nVent SCHROFF
Guardian Management Gateway

User Manual

Release 63998-20558

29.11.2021

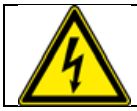
Doc-No.: 63972-383

Schroff GmbH
Langenalber Str. 96-100
75334 Straubenhardt/Germany
Schroff.nVent.com

Table of contents

1 Safety

	**Read hardware manual and quick start guides** The nVent SCHROFF Guardian Management Gateway is intended to be installed and maintained by qualified and trained personnel in compliance with local and national electrical codes and safety regulations. The hardware description and the corresponding safety instructions are not scope of this manual. Before initial operation, read the resp. manuals.

	Before using the devices, check connectors and electrical cables. All connectors and cables must be designed and rated in accordance with the technical data.

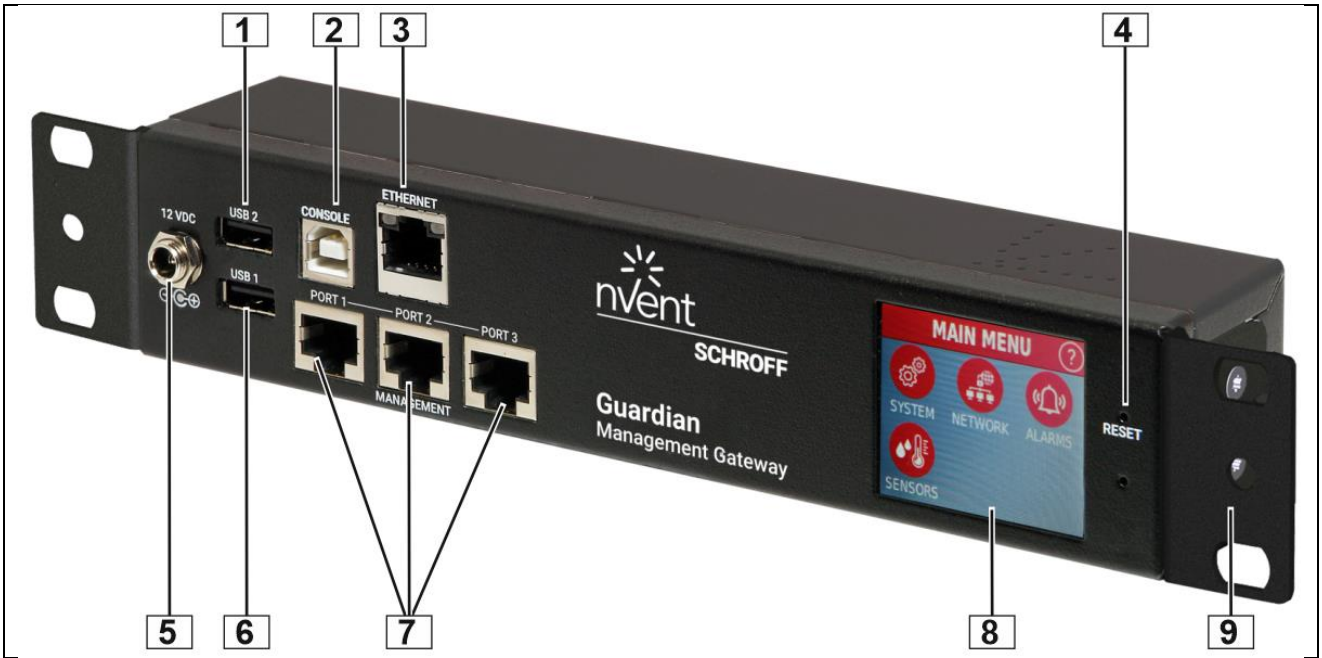
1.1 Intended Use

The nVent SCHROFF Smart Gateway Platform (SGP) is an environmental monitoring platform designed to sense, track, store and alarm health and security parameters in an IT-datacenter infrastructure.
The heart of the platform is a compact control unit with just 1U in height/depth and 250 mm in width, it can be installed as 19” unit or into any available space in a data center rack.
The Guardian Management Gateway is based on the nVent SCHROFF Smart Gateway Platform (SGP) and offers three sensor management ports with each port being able to monitor up to 16 sensor devices with a total cable length of 40 meters per port, allowing a single Guardian Management Gateway unit to monitor multiple racks or complete rack aisles.
Besides monitoring physical parameters like temperature, humidity, smoke, door status or water intrusion, the Guardian Management Gateway can also monitor Schroff RackChiller and In-Row Coolers – with an easy plug and play installation.
Set-up of the Guardian Management Gateway with security features, sensor configuration, user management, alarm and log management can be easily done through a built in Web Interface.
Main access to the Guardian Management Gateway is through the 1 GbE Network interface, supporting industry standard protocols like SNMP, SMTP, HTTPS, BACnet, Modbus/TCP and HPI.

Features:

- Data Center environmental monitoring platform
- Compact Design, fits anywhere in a data center rack
- Auto orientation LCD Touch Display
- Web browser GUI or Command Line Interface (CLI) for setup and maintenance.
- Three management ports to connect external sensors and Modbus devices
- Up to 16 sensors/Modbus devices per management port with a cable length of 40 m
- Supports Industry standard network protocols (HTTPS, SNMP, SMTP, Modbus/TCP)
- BACnet support
- IoT support
- RedFish support

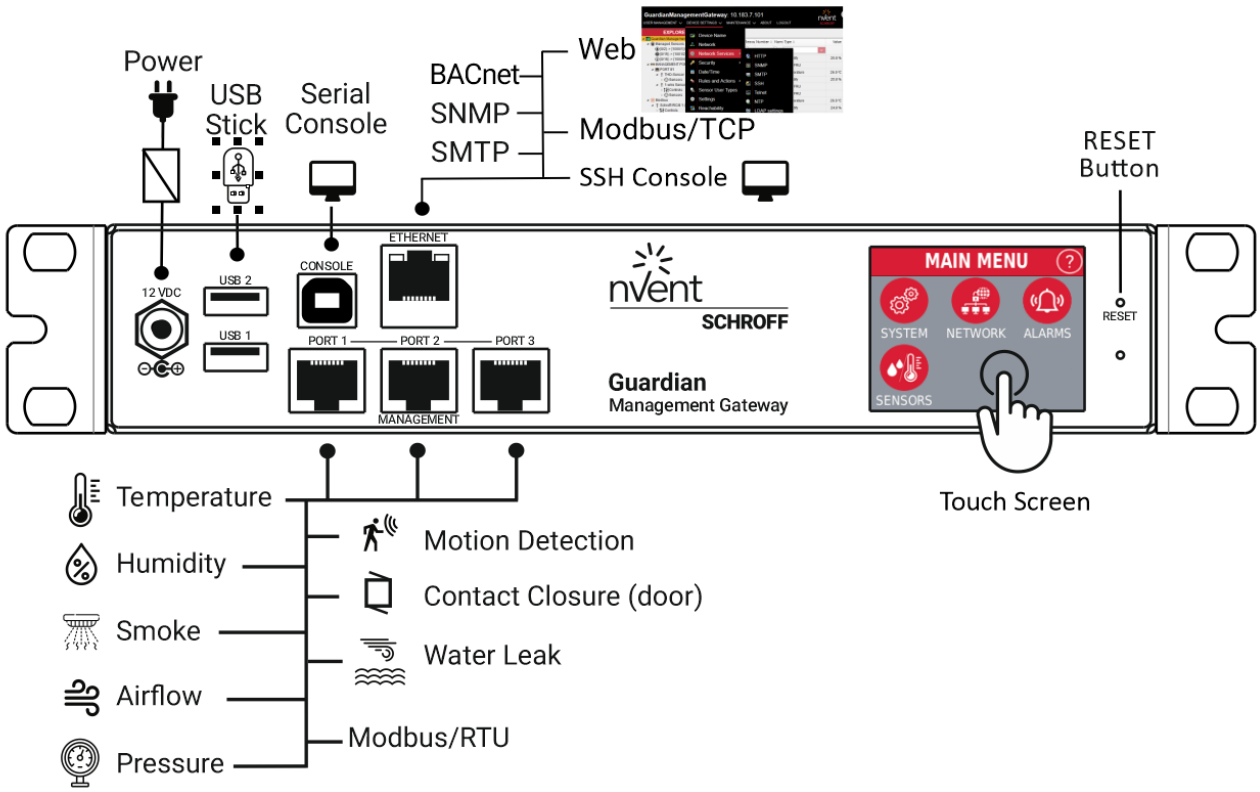
2 Product Overview Guardian Management Gateway



1	USB Type A Interface 2	6	USB Type A Interface 1
2	USB Type B interface (Console)	7	Sensor Ports
3	Ethernet Interface	8	Touch Display
4	RESET Button	9	Mounting Brackets
5	12 VDC Power Input		

2.1 Guardian Management Gateway Interfaces

--



The Guardian Management Gateway provides the following interfaces and connectors:

- 1 USB Type-B interface (Console)
- 3 sensor and Modbus device interfaces (RJ45)
- 2 USB Type-A interfaces (USB 1 and USB 2)
- 1 Ethernet interface (RJ45)
- 1 LCD touch display
- 1 Power Barrel Connector 2.1/5.5 mm, female

To support firmware upgrades, importing/exporting configuration files and exporting log files via a USB Flash drive, the device must be inserted into the USB 2 port. These features are not supported on the USB 1 port.

3 Installing and configuring

	This manual describes how to operate and configure the Guardian Management Gateway via the web interface. Advanced users can operate and configure the Guardian Management Gateway using a terminal program via the command line interface. A User Manual with the Command Line commands is available on request under order number 63972-385.

	For rack mounting and first steps, see the quick start guide, order no.: 63972-380

3.1 Connect to Network

3.1.1 Wired Connection to LAN

To make a wired connection insert a network cable with RJ45 connector into the socket labelled “Ethernet” and connect the other end to your network device. Once you have a wired connection, you can use the Command Line Interface (CLI) or the Web Interface to access the Guardian Management Gateway.

3.1.2 Serial Interface via USB Type B Connector

To use a Command Line Interface (CLI) via the serial interface, connect your computer to the USB Type B connector labelled “CONSOLE”.

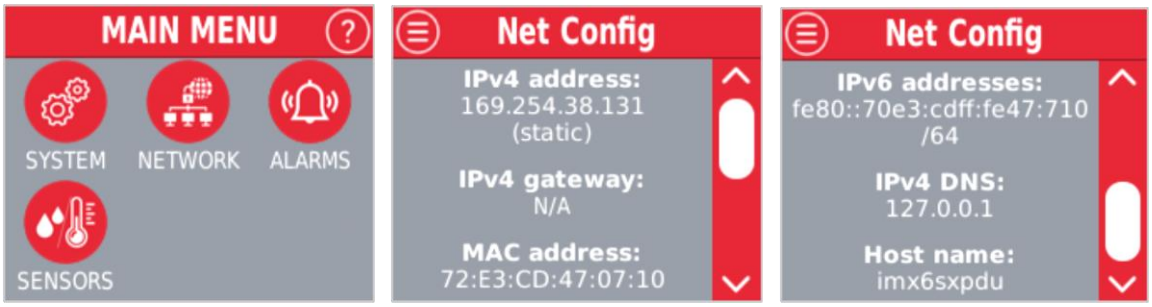
4 Setting up Guardian Management Gateway

After the Guardian Management Gateway is installed and connected to the network, a user can use Command Line Interface (CLI) or Web interface to connect and start communicating with it. For that, the user should know the IP address of the Guardian Management Gateway.

	By default, the Guardian Management Gateway is configured to obtain its IPv4 address from a DHCP server.

A DHCP server can be configured either to give the IP address to the Guardian Management Gateway from a dynamic pool (in which case it is not known in advance) or to assign a static IP address based on the MAC address of the Guardian Management Gateway.

In any case, the assigned IP address can be seen by pressing the NETWORK button on the LCD screen of the Guardian Management Gateway.



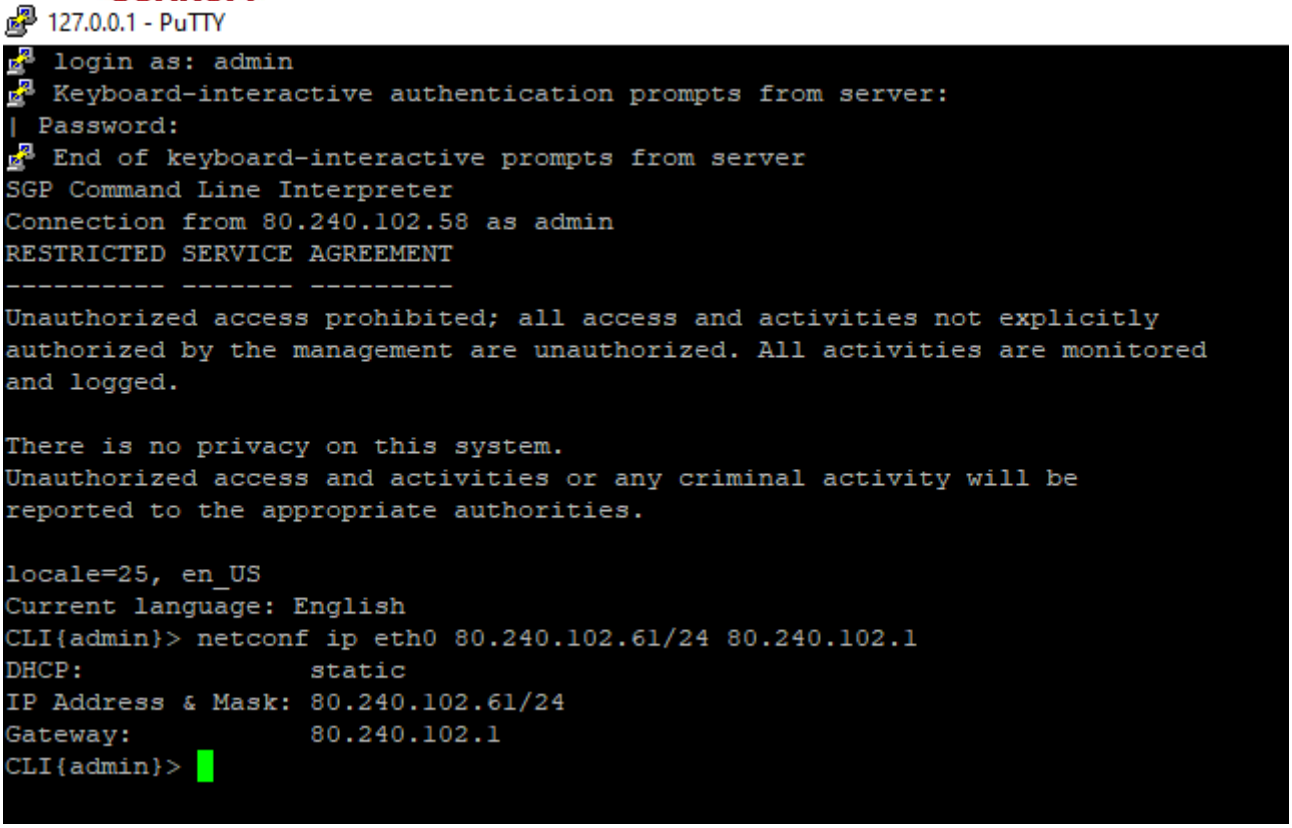
Assign static IP address

To assign a static IP address, complete the following steps:

- Connect your computer to the USB-B port labelled “CONSOLE” with an USB-A/USB-B cable.
- Determine COM port assigned by your computer to the USB Serial connection (Control Panel → System → Hardware → Device Manager → Ports > USB Serial Port).
- Open a terminal program (e.g. PuTTY), set Serial line to the assigned COM port (e.g. COM3), the Speed to 115200, and the Connection type to Serial
- Log in as “admin”
- Password: “admin”
- Assign the IPv4 configuration attributes for the network interface with CLI by entering the following command:
  netconf ip <interface> <ip_address>/<mask> [<ipv4_gateway>]
  - *<interface>* is the name of the adapter (eth0)
  - *<ip_address>* is the IPv4 address assigned to the interface, in the decimal-dot notation
  - *<mask>* is the subnet mask as the number of significant bits; the address with mask may look like *10.183.7.110/24*
  - *<ipv4_gateway>* is the default gateway address in the decimal-dot notation, it is optional here.

