the /api/v1/satellites/##/channels API calls

Data Model

Key	Type	Value
channelId	Number	A unique ID of this ChannelData record (auto generated).
satelliteId	Number	The ID of the satellite this ChannelData record refers to.
name	String	The user defined channel name.
fullName	String	The full name of this channel as it appears as a selection in the user interface. The full name is built from the satellite's name, a '-' character and the *name* field in this record. The full name must be unique because the operator selects satellite channels by this name. With POST or PATCH calls to the channel database the backend ignores this field, it automatically generates the *fullName* from the satellite name and *name* .
user	String	The name of the user who lastly edited the channel
programTitle	String	Name of the program transmitted or received with this channel
comment	String	A one line comment describing this satellite channel
rxFrequency	Number	The receive frequency in MHz (3 digits precision)
rxPolarization	String	The receive polarization, one of 'X', 'Y', 'L' or 'R'
txFrequency	Number	The transmit frequency in MHz (3 digits precision)
txPolarization	String	The transmit polarization, one of 'X', 'Y', 'L' or 'R'
dvbMode	String	The DVB mode, one of 'AMCD', 'AMCDL', 'AMCDVBS', 'AMCNBC', 'AMDVBS', 'ATSC', 'AUTO', 'DSNG', 'DSS', 'DVBS', 'DVBS2', 'DVBS2X', 'DVBT', 'DVBT2', 'NS3', 'NS4' or 'TURBO'
nlcMode	String	The NLC mode, one of 'OFF' or 'ON'
symbolRate	Number	The symbol rate in Msym/sec (4 digits precision)

Key	Type	Value
fec	String	The FEC rate, one of 'AUTO', '1/1', '1/16', '1/2', '1/3', '1/32', '1/4', '1/5', '1/8', '2/2', '2/3', '2/5', '2/9', '3/16', '3/2', '3/3', '3/4', '3/5', '4/15', '4/2', '4/3', '4/5', '4/9', '5/16', '5/2', '5/6', '5/9', '6/2', '6/5', '6/7', '7/15', '7/2', '7/8', '7/9', '8/10', '8/15', '8/16', '8/2', '8/9', '9/10', '9/2', '9/20', '10/11', '10/2', '11/12', '11/15', '11/2', '11/20', '11/45', '12/2', '12/24', '13/14', '13/15', '13/18', '13/2', '13/24', '13/30', '13/45', '14/24', '14/45', '15/16', '15/25', '16/25', '17/30', '19/20', '19/30', '21/22', '21/44', '22/45', '23/36', '25/36', '26/45', '28/45', '29/45', '31/45', '32/45', '37/45', '41/5', '43/27', '44/27', '45/180', '45/28', '46/28', '47/26', '48/26', '50/15', '55/56', '60/180', '64/57', '70/140', '70/150', '72/180', '77/90', '80/100', '80/180', '90/180', '90/30', '100/180', '108/180', '114/180', '120/180', '126/112', '126/180', '135/180', '144/180', '150/180', '160/180', '162/180', '188/204', '194/178', '204/188', '208/192', '219/201', '225/205', '380/400', '442/485', '485/422', '510/511' or '510/512'
modulation	String	The modulation type, one of '128APSK', '128QAM', '16APSK', '16PSK', '16QAM', '16QUAM', '256APSK', '256QAM', '32APSK', '32QAM', '64APSK', '64QAM', '8APSK', '8PSK', '8QAM', 'AQPSK', 'AUTO', 'BPSK', 'OQPSK', 'PSK8', 'QAM', 'QAMAUTO', 'QPSK', 'SEQ', 'SOQPSK', 'SQPSK', 'UQPSK', 'VIT', 'VSB16' or 'VSB8'
rolloff	String	The roll off factor, one of '0.02', '0.05', '0.10', '0.15', '0.20', '0.25' or '0.35'
pilots	String	The pilots on/off switch, one of 'OFF' or 'ON'
bitRate	Number	The data rate in Mbit/sec (4 digits precision)
useRate	String	Selects the way the data rate is specified, one of 'SYMBOLRATE' or 'BITRATE'
packetSize	String	The packet size, one of '188' or '204'
frameSize	String	The frame size, one of 'SHORT' or 'NORMAL'
videoRate	Number	The video data rate in Mbit/sec (4 digits precision)
vrAuto	String	The video rate auto switch, one of 'FIXED' or 'AUTO'

Key	Type	Value
profile	String	The video encoding profile, one of 'MPEG2-MP@ML-420', 'MPEG2-MP@LL-420', 'MPEG2-HP@ML-422', 'MPEG2-HP@LL-422', 'MPEG2-SP@LL-420', 'MPEG2-SP@ML-420', 'MPEG2-MP@H14-420', 'MPEG2-MP@HL-420', 'MPEG2-HP@H14-422', 'MPEG2-HP@HL-422', 'MPEG2-HP@HL-420', 'MPEG2-HP@H14-420', 'MPEG2-HP@ML-420', 'MPEG4-BP@L2-420', 'MPEG4-BP@L3-420', 'MPEG4-BP@L4-420', 'MPEG4-MP@L3-420', 'MPEG4-MP@L4-420', 'MPEG4-MP@L4.2-420', 'MPEG4-HP@L3-420', 'MPEG4-HP@L4-420', 'MPEG4-HP@L4.2-420', 'MPEG4-HP@L3-422', 'MPEG4-HP@L4-422' or 'MPEG4-HP@L4.2-422'
eirp	Number	The transmit EIRP in dBW (2 digits precision)
redEirp	Number	The reduced EIRP for line up in dBW (2 digits precision)
autoLineup	String	Configures automatic line up, one of 'OFF' or 'ON'
audioRate1	String	Audio channel 1 data rate, one of 'DISABLED', '64k', '96k', '128k', '160k', '192k', '224k', '256k', '320k' or '384k"
audioRate2	String	Audio channel 2 data rate, one of 'DISABLED', '64k', '96k', '128k', '160k', '192k', '224k', '256k', '320k' or '384k"
audioRate3	String	Audio channel 3 data rate, one of 'DISABLED', '64k', '96k', '128k', '160k', '192k', '224k', '256k', '320k' or '384k"
audioRate4	String	Audio channel 4 data rate, one of 'DISABLED', '64k', '96k', '128k', '160k', '192k', '224k', '256k', '320k' or '384k"
audioChannels	Number	Number of audio channels, (integer, 1-4)
decoderInput	String	Decoder input, one of 'SAT', 'IP' or 'ASI'
ipType	String	IP adressing type, one of 'UNICAST' or 'MULTICAST'
ipProtocol	String	IP protocol, one of 'UDP', 'RTP', 'TCP' or 'ZIXI'
ipSourceAddress	String	Source IP address in dotted quad notation
ipSourcePort	Number	Source port number (integer, 0-65535)
ipDestinationAddress	String	Destination IP address in dotted quad notation

Key	Type	Value
ipDestinationPort	Number	Destination port number (integer, 0-65535)
ipFec	String	IP FEC selection, one of 'OFF' or 'ON'
ipBufferSize	Number	Buffer size (integer)
encoderPhysInterface	Number	Encoder physical output number (integer, 1-4)
decoderPhysInterface	Number	Decoder physical input number (integer, 1-4)
modified	String	timestamp, generated by database at creation / modification time

5.5.86 StreamKeyData

The StreamKeyData data record holds the information about one BISS or SRT key in the *streamkeys* table of the database . It is used by the /api/v1/streamkeys API calls.

Data Model

Key	Type	Value
keyId	Number	A unique ID of this StreamKeyData record (auto generated).
name	String	The user defined name of this stream key. This name also must be unique because the operator selects the key by this name.
keyType	String	The key type, one of 'BISS-1', 'BISS-E' or 'PASSPHRASE'.
key	String	The key value associated with the id and name in this record. For biss keys this is a string entirely consisting of the hexadeximal characters 0123456789ABCDEF, it is either 12 or 16 characters long. For other key types the key is free text.
modified	String	timestamp, generated by database at creation / modification time

5.5.87 TreeViewNode

The TreeViewNode data record holds the information about one node in the sat-nms tree view and the tree below this node. The complete tree is partially autogenerated and user defined, see below about the standard tree structure used in the sat-nms software.

Data Model

Key	Type	Value

Key	Type	Value
nodeName	String	The (displayed) name of this node. This may be a M&C name, a subsystem name, a device name or a screen name, depending on the type of this node. See below on details about node types and names.
nodeType	String	One of ROOT, MNCNODE, DEVICELIST, SUBSYSTEM or DEVICE. See below on details about node types and names.
children	Array	An array of other TreeViewNode objects being the children of this node. May be null, this denotes a leaf node.

Standard Tree Structure

The general structure of the tree view ist autogenerated by the software. User defined branches with subsystem definitions are integrated at certain points of the tree.

```
ROOT
  +-- MNC001
  +-- MNC002
    +-- Devices
    +-- SUBSYSTEM-1
    +-- SUBSYSTEM-2
      +-- SUBSUBSYSTEM-1
        +-- DEVICE-1
        +-- DEVICE-2
    +-- DEVICE-3
    +-- DEVICE-4
  +-- MNC003
```

A tree always consists of one ROOT node. The children of this node are the M&C systems managed by the backend (one MNCNODE for each M&C system). Each MNCNODE has at least one child, a DEVICELIST node containing the list of devices managed by this M&C. Up to here all nodes are auto-generated and not c hangeable by the user. Additionally, each MNCNODE may have zero or more SUBSYSTEM nodes with user defined groups of devices or nested subsystems.

Node Types and Names

A TreeViewNode may be one of the types ROOT, MNCNODE, DEVICELIST, SUBSYSTEM or DEVICE. The table below lists the properties of these node types in short form:

type	***auto-generated***	***may contain***	***name***
ROOT	yes	MNCNODE	always 'TreeView'

type	***auto-generated***	***may contain***	***name***
MNCNODE	yes	DEVICELIST, SUBSYSTEM	M&C name
DEVICELIST	yes	DEVICE	always 'Devices'
SUBSYSTEM	no	SUBSYSTEM, DEVICE	subsystem name
DEVICE	within a DEVICELIST	-/- (leaf node)	device name

The **ROOT** node is - as the name suggests - the root of the tree view tree. There is exactly one **ROOT** node in the tree, its children are one or more**MNCNODE** nodes, as many as M&C systems are configured to be managed by the backend. The displayed name of each **MNCNODE** node id the ID / name of this M&C, this is the same as used in message IDs. The list of **MNCNODE** nodes is auto-generated by the backend, derived from its configuration data.

The first child of each**MNCNODE** node always is a **DEVICELIST** node containing the complete list of devices configured on this M&C. This node also is automatically created by the backend, the device list is in 'natural' order, meaning that the devices appear in the same order as they do in the M&C configuration file / screen.

DEVICE nodes are the leaf nodes of the tree. The name of a**DEVICE** node is the name of the device without the name of the M&C it is contained in.

SUBSYSTEM nodes define a group of devices or nested subsystems. They are the only user defined node objects in the tree. The name of a **SUBSYSTEM** node is user defined, but as this name appears in the message ID of the summary fault state of a subsystem, the same restrictions applay for subsystem names as for device names. They may consist of upper case letters 'A' .. 'Z', digits and the '-' - character.

Subsystems

Subsystems defined by a **SUBSYSTEM** node in the tree are managed by the M&C like a virtual device which has a summary fault flag which reads as the maximum fault condition that appears at one of the elements the subsystem contains. The definition of all configured subsystems in a M&C is stored by the M&C itself, not by the backend. Due to the M&C-local architecture of subsystems the following rules apply to them:

The name of a subsystem consists of the same limited set of characters like device names do.

A subsystem may contain devices or other subsystems. These children must be located at the same M&C as their parent subsystem. Subsystems cannot be defined across multiple M&Cs.

Each subsystem distributes a summary fault variable which can be used to display the fault state of the subsystem. The full messgeId of this variable is

```
{M&C-name}.SUBSYSTEM.{subsystem-name}.fault
```

or for nested subsystems

`{M&C-name}.SUBSYSTEM.{subsystem-name}.{sub-subsystem-name}....fault`

These variables - like a .fault value of a device - read one of the strings "OK.", "WARNING" or "Summary FAULT".

5.5.88 DatabaseVersion

The DatabaseVersion reports the actual version (migration_step) of the database together with the version required by the backend and a fault flag indicating that the database version is too old for this backend.

Data Model

Key	Type	Value
reported	Number	The version / migration_step reported by the database. a value of -1 means 'undefined'
required	Number	The version / migration_step required by the backend
fault	Boolean	True of the database version is too old

5.5.89 DebugMessage

The DebugMessage data record is used for the communication with a M&C debug terminal. It transports the lines to be displayed in the terminal via the websocket and also the commands to be sent to the debug terminal using the /api/v1/debug API call the other way round.

Data Model

Key	Type	Value
msg	String	The line to be transferred (without any newline character at the end)
error	Boolean	True if this is an error message issued by the backend rather than a message from the M&C

5.5.90 RedundancyState

The RedundancyState data record carries the actual state of a particular M&C redundancy switching logic. It is returned by a GET /api/v1/redundancy API call.

Data Model

Key	Type	Value

Key	Type	Value
mncName	String	The name of the redundant M&C pair for which this data record reports the state
enabled	Boolean	Reports if this redundancy is enabled (true/false)
state	String	Reports the actual state of the redudancy switching logic, one of 'PRIMARY', 'BACKUP', 'NONE'

5.5.91 RedundancySummary

The RedundancySummary data record carries the actual state of the M&C redundancy switching logic for all redundant M&Cs. It is returned by a GET /api/v1/redundancy API call.

Data Model

Key	Type	Value
redundantMnCs	Array	An array of RedundancyState objects, reporting the states to all redundant M&Cs. The array may be empty if no redundant M&Cs are configured in the backend.

5.5.92 RedundancyCmd

The RedundancyCmd data record carries the command to be executed with a /api/v1/redundancy API call.

Data Model

Key	Type	Value
cmd	String	The command to be executed. Valid commands are: 'ENABLE' = enable redundancy switching in case of a communication fault. 'DISABLE' = disable redundancy switching. 'RESET' = switch back to the primary IP address for the M&C 'FORCE' = force a redundancy switch to the backup IP address for the M&C even if the communication at the primary IP address works fine.

5.5.93 RestartCmd

The RestartCmd data record carries the command to be executed with a/api/v1/restart API call.

Data Model

Key	Type	Value

Key	Type	Value
cmd	String	The command to be executed. The only valid command is "RESTART". The restart is only executed if *cmd* has this value.

5.5.94 ScheduleEvent

The ScheduleEvent data record carries the data of one event in the M&C macro scheduler.

Data Model

Key	Type	Value
eventId	Number	An integer value in the range 1 .. 2147483647 uniquely identifying this event in the schedule. A value of 0 means undefined.
groupId	Number	The event's group identifyer. this is the *eventId* if the master event in the group. Even this master event has its own *eventId* set as *groupId*, a *groupId* of 0 means this event is not grouped.
state	String	One of the following: DONE, UPCOMING or MISSED
repeatMode	String	One of the following: NONE, DAILY, WEEKLY or MONTHLY
enabled	Boolean	*true* enables the execution of the event's macro, *false* inhibits this.
description	String	A description for this event given by the operator.
firstExecution	String	The time of the first execution of the event.
nextExecution	String	The time of the next execution of the event.
lastExecution	String	The date of the last execution of the event if a repeat mode is set. For non-repeating events this reads null. The time stamp should contain a time (e.g. 00:00:00) although the scheduler ignores this time.
macro	String	The macro to be executed. newline characters on the macro thave to written as the character sequence "\n".
dict	Array	An optional array of KeyValPair definitions which may be used to append additional information to the event.

Remarks

- New events shall use the lowest *eventId* above 1000 which is not used in the schedule.
- All time stamps are formatted as 'YYYY-MM-DDTHH:MM:SSZ' ans expressed in UTC.

- In the macro text newline characters are expressed as a "\n" character sequence.
- The fields *groupId* and *dict* are not used by the macro scheduler appliaction. The Web GUI shall not show these fields snd shall not present them for editing.

5.5.95 Schedule

The Schedule data record carries the complete shedule of an M&C macro scheduler. It is used with the /api/v1/schedule API call.

Data Model

Key	Type	Value
events	Array	An array of ScheduleEvent data objects defining the complete schedule.

5.5.96 DeviceDefinition

The DeviceDefinition data record carriers the information about one device in an M&C's device setup.

Key	Type	Value
name	String	The name of the device
driver	String	The device driver

5.5.97 DeviceThreadDefinition

The DeviceThreadDefinition data record carriers the information about one device thread in an M&C's device setup.

Key	Type	Value
port	String	The communication port / interface this thread uses
protocol	String	The protocol this thread uses
idle	Number	The idle time of this thread (msecs)
devices	Array	An array of DeviceDefinition objects containing the devices of this thread

5.5.98 DeviceSetup

The DeviceSetup data record carriers the complete information of an M&C's device setup.

Key	Type	Value
comment	String	A short one line description (e.g. time stamp an user name of the last change)
threads	Array	An array of DeviceThreadDefinition objects containing the device thread definitions of this setup

5.5.99 FRViewProperties

The FRViewProperties data record contains the setting stored for a file recorder viewer.

Key	Type	Value
title	String	The diagram title
traces	Array	An array of FRViewTraceDescription objects containing the trace names and unit strings. The File-Recorder device records four traces of data, so this array always contains four elements. When sent to the backend also four elements must be supplied.
presets	Array	An array of FRViewTracePreset objects containing the scaling presets defined for this file recorder view. This array always contains eight elements, when sent to the backend also eight elements must be supplied. In the Java-UI the presets are identified by position in this array, so the front end should not re-order them.

5.5.100 FRViewTraceDescription

The FRViewTraceDescription data record contains a basic description (label, unit) for one trace within the FRViewProperties data record.

Key	Type	Value
unit	String	The unit string to be displayed at the axis for this trace
name	String	The name of the trace

5.5.101 FRViewPreset

The FRViewPreset data record contains the data stored for one user preset within the FRViewProperties data record. This includes the diagram scaling and some additional information like trace colors etc.

Key	Type	Value

Key	Type	Value
absstart	String	The (absolute) starttime of the diagram, expressed as an ISO 8601 string, example: `2021-03-26T08:02:30Z` , time zone is always UTC.
relstart	Number	The starttime of the diagram, in seconds relative to the newest data record recorded.
tscale	String	The time axis scaling, one of ("15s", "30s", "1m", "3m", "5m", "10m", "30m", "1h", "3h", "6h", "12h", "1d", "2d", "6d", "30d"). The values are per division of the time axis
traces	Array	An array of FRViewTraceScaling objects, each describing the scaling properties for one diagram trace. The backend will always report four traces, even if not all four traces are configured in the File-Recorder device. When sent to the backend, each preset **must** provide four elements in this array to be recognized properly.
name	String	The (optional) name of the preset

5.5.102 FRViewTraceScaling

The FRViewTraceScaling data record contains the scaling information for one diagram trace within the FRViewPreset data record.

Key	Type	Value
yref	Number	The reference value (the value shown at the half diagram height)
yscale	Number	The Y-scale value (y-units / division with 10 divisions diagram height). The values must match a 1/2/5 stepping, they will be read back rounded to this if other values are supplied.
enabled	Boolean	TRUE means this trace shall be shown
color	String	The trace color, expressed as a hexadecimal RGB value preceeded by a '#' character

5.5.103 AcuTarget

The AcuTarget data model contains the basic properties (name, az, el, pol) for one target / satellite stored on an ACU device

Key	Type	Value
targetNo	Number	The target number on the ACU

Key	Type	Value
satelliteName	String	The name of the target
azimuth	Number	The stored azimuth angle
elevation	Number	The stored elevation angle
polarization	Number	The stored polarization angle

5.5.104 AcuTargetList

The AcuTargetList data model contains the target list read from an ACU device, showing all used target memories as an array of AcuTarget objects.

Key	Type	Value
acuName	String	Name of the ACU device
acuTargets	Array	Target list as an array ofAcuTarget objects.

5.5.105 InventoryItem

The InventoryItem data record contains the data stored for one particular item in the inventory database. It is used to report, add or modify the properties of an inventory item thru the API. Beside this, the InventoryItem record is used to specify a filter when getting a list of InventoryItem records. in such a case, all fields which are missing or null are interpreted as a wildcard, specified fields must match to include the item in the list.

Key	Type	Value
itemId	Number	A unique ID of this record (auto generated).
serialNo	String	The serial number of the item / device
vendor	String	The vendor name (for serach filters)
model	String	The modem name (for serach filters)
state	String	The administrative state of the device
comment	String	free text
modified	TimeStamp	automatically created timestamp of last change

5.5.106 InventoryDevice

The InventoryDevice data record contains the device mappings for a InventoryItem. There may be multiple devices mapped to one item in the inventory database.

Key	Type	Value
deviceId	Number	A unique ID of this record (auto generated).
itemId	Number	The ID of the InventoryItem this record refers to
mncName	String	The serial number of the item / device
deviceName	String	The vendor name (for serach filters)
state	String	The administrative state of the device
online	Boolean	true means the device is present
modified	TimeStamp	automatically created timestamp of last change

5.5.107 InventoryLogEntry

The InventoryLogEntry data record contains the data of one entry in the inventory life cycle log. It is used to report, add or modify the properties of an inventory log entry the API. Beside this, the InventoryLogEntry record is used to specify a filter when getting a list of InventoryLogEntry records. in such a case, all fields which are missing or null are interpreted as a wildcard, specified fields must match to include the item in the list.

Key	Type	Value
entryId	Number	A unique ID of this record (auto generated).
itemId	Number	Links to the inventory_items entry this log entry refers to.
mncName	String	The M&C which issued this message or null if the message was entered at the UI
devName	String	The satnms device name which issued this message
message	String	the message text
created	TimeStamp	timestamp, generated by database at creation time

5.5.108 DocumentItem

The DocumentItem data record contains the data of one entry in the backend's document list. It is used to report, add or modify the properties of a stored document thru the API. Beside this, the InventoryLogEntry record is used to specify a filter when getting a list of DocumentItem records. in such a case, all fields which are missing or null are interpreted as a wildcard, specified fields must match to include the item in the list.

Key	Type	Value
documentId	Number	A unique ID of this record (auto generated).
fileName	String	Name of the file. The name must be unique, all files are stored in the same directory.
contentType	String	Content type of the file. must be one of application/pdf, image/png or image/jpg.
fileContent	String	The file content as a BASE64 coded string
driver	String	The satnms driver name this document refers to or null if it does not refer to a driver / device type.
mncName	String	The M&C name this document refers to or null if it does not refer to a M&C.
comment	String	free text
modified	TimeStamp	automatically created timestamp of last change

5.5.109 DocumentList

The DocumentList data record contain a list of DocumentItem records.

Key	Type	Value
documents	Array	An array of DocumentItem records.

5.5.110 Thumbnail

The Thumbnail data record contains the data fro a thumbnail image when fetched using the /api/v1/thumbnail API call.

Key	Type	Value
messageId	String	The full message ID of the thumbnail image
image	String	The binary thumbnail image (jpf or png), base64 encoded.

5.5.111 BackendInfo

Key	Type	Value

Key	Type	Value
backendName	String	The current backend name
role	String	Role of current backend
isPrimary	Boolean	Whether the backend is main or not
backendNames	String []	Array of backend names (string) to set the activeBackend name

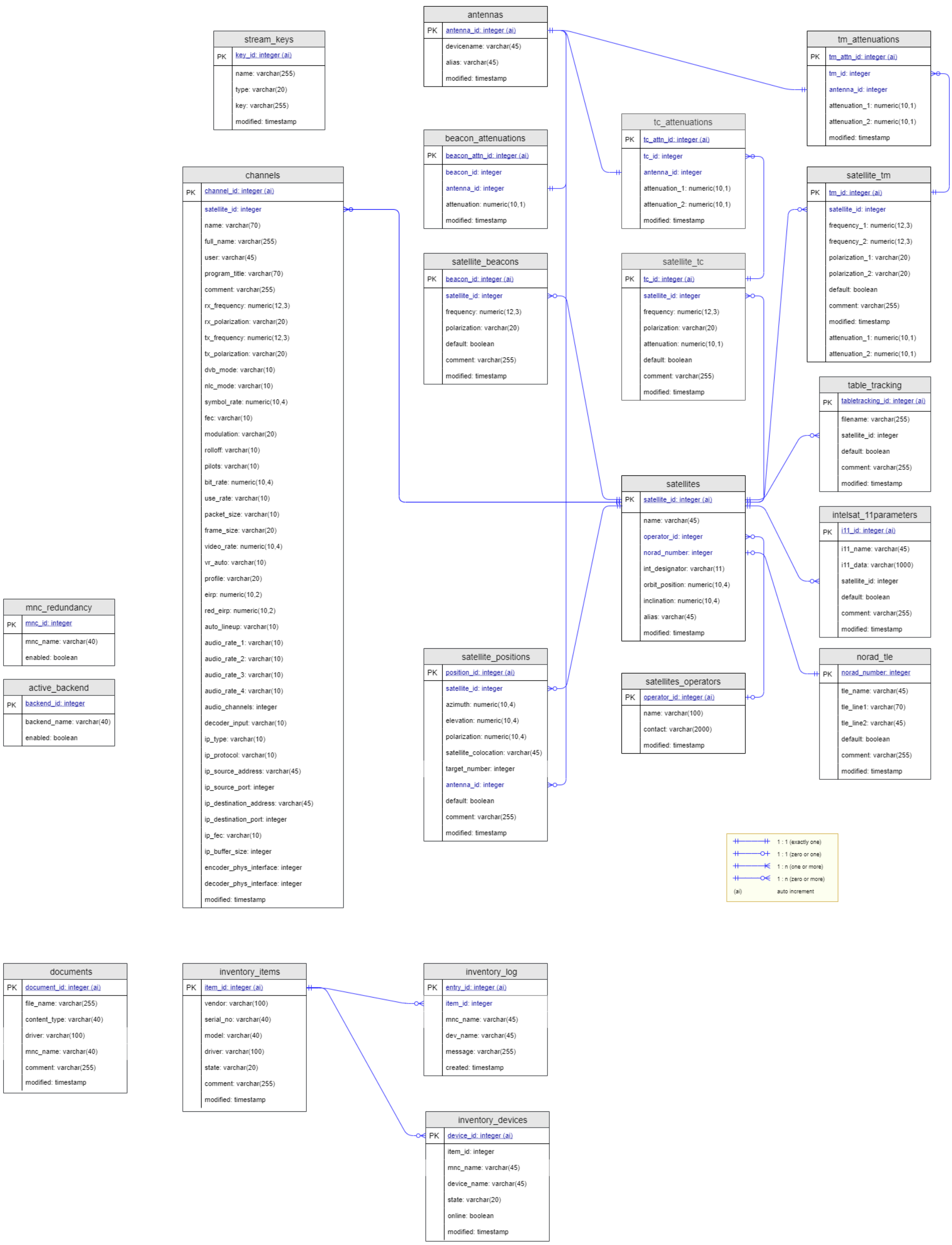
5.5.112 UISettings

The User-Settings data record contains changed settings by the user. Not saved or changed settings will be generated from Backend.

Key	Type	Value
settingsId	Integer	A Unique Id of an setting
key	String	Setting name
value	TEXT	The saved value of the setting
username	String	Username to which the setting belongs
isLocked	Boolean	Whether the setting is locked for user to overwrite it
modified	TimeStamp	automatically created timestamp of last change

5.6 Satellite Database

The satnms4 backend stores all information about satellites, antenna pointing, ephemeris data etc. In a SQL database which is accessible through the backend's API. The following paragraphs describe the table structure of the database and the meaning of each particular table column as an appendix to the API description given above.


stream_keys
PK key_id: integer (ai)
name: varchar(255)
type: varchar(20)
key: varchar(255)
modified: timestamp
antennas
PK antenna_id: integer (ai)
devicename: varchar(45)
alias: varchar(45)
modified: timestamp
tm_attenuations
PK tm_attn_id: integer (ai)
tm_id: integer
antenna_id: integer
attenuation_1: numeric(10,1)
attenuation_2: numeric(10,1)
modified: timestamp
beacon_attenuations
PK beacon_attn_id: integer (ai)
beacon_id: integer
antenna_id: integer
attenuation: numeric(10,1)
modified: timestamp
tc_attenuations
PK tc_attn_id: integer (ai)
tc_id: integer
antenna_id: integer
attenuation_1: numeric(10,1)
attenuation_2: numeric(10,1)
modified: timestamp
channels
PK channel_id: integer (ai)
satellite_id: integer
name: varchar(70)
full_name: varchar(255)
user: varchar(45)
program_title: varchar(70)
comment: varchar(255)
rx_frequency: numeric(12,3)
rx_polarization: varchar(20)
tx_frequency: numeric(12,3)
tx_polarization: varchar(20)
dvb_mode: varchar(10)
nlc_mode: varchar(10)
symbol_rate: numeric(10,4)
fec: varchar(10)
modulation: varchar(20)
rolloff: varchar(10)
pilots: varchar(10)
bit_rate: numeric(10,4)
use_rate: varchar(10)
packet_size: varchar(10)
frame_size: varchar(20)
video_rate: numeric(10,4)
vr_auto: varchar(10)
profile: varchar(20)
eirp: numeric(10,2)
red_eirp: numeric(10,2)
auto_lineup: varchar(10)
audio_rate_1: varchar(10)
audio_rate_2: varchar(10)
audio_rate_3: varchar(10)
audio_rate_4: varchar(10)
audio_channels: integer
decoder_input: varchar(10)
ip_type: varchar(10)
ip_protocol: varchar(10)
ip_source_address: varchar(45)
ip_source_port: integer
ip_destination_address: varchar(45)
ip_destination_port: integer
ip_fec: varchar(10)
ip_buffer_size: integer
encoder_phys_interface: integer
decoder_phys_interface: integer
modified: timestamp
satellite_tm
PK tm_id: integer (ai)
satellite_id: integer
frequency_1: numeric(12,3)
frequency_2: numeric(12,3)
polarization_1: varchar(20)
polarization_2: varchar(20)
default: boolean
comment: varchar(255)
modified: timestamp
attenuation_1: numeric(10,1)
attenuation_2: numeric(10,1)
satellite_beacons
PK beacon_id: integer (ai)
satellite_id: integer
frequency: numeric(12,3)
polarization: varchar(20)
default: boolean
comment: varchar(255)
modified: timestamp
satellite_tc
PK tc_id: integer (ai)
satellite_id: integer
frequency: numeric(12,3)
polarization: varchar(20)
attenuation: numeric(10,1)
default: boolean
comment: varchar(255)
modified: timestamp
table_tracking
PK tabletracking_id: integer (ai)
filename: varchar(255)
satellite_id: integer
default: boolean
comment: varchar(255)
modified: timestamp
satellites
PK satellite_id: integer (ai)
name: varchar(45)
operator_id: integer
norad_number: integer
int_designator: varchar(11)
orbit_position: numeric(10,4)
inclination: numeric(10,4)
alias: varchar(45)
modified: timestamp
intelsat_11parameters
PK i11_id: integer (ai)
i11_name: varchar(45)
i11_data: varchar(1000)
satellite_id: integer
default: boolean
comment: varchar(255)
modified: timestamp
mnc_redundancy
PK mnc_id: integer
mnc_name: varchar(40)
enabled: boolean
satellite_positions
PK position_id: integer (ai)
satellite_id: integer
azimuth: numeric(10,4)
elevation: numeric(10,4)
polarization: numeric(10,4)
satellite_colocation: varchar(45)
target_number: integer
antenna_id: integer
default: boolean
comment: varchar(255)
modified: timestamp
satellites_operators
PK operator_id: integer (ai)
name: varchar(100)
contact: varchar(2000)
modified: timestamp
norad_tle
PK norad_number: integer
tle_name: varchar(45)
tle_line1: varchar(70)
tle_line2: varchar(45)
default: boolean
comment: varchar(255)
modified: timestamp
active_backend
PK backend_id: integer
backend_name: varchar(40)
enabled: boolean
1 : 1 (exactly one)
1 : 1 (zero or one)
1 : n (one or more)
1 : n (zero or more)
(ai) auto increment
documents
PK document_id: integer (ai)
file_name: varchar(255)
content_type: varchar(40)
driver: varchar(100)
mnc_name: varchar(40)
comment: varchar(255)
modified: timestamp
inventory_items
PK item_id: integer (ai)
vendor: varchar(100)
serial_no: varchar(40)
model: varchar(40)
driver: varchar(100)
state: varchar(20)
comment: varchar(255)
modified: timestamp
inventory_log
PK entry_id: integer (ai)
item_id: integer
mnc_name: varchar(45)
dev_name: varchar(45)
message: varchar(255)
created: timestamp
inventory_devices
PK device_id: integer (ai)
item_id: integer
mnc_name: varchar(45)
device_name: varchar(45)
state: varchar(20)
online: boolean
modified: timestamp

5.6.1 Table Satellites (satellites)

All satellites known to this system. Number may vary from a dozen to hundreds.