  Example:

  netconf ip eth0 10.183.7.101/24 10.183.7.249

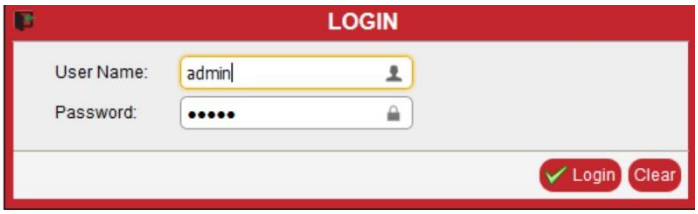


	To communicate with the Guardian Management Gateway using CLI or Web interface, the user should know a user name and the corresponding password. By default, three users are created: **admin** (password “admin”) with administrative privileges, **user** (password “user”) with normal user privileges, **guest** (password “guest”) with low privileges. The **user** and **guest** accounts are disabled by default. Additional configuration of users can be done after logging in as **admin**. For security reasons, the password for the **admin**, **user** and **guest** user accounts request password change at the first successful login.

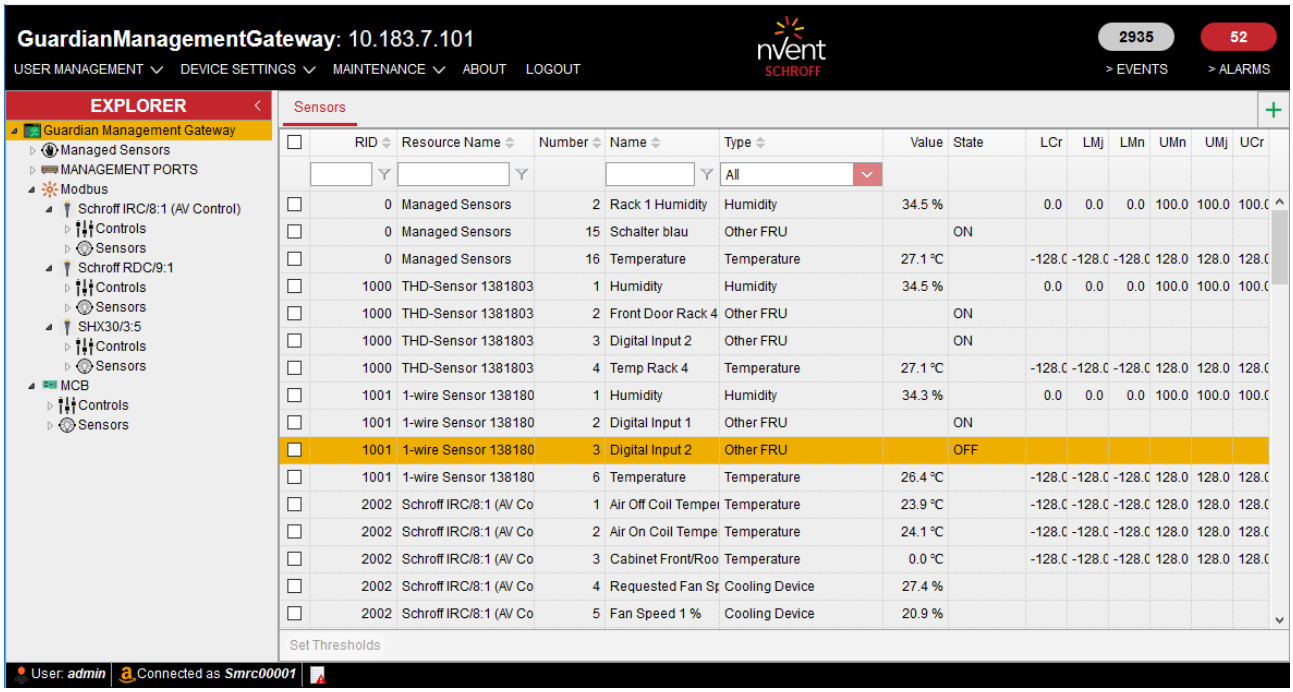
5 Getting Started

5.1 Log in using the Web interface

To log in to the Guardian Management Gateway using the Web interface, open the Web browser and point it to the Guardian Management Gateway IP address. The login dialog box appears:



After entering the user name and password, and pressing the “Login” button, the main Web interface screen appears:



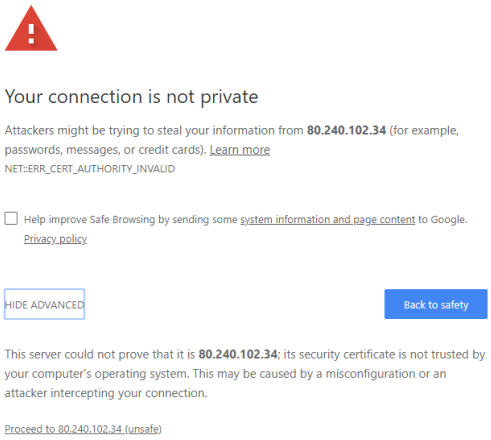
5.2 Change Password

To change the password for the current user, invoke the menu command “USER MANAGEMENT” -> “Change Password”. The “CHANGE PASSWORD” dialog appears, in which the user should type the old password and the new password (two times):

The password will be changed.

5.3 HTTPS Connection

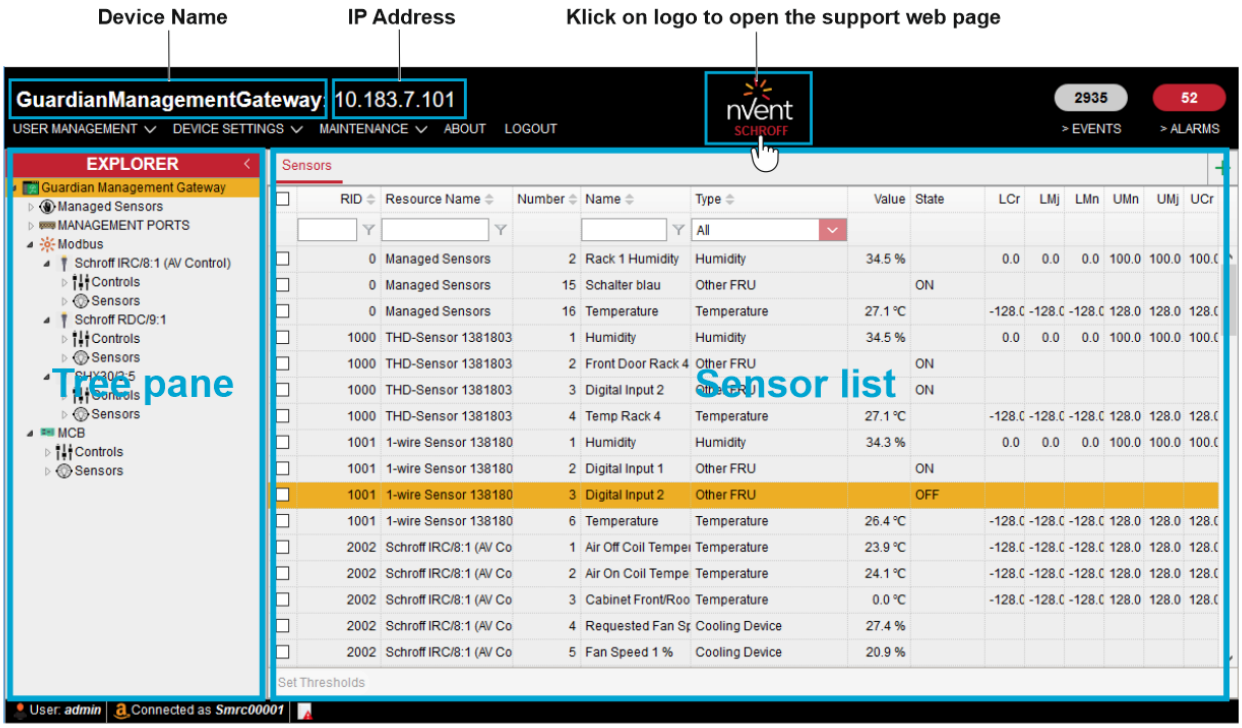
It’s possible to establish a secure HTTP (HTTPS) connection to the Guardian Management Gateway. However, the certificate that is originally installed on the Guardian Management Gateway is self-signed, and a warning like this is issued when the connection is established:



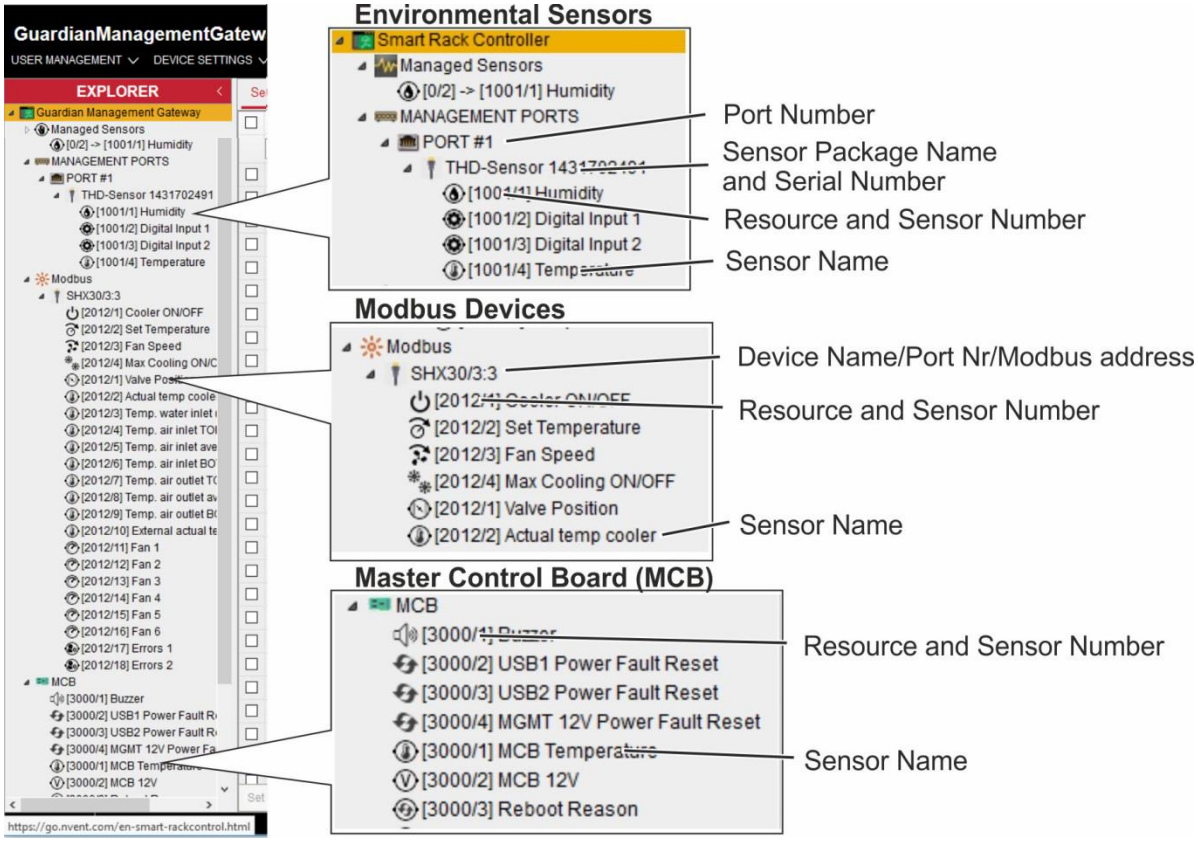
To get rid of this warning, it is necessary to install a properly signed SSL certificate on the Guardian Management Gateway; it is the user’s responsibility to obtain such a certificate.