Field	null	Description
satellite_id	no	unique, primary key, auto increment
name	no	free text, typically the name as given by satellite operator
operator_id	yes	1:1 relation to table *satellite-operators*
norad_number	yes	unique name (world wide standardized), 1:1 relation to table *norad_tle*
int_designator	yes	unique name (world wide standardized)
orbit_position	yes	satellite orbit position in degrees
inclination	yes	satellite inclination in degrees
alias	yes	free text, can be used by customer to add own name
modified	no	automatically created timestamp of last change

5.6.2 Table Satellite Operators (satellites_operators)

All satellites operators known to this system. Usually only a small number (2-10) but not limited to this.

Field	null	Description
operator_id	no	unique, primary key, auto generated
name	no	free text, company name of satellite operator
contact	yes	free text, multiline, hotline phone, e-mail etc.
modified	no	automatically created timestamp of last change

5.6.3 Antennas (antennas)

All antennas known to this system. Usually only a small number (2-10) but not limited to this.

Field	null	Description
antenna_id	no	unique, primary key, auto generated

Field	null	Description
devicename	no	free text, max. 11 characters (at the moment) because it is a device name from the MNC system
alias	yes	free text, can be used by customer to add own name
modified	no	automatically created timestamp of last change

5.6.4 Table Satellite Beacons (satellite_beacons)

All satellites beacons known to this system. Normally every satellite has one or more beacon signals, but it is not necessary to store these information because its only needed if you have to track the satellite. So this table could be completely empty.

Field	null	Description
beacon_id	no	unique, primary key, auto generated
satellite_id	no	ID of the satellite this beacon belongs to
frequency	no	beacon frequency, 3 digits precision (MHz)
polarization	no	integer coded receive polarization
attenuation	yes	an attenuation value to be set at the beacon receiver
default	no	boolean, true meaning this is the default beacon for this satellite
modified	no	automatically created timestamp of last change

5.6.5 Table Satellite TC (satellite_tc)

All satellite telecommand frequencies / channels known to this system. Every satellite may have one or more TC channels, but these are only required if the application deals with satellite TT&C.

Field	null	Description
tc_id	no	unique, primary key, auto generated
satellite_id	no	ID of the satellite this telecommand channel belongs to
frequency	no	TC signal frequency, 3 digits precision (MHz)
polarization	no	integer coded transmit polarization
attenuation	no	an attenuation value to be set at the TC transmitter

Field	null	Description
default	no	boolean, true meaning this is the default beacon for this satellite
modified	no	automatically created timestamp of last change

5.6.6 Table Satellite TM (satellite_tm)

All satellite telemetry frequencies / channels known to this system. Every satellite may have one or more TM channels, but these are only required if the application deals with satellite TT&C.

Field	null	Description
tm_id	no	unique, primary key, auto generated
satellite_id	no	ID of the satellite this telemetry channel belongs to
frequency_1	no	primary receive frequency, 3 digits precision (MHz)
frequency_2	no	secondary receive frequency, 3 digits precision (MHz)
polarization_1	no	integer coded primary receive polarization
polarization_2	no	integer coded secondary receive polarization
attenuation_1	no	attenuation value to be set at the primary telemetry receiver
attenuation_2	no	attenuation value to be set at the secondary telemetry receiver
default	no	boolean, true meaning this is the default beacon for this satellite
modified	no	automatically created timestamp of last change

5.6.7 Table Satellite Positions (satellite_positions)

This tables stores an antenna position (Azimuth/Elevation/Polarization) for a given Satellite and a specific Antenna.

Field	null	Description
position_id	no	unique, primary key, auto generated
satellite_id	no	ID of the satellite this position belongs to
azimuth	yes	antenna azimuth position in degrees
elevation	yes	antenna elevation position in degrees

Field	null	Description
polarization	yes	antenna polarization position in degrees
satellite-colocation	yes	TODO we need to recap how this field will be used, for now its just free text
target_number	yes	a target number for this satellite to command at the antenna controller (device specific)
antenna_id	no	id of antenna, 1:1 relation to *antenna* table
default	no	boolean, true meaning this is the default satellite position
modified	no	automatically created timestamp of last change

Should later provide a dropdown list with the valid antenna (device) names.

5.6.8 Table Table Tracking (table_tracking)

This tables stores "table tracking" files names for satellites. It could have none or one filename per satellite.

Field	null	Description
tabletracking_id	no	unique, primary key, auto generated
filename	no	free text
satellite_id	no	ID of the satellite this tracking table belongs to
default	no	boolean, true meaning this is the default table track
comment	yes	free text
modified	no	automatically created timestamp of last change

5.6.9 Table Intelsat 11 Parameters (intelsat_11parameters)

This tables stores "Intelsat 11 Parameter Sets" for satellites. It could have none or one entry per satellite.

Field	null	Description
i11_id	no	unique, primary key, auto generated
satellite_id	no	ID of the satellite these I11 parameters belong to
i11_name	no	free text, name of I11

Field	null	Description
i11_data	no	free text, (raw) data of I11 (not split into 11 single parameters)
default	no	boolean, true meaning this is the default table track
comment	yes	free text
modified	no	timestamp, generated by database at creation time

5.6.10 Table Norad TLE (norad_tle)

This tables stores "Norad Two Line Elements (TLE) " for satellites. It could have none or one entry per satellite.

Field	null	Description
norad_number	no	unique Norad Number, primary key, defined by the satellite
tle_name	no	free text, name of TLE
tle_line1	no	free text, first line of TLE
tle_line2	no	free text, second line of TLE
default	no	boolean, true meaning this is the default table track
comment	yes	free text
modified	no	timestamp, generated by database at creation time

5.6.11 Table Channels (channels)

This tables stores satellite channel records as described below.

Field	null	Description
channel_id	no	A unique ID of this ChannelData record (auto generated).
satellite_id	no	The ID of the satellite this ChannelData record refers to.
name	no	The user defined channel name.

Field	null	Description
full_name	no	The full name of this channel as it appears as a selection in the user interface. The full name is built from the satellite's name, a '-' character and the *name* field in this record. The full name must be unique because the operator selects satellite channels by this name. With POST or PATCH calls to the channel database the backend ignores this field, it automatically generates the *fullName* from the satellite name and *name* .
user	yes	The name of the user who lastly edited the channel
program_title	yes	Name of the program transmitted or received with this channel
comment	yes	A one line comment describing this satellite channel
rx_frequency	yes	The receive frequency in MHz (3 digits precision)
rx_polarization	yes	The receive polarization, one of 'X', 'Y', 'L' or 'R'
tx_frequency	yes	The transmit frequency in MHz (3 digits precision)
tx_polarization	yes	The transmit polarization, one of 'X', 'Y', 'L' or 'R'
dvb_mode	yes	The DVB mode, one of 'DVBS', 'DVBS2', 'NS3' or 'NS4'
nlc_mode	yes	The NLC mode, one of 'OFF' or 'ON'
symbol_rate	yes	The symbol rate in Msym/sec (4 digits precision)
fec	yes	The FEC rate, one of 'AUTO', '1/2', '1/3', '1/4', '2/3', '3/4', '2/5', '3/5', '4/5', '5/6', '6/7', '7/8', '8/9' or '9/10'
modulation	yes	The modulation type, one of 'BPSK', 'QPSK', 'OQPSK', '8PSK', '16PSK', '16QAM', '64QAM', '256QAM', '16APSK' or '32APSK'
rolloff	yes	The roll off factor, one of '0.02', '0.05', '0.10', '0.15', '0.20', '0.25' or '0.35'
pilots	yes	The pilots on/off switch, one of 'OFF' or 'ON'
bit_rate	yes	The data rate in Mbit/sec (4 digits precision)
use_rate	yes	Selects the way the data rate is specified, one of 'SYMBOLRATE' or 'BITRATE'
packet_size	yes	The packet size, one of '188' or '204'

Field	null	Description
frame_size	yes	The frame size, one of 'SHORT' or 'NORMAL'
video_rate	yes	The video data rate in Mbit/sec (4 digits precision)
vr_auto	yes	The video rate auto switch, one of 'FIXED' or 'AUTO'
profile	yes	The video encoding profile, one of 'MPEG2-MP@ML-420', 'MPEG2-MP@LL-420', 'MPEG2-HP@ML-422', 'MPEG2-HP@LL-422', 'MPEG2-SP@LL-420', 'MPEG2-SP@ML-420', 'MPEG2-MP@H14-420', 'MPEG2-MP@HL-420', 'MPEG2-HP@H14-422', 'MPEG2-HP@HL-422', 'MPEG2-HP@HL-420', 'MPEG2-HP@H14-420', 'MPEG2-HP@ML-420', 'MPEG4-BP@L2-420', 'MPEG4-BP@L3-420', 'MPEG4-BP@L4-420', 'MPEG4-MP@L3-420', 'MPEG4-MP@L4-420', 'MPEG4-MP@L4.2-420', 'MPEG4-HP@L3-420', 'MPEG4-HP@L4-420', 'MPEG4-HP@L4.2-420', 'MPEG4-HP@L3-422', 'MPEG4-HP@L4-422' or 'MPEG4-HP@L4.2-422'
eirp	yes	The transmit EIRP in dBW (2 digits precision)
red_eirp	yes	The reduced EIRP for line up in dBW (2 digits precision)
auto_lineup	yes	Configures automatic line up, one of 'OFF' or 'ON'
audio_rate_1	yes	Audio channel 1 data rate, one of 'DISABLED', '64k', '96k', '128k', '160k', '192k', '224k', '256k', '320k' or '384k"
audio_rate_2	yes	Audio channel 2 data rate, one of 'DISABLED', '64k', '96k', '128k', '160k', '192k', '224k', '256k', '320k' or '384k"
audio_rate_3	yes	Audio channel 3 data rate, one of 'DISABLED', '64k', '96k', '128k', '160k', '192k', '224k', '256k', '320k' or '384k"
audio_rate_4	yes	Audio channel 4 data rate, one of 'DISABLED', '64k', '96k', '128k', '160k', '192k', '224k', '256k', '320k' or '384k"
audio_channels	yes	Number of audio channels, (integer, 1-4)
decoder_input	yes	Decoder input, one of 'SAT', 'IP' or 'ASI'
ip_type	yes	IP addressing type, one of 'UNICAST' or 'MULTICAST'
ip_protocol	yes	IP protocol, one of 'UDP', 'RTP', 'TCP' or 'ZIXI'

Field	null	Description
ip_source_address	yes	Source IP address in dotted quad notation
ip_source_port	yes	Source port number (integer, 0-65535)
ip_destination_address	yes	Destination IP address in dotted quad notation
ip_destination_port	yes	Destination port number (integer, 0-65535)
ip_fec	yes	IP FEC selection, one of 'OFF' or 'ON'
ip_buffer_size	yes	Buffer size (integer)
encoder_phys_interface	yes	Encoder physical output number (integer, 1-4)
decoder_phys_interface	yes	Decoder physical input number (integer, 1-4)
modified	no	automatically created timestamp of last change

5.6.12 Table StreamKeys (stream_keys)

This table stores stream keys (BISS keys, SRT passwords) as described below.

Field	null	Description
key_id	no	A unique ID of this StreamKeyData record (auto generated).
name	no	The user defined name of this stream key. This name also must be unique because the operator selects the key by this name.
type	no	The key type, one of 'BISS-1', 'BISS-E' or 'PASSPHRASE'.
key	no	The key value associated with the id and name in this record. For biss keys this is a string entirely consisting of the hexadeximal characters 0123456789ABCDEF, it is either 12 or 16 characters long. For other key types the key is free text.
modified	no	automatically created timestamp of last change

5.6.13 Table BeaconAttenuations (beacon_attenuations)

This table stores the attenuation values for a particular beacon / antenna combination

Field	null	Description
beacon_attn_id	no	A unique ID of this record (auto generated).
beacon_id	no	The ID of the beacon this attenuation belongs to.

Field	null	Description
antenna_id	no	The ID of the antenna this attenuation belongs to,
attenuation	no	The attenuation value (dB)
modified	no	automatically created timestamp of last change

5.6.14 Table TcAttenuations (tc_attenuations)

This table stores the attenuation values for a particular TC channel / antenna combination

Field	null	Description
tc_attn_id	no	A unique ID of this record (auto generated).
beacon_id	no	The ID of the beacon this attenuation belongs to.
antenna_id	no	The ID of the antenna this attenuation belongs to,
attenuation_1	no	The upconverter attenuation value (dB)
attenuation_2	no	The HPA attenuation value (dB)
modified	no	automatically created timestamp of last change

5.6.15 Table TmAttenuations (tm_attenuations)

This table stores the attenuation values for a particular TM channel / antenna combination

Field	null	Description
tm_attn_id	no	A unique ID of this record (auto generated).
beacon_id	no	The ID of the beacon this attenuation belongs to.
antenna_id	no	The ID of the antenna this attenuation belongs to,
attenuation_1	no	The primary TM signal attenuation value (dB)
attenuation_2	no	The secondary TM signal value (dB)
modified	no	automatically created timestamp of last change

5.6.16 Table InventoryItems (inventory_items)

The table inventory_items stores the properties of all items known to the inventory database.

Field	null	Description
item_id	no	A unique ID of this record (auto generated).
serial_no	yes	The serial number of the item / device
vendor	yes	The vendor name (for search filters)
model	yes	The model name (for search filters)
state	no	The administrative state of the device
comment	yes	free text
modified	no	automatically created timestamp of last change

5.6.17 Table InventoryDevices (inventory_devices)

The table inventory_devices stores the device mappings of all items known to the inventory database.

Field	null	Description
device_id	no	A unique ID of this record (auto generated).
item_no	yes	The ID of the item this device is mapped to
mnc_name	yes	The M&C name
device_name	yes	The device name
state	no	The administrative state of the device
online	yes	Boolean flag indicating if the device is online
modified	no	automatically created timestamp of last change

5.6.18 Table InventoryLog (inventory_log)

The table inventory_log stores the live cycle logs of all items which are registered in the inventory_items table.

Field	null	Description
entry_id	no	A unique ID of this record (auto generated).
item_id	no	Links to the inventory_items entry this log entry refers to.

Field	null	Description
mnc_name	yes	The M&C which issued this message or null if the message was entered at the UI
dev_name	yes	The satnms device name which issued this message
message	no	the message text
created	no	timestamp, generated by database at creation time

5.6.19 Table Documents (documents)

The table documents stores information about the documents (pdf, png, jpg) stored in the backend.

Field	null	Description
document_id	no	A unique ID of this record (auto generated).
file_name	no	Name of the file.
content_type	no	Content type of the file. must be one of application/pdf, image/png or image/jpg.
driver	yes	The satnms driver name this document refers to or null if it does not refer to a driver / device type.
mnc_name	yes	The M&C name this document refers to or null if it does not refer to a M&C.
comment	yes	free text
modified	no	automatically created timestamp of last change

The backend handles three types of documents:

- 'M&C related' documents refer to a particular M&C. They have the 'mnc_name' field set.
- 'Driver related' documents refer to a particular device type / driver. They have the 'driver' field set.
- 'global' documents do not refer to anything

Documents are stored on the backend machine in a common directory tree. Global documents are stored at the root of this tree. For each M&C and for each device driver the backend creates a subdirectory. M&C or driver related documents are stored in one of these subdirectories. File names must be unique within the subdirectory where they are are stored.

5.6.20 Table GlobalSettings (globalSettings)

The global_settings stores the properties of changed settings from default by admins.

Field	null	Description
settings_id	Integer	A Unique Id of an setting
key	no	Setting name as key
value	no	Value of the setting
username	yes	Username to which the setting belongs
isLocked	no	Whether the setting is locked for user to overwrite it
modified	no	timestamp, generated by database at creation time

5.6.21 Table User Settings (userSettings)

The user_settings stores the properties of changed settings from default by user.

Field	null	Description
settings_id	Integer	A Unique Id of an setting
key	no	Setting name as key
value	no	Value of the setting
username	yes	Username to which the setting belongs
isLocked	no	Whether the setting is locked for user to overwrite it
modified	no	timestamp, generated by database at creation time

5.7 Fonts

The satnms software uses a number of predefined fonts to be used in the user interface. The table below shows which font types / sizes match the pixel by pixel position and size values of existing screen definitions.

Name	Font	Size	Style
small	Arial	10px	PLAIN
plain	Arial	12px	PLAIN
bold	Arial	12px	BOLD

Name	Font	Size	Style
title	Arial	16px	BOLD
huge	Arial	28px	PLAIN
typewriter	Courier New	10px	PLAIN

6 Device Driver Descriptions

The following pages describe all device drivers coming with the sat-nms software. Each device driver description contains a short paragraph about what type of equipment this driver is for and how should be configured that it works seamlessly with the device.

This manual contains three tables which give you fast access to the device driver documentation:

- Device Drivers By Name --- contains a table which lists all device drivers sorted by their driver name.
- Device Drivers By Vendor --- contains a table which lists all device drivers sorted by the name of the vendor of the device to control
- Device Drivers By Tag --- contains a table which lists all device drivers sorted by tags assigned to the device drivers. These tags classify devices to groups like HPAs, antenna controllers, IRDs and more.

6.1 Device Drivers By Name

Driver	***Version***	***Vendor***	***Tags***
Antenna-Management	1.09 251126	SatService	logical
Antenna-Tracking	1.06 171221	SatService	logical
Execute	2.0 211025	SatService	logical
File-Recorder	2.03 190616	SatService	logical
Gate-Array	1.03 190506	SatService	logical
Level-Set	2.08 200310	SatService	logical
Pointing-Robot	2.05 180105	SatService	logical
SatService-ACU-ODM-Serial	1.06 230718	SatService	Antenna-Controller
SatService-ACU-ODM	3.05 230719	SatService	Antenna-Controller
SatService-ACU-Prediction	1.02 210621	SatService	logical

Driver	***Version***	***Vendor***	***Tags***
SatService-ACU2-Serial	2.06 230920	SatService	Antenna-Controller
SatService-ACU2	2.05 240423	SatService	Antenna-Controller
SatService-Beacon-Receiver	4.02 230911	SatService	Tracking-Receiver
Sysinfo	3.15 250728	SatService	Server

6.2 Device Drivers By Vendor

Vendor	***Driver***	***Version***	***Tags***
SatService	Execute	2.0 211025	logical
	File-Recorder	2.03 190616	logical
	Pointing-Robot	2.05 180105	logical
	SatService-ACU-ODM	3.05 230719	Antenna-Controller
	SatService-ACU-ODM-Serial	1.06 230718	Antenna-Controller
	SatService-ACU2	2.05 240423	Antenna-Controller
	SatService-ACU2-Serial	2.06 230920	Antenna-Controller
	SatService-ACU-Prediction	1.02 210621	logical
	SatService-Beacon-Receiver	4.02 230911	Tracking-Receiver
	Sysinfo	3.15 250728	Server
	Antenna-Management	1.09 251126	logical

Vendor	*Driver*	*Version*	*Tags*
	Antenna-Tracking	1.06 171221	logical
	Gate-Array	1.03 190506	logical
	Level-Set	2.08 200310	logical

6.3 Device Drivers By Tag

Tag	*Driver*	*Version*	*Vendor*
Antenna-Controller	SatService-ACU-ODM	3.05 230719	SatService
	SatService-ACU-ODM-Serial	1.06 230718	SatService
	SatService-ACU2	2.05 240423	SatService
	SatService-ACU2-Serial	2.06 230920	SatService
logical	Execute	2.0 211025	SatService
	File-Recorder	2.03 190616	SatService
	Pointing-Robot	2.05 180105	SatService
	SatService-ACU-Prediction	1.02 210621	SatService
	Antenna-Management	1.09 251126	SatService
	Antenna-Tracking	1.06 171221	SatService
	Gate-Array	1.03 190506	SatService
	Level-Set	2.08 200310	SatService

Tag	*Driver*	*Version*	*Vendor*
Server	Sysinfo	3.15 250728	SatService
Tracking-Receiver	SatService-Beacon-Receiver	4.02 230911	SatService

6.1 Execute

This logical device executes a script every time a parameter changes and submits this parameter to the script.

Typical application is to react on the summary fault and trigger a local script or program on the MNC server.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- ℹ --- The device info page.
- 🔧 --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

parameter	*description*
script	The name of the script which should be executed
inputId	Enter the message identifier here of the parameter the device shall listen

Security advice and Remarks

The Execute device runs programs and scripts of the Linux operating system which are located in /home/satnms/scripts/.

Using the Execute device may be dangerous, if the executed script does not behave properly and if it does not handle input values in a proper way.

1. Scripts may be scripts requiring bash, perl etc and Linux executables as well
2. Scripts and programs must be locates in the /home/satnms/scripts/ subdirectory, they must be readable / executable to the 'satnms' user.
3. The software reads the output of the program to stdout but ignores the stderr output. This is useful for some Linux programs which send their output to stdout and status reports to stderr. If the stderrr output is required, the program should be enclosed in a

bash script performing the appropriate redirection.

4. There are no timeouts with the Script protocol. If a script or program doesn't terminate, it blocks the device thread containing the driver forever without notice for the operator. When the MNC server terminates, the subprocess is forced to terminate as well.
5. There is no explicit limitation of the volume of data collected from the script's output. Hence, an endless running program which continuously output e.g. some logging data, will eat up the complete memory of the MNC server process an bring the system down.

Be careful when using the Execute device!

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.lastMessage	TEXT	R/O	StringRange R/O
config.script	TEXT	SAVE SETUP	StringRange
config.inputId	TEXT	SAVE SETUP	StringRange
input	TEXT	NOPRESET	StringRange
run	TEXT	NOPRESET	StringRange
arguments	TEXT	SAVE NOPRESET	StringRange

6.2 File-Recorder

The File-Recorder logical device provides a simple facility to write measurement or state values in regular intervals to a file. In user configurable intervals, the File-Recorder writes a line consisting of a time stamp and up to 4 measurement values to the file. A file size limit may be defined to prevent the File-Recorder from creating arbitrary sized files.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- 👁 --- This page contain only one button which launches a History File Viewer to display the recorded data graphically.
- ℹ --- The device info page.
- 🔧 --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the File-Recorder device with the devices which origin the measurement values to record.

- *parameter* --- *description*

- channel1Id --- The message identifier of the parameter to record as 'channel 1'. If left empty, an empty column is written to the file for this channel.

- channel2Id --- The message identifier of the parameter to record as 'channel 2'. If left empty, an empty column is written to the file for this channel.

- channel3Id --- The message identifier of the parameter to record as 'channel 3'. If left empty, an empty column is written to the file for this channel.

- channel4Id --- The message identifier of the parameter to record as 'channel 4'. If left empty, an empty column is written to the file for this channel.

- interval --- The time mean interval between two lines written to the file. The true interval may vary by about one second. Below some long term intervals, expressed in seconds.

  - 1 hour: --- 3600 seconds
  - 24 hours: --- 86400 seconds
  - 7 days: --- 604800 seconds
  - 30 days: --- 2592000 seconds

- liveInterval --- The time mean interval for updating the live view buffer.

- maxSize --- The maximum file size (KBytes) the recorder will produce. If set to a non-zero value, the File-Recorder backs up the output file to a file named 'XXXX. old.txt' as soon as the size limit is reached. The recent values are written to a new file 'XXXX.txt'. A value of zero disables the file size checking.**Using the File-Recorder device with a small 'interval' value may generate a huge amount of data. Limiting the file size is strongly recommended!**

- directory --- The location where to store the file recorder's data. By default, File-Recorder devices store their files in the base directory of the sat-nms installation. This is /home/satnms/ in most cases. You may define any subdirectoy of this base directory as the location where this File-Recorder shall store its data. The directory already must exist.When you change a File-Recorder's directory, any existing files in the previous directory are moved to the new one.

File Format

The File-Recorder writes the sample to a file 'XXXX.txt', where 'XXXX' is the device name of the File-Recorder device. The file is placed in the Server's base directory unless the 'directory' configuration setting is set to a subdirectory of this.

The file entirely consists of lines, separated by CR/LF pairs. Each line contains a set of measurement values, preceeded by a time stamp. Columns are separated by semicolons, no space characters are inserted between the columns.

Reading A File From Remote

The file written by the File-Recorder (also the backed up data file, if size limiting is enabled) may be read from a remote computer via http. The URL to read the file is:

http://xxx.xxx.xxx.xxx:2007/XXXX.txt or **http://xxx.xxx.xxx.xxx:2007/XXXX. old.txt** respectively.

Hereby 'xxx.xxx.xxx.xxx' is the IP Address or host name of the M&C / VLC server, 'XXXX' is the name of the File-Recorder device on this machine.

If the 'directory' configuration setting is set to e.g. 'somedir', the URLs to read the files are:

http://xxx.xxx.xxx.xxx:2007/somedir/XXXX.txt or
http://xxx.xxx.xxx.xxx:2007/somedir/XXXX.old.txt respectively.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.directory	TEXT	R/O	StringRange R/O
config.channel1Id	TEXT	SAVE SETUP	StringRange
config.channel2Id	TEXT	SAVE SETUP	StringRange
config.channel3Id	TEXT	SAVE SETUP	StringRange
config.channel4Id	TEXT	SAVE SETUP	StringRange
config.interval	INTEGER	SAVE SETUP	IntegerRange (1 .. 2592000)
config.liveInterval	INTEGER	SAVE SETUP	IntegerRange (1 .. 2592000)

name	*type*	*flags*	*range*
config.maxSize	INTEGER	SAVE SETUP	IntegerRange (0 .. 100000)
config.directory	TEXT	SAVE SETUP	StringRange
liveView	TEXT	R/O	StringRange R/O
directory	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Invalid directory)

6.3 Pointing-Robot

This page describes the device driver and the device window for the Pointing-Robot logical device. This device lets you make a motorized antenna follow a satellite based on it's (known) orbit position. For this the Pointing-Robot implements orbit position calculations based on common ephemeris data models. The basic functions of a Pointing-Robot device are:

- Point the antenna to the nominal position of a satellite.
- Move the antenna along a satellite position calculated from ephemeris data.
- Move the antenna along a path given by a table.

The Pointing-Robot is part of the sat-nms ACU Indoor Unit. It supplements the sat-nms ACU-ODM with the orbit calculation routines described below. The Pointing-Robot principally works together with all drivers for antenna controllers which are part of the sat-nms software. However some functions like automatically switching off the ACU's tracking facility are uniquely designed for the sat-nms ACU-ODM.

Move the antenna along a satellite position calculated from ephemeris data

You can make the pointing robot follow a satellite position computed from usual ephemeris data sets. The robot recognizes the following data formats:

- **Keplerian elements** --- The robot recognizes and parses keplerian elements in the NASA 2-line format. The two lines of element definition may be prepended by an additional line with the satellite's name. A common source for Keplerian elements in this format is www.celestrak.com . The pointing robot uses the NORAD SDP4 model to predict a satellite's position from this type of data.
- **Intelsat elements** --- Intelsat publish ephemeris data for their communication satellites at their web site. The data format and the mathematical model for this data is Intelsat proprietary. The pointing robot recognizes this data type as well. You should copy the complete text of the ephemeris data set into the entry field to make the software recognize the data correctly.

Pointing the antenna to the nominal position of a satellite

If a single floating point number is entered to the ephemeris data field, the pointing robot interprets this as a nominal orbit position of the satellite (°E).

Move the antenna along a path given by a table

The pointing robot is capable to move the antenna along a path defined in a text file table. You either may copy the file to the IDU computer by means of FTP and enter the name of file to the ephemeris field or you may enter or cot/paste the table directly to this field. If you do the latter, the size of the table is limited to 32000 characters.

The Pointing-Robot provides a file selection dropdown widget to select files located in the /home/satnms/robot directory on the M&C&VLC server. To use this feature, select the file from the dropdown list and then click "SELECT". This copies the file name into the ephemeris data field and thus selects the file.

Files in the /home/satnms/robot directory which are no longer used may be deleted by selecting the file do delete from the drop down list and clicking DELETE. A file actually in use cannot be deleted, the request is silently ignored.

Polarization calculation

When calculating the antenna pointing from a orbital model or when interpolating the pointing from the table data, the Pointing-Robot also calculates a polarization angle. As each satellite uses its individual polarization offset, the calculated angle is normally for information only.

With table tracking, the pointing robot may also command the calculated polarization angle at the antenna. Setting "apply pol" to YES enables this feature. "apply pol" is not available with orbital elements.

Tracking table format

The format of the table must be as follows: It contains the data as a three or four column table. Empty lines are ignored, comments starting with a '#' as well. The numbers in the table are parsed as floating point numbers which only may consist of decimal digits, one decimal point and an optional leading '-' if negative. The columns must be separated by an arbitrary number of space or tabulator characters. They have the following meanings:

- 1 --- time stampThe time stamp must be a Julian date with the time of day coded as a fraction of a day. Example: The Julian date for the common base of most computer clocks (1970-01-0100:00:00) is 2440587.5
- 2 --- azimuth angleThe azimuth angle in degrees
- 3 --- elevation angleThe elevation angle in degrees

File import

The Pointing Robot GUI offers an import function which permits to copy pointing tables from the client computer to the server and thereby convert the file format to that one internally used by the sat-nms software.

Parsing order

If the pointing robot gets entered a new ephemeris data set, it tries to interpret the data in the following way:

1. It checks if the first line of the data specifies an existing file (residing on the IDU computer, therefore the file path should use Unix-style ('/') directory separators if necessary. If there is the specified file, it gets read an interpreted in a way, its file extension suggests: '*.tle' or '*.TLE': NASA 2-line ephemeris data. '*.i11' or '*.I11': Intelsat ephemeris data. Files with other extensions are interpreted as table data as described above.
2. It checks if the data is a valid NASA 2-line data set. A satellite name before the 2-line data is accepted and parsed separately if present.
3. It checks if the data contains valid Intelsat ephemeris data.
4. It checks if the first number in the first line of the data is a valid Julian date in the range 2005 .. 2050. In this case the robot assumes, that the ephemeris data contain a pointing table.
5. Finally, if nothing of the above mentioned formats did match, the robot tries to interpret the first line of the data as the nominal orbit position of the satellite.

If all of the parsing attempts fail, the pointing robot reports the model type to be UNKNOWN. Mostly this happens due to artifacts of previously entered elements which had not been deleted before the new ephemeris data was entered / pasted.

Remote control

The Pointing-Robot may be updated easily with updated pointing tables or ephemeris data by copying such files to the /home/public directory of the M&C system using the SMB file sharing protocol and then commanding the name of this file to the 'elements.input' variable of the Pointing-Robot using one of the M&C system's remote control ports.

ACU fault monitoring

The device can monitor the ACU's summary fault and show it in the device screen. The parameter 'antenna fault detection' may be set to one of the following settings:

- **IGNORE-ACU-FAULT** : The pointing robot ignores ACU faults, does not show them.
- **SHOW-ACU-FAULT** : The pointing robot shows ACU faults, continues to command the antenna controller even if there is an ACU fault.
- **STOP-ON-ACU-FAULT** : The pointing robot stops when it detects a fault of the antenna controller. It must be restarted manually after the ACU fault has been cleared.

Fault flags

In table tracking mode the Pointing-Robot reports two fault conditions regarding the end of table data:

File expired is reported if the end of the table data is reached. Some hours before (configurable in the device setup) the **File expires soon** fault is reported. The latter fault condition appears as a WARNING by default.