6 Web Interface GUI

6.1 Overview

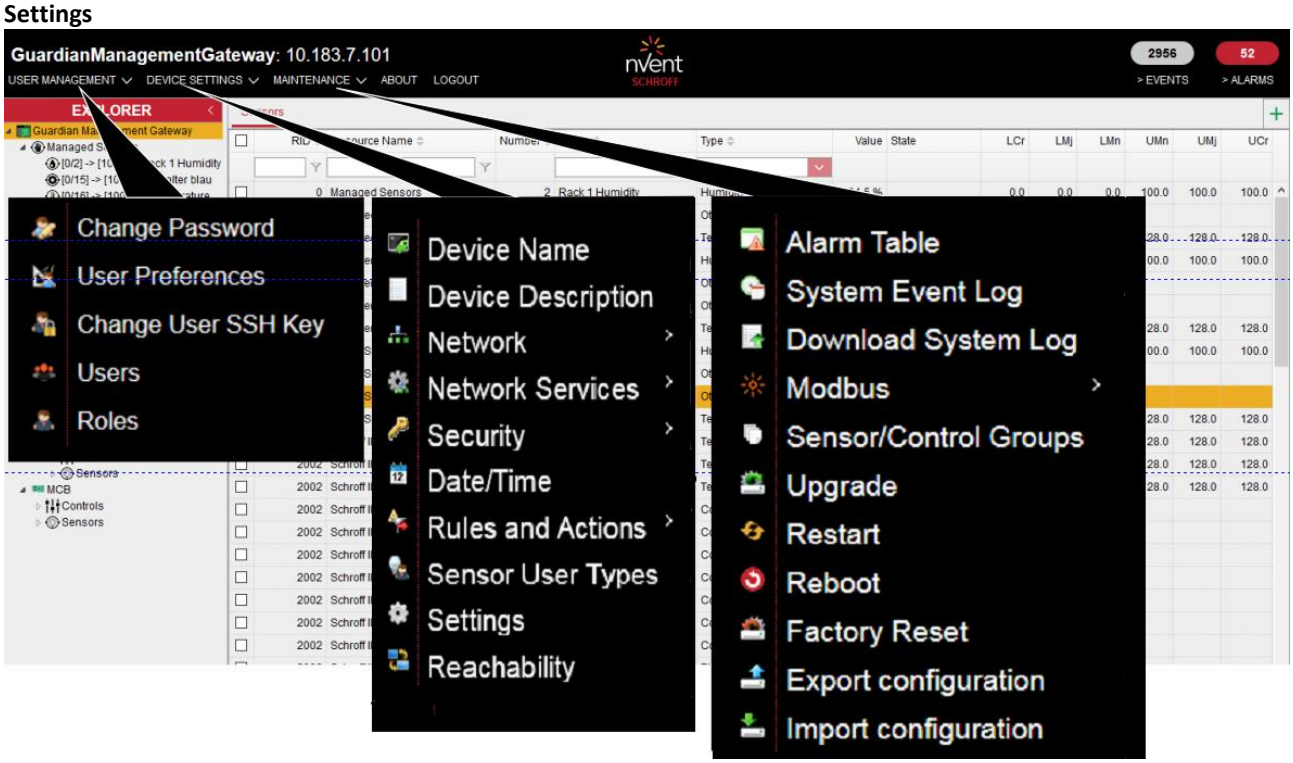


6.1.1 **Overview tree pane**



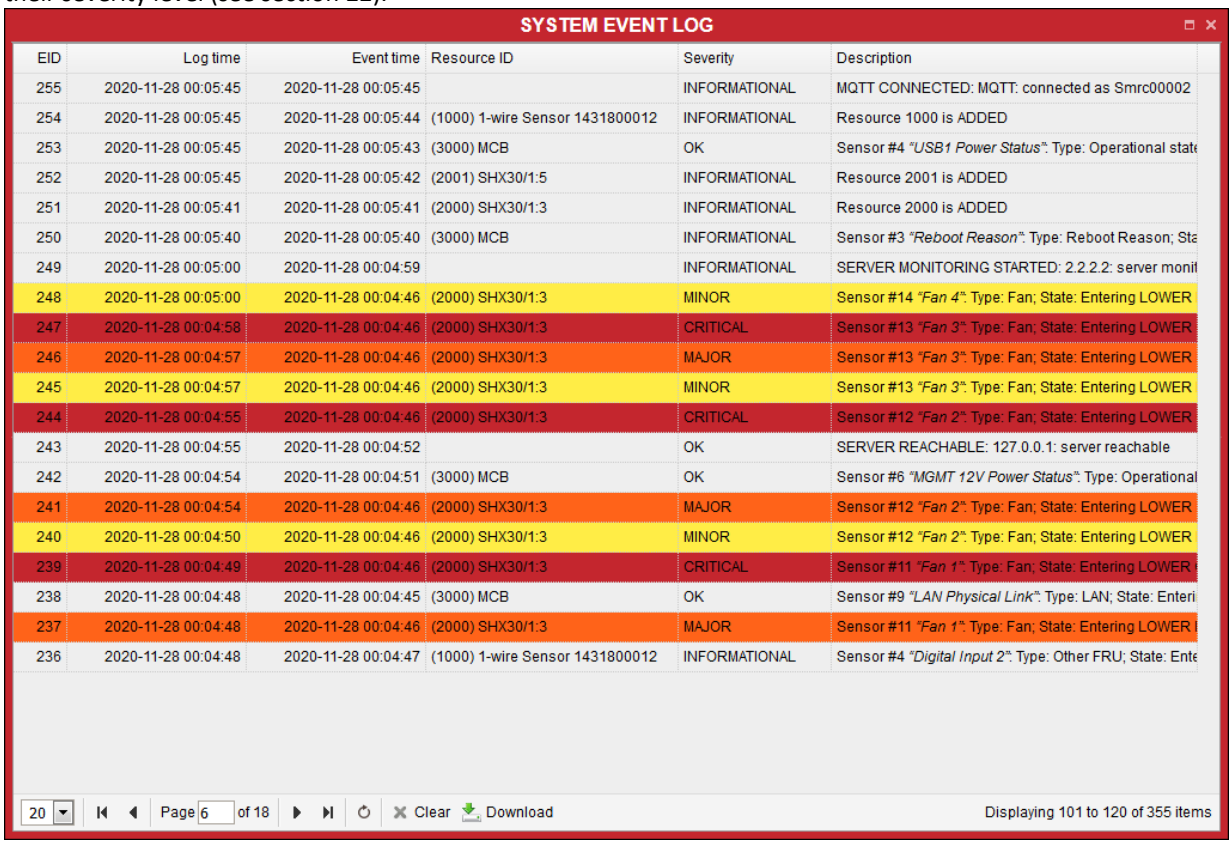
6.1.2 Overview drop down menu

Settings

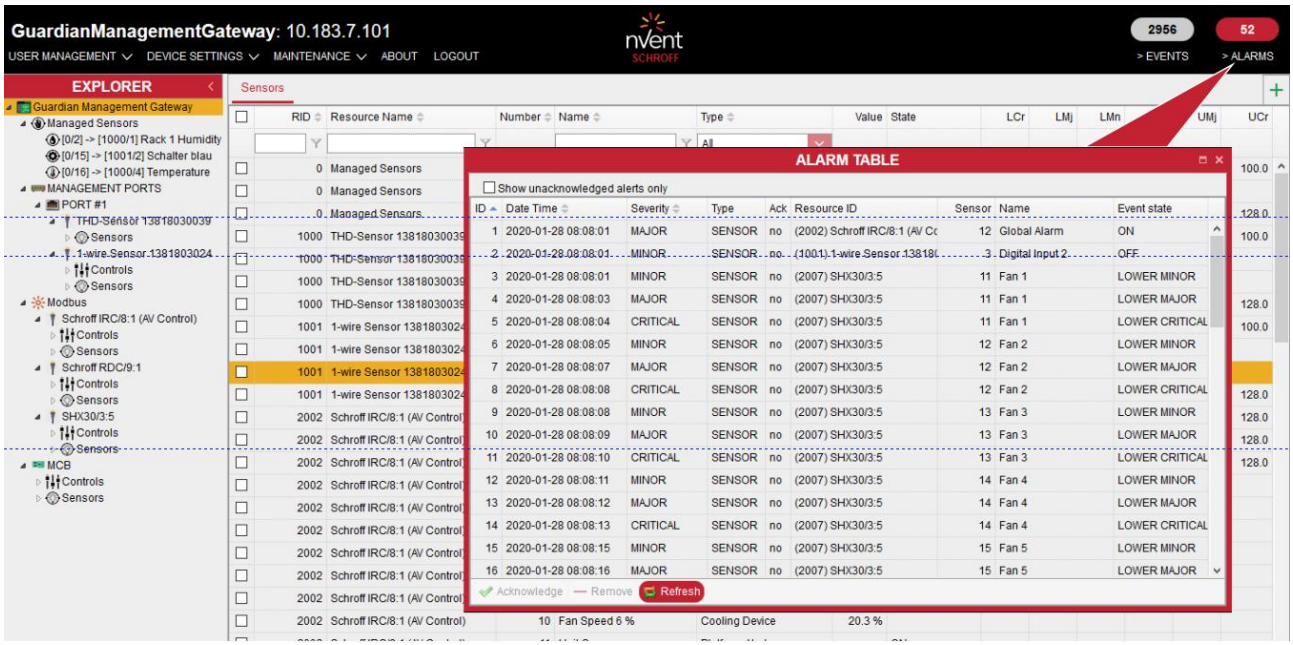


6.1.3 System Event Log

Click on the “Events” label and the “System Event Log” window opens. The rows of the log are colored according to their severity level (see section 22).



6.1.4 **Alarm Table**



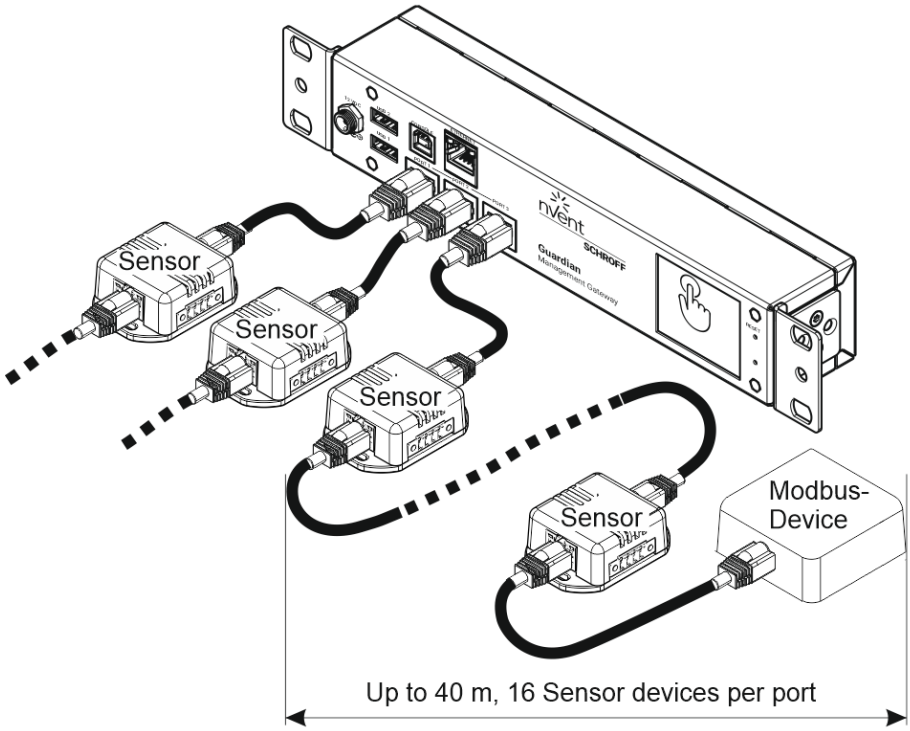
See section 23.

7 Managing External Devices

7.1 Managing Schroff environmental sensors

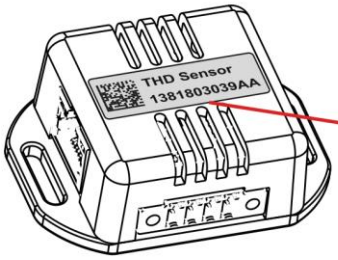
	Only nVent SCHROFF Guardian sensor devices are supported by Guardian Management Gateway.

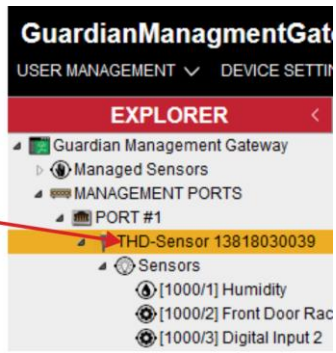
The Guardian Management Gateway offers three sensor management ports with each port capable of monitoring up to 16 sensor devices with a total cable length of 40 meters per port.
The sensors can be chained together and connected to one of the three interface (RJ45) ports labelled: “MANAGEMENT” on the Guardian Management Gateway.
The sensors are hot-pluggable, that means, they can be connected and disconnected at runtime, without restart or reboot.



Example sensor types:

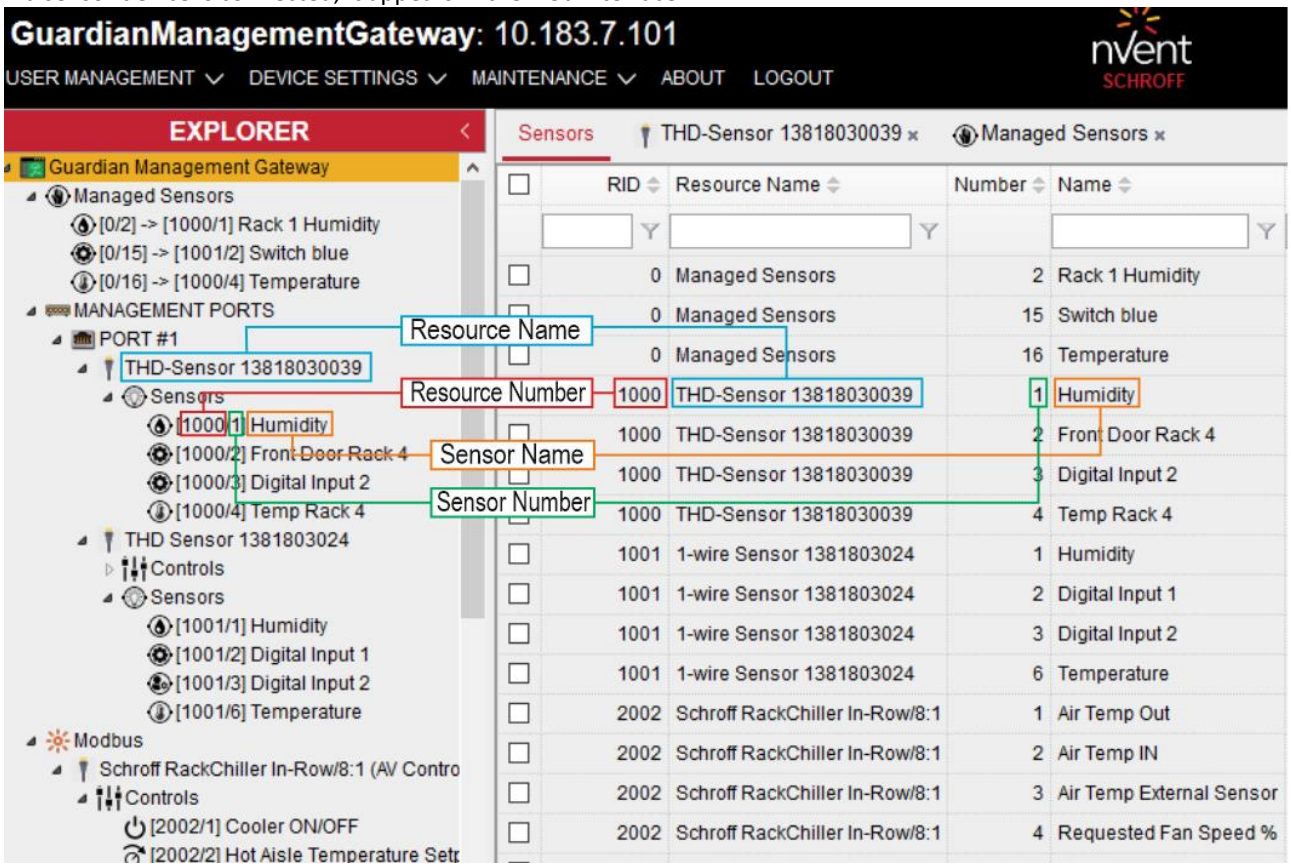
(T) Temperature Sensor
(TH) Multi Sensor (Temperature / Humidity)
(THD) Multi Sensor (Temperature / Humidity / 2x Digital Input)
(D) Digital Sensor 1 (2x Digital Input)





7.1.1 **Overview Schroff sensor devices**

If a sensor device is connected, it appears in the web interface.



Each sensor device is represented as a separate resource with a number in the range 1000-1999, each resource represents one device.