Both fault conditions are reported if the Pointing-Robot is in table tracking mode, even if the tracking is stopped. To clear faults concerning an outdated but unused table, simply clear the "ephemeris data" field.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- ⌨ --- The main parameter page displays/controls the antenna pointing. Actual readings and set points are shown in separate fields, so you can see where the antenna is going to while it is moving.
- F! --- The faults page.
- ℹ --- The device info page.
- 🔧 --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
antennaName	The name of the antenna ODM device to control. With the sat-nms IDU, this parameter is preconfigured to 'ODM' and should not be changed.
longitude	The antenna's longitude (°E).
latitude	The antenna's latitude (°N).
altitude	The antenna's altitude over sea (m)
expireWarning	Defines with table tracking how many hours before the end of data a "Files expires soon" warning / fault shall be rise.

Remarks

The accuracy of the calculations made by Pointing-Robot depend on the following:

- The geodetic location of the antenna must be known very precisely.
- The antenna's azimuth axis alignment must be precisely perpendicular.
- The clock of the M&C computer must be set to UTC.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.antennaName	TEXT	SAVE SETUP	StringRange
config.longitude	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.latitude	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.altitude	FLOAT	SAVE SETUP	DoubleRange (0 .. 0)
tracking	CHOICE	SAVE	EnumRange (OFF ON)
interval	INTEGER	SAVE	IntegerRange (0 .. 3600)
calculatePol	CHOICE	SAVE	EnumRange (NO YES)
antFaultMode	CHOICE	SAVE	EnumRange (IGNORE-ACU-FAULT SHOW-ACU-FAULT STOP-ON-ACU-FAULT)
expireWarning	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
elements.input	TEXT		StringRange
eval.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
eval.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
eval.pl	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
eval.orbit	TEXT	R/O	StringRange R/O
azimuth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
elevation	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
polarization	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
state.activeFault	BOOLEAN	R/O	BooleanRange R/O (true false)
state.latchedFault	BOOLEAN	R/O	BooleanRange R/O (true false)
state.short	TEXT	R/O	StringRange R/O
state.long	TEXT	R/O	StringRange R/O
state.time	TEXT	R/O	StringRange R/O
offset.az	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
offset.el	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
offset.pl	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
hysteresis.az	FLOAT	SAVE	DoubleRange (0.000 .. 10.000)
hysteresis.el	FLOAT	SAVE	DoubleRange (0.000 .. 10.000)
hysteresis.pl	FLOAT	SAVE	DoubleRange (0.000 .. 10.000)
elements.type	TEXT	R/O	StringRange R/O
elements.age	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
elements.epoch	TEXT	R/O	StringRange R/O
elements.validFrom	TEXT	R/O	StringRange R/O
elements.validTo	TEXT	R/O	StringRange R/O
evalByElements	TEXT	NOPRESET	StringRange
moveEvaluated	TEXT	NOPRESET	StringRange
resetLatched	TEXT	NOPRESET	StringRange
files	CHOICE		EnumRange (NONE)
selectFile	TEXT	NOPRESET	StringRange
deleteFile	TEXT	NOPRESET	StringRange
forceUpdateFileList	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (File expired)
faults.02	ALARM	R/O	AlarmFlagRange R/O (File expires soon)

6.4 SatService-ACU-ODM

This page describes the device driver and the device window for the sat-nms ACU outdoor module. The driver treats the ACU ODM as a plain antenna positioner, communicating via Ethernet/HTTP to it. The configuration parameters of the ACU ODM are accessible through the ACU web based user interface, they are not replicated as configuration parameters in the M&C device driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- ⌨ --- The main parameter page displays/controls the antenna pointing. Actual readings and set points are shown in separate fields, so you can see where the antenna is going to while it is moving.
- F! --- The faults page.
- ℹ --- The device info page.
- 🔧 --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
address	This is the only parameter that must be set. Enter the antenna controller's IP address in 'dotted quad' notation here. Example: " **192.168.2.81** ".
timeSync	Enables or disables the date/time synchronization of the ACU-ODM with the MNC/ACU-IDU system time.

Time Synchronization

If enabled via the setup option described above, the M&C System or ACU Indoor Unit (running this drivers) will send date and time in regular intervals to the ACU Outdoor Module. The ACU-ODM will update its internal clock.

The ACU-ODM does not have a time zone configuration, so it simply uses the time send by the MNC/ACU-IDU. For example: If the system clock of the MNC/ACU-IDU runs on UTC, the ACU-ODM clock is also set to UTC

Do not enable time synchronization for the first time if you just doing step or adaptive tracking, because a large leap in time will produce a wrong tracking model. After first synchronization you can re-enable step or adaptive tracking again. It need approx. 40 seconds after enabling until the ACU-ODM updates its internal clock. Use the web browser interface to check the current date and time of the ACU-ODM.

This feature syncs the date and time between ACU-ODM and MNC/ACU-IDU. If you want to synchronize the date and time of the MNC/ACU-IDU server you can use an external NTP server. Please refer to the sat-nms M&C Installation Manual.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.caps.tracking	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.polar	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.memtrack	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.clswitch	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.rellevel	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.standby	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.200targets	BOOLEAN	R/O	BooleanRange R/O (true false)
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.poslog	TEXT	R/O	StringRange R/O
info.beaconType	CHOICE		EnumRange (SATNMS VOLTAGE)
internal.azLoLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.azHiLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.elLoLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
internal.elHiLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.plLoLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.plHiLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.ibit	HEX	R/O	HexRange R/O (0 .. 0)
internal.ibitaux	HEX	R/O	HexRange R/O (0 .. 0)
internal.obit	HEX	R/O	HexRange R/O (0 .. 0)
internal.tflt	HEX	R/O	HexRange R/O (0 .. 0)
internal.pcnt1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pcnt2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.caps	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.time	TEXT	R/O	StringRange R/O
internal.yy	TEXT	R/O	StringRange R/O
internal.mo	TEXT	R/O	StringRange R/O
internal.dd	TEXT	R/O	StringRange R/O
internal.hh	TEXT	R/O	StringRange R/O
internal.mi	TEXT	R/O	StringRange R/O
internal.ss	TEXT	R/O	StringRange R/O
internal.state	TEXT	R/O	StringRange R/O
internal.sleep	TEXT	R/O	StringRange R/O
internal.cmod	TEXT	R/O	StringRange R/O DIS
initialized	TEXT	R/O	StringRange R/O
config.timeSync	CHOICE	SAVE SETUP	EnumRange (ENABLED DISABLED)
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.00 .. 0.00)

name	***type***	***flags***	***range***
clswitch	CHOICE		EnumRange DIS (LINEAR CIRCULAR UNKNOWN)
step	TEXT	NOPRESET	StringRange
standby	TEXT	NOPRESET	StringRange DIS
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.cl	TEXT	R/O	StringRange R/O DIS
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
state.stopped	BOOLEAN	R/O	BooleanRange R/O (true false)
state.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.absLevel	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
state.mode1	TEXT	R/O	StringRange R/O
state.mode4	GENERIC	R/O	ObjectRange R/O generic object
state.azJitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.elJitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.azAmplitude	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.elAmplitude	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.trackHours	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.azModel	TEXT	R/O	StringRange R/O
state.elModel	TEXT	R/O	StringRange R/O
state.modelAge	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
trackMode	CHOICE		EnumRange (OFF STEP ADAPTIVE)
interval	INTEGER		IntegerRange DIS (1 .. 1638)
stepSize	INTEGER		IntegerRange DIS (1 .. 100)

name	***type***	***flags***	***range***
measDelay	INTEGER		IntegerRange DIS (1 .. 9999)
recoveryDelay	INTEGER		IntegerRange DIS (100 .. 30000)
averaging	INTEGER		IntegerRange DIS (1 .. 25)
frequency	FLOAT		DoubleRange DIS (950.000 .. 40000.000)
levelThreshold	FLOAT		DoubleRange DIS (-999.00 .. 100.00)
jitterThreshold	INTEGER		IntegerRange DIS (0 .. 100)
smoothing	INTEGER		IntegerRange DIS (0 .. 6)
azMaxModel	CHOICE		EnumRange DIS (SMALL MEDIUM LARGE)
elMaxModel	CHOICE		EnumRange DIS (SMALL MEDIUM LARGE)
levelOffset	FLOAT		DoubleRange DIS (0.00 .. 0.00)
levelZero	TEXT	NOPRESET	StringRange DIS
retryOnFault	CHOICE		EnumRange DIS (NEVER ONCE FOREVER)
reset	TEXT	NOPRESET	StringRange
motorStop	TEXT	NOPRESET	StringRange
motorReset	TEXT	NOPRESET	StringRange
clearTracking	TEXT	NOPRESET	StringRange
logPosition	TEXT	NOPRESET	StringRange
target.name	TEXT		StringRange
target.0	TEXT	R/O	StringRange R/O
target.1	TEXT	R/O	StringRange R/O
target.2	TEXT	R/O	StringRange R/O
target.3	TEXT	R/O	StringRange R/O
target.4	TEXT	R/O	StringRange R/O
target.5	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.6	TEXT	R/O	StringRange R/O
target.7	TEXT	R/O	StringRange R/O
target.8	TEXT	R/O	StringRange R/O
target.9	TEXT	R/O	StringRange R/O
target.10	TEXT	R/O	StringRange R/O
target.11	TEXT	R/O	StringRange R/O
target.12	TEXT	R/O	StringRange R/O
target.13	TEXT	R/O	StringRange R/O
target.14	TEXT	R/O	StringRange R/O
target.15	TEXT	R/O	StringRange R/O
target.16	TEXT	R/O	StringRange R/O
target.17	TEXT	R/O	StringRange R/O
target.18	TEXT	R/O	StringRange R/O
target.19	TEXT	R/O	StringRange R/O
target.20	TEXT	R/O	StringRange R/O
target.21	TEXT	R/O	StringRange R/O
target.22	TEXT	R/O	StringRange R/O
target.23	TEXT	R/O	StringRange R/O
target.24	TEXT	R/O	StringRange R/O
target.25	TEXT	R/O	StringRange R/O
target.26	TEXT	R/O	StringRange R/O
target.27	TEXT	R/O	StringRange R/O
target.28	TEXT	R/O	StringRange R/O
target.29	TEXT	R/O	StringRange R/O
target.30	TEXT	R/O	StringRange R/O
target.31	TEXT	R/O	StringRange R/O
target.32	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.33	TEXT	R/O	StringRange R/O
target.34	TEXT	R/O	StringRange R/O
target.35	TEXT	R/O	StringRange R/O
target.36	TEXT	R/O	StringRange R/O
target.37	TEXT	R/O	StringRange R/O
target.38	TEXT	R/O	StringRange R/O
target.39	TEXT	R/O	StringRange R/O
target.40	TEXT	R/O	StringRange R/O
target.41	TEXT	R/O	StringRange R/O
target.42	TEXT	R/O	StringRange R/O
target.43	TEXT	R/O	StringRange R/O
target.44	TEXT	R/O	StringRange R/O
target.45	TEXT	R/O	StringRange R/O
target.46	TEXT	R/O	StringRange R/O
target.47	TEXT	R/O	StringRange R/O
target.48	TEXT	R/O	StringRange R/O
target.49	TEXT	R/O	StringRange R/O
target.50	TEXT	R/O	StringRange R/O
target.51	TEXT	R/O	StringRange R/O
target.52	TEXT	R/O	StringRange R/O
target.53	TEXT	R/O	StringRange R/O
target.54	TEXT	R/O	StringRange R/O
target.55	TEXT	R/O	StringRange R/O
target.56	TEXT	R/O	StringRange R/O
target.57	TEXT	R/O	StringRange R/O
target.58	TEXT	R/O	StringRange R/O

name	type	flags	range
target.59	TEXT	R/O	StringRange R/O
target.60	TEXT	R/O	StringRange R/O
target.61	TEXT	R/O	StringRange R/O
target.62	TEXT	R/O	StringRange R/O
target.63	TEXT	R/O	StringRange R/O
target.64	TEXT	R/O	StringRange R/O
target.65	TEXT	R/O	StringRange R/O
target.66	TEXT	R/O	StringRange R/O
target.67	TEXT	R/O	StringRange R/O
target.68	TEXT	R/O	StringRange R/O
target.69	TEXT	R/O	StringRange R/O
target.70	TEXT	R/O	StringRange R/O
target.71	TEXT	R/O	StringRange R/O
target.72	TEXT	R/O	StringRange R/O
target.73	TEXT	R/O	StringRange R/O
target.74	TEXT	R/O	StringRange R/O
target.75	TEXT	R/O	StringRange R/O
target.76	TEXT	R/O	StringRange R/O
target.77	TEXT	R/O	StringRange R/O
target.78	TEXT	R/O	StringRange R/O
target.79	TEXT	R/O	StringRange R/O
target.80	TEXT	R/O	StringRange R/O
target.81	TEXT	R/O	StringRange R/O
target.82	TEXT	R/O	StringRange R/O
target.83	TEXT	R/O	StringRange R/O
target.84	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.85	TEXT	R/O	StringRange R/O
target.86	TEXT	R/O	StringRange R/O
target.87	TEXT	R/O	StringRange R/O
target.88	TEXT	R/O	StringRange R/O
target.89	TEXT	R/O	StringRange R/O
target.90	TEXT	R/O	StringRange R/O
target.91	TEXT	R/O	StringRange R/O
target.92	TEXT	R/O	StringRange R/O
target.93	TEXT	R/O	StringRange R/O
target.94	TEXT	R/O	StringRange R/O
target.95	TEXT	R/O	StringRange R/O
target.96	TEXT	R/O	StringRange R/O
target.97	TEXT	R/O	StringRange R/O
target.98	TEXT	R/O	StringRange R/O
target.99	TEXT	R/O	StringRange R/O
target.100	TEXT	R/O	StringRange R/O
target.101	TEXT	R/O	StringRange R/O
target.102	TEXT	R/O	StringRange R/O
target.103	TEXT	R/O	StringRange R/O
target.104	TEXT	R/O	StringRange R/O
target.105	TEXT	R/O	StringRange R/O
target.106	TEXT	R/O	StringRange R/O
target.107	TEXT	R/O	StringRange R/O
target.108	TEXT	R/O	StringRange R/O
target.109	TEXT	R/O	StringRange R/O
target.110	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.111	TEXT	R/O	StringRange R/O
target.112	TEXT	R/O	StringRange R/O
target.113	TEXT	R/O	StringRange R/O
target.114	TEXT	R/O	StringRange R/O
target.115	TEXT	R/O	StringRange R/O
target.116	TEXT	R/O	StringRange R/O
target.117	TEXT	R/O	StringRange R/O
target.118	TEXT	R/O	StringRange R/O
target.119	TEXT	R/O	StringRange R/O
target.120	TEXT	R/O	StringRange R/O
target.121	TEXT	R/O	StringRange R/O
target.122	TEXT	R/O	StringRange R/O
target.123	TEXT	R/O	StringRange R/O
target.124	TEXT	R/O	StringRange R/O
target.125	TEXT	R/O	StringRange R/O
target.126	TEXT	R/O	StringRange R/O
target.127	TEXT	R/O	StringRange R/O
target.128	TEXT	R/O	StringRange R/O
target.129	TEXT	R/O	StringRange R/O
target.130	TEXT	R/O	StringRange R/O
target.131	TEXT	R/O	StringRange R/O
target.132	TEXT	R/O	StringRange R/O
target.133	TEXT	R/O	StringRange R/O
target.134	TEXT	R/O	StringRange R/O
target.135	TEXT	R/O	StringRange R/O
target.136	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.137	TEXT	R/O	StringRange R/O
target.138	TEXT	R/O	StringRange R/O
target.139	TEXT	R/O	StringRange R/O
target.140	TEXT	R/O	StringRange R/O
target.141	TEXT	R/O	StringRange R/O
target.142	TEXT	R/O	StringRange R/O
target.143	TEXT	R/O	StringRange R/O
target.144	TEXT	R/O	StringRange R/O
target.145	TEXT	R/O	StringRange R/O
target.146	TEXT	R/O	StringRange R/O
target.147	TEXT	R/O	StringRange R/O
target.148	TEXT	R/O	StringRange R/O
target.149	TEXT	R/O	StringRange R/O
target.150	TEXT	R/O	StringRange R/O
target.151	TEXT	R/O	StringRange R/O
target.152	TEXT	R/O	StringRange R/O
target.153	TEXT	R/O	StringRange R/O
target.154	TEXT	R/O	StringRange R/O
target.155	TEXT	R/O	StringRange R/O
target.156	TEXT	R/O	StringRange R/O
target.157	TEXT	R/O	StringRange R/O
target.158	TEXT	R/O	StringRange R/O
target.159	TEXT	R/O	StringRange R/O
target.160	TEXT	R/O	StringRange R/O
target.161	TEXT	R/O	StringRange R/O
target.162	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.163	TEXT	R/O	StringRange R/O
target.164	TEXT	R/O	StringRange R/O
target.165	TEXT	R/O	StringRange R/O
target.166	TEXT	R/O	StringRange R/O
target.167	TEXT	R/O	StringRange R/O
target.168	TEXT	R/O	StringRange R/O
target.169	TEXT	R/O	StringRange R/O
target.170	TEXT	R/O	StringRange R/O
target.171	TEXT	R/O	StringRange R/O
target.172	TEXT	R/O	StringRange R/O
target.173	TEXT	R/O	StringRange R/O
target.174	TEXT	R/O	StringRange R/O
target.175	TEXT	R/O	StringRange R/O
target.176	TEXT	R/O	StringRange R/O
target.177	TEXT	R/O	StringRange R/O
target.178	TEXT	R/O	StringRange R/O
target.179	TEXT	R/O	StringRange R/O
target.180	TEXT	R/O	StringRange R/O
target.181	TEXT	R/O	StringRange R/O
target.182	TEXT	R/O	StringRange R/O
target.183	TEXT	R/O	StringRange R/O
target.184	TEXT	R/O	StringRange R/O
target.185	TEXT	R/O	StringRange R/O
target.186	TEXT	R/O	StringRange R/O
target.187	TEXT	R/O	StringRange R/O
target.188	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.189	TEXT	R/O	StringRange R/O
target.190	TEXT	R/O	StringRange R/O
target.191	TEXT	R/O	StringRange R/O
target.192	TEXT	R/O	StringRange R/O
target.193	TEXT	R/O	StringRange R/O
target.194	TEXT	R/O	StringRange R/O
target.195	TEXT	R/O	StringRange R/O
target.196	TEXT	R/O	StringRange R/O
target.197	TEXT	R/O	StringRange R/O
target.198	TEXT	R/O	StringRange R/O
target.199	TEXT	R/O	StringRange R/O
target.list	TEXT	R/O	StringRange R/O
target.goto	INTEGER	NOPRESET	IntegerRange (0 .. 99)
target.save	INTEGER	NOPRESET	IntegerRange (0 .. 99)
target.delete	INTEGER	NOPRESET	IntegerRange (0 .. 99)
goto	INTEGER	NOPRESET	IntegerRange (0 .. 99)
grmc	TEXT	R/O	StringRange R/O
aux.in.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.out.1	INTEGER		IntegerRange (0 .. 1)
aux.out.2	INTEGER		IntegerRange (0 .. 1)
aux.out.3	INTEGER		IntegerRange (0 .. 1)
aux.out.4	INTEGER		IntegerRange (0 .. 1)
aux.out.5	INTEGER		IntegerRange (0 .. 1)

name	***type***	***flags***	***range***
aux.out.6	INTEGER		IntegerRange (0 .. 1)
aux.out.7	INTEGER		IntegerRange (0 .. 1)
aux.out.8	INTEGER		IntegerRange (0 .. 1)
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ high limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ low limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EL high limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (EL low limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (PL high limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (PL low limit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (AZ motor fault)
faults.08	ALARM	R/O	AlarmFlagRange R/O (AZ motor timeout)
faults.09	ALARM	R/O	AlarmFlagRange R/O (EL motor fault)
faults.10	ALARM	R/O	AlarmFlagRange R/O (EL motor timeout)
faults.11	ALARM	R/O	AlarmFlagRange R/O (PL motor fault)
faults.12	ALARM	R/O	AlarmFlagRange R/O (PL motor timeout)
faults.13	ALARM	R/O	AlarmFlagRange R/O (AZ motor stopped)
faults.14	ALARM	R/O	AlarmFlagRange R/O (EL motor stopped)
faults.15	ALARM	R/O	AlarmFlagRange R/O (PL motor stopped)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Beacon level)
faults.17	ALARM	R/O	AlarmFlagRange R/O (AZ peaking)
faults.18	ALARM	R/O	AlarmFlagRange R/O (EL peaking)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Model fit)

name	*type*	*flags*	*range*
faults.20	ALARM	R/O	AlarmFlagRange R/O (Model reset)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Hub/Limit fault)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Emergency stop)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Cabinet open/Handheld)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Beacon receiver)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Soft Limits)

6.5 SatService-ACU-ODM-Serial

This page describes the device driver and the device window for the sat-nms ACU outdoor module. The driver treats the ACU ODM as a plain antenna positioner, communicating via Serial Interface to it. The configuration parameters of the ACU ODM are accessible through the ACU web based user interface, they are not replicated as configuration parameters in the M&C device driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- ⌨ --- The main parameter page displays/controls the antenna pointing. Actual readings and set points are shown in separate fields, so you can see where the antenna is going to while it is moving.
- F! --- The faults page.
- ℹ --- The device info page.
- 🔧 --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	*description*
address	n/a
timeSync	Enables or disables the date/time synchronization of the ACU-ODM with the MNC/ACU-IDU system time.

parameter	***description***
Serial	Remember to set the line parameters for the serial interface connected to the ACU-ODM. The RS232 interface of sat-nms ACU-ODM always operates at 9600 baud, no parity, 8 data bits, one stop bit. *9600,8,N,1* .

Serial protocol selection Depending on the device address set, the sat-nms ACU-ODM either runs framed protocol with start/stop characters and checksum or it provides a dumb terminal interface. If the address is set in the ACU configuration (web page) to 'A' .. 'G' use Miteq-MOD95 protocol, if the address is 'NONE' select Terminal-CR-CRLF protocol.

Time Synchronization

If enabled via the setup option described above, the M&C System or ACU Indoor Unit (running this drivers) will send date and time in regular intervals to the ACU Outdoor Module. The ACU-ODM will update its internal clock.

The ACU-ODM does not have a time zone configuration, so it simply uses the time send by the MNC/ACU-IDU. For example: If the system clock of the MNC/ACU-IDU runs on UTC, the ACU-ODM clock is also set to UTC

Do not enable time synchronization for the first time if you just doing step or adaptive tracking, because a large leap in time will produce a wrong tracking model. After first synchronization you can re-enable step or adaptive tracking again. It need approx. 40 seconds after enabling until the ACU-ODM updates its internal clock. Use the web browser interface to check the current date and time of the ACU-ODM.

This feature syncs the date and time between ACU-ODM and MNC/ACU-IDU. If you want to synchronize the date and time of the MNC/ACU-IDU server you can use an external NTP server. Please refer to the sat-nms M&C Installation Manual.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
info.frame	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.caps.tracking	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.polar	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.memtrack	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.clswitch	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.rellevel	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.standby	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.200targets	BOOLEAN	R/O	BooleanRange R/O (true false)
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.poslog	TEXT	R/O	StringRange R/O
info.beaconType	CHOICE		EnumRange (SATNMS VOLTAGE)
internal.ibit	HEX	R/O	HexRange R/O (0 .. 0)
internal.ibitaux	HEX	R/O	HexRange R/O (0 .. 0)
internal.obit	HEX	R/O	HexRange R/O (0 .. 0)
internal.tflt	HEX	R/O	HexRange R/O (0 .. 0)
internal.pcnt1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pcnt2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.caps	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.time	TEXT	R/O	StringRange R/O
internal.yy	TEXT	R/O	StringRange R/O
internal.mo	TEXT	R/O	StringRange R/O
internal.dd	TEXT	R/O	StringRange R/O
internal.hh	TEXT	R/O	StringRange R/O
internal.mi	TEXT	R/O	StringRange R/O
internal.ss	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
internal.state	TEXT	R/O	StringRange R/O
internal.sleep	TEXT	R/O	StringRange R/O
internal.cmod	TEXT	R/O	StringRange R/O DIS
initialized	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.timeSync	CHOICE	SAVE SETUP	EnumRange (ENABLED DISABLED)
config.azLoLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.azHiLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.elLoLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.elHiLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.plLoLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.plHiLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.00 .. 0.00)
clswitch	CHOICE		EnumRange DIS (LINEAR CIRCULAR UNKNOWN)
step	TEXT	NOPRESET	StringRange
standby	TEXT	NOPRESET	StringRange DIS
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.cl	TEXT	R/O	StringRange R/O DIS
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)

name	***type***	***flags***	***range***
state.stopped	BOOLEAN	R/O	BooleanRange R/O (true false)
state.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.absLevel	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
state.mode1	TEXT	R/O	StringRange R/O
state.azJitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.elJitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.azAmplitude	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.elAmplitude	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.trackHours	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.azModel	TEXT	R/O	StringRange R/O
state.elModel	TEXT	R/O	StringRange R/O
state.modelAge	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
trackMode	CHOICE		EnumRange (OFF STEP ADAPTIVE)
interval	INTEGER		IntegerRange DIS (1 .. 1638)
stepSize	INTEGER		IntegerRange DIS (1 .. 100)
measDelay	INTEGER		IntegerRange DIS (1 .. 9999)
recoveryDelay	INTEGER		IntegerRange DIS (100 .. 30000)
averaging	INTEGER		IntegerRange DIS (1 .. 25)
frequency	FLOAT		DoubleRange DIS (950.000 .. 40000.000)
levelThreshold	FLOAT		DoubleRange DIS (-999.00 .. 100.00)
jitterThreshold	INTEGER		IntegerRange DIS (0 .. 100)
smoothing	INTEGER		IntegerRange DIS (0 .. 6)
azMaxModel	CHOICE		EnumRange DIS (SMALL MEDIUM LARGE)
elMaxModel	CHOICE		EnumRange DIS (SMALL MEDIUM LARGE)

name	***type***	***flags***	***range***
levelOffset	FLOAT		DoubleRange DIS (0.00 .. 0.00)
levelZero	TEXT	NOPRESET	StringRange DIS
retryOnFault	CHOICE		EnumRange DIS (NEVER ONCE FOREVER)
reset	TEXT	NOPRESET	StringRange
motorStop	TEXT	NOPRESET	StringRange
motorReset	TEXT	NOPRESET	StringRange
clearTracking	TEXT	NOPRESET	StringRange
logPosition	TEXT	NOPRESET	StringRange
target.name	TEXT		StringRange
target.0	TEXT	R/O	StringRange R/O
target.1	TEXT	R/O	StringRange R/O
target.2	TEXT	R/O	StringRange R/O
target.3	TEXT	R/O	StringRange R/O
target.4	TEXT	R/O	StringRange R/O
target.5	TEXT	R/O	StringRange R/O
target.6	TEXT	R/O	StringRange R/O
target.7	TEXT	R/O	StringRange R/O
target.8	TEXT	R/O	StringRange R/O
target.9	TEXT	R/O	StringRange R/O
target.10	TEXT	R/O	StringRange R/O
target.11	TEXT	R/O	StringRange R/O
target.12	TEXT	R/O	StringRange R/O
target.13	TEXT	R/O	StringRange R/O
target.14	TEXT	R/O	StringRange R/O
target.15	TEXT	R/O	StringRange R/O
target.16	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.17	TEXT	R/O	StringRange R/O
target.18	TEXT	R/O	StringRange R/O
target.19	TEXT	R/O	StringRange R/O
target.20	TEXT	R/O	StringRange R/O
target.21	TEXT	R/O	StringRange R/O
target.22	TEXT	R/O	StringRange R/O
target.23	TEXT	R/O	StringRange R/O
target.24	TEXT	R/O	StringRange R/O
target.25	TEXT	R/O	StringRange R/O
target.26	TEXT	R/O	StringRange R/O
target.27	TEXT	R/O	StringRange R/O
target.28	TEXT	R/O	StringRange R/O
target.29	TEXT	R/O	StringRange R/O
target.30	TEXT	R/O	StringRange R/O
target.31	TEXT	R/O	StringRange R/O
target.32	TEXT	R/O	StringRange R/O
target.33	TEXT	R/O	StringRange R/O
target.34	TEXT	R/O	StringRange R/O
target.35	TEXT	R/O	StringRange R/O
target.36	TEXT	R/O	StringRange R/O
target.37	TEXT	R/O	StringRange R/O
target.38	TEXT	R/O	StringRange R/O
target.39	TEXT	R/O	StringRange R/O
target.40	TEXT	R/O	StringRange R/O
target.41	TEXT	R/O	StringRange R/O
target.42	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.43	TEXT	R/O	StringRange R/O
target.44	TEXT	R/O	StringRange R/O
target.45	TEXT	R/O	StringRange R/O
target.46	TEXT	R/O	StringRange R/O
target.47	TEXT	R/O	StringRange R/O
target.48	TEXT	R/O	StringRange R/O
target.49	TEXT	R/O	StringRange R/O
target.50	TEXT	R/O	StringRange R/O
target.51	TEXT	R/O	StringRange R/O
target.52	TEXT	R/O	StringRange R/O
target.53	TEXT	R/O	StringRange R/O
target.54	TEXT	R/O	StringRange R/O
target.55	TEXT	R/O	StringRange R/O
target.56	TEXT	R/O	StringRange R/O
target.57	TEXT	R/O	StringRange R/O
target.58	TEXT	R/O	StringRange R/O
target.59	TEXT	R/O	StringRange R/O
target.60	TEXT	R/O	StringRange R/O
target.61	TEXT	R/O	StringRange R/O
target.62	TEXT	R/O	StringRange R/O
target.63	TEXT	R/O	StringRange R/O
target.64	TEXT	R/O	StringRange R/O
target.65	TEXT	R/O	StringRange R/O
target.66	TEXT	R/O	StringRange R/O
target.67	TEXT	R/O	StringRange R/O
target.68	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.69	TEXT	R/O	StringRange R/O
target.70	TEXT	R/O	StringRange R/O
target.71	TEXT	R/O	StringRange R/O
target.72	TEXT	R/O	StringRange R/O
target.73	TEXT	R/O	StringRange R/O
target.74	TEXT	R/O	StringRange R/O
target.75	TEXT	R/O	StringRange R/O
target.76	TEXT	R/O	StringRange R/O
target.77	TEXT	R/O	StringRange R/O
target.78	TEXT	R/O	StringRange R/O
target.79	TEXT	R/O	StringRange R/O
target.80	TEXT	R/O	StringRange R/O
target.81	TEXT	R/O	StringRange R/O
target.82	TEXT	R/O	StringRange R/O
target.83	TEXT	R/O	StringRange R/O
target.84	TEXT	R/O	StringRange R/O
target.85	TEXT	R/O	StringRange R/O
target.86	TEXT	R/O	StringRange R/O
target.87	TEXT	R/O	StringRange R/O
target.88	TEXT	R/O	StringRange R/O
target.89	TEXT	R/O	StringRange R/O
target.90	TEXT	R/O	StringRange R/O
target.91	TEXT	R/O	StringRange R/O
target.92	TEXT	R/O	StringRange R/O
target.93	TEXT	R/O	StringRange R/O
target.94	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.95	TEXT	R/O	StringRange R/O
target.96	TEXT	R/O	StringRange R/O
target.97	TEXT	R/O	StringRange R/O
target.98	TEXT	R/O	StringRange R/O
target.99	TEXT	R/O	StringRange R/O
target.100	TEXT	R/O	StringRange R/O
target.101	TEXT	R/O	StringRange R/O
target.102	TEXT	R/O	StringRange R/O
target.103	TEXT	R/O	StringRange R/O
target.104	TEXT	R/O	StringRange R/O
target.105	TEXT	R/O	StringRange R/O
target.106	TEXT	R/O	StringRange R/O
target.107	TEXT	R/O	StringRange R/O
target.108	TEXT	R/O	StringRange R/O
target.109	TEXT	R/O	StringRange R/O
target.110	TEXT	R/O	StringRange R/O
target.111	TEXT	R/O	StringRange R/O
target.112	TEXT	R/O	StringRange R/O
target.113	TEXT	R/O	StringRange R/O
target.114	TEXT	R/O	StringRange R/O
target.115	TEXT	R/O	StringRange R/O
target.116	TEXT	R/O	StringRange R/O
target.117	TEXT	R/O	StringRange R/O
target.118	TEXT	R/O	StringRange R/O
target.119	TEXT	R/O	StringRange R/O
target.120	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.121	TEXT	R/O	StringRange R/O
target.122	TEXT	R/O	StringRange R/O
target.123	TEXT	R/O	StringRange R/O
target.124	TEXT	R/O	StringRange R/O
target.125	TEXT	R/O	StringRange R/O
target.126	TEXT	R/O	StringRange R/O
target.127	TEXT	R/O	StringRange R/O
target.128	TEXT	R/O	StringRange R/O
target.129	TEXT	R/O	StringRange R/O
target.130	TEXT	R/O	StringRange R/O
target.131	TEXT	R/O	StringRange R/O
target.132	TEXT	R/O	StringRange R/O
target.133	TEXT	R/O	StringRange R/O
target.134	TEXT	R/O	StringRange R/O
target.135	TEXT	R/O	StringRange R/O
target.136	TEXT	R/O	StringRange R/O
target.137	TEXT	R/O	StringRange R/O
target.138	TEXT	R/O	StringRange R/O
target.139	TEXT	R/O	StringRange R/O
target.140	TEXT	R/O	StringRange R/O
target.141	TEXT	R/O	StringRange R/O
target.142	TEXT	R/O	StringRange R/O
target.143	TEXT	R/O	StringRange R/O
target.144	TEXT	R/O	StringRange R/O
target.145	TEXT	R/O	StringRange R/O
target.146	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.147	TEXT	R/O	StringRange R/O
target.148	TEXT	R/O	StringRange R/O
target.149	TEXT	R/O	StringRange R/O
target.150	TEXT	R/O	StringRange R/O
target.151	TEXT	R/O	StringRange R/O
target.152	TEXT	R/O	StringRange R/O
target.153	TEXT	R/O	StringRange R/O
target.154	TEXT	R/O	StringRange R/O
target.155	TEXT	R/O	StringRange R/O
target.156	TEXT	R/O	StringRange R/O
target.157	TEXT	R/O	StringRange R/O
target.158	TEXT	R/O	StringRange R/O
target.159	TEXT	R/O	StringRange R/O
target.160	TEXT	R/O	StringRange R/O
target.161	TEXT	R/O	StringRange R/O
target.162	TEXT	R/O	StringRange R/O
target.163	TEXT	R/O	StringRange R/O
target.164	TEXT	R/O	StringRange R/O
target.165	TEXT	R/O	StringRange R/O
target.166	TEXT	R/O	StringRange R/O
target.167	TEXT	R/O	StringRange R/O
target.168	TEXT	R/O	StringRange R/O
target.169	TEXT	R/O	StringRange R/O
target.170	TEXT	R/O	StringRange R/O
target.171	TEXT	R/O	StringRange R/O
target.172	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.173	TEXT	R/O	StringRange R/O
target.174	TEXT	R/O	StringRange R/O
target.175	TEXT	R/O	StringRange R/O
target.176	TEXT	R/O	StringRange R/O
target.177	TEXT	R/O	StringRange R/O
target.178	TEXT	R/O	StringRange R/O
target.179	TEXT	R/O	StringRange R/O
target.180	TEXT	R/O	StringRange R/O
target.181	TEXT	R/O	StringRange R/O
target.182	TEXT	R/O	StringRange R/O
target.183	TEXT	R/O	StringRange R/O
target.184	TEXT	R/O	StringRange R/O
target.185	TEXT	R/O	StringRange R/O
target.186	TEXT	R/O	StringRange R/O
target.187	TEXT	R/O	StringRange R/O
target.188	TEXT	R/O	StringRange R/O
target.189	TEXT	R/O	StringRange R/O
target.190	TEXT	R/O	StringRange R/O
target.191	TEXT	R/O	StringRange R/O
target.192	TEXT	R/O	StringRange R/O
target.193	TEXT	R/O	StringRange R/O
target.194	TEXT	R/O	StringRange R/O
target.195	TEXT	R/O	StringRange R/O
target.196	TEXT	R/O	StringRange R/O
target.197	TEXT	R/O	StringRange R/O
target.198	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.199	TEXT	R/O	StringRange R/O
target.list	TEXT	R/O	StringRange R/O
target.goto	INTEGER	NOPRESET	IntegerRange (0 .. 99)
target.save	INTEGER	NOPRESET	IntegerRange (0 .. 99)
target.delete	INTEGER	NOPRESET	IntegerRange (0 .. 99)
target.loadList	TEXT	NOPRESET	StringRange
goto	INTEGER	NOPRESET	IntegerRange (0 .. 99)
grmc	TEXT	R/O	StringRange R/O
aux.in.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.out.1	INTEGER		IntegerRange (0 .. 1)
aux.out.2	INTEGER		IntegerRange (0 .. 1)
aux.out.3	INTEGER		IntegerRange (0 .. 1)
aux.out.4	INTEGER		IntegerRange (0 .. 1)
aux.out.5	INTEGER		IntegerRange (0 .. 1)
aux.out.6	INTEGER		IntegerRange (0 .. 1)
aux.out.7	INTEGER		IntegerRange (0 .. 1)
aux.out.8	INTEGER		IntegerRange (0 .. 1)
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ high limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ low limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EL high limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (EL low limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (PL high limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (PL low limit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (AZ motor fault)

name	***type***	***flags***	***range***
faults.08	ALARM	R/O	AlarmFlagRange R/O (AZ motor timeout)
faults.09	ALARM	R/O	AlarmFlagRange R/O (EL motor fault)
faults.10	ALARM	R/O	AlarmFlagRange R/O (EL motor timeout)
faults.11	ALARM	R/O	AlarmFlagRange R/O (PL motor fault)
faults.12	ALARM	R/O	AlarmFlagRange R/O (PL motor timeout)
faults.13	ALARM	R/O	AlarmFlagRange R/O (AZ motor stopped)
faults.14	ALARM	R/O	AlarmFlagRange R/O (EL motor stopped)
faults.15	ALARM	R/O	AlarmFlagRange R/O (PL motor stopped)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Beacon level)
faults.17	ALARM	R/O	AlarmFlagRange R/O (AZ peaking)
faults.18	ALARM	R/O	AlarmFlagRange R/O (EL peaking)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Model fit)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Model reset)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Hub/Limit fault)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Emergency stop)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Cabinet open/Handheld)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Beacon receiver)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Soft Limits)

6.6 SatService-ACU2

This page describes the device driver and the device window for the sat-nms ACU2 antenna controller. This driver uses Ethernet/HTTP for communication with it. The configuration parameters of the ACU2 are accessible through the ACU web based user interface, they are not replicated as configuration parameters in the MNC device driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page displays/controls the antenna pointing. Actual readings and set points are shown in separate fields, so you can see where the antenna is going to while it is moving.
- --- Show the tracking state.
- --- Target list to load, save and delete targets on the ACU
- --- Set the tracking parameters
- --- Editor for TLE's parameters
- --- Editor for I11 (Intelsat) Parameters.
- --- Target Editor
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
address	This is the only parameter that must be set. Enter the antenna controller's IP address in 'dotted quad' notation here. Example: " **192.168.2.81** ".
timeSync	Enables or disables the date/time synchronization of the ACU with the MNC/ACU-IDU system time.

Time Synchronization

If enabled via the setup option described above, the M&C System or ACU Indoor Unit (running this drivers) will send date and time in regular intervals to the ACU . The ACU will update its internal clock.

The ACU does not have a time zone configuration, so it simply uses the time send by the MNC/ACU-IDU. For example: If the system clock of the MNC/ACU-IDU runs on UTC, the ACU clock is also set to UTC

Do not enable time synchronization for the first time if you just doing step or adaptive tracking, because a large leap in time will produce a wrong tracking model. After first synchronization you can re-enable step or adaptive tracking again. It need approx. 40 seconds after enabling until the ACU updates its internal clock. Use the web browser interface to check the current date and time of the ACU.

This feature syncs the date and time between ACU and MNC/ACU-IDU. If you want to synchronize the date and time of the MNC/ACU-IDU server you can use an external NTP server. Please refer to the sat-nms M&C Installation Manual.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.caps.tracking	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.polar	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.memtrack	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.clswitch	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.rellevel	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.standby	BOOLEAN	R/O	BooleanRange R/O (true false)

name	***type***	***flags***	***range***
info.caps.200targets	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.i11tle	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.4axes	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.mhyst	BOOLEAN	R/O	BooleanRange R/O (true false)
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.poslog	TEXT	R/O	StringRange R/O
info.beaconType	CHOICE		EnumRange (SATNMS VOLTAGE)
internal.azLoLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.azHiLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.elLoLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.elHiLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.plLoLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.plHiLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.pl2LoLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.pl2HiLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.ibit	HEX	R/O	HexRange R/O (0 .. 0)
internal.ibi2	HEX	R/O	HexRange R/O (0 .. 0)
internal.ibitaux	HEX	R/O	HexRange R/O (0 .. 0)
internal.obit	HEX	R/O	HexRange R/O (0 .. 0)

name	***type***	***flags***	***range***
internal.obi2	HEX	R/O	HexRange R/O (0 .. 0)
internal.tflt	HEX	R/O	HexRange R/O (0 .. 0)
internal.pcnt1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pcnt2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.caps	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.time	TEXT	R/O	StringRange R/O
internal.yy	TEXT	R/O	StringRange R/O
internal.mo	TEXT	R/O	StringRange R/O
internal.dd	TEXT	R/O	StringRange R/O
internal.hh	TEXT	R/O	StringRange R/O
internal.mi	TEXT	R/O	StringRange R/O
internal.ss	TEXT	R/O	StringRange R/O
internal.state	TEXT	R/O	StringRange R/O
internal.sleep	TEXT	R/O	StringRange R/O
internal.cmod	TEXT	R/O	StringRange R/O DIS
initialized	TEXT	R/O	StringRange R/O
pl2StepEnable	BOOLEAN	R/O	BooleanRange R/O (true false)
config.timeSync	CHOICE	SAVE SETUP	EnumRange (ENABLED DISABLED)
config.bcrx.polAlias	TEXT	SETUP	StringRange
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.00 .. 0.00)
polarization2	FLOAT		DoubleRange DIS (0.00 .. 0.00)
clswitch	CHOICE		EnumRange DIS (LINEAR CIRCULAR UNKNOWN)

name	***type***	***flags***	***range***
step	TEXT	NOPRESET	StringRange
standby	TEXT	NOPRESET	StringRange DIS
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pl2	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
state.cl	TEXT	R/O	StringRange R/O DIS
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
state.stopped	BOOLEAN	R/O	BooleanRange R/O (true false)
state.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.absLevel	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
state.mode1	TEXT	R/O	StringRange R/O
state.mode4	GENERIC	R/O	ObjectRange R/O generic object
state.azJitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.elJitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.azAmplitude	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.elAmplitude	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.trackHours	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.azModel	TEXT	R/O	StringRange R/O
state.elModel	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
state.modelAge	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.targetTainted	TEXT	R/O	StringRange R/O
state.taintedText	TEXT	R/O	StringRange R/O
state.selectedTarget	INTEGER	R/O	IntegerRange R/O (0 .. 0)
trackMode	CHOICE		EnumRange (OFF STEP ADAPTIVE)
initMode	CHOICE		EnumRange DIS (STORED-POSITION ORBIT I11 TLE MODEL)
interval	INTEGER		IntegerRange DIS (1 .. 1638)
stepSize	INTEGER		IntegerRange DIS (1 .. 100)
measDelay	INTEGER		IntegerRange DIS (1 .. 9999)
recoveryDelay	INTEGER		IntegerRange DIS (100 .. 30000)
averaging	INTEGER		IntegerRange DIS (1 .. 25)
frequency	FLOAT		DoubleRange DIS (950.000 .. 40000.000)
levelThreshold	FLOAT		DoubleRange DIS (-999.00 .. 100.00)
jitterThreshold	INTEGER		IntegerRange DIS (0 .. 100)
smoothing	INTEGER		IntegerRange DIS (0 .. 6)
azMaxModel	CHOICE		EnumRange DIS (SMALL MEDIUM LARGE)
elMaxModel	CHOICE		EnumRange DIS (SMALL MEDIUM LARGE)
levelOffset	FLOAT		DoubleRange DIS (0.00 .. 0.00)
levelZero	TEXT	NOPRESET	StringRange DIS
spindleSaveMode	CHOICE		EnumRange DIS (OFF 1 2 3 4 5)

name	***type***	***flags***	***range***
spindleSaveThresh	INTEGER		IntegerRange DIS (0 .. 100)
modelHysteresis	INTEGER		IntegerRange DIS (0 .. 100)
modelBeforeTrack	CHOICE		EnumRange DIS (OFF ON)
azOffset	FLOAT		DoubleRange DIS (-180.000 .. 180.000)
elOffset	FLOAT		DoubleRange DIS (-180.000 .. 180.000)
orbit	FLOAT		DoubleRange DIS (0.000 .. 0.000)
selectedTLE	CHOICE		EnumRange DIS (00 - NONE)
selectedI11	CHOICE		EnumRange DIS (00 - NONE)
tleNo	INTEGER		IntegerRange DIS (0 .. 99)
i11No	INTEGER		IntegerRange DIS (0 .. 99)
maxAge	FLOAT		DoubleRange DIS (0.0 .. 999.0)
retryOnFault	CHOICE		EnumRange DIS (NEVER ONCE FOREVER)
polPredict	CHOICE		EnumRange DIS (OFF ON)
inclination	FLOAT		DoubleRange DIS (-45.000 .. 45.000)
satPolOffset	FLOAT		DoubleRange DIS (-45.000 .. 45.000)
reset	TEXT	NOPRESET	StringRange
motorStop	TEXT	NOPRESET	StringRange
motorReset	TEXT	NOPRESET	StringRange
clearTracking	TEXT	NOPRESET	StringRange
logPosition	TEXT	NOPRESET	StringRange
clearTainted	TEXT	NOPRESET	StringRange
bcrx.attenuation	CHOICE		EnumRange DIS (30 dB 20 dB 10 dB 0 dB)

name	***type***	***flags***	***range***
bcrx.averaging	CHOICE		EnumRange DIS (0.1 Hz 0.2 Hz 0.5 Hz 1 Hz 2 Hz 5 Hz)
bcrx.frequency	FLOAT		DoubleRange DIS (950.000 .. 40000.000)
bcrx.bandwidth	CHOICE		EnumRange DIS (6 kHz 12 kHz 30 kHz 100 kHz)
bcrx.polarization	CHOICE		EnumRange DIS (H V)
bcrx.threshold	FLOAT		DoubleRange DIS (-999.99 .. 0.00)
bcrx.speccomp	CHOICE		EnumRange DIS (OFF)
bcrx.ftrack.on	CHOICE		EnumRange DIS (OFF ON)
bcrx.ftrack.width	INTEGER		IntegerRange DIS (10 .. 1000)
bcrx.ftrack.interval	INTEGER		IntegerRange DIS (1 .. 21600)
bcrx.ssrch.delay	INTEGER		IntegerRange DIS (0 .. 600)
bcrx.ssrch.enable	CHOICE		EnumRange DIS (OFF ON)
bcrx.cn.on	CHOICE		EnumRange DIS (OFF C/N C/NO)
bcrx.cn.frequency	FLOAT		DoubleRange DIS (950.000 .. 40000.000)
bcrx.cn.interval	INTEGER		IntegerRange DIS (0 .. 0)
bcrx.offset	FLOAT		DoubleRange DIS (0.00 .. 0.00)
bcrx.output	FLOAT		DoubleRange DIS (0.000 .. 0.000)
edit.tle.load	CHOICE		EnumRange DIS (00 - NONE)
edit.tle.no	INTEGER		IntegerRange DIS (1 .. 99)
edit.tle.save	TEXT	NOPRESET	StringRange DIS
edit.tle.destination	INTEGER		IntegerRange DIS (1 .. 99)
edit.tle.name	TEXT		StringRange DIS

name	***type***	***flags***	***range***
edit.tle.line1	TEXT		StringRange DIS
edit.tle.line2	TEXT		StringRange DIS
edit.i11.load	CHOICE		EnumRange DIS (00 - NONE)
edit.i11.no	INTEGER		IntegerRange DIS (1 .. 99)
edit.i11.save	TEXT	NOPRESET	StringRange DIS
edit.i11.destination	INTEGER		IntegerRange DIS (1 .. 99)
edit.i11.name	TEXT		StringRange DIS
edit.i11.epoch	TEXT		StringRange DIS
edit.i11.lm0	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.i11.lm1	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.i11.lm2	FLOAT		DoubleRange DIS (0.000000 .. 0.000000)
edit.i11.lonc	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.i11.lonc1	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.i11.lons	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.i11.lons1	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.i11.latc	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.i11.latc1	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.i11.lats	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.i11.lats1	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)

name	***type***	***flags***	***range***
edit.i11.predlon	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.i11.predlat	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.target.load	CHOICE		EnumRange DIS (000 - ADAPTIVE TRACKING MEMORY)
edit.target.no	INTEGER		IntegerRange DIS (1 .. 199)
edit.target.save	TEXT	NOPRESET	StringRange DIS
edit.target.destination	INTEGER		IntegerRange DIS (1 .. 199)
edit.target.selectedTLE	CHOICE		EnumRange DIS (00 - NONE)
edit.target.selectedI11	CHOICE		EnumRange DIS (00 - NONE)
edit.target.tleNo	INTEGER		IntegerRange DIS (0 .. 99)
edit.target.i11No	INTEGER		IntegerRange DIS (0 .. 99)
edit.target.trackMode	CHOICE		EnumRange DIS (OFF STEP ADAPTIVE MEMORY STEP-TLE STEP-I11 TLE I11)
edit.target.initMode	CHOICE		EnumRange DIS (STORED-POSITION ORBIT I11 TLE)
edit.target.interval	INTEGER		IntegerRange DIS (60 .. 1638)
edit.target.stepSize	INTEGER		IntegerRange DIS (1 .. 100)
edit.target.measDelay	INTEGER		IntegerRange DIS (1 .. 9999)
edit.target.recoveryDelay	INTEGER		IntegerRange DIS (100 .. 30000)
edit.target.averaging	INTEGER		IntegerRange DIS (1 .. 25)
edit.target.levelThreshold	FLOAT		DoubleRange DIS (-999.00 .. 100.00)
edit.target.jitterThreshold	INTEGER		IntegerRange DIS (0 .. 100)
edit.target.smoothing	INTEGER		IntegerRange DIS (0 .. 6)

name	***type***	***flags***	***range***
edit.target.azMaxModel	CHOICE		EnumRange DIS (SMALL MEDIUM LARGE)
edit.target.elMaxModel	CHOICE		EnumRange DIS (SMALL MEDIUM LARGE)
edit.target.levelOffset	FLOAT		DoubleRange DIS (0.00 .. 0.00)
edit.target.spindleSaveMode	CHOICE		EnumRange DIS (OFF 1 2 3 4 5 6 7 8 9 10 11 12)
edit.target.spindleSaveThresh	INTEGER		IntegerRange DIS (0 .. 100)
edit.target.modelHysteresis	INTEGER		IntegerRange DIS (0 .. 100)
edit.target.modelBeforeTrack	CHOICE		EnumRange DIS (OFF ON)
edit.target.azOffset	FLOAT		DoubleRange DIS (-180.000 .. 180.000)
edit.target.elOffset	FLOAT		DoubleRange DIS (-180.000 .. 180.000)
edit.target.orbit	FLOAT		DoubleRange DIS (0.000 .. 0.000)
edit.target.maxAge	FLOAT		DoubleRange DIS (0.0 .. 999.0)
edit.target.polPredict	CHOICE		EnumRange DIS (OFF ON)
edit.target.inclination	FLOAT		DoubleRange DIS (-45.000 .. 45.000)
edit.target.satPolOffset	FLOAT		DoubleRange DIS (-45.000 .. 45.000)
edit.target.retryOnFault	CHOICE		EnumRange DIS (NEVER ONCE FOREVER)
edit.target.azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
edit.target.elevation	FLOAT		DoubleRange (0.000 .. 0.000)
edit.target.polarization	FLOAT		DoubleRange (0.00 .. 0.00)
edit.target.polarization2	FLOAT		DoubleRange DIS (0.00 .. 0.00)

name	***type***	***flags***	***range***
edit.target.name	TEXT		StringRange
edit.target.frequency	FLOAT		DoubleRange (1000.000 .. 40000.000)
edit.target.bcrx.attenuation	CHOICE		EnumRange DIS (30 dB 20 dB 10 dB 0 dB)
edit.target.bcrx.averaging	CHOICE		EnumRange DIS (0.1 Hz 0.2 Hz 0.5 Hz 1 Hz 2 Hz 5 Hz)
edit.target.bcrx.frequency	FLOAT		DoubleRange DIS (950.000 .. 40000.000)
edit.target.bcrx.bandwidth	CHOICE		EnumRange DIS (6 kHz 12 kHz 30 kHz 100 kHz)
edit.target.bcrx.polarization	CHOICE		EnumRange DIS (H V)
edit.target.bcrx.threshold	FLOAT		DoubleRange DIS (-999.99 .. 0.00)
edit.target.bcrx.speccomp	CHOICE		EnumRange DIS (OFF)
edit.target.bcrx.ftrack.on	CHOICE		EnumRange DIS (OFF ON)
edit.target.bcrx.ftrack.width	INTEGER		IntegerRange DIS (10 .. 1000)
edit.target.bcrx.ftrack.interval	INTEGER		IntegerRange DIS (1 .. 21600)
edit.target.bcrx.ssrch.delay	INTEGER		IntegerRange DIS (0 .. 600)
edit.target.bcrx.ssrch.enable	CHOICE		EnumRange DIS (OFF ON)
edit.target.bcrx.cn.on	CHOICE		EnumRange DIS (OFF C/N C/NO)
edit.target.bcrx.cn.frequency	FLOAT		DoubleRange DIS (950.000 .. 40000.000)
edit.target.bcrx.cn.interval	INTEGER		IntegerRange DIS (0 .. 0)
edit.target.bcrx.offset	FLOAT		DoubleRange DIS (0.00 .. 0.00)
edit.target.bcrx.output	FLOAT		DoubleRange DIS (0.000 .. 0.000)

name	*type*	*flags*	*range*
saveMode	CHOICE		EnumRange (ALL NO-POINTING)
initialPointingOverride	CHOICE		EnumRange (STORED-POSITION ORBIT TLE I11 MODEL FROM-TARGET)
target.name	TEXT		StringRange
target.0	TEXT	R/O	StringRange R/O
target.1	TEXT	R/O	StringRange R/O
target.2	TEXT	R/O	StringRange R/O
target.3	TEXT	R/O	StringRange R/O
target.4	TEXT	R/O	StringRange R/O
target.5	TEXT	R/O	StringRange R/O
target.6	TEXT	R/O	StringRange R/O
target.7	TEXT	R/O	StringRange R/O
target.8	TEXT	R/O	StringRange R/O
target.9	TEXT	R/O	StringRange R/O
target.10	TEXT	R/O	StringRange R/O
target.11	TEXT	R/O	StringRange R/O
target.12	TEXT	R/O	StringRange R/O
target.13	TEXT	R/O	StringRange R/O
target.14	TEXT	R/O	StringRange R/O
target.15	TEXT	R/O	StringRange R/O
target.16	TEXT	R/O	StringRange R/O
target.17	TEXT	R/O	StringRange R/O
target.18	TEXT	R/O	StringRange R/O
target.19	TEXT	R/O	StringRange R/O
target.20	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.21	TEXT	R/O	StringRange R/O
target.22	TEXT	R/O	StringRange R/O
target.23	TEXT	R/O	StringRange R/O
target.24	TEXT	R/O	StringRange R/O
target.25	TEXT	R/O	StringRange R/O
target.26	TEXT	R/O	StringRange R/O
target.27	TEXT	R/O	StringRange R/O
target.28	TEXT	R/O	StringRange R/O
target.29	TEXT	R/O	StringRange R/O
target.30	TEXT	R/O	StringRange R/O
target.31	TEXT	R/O	StringRange R/O
target.32	TEXT	R/O	StringRange R/O
target.33	TEXT	R/O	StringRange R/O
target.34	TEXT	R/O	StringRange R/O
target.35	TEXT	R/O	StringRange R/O
target.36	TEXT	R/O	StringRange R/O
target.37	TEXT	R/O	StringRange R/O
target.38	TEXT	R/O	StringRange R/O
target.39	TEXT	R/O	StringRange R/O
target.40	TEXT	R/O	StringRange R/O
target.41	TEXT	R/O	StringRange R/O
target.42	TEXT	R/O	StringRange R/O
target.43	TEXT	R/O	StringRange R/O
target.44	TEXT	R/O	StringRange R/O
target.45	TEXT	R/O	StringRange R/O
target.46	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.47	TEXT	R/O	StringRange R/O
target.48	TEXT	R/O	StringRange R/O
target.49	TEXT	R/O	StringRange R/O
target.50	TEXT	R/O	StringRange R/O
target.51	TEXT	R/O	StringRange R/O
target.52	TEXT	R/O	StringRange R/O
target.53	TEXT	R/O	StringRange R/O
target.54	TEXT	R/O	StringRange R/O
target.55	TEXT	R/O	StringRange R/O
target.56	TEXT	R/O	StringRange R/O
target.57	TEXT	R/O	StringRange R/O
target.58	TEXT	R/O	StringRange R/O
target.59	TEXT	R/O	StringRange R/O
target.60	TEXT	R/O	StringRange R/O
target.61	TEXT	R/O	StringRange R/O
target.62	TEXT	R/O	StringRange R/O
target.63	TEXT	R/O	StringRange R/O
target.64	TEXT	R/O	StringRange R/O
target.65	TEXT	R/O	StringRange R/O
target.66	TEXT	R/O	StringRange R/O
target.67	TEXT	R/O	StringRange R/O
target.68	TEXT	R/O	StringRange R/O
target.69	TEXT	R/O	StringRange R/O
target.70	TEXT	R/O	StringRange R/O
target.71	TEXT	R/O	StringRange R/O
target.72	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.73	TEXT	R/O	StringRange R/O
target.74	TEXT	R/O	StringRange R/O
target.75	TEXT	R/O	StringRange R/O
target.76	TEXT	R/O	StringRange R/O
target.77	TEXT	R/O	StringRange R/O
target.78	TEXT	R/O	StringRange R/O
target.79	TEXT	R/O	StringRange R/O
target.80	TEXT	R/O	StringRange R/O
target.81	TEXT	R/O	StringRange R/O
target.82	TEXT	R/O	StringRange R/O
target.83	TEXT	R/O	StringRange R/O
target.84	TEXT	R/O	StringRange R/O
target.85	TEXT	R/O	StringRange R/O
target.86	TEXT	R/O	StringRange R/O
target.87	TEXT	R/O	StringRange R/O
target.88	TEXT	R/O	StringRange R/O
target.89	TEXT	R/O	StringRange R/O
target.90	TEXT	R/O	StringRange R/O
target.91	TEXT	R/O	StringRange R/O
target.92	TEXT	R/O	StringRange R/O
target.93	TEXT	R/O	StringRange R/O
target.94	TEXT	R/O	StringRange R/O
target.95	TEXT	R/O	StringRange R/O
target.96	TEXT	R/O	StringRange R/O
target.97	TEXT	R/O	StringRange R/O
target.98	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.99	TEXT	R/O	StringRange R/O
target.100	TEXT	R/O	StringRange R/O
target.101	TEXT	R/O	StringRange R/O
target.102	TEXT	R/O	StringRange R/O
target.103	TEXT	R/O	StringRange R/O
target.104	TEXT	R/O	StringRange R/O
target.105	TEXT	R/O	StringRange R/O
target.106	TEXT	R/O	StringRange R/O
target.107	TEXT	R/O	StringRange R/O
target.108	TEXT	R/O	StringRange R/O
target.109	TEXT	R/O	StringRange R/O
target.110	TEXT	R/O	StringRange R/O
target.111	TEXT	R/O	StringRange R/O
target.112	TEXT	R/O	StringRange R/O
target.113	TEXT	R/O	StringRange R/O
target.114	TEXT	R/O	StringRange R/O
target.115	TEXT	R/O	StringRange R/O
target.116	TEXT	R/O	StringRange R/O
target.117	TEXT	R/O	StringRange R/O
target.118	TEXT	R/O	StringRange R/O
target.119	TEXT	R/O	StringRange R/O
target.120	TEXT	R/O	StringRange R/O
target.121	TEXT	R/O	StringRange R/O
target.122	TEXT	R/O	StringRange R/O
target.123	TEXT	R/O	StringRange R/O
target.124	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.125	TEXT	R/O	StringRange R/O
target.126	TEXT	R/O	StringRange R/O
target.127	TEXT	R/O	StringRange R/O
target.128	TEXT	R/O	StringRange R/O
target.129	TEXT	R/O	StringRange R/O
target.130	TEXT	R/O	StringRange R/O
target.131	TEXT	R/O	StringRange R/O
target.132	TEXT	R/O	StringRange R/O
target.133	TEXT	R/O	StringRange R/O
target.134	TEXT	R/O	StringRange R/O
target.135	TEXT	R/O	StringRange R/O
target.136	TEXT	R/O	StringRange R/O
target.137	TEXT	R/O	StringRange R/O
target.138	TEXT	R/O	StringRange R/O
target.139	TEXT	R/O	StringRange R/O
target.140	TEXT	R/O	StringRange R/O
target.141	TEXT	R/O	StringRange R/O
target.142	TEXT	R/O	StringRange R/O
target.143	TEXT	R/O	StringRange R/O
target.144	TEXT	R/O	StringRange R/O
target.145	TEXT	R/O	StringRange R/O
target.146	TEXT	R/O	StringRange R/O
target.147	TEXT	R/O	StringRange R/O
target.148	TEXT	R/O	StringRange R/O
target.149	TEXT	R/O	StringRange R/O
target.150	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.151	TEXT	R/O	StringRange R/O
target.152	TEXT	R/O	StringRange R/O
target.153	TEXT	R/O	StringRange R/O
target.154	TEXT	R/O	StringRange R/O
target.155	TEXT	R/O	StringRange R/O
target.156	TEXT	R/O	StringRange R/O
target.157	TEXT	R/O	StringRange R/O
target.158	TEXT	R/O	StringRange R/O
target.159	TEXT	R/O	StringRange R/O
target.160	TEXT	R/O	StringRange R/O
target.161	TEXT	R/O	StringRange R/O
target.162	TEXT	R/O	StringRange R/O
target.163	TEXT	R/O	StringRange R/O
target.164	TEXT	R/O	StringRange R/O
target.165	TEXT	R/O	StringRange R/O
target.166	TEXT	R/O	StringRange R/O
target.167	TEXT	R/O	StringRange R/O
target.168	TEXT	R/O	StringRange R/O
target.169	TEXT	R/O	StringRange R/O
target.170	TEXT	R/O	StringRange R/O
target.171	TEXT	R/O	StringRange R/O
target.172	TEXT	R/O	StringRange R/O
target.173	TEXT	R/O	StringRange R/O
target.174	TEXT	R/O	StringRange R/O
target.175	TEXT	R/O	StringRange R/O
target.176	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.177	TEXT	R/O	StringRange R/O
target.178	TEXT	R/O	StringRange R/O
target.179	TEXT	R/O	StringRange R/O
target.180	TEXT	R/O	StringRange R/O
target.181	TEXT	R/O	StringRange R/O
target.182	TEXT	R/O	StringRange R/O
target.183	TEXT	R/O	StringRange R/O
target.184	TEXT	R/O	StringRange R/O
target.185	TEXT	R/O	StringRange R/O
target.186	TEXT	R/O	StringRange R/O
target.187	TEXT	R/O	StringRange R/O
target.188	TEXT	R/O	StringRange R/O
target.189	TEXT	R/O	StringRange R/O
target.190	TEXT	R/O	StringRange R/O
target.191	TEXT	R/O	StringRange R/O
target.192	TEXT	R/O	StringRange R/O
target.193	TEXT	R/O	StringRange R/O
target.194	TEXT	R/O	StringRange R/O
target.195	TEXT	R/O	StringRange R/O
target.196	TEXT	R/O	StringRange R/O
target.197	TEXT	R/O	StringRange R/O
target.198	TEXT	R/O	StringRange R/O
target.199	TEXT	R/O	StringRange R/O
target.list	TEXT	R/O	StringRange R/O
target.goto	INTEGER	NOPRESET	IntegerRange (0 .. 99)
target.save	INTEGER	NOPRESET	IntegerRange (0 .. 99)

name	***type***	***flags***	***range***
target.delete	INTEGER	NOPRESET	IntegerRange (0 .. 99)
goto	INTEGER	NOPRESET	IntegerRange (0 .. 99)
grmc	TEXT	R/O	StringRange R/O
aux.in.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.5	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
aux.in.6	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
aux.in.7	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
aux.in.8	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
aux.out.1	INTEGER		IntegerRange (0 .. 1)
aux.out.2	INTEGER		IntegerRange (0 .. 1)
aux.out.3	INTEGER		IntegerRange (0 .. 1)
aux.out.4	INTEGER		IntegerRange (0 .. 1)
aux.out.5	INTEGER		IntegerRange (0 .. 1)
aux.out.6	INTEGER		IntegerRange (0 .. 1)
aux.out.7	INTEGER		IntegerRange (0 .. 1)
aux.out.8	INTEGER		IntegerRange (0 .. 1)
aux.out.9	INTEGER		IntegerRange DIS (0 .. 1)
aux.out.10	INTEGER		IntegerRange DIS (0 .. 1)
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ high limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ low limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EL high limit)

name	***type***	***flags***	***range***
faults.04	ALARM	R/O	AlarmFlagRange R/O (EL low limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (PL high limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (PL low limit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (AZ motor fault)
faults.08	ALARM	R/O	AlarmFlagRange R/O (AZ motor timeout)
faults.09	ALARM	R/O	AlarmFlagRange R/O (EL motor fault)
faults.10	ALARM	R/O	AlarmFlagRange R/O (EL motor timeout)
faults.11	ALARM	R/O	AlarmFlagRange R/O (PL motor fault)
faults.12	ALARM	R/O	AlarmFlagRange R/O (PL motor timeout)
faults.13	ALARM	R/O	AlarmFlagRange R/O (AZ motor stopped)
faults.14	ALARM	R/O	AlarmFlagRange R/O (EL motor stopped)
faults.15	ALARM	R/O	AlarmFlagRange R/O (PL motor stopped)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Beacon level)
faults.17	ALARM	R/O	AlarmFlagRange R/O (AZ peaking)
faults.18	ALARM	R/O	AlarmFlagRange R/O (EL peaking)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Model fit)

name	***type***	***flags***	***range***
faults.20	ALARM	R/O	AlarmFlagRange R/O (Model reset)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Hub/Limit fault)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Emergency stop)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Cabinet open/Handheld)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Beacon receiver)
faults.25	ALARM	R/O	AlarmFlagRange R/O (TLE data age)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TLE data parse)
faults.27	ALARM	R/O	AlarmFlagRange R/O (I11 data age)
faults.28	ALARM	R/O	AlarmFlagRange R/O (I11 data parse)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Soft Limits)
faults.30	ALARM	R/O	AlarmFlagRange R/O (PL2 high limit)
faults.31	ALARM	R/O	AlarmFlagRange R/O (PL2 low limit)
faults.32	ALARM	R/O	AlarmFlagRange R/O (PL2 motor fault)
faults.33	ALARM	R/O	AlarmFlagRange R/O (PL2 motor timeout)
faults.34	ALARM	R/O	AlarmFlagRange R/O (PL2 motor stopped)

6.7 SatService-ACU2-Serial

This page describes the device driver and the device window for the sat-nms ACU2 antenna controller. This driver uses the Serial Interface for communication with it. The configuration parameters of the ACU2 are accessible through the ACU web based user interface, they are not replicated as configuration parameters in the MNC device driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page displays/controls the antenna pointing. Actual readings and set points are shown in separate fields, so you can see where the antenna is going to while it is moving.
- --- Show the tracking state.
- --- Target list to load, save and delete targets on the ACU
- --- Set the tracking parameters
- --- Editor for TLE's parameters
- --- Editor for I11 (Intelsat) Parameters.
- --- Target Editor
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
address	n/a
timeSync	Enables or disables the date/time synchronization of the ACU with the MNC/ACU-IDU system time.
Serial	Remember to set the line parameters for the serial interface connected to the ACU. The RS232 interface of sat-nms ACU always operates at 9600 baud, no parity, 8 data bits, one stop bit. Defaults is *9600,8,N,1* .