Each resource has the following properties:

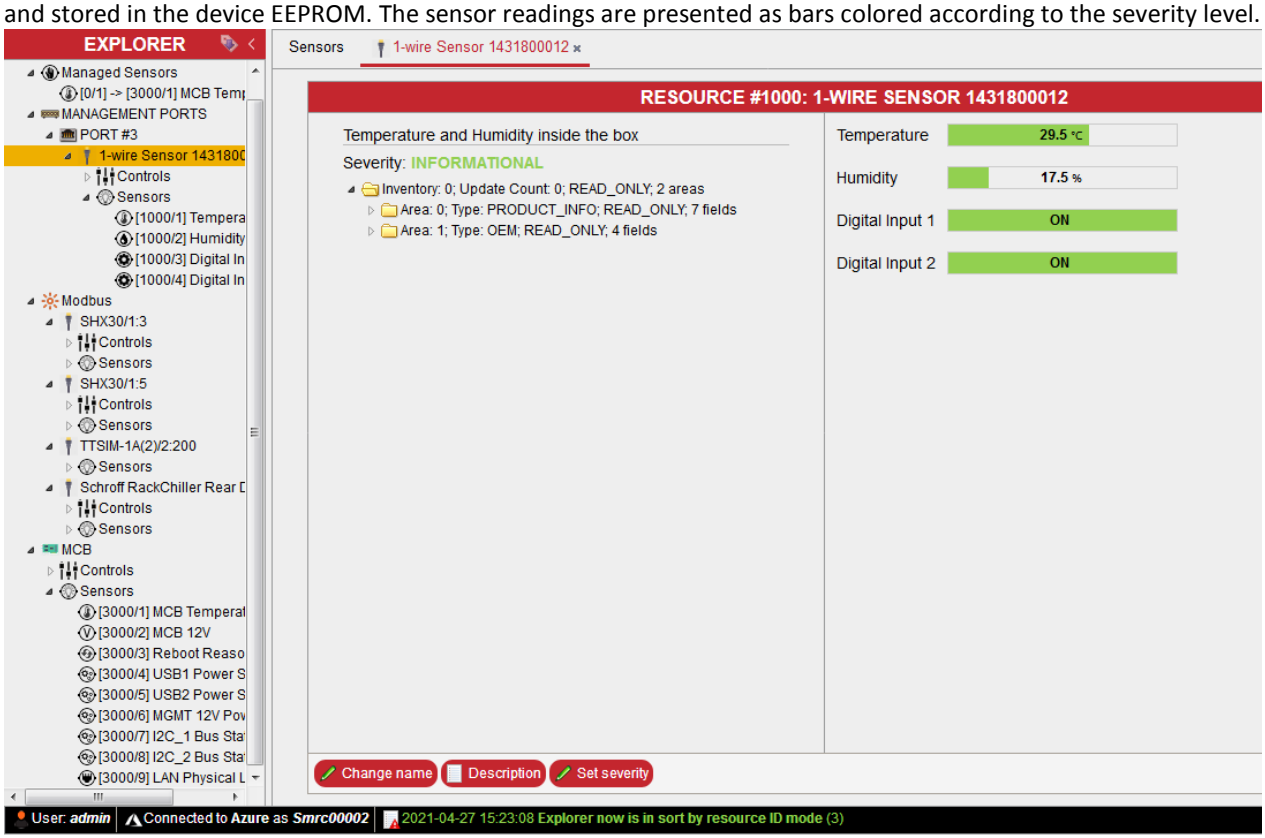
- Resource number, which uniquely identifies the resource in the system
  - The resource number is assigned when the device is first connected to the Guardian Management Gateway. Since each sensor device has a unique serial number, the resource number is associated with the device serial number at this point. When the device is extracted and reinstalled later, Guardian Management Gateway will try to keep the same resource number for it.
- Resource name (also called “resource tag”); this is a human-readable name of the resource that can be changed by the user

Each sensor device exposes the following sensors and controls:

- Sensor “Temperature” report the temperature measured on the device, in degrees C
- Sensor “Humidity” reports the humidity measured on the device, in percentage values
- Sensor “Digital Input 1” this discrete sensor reports the actual state of the Digital Input 1 (*ON* or *OFF*)
- Sensor “Digital Input 2” this discrete sensor reports the actual state of the Digital Input 2 (*ON* or *OFF*)
- Controls “Pin 0 Control” and “Pin 1 Control” (*ON* or *OFF*) (optional)

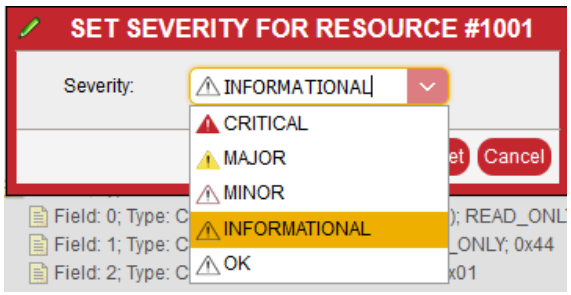
	The digital inputs are pulled to “High”, so the default state is *ON*.

The device DS2406 (family code 12) may be daisy-chained to a 1-wire sensor device. This device provides GPIOs with latching support. The DS2406 has two GPIOs, the Guardian Management Gateway can be configured to use each of them as input GPIO (control) or output GPIO sensor.
The inventory #0 is present on the resource representing the environmental sensor device. This inventory is in IPMI FRU information format and contains minimal information about the device, including its part number, serial number and manufacturer name, and the Device Identification record in the nVent OEM format. The inventory is read-only and stored in the device EEPROM. The sensor readings are presented as bars colored according to the severity level.



Severity

When a resource (sensor device) is removed, an event or alarm is generated. The severity can be set by clicking on the “Set severity” button.



For more information, see HPI model: resources, sensors, controls.

Description

The description of a resource can be set by clicking on the “Description” button. The “Description” attribute is used in BACnet applications.

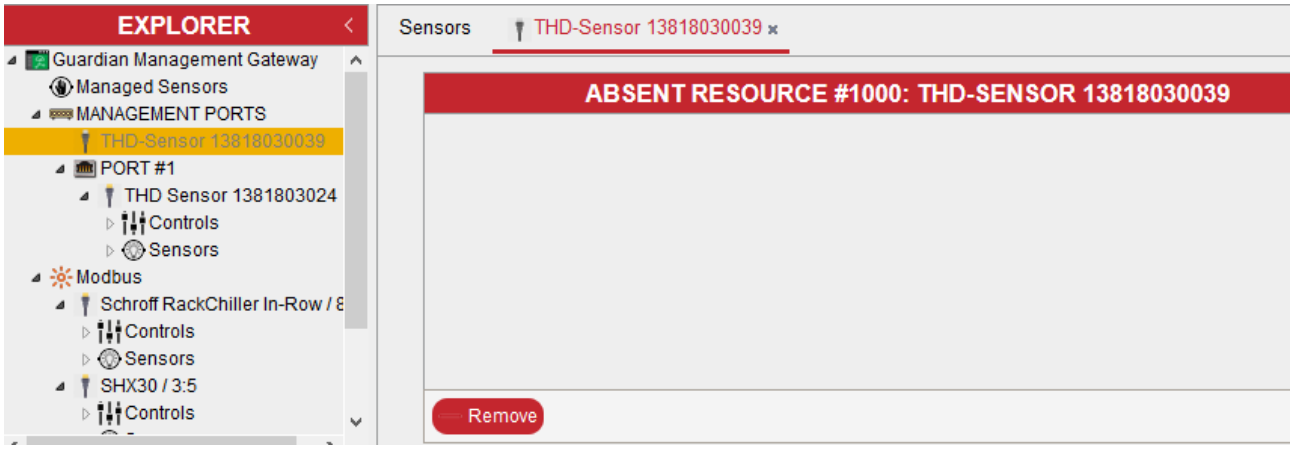

SCHROFF
DESCRIPTION FOR RESOURCE #1000
Temperature and Humidity inside the box
Save
Cancel

7.1.2 Remove sensor device permanently

Sensor devices which are absent (disconnected or broken) are grayed out in the tree pane.

When the sensor is reconnected, it will appear normal again.

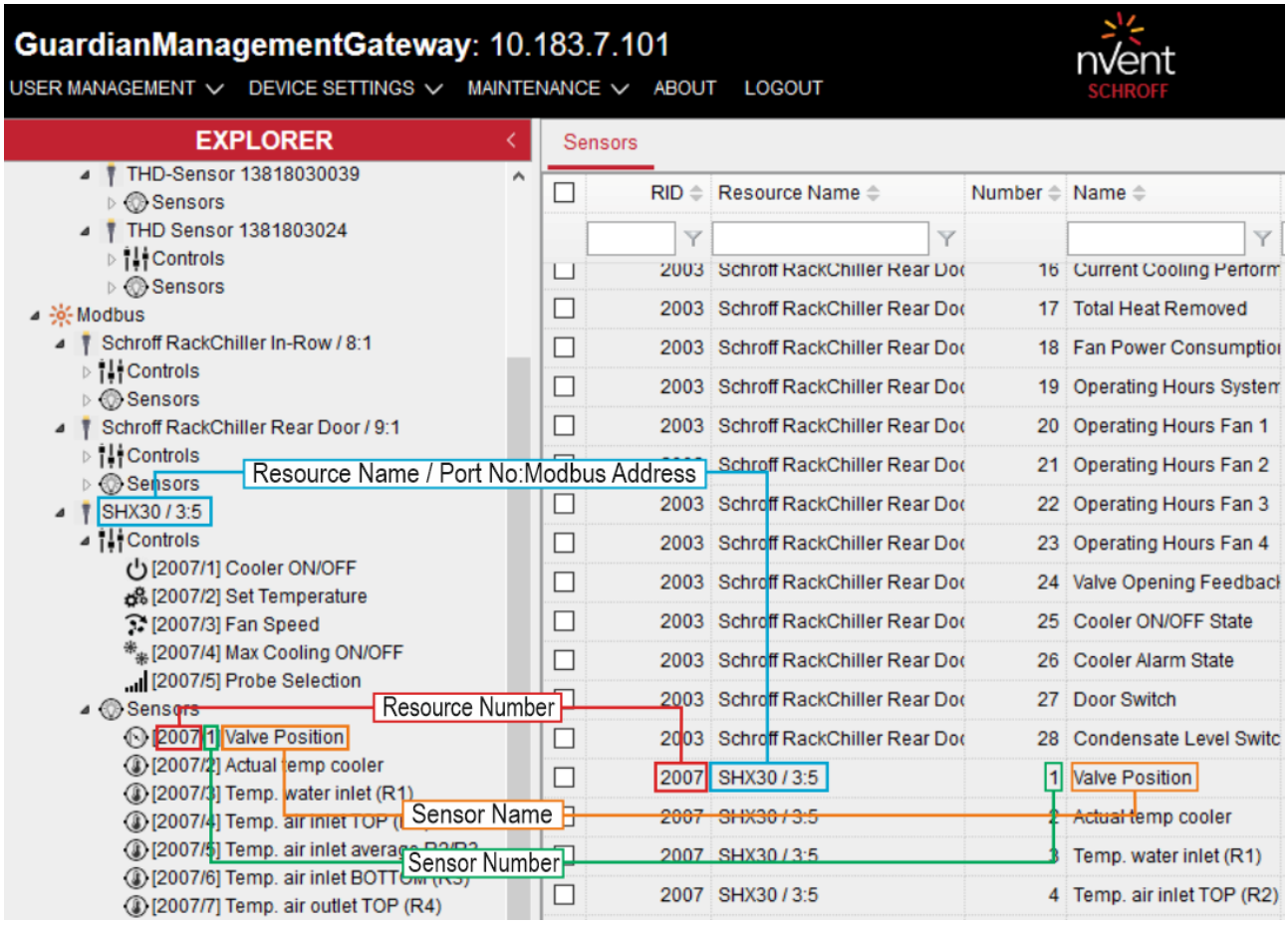
If the sensor is to be permanently removed and thus the resource number released again, the sensor must be removed by pressing the “Remove” button.



7.2 Managing Modbus devices

The Guardian Management Gateway supports external Modbus devices communicating over Modbus TCP or the Modbus serial protocol.
Modbus devices communicating over Modbus serial protocol must be connected to the external interface (management) ports, sharing these ports with environmental sensor devices.
From the software perspective, currently Schroff Side Heat Exchangers (SHX30 and compatibles), TT_SIM leak detection cable controllers are supported; new versions of firmware may add support for other devices.
Modbus devices connected over TCP are also supported. Here is the convention for TCP-connected Modbus devices: the IP address is associated with a specific interface (8 to 255). For a given interface, the Modbus address (1 to 255) is used to select a specific device behind this IP address. Modbus specification assumes that several Modbus devices can be located at the same IP address and can be differentiated by their Modbus address. So we need a Modbus address to distinguish between Modbus devices at the given IP address. RackChiller controller however is a special case that it is the only one controller behind its IP address and responds to any Modbus address in the range 1 to 255, though officially its Modbus address is 1.
From the software perspective, currently of these devices only the RackChiller controllers are supported.
Modbus devices are represented by the HPI resources in the range 2000 - 2999, each resource represents one device. Each device must have a unique combination of a Modbus address and the number of the interface (management port) to which they are connected (1, 2 or 3).

	Port number = interface number! For TCP-connected devices the interface number is 8 and higher.

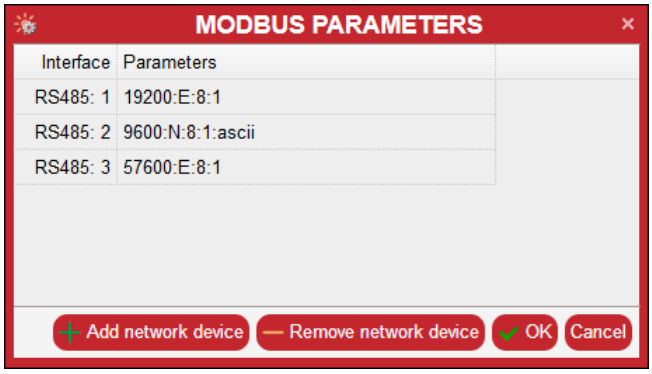


7.2.1 Connecting serial Modbus devices

Before connecting a serial Modbus device to one of the three interface (RJ45) ports labelled: “MANAGEMENT”, adjust the serial port settings (baud rate, number of data and stop bits, parity, and the type of Modbus protocol: ASCII or binary).
For example, Schroff Side Heat Exchangers (SHX30) use the speed of 19200 – 57600 baud, even parity, 8 data bits, 1 stop bit and binary Modbus protocol.
TT_SIM leak detection cable controllers use 9600 baud, no parity, 8 data bits, 1 stop bit and ASCII Modbus protocol.
To accommodate different Modbus devices, Guardian Management Gateway supports its own set of settings for each external interface ports.
Devices with different requirements to the serial port settings should be connected to different interface ports.

7.2.2 Configure Port Settings

To manage Modbus serial settings in the Web interface, invoke the dialog “Modbus Parameters” via the menu command “Maintenance” -> “Modbus” -> “Configure Modbus Parameters”. The dialog allows the user to edit serial settings (as strings) for all supported external interfaces. After changing the settings, press the “OK” button to apply the changes.

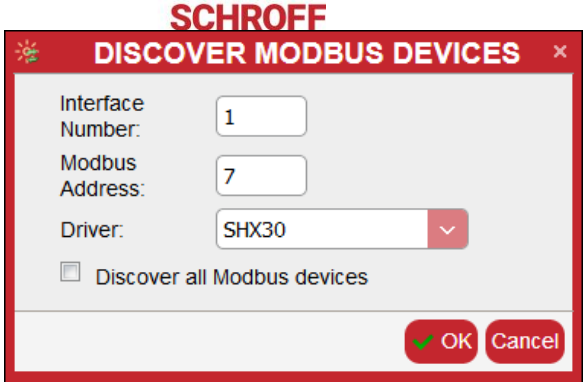


The format of the Parameters field value is the following:

```
<param> ::= <speed>:<parity>:<data-bits>:<stop-bits>[:ascii]
<speed> ::= integer
<parity> ::= N | O | E
<data-bits> ::= 5 | 6 | 7 | 8
<stop-bits> ::= 1 | 2
```

7.2.3 Discovering serial Modbus devices

Modbus devices are semi hot-pluggable: hot extraction is recognized automatically, but to recognize hot inserted devices, a special discovery process should be run (this is because the discovery of new Modbus devices can be quite slow and resource-consuming).
To discover Modbus devices in the Web interface, invoke the command “Maintenance” -> “Modbus” -> “Discover Modbus Devices”. A dialog appears where the user can specify the interface number, the address for directed discovery of a specific device, and a driver for this device.
If the target address is not known, press the checkbox “Discover all Modbus devices” to discover all devices.
If the address of the new Modbus device is known, it is recommended to perform the “directed discovery” which is much faster.
If the driver of the new Modbus device is known, it is recommended to select the driver in the drop-down list. Otherwise, select the “Automatic” item in the drop-down list.
Press the “OK” button to perform the discovery (directed or generic).



The following drivers for serial Modbus devices are installed by default: “SHX30”, “TTSIM”, “Omron”, “Raychem”, “HoffmanCNQDValue”, “Yeeka15xx”. Other drivers may be added via the “Modbus JSON Drivers” dialog window, which is invoked by the “Maintenance” -> “Modbus” -> “Modbus JSON Drivers” menu item (see section 0).

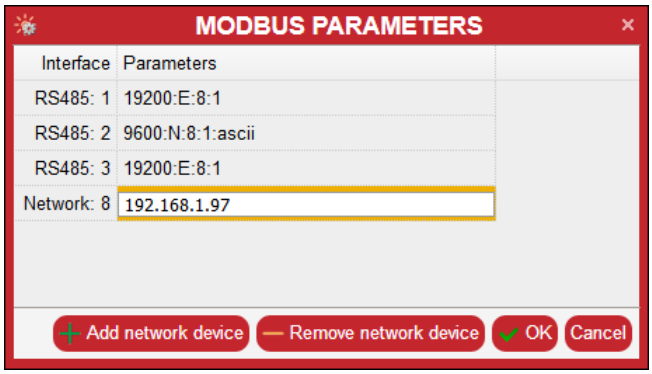
7.2.4 Connecting TCP-connected Modbus devices

	Before you can configure your interface settings for a Modbus TCP device, be sure that the Modbus device is already connected to your network, otherwise the configuration failed!

To establish a TCP connection to the target Modbus device via the Web interface, invoke the dialog:

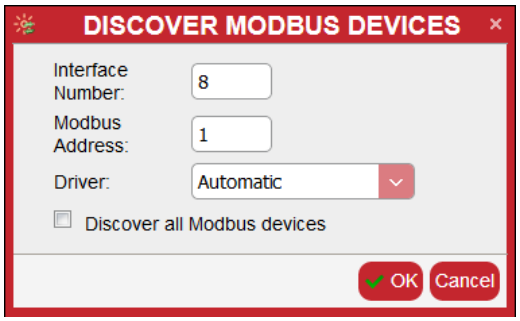
“Modbus Parameters” via the menu command “Maintenance” -> “Modbus” -> “Configure Modbus Parameters”.

The dialog allows the user to enter or edit the IP address for a Modbus TCP interface (The virtual interface number for TCP-connected devices is 8 or higher).
After setting or changing the IP address, press the “OK” button to apply the changes; a failure will not be reported if a TCP connection to the target address cannot be established.



7.2.5 Discovering TCP-connected Modbus devices

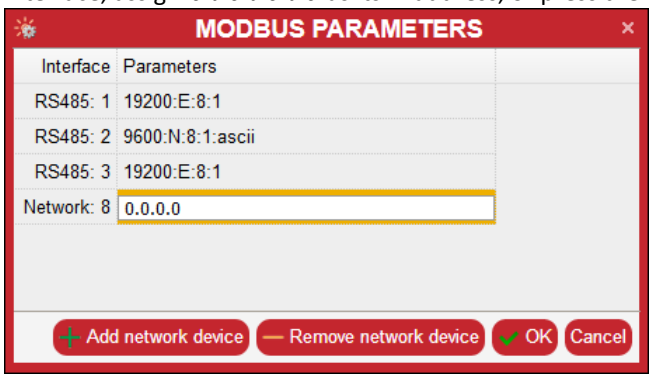
To discover TCP-connected Modbus devices in the Web interface, invoke the command:
“Maintenance” -> “Modbus” -> “Discover Modbus Devices”.
A dialog appears where the user can specify the interface number and address* for directed discovery of a specific device, and a driver for this device.
If the driver of the new Modbus device is known, it is recommended to select the driver in the drop-down list. Otherwise, select the “Automatic” item in the drop-down list. Then, press the “OK” button to perform the directed discovery of the TCP connected device.



The following drivers for TCP-connected devices are installed by default: “InRowCooler”, “Hoffman”, “Carel”. Other drivers may be added via the “Modbus JSON Drivers” dialog window, which is invoked by the “Maintenance” -> “Modbus” -> “Modbus JSON Drivers” menu item (see section 0).

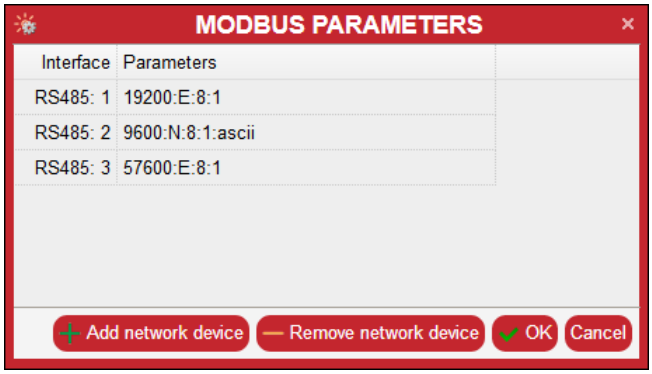
	*The address can be individually assigned by the user in the range 1 to 255. This address is not the IP-address!

To reset a TCP connection to the target Modbus device via the Web interface, invoke the dialog:
“Modbus Parameters” via the menu command “Maintenance” -> Modbus” -> “Configure Modbus Parameters”.
To add a new network interface press the “Add network interface” button. To disable the supported network interface, assign 0.0.0.0 as its IP address, or press the “Remove network device” button.



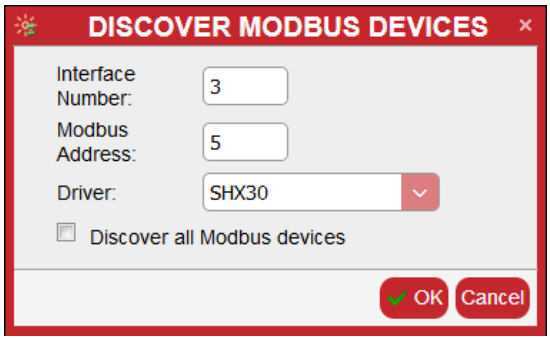
7.2.6 Managing Schroff Side Heat Exchangers SHX30

Configure/check serial port setting via the menu command “Maintenance” -> “Modbus” -> “Configure Modbus Parameters”.
In this example port (interface) *1* and *3* are configured for the SHX30.



	Schroff Side Heat Exchangers (SHX30) use the speed of 19200 – 57600 baud, even parity, 8 data bits, 1 stop bit and binary Modbus protocol.

Connect the SHX30 to a management port (assuming it is connected to port *3* and the Modbus address is *5*) and invoke the command: “Maintenance” -> “Modbus” -> “Discover Modbus Devices”.



After entering the port and Modbus address, click OK. It discovers the Modbus present at that location.

For Schroff Side Heat Exchangers (SHX30), the following sensors and controls are available:

Controls:

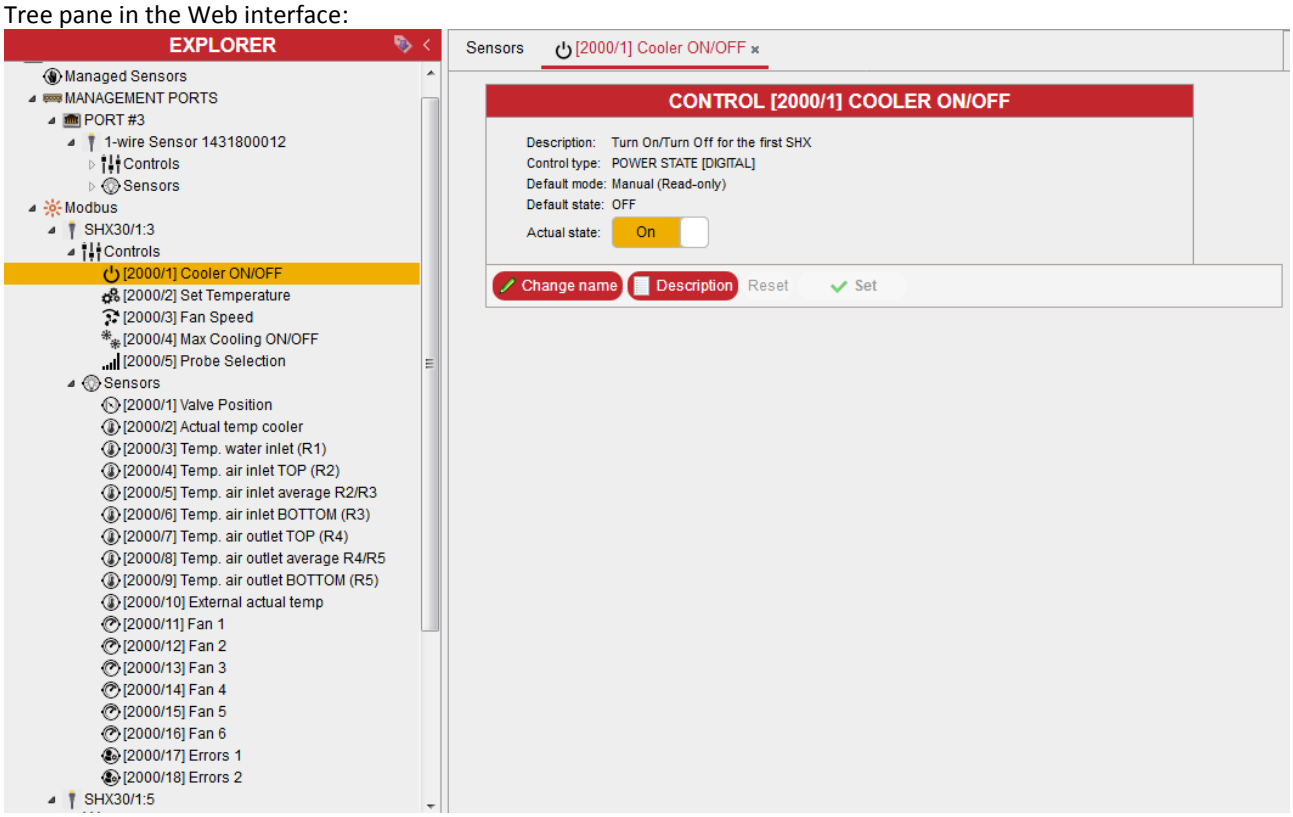
1	**Cooler ON/OFF**	This digital control can be used to turn the cooler on or off
2	**Set Temperature**	This control is numeric, it sets the temperature set point for the cooler in the range between 18 and 40 degrees C
3	**Fan Speed**	This control is numeric, it specifies the desired fan speed in percentages between 30% and 100%
4	**Max Cooling ON/OFF**	A digital control that allows the user to turn on or off maximum cooling mode
5	**Probe Selection**	A discrete control. The following states are supported: (0) None (1) Inlet Air temp. Top (2) Average Inlet Air temp. (3) Inlet Air temp. Bottom (4) Outlet Air temp. Top (5) Average Outlet Air temp. (6) Outlet Air temp. Bottom (7) Room temp.

Sensors:

1	**Valve Position**	Reports the current valve position, in percentages of fully open state (0 to 100)
2	**Actual temp cooler**	Reports the actual average outlet temperature
3	**Temp. water inlet (R1)**	Reports the cooling water inlet temperature
4	**Temp. air inlet TOP (R2)**	Reports the upper air inlet temperature
5	**Temp. air inlet average R2/R3**	Reports the average air inlet temperature
6	**Temp. air inlet BOTTOM (R3)**	Reports the bottom air inlet temperature
7	**Temp. air outlet TOP (R4)**	Reports the upper air outlet temperature
8	**Temp. air outlet average R4/R5**	Reports the average air inlet temperature
9	**Temp. air outlet BOTTOM (R5)**	Reports the bottom air inlet temperature
10	**External actual temp**	Reports the temperature of an external temp. sensor
11	**Fan 1**	Reports the speed of Fan 1 in revs/min
12	**Fan 2**	Reports the speed of Fan 2 in revs/min
13	**Fan 3**	Reports the speed of Fan 3 in revs/min
14	**Fan 4**	Reports the speed of Fan 4 in revs/min
15	**Fan 5**	Reports the speed of Fan 5 in revs/min
16	**Fan 6**	Reports the speed of Fan 6 in revs/min
17	**Errors 1**	Discrete sensors that report various errors detected by the heat exchanger in their state masks; multiple bits may be simultaneously set in their state masks
18	**Errors2**	Discrete sensors that report various errors detected by the heat exchanger in their state masks; multiple bits may be simultaneously set in their state masks

The “SHX30” driver corresponds to a device of this type.

Tree pane in the Web interface:



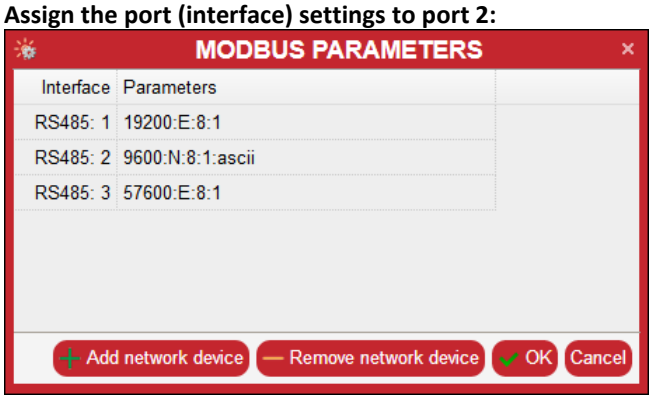
To retrieve the sensor or control data, click on the sensor or control for which you want to know the data.

Inventory of Schroff Side Heat Exchangers (SHX) contains Product Information area only.

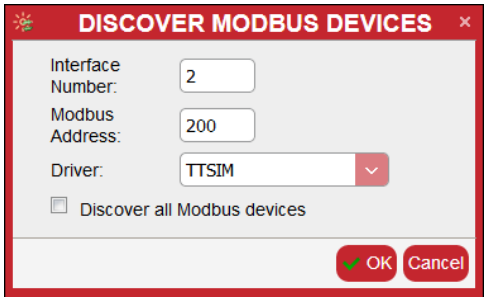
7.2.7 Managing TT_SIM Leak detection cable controllers

Connect the device to a management port (assuming it is connected to Interface (Management Port) *2* and the Modbus address is *200*).

Assign the port (interface) settings to port 2:



Discover the Modbus device:



For TT_SIM Leak detection cable controllers, the following sensors are available:

1	**Leak**	A discrete sensor that reports whether a leak has been detected
2	**Contamination**	A discrete sensor that reports whether cable contamination has been detected
3	**Leak Location**	Reports the leak location, in meters
4	**Cable Break**	A discrete sensor that reports whether the cable has been physically broken
5	**Fault**	A discrete sensor that reports whether any other fault has been detected
6	**Circuit Length**	Reports the total cable length, in meters
7	**Detection Current**	Reports the current in the cable, in milliamperes
8	**Status**	Reports the current contents of the controller status word, as an opaque numeric value

The “TTSIM” driver corresponds to a device of this type.

To retrieve sensor values for the TT_SIM Leak detection cable controllers, click on the resp. sensor in the tree pane of the Web interface

Inventory and controls are not available for TT_SIM Leak detection cable controllers.

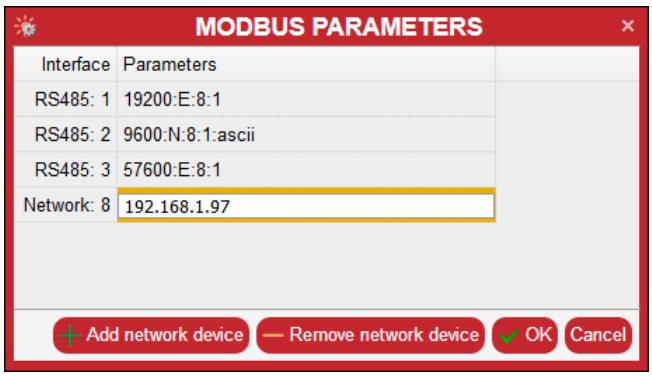
	The default Modbus address of TT_SIM Leak detection cable controller is either 199 or 200.

7.2.8 Managing Schroff RackChiller devices

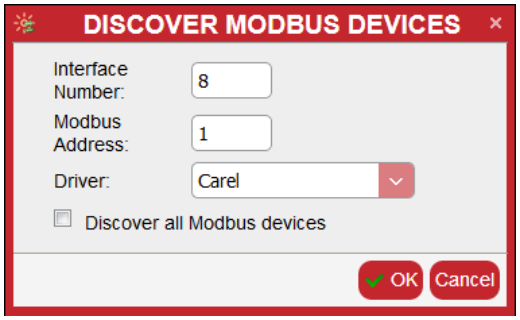
	Schroff RackChiller devices can be accessed only via TCP.

Connect the device to your network (assuming the IP address is *192.168.1.97*).