Serial protocol selection Depending on the device address set, the sat-nms ACU either runs framed protocol with start/stop characters and checksum or it provides a dumb terminal interface. If the address is set in the ACU configuration (web page) to 'A' .. 'G' use Miteq-MOD95 protocol, if the address is 'NONE' select Terminal-CR-CRLF protocol.

Please remember to set the baudrate on the ACU.

Time Synchronization

If enabled via the setup option described above, the M&C System or ACU Indoor Unit (running this drivers) will send date and time in regular intervals to the ACU . The ACU will update its internal clock.

The ACU does not have a time zone configuration, so it simply uses the time send by the MNC/ACU-IDU. For example: If the system clock of the MNC/ACU-IDU runs on UTC, the ACU clock is also set to UTC

Do not enable time synchronization for the first time if you just doing step or adaptive tracking, because a large leap in time will produce a wrong tracking model. After first synchronization you can re-enable step or adaptive tracking again. It need approx. 40 seconds after enabling until the ACU updates its internal clock. Use the web browser interface to check the current date and time of the ACU.

This feature syncs the date and time between ACU and MNC/ACU-IDU. If you want to synchronize the date and time of the MNC/ACU-IDU server you can use an external NTP server. Please refer to the sat-nms M&C Installation Manual.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.caps.tracking	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.polar	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.memtrack	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.clswitch	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.rellevel	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
info.caps.standby	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.200targets	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.i11tle	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.4axes	BOOLEAN	R/O	BooleanRange R/O (true false)
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.poslog	TEXT	R/O	StringRange R/O
info.beaconType	CHOICE		EnumRange (SATNMS VOLTAGE)
internal.ibit	HEX	R/O	HexRange R/O (0 .. 0)
internal.ibi2	HEX	R/O	HexRange R/O (0 .. 0)
internal.ibitaux	HEX	R/O	HexRange R/O (0 .. 0)
internal.ibitaux2	HEX	R/O	HexRange R/O (0 .. 0)
internal.obit	HEX	R/O	HexRange R/O (0 .. 0)
internal.obi2	HEX	R/O	HexRange R/O (0 .. 0)
internal.tflt	HEX	R/O	HexRange R/O (0 .. 0)
internal.pcnt1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pcnt2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.caps	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.target	TEXT	R/O	StringRange R/O
internal.targetNo	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.time	TEXT	R/O	StringRange R/O
internal.yy	TEXT	R/O	StringRange R/O
internal.mo	TEXT	R/O	StringRange R/O
internal.dd	TEXT	R/O	StringRange R/O
internal.hh	TEXT	R/O	StringRange R/O
internal.mi	TEXT	R/O	StringRange R/O
internal.ss	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
internal.state	TEXT	R/O	StringRange R/O
internal.sleep	TEXT	R/O	StringRange R/O
internal.cmod	TEXT	R/O	StringRange R/O DIS
initialized	TEXT	R/O	StringRange R/O
pl2StepEnable	BOOLEAN	R/O	BooleanRange R/O (true false)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.timeSync	CHOICE	SAVE SETUP	EnumRange (ENABLED DISABLED)
config.bcrx.polAlias	TEXT	SETUP	StringRange
config.azLoLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.azHiLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.elLoLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.elHiLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.plLoLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.plHiLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.pl2LoLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.pl2HiLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.00 .. 0.00)
polarization2	FLOAT		DoubleRange DIS (0.00 .. 0.00)
clswitch	CHOICE		EnumRange DIS (LINEAR CIRCULAR UNKNOWN)
step	TEXT	NOPRESET	StringRange
standby	TEXT	NOPRESET	StringRange DIS
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	***type***	***flags***	***range***
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pl2	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
state.cl	TEXT	R/O	StringRange R/O DIS
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
state.stopped	BOOLEAN	R/O	BooleanRange R/O (true false)
state.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.absLevel	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
state.mode1	TEXT	R/O	StringRange R/O
state.azJitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.elJitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.azAmplitude	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.elAmplitude	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.trackHours	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.azModel	TEXT	R/O	StringRange R/O
state.elModel	TEXT	R/O	StringRange R/O
state.modelAge	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.targetTainted	TEXT	R/O	StringRange R/O
state.taintedText	TEXT	R/O	StringRange R/O
state.selectedTarget	INTEGER	R/O	IntegerRange R/O (0 .. 0)
trackMode	CHOICE		EnumRange (UNKNOWN OFF STEP ADAPTIVE)
interval	INTEGER		IntegerRange DIS (1 .. 1638)
stepSize	INTEGER		IntegerRange DIS (1 .. 100)
measDelay	INTEGER		IntegerRange DIS (1 .. 9999)
recoveryDelay	INTEGER		IntegerRange DIS (100 .. 30000)

name	***type***	***flags***	***range***
averaging	INTEGER		IntegerRange DIS (1 .. 25)
frequency	FLOAT		DoubleRange DIS (950.000 .. 40000.000)
levelThreshold	FLOAT		DoubleRange DIS (-999.00 .. 100.00)
jitterThreshold	INTEGER		IntegerRange DIS (0 .. 100)
smoothing	INTEGER		IntegerRange DIS (0 .. 6)
azMaxModel	CHOICE		EnumRange DIS (SMALL MEDIUM LARGE)
elMaxModel	CHOICE		EnumRange DIS (SMALL MEDIUM LARGE)
levelOffset	FLOAT		DoubleRange DIS (0.00 .. 0.00)
levelZero	TEXT	NOPRESET	StringRange DIS
spindleSaveMode	CHOICE		EnumRange DIS (OFF 1 2 3 4 5)
spindleSaveThresh	INTEGER		IntegerRange DIS (0 .. 100)
azOffset	FLOAT		DoubleRange DIS (-180.000 .. 180.000)
elOffset	FLOAT		DoubleRange DIS (-180.000 .. 180.000)
orbit	FLOAT		DoubleRange DIS (0.000 .. 0.000)
retryOnFault	CHOICE		EnumRange DIS (NEVER ONCE FOREVER)
polPredict	CHOICE		EnumRange DIS (OFF ON)
inclination	FLOAT		DoubleRange DIS (-45.000 .. 45.000)
satPolOffset	FLOAT		DoubleRange DIS (-45.000 .. 45.000)
reset	TEXT	NOPRESET	StringRange
motorStop	TEXT	NOPRESET	StringRange
motorReset	TEXT	NOPRESET	StringRange
clearTracking	TEXT	NOPRESET	StringRange
logPosition	TEXT	NOPRESET	StringRange

name	***type***	***flags***	***range***
clearTainted	TEXT	NOPRESET	StringRange
bcrx.attenuation	CHOICE		EnumRange DIS (30 dB 20 dB 10 dB 0 dB)
bcrx.averaging	CHOICE		EnumRange DIS (0.1 Hz 0.2 Hz 0.5 Hz 1 Hz 2 Hz 5 Hz)
bcrx.frequency	FLOAT		DoubleRange DIS (950.000 .. 40000.000)
bcrx.bandwidth	CHOICE		EnumRange DIS (6 kHz 12 kHz 30 kHz 100 kHz)
bcrx.polarization	CHOICE		EnumRange DIS (H V)
bcrx.threshold	FLOAT		DoubleRange DIS (-999.99 .. 0.00)
bcrx.speccomp	CHOICE		EnumRange DIS (OFF)
bcrx.ftrack.on	CHOICE		EnumRange DIS (OFF ON)
bcrx.ftrack.width	INTEGER		IntegerRange DIS (10 .. 1000)
bcrx.ftrack.interval	INTEGER		IntegerRange DIS (1 .. 21600)
bcrx.ssrch.delay	INTEGER		IntegerRange DIS (0 .. 600)
bcrx.ssrch.enable	CHOICE		EnumRange DIS (OFF ON)
bcrx.cn.on	CHOICE		EnumRange DIS (OFF C/N C/NO)
bcrx.cn.frequency	FLOAT		DoubleRange DIS (950.000 .. 40000.000)
bcrx.cn.interval	INTEGER		IntegerRange DIS (0 .. 0)
bcrx.offset	FLOAT		DoubleRange DIS (0.00 .. 0.00)
bcrx.output	FLOAT		DoubleRange DIS (0.000 .. 0.000)
saveMode	CHOICE		EnumRange (ALL NO-POINTING)
target.name	TEXT		StringRange
target.0	TEXT	R/O	StringRange R/O
target.1	TEXT	R/O	StringRange R/O
target.2	TEXT	R/O	StringRange R/O
target.3	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.4	TEXT	R/O	StringRange R/O
target.5	TEXT	R/O	StringRange R/O
target.6	TEXT	R/O	StringRange R/O
target.7	TEXT	R/O	StringRange R/O
target.8	TEXT	R/O	StringRange R/O
target.9	TEXT	R/O	StringRange R/O
target.10	TEXT	R/O	StringRange R/O
target.11	TEXT	R/O	StringRange R/O
target.12	TEXT	R/O	StringRange R/O
target.13	TEXT	R/O	StringRange R/O
target.14	TEXT	R/O	StringRange R/O
target.15	TEXT	R/O	StringRange R/O
target.16	TEXT	R/O	StringRange R/O
target.17	TEXT	R/O	StringRange R/O
target.18	TEXT	R/O	StringRange R/O
target.19	TEXT	R/O	StringRange R/O
target.20	TEXT	R/O	StringRange R/O
target.21	TEXT	R/O	StringRange R/O
target.22	TEXT	R/O	StringRange R/O
target.23	TEXT	R/O	StringRange R/O
target.24	TEXT	R/O	StringRange R/O
target.25	TEXT	R/O	StringRange R/O
target.26	TEXT	R/O	StringRange R/O
target.27	TEXT	R/O	StringRange R/O
target.28	TEXT	R/O	StringRange R/O
target.29	TEXT	R/O	StringRange R/O
target.30	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.31	TEXT	R/O	StringRange R/O
target.32	TEXT	R/O	StringRange R/O
target.33	TEXT	R/O	StringRange R/O
target.34	TEXT	R/O	StringRange R/O
target.35	TEXT	R/O	StringRange R/O
target.36	TEXT	R/O	StringRange R/O
target.37	TEXT	R/O	StringRange R/O
target.38	TEXT	R/O	StringRange R/O
target.39	TEXT	R/O	StringRange R/O
target.40	TEXT	R/O	StringRange R/O
target.41	TEXT	R/O	StringRange R/O
target.42	TEXT	R/O	StringRange R/O
target.43	TEXT	R/O	StringRange R/O
target.44	TEXT	R/O	StringRange R/O
target.45	TEXT	R/O	StringRange R/O
target.46	TEXT	R/O	StringRange R/O
target.47	TEXT	R/O	StringRange R/O
target.48	TEXT	R/O	StringRange R/O
target.49	TEXT	R/O	StringRange R/O
target.50	TEXT	R/O	StringRange R/O
target.51	TEXT	R/O	StringRange R/O
target.52	TEXT	R/O	StringRange R/O
target.53	TEXT	R/O	StringRange R/O
target.54	TEXT	R/O	StringRange R/O
target.55	TEXT	R/O	StringRange R/O
target.56	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.57	TEXT	R/O	StringRange R/O
target.58	TEXT	R/O	StringRange R/O
target.59	TEXT	R/O	StringRange R/O
target.60	TEXT	R/O	StringRange R/O
target.61	TEXT	R/O	StringRange R/O
target.62	TEXT	R/O	StringRange R/O
target.63	TEXT	R/O	StringRange R/O
target.64	TEXT	R/O	StringRange R/O
target.65	TEXT	R/O	StringRange R/O
target.66	TEXT	R/O	StringRange R/O
target.67	TEXT	R/O	StringRange R/O
target.68	TEXT	R/O	StringRange R/O
target.69	TEXT	R/O	StringRange R/O
target.70	TEXT	R/O	StringRange R/O
target.71	TEXT	R/O	StringRange R/O
target.72	TEXT	R/O	StringRange R/O
target.73	TEXT	R/O	StringRange R/O
target.74	TEXT	R/O	StringRange R/O
target.75	TEXT	R/O	StringRange R/O
target.76	TEXT	R/O	StringRange R/O
target.77	TEXT	R/O	StringRange R/O
target.78	TEXT	R/O	StringRange R/O
target.79	TEXT	R/O	StringRange R/O
target.80	TEXT	R/O	StringRange R/O
target.81	TEXT	R/O	StringRange R/O
target.82	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.83	TEXT	R/O	StringRange R/O
target.84	TEXT	R/O	StringRange R/O
target.85	TEXT	R/O	StringRange R/O
target.86	TEXT	R/O	StringRange R/O
target.87	TEXT	R/O	StringRange R/O
target.88	TEXT	R/O	StringRange R/O
target.89	TEXT	R/O	StringRange R/O
target.90	TEXT	R/O	StringRange R/O
target.91	TEXT	R/O	StringRange R/O
target.92	TEXT	R/O	StringRange R/O
target.93	TEXT	R/O	StringRange R/O
target.94	TEXT	R/O	StringRange R/O
target.95	TEXT	R/O	StringRange R/O
target.96	TEXT	R/O	StringRange R/O
target.97	TEXT	R/O	StringRange R/O
target.98	TEXT	R/O	StringRange R/O
target.99	TEXT	R/O	StringRange R/O
target.100	TEXT	R/O	StringRange R/O
target.101	TEXT	R/O	StringRange R/O
target.102	TEXT	R/O	StringRange R/O
target.103	TEXT	R/O	StringRange R/O
target.104	TEXT	R/O	StringRange R/O
target.105	TEXT	R/O	StringRange R/O
target.106	TEXT	R/O	StringRange R/O
target.107	TEXT	R/O	StringRange R/O
target.108	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.109	TEXT	R/O	StringRange R/O
target.110	TEXT	R/O	StringRange R/O
target.111	TEXT	R/O	StringRange R/O
target.112	TEXT	R/O	StringRange R/O
target.113	TEXT	R/O	StringRange R/O
target.114	TEXT	R/O	StringRange R/O
target.115	TEXT	R/O	StringRange R/O
target.116	TEXT	R/O	StringRange R/O
target.117	TEXT	R/O	StringRange R/O
target.118	TEXT	R/O	StringRange R/O
target.119	TEXT	R/O	StringRange R/O
target.120	TEXT	R/O	StringRange R/O
target.121	TEXT	R/O	StringRange R/O
target.122	TEXT	R/O	StringRange R/O
target.123	TEXT	R/O	StringRange R/O
target.124	TEXT	R/O	StringRange R/O
target.125	TEXT	R/O	StringRange R/O
target.126	TEXT	R/O	StringRange R/O
target.127	TEXT	R/O	StringRange R/O
target.128	TEXT	R/O	StringRange R/O
target.129	TEXT	R/O	StringRange R/O
target.130	TEXT	R/O	StringRange R/O
target.131	TEXT	R/O	StringRange R/O
target.132	TEXT	R/O	StringRange R/O
target.133	TEXT	R/O	StringRange R/O
target.134	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.135	TEXT	R/O	StringRange R/O
target.136	TEXT	R/O	StringRange R/O
target.137	TEXT	R/O	StringRange R/O
target.138	TEXT	R/O	StringRange R/O
target.139	TEXT	R/O	StringRange R/O
target.140	TEXT	R/O	StringRange R/O
target.141	TEXT	R/O	StringRange R/O
target.142	TEXT	R/O	StringRange R/O
target.143	TEXT	R/O	StringRange R/O
target.144	TEXT	R/O	StringRange R/O
target.145	TEXT	R/O	StringRange R/O
target.146	TEXT	R/O	StringRange R/O
target.147	TEXT	R/O	StringRange R/O
target.148	TEXT	R/O	StringRange R/O
target.149	TEXT	R/O	StringRange R/O
target.150	TEXT	R/O	StringRange R/O
target.151	TEXT	R/O	StringRange R/O
target.152	TEXT	R/O	StringRange R/O
target.153	TEXT	R/O	StringRange R/O
target.154	TEXT	R/O	StringRange R/O
target.155	TEXT	R/O	StringRange R/O
target.156	TEXT	R/O	StringRange R/O
target.157	TEXT	R/O	StringRange R/O
target.158	TEXT	R/O	StringRange R/O
target.159	TEXT	R/O	StringRange R/O
target.160	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.161	TEXT	R/O	StringRange R/O
target.162	TEXT	R/O	StringRange R/O
target.163	TEXT	R/O	StringRange R/O
target.164	TEXT	R/O	StringRange R/O
target.165	TEXT	R/O	StringRange R/O
target.166	TEXT	R/O	StringRange R/O
target.167	TEXT	R/O	StringRange R/O
target.168	TEXT	R/O	StringRange R/O
target.169	TEXT	R/O	StringRange R/O
target.170	TEXT	R/O	StringRange R/O
target.171	TEXT	R/O	StringRange R/O
target.172	TEXT	R/O	StringRange R/O
target.173	TEXT	R/O	StringRange R/O
target.174	TEXT	R/O	StringRange R/O
target.175	TEXT	R/O	StringRange R/O
target.176	TEXT	R/O	StringRange R/O
target.177	TEXT	R/O	StringRange R/O
target.178	TEXT	R/O	StringRange R/O
target.179	TEXT	R/O	StringRange R/O
target.180	TEXT	R/O	StringRange R/O
target.181	TEXT	R/O	StringRange R/O
target.182	TEXT	R/O	StringRange R/O
target.183	TEXT	R/O	StringRange R/O
target.184	TEXT	R/O	StringRange R/O
target.185	TEXT	R/O	StringRange R/O
target.186	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
target.187	TEXT	R/O	StringRange R/O
target.188	TEXT	R/O	StringRange R/O
target.189	TEXT	R/O	StringRange R/O
target.190	TEXT	R/O	StringRange R/O
target.191	TEXT	R/O	StringRange R/O
target.192	TEXT	R/O	StringRange R/O
target.193	TEXT	R/O	StringRange R/O
target.194	TEXT	R/O	StringRange R/O
target.195	TEXT	R/O	StringRange R/O
target.196	TEXT	R/O	StringRange R/O
target.197	TEXT	R/O	StringRange R/O
target.198	TEXT	R/O	StringRange R/O
target.199	TEXT	R/O	StringRange R/O
target.list	TEXT	R/O	StringRange R/O
target.goto	INTEGER	NOPRESET	IntegerRange (0 .. 99)
target.save	INTEGER	NOPRESET	IntegerRange (0 .. 99)
target.delete	INTEGER	NOPRESET	IntegerRange (0 .. 99)
target.loadList	TEXT	NOPRESET	StringRange
target.needUpdate	BOOLEAN	NOPRESET	BooleanRange (true false)
goto	INTEGER	NOPRESET	IntegerRange (0 .. 99)
grmc	TEXT	R/O	StringRange R/O
aux.in.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.5	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	***type***	***flags***	***range***
aux.in.6	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
aux.in.7	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
aux.in.8	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
aux.out.1	INTEGER		IntegerRange (0 .. 1)
aux.out.2	INTEGER		IntegerRange (0 .. 1)
aux.out.3	INTEGER		IntegerRange (0 .. 1)
aux.out.4	INTEGER		IntegerRange (0 .. 1)
aux.out.5	INTEGER		IntegerRange (0 .. 1)
aux.out.6	INTEGER		IntegerRange (0 .. 1)
aux.out.7	INTEGER		IntegerRange (0 .. 1)
aux.out.8	INTEGER		IntegerRange (0 .. 1)
aux.out.9	INTEGER		IntegerRange DIS (0 .. 1)
aux.out.10	INTEGER		IntegerRange DIS (0 .. 1)
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ high limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ low limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EL high limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (EL low limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (PL high limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (PL low limit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (AZ motor fault)
faults.08	ALARM	R/O	AlarmFlagRange R/O (AZ motor timeout)
faults.09	ALARM	R/O	AlarmFlagRange R/O (EL motor fault)
faults.10	ALARM	R/O	AlarmFlagRange R/O (EL motor timeout)
faults.11	ALARM	R/O	AlarmFlagRange R/O (PL motor fault)

name	*type*	*flags*	*range*
faults.12	ALARM	R/O	AlarmFlagRange R/O (PL motor timeout)
faults.13	ALARM	R/O	AlarmFlagRange R/O (AZ motor stopped)
faults.14	ALARM	R/O	AlarmFlagRange R/O (EL motor stopped)
faults.15	ALARM	R/O	AlarmFlagRange R/O (PL motor stopped)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Beacon level)
faults.17	ALARM	R/O	AlarmFlagRange R/O (AZ peaking)
faults.18	ALARM	R/O	AlarmFlagRange R/O (EL peaking)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Model fit)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Model reset)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Hub/Limit fault)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Emergency stop)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Cabinet open/Handheld)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Beacon receiver)
faults.25	ALARM	R/O	AlarmFlagRange R/O (TLE data age)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TLE data parse)
faults.27	ALARM	R/O	AlarmFlagRange R/O (I11 data age)
faults.28	ALARM	R/O	AlarmFlagRange R/O (I11 data parse)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Soft Limits)
faults.30	ALARM	R/O	AlarmFlagRange R/O (PL2 high limit)
faults.31	ALARM	R/O	AlarmFlagRange R/O (PL2 low limit)
faults.32	ALARM	R/O	AlarmFlagRange R/O (PL2 motor fault)
faults.33	ALARM	R/O	AlarmFlagRange R/O (PL2 motor timeout)

name	***type***	***flags***	***range***
faults.34	ALARM	R/O	AlarmFlagRange R/O (PL2 motor stopped)

6.8 SatService-ACU-Prediction

This page describes the *SatService-ACU-Prediction* device. This device is a supplement to the SatService-ACU-ODM device driver for the sat-nms ACU outdoor module.

The *SatService-ACU-Prediction* device adds access to the ACU's feature to calculate a 100 datapoint table from a given I11 or TLE dataset stored in the ACU's memory. This requires an ACU2 outdoor module (software version 3.0 or higher). The *SatService-ACU-Prediction* device works like a separate device driver which talks to the ACU-ODM in parallel to the basic SatService-ACU-ODM driver. The *SatService-ACU-Prediction* device may be configured in the same or in a different device thread (aka. INTERFACE), the HTTP protocol used does not demand any restriction on this.

To use the *SatService-ACU-Prediction* device, select the Type of ephemerides to use, the memory number, start time and time increment. Finally click "SUBMIT" and the result will be shown.

Using the device in an automated or remote controlled environment works exactly in the same way: Set the variables 'ephemerisMode', 'memoryNumber', 'startTime' and 'timeIncrement' to specify the parameters of the request. Finally set the 'predict' variable to an arbitrary value to send the request to the ACU-ODM

The 100 data points then can be read from the 'result.nn' variables, where 'nn' is in the range of '00' to '99'. The variable 'state' holds information about success or failure of the operation. If the request failed, all 'result.nn' variables contain empty strings and the 'state' variable tells about the failure's reason. Please give the device about one second time to answer after sending the request and before you read the result strings.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- ⌨ --- The main parameter page displays/controls request parameters and the result.
- ℹ --- The device info page.
- 🔧 --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there is one configuration parameter which must be set to make the software talk to the antenna controller.

parameter	***description***
address	This is the only parameter that must be set. Enter the antenna controller's IP address in 'dotted quad' notation here. Example: " **192.168.2.81** ".

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
ephemerisType	CHOICE		EnumRange (TLE I11)
memoryNumber	INTEGER		IntegerRange (1 .. 99)
selectedTLE	CHOICE		EnumRange (00 - NONE)
selectedI11	CHOICE		EnumRange DIS (00 - NONE)
startTime	TEXT		StringRange
timeIncrement	FLOAT		DoubleRange (0.01 .. 24.00)
predict	TEXT	NOPRESET	StringRange
result	TEXT	R/O	StringRange R/O
state	TEXT	R/O	StringRange R/O
result.00	TEXT	R/O	StringRange R/O
result.01	TEXT	R/O	StringRange R/O
result.02	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
result.03	TEXT	R/O	StringRange R/O
result.04	TEXT	R/O	StringRange R/O
result.05	TEXT	R/O	StringRange R/O
result.06	TEXT	R/O	StringRange R/O
result.07	TEXT	R/O	StringRange R/O
result.08	TEXT	R/O	StringRange R/O
result.09	TEXT	R/O	StringRange R/O
result.10	TEXT	R/O	StringRange R/O
result.11	TEXT	R/O	StringRange R/O
result.12	TEXT	R/O	StringRange R/O
result.13	TEXT	R/O	StringRange R/O
result.14	TEXT	R/O	StringRange R/O
result.15	TEXT	R/O	StringRange R/O
result.16	TEXT	R/O	StringRange R/O
result.17	TEXT	R/O	StringRange R/O
result.18	TEXT	R/O	StringRange R/O
result.19	TEXT	R/O	StringRange R/O
result.20	TEXT	R/O	StringRange R/O
result.21	TEXT	R/O	StringRange R/O
result.22	TEXT	R/O	StringRange R/O
result.23	TEXT	R/O	StringRange R/O
result.24	TEXT	R/O	StringRange R/O
result.25	TEXT	R/O	StringRange R/O
result.26	TEXT	R/O	StringRange R/O
result.27	TEXT	R/O	StringRange R/O
result.28	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
result.29	TEXT	R/O	StringRange R/O
result.30	TEXT	R/O	StringRange R/O
result.31	TEXT	R/O	StringRange R/O
result.32	TEXT	R/O	StringRange R/O
result.33	TEXT	R/O	StringRange R/O
result.34	TEXT	R/O	StringRange R/O
result.35	TEXT	R/O	StringRange R/O
result.36	TEXT	R/O	StringRange R/O
result.37	TEXT	R/O	StringRange R/O
result.38	TEXT	R/O	StringRange R/O
result.39	TEXT	R/O	StringRange R/O
result.40	TEXT	R/O	StringRange R/O
result.41	TEXT	R/O	StringRange R/O
result.42	TEXT	R/O	StringRange R/O
result.43	TEXT	R/O	StringRange R/O
result.44	TEXT	R/O	StringRange R/O
result.45	TEXT	R/O	StringRange R/O
result.46	TEXT	R/O	StringRange R/O
result.47	TEXT	R/O	StringRange R/O
result.48	TEXT	R/O	StringRange R/O
result.49	TEXT	R/O	StringRange R/O
result.50	TEXT	R/O	StringRange R/O
result.51	TEXT	R/O	StringRange R/O
result.52	TEXT	R/O	StringRange R/O
result.53	TEXT	R/O	StringRange R/O
result.54	TEXT	R/O	StringRange R/O
result.55	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
result.56	TEXT	R/O	StringRange R/O
result.57	TEXT	R/O	StringRange R/O
result.58	TEXT	R/O	StringRange R/O
result.59	TEXT	R/O	StringRange R/O
result.60	TEXT	R/O	StringRange R/O
result.61	TEXT	R/O	StringRange R/O
result.62	TEXT	R/O	StringRange R/O
result.63	TEXT	R/O	StringRange R/O
result.64	TEXT	R/O	StringRange R/O
result.65	TEXT	R/O	StringRange R/O
result.66	TEXT	R/O	StringRange R/O
result.67	TEXT	R/O	StringRange R/O
result.68	TEXT	R/O	StringRange R/O
result.69	TEXT	R/O	StringRange R/O
result.70	TEXT	R/O	StringRange R/O
result.71	TEXT	R/O	StringRange R/O
result.72	TEXT	R/O	StringRange R/O
result.73	TEXT	R/O	StringRange R/O
result.74	TEXT	R/O	StringRange R/O
result.75	TEXT	R/O	StringRange R/O
result.76	TEXT	R/O	StringRange R/O
result.77	TEXT	R/O	StringRange R/O
result.78	TEXT	R/O	StringRange R/O
result.79	TEXT	R/O	StringRange R/O
result.80	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
result.81	TEXT	R/O	StringRange R/O
result.82	TEXT	R/O	StringRange R/O
result.83	TEXT	R/O	StringRange R/O
result.84	TEXT	R/O	StringRange R/O
result.85	TEXT	R/O	StringRange R/O
result.86	TEXT	R/O	StringRange R/O
result.87	TEXT	R/O	StringRange R/O
result.88	TEXT	R/O	StringRange R/O
result.89	TEXT	R/O	StringRange R/O
result.90	TEXT	R/O	StringRange R/O
result.91	TEXT	R/O	StringRange R/O
result.92	TEXT	R/O	StringRange R/O
result.93	TEXT	R/O	StringRange R/O
result.94	TEXT	R/O	StringRange R/O
result.95	TEXT	R/O	StringRange R/O
result.96	TEXT	R/O	StringRange R/O
result.97	TEXT	R/O	StringRange R/O
result.98	TEXT	R/O	StringRange R/O
result.99	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange

6.9 SatService-Beacon-Receiver

This page describes the device driver and the device window for the sat-nms LBRX beacon receiver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the receive level and lets you set the basic receive parameters like frequency, bandwidth and input attenuation.
- --- This page lets you configure the C/N measurement mode of the receiver and displays the signal C/N if this mode is active.
- --- This page contains the parameters to configure the frequency tracking facility of the receiver.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

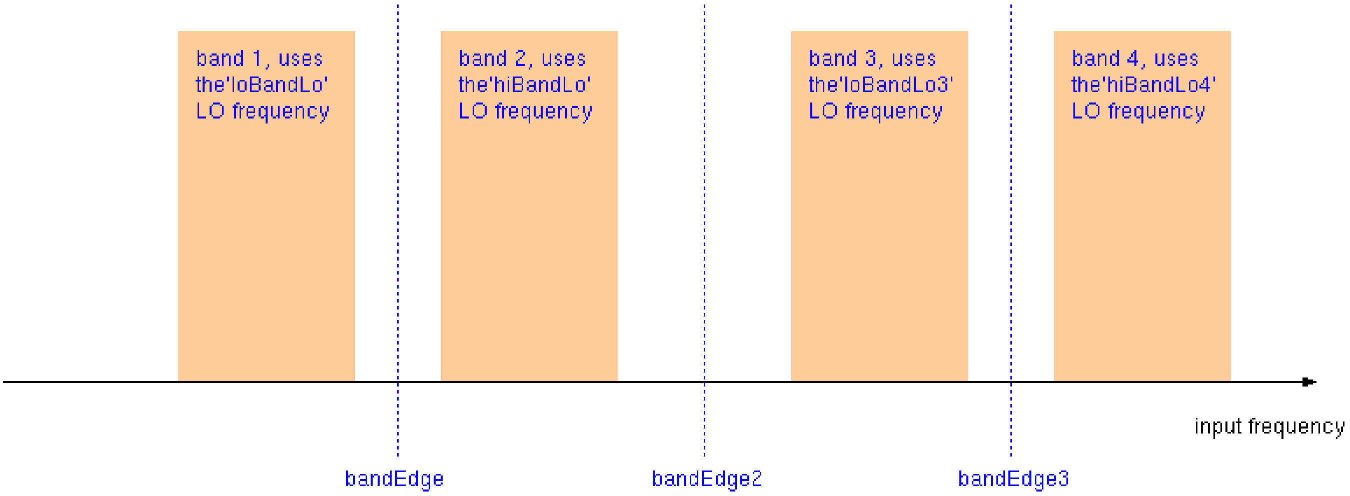
LO Frequency Calculations

The beacon receiver may store one or more local oscillator frequencies which are used to calculate the L-band frequency at the receiver input from the receive frequency at the antenna. Most beacon receivers support two frequency bands with separate LO frequencies. A model with four frequency bandy and 8-way input switch is also available. With the LO frequencies and frequency band edges setup correctly, the beacon receiver always shows and accepts the input frequency in terms of true receive frequency at the antenna rather than in terms of L-band frequency. Optionally a 22kHz tone may be modulated on the LNB power supply to switch the band.

All LO frequencies are to be specified in MHz. The LO is assumed to be below the input frequency unless the frequency is given as a negative value. In the latter case, the beacon receiver calculates for a LO frequency above the carrier.

Two-band receivers select the frequency band by the 'bandEdge' configuration parameter. Input frequencies below this value get the 'loBandLo' LO frequency applies, frequencies above the edge are calculated to L-band using the 'hiBandLo' LO frequency.

Four-band receivers provide three frequency band edges which divide the input frequency range into four bands. Four separate LO frequencies are applied are applied according to the actual band the selected frequency is within. For compatibility reasons, the lower frequency bands and the band edge in between are controlled by the parameters mentioned above. Two additional band edges and LO frequencies control the upper two frequency bands:


band 1, uses the'loBandLo' LO frequency
band 2, uses the'hiBandLo' LO frequency
band 3, uses the'loBandLo3' LO frequency
band 4, uses the'hiBandLo4' LO frequency
input frequency
bandEdge
bandEdge2
bandEdge3

With the 22kHzTone parameter set to AUTO, the 22 kHz tone gets activated if the input frequency is in frequency band 2 or in in band 4.

C/N Measurement Mode

In addition to the normal level measurement, the sat-nms LBRX beacon receiver is able to perform C/N measurements. This is done by measuring the level of the received noise at a frequency other than the receive frequency in regular intervals. Each measured signal level then gets converted to a C/N value by referring it to the noise level value. While the receiver measures the noise level, it freezes the displayed C/N value as well as the analog output voltage for this time.

The behavior of the C/N measurement function is controlled by three operational parameters:

- *Parameter Name* --- *Description*
- C/N Noise measurement --- This parameter controls if the receiver shall perform a plain input level measurement or a C/N measurement. The following measurement modes are available:
  - **OFF** --- The receiver performs a plain level measurement.
  - **IN** --- The receiver measures the signal / noise ratio. The The C/N measurement page shows the C/N value in dB. and the normalizes C/No value as well.
- Noise Measurement Frequency --- This parameter specifies the frequency at which the receiver shall measure the noise level at a certain interval. Like with the receive frequency, the LO frequency settings made at the Setup page are taken into account also for this frequency value.To get reasonable results with a C/N measurement, you should consider the following:
  1. The receiver does not change the LNB frequency band setting when it switches from the level measurement to the noise measurement. The LNB probably would change it's gain in this case. The noise measurement frequency hence must be in the same frequency band as the receive frequency.
  2. Measuring the noise level at the band edge may falsify the result due to the LNB's

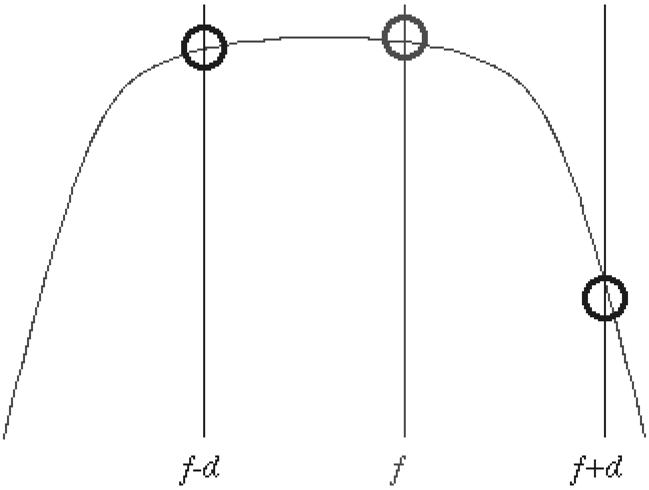
band filter. The measured noise level may be too low in this case.
3. You should verify with a spectrum analyzer, that no signal disturbs the noise measurement at the selected frequency.

- Noise Measurement Interval --- This parameter defines the interval at which the receiver inserts noise measurements in the C/N modes. The time is specified in seconds. 3600 secs being one hour is a suitable setting in most cases.

Frequency Tracking

The sat-nms LBRX beacon receiver using it's non-coherent receiver design, does not automatically follow an input signal drifting in frequency like a PLL receiver would do. To let the receiver compensate effects like an LNC frequency drift due to temperature or Doppler frequency shift, a frequency tracking algorithm has been implemented in the receiver.

With frequency tracking enabled, the receiver tunes the receive frequency a small amount up and down at certain intervals. Such a search step takes less than one second, while the frequency is detuned, the level reading and the analog output of the receiver stays frozen.



The step size used for a frequency search step depends in the measurement bandwidth, the receiver automatically selects the appropriate step size. Receiving a C/W signal, the receiver software expects a signal degradation of 1..3 dB when the frequency has been tuned up or down. The receiver evaluates the signal center frequency by comparing the measured level, at the old frequency *f* , at *f+d* and *f-d* to the known shape of the filter.

The behavior of the frequency track facility is controlled by three operational parameters:

Parameter Name	***Description***
Frequency Tracking	This parameter switches the the frequency tracking facility of the receiver ON or OFF. Please note, that switching off the frequency tracking does not reset the frequency offset which is applied by the frequency tracking. To reset the offset, set the receiver's frequency setting.

Parameter Name	***Description***
Frequency Tracking Interval	This parameter sets the interval on which the frequency tracking procedure operates. The value is in seconds. Recommended settings are 15 seconds to tune the receiver quickly to a frequency you do not know precisely. For normal operation a frequency tracking interval of one hour (3600 secs) is recommended.
Frequency Tracking width	With this setting you limit the frequency offset the frequency tracking procedure may apply to the nominal frequency. The frequency tracking never tunes the receiver to a frequency outside the specified range, a frequency track fault is generated if the tracked frequency reaches the limit.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the receiver.

- *parameter* --- *description*
- address --- The IP address of the receiver to control.
- loBandLo --- Enter the LO frequency for the lower frequency band here (MHz), if you want to specify the receive frequency in terms of true RF receive frequency. 'loBandLo' is applied for frequencies below 'bandEdge'. Enter 0 if you want the receive frequency to be displayed as receiver input (L-band) frequency.
- hiBandLo --- Enter the LO frequency for the upper frequency band here (MHz), if you want to specify the receive frequency in terms of true RF receive frequency. 'hiBandLo' is applied for frequencies above 'bandEdge'. Enter 0 if you want the receive frequency to be displayed as receiver input (L-band) frequency.
- loBandLo3 --- This LO frequency ist only available with beacon receivers which support 4 frequency bands. Enter the LO frequency for the frequency band between 'bandEdge2' and 'bandEdge3' here (MHz), if you want to specify the receive frequency in terms of true RF receive frequency. Enter 0 if you want the receive frequency to be displayed as receiver input (L-band) frequency.
- hiBandLo4 --- This LO frequency ist only available with beacon receivers which support 4 frequency bands. Enter the LO frequency for the frequency band above 'bandEdge3' here (MHz), if you want to specify the receive frequency in terms of true RF receive frequency. Enter 0 if you want the receive frequency to be displayed as receiver input (L-band) frequency.
- bandEdge --- Enter the edge between the receive frequency bands here (MHz). The receiver applies the 'loBandLo' LO frequency if the receive frequency is below the edge, the 'hiBandLo' LO frequency if it is above. If the 22kHzTone parameter is set to AUTO, the

22 kHz ist set accordingly.

- bandEdge2 --- This band edge value is only available with beacon receivers which support four frequency bands. Enter the edge between the receive frequency bands 2 and 3 here (MHz). The receiver applies the 'loBandLo' and 'hiBandLo' LO frequencies if the receive frequency is below this edge, the 'loBandLo3' and 'hiBandLo4' LO frequencies if it is above.

- bandEdge3 --- This band edge value is only available with beacon receivers which support four frequency bands. Enter the edge between the upper receive frequency bands 3 and 4 here (MHz). The receiver applies the 'loBandLo3' LO frequency if the receive frequency is below the edge, the 'hiBandLo4' LO frequency if it is above. If the 22kHzTone parameter is set to AUTO, the 22 kHz ist set accordingly.

- lnbVolt --- This parameter controls the LNB supply voltage provided by the receiver at it's input connector. The following settings are available:

  - **OFF** --- The D/C voltage is completely switched off.
  - **14V** --- The LNB supply voltage is 14V
  - **18V** --- The LNB supply voltage is 18V
  - **AUTO** --- The LNB supply voltage is switched on, the voltage depends on the 'Polarization' parameter. The voltage is 14V for vertical polarization, 18V for horizontal polarization.

- 22kHzTone --- This parameter controls the presence of a 22 kHz tone on the LNB supply voltage. The following settings are available:

  - **OFF** --- The 22 kHz tone is switched OFF.
  - **ON** --- The 22 kHz tone is switched ON.
  - **AUTO** --- The receiver automatically enables the 22 kHz tone depending on the receive frequency set. The tone is switched OFF for frequencies below the band edge, switched ON for frequencies above the band edge. The band edge is set with the 'bandEdge' parameter above.

- vOutScale --- This parameter defines the slope of the receiver's voltage output in V/dB. The output voltage has a range of 0 .. 10 V. Setting this parameter to 0.25V/dB lets the analog output cover a dynamic range of 40 dB.

- vOutOffset --- This parameter defines, which input level gives 0V output.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.note	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.loMode	TEXT	R/O	StringRange R/O
info.inputs	TEXT	R/O	StringRange R/O
config.showVoltage	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.loMode	CHOICE	SETUP	EnumRange (LEGACY 8-LO-MODE)
config.useMSwitch	CHOICE	SETUP	EnumRange (OFF ON)
config.minFreq	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.maxFreq	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.aliasPolH	TEXT	SAVE SETUP	StringRange
config.aliasPolV	TEXT	SAVE SETUP	StringRange
config.vOutScale	FLOAT	SETUP	DoubleRange (-5.0000 .. 5.0000)
config.vOutOffs	FLOAT	SETUP	DoubleRange (-200.00 .. 0.00)
config.lnbVolt	CHOICE	SETUP	EnumRange (OFF 14V 18V AUTO)
config.22kHzTone	CHOICE	SETUP	EnumRange (OFF ON AUTO)

name	***type***	***flags***	***range***
config.inputs	CHOICE	SAVE SETUP	EnumRange (4 8 16)
config.commandDelay	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 1.000)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL-ACCESS READONLY)
config.bandEdge	FLOAT		DoubleRange (0.000 .. 25000.000)
config.bandEdge2	FLOAT		DoubleRange (0.000 .. 25000.000)
config.bandEdge3	FLOAT		DoubleRange (0.000 .. 25000.000)
config.bandEdge4	FLOAT		DoubleRange (0.000 .. 25000.000)
config.bandEdge5	FLOAT		DoubleRange (0.000 .. 25000.000)
config.bandEdge6	FLOAT		DoubleRange (0.000 .. 25000.000)
config.bandEdge7	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo1Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo2Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo3Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo4Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo5Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo6Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo7Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo8Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo9Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)

name	***type***	***flags***	***range***
config.lo10Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo11Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo12Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo13Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo14Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo15Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo16Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo1HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo2HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo3HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo4HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo5HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo6HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo7HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo8HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo9HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo10HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo11HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo12HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo13HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo14HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo15HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)

name	***type***	***flags***	***range***
config.lo16HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo1LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo2LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo3LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo4LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo5LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo6LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo7LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo8LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo9LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo10LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo11LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo12LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo13LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo14LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo15LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo16LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo1Pol	CHOICE		EnumRange (H V H/V)
config.lo2Pol	CHOICE		EnumRange (H V H/V)
config.lo3Pol	CHOICE		EnumRange (H V H/V)
config.lo4Pol	CHOICE		EnumRange (H V H/V)
config.lo5Pol	CHOICE		EnumRange (H V H/V)
config.lo6Pol	CHOICE		EnumRange (H V H/V)
config.lo7Pol	CHOICE		EnumRange (H V H/V)
config.lo8Pol	CHOICE		EnumRange (H V H/V)
config.lo9Pol	CHOICE		EnumRange (H V H/V)

name	***type***	***flags***	***range***
config.lo10Pol	CHOICE		EnumRange (H V H/V)
config.lo11Pol	CHOICE		EnumRange (H V H/V)
config.lo12Pol	CHOICE		EnumRange (H V H/V)
config.lo13Pol	CHOICE		EnumRange (H V H/V)
config.lo14Pol	CHOICE		EnumRange (H V H/V)
config.lo15Pol	CHOICE		EnumRange (H V H/V)
config.lo16Pol	CHOICE		EnumRange (H V H/V)
config.lo1Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo2Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo3Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo4Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo5Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo6Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo7Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo8Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo9Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)

name	***type***	***flags***	***range***
config.lo10Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo11Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo12Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo13Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo14Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo15Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo16Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo1LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo2LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo3LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo4LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo5LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo6LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo7LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo8LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo9LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo10LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo11LnbVolt	CHOICE		EnumRange (14V 18V)

name	***type***	***flags***	***range***
config.lo12LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo13LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo14LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo15LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo16LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo1Tone	CHOICE		EnumRange (OFF ON)
config.lo2Tone	CHOICE		EnumRange (OFF ON)
config.lo3Tone	CHOICE		EnumRange (OFF ON)
config.lo4Tone	CHOICE		EnumRange (OFF ON)
config.lo5Tone	CHOICE		EnumRange (OFF ON)
config.lo6Tone	CHOICE		EnumRange (OFF ON)
config.lo7Tone	CHOICE		EnumRange (OFF ON)
config.lo8Tone	CHOICE		EnumRange (OFF ON)
config.lo9Tone	CHOICE		EnumRange (OFF ON)
config.lo10Tone	CHOICE		EnumRange (OFF ON)
config.lo11Tone	CHOICE		EnumRange (OFF ON)
config.lo12Tone	CHOICE		EnumRange (OFF ON)
config.lo13Tone	CHOICE		EnumRange (OFF ON)
config.lo14Tone	CHOICE		EnumRange (OFF ON)
config.lo15Tone	CHOICE		EnumRange (OFF ON)
config.lo16Tone	CHOICE		EnumRange (OFF ON)
frequency	FLOAT		DoubleRange (950.000 .. 40000.000)
attenuation	CHOICE		EnumRange (30 dB 20 dB 10 dB 0 dB)
gain	CHOICE		EnumRange (-30 dB -20 dB -10 dB 0 dB)

name	***type***	***flags***	***range***
bandwidth	CHOICE		EnumRange (6 kHz 12 kHz 30 kHz 100 kHz)
averaging	CHOICE		EnumRange (0.1 Hz 0.2 Hz 0.5 Hz 1 Hz 2 Hz 5 Hz)
threshold	FLOAT		DoubleRange (-999.99 .. 0.00)
polarization	CHOICE		EnumRange (H V)
speccomp	CHOICE		EnumRange (OFF)
ftrack.on	CHOICE		EnumRange (OFF ON)
ftrack.width	INTEGER		IntegerRange (10 .. 1000)
ftrack.interval	INTEGER		IntegerRange (1 .. 21600)
ftrack.offset	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ftrack.running	BOOLEAN	R/O	BooleanRange R/O (true false)
bg.inhibit	CHOICE		EnumRange (OFF ON)
power	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
cton	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
cton0	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
cn.on	CHOICE		EnumRange (OFF ON)
cn.frequency	FLOAT		DoubleRange (950.000 .. 19000.000)
cn.interval	INTEGER		IntegerRange (1 .. 21600)
ssrch.active	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ssrch.enable	CHOICE		EnumRange (OFF ON)
ssrch.delay	INTEGER		IntegerRange (0 .. 600)
ssrch.force	TEXT	NOPRESET	StringRange
internal.sccl	TEXT	R/O	StringRange R/O
internal.v	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.has4lo	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
internal.has8lo	TEXT	R/O	StringRange R/O
internal.haslimits	TEXT	R/O	StringRange R/O
internal.hasAlias	TEXT	R/O	StringRange R/O
internal.hasSummary	TEXT	R/O	StringRange R/O
internal.pol	CHOICE		EnumRange (H V)
internal.aliasPol	TEXT		StringRange
internal.hasNoInputs	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (D/C power supply)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Level threshold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Synth lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Frequency track)

6.10 Sysinfo

The Sysinfo logical device displays parameters of the server machine. This includes disk usage, cpu load, memory and sensor information from the mainboard if available on the specific hardware platform.

It is possible to configure alarm thresholds to create alarm events if the disk capacity goes low, if the 15min average cpu load is to high, if the available memory is to low or if a temperature is to high.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows cpu load and disk usage etc.
- FT --- The faults page.
- ℹ --- The device info page.
- 🔧 --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

parameter	***description***
diskThreshold	The threshold for the available disk capacity
loadThreshold	The threshold for the 15min average cpu load
memThreshold	The threshold for RAM
core0Threshold	The threshold for CPU temperature core0 if available
core1Threshold	The threshold for CPU temperature core1 if available
core1Threshold	The threshold for mainboard temperature 1 if available
core1Threshold	The threshold for mainboard temperature 2 if available
core1Threshold	The threshold for mainboard temperature 3 if available

Fault flags

The device provides the following fault flags:

- 'low disk capacity': the available space on the internal disk is lower than the configured threshold
- 'high average load': the average cpu load of the last 15min was higher than the configured threshold
- 'low memory': the available memory (RAM) is lower than the configured threshold
- 'high .. temperature': the temperature is higher than the configured threshold

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.hw	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.hostname	TEXT	R/O	StringRange R/O
info.cores	TEXT	R/O	StringRange R/O
info.devLicUsed	TEXT	R/O	StringRange R/O
config.execHost	CHOICE	SAVE SETUP	EnumRange (LOCAL REMOTE)
config.diskName	TEXT	SAVE SETUP	StringRange
config.diskThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loadThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.core0Threshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.core1Threshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.temp1Threshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.temp2Threshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.temp3Threshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
disk.name	TEXT	R/O	StringRange R/O
disk.size	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
disk.used	INTEGER	R/O	IntegerRange R/O (0 .. 0)
disk.available	INTEGER	R/O	IntegerRange R/O (0 .. 0)
disk.usage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mem.total	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mem.used	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mem.free	INTEGER	R/O	IntegerRange R/O (0 .. 0)
stat.uptime	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
stat.loadAvg1min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
stat.loadAvg5min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
stat.loadAvg15min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
temperature.core0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.core1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
fan.speed.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.speed.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.speed.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.speed.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.speed.5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.in0	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage.in1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage.in2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage.in3	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage.in4	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage.in5	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	***type***	***flags***	***range***
voltage.in6	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage.in7	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage.vbat	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
ps.state.1	TEXT	R/O	StringRange R/O
ps.state.2	TEXT	R/O	StringRange R/O
internal.ps1	TEXT	R/O	StringRange R/O
internal.ps2	TEXT	R/O	StringRange R/O
mail.count	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mail.subjects	TEXT	R/O	StringRange R/O
mnc.redundancy	CHOICE	R/O	EnumRange R/O (ERROR NONE MAIN BACKUP)
mnc.state	CHOICE	R/O	EnumRange R/O (ERROR ACTIVE STANDBY)
mnc.remote	TEXT	R/O	StringRange R/O
mnc.mainAddress	TEXT	R/O	StringRange R/O
mnc.backupAddress	TEXT	R/O	StringRange R/O
backend.redundancy	CHOICE	R/O	EnumRange R/O (ERROR NONE MAIN BACKUP)
backend.state	CHOICE	R/O	EnumRange R/O (ERROR ACTIVE STANDBY)
backend.remote	TEXT	R/O	StringRange R/O
backend.mainAddress	TEXT	R/O	StringRange R/O
backend.backupAddress	TEXT	R/O	StringRange R/O
backend.running	CHOICE	R/O	EnumRange R/O (YES NO)
internal.version	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.os	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.path	TEXT	R/O	StringRange R/O
internal.loadAvg1min	INTEGER	R/O	IntegerRange R/O (0 .. 0)

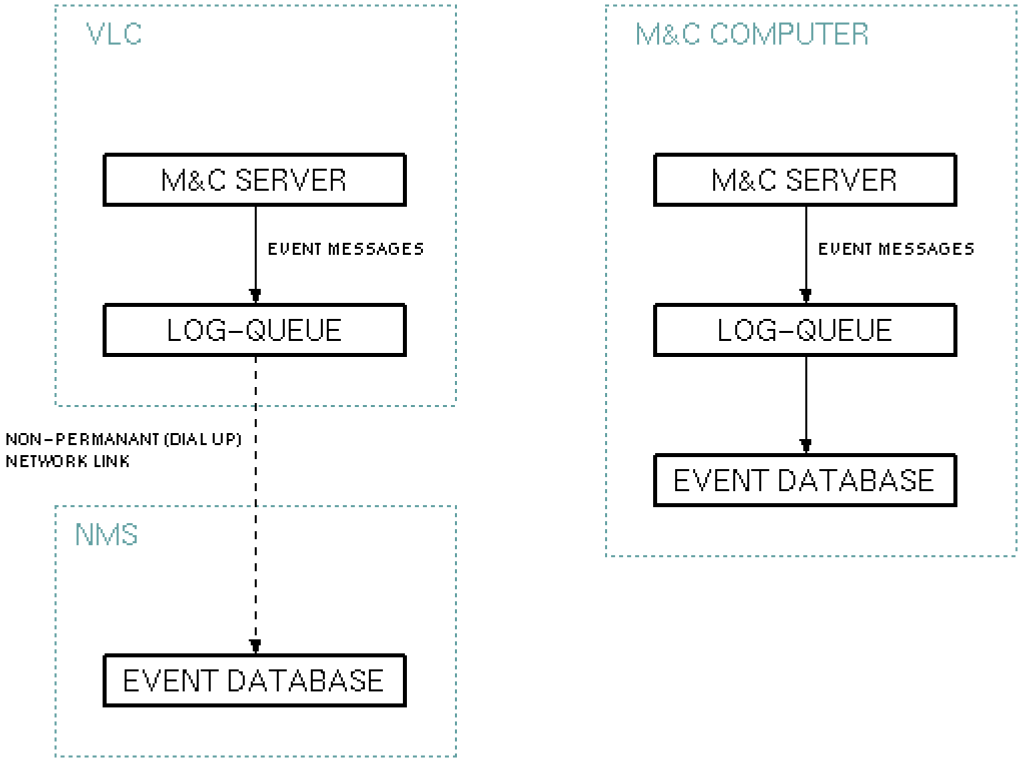
name	***type***	***flags***	***range***
internal.loadAvg5min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.loadAvg15min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (low disk capacity)
faults.02	ALARM	R/O	AlarmFlagRange R/O (high average load)
faults.03	ALARM	R/O	AlarmFlagRange R/O (high core 0 temp)
faults.04	ALARM	R/O	AlarmFlagRange R/O (high core 1 temp)
faults.05	ALARM	R/O	AlarmFlagRange R/O (high board 1 temp)
faults.06	ALARM	R/O	AlarmFlagRange R/O (high board 2 temp)
faults.07	ALARM	R/O	AlarmFlagRange R/O (high board 3 temp)
faults.08	ALARM	R/O	AlarmFlagRange R/O (low memory)
faults.09	ALARM	R/O	AlarmFlagRange R/O (power supply 1)
faults.10	ALARM	R/O	AlarmFlagRange R/O (power supply 2)
faults.11	ALARM	R/O	AlarmFlagRange R/O (ps 1 not installed)
faults.12	ALARM	R/O	AlarmFlagRange R/O (ps 2 not installed)
faults.13	ALARM	R/O	AlarmFlagRange R/O (new mail)
faults.21	ALARM	R/O	AlarmFlagRange R/O (fan 1)
faults.22	ALARM	R/O	AlarmFlagRange R/O (fan 2)
faults.23	ALARM	R/O	AlarmFlagRange R/O (fan 3)
faults.24	ALARM	R/O	AlarmFlagRange R/O (fan 4)
faults.25	ALARM	R/O	AlarmFlagRange R/O (fan 5)

name	***type***	***flags***	***range***
faults.26	ALARM	R/O	AlarmFlagRange R/O (mnc switched)
faults.27	ALARM	R/O	AlarmFlagRange R/O (backup mnc)
faults.28	ALARM	R/O	AlarmFlagRange R/O (backend switches)
faults.29	ALARM	R/O	AlarmFlagRange R/O (backup backend)
faults.30	ALARM	R/O	AlarmFlagRange R/O (backend running)

6.11 System

The System device with the name SYSTEM is present on every M&C/VLC server. It does not appear in the "vlc.setup" file, the server creates it automatically as a gateway to some of it's internal services. You mainly use the system device to tell the M&C/VLC server on which way and on which conditions the server shall send event messages to the event log.

To understand the parameters provided by the System device, it is necessary to know that the M&C/VLC server routes all faults and other messages through the so called "log-queue".



With a NMS/VLC network control system, all event messages are stored at a central place at the NMS. As the VLC usually has no permanent data link to the NMS, the log-queue holds the messages until a network link to the NMS is available and the queue gets flushed to the central event database.

The VLC is capable to dial to the NMS by itself. To limit the costs of the dial up connections, time delays may be specified to prevent the VLC from dialing with each message that occurs. If the log-queue gets flushed (which implies a dial-up from the VLC to the NMS) depends on the following. The queue gets flushed if:

- it contains at least one message of priority "ALARM", regardless of the age of the message.
- it contains at least one message of priority "FAULT" which is older than the parameter "fault delay".
- it contains at least one message of priority "INFO" which is older than the parameter "informational delay".
- the queue is full.
- a dial-up connection setup from NMS becomes on-line.

With a M&C system, the M&C computer itself provides the event database in most cases. The M&C server still routes the event messages through the log-queue, the timing parameter described above still apply, but they are set in a way that all messages go straight through into the event database.

Clock synchronisation

Another task of the SYSTEM device is to control the synchronization of the VLC's clock to the NMS time. This synchronization is essential for the link management as scheduled links are planned on the base of the NMS' clock but executed following the VLC's clock. The synchronization works as follows:

Each time, the NMS connects to a VLC, it sends the actual time to the VLC. The SYSTEM device compares the local time to the time received from the NMS and -- depending on the difference -- tunes the VLC's clock a little bit faster or slower. With VLCs being accessed at least once a month, this keeps the VLC's clock well in tune with the NMS without producing any leaps in the local time. This clock synchronization is made once when a VLC becomes 'on-line'.

If the clocks of NMS and VLC differ by more than 30 seconds, the SYSTEM device treas this as an error. It states a FAULT (it's icon at the user interface becomes red) and writes a message to the event log with a message text like:

Clock differs by 1523 secs from NMS time. Please reboot.

There may be several reasons for this fault:

- You have set the NMS clock manually, now each VLC states this fault until all VLCs are in tune again.
- The VLC was not accessed for a long time. A time drift of 15 secs / month is quite usual.
- The same applies if the VLC has been on-line for a very long time without interruption. As the clock synchronization only happens when the VLC *becomes* on-line, the clock will drift out of tune in this case.

When the SYSTEM device has recognized this fault, the VLC already is about to adjust the clock to the right time. It however cannot set the correct time immediately because the time leap could mess up the software's timeout recognition. The VLC smoothly tunes the clock with a maximum rate of 10 minutes per day. Hence, with relative small time differences you might ignore the fault, for larger differences a re-boot of the VLC will speed up this procedure, the correct time is set when the VLC starts (see section 'SYSTEM Device fault' below for details).

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- ☎ --- The parameters at this page define on which way event messages shall be sent to the event database.
    - NMSnetworkaddress --- Defines the host name / IP address of the NMS / event database server.
        - If you are configuring a VLC and this unit is connected to the NMS via Ethernet or a permanent SCC link, enter the IP address of the NMS.
        - For a VLC exclusively using dial up connections to the NMS enter the word "offline".
        - For M&C computers enter "localhost" or "127.0.0.1".
    - dial command 1 dial command 2 dial command 3 --- For a VLC you can specify up to three dial commands (AT.. command sequences) the VLC shall use to connect to the NMS. The VLC always tries dial command 1 first, if this fails it tries the second one an so on. You must enter the complete modem AT command, not only the phone number. This lets you prepend some additional commands like "X3" or some switches in front of the "D" command.With a M&C system or a VLC which communicates to NMS via Ethernet, let all three dial commands empty. This prevents the machine from doing any dial attempts.
- ❐ --- Contains the size and timing parameter of the log queue.
    - size --- The number of event messages the queue can hold. If the queue is full, the last (youngest) message in the queue gets replaced by a overflow message and all further messages are discarded.
    - fault delay --- The time delay for FAULT messages. With a M&C system, you should set this value to 0.
    - informational --- The time delay for INFO messages. With a M&C system, you should set this value to 0.
    - redial delay --- Defines the pause between two dialing attempts a VLC makes. This should be set to about 5 minutes, some modems decline more frequent dialing commands to the same phone number.
- 💾 --- Contains the macro backup function which permits to store all parameter settings known to the M&C/VLC into a macro definition for later retrieval.