To establish a TCP connection to the target Modbus device via the Web interface, invoke the dialog:
“Modbus Parameters” via the menu command “Maintenance” -> “Modbus” -> “Configure Modbus Parameters”.
After setting or changing the IP address, press the OK button to apply the changes; a failure will not be reported if a TCP connection to the target address cannot be established.



Discover the Modbus device:
To discover TCP-connected Modbus devices in the Web interface, invoke the command:
“Maintenance” -> “Modbus” -> “Discover Modbus Devices”.
Specify the interface number and address*, then press the “OK” button to perform the directed.



	*The address can be individually assigned by the user in the range 1 to 255. This address is not the IP-address!

RackChiller Rear Door:
For Schroff RackChiller Rear Door devices, the following sensors and controls are available (NOTE: this list may be expanded in the future):

Controls:

1	**Cooler ON/OFF**	This digital control can be used to turn the cooler on or off
2	**Max Cooling ON/OFF**	Allows the user to turn on or off maximum cooling mode
3	**Temperature Control Variable**	Set the control variable for the temperature regulation. The following parameters are available: (0) Manual (Opening ration water valve in %) (1) Outlet Temp Air Top (2) Outlet Temp Air Bottom (3) Average Outlet Air Temp (default) (4) Temp Water Outlet (5) dT Water Inlet/Outlet
4	**Temperature Setpoint**	Setpoint for the temperature. If the control is made by a temperature sensor, the temperature can be set in °C (°F), with

		manual control, the opening ratio of the water valve can be set manually.
5	**Fan Speed Control Mode**	Control variable for the fan speed. The following parameters are available: (0) Manual (%) (1) Pressure Difference
11	**Pressure Differential Setpoint**	Setpoint for controlling the fan speed. If the control is via the differential pressure sensor, the pressure can be set in the range of -150 Pa to +150 Pa. Negative differential pressure means that the pressure in the cabinet is higher than the ambient pressure. A setting of approx. +20 Pa is recommended.
12	**Manual Water Valve Position**	When the Temperature Control Variable is set to “Manual”, the opening ratio of the water valve can be set manually in %.
13	**Manual Fan Level**	When the Fan Speed Control Variable is set to “Manual”, the fan speed can be set from 20 - 100 %.

Sensors:

1	**Air Temp Out Top**	Reading of the upper temperature sensor at the air outlet (Cold air)
2	**Air Temp Out Bottom**	Reading of the lower temperature sensor at the air outlet (Cold air)
3	**Air Temp In Top**	Reading of the upper temperature sensor located in front of the RackChiller (Warm air)
4	**Air Temp In Bottom**	Reading of the lower temperature sensor located in front of the RackChiller (Warm air)
5	**Air Differential Pressure**	Differential Pressure Inside/Outside cabinet
6	**Water Temp In**	Temperature Water inlet
7	**Water Temp Out**	Temperature Water outlet
8	**Water Flow**	Water flow
9	**Water Pressure**	Water pressure
10	**Requested Valve Opening**	Requested opening ratio of the water valve in %
11	**Requested Fan Speed**	Reports the requested fan speed by the controller
12	**Speed Fan 1**	Reports the actual speed of fan 1 (upper fan)
13	**Speed Fan 2**	Reports the actual speed of fan 2
14	**Speed Fan 3**	Reports the actual speed of fan 3
15	**Speed Fan 4**	Reports the actual speed of fan 4 (lower fan)
16	**Current Cooling Performance**	Reports the Current Cooling Performance
17	**Total Heat Removed**	Reports the Total Heat Removed in kW/h
18	**Fan Power Consumption**	Reports the actual power consumption of the fans
19	**Operating Hours System**	Reports the operating hours of the RackChiller
20	**Operating Hours Fan 1**	Reports the operating hours of fan 1
21	**Operating Hours Fan 2**	Reports the operating hours of fan 2
22	**Operating Hours Fan 3**	Reports the operating hours of fan 3
23	**Operating Hours Fan 4**	Reports the operating hours of fan 4
24	**Valve Opening Feedback**	Reports the actual valve opening ratio
25	**Cooler ON/OFF State**	Reports the Cooler ON/OFF State
26	**Cooler Alarm State**	Reports the cooler alarm state
27	**Door Switch**	Reports the state of an optional door switch
28	**Condensate Level Switch**	Reports the state of an optional switch for the water level in the condensate tray

The “Carel” driver corresponds to a device of this type.

RackChiller In-Row:

For Schroff RackChiller In-Row devices, the following sensors and controls are available (NOTE: this list may be expanded in the future):

Controls:

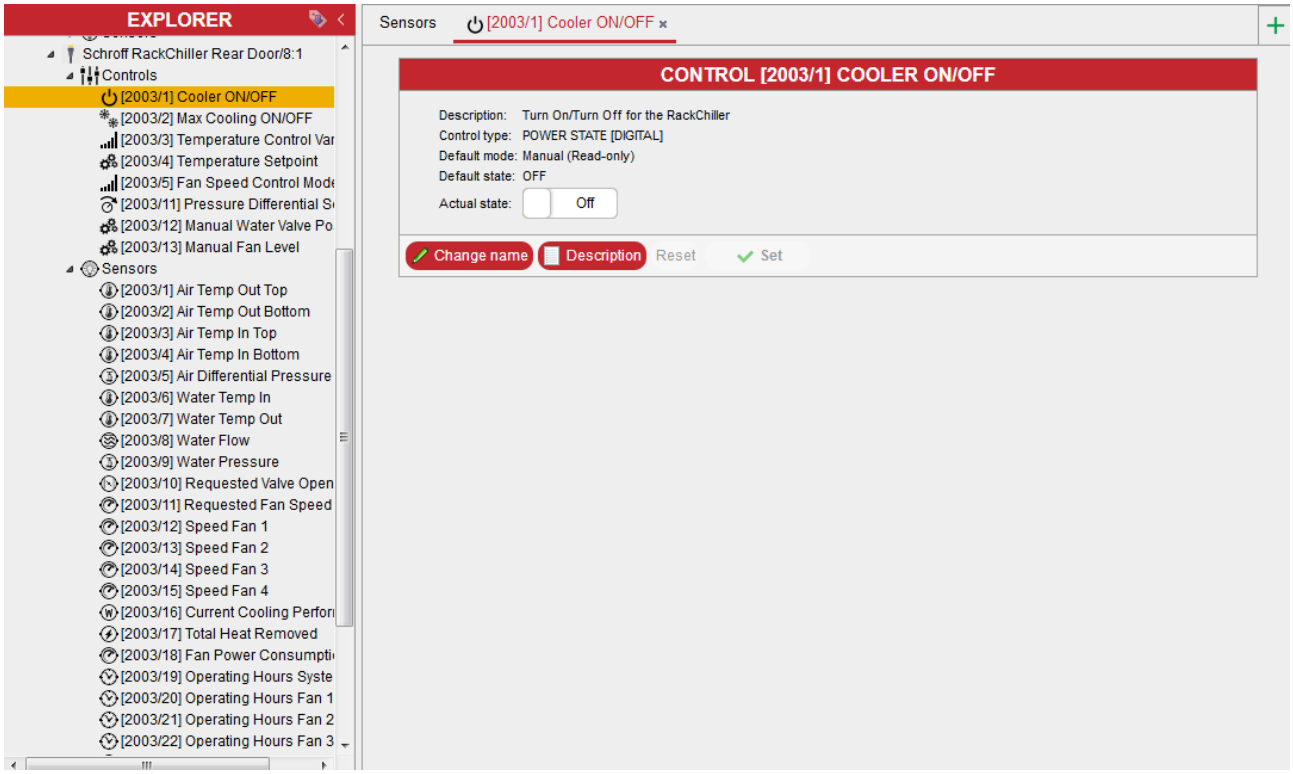
1	**Cooler ON/OFF**	This digital control can be used to turn the cooler on or off
2	**Hot Aisle Temperature Setpoint**	
3	**Hot Aisle Temperature Differential**	
4	**Cold Aisle Temperature Differential**	

Sensors:

1	**Air Temp Out**	Reading of the temperature sensor at the air outlet (Cold air)
2	**Air Temp In**	Reading of the temperature sensor at the air inlet (Warm air)
3	**Air Temp External Sensor**	Reading of the external temperature sensor
4	**Requested Fan Speed %**	Reports the requested fan speed by the controller
5	**Requested Valve Opening**	Requested opening ratio of the water valve in %
6	**Valve Opening Feedback**	Reports the actual valve opening ratio
7	**Fan Speed 1 %**	Reports the actual speed of fan 1
8	**Fan Speed 2 %**	Reports the actual speed of fan 2
9	**Fan Speed 3 %**	Reports the actual speed of fan 3
10	**Fan Speed 4 %**	Reports the actual speed of fan 4
11	**Fan Speed 5 %**	Reports the actual speed of fan 5
12	**Fan Speed 6 %**	Reports the actual speed of fan 6
13	**Cooler ON/OFF State**	Reports the Cooler ON/OFF State
14	**Cooler Alarm State**	Reports the cooler alarm state

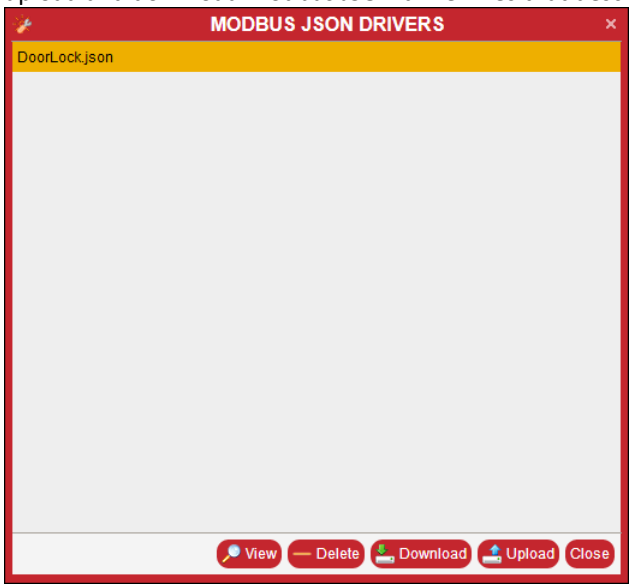
The “InRowCooler” driver corresponds to a device of this type.

To retrieve control or sensor values for the Rack Chiller, click on the resp. control or sensor in the tree pane of the Web interface.



7.2.9 Modbus JSON drivers

Drivers other than the default ones may be added via the “Modbus JSON Drivers” dialog window, which is invoked by the “Maintenance” -> “Modbus” -> “Modbus JSON Drivers” menu item. This dialog enables a user to view, delete, upload and download Modbus JSON driver files that describe correspondent drivers.



The syntax of Modbus JSON driver files is described in “Modbus Driver JSON Description” document.

7.3 Reachability

For user convenience, Guardian Management Gateway provides a facility to detect whether a certain system (server) is reachable over the network. It does this by periodically pinging the given address and storing the results in a special table. When a registered system becomes reachable (ping becomes successful) or becomes unreachable (ping becomes unsuccessful), Guardian Management Gateway changes the state of the target system in the table and generates a corresponding event ("Server reachable" or "Server unreachable"). Another pair of events is generated when systems are added to the reachability verification list ("Server Monitoring Starts") or deleted from the list ("Server Monitoring Stops"). These events are sent as HPI software events, are subject to event filtering and are placed into the System Event Log.

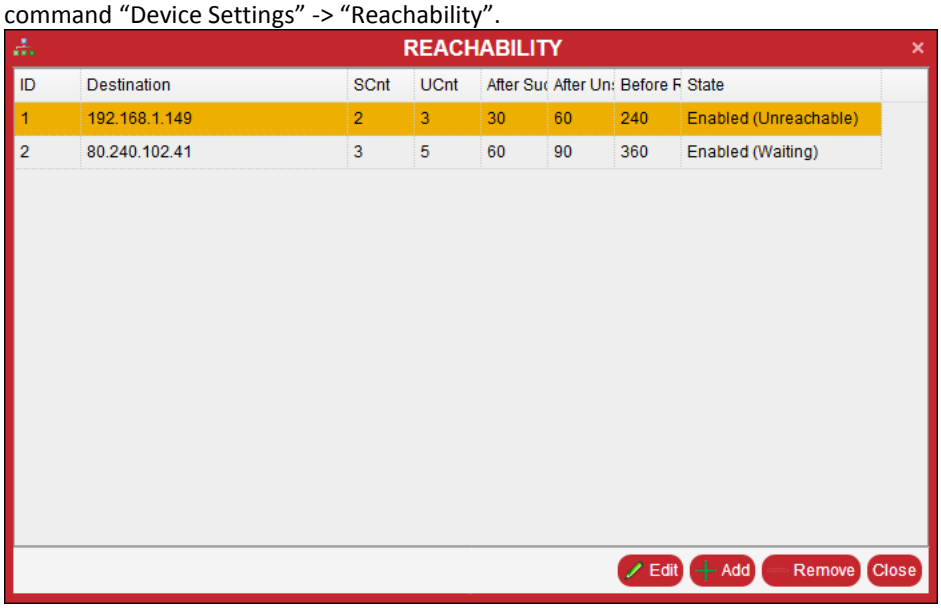
The following actions are available for user in connection with the Reachability feature:

- Add an IP address or the name of the system to the reachability verification list, and specify ping parameters
- Update ping parameters for the specified position in the list
- Enable/disable pinging for the previously specified system, by its position in the list
- Get the current reachability verification list, with system names or IP addresses, their status and ping parameters.

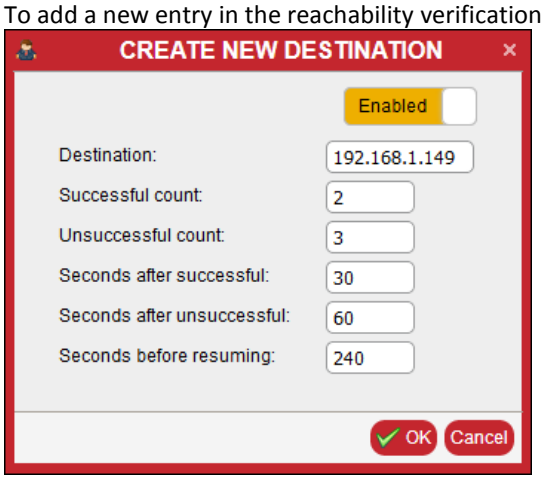
The following ping parameters can be specified for a certain system:

- Success count: after how many successful pings the system should be considered reachable
- Unsuccessful count: after how many unsuccessful pings the system should be considered unreachable
- Seconds after successful: a delay in seconds between a successful ping and the next ping
- Seconds after unsuccessful: a delay in seconds between an unsuccessful ping and the next ping (unless the target has been considered unreachable after this unsuccessful ping)
- Seconds before resuming: a delay in seconds to resume pinging after that target has been considered unreachable.
- Whether to enable reachability test for this system (true/false).

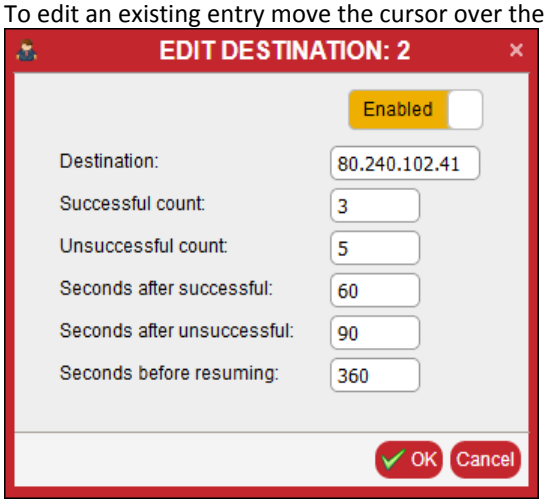
To manage the reachability verification list from the Web interface, use the Reachability dialog, invoked via the menu command "Device Settings" -> "Reachability".