- ℹ --- The device info page. You see the software revision number/date of the M&C/VLC server at this place

SYSTEM Device Fault

If the SYSTEM device states a fault, this indicates that the clocks of NMS and VLC are not in tune. The VLC will correct the clock difference automatically after some time, but actually any scheduled links from this VLC will start or stop at the wrong time.

If the time difference is relative small you may ignore the fault. The time difference is stated in the event log, search for the VLC ID and the text 'Clock differs'. Clicking the 'RESET' button on 'Maintenance' page of the device window will reset the fault until the VLC goes on-line the next time.

If the time difference is larger or if you need the clock to be set precisely at once, you may re-boot the VLC. This means to restart the VLC's operating system, not only to restart the VLC application. The procedure for this is as follows:

1. Establish an on-line connection to the VLC.
2. Open a telnet window to the VLC (Depending on your network settings, this might require that you are working directly at the NMS server).
3. Log in as user 'satnms' (password 'satnms').
4. Enter 'su', enter the password 'root' when prompted.
5. Enter 'reboot'.

The connection to the VLC will break after this, about 3 minutes later the VLC will be accessible again with the clock set properly.

Remarks

- When setting up a M&C system, ensure that "localhost" is set as the NMS network address and both delays for FAULT and INFO messages are set to 0. You will see the event messages delayed or event not at all in the event log if you leave these parameters unconfigured.
- VLC may be configured to talk to the NMS using a permanent network link and a dial up connection as a fall back. You might use this constellation with an SCC connection. If the satellite link fails, the VLC tries to use the terrestrial phone line to send it's fault messages.
- Setting the "fault delay" and "informational delay" parameters for a VLC which uses a dial-up connection to the NMS always is a compromise. If you are small values, you get informed earlier about malfunctions in the VLC, but there will be higher connection costs.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.type	TEXT	R/O	StringRange R/O

6.12 Antenna-Management

This page describes the device driver and the device window for the Antenna-Management logical device. This device lets you make a motorized antenna follow a satellite based on it's orbit position or ephemeris data. The Antenna-Management device is very similar to the Pointing-Robot device, extends this with a common database which stores the all satellite data like ephemeris data, stored center of box positions and administrative data of the satellite.

If used as a replacement for the legacy Antenna-Pointing device, the Antenna-Management device may be operated in "SIMPLE" mode, which reduces the number of parameter to focus the operation to a simple "mode the antenna to the stored position" for the selected satellite. See "Simple Mode" below for more information about this.

Using the device for recalling data from the database

To use the Antenna-Management device with the database, set the 'tracking data source' parameter to DATABASE. The workflow to track a satellite from the database is as follows:

1. Select the satellite from the 'satellite' list. You may enter partial satellite names to filter the list. There are several sort modes available for the satellite list, the list may be sorted by name, alias, orbit position and other satellite properties.
2. If the Antenna-Management device is setup to control a beacon receiver, select the satellite beacon to use.
3. If the Antenna-Management device is setup to control a TT&C equipment, select the telemetry and telecommand channels to use.
4. If you want to position or track following a tracking table, select the table file name set to use.
5. If you want to position or track following Intelsat I11 ephemerides, select the I11 data set to use.
6. Select the tracking mode to use (see below)
7. Click START to point the antenna to the satellite and to start tracking

The Antenna-Management device offers several tracking modes:

tracking mode	description
POINTING	The antenna moves to the pointing stored for the selected satellite in the database and stops.
STEPTRACK	The antenna moves to the pointing stored for the selected satellite in the database and then activates step track at the antenna controller.
MONOPULSE	The antenna moves to the pointing stored for the selected satellite in the database and then activates monopulse tracking at the antenna controller.

tracking mode	description
TLE	The antenna moves to the pointing calculated from the satellites TLE ephemerides and then tracks the satellite using this data.
I11	The antenna moves to the pointing calculated from the selected I11 ephemerides and then tracks the satellite using this data.
TABLETRACK	The antenna moves to the pointing calculated from the selected tracking table and then tracks the satellite using this data.
TLE-STEP	The antenna moves to the pointing calculated from the satellite's TLE ephemerides in the database and then activates step track at the antenna controller.
I11-STEP	The antenna moves to the pointing calculated from the selected I11 ephemerides in the database and then activates step track at the antenna controller.
TABLE-STEP	The antenna moves to the pointing calculated from the selected tracking table and then activates step track at the antenna controller.
TLE-MONOPULSE	The antenna moves to the pointing calculated from the satellite's TLE ephemerides in the database and then activates monopulse tracking at the antenna controller.
I11-MONOPULSE	The antenna moves to the pointing calculated from the selected I11 ephemerides in the database and then activates monopulse tracking at the antenna controller.
TABLE-MONOPULSE	The antenna moves to the pointing calculated from the selected tracking table and then activates monopulse at the antenna controller.

To stop tracking again click the STOP button.

Simple Mode

In SIMPLE mode only the tracking modes POINTING and STEPTRACK are available. The operation sequence differs from that used in FULL mode: First you select the desired tracking mode, then the satellite. The antenna moves to stored pointing for the satellite as soon as the satellite is selected from the list. No additional click to "start" is necessary.

"actual satellite" displays the satellite which recently has been selected for positioning and/or tracking. This persists over a restart of the sat-nms software.

Tracking file handling

The satellite database stores file names for table tracking. These names are assigned to a pair of satellite and antenna. During the satellite selection process the desired file name may be selected using the 'table tracking file name' parameter. The physical file however is **not** read from the database, it is read from the ~/robots directory ad the M&C computer. If the selected file is not found there, the Antenna-Management device complains with an error message.

Controlling the antenna without using the database

To use the TLE / I11 / table Track capabilities of the Antenna-Management device for a satellite which is not stored in the database or has no valid ephemeris data stored there, set the 'tracking data source' parameter to MANUAL and enter the tracking data at the 'Pointing Robot' page. The typical workflow there is to copy the ephemeris data from e.g. a webbrowser window and paste it into the 'ephemeris data' field at this page. The paragraphs below describe the various formats the Antenna-Management device can interpret.

Move the antenna along a satellite position calculated from ephemeris data

You can make the pointing robot follow a satellite position computed from usual ephemeris data sets. The robot recognizes the following data formats:

- **Keplerian elements** --- The robot recognizes and parses keplerian elements in the NASA 2-line format. The two lines of element definition may be prepended by an additional line with the satellite's name. A common source for Keplerian elements in this format is www.celestrak.com . The pointing robot uses the NORAD SDP4 model to predict a satellite's position from this type of data.
- **Intelsat elements** --- Intelsat publish ephemeris data for their communication satellites at their web site. The data format and the mathematical model for this data is Intelsat proprietary. The pointing robot recognizes this data type as well. You should copy the complete text of the ephemeris data set into the entry field to make the software recognize the data correctly.

Pointing the antenna to the nominal position of a satellite

If a single floating point number is entered to the ephemeris data field, the pointing robot interprets this as a nominal orbit position of the satellite (°E).

Move the antenna along a path given by a table

The pointing robot is capable to move the antenna along a path defined in a text file table. You either may copy the file to the IDU computer by means of FTP and enter the name of file to the ephemeris field or you may enter or cot/paste the table directly to this field. If you do the latter, the size of the table is limited to 32000 characters.

The Pointing-Robot provides a file selection dropdown widget to select files located in the /home/satnms/robot directory on the M&C&VLC server. To use this feature, select the file from the dropdown list and then click "SELECT". This copies the file name into the ephemeris data field and thus selects the file.

Files in the /home/satnms/robot directory which are no longer used may be deleted by selecting the file do delete from the drop down list and clicking DELETE. A file actually in use cannot be deleted, the request is silently ignored.

Polarization calculation

When calculating the antenna pointing from a orbital model or when interpolating the pointing from the table data, the Pointing-Robot also calculates a polarization angle. As each satellite uses its individual polarization offset, the calculated angle is normally for information only. With table tracking, the pointing robot may also command the calculated polarization angle at the antenna.

Setting "apply pol" to CALCED enables this feature. Setting "apply pol" to SAT-DB makes the devices use the polarization angle stored for this satellite in the database.

Tracking table format

The format of the table must be as follows: It contains the data as a three or four column table. Empty lines are ignored, comments starting with a '#' as well. The numbers in the table are parsed as floating point numbers which only may consist of decimal digits, one decimal point and an optional leading '-' if negative. The columns must be separated by an arbitrary number of space or tabulator characters. They have the following meanings:

- 1 --- time stamp The time stamp must be a Julian date with the time of day coded as a fraction of a day. Example: The Julian date for the common base of most computer clocks (1970-01-0100:00:00) is 2440587.5
- 2 --- azimuth angle (degrees, 0.0 == north)
- 3 --- elevation angle (degrees)

Beside this format, the Antenna-Management device accepts MMS files with the following column definitions:

- 1 --- date in the format YY/MM/DD
- 2 --- time in the format HH:MM:SS.sss (UTC)
- 3 --- elevation angle (degrees)
- 4 --- azimuth angle (degrees, 0.0 == north)

File import

The Pointing Robot GUI offers an import function which permits to copy pointing tables from the client computer to the server and thereby convert the file format to that one internally used by the sat-nms software.

Parsing order

If the pointing robot gets entered a new ephemeris data set, it tries to interpret the data in the following way:

1. It checks if the first line of the data specifies an existing file (residing on the IDU computer, therefore the file path should use Unix-style ('/') directory separators if necessary. If there is the specified file, it gets read an interpreted in a way, its file extension suggests: '*.tle' or '*.TLE': NASA 2-line ephemeris data. '*.i11' or '*.I11': Intelsat ephemeris data. Files with other extensions are interpreted as table data as described above.

2. It checks if the data is a valid NASA 2-line data set. A satellite name before the 2-line data is accepted and parsed separately if present.
3. It checks if the data contains valid Intelsat ephemeris data.
4. It checks if the first number in the first line of the data is a valid Julian date in the range 2005 .. 2050. In this case the robot assumes, that the ephemeris data contain a pointing table.
5. Finally, if nothing of the above mentioned formats did match, the robot tries to interpret the first line of the data as the nominal orbit position of the satellite.

If all of the parsing attempts fail, the pointing robot reports the model type to be UNKNOWN. Mostly this happens due to artifacts of previously entered elements which had not been deleted before the new ephemeris data was entered / pasted.

Remote control

The Pointing-Robot may be updated easily with updated pointing tables or ephemeris data by copying such files to the /home/public directory of the M&C system using the SMB file sharing protocol and then commanding the name of this file to the 'elements.input' variable of the Pointing-Robot using one of the M&C system's remote control ports.

ACU fault monitoring

The device can monitor the ACU's summary fault and show it in the device screen. The parameter 'antenna fault detection' may be set to one of the following settings:

- **IGNORE-ACU-FAULT** : The pointing robot ignores ACU faults, does not show them.
- **SHOW-ACU-FAULT** : The pointing robot shows ACU faults, continues to command the antenna controller even if there is an ACU fault.
- **STOP-ON-ACU-FAULT** : The pointing robot stops when it detects a fault of the antenna controller. It must be restarted manually after the ACU fault has been cleared.

Fault flags

In table tracking mode the Pointing-Robot reports two fault conditions regarding the end of table data:

File expired is reported if the end of the table data is reached.

Some hours before (configurable in the device setup) the**File expires soon** fault is reported. The latter fault condition appears as a WARNING by default.

Both fault conditions are reported if the Pointing-Robot is in table tracking mode, even if the tracking is stopped. To clear faults concerning an outdated but unused table, simply clear the "ephemeris data" field.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- 📡 --- The main parameter page lets you select a satellite and make the antenna track along the data read from the database for the selected satellite.
- 🤖 --- The "Pointing Robot" page lets you manually enter ephemeris data for the satellite to track and control the Pointing-Robot integrated into the Antenna-Management directly.
- 💾 --- The "Create Robot File" lets you create a table tracking file from ephemeris date read from the database for a given time span for offline use.
- F! --- The faults page.
- ℹ --- The device info page.
- 🔧 --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
azId	The parameter ID used to control the antenna's azimuth angle.
elId	The parameter ID used to control the antenna's elevation angle.
polId	The parameter ID used to control the antenna's polarization angle.

```
        // the connection may be closed meanwhile
```

| azStateId | The parameter ID reporting the antenna's actual azimuth angle. | | elStateId | The parameter ID reporting the antenna's actual elevation angle. | | polStateId | The parameter ID reporting the antenna's actual polarization angle. | | posReachedId | A parameter ID reporting the antenna's 'position reached' state as a boolean parameter. | | motorStopId | The parameter ID used to issue a "motor stop" at the antenna. Leave this empty if the antenna controller does not support this | | motorStopVal | The value to be sent to the parameter ID above in order to stop the motors | | memoryClearId | The parameter ID used to make the antenna clear its tracking memory. Leave this empty if the antenna controller does not support this | | memoryClearVal | The value to be sent to the parameter ID above in order to clear the tracking memory | | stepTrackId | The parameter ID used to switch the antenna controller's step track ON/OFF. Leave this empty if the antenna controller does not support step track | | stepTrackOn | The value to be sent to the parameter ID above in order to enable step track at the antenna controller | | stepTrackOff | The value to be sent to the parameter ID above in order to disable step track at the antenna controller | | targetId | This is a special configuration parameter for using the Antenna-Management device with a Vertex 7200 ACU. This ACU requires a target number to be set in order to run a step track. Set this to '.selectTarget' when controlling a Vertex-7200 ACU. | | monopulseTrackId | The parameter ID used to switch the antenna controller's monopulse tracking ON/OFF. Leave this empty if the antenna controller does not support monopulse tracking | | monopulseTrackOn | The value to be sent to the parameter ID above in order to enable monopulse tracking at the antenna controller | | monopulseTrackOff |

The value to be sent to the parameter ID above in order to disable monopulse tracking at the antenna controller | | targetNameId | If set, the Antenna-Management device sends the name and orbit position of the satellite actually tracked to this parameter. This can be used, to display the actually tracked satellite in the device screen of an ACU supporting such a parameter | | beacon.freqId | The parameter ID used to control the beacon receiver's frequency. The value sent is the receive frequency in MHz as the true RF frequency of the signal. Leave this empty if the Antenna-Management device shall not control a beacon receiver | | beacon.attenId | The parameter ID used to control the beacon receiver's attenuation. The parameter value is sent in dB as a floating point value | | beacon.levelId | The parameter ID reporting the beacon level. The parameter addressed by this ID should report the level as a dB value (either absolute or relative) | | beacon.polId | The parameter ID used to control the beacon polarization pane. | | tc.freqId | The parameter ID used to control the telecommand transmit frequency. The value sent is the transmit frequency in MHz as the true RF frequency of the signal. Leave this empty if the Antenna-Management device shall not control a telecommand transmitter | | tc.attenId | The parameter ID used to control the telecommand transmitter attenuation. The parameter value is sent in dB as a floating point value | | tc.polId | The parameter ID used to control the telecommand signal polarization pane. | | tm1.freqId | The parameter ID used to control the primary telemetry receiver's frequency. The value sent is the receive frequency in MHz as the true RF frequency of the signal. Leave this empty if the Antenna-Management device shall not control a telemetry receiver | | tm1.attenId | The parameter ID used to control the primary telemetry receiver attenuation. The parameter value is sent in dB as a floating point value | | tm1.polId | The parameter ID used to control the primary telemetry receiver polarization pane. | | tm2.freqId | The parameter ID used to control the secondary telemetry receiver's frequency. The value sent is the receive frequency in MHz as the true RF frequency of the signal. Leave this empty if the Antenna-Management device shall not control a telemetry receiver | | tm2.attenId | The parameter ID used to control the secondary telemetry receiver attenuation. The parameter value is sent in dB as a floating point value | | tm2.polId | The parameter ID used to control the secondary telemetry receiver polarization pane. | | reserved | Used by a RX-Channel device to lock the antenna. | | reserved.lo.x | Used by a RX-Channel device to lock one RX band of the antenna. | | reserved.lo.y | Used by a RX-Channel device to lock one RX band of the antenna. | | reserved.hi.x | Used by a RX-Channel device to lock one RX band of the antenna. | | reserved.hi.y | Used by a RX-Channel device to lock one RX band of the antenna. | | beamwidth | The antenna's half 3dB beam width in °. | | diameter | The antenna diameter in m. | | latitude | The antenna's geodetic location (latitude, °N). | | longitude | The antenna's geodetic location (longitude, °E). | | altitude | The antenna's altitude over sea level in m. | | ephemeris.update | If this is set to MANUAL, TLE and I11 data sets are read once from the database when tracking is started. If set to AUTO, TLE and I11 parameter are updated automatically while tracking is active and the data gets updated in the database. |

Database configuration

To give the Antenna-Management device access to the satellite database, the database server IP address as well as user name and password must be defined in the vlc.properties file. The property keys used for this are:

```
db.satlist.host=<IP4-address>
db.satlist.user=<user>
db.satlist.password=<password>
db.satlist.usessl=false
```

If SSL encryption shall be used for the database communication, set the #usessl' key to 'true'. In this case the M&C application's home directory must contain a hidden '.postgresql' directory with the database server's certificate, key and root certificate files. These files can be obtained from the database administrator, they have to be named postgresql.crt, postgresql.key.der and root.crt.

Changes to the vlc.properties file become effective after the next M&C restart.

Remarks

The accuracy of the calculations made by Pointing-Robot depend on the following:

- The geodetic location of the antenna must be known very precisely.
- The antenna's azimuth axis alignment must be precisely perpendicular.
- The clock of the M&C computer must be set to UTC.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.azId	TEXT	SAVE SETUP	StringRange
config.elId	TEXT	SAVE SETUP	StringRange
config.polId	TEXT	SAVE SETUP	StringRange
config.azStateId	TEXT	SAVE SETUP	StringRange
config.elStateId	TEXT	SAVE SETUP	StringRange
config.polStateId	TEXT	SAVE SETUP	StringRange

name	***type***	***flags***	***range***
config.posReachedId	TEXT	SAVE SETUP	StringRange
config.stepTrackId	TEXT	SAVE SETUP	StringRange
config.stepTrackOn	TEXT	SAVE SETUP	StringRange
config.stepTrackOff	TEXT	SAVE SETUP	StringRange
config.targetId	TEXT	SAVE SETUP	StringRange
config.targetNameId	TEXT	SAVE SETUP	StringRange
config.monopulseTrackId	TEXT	SAVE SETUP	StringRange
config.monopulseTrackOn	TEXT	SAVE SETUP	StringRange
config.monopulseTrackOff	TEXT	SAVE SETUP	StringRange
config.motorStopId	TEXT	SAVE SETUP	StringRange
config.motorStopVal	TEXT	SAVE SETUP	StringRange
config.clearMemoryId	TEXT	SAVE SETUP	StringRange
config.clearMemoryVal	TEXT	SAVE SETUP	StringRange
config.beacon.freqId	TEXT	SAVE SETUP	StringRange
config.beacon.levelId	TEXT	SAVE SETUP	StringRange
config.beacon.attenId	TEXT	SAVE SETUP	StringRange

name	***type***	***flags***	***range***
config.beacon.polId	TEXT	SAVE SETUP	StringRange
config.tc.freqId	TEXT	SAVE SETUP	StringRange
config.tc.atten1Id	TEXT	SAVE SETUP	StringRange
config.tc.atten2Id	TEXT	SAVE SETUP	StringRange
config.tc.polId	TEXT	SAVE SETUP	StringRange
config.tm1.freqId	TEXT	SAVE SETUP	StringRange
config.tm1.attenId	TEXT	SAVE SETUP	StringRange
config.tm1.polId	TEXT	SAVE SETUP	StringRange
config.tm2.freqId	TEXT	SAVE SETUP	StringRange
config.tm2.attenId	TEXT	SAVE SETUP	StringRange
config.tm2.polId	TEXT	SAVE SETUP	StringRange
config.reserved	CHOICE	SETUP	EnumRange (NO YES)
config.reserved.lo.x	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.reserved.lo.y	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.reserved.hi.x	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.reserved.hi.y	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.beamwidth	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)

name	***type***	***flags***	***range***
config.diameter	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.latitude	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.longitude	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.altitude	FLOAT	SAVE SETUP	DoubleRange (0 .. 0)
config.ephemeris.update	CHOICE	SAVE SETUP	EnumRange (MANUAL AUTO)
config.uiMode	CHOICE	SAVE SETUP	EnumRange (FULL SIMPLE)
config.fixedAntenna	CHOICE	SAVE SETUP	EnumRange (NO YES)
dataSource	CHOICE	SAVE	EnumRange (DATABASE MANUAL)
satsort	CHOICE	SAVE	EnumRange (NAME ALIAS ORBIT NORAD FLIGHT)
reconnect	TEXT	NOPRESET	StringRange
poslist	CHOICE		EnumRange (?)
satlist	CHOICE		EnumRange (?)
satellite	CHOICE		EnumRange (?)
copositioned	TEXT	R/O	StringRange R/O
noradNumber	INTEGER	R/O	IntegerRange R/O (0 .. 0)
intDesignator	TEXT	R/O	StringRange R/O
orbit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
inclination	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
alias	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
trackMode	CHOICE	SAVE	EnumRange (POINTING STEPTRACK MONOPULSE TLE I11 TABLETRACK TLE-STEP I11-STEP TABLE-STEP TLE-MONOPULSE I11-MONOPULSE TABLE-MONOPULSE)
beacon	CHOICE		EnumRange DIS (?)
tcchan	CHOICE		EnumRange DIS (?)
tmchan	CHOICE		EnumRange DIS (?)
i11	CHOICE		EnumRange (?)
table	CHOICE		EnumRange (?)
ephemeris	TEXT	R/O	StringRange R/O
start	TEXT	NOPRESET	StringRange
stop	TEXT	NOPRESET	StringRange
reload	TEXT	NOPRESET	StringRange
gotoTLE	TEXT	NOPRESET	StringRange
mayStart	BOOLEAN	R/O	BooleanRange R/O (true false)
mayStop	BOOLEAN	R/O	BooleanRange R/O (true false)
azimuth	FLOAT	NOPRESET	DoubleRange (0.000 .. 0.000)
elevation	FLOAT	NOPRESET	DoubleRange (0.000 .. 0.000)
polarization	FLOAT	NOPRESET	DoubleRange (0.000 .. 0.000)
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.activeFault	BOOLEAN	R/O	BooleanRange R/O (true false)
state.latchedFault	BOOLEAN	R/O	BooleanRange R/O (true false)
state.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.usePol	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
initial.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
initial.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
initial.pl	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
initial.state	TEXT	R/O	StringRange R/O
satoperator.name	TEXT	R/O	StringRange R/O
satoperator.contact	TEXT	R/O	StringRange R/O
satoperator.exists	BOOLEAN	R/O	BooleanRange R/O (true false)
saved.satelliteid	INTEGER	SAVE	IntegerRange (0 .. 0)
saved.i11id	INTEGER	SAVE	IntegerRange (0 .. 0)
saved.tableid	INTEGER	SAVE	IntegerRange (0 .. 0)
saved.trackMode	INTEGER	SAVE	IntegerRange (0 .. 0)
tracking	CHOICE	SAVE	EnumRange (OFF ON)
interval	INTEGER	SAVE	IntegerRange (0 .. 3600)
calculatePol	CHOICE	SAVE	EnumRange (OFF CALCED SAT-DB)
antFaultMode	CHOICE	SAVE	EnumRange (IGNORE-ACU-FAULT SHOW-ACU-FAULT STOP-ON-ACU-FAULT)
expireWarning	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
elements.input	TEXT		StringRange
eval.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
eval.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
eval.pl	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
eval.orbit	TEXT	R/O	StringRange R/O
state.short	TEXT	R/O	StringRange R/O
state.long	TEXT	R/O	StringRange R/O
state.time	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
state.satellite	TEXT	R/O	StringRange R/O
state.alias	TEXT	R/O	StringRange R/O
state.intDesignator	TEXT	R/O	StringRange R/O
state.noradNumber	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.trackMode	TEXT	R/O	StringRange R/O
state.stepTrack	TEXT	R/O	StringRange R/O
offset.az	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
offset.el	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
offset.pl	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
hysteresis.az	FLOAT	SAVE	DoubleRange (0.000 .. 10.000)
hysteresis.el	FLOAT	SAVE	DoubleRange (0.000 .. 10.000)
hysteresis.pl	FLOAT	SAVE	DoubleRange (0.000 .. 10.000)
elements.type	TEXT	R/O	StringRange R/O
elements.age	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
elements.epoch	TEXT	R/O	StringRange R/O
elements.validFrom	TEXT	R/O	StringRange R/O
elements.validTo	TEXT	R/O	StringRange R/O
resetLatched	TEXT	NOPRESET	StringRange
files	CHOICE		EnumRange (NONE)
selectFile	TEXT	NOPRESET	StringRange
deleteFile	TEXT	NOPRESET	StringRange
forceUpdateFileList	TEXT	NOPRESET	StringRange
wf.startTime	TEXT		StringRange
wf.endTime	TEXT		StringRange
wf.step	INTEGER		IntegerRange (1 .. 7200)
wf.fileName	TEXT		StringRange
wf.write	TEXT	NOPRESET	StringRange

name	***type***	***flags***	***range***
wf.mayWrite	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Database access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Antenna Controller)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Beacon Receiver)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Input Device)
faults.05	ALARM	R/O	AlarmFlagRange R/O (File expired)
faults.06	ALARM	R/O	AlarmFlagRange R/O (File expires soon)
faults.07	ALARM	R/O	AlarmFlagRange R/O (ACU Latched)

6.13 Antenna-Tracking

Please note, the Antenna-Tracking device is no longer maintained andshould not be used for new M&C installations.

The Antenna-Tracking logical device is a 'high level' driver for motor driven antennas. It is used together with a driver for the type of antenna controller used. To operate the device in step track modes, also a beacon receiver is required. The Antenna-Tracking device provides three mayor tracking modes:

- **STEP** --- In plain step track mode, the device performs small test steps with the antenna in regular intervals to see if the receive level becomes better or worse for another antenna position. With the knowledge of the antenna pattern's shape, the tracking engine computes an optimized position from measurements taken at the old position and at the new one. Many parameters such as the step size, averaging times and the overall repetition interval are configurable by the operator.
- **ADAPTIVE** --- The adaptive tracking mode works much like the plain step track, however the tracking engine in background computes a mathematical model of the antenna motions from the step track results. After a tracking step the antenna is moved to the position computed from this model rather than to the measured maximum. This significantly smooths the path at which the antenna follows the satellite. Another great advantage of this mode is that the antenna follows the computed model even if the

beacon reception fails. With the plain step track mode, the antenna would freeze in this case until the satellite beacon appears again.

- **PROGRAM** --- With the third tracking mode called program track, the antenna follows a position computed from the satellite's ephemeris data. The tracking engine accepts Keplerian elements in NASA 2-line format or the proprietary Intelsat ephemeris data format for this. For this tracking mode, no beacon reception is required, however the ephemeris data sets must be updated every few days.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Basic tracking parametersThe primary page contains the basic tracking parameters like the nominal antenna position or the mayor tracking mode. This page also gives access to the logging and analysis tools built into the tracking device. The most important parameters are explained below:**satellite name:** This ist the name of the satellite as it appears in the list of stored targets. When you read in ephemeris data which contains a satellite name, the latter gets copied into this field. You may however enter any other name.**nominal orbit:** With this parameter the satellite's nominal orbit position is specified. There are two applications for the nominal orbit value in the software: The tracking device is capable to evaluate the antenna pointing angles for the orbit position, you may move the antenna to the resulting position in order to find a new satellite quickly. Also the orbit value is used to identify and sort satellites in the target list.The nominal orbit position has to be entered as 'deg. east' To specify a satellite at 10° West, either enter -10.0 or 350.0, the software accepts both notations.**tracking mode:** With this parameter you select the major tracking mode. Available modes are:
    - STEP --- Plain step track mode.
    - ADAPTIVE --- Step track with mathematical model assistance / backup.
    - PROGRAM --- Program track following given ephemeris data. **interval:** Defines the overall tracking interval. The tracking device performs it's steps not faster than defined with this parameter. If an optimization step takes longer than this interval, this is not treated an a fault. The tracking device will start the next step immediately after the first one has finished. Set this value to 0 in order to force the tracking to operate at a maximum rate.The effective duration of one optimization step primarily depends on the integration times and delays defined with the advanced step track parameters. But also the fluctuation of the beacon level reading has an impact to the duration of a tracking step. If the standard variation of the readings will exceed a configurable threshold, the tracking algorithm will try to repeat the measurement several times.**az/el nominal:** These fields define the nominal position of the antenna for this satellite. The nominal position is the center of the box in which the antenna tracks the satellite. The size of this box is set on the 'advanced tracking parameters' page described below. The nominal position also is that one the antenna moves to, if the satellite's tracking settings get recalled from a preset memory.Changing one of the nominal position values while the tracking mode is 'OFF' immediately moves the antenna to the given position. With tracking

enabled, changing a nominal position value does not move the antenna, only the box limits are shifted accordingly.**polarization:** The polarization value directly controls the antenna's polarization offset motor (if available). None of the tracking algorithms changes this value, in must be optimized manually when a satellite is going to be tracked the first time.**beacon frequency:** This parameter controls the beacon receiver's receive frequency. The true RF receive frequency value is required, not an L-band frequency. It is necessary to enter this value even if the beacon receiver is not controlled by the tracking M&C system. This is because the tracking engine uses the beacon frequency to compute the tracking step sizes.**beacon pol.:** With beacon receivers which are capable to switch the input polarization, this parameter permits to select the polarization plane of the beacon to receive.**Y/T PLOT:** Clicking to the 'Y/T PLOT' button opens a window which shows a diagram of the antenna movement and the beacon level over the time. If you leave this window opened, it gets updated with each tracking step the antenna makes. See the paragraph 'built in analysis tools' below for further information.**X/Y PLOT:** Clicking to the 'X/Y PLOT' button opens a window which shows a diagram of the antenna movement in both axes. If you leave this window opened, it gets updated with each tracking step the antenna makes. See the paragraph 'built in analysis tools' below for further information.**STEP LOG:** Clicking to the 'STEP LOG' button opens a window where the program shows a live report what the tracking engine actually is doing. The log in this window is cleared with each new tracking step. If 'Y/T PLOT' diagram shows many unsuccessful tracking steps, watching the 'STEP LOG' reports may give an explanation why the tracking engine discards so many steps.**RESET POS:** Moves the antenna back to it's nominal position**RESET MEM:** Clears the tracking memory used by the ADAPTIVE tracking mode.**orbital model:** Displays the type and age of the orbital model used by the PROGRAM tracking mode. While in ADAPTIVE tracking mode 'ADAPTIVE' is displayed.**operation state:** Reports the actual operation state of the tracking engine. This mainly reflects the major tracking mode, but also contains some additional information, e.g. which type of ADAPTIVE model the tracking engine actually uses.**tracking step:** Reports the tracking step which recently has been done. This exactly matches the last line of the 'STEP LOG' display.**tracking offset:** These fields report the actual offset of the antenna pointing from the nominal position.**beacon level:** Displays the beacon level as actually read from the beacon receiver.

- 💾 --- Satellite listThe satellite list page permits to store tracking settings in a target memory pool which is unique to this instance of the Antenna-Tracking device. This is unlike the standard device preset function if the sat-nms software which stores device settings for all instances of one device type at a central place. The following functions are provided by this page:**Save actual satellite:** Clicking to this button stores the actual tracking parameters together with the nominal antenna pointing into a data file named after the actual satellite name followed by the nominal orbit position. If a file of this name already exists, it gets replaced.**Delete seleted entry** Deletes the selected (yellow colored) target memory file.**Load selected parameters** Loads all tracking parameters except the nominal antenna pointing from the selected target memory. This is intended to be used to copy the tracking settings from an already known satellite to a new one, which actually is being set up.**Track selected satellite** Loads all settings including the nominal antenna pointing and the tracking mode. Moves the antenna to the nominal pointing position and starts the selected tracking mode there.