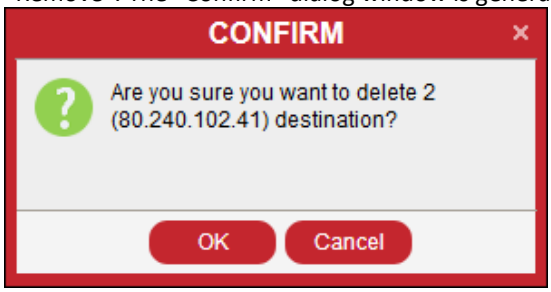
To add a new entry in the reachability verification list press the button “Add”.



To edit an existing entry move the cursor over the entry and press the button “Edit”.



To delete an existing entry from the reachability verification list move the cursor over the entry and press the button “Remove”. The “Confirm” dialog window is generated.



8 HPI model: resources, sensors, controls

The software architecture of the Guardian Management Gateway conforms to the Hardware Platform Interface (HPI) model by Service Availability Forum. This model is defined in the Hardware Platform Interface Specification.

Hardware Platform Interface provides an abstract model of underlying hardware, using abstract concepts of resources, sensors, controls and inventory.

A system comprises multiple resources, and the resource population is dynamic, that is, it may change over time.

Existing resources may be removed from the system and new resources may appear.

Each resource abstracts a hardware field-replaceable unit (FRU) and includes multiple sensors, controls and an optional inventory.

- Sensors abstract physical sensor devices
- Controls abstract physical control mechanisms (e.g. GPIO pins in the output mode)
- Inventory is a data storage that contains information about the resource in a standardized format.

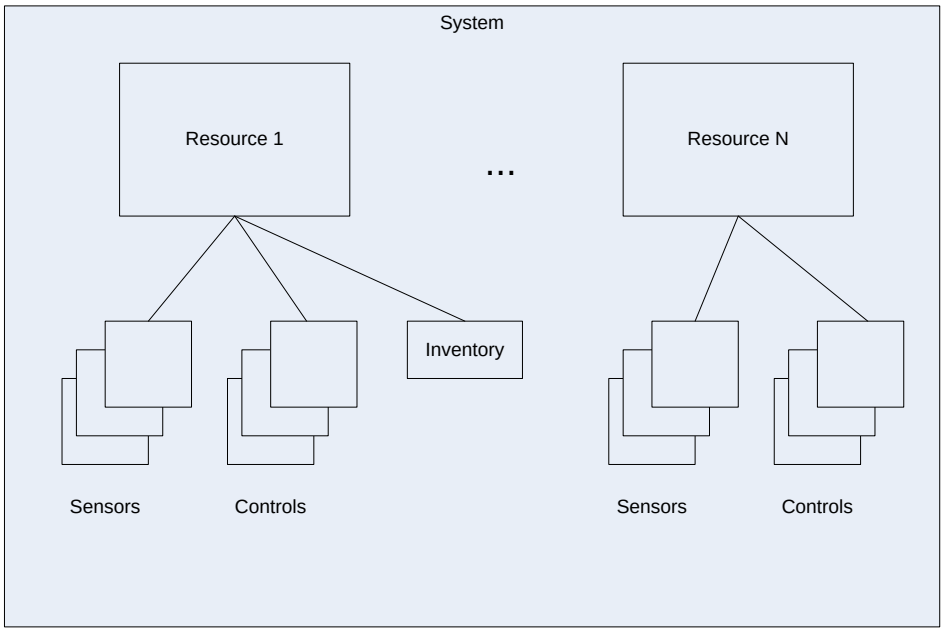
When something important happens in the system (e.g. a configuration change or an alert condition on a sensor) an event is generated.

Events are data packets in a standardized format, they are processed by the event filters (and can generated subsidiary actions, like sending an e-mail message or an SMS, or executing some predefined actions, or generating an SNMP trap).

All events are stored in the system event log where they can be examined in a later time.

8.1 Resources

In the Guardian Management Gateway architecture, resources normally represent hardware FRUs (though some of them may be hot-inserted and removed while some remain static).



Each resource has the following properties:

- Resource number, which uniquely identifies the resource in the system
- Resource name (also called “resource tag”); this is a human-readable name of the resource that can be changed by the user
- Resource description; the “Description” attribute is used in BACnet applications
- Capabilities; this is the mask of binary flags that identifies what capabilities the resource has. The most commonly used capabilities are:
  - Resource contains sensors
  - Resource contains controls
  - Resource hosts an inventory
- Resource entity path, which identifies the position of the resource in the hierarchy of entities in the system (in the machine-readable form).
- Resource severity, that identifies the severity of an event generated when this resource is removed.

The resource numbers are fixed and assigned as follows:

Resource 0 (“Managed Sensors”): this resource is virtual. It holds managed sensors and the inventory for the whole Guardian Management Gateway.

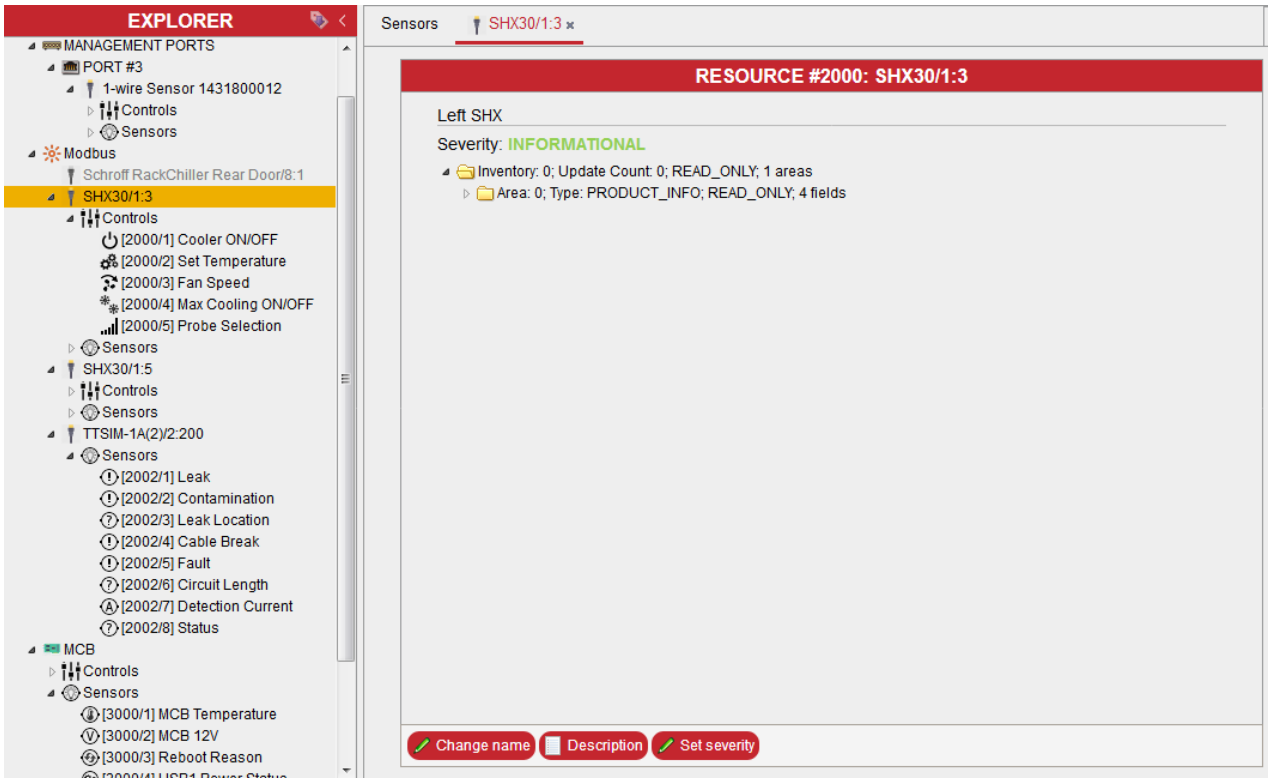
Resources 1000 - 1999: these resources represent Schroff environmental sensor devices. These devices are hot-swappable and carry several sensors and controls on them. Each sensor device holds an inventory that contains the serial number of the device.

Resources 2000 - 2999: these resources represent Modbus devices.

Currently, the Schroff SHX30 cooling units, the Schroff RackChillers and the TraceTek TTSIM-1A leak detection alarm unit are supported as Modbus devices. These devices are hot-swappable, but in the case of hot insertion, a manual discovery of a new device should be initiated by the user.

Resource 3000: the master control board (MCB). This is the board which hosts the single-board computer on which the management software is run.

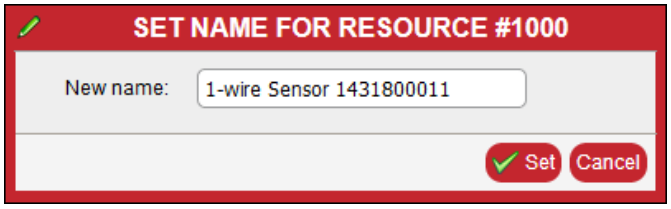
In the Web interfaces, resources are visible in the left (tree) pane of the screen as tree nodes.



When a Modbus device or a Schroff environmental sensor device is either extracted or disconnected from the Guardian Management Gateway the correspondent node in the tree pane turns grey.

8.1.1 Change a resource name

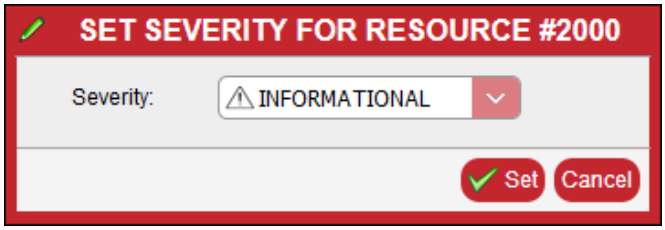
To change the resource name, select the resource in the tree pane. In the middle pane, the resource inventory will be shown. Press the button “Change name” at the bottom of the screen.
The dialog box “Set Name for Resource #NNN” appears:



Change the current resource name and press the “Set” button.

8.1.2 Change the resource severity

Press the button “Set severity” at the bottom of the screen. The dialog box “Set Severity for Resource #NNN” appears:



Change the current resource severity and press the “Set” button.

8.1.3 Change the resource description

Press the button “Description” at the bottom of the screen. The dialog box “Description for Resource #NNN” appears:

Change the resource description and press the “Save” button.

8.2 Sensors

Sensors in the HPI model represent devices that collect and report information about the environment and the state of the system itself. These devices can be physical or logical, and are considered components of FRUs (which are represented as resources).

	Each sensor belongs to some resource.

8.2.1 Numeric and discrete sensors

There are two classes of sensors: numeric and discrete.

Numeric sensors

Numeric sensors report a numeric reading and the state mask.
The reading can be signed integer, unsigned integer or a floating-point number.
In all cases the values are 64 bits in size.

Thresholds

Thresholds can be specified for a numeric sensor.
There are three upper thresholds:

- Upper Critical Threshold
- Upper Major Threshold
- Upper Minor Threshold

and three lower thresholds:

- Lower Critical Threshold
- Lower Major Threshold
- Lower Minor Threshold

Not all thresholds need to be defined. Normally the sensor value should be between lower and upper thresholds. The state mask for numeric sensors reports which thresholds are crossed at a given moment of time. When a sensor value crosses a threshold, this is considered an abnormal situation, and an event may be generated (if the sensor configuration and event enable mask allows it).

Hysteresis

Hysteresis values can be specified to prevent generation of numerous events when the sensor reading oscillates in a vicinity of a certain threshold.

- Positive Hysteresis
- Negative Hysteresis

Usually the position of the thresholds corresponds to the figure below, and the sensor value is normally located between lower minor and upper minor thresholds.

A hysteresis is taken into account when the sensor value goes back into the normal range, crossing an upper threshold in the downside direction or lower threshold in the upside direction. For a de-assertion event, the sensor value should become less than Threshold – Positive Hysteresis in the first case, and greater than Threshold + Negative Hysteresis in the second case.

Discrete sensors

Discrete sensors do not report a numeric reading, they report only the state mask.
The state mask can comprise up to 16 states.
A sensor may be in several states simultaneously, but most often it is only in one single state at a given moment of time.
Discrete sensors may generate events when changing states.
An example of a discrete sensor can be:

- A presence sensor (some entity is present or absent)
- A failure sensor for a component (component operational / component failed) or
- A reboot reason sensor (with the set of states corresponding to different reasons of last reboot, e.g. “power up”, “hardware reset”, “software reboot”, “reset after upgrade”, etc.).

For a discrete sensor, event severity is assigned to each state and can be changed by a user.
This severity is propagated to the event that is generated when the sensor gets into this state, and allows to distinguish between “normal” and “abnormal” states for the sensor.

- A “normal” state should be assigned the severity “OK” or “Informational”
- An “abnormal” state should be assigned the severity “Minor”, “Major” or “Critical” depending on the severity of this abnormality.

For sensors with thresholds, event severity corresponds to the severity of the corresponding threshold and should not be changed by a user.

8.2.2 Sensor attributes and configuration parameters

Each sensor has static attributes and dynamic configuration parameters.

- Static attributes are defined when the sensor is created by the system and are read-only for a user.
- Dynamic configuration parameters can be changed by a user.

Static sensor attributes:

- Sensor number
- Sensor type (e.g. temperature, voltage, humidity, presence)
- Event category (threshold-based, state asserted/deasserted, reboot reason)
- Can sensor be dynamically enabled or disabled?
- Event control: can events be globally enabled/disabled, has the sensor per-state event control?
- Bit mask of supported states
- Data format: is sensor numeric or discrete?
- For a numeric sensor:
  - Type of the numeric reading (integer, unsigned, float)
  - Base units (meters, volts, amperes, etc.)
  - Modifier units (e.g, seconds)
  - Modifier use (multiply or divide)
  - Base unit's factor (e.g. kilo=10^3, milli=10^{-3}, etc.)
  - Modifier unit's factor
  - Is sensor reading a percentage?
  - Range of valid sensor readings
  - Accuracy, resolution, tolerance for the sensor reading
  - Supported thresholds and hysteresis, as a bit mask
  - Writable thresholds and hysteresis, as a bit mask (those that can be changed by a user)
- For a discrete sensor:
  - Default assignment of severities to event states (identifies normal and abnormal states for the sensor)

Dynamic sensor attributes:

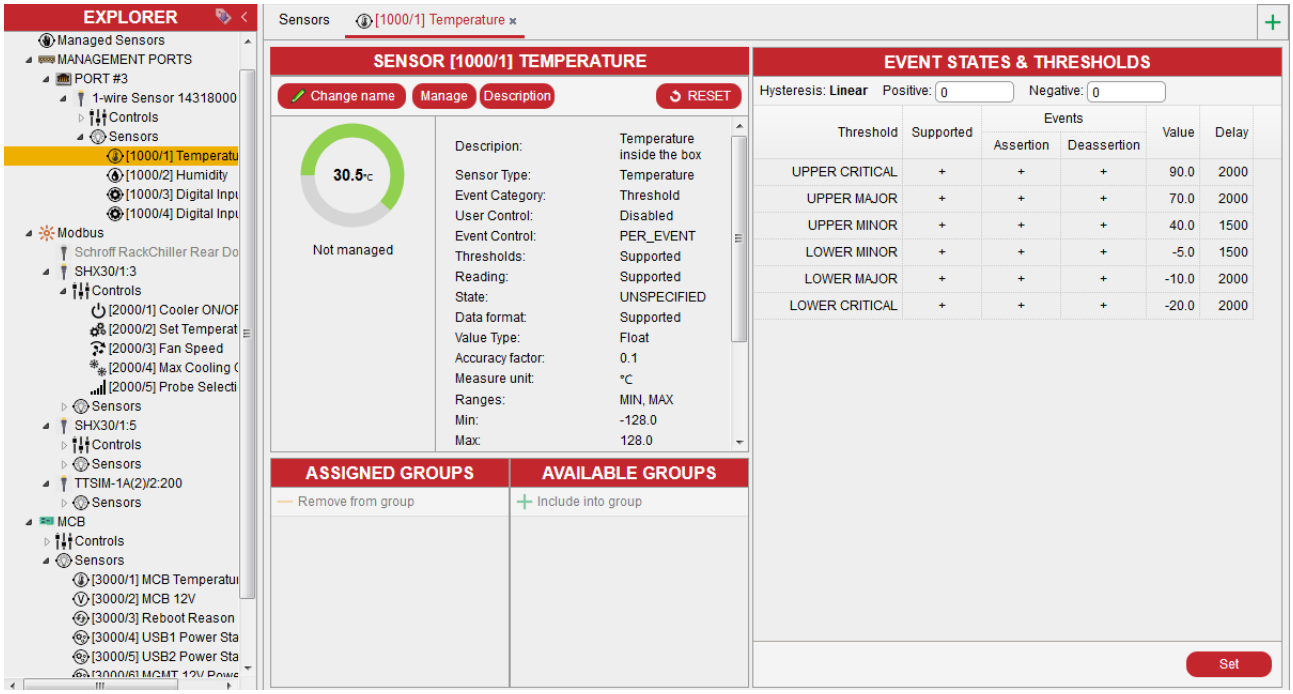
The user can change the following configuration parameters for a sensor:

- Enable or disable the sensor (if static attributes allow that)
- Enable or disable events for specific states and/or globally (if static attributes allow that)
- Change sensor human-readable name
- Change sensor human-readable description (the "Description" attribute is used by BACnet applications)
- Change values of supported thresholds and hysteresis
- Change polling period for the sensor (in milliseconds)
- Change either the Assertion Delay count or the Per-Threshold Assertion Delays (for how long a threshold should be crossed to generate an event); this setting prevents spontaneous events in the case of random errors in sensor readings
- Change severities assigned to specific event states.