- --- Orbital modelAt this page you may define an orbital model for the satellite which is used with the PROGRAM tracking mode. The orbital model also may be used to find a satellite more precisely than by it's nominal orbital position.The tracking engine accepts either Keplerian elements in NASA 2-line format or Intelsat ephemeris data. The software automatically recognized the data format and chooses the appropriate orbital model. A common source for actual Keplerian ephemeris data is www.celetrak.com , weekly updated Intelsat orbital data may be found on the Intelsat web site.The easiest way to enter the orbital data to the tracking system is by copy & paste from the source web site. Clear the orbital data field, mark the desired orbital data in the web browser window and copy it into the orbital data entry field (use CTRL-C/CTRL-V on Windows based PC, the middle mouse button on Linux boxes like the sat-nms ACU indoor unit. For Intelsat data, copy the whole web page which shows the ephemeris data set. For Keplerian data copying the two lines containing the element data is required. When using data from Celestrak, you optionally may copy also the comment line above the element data, the satellite name will be adapted automatically in this case.Once you entered / copied the orbital data and change the input focus to another field, the software analyzed the data and updates the fields at the top of the page. The 'recognized orbital model' shows the type and age of the interpreted ephemeris data. The 'satellite name' and 'nominal orbit' are updated accordingly, if this information is present in the entered data. If you select the PROGRAM tracking mode on the primary parameter page, the antenna follows the position of the satellite as computed from the ephemeris data.Using the buttons below the orbital data entry field you may compute the actual antenna pointing either from the entered ephememeris data or from the satellite's nominal orbit position. Clicking to the large button at the bottom move the antenna to the computed position. This function is intended to be used to find a new satellite quickly. If there is no orbital data available for the satellite you search, you may enter it's nominal orbit position directly and compute the antenna pointing from this.The accuracy of the calculated antenna positions heavily depends on the antenna pointing calibration and on the accuracy of the antenna's geodetic location as defined with the configuration parameters.

- --- Advanced step track parametersThis page contains a number of parameter which are used to tune the tracking algorithm. Most of these parameters affect the STEP and ADAPTIVE tracking modes as well.**step size:** This parameter controls the sizes of the search steps the tracking engine performs in the STEP and ADAPTIVE modes. With the step size set to 100%, the elevation test step is one half of the antenna's 1dB bandwidth. The azimuth step size is computed the same way, but divided by *cos(elevation)* to deskew look angle to the satellite. The antenna's beamwidth (approximately) is computed from the beacon frequency and the antenna diameter. A good initial value for the step size is 60%.**settling time:** The settling time is the time the tracking engine waits after it issued an antenna pointing command until it takes the first sample of the beacon level. Specially with small steps of the antenna position, the tracking engine has no real feedback when the antenna has reached it's final position. This delay time is used to ensure that the antenna pointing command has been executes and the beacon level has settled.**step mode:** In STEP mode, this parameter controls how the tracking engine selects the direction of a search step. With the SLOW step mode, search steps alternately are made in the one or the other direction. This step mode gives good results with satellites that move only a small amount. With the FAST step mode, each search step is made in the same direction of the last successful optimization step. With any unsuccessful step the direction changes. This mode is recommended to track inclined orbit satellites, because the tracking does not perform search steps against the satellite movement in the majority of cases. In the ADAPTIVE tracking mode, the tracking engine observes the step mode selection only as long as it cannot compute a usable model from the records steptrack data. Later it uses the the modeled antenna pointing to decide where to go for a search step.**max optimization:** The step track algorithm uses the measured beacon levels before a search step and after it together with some knowledge about the shape of the antenna beam to compute the peak location. Unexpected variations of the beacon level may disturb the measurement and may result to a wrong peak evaluation. To minimize the impact of this to the antenna pointing, the tracking algorithm limits the amount of an optimization step to this value. Setting this parameter to 0% forces the tracking engine to limit the peak position to a window between the antenna positions before and after the search step. Increasing the 'max optimization' parameter widens this window at both edges with the actual search step size multiplied by the 'max optimization' value.With the ADAPTIVE tracking mode this parameter is used, too. Here however it acts as a fault threshold: If a peak is outside, is is marked as invalid and not included to the tracking model computation.**az/el search step:** These fields show the actual search step sizes as absolute angles for you orientation. The fields get updated each time you change the 'step size' parameter.**az/el window:** These parameters define a rectangular window around the nominal antenna position. The tracking engine limits the optimization movements to this window.**beacon averaging time:** This parameter defines the time, while samples read from the beacon receiver get averaged to achieve one beacon level value. Recommended values are between 15 and 30 seconds, if the beacon receiver itself performs an effective averaging/smoothing of the level reading, shorter values may be possible.**max standard deviation:** While the program averages the beacon level readings, it also computes the standard deviation of the reading as a measure of the (short term) stability of readings. If the standard deviation exceeds this limit, the measurement is discarded.**max signal drift:** Before performing a search step, the tracking algorithm requires two beacon measurement series to differ not more than the value defined here. Otherwise the beacon level is assumed to drift, the search step is postponed or canceled.**max drop in search step:** If with a search step the beacon level drops more than the value set here, the antenna moves back to the previous or predicted position, no peak evaluation is done.**min beacon level:** If the measured beacon level falls below this value, a beacon level fault is risen and the measurement is marked as invalid. With the STEP tracking mode no tracking is performed until the beacon level is above the threshold again. In ADAPTIVE mode, the tracking engine tracks the satellite following the mathematical model made from the step track peaks.

- F! --- Faults and fault mask**Beacon receiver:** The beacon receiver device states a fault. This may be a hardware fault or a loss of signal. It is recommended to switch of any level threshold monitoring at the beacon receiver and let the tracking device monitor the beacon level. The tracking device applies threshold monitoring only during the measurement phases, signal dropouts during the antenna movement phases are ignored.**Beacon level:** The measured beacon level is below the threshold set at the advanced step track parameters page.**AZ/EL window limit:** The recent step track peak found is outside the window limits set at the advanced step track parameters page. The antenna has been moved to the window limit rather than to the peak position. The fault stays active until the next tracking step.**Antenna:** The antenna positioning controller states a fault. This may be a motor failure, limit switch or another specific antenna fault.**Search step size:** The tracking engine is not capable to compute a valid tracking step size from the given data. This happen if either the antenna diameter (a setup parameter) or the beacon frequency is not set properly. The fault stays active until the missing value is set.**AZ/EL Model match:** In ADAPTIVE tracking mode, a tracking step found a peak which considered to be invalid at it is too far away from the position computed by the model. The fault stays active until the next tracking step.

- ℹ --- The device info page.
- 🔧 --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the Antenna-Tracking device work properly.

parameter	***description***
config.azimuthId	Enter here the parameter ID of the azimuth target value the tracking unit shall use. Example: 'ODU.azimuth'
config.stateAzId	Enter here the parameter ID of the azimuth read out value the tracking unit shall use. Example: 'ODU.state.az'
config.elevationId	Enter here the parameter ID of the elevation target value the tracking unit shall use. Example: 'ODU.elevation'
config.stateElId	Enter here the parameter ID of the elevation read out value the tracking unit shall use. Example: 'ODU.state.el'
config.polarizationId	Enter here the parameter ID of the polarization offset target value the tracking unit shall use. The tracking unit does no polarization, however a fixed polarization offset value is stored together with each target. Example: 'ODU.polarization'
config.antennaFaultId	Enter here the parameter ID of the antenna fault, the tracking unit shall monitor.

parameter	***description***
config.bcrxFreqId	Enter here the parameter ID of the beacon receiver frequency setting. The tracking unit sends the receive frequency stored with a target to this parameter (RF frequency, not L-band). Leave this field empty, if the beacon receiver frequency cannot be controlled through the sat-nms software. Please note, that in the tracking unit the beacon frequency must be set to the correct value even if it is not passed to the beacon receiver. This is because the tracing unit computes the step track search sizes from the beacon frequency.
config.bcrxLevelId	Enter here the parameter ID of the beacon receiver level reading. The value is expected to be a receive level in dBm or a C/N value [dB]. To read the analog output value of a tracking receiver, you need an external device which reads this voltage and converts the readint to dBm. The sat-nms Antenna Control Unit ODM is capable to do this.
config.bcrxFTrackId	If you are using a sat-nms L-band beacon receiver together with this tracking device, enter 'BCRX.ftrack.fast' here, replace BCRX by the device name of the beacon receiver. This tells the tracking engine to perform a synchronous frequency tracking step on the beacon receiver before each antenna tracking step.
config.bcrxFaultId	Enter here the parameter ID of the beacon receiver fault, the tracking unit shall monitor.
config.xDiameter	This parameter tells the tracking device the width of the antenna dish. The tracking device computes the azimuth tracking step size from this parameter and the beacon frequency value.
config.yDiameter	This parameter tells the tracking device the height of the antenna dish. The tracking device computes the elevation tracking step size from this parameter and the beacon frequency value.
longitude	The antenna's longitude (degree east). Provide this value with an accuracy of 0.01° to ensure a precise orbit to pointing calculation
latitude	The antenna's latitude (degree north). Provide this value with an accuracy of 0.01° to ensure a precise orbit to pointing calculation
altitude	The antenna's altitude over sea. Provide this value with an accuracy of 10 meters to ensure a precise orbit to pointing calculation

Built in analysis tools

The Antenna-Tracking device is capable to show two different types of diagrams which support you to monitor and optimize the tracking behaviour.

The first diagram tool is the X/Y display of the antenna movements. This display works completely autoscaled, it shows one green dot for each antenna position the tracking memorizes (usually the recent 48 hours). The actual antenna pointing is marked with a slightly larger yellow dot. After at least 24 hours of continuous operation, the antenna position dots give the curve of the daily movement of the satellite, from the antenna's angle of look. With a mouse click to the diagram area, the display cas be switched into a Y/T diagram showing the recent 48 hours. This condensed display of the azimuth, elevation and beacon level variations over 48 hours is meant as an overview display. A second mouse click switches back to the previous view.

The separate Y/T display window gives more detailed information about what the tracking engine did during the recorded 48 hours. It provides variable scales, and permits to display additional information. The display shows either the azimuth or the elevation pointing found at the beginning of each each tracking step together with the beacon level measured at this time. The following additional information may be included to the diagram:

- ⊥? --- If this toolbar button is pressed, the diagram shows the search step and the beacon level change resulting from this as a dimmed peak at the left edge of each data point. You can easily monitor the signal degradation which is caused temporarily by the tracking search steps with this function enabled.
- ⊥! --- If this toolbar button is pressed, the diagram includes the evaluated peak positions. In STEP track mode, this actually is the location where the antenna gets moved to. In ADAPTIVE tracking mode, the antenna moves to the location resulting from the mathematical model which is computed from the step track peaks. In the latter case, the peak positions may be interpreted as the actual difference between the modeled pointing and the peak position.

Recommended beacon receiver settings

Below some recommended settings for the sat-nms L-band beacon receiver for the use with the tracking device are listed. These settings have proved to work satisfactory in most cases. If you are using a third party beacon receiver, use analogous values a starting point.

- Measurement bandwidth --- 30kHz
- Post detector filter --- 0.5 Hz
- Frequency tracking --- OFF (the tracking device switches this ON/OFF automatically with a sat-nms L-band beacon receiver)

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

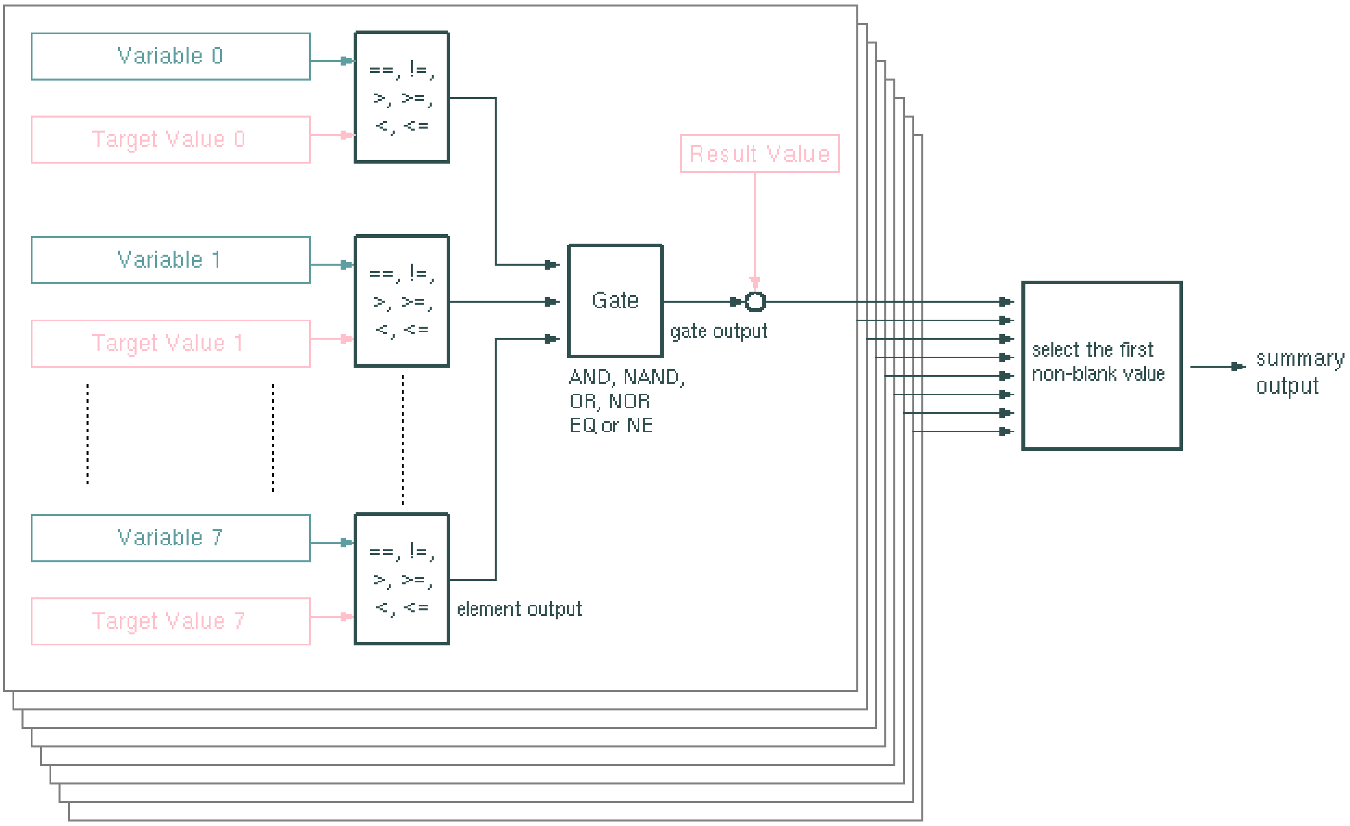
name	***type***	***flags***	***range***
info.frame	TEXT	R/O	StringRange R/O
config.azimuthId	TEXT	SAVE SETUP	StringRange
config.stateAzId	TEXT	SAVE SETUP	StringRange
config.elevationId	TEXT	SAVE SETUP	StringRange
config.stateElId	TEXT	SAVE SETUP	StringRange
config.polarizationId	TEXT	SAVE SETUP	StringRange
config.antennaFaultId	TEXT	SAVE SETUP	StringRange
config.bcrxFreqId	TEXT	SAVE SETUP	StringRange
config.bcrxLevelId	TEXT	SAVE SETUP	StringRange
config.bcrxPolId	TEXT	SAVE SETUP	StringRange
config.bcrxInhibitId	TEXT	SAVE SETUP	StringRange
config.bcrxFaultId	TEXT	SAVE SETUP	StringRange
config.xDiameter	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.yDiameter	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.longitude	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.latitude	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.altitude	FLOAT	SAVE SETUP	DoubleRange (0 .. 0)

name	***type***	***flags***	***range***
azimuth	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
elevation	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
polarization	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
trackMode	CHOICE	SAVE	EnumRange (OFF STEP ADAPTIVE PROGRAM)
polPlane	CHOICE		EnumRange DIS (H V)
frequency	FLOAT		DoubleRange (0.000 .. 0.000)
interval	INTEGER	SAVE	IntegerRange (0 .. 60)
name	TEXT	SAVE	StringRange
elements	TEXT	SAVE	StringRange
orbit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
stepSize	INTEGER	SAVE	IntegerRange (0 .. 200)
stepMode	CHOICE	SAVE	EnumRange (FAST SLOW)
optimizeSize	INTEGER	SAVE	IntegerRange (0 .. 200)
averagingTime	INTEGER	SAVE	IntegerRange (1 .. 120)
settlingTime	INTEGER	SAVE	IntegerRange (1 .. 120)
dazMax	FLOAT	SAVE	DoubleRange (0.0 .. 10.0)
delMax	FLOAT	SAVE	DoubleRange (0.0 .. 10.0)
driftMax	FLOAT	SAVE	DoubleRange (0.00 .. 10.00)
dropMax	FLOAT	SAVE	DoubleRange (0.00 .. 10.00)
sdevMax	FLOAT	SAVE	DoubleRange (0.00 .. 10.00)
beaconThreshold	FLOAT	SAVE	DoubleRange (-100.0 .. 0.0)
state.saz	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.sel	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.daz	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.del	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
eval.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	***type***	***flags***	***range***
eval.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
eval.pl	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
state.mode3	TEXT	R/O	StringRange R/O
state.mode4	TEXT	R/O	StringRange R/O
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.elements	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
evalByOrbit	TEXT	NOPRESET	StringRange
evalByElements	TEXT	NOPRESET	StringRange
moveEvaluated	TEXT	NOPRESET	StringRange
slist.select	CHOICE	NOPRESET	EnumRange ()
slist.action	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Beacon receiver)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Beacon level)
faults.03	ALARM	R/O	AlarmFlagRange R/O (AZ window limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (EL window limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Antenna)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Search step size)
faults.07	ALARM	R/O	AlarmFlagRange R/O (AZ Model match)
faults.08	ALARM	R/O	AlarmFlagRange R/O (EL Model match)

6.14 Gate-Array

The Gate-Array and Gate-Array-20 devices provide a set of 8 logical gates with 8 inputs each (20 with the Gate-Array-20) which may be used to merge status variables in a user programmed way. Gate-Array and Gate-Array-20 behave exactly the same way, the only difference is the number of inputs of each gate. Each gate may be configured to work as a AND, NAND, OR, NOR, NE (XOR) or EQ gate. The inputs of a gate are so called elements, which compare the contents of a sat-nms parameter to a fixed target value.



Elements

The Gate-Array altogether provides 64 'elements' (160 with the Gate-Array-20) which compare a parameter value to a fixed target value. Each element of a Gate-Array may be used / programmed individually.

The output of an element is a read-only text-type variable which either reads 'true' or 'false'. This output may may be monitored by other devices, the parameter name is ' *<device-name>.<gate-no.>.<element-no.>.output* '. Counting the gate or element numbers starts at zero, the name of the second element of the first gate e.g. is ' *<device-name>.0.1.output* '.

The parameter 'message id' defines the full name of the parameter the element shall monitor. The 'target' value is the constant value the parameter is compared against. The comparison operator is selected from a choice box:

- EQ --- 'equal' The output of the element becomes 'true', if the monitored parameter is *textual* equal to the target value. If you compare numeric values with EQ, you have to consider the textual representation of a parameter: '1.0' is not equal to '1.00'! The monitored parameter is read like it would appear in the user interface, but without a unit string.
- NE --- 'not equal' The inverted form of EQ.
- GT --- 'greater than' The output of the element becomes 'true', if the monitored parameter is *numerically* greater than the target value. If any of the compared values cannot be successfully parsed as a floating point number, the value is interpreted as zero.

- GE --- 'greater or equal' The output of the element becomes 'true', if the monitored parameter is *numerically* greater or equal the target value. If any of the compared values cannot be successfully parsed as a floating point number, the value is interpreted as zero.
- LT --- 'less than' The output of the element becomes 'true', if the monitored parameter is *numerically* less than the target value. If any of the compared values cannot be successfully parsed as a floating point number, the value is interpreted as zero.
- LE --- 'less or equal' The output of the element becomes 'true', if the monitored parameter is *numerically* less or equal the target value. If any of the compared values cannot be successfully parsed as a floating point number, the value is interpreted as zero.

Elements which have no or no valid message ID set, are not taken into account in the downstream gate. The same applies to elements which indeed have a valid message ID set, but the message source did not supply a value since system start.

Gates

Every 8/20 elements are connected to one gate (There are 8 gates in total). Each gated can be set to perform one of the logical functions AND, NAND, OR, NOR, EQ and NE. A gate likewise provides an output variable, named ' *<device-name>.<gate-no.>.output* ' (' *<device-name>.0.output* ' designates the output of the first gate). A output either reads 'true' or 'false' depending on the result of the logical function it performs. For the logical function only these inputs contribute which really are in use. The outputs of elements which have no message identifier set are not taken into account. The available logical functions for the gate are:

- AND --- Output is *true* if all active input comparisons are true.
- NAND --- Output is *false* if all active input comparisons are true.
- OR --- Output is *true* if at least one of the active input comparisons is true.
- NOR --- Output is *false* if at least one of the active input comparisons is true.
- EQ --- Output is *true* if all active input comparisons result to the same logical state
- NE --- Output is *false* if all active input comparisons result to the same logical state

Beside this, each gate provides a parameter called 'result'. This parameter is an arbitrary text value which may become the summary output value of the gate array under certain circumstances.

The Summary Output

The Gate-Array provides a summary output variable named ' *<device-name>.output* '. This summary output carries the value of the 'result' parameter of the first gate which has a 'true' output. If no gate has a 'true' output, the Gate-Array's master output reads an empty string. Example: If the gates 4 and 6 in the Gate-Array are 'true', the summary output is set to the value of the 'result' variable of gate 4.

Individual Logical Gate Outputs

The Gate-Array provides variables which report the local state of each comparison element, each gate and additionally a summary output which combines the outputs of all gates with a logical OR. For compatibility reasons these variables all are read-only TEXT variables reporting either 'true' or 'false'.

On element level (the result of each individual comparison) the variables are named '*<device-name>.<gate-no.>.<element-no.>.output*'. Counting the gate or element numbers starts at zero, the name of the second element of the first gate e.g. is ' *<device-name>.0.1.output* '.

On gate level (the result of each of the eight gates) the variables are named '*<device-name>.<gate-no.>.output*'. Counting the gate numbers starts at zero.

Beside the textual summary output described above, the Gate-Array provides a second summary output called ' *<device-name>.ored* '. This output reports 'true' if at least one of the gates reports 'true', 'false' otherwise.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- 0 .. 7 --- The configuration and status pages each contain the parameters for one gate and the eight elements which are connected to this gate. The actual state of the element outputs and the gate output is signalled by the color of the arrows which show the logical signal flow. Gray arrows signal a 'false' or inactive state, green arrows signal 'true' state.At the bottom of each page the actual summary (master) output of the Gate-Array and the output states of all eight gates ist shown.
- ℹ --- The device info page.
- 🔧 --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

Beside the standard debug settings 'verbose' and 'logParameterChanges' there is one configuration parameter available for the Gate-Array logical device.

parameter	***description***
outputDefault	If this parameter is set to a non-blanc value, the Gate-Arrays 'output' shows this value instead of an empty string if none of the gates results 'true'.

Remarks

- The Gate-Array is particularly suitable to generate color selections for the user interface from complex status information in the M&C. For example, switch settings, rf-on states and fault signals may be combined to the information if a certain signal path is carrying signal, switched off or being faulty. The summary output of the Gate-Array may be monitored by the line elements in the user interface which symbolize this signal path. They will change their color following the detected state.
- The gates of a Gate-Array may be cascaded by referencing the output of one gate as the message ID of another's gate input element. Also 'feedback' routings are possible. If you couple element inputs to an output of the same gate array, you should consider, that a

gate's output changes with up to one second delay after the gate's input changed. If you cascade Gates, the delay times are accumulating. If you feed-back signals, oscillation due to the delay may be the result.

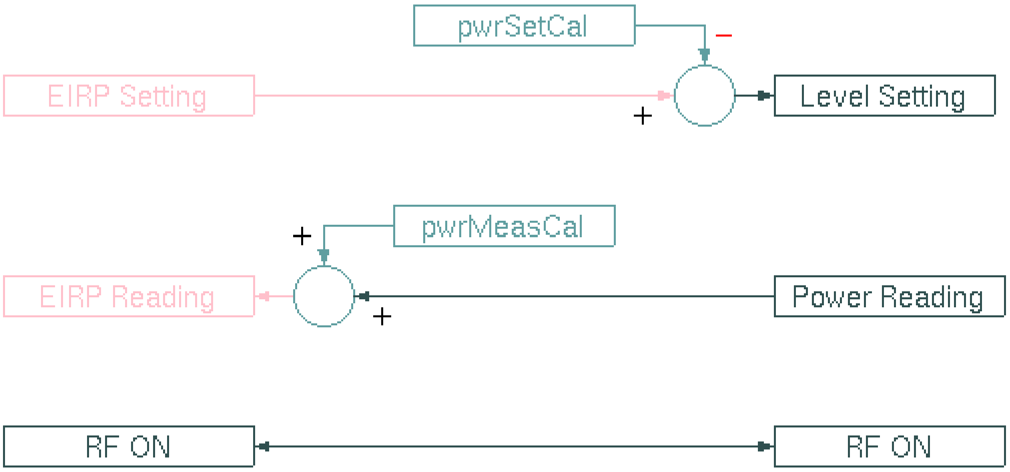
- The variables used to monitor and control a Gate-Array are created dynamically at runtime. The neither appear in the device driver file nor do the in the variable list below. For the same reason it is not possible to map the output variables of a Gate-Array into another M&C via a Proxy-Device.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.outputDefault	TEXT	SAVE SETUP	StringRange

6.15 Level-Set

The Level-Set logical device provides a simple mechanism to set (and monitor) transmit power values in EIRP. It is intended to be used with small VSAT stations using a SSPA at a fixed frequency. The Level-Set device uses fixed calibrations offsets or optionally calibration tables.



The diagram above illustrates the function of the Level-Set device. The device provides two calibrations offsets. One is used to translate an EIRP setting into a gain / output power setting to be applied at a device. The other offset translates a power reading into an EIRP value which is reported to the user. The RF-ON switch simply is routed through the device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- ⌨ --- This page shows the operational parameters.
- ℹ --- The device info page.
- 🔧 --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Level-Set device to the devices it monitors and controls.

parameter	***description***
rfOnId	The message ID controlling the RF ON state, e.g. **KSTAR.tx.on**
pwrSetId	The message ID for the drive setting.
pwrMeasId	The message ID for the output power reading.
pwrSetCal	The calibration offset to be applied at the drive setting. The offset is defined in a way that the drive setting is calculated as:
drive-setting = EIRP(dBW) - pwrSetCal .	
The Level-Set device optionally may use calibration tables (see below), the pwrSetCal value then gets recalculated with each change of the transmit frequency or the signal path.	
pwrMeasCal	The calibration offset to be applied at the power reading. The offset is defined that the displayed EIRP is calculated as:
EIRP(dBW) = power-reading + pwrMeasCal .	

parameter	***description***
The Level-Set device optionally may use calibration tables (see below), the pwrSetCal value then gets recalculated with each change of the transmit frequency or the signal path.	
txFrequencyId	The message ID of the transmit frequency, usually the upconverter tx.frequency setting. The Level-Set device listens to this parameter to perform the amplifier output power vs. EIRP calculations based on the actual frequency.
txSwitch1Id	The message ID of the first of two switches in the transmit signal path which affect the EIRP calibration. Usually here the ID of the polarization switch is entered.
txSwitch2Id	The message ID of a second switch in the transmit signal path.
fileName	The base name of the files containing the calibration tables (see below). The name is without suffix.
maxDisplay	The EIRP value to be displayed at the right edge of a GaugeElement which uses the 'gauge' status variable provided by this device. Set maxDisplay and minDisplay both to 0.0 in order to set the gauge's display range to the complete valid range of the eirp setting.
minDisplay	The EIRP value to be displayed at the left edge of a GaugeElement which uses the 'gauge' status variable provided by this device. Set maxDisplay and minDisplay both to 0.0 in order to set the gauge's display range to the complete valid range of the eirp setting.

EIRP Calibration Tables

The Level-Set device may use fixed calibration values to convert the measures amplifier power to EIRP and to convert the desired EIRP to a gain or output power setting of an arbitrary device. For applications which require a more precise power setting/display, the Level-Set device alternatively may use calibration tables. With these tables, the calibration offsets are calculated frequency dependent for up to four signal paths.

The calibration tables are plain text files, called "XXXX.eirpCal" and "XXXX.driveCal" where XXXX is the 'fileName' configuration setting. The files must reside in the M&C software's home directory. If the files exist and the Level-Set device receives a valid frequency setting, the calibration values calculated from the tables replace the pwrSetCal and pwrMeasCal values in the configuration setup.

Below an example for such a file is shown:

```
#
# /home/mnc/EIRP-1.eirpCal
#
#          TRANSMIT 1    TRANSMIT-2
#          X      Y      X      Y
  14000.0  20.2   20.2   20.2   20.2
  14100.0  20.27  20.27  20.27  20.27
  14200.0  20.31  20.31  20.31  20.31
  14300.0  20.44  20.44  20.44  20.44
  14400.0  20.5   20.5   20.5   20.5
  14500.0  20.5   20.5   20.5   20.5
```

The file is formatted as a table of 5 columns and an arbitrary number of lines. Empty lines and lines starting with a "#" are ignored. Columns are separated by spaces.

column	***description***
1	The frequency value in MHz. The frequency values must be sorted in the order low values first.
2	The calibration value for the signal path indicated by SWITCH1=A (OFF), SWITCH2=A (OFF).
3	The calibration value for the signal path indicated by SWITCH1=B (ON), SWITCH2=A(OFF).
4	The calibration value for the signal path indicated by SWITCH1=A (OFF), SWITCH2=B (ON).
5	The calibration value for the signal path indicated by SWITCH1=B (ON), SWITCH2=B (ON).

The calibration values in the '*.eirpCal' table are defined the same way as the pwrMeasCal setup value, the values in the '*.driveCal' table are defined the same way as the pwrSetCal value.

Dynamic Change Of Calibration Settings

If the transmit frequency or the signal path changes, the Level-Set device keeps the drive setting constant and adjusts the diyplayed EIRP value according to the changed calibarion offsets.

The device provides a variable called 'adjustDrive'. If an arbitrary value gets written to this variable, the device adjusts the drive setting in order to meet the previously set EIRP value.

Limit Checking

At the primary device page, two pairs of min / max threshold values may be defined. These values apply to the measured EIRP, they are checked while the Level-Set device is set RF-ON. If the measured EIRP exceeds the limit values a fault is generated. The fault gets cleared if the EIRP value returns inside the limit window of id the EIRP is switched off. If the Level-Set device shall not do any limit checking, set both threshold values to 0.0.

The minEirp/maxEirp settings are included into the 'gauge' parameter which controls the appearance of a Gauge screen element which follows this parameter. There is a second parameter of the same format called 'gauge2' which includes the minEirp2/maxEirp2 limits instead.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.state	TEXT	R/O	StringRange R/O
config.rfOnId	TEXT	SAVE SETUP	StringRange
config.pwrSetId	TEXT	SAVE SETUP	StringRange
config.pwrMeasId	TEXT	SAVE SETUP	StringRange
config.txFrequencyId	TEXT	SAVE SETUP	StringRange
config.txSwitch1Id	TEXT	SAVE SETUP	StringRange
config.txSwitch2Id	TEXT	SAVE SETUP	StringRange
config.pwrSetCal	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)

name	***type***	***flags***	***range***
config.pwrMeasCal	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.fileName	TEXT	SAVE SETUP	StringRange
config.maxDisplay	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.minDisplay	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
eirp	FLOAT		DoubleRange (0.00 .. 0.00)
on	CHOICE		EnumRange (OFF ON)
minEirp	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
maxEirp	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
minEirp2	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
maxEirp2	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
adjustDrive	TEXT	NOPRESET	StringRange
measuredEirp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
gauge	TEXT	R/O	StringRange R/O
gauge2	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (EIRP low threshold)
faults.02	ALARM	R/O	AlarmFlagRange R/O (EIRP high threshold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EIRP low threshold 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (EIRP high threshold 2)