8.2.3 Managing sensors with the Web interface

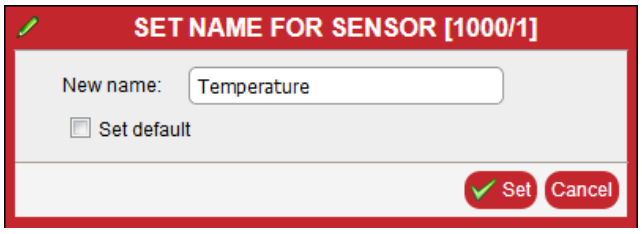
Managing specific sensors

To manage a specific sensor, choose it in the left (tree) pane; instruments are shown under the resources that own them:



When a numeric sensor is selected in the left pane, the middle pane shows the sensor reading and properties, and the right pane shows threshold and hysteresis values and the per-threshold assertion delays (in milliseconds). The middle lower pane shows the groups to which the sensor belongs.

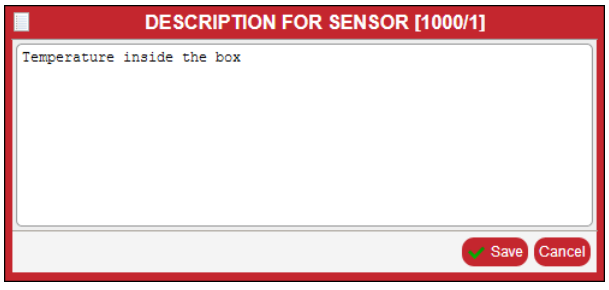
To change the name of the sensor, press the button “Change name”. The dialog for choosing the new name will open:



Type the new name and press the “Set” button; the sensor name will be changed.

To restore the default name of the sensor, check the “Set default” checkbox and press the “Set” button.

To change the human-readable description of the sensor, press the button “Description”. The dialog for setting the human-readable description will open:



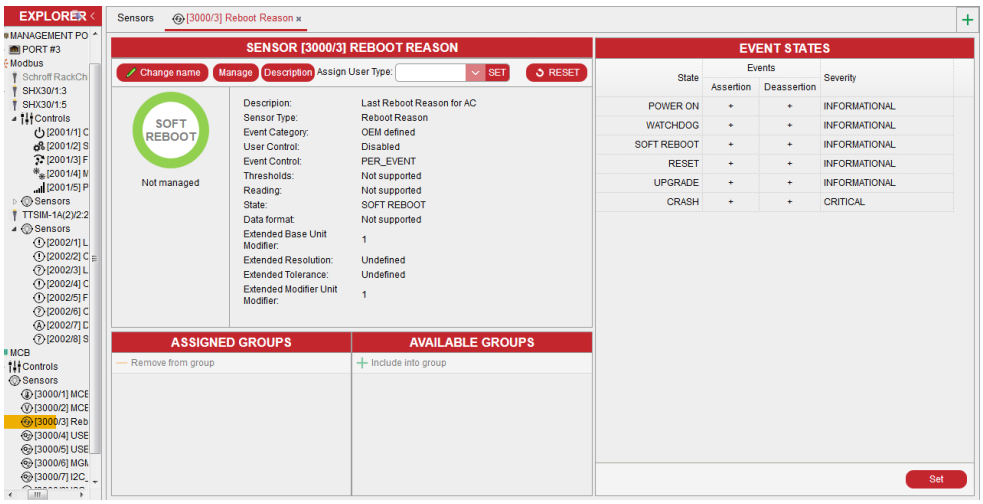
Type the new description and press the “Save” button; the human-readable description of the sensor will be changed.

To include the sensor to a group, check the correspondent checkbox in the “Groups” pane.

To change thresholds, hysteresis and the per-threshold assertion delays, change the corresponding values in the right pane and press the “Set” button in the right pane.

Managing discrete sensors:

When a discrete sensor is selected in the left pane, the middle upper pane shows the sensor state and properties, the middle lower pane shows the groups to which the sensor belongs, and the right pane shows supported event states, their event severities and event enables.



To change the name of the sensor, press the button “Change name”, as in the case of a numeric sensor.

To change the human-readable description of the sensor, press the button “Description”, as in the case of a numeric sensor.

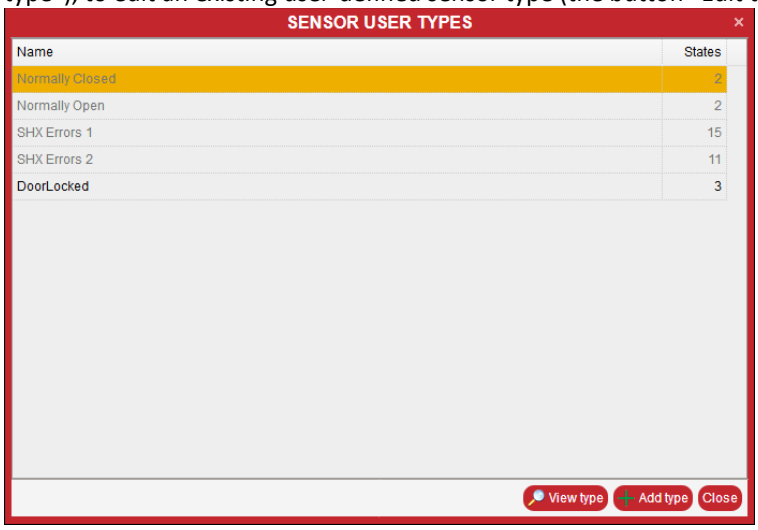
To change event enables and event severities, change the corresponding values in the right pane and press the “Set” button in the right pane.

To include the sensor to a group, check the correspondent checkbox in the “Groups” pane.

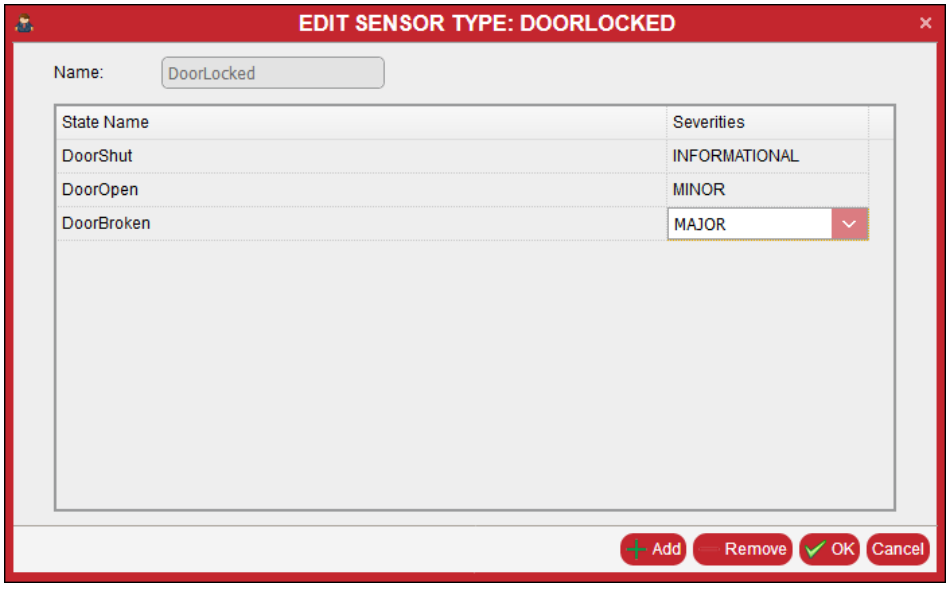
8.2.4 User-Defined Sensor Types

There are a number of built-in sensor types which are identified by small integer numbers and listed in Table 7. However, a user can define his/her own type for discrete sensors, in order to specify meaningful names to the states of the corresponding sensor and define severities appropriately. Then these user-defined sensor types can be assigned to sensors.
User-defined sensor types must have unique names that identify them.
They are also assigned numeric identifiers from a specially designated range, so that these types can be used in sensor events and other data structures where numeric sensor types are required.
There are four pre-defined sensor types: “Normally Closed”, “Normally Open”, “SHX Errors 1”, “SHX Errors 2”.
These types are included in the user-defined sensor types, but they can’t be edited or deleted. The sensor types “Normally Closed” and “Normally Open” may be assigned to a discrete sensor with two states. The sensor types “SHX Errors 1” and “SHX Errors 2” have 15 and 11 states, respectively.

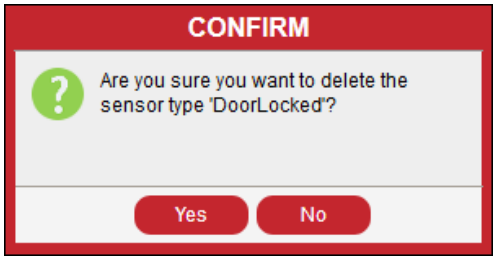
From the Web interface, to get the list of the user-defined sensor types, invoke the dialog box with the menu items “Device Settings” -> “Sensor User Types”. The dialog box “Sensor User Types” allows the user to create a new user-defined sensor type (the button “Add type”), to delete an existing user-defined sensor type (the button “Remove type”), to edit an existing user-defined sensor type (the button “Edit type”).



To edit an existing user-defined sensor type, press the “Edit type” button. The “Add” and “Remove” buttons in the “Edit Sensor Type” window allow to add and to remove named sensor states (and their severities), respectively. Severities are chosen from a drop-down box with a predefined list of values.



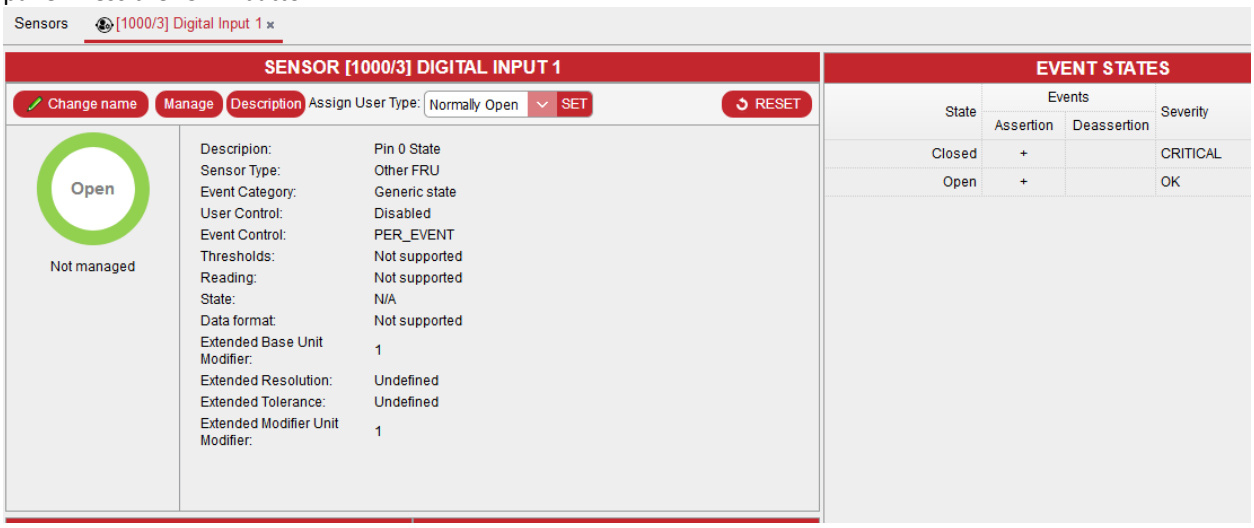
To delete an existing user-defined sensor type, press the “Remove type” button; a confirmation dialog appears that asks the user to confirm the deletion of the specific sensor type:



8.2.5 Assigning sensor types to sensors

It is possible to assign either a built-in type or a user-defined type to a sensor. In the case of a built-in type, the type is designated by the numeric identifier. In the case of a user-defined type, the type should be defined by its name. If the type is specified by its numeric identifier, the event category number is also specified, because the meaning of sensor states depends not only on the sensor type but on the event category as well.
Event category numbers are described in Table 9. For user-defined sensor types specified by name, the event category is set to the value $0x7E$ (Sensor-specific events).

To assign a user-defined sensor type to a discrete sensor, select this discrete sensor in the left pane, then choose a user-defined sensor type from a drop-down box with the list of all the user-defined sensor types in the middle upper pane. Press the “SET” button.



8.3 Controls

Controls represent the means to change state of some physical or logical objects programmatically. For example, a GPIO that controls the state of a door lock (open/closed) can be represented as a control in the HPI model. Another example of a control can be a PWM register that determines the speed of a fan.
A control has the following attributes:

- Control number, that identifies the control within the resource that owns it
- Human-readable control name
- Human-readable control description (the “Description” attribute is used in BACnet applications)
- Control type
- Output type: the type of physical control output, e.g. dry contact-closure, fan speed or LED
- Actual mode (automatic or manual) and state/value
- Default mode and state/value
- For analog controls, the allowed range of values.

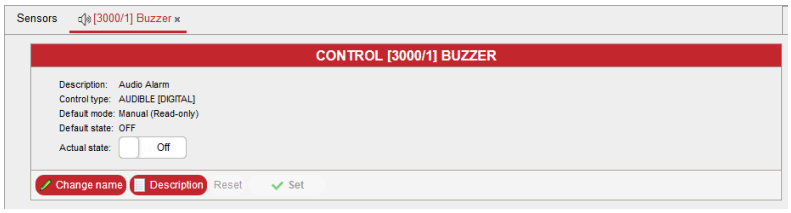
The following types exist for controls:

- Digital: these controls can be in one of the two states, On and Off. In addition, pulse operations (Pulse On and Pulse Off), may be supported for digital controls; these operations set the specific state for a control for a small period of time, and then return back to the previous state
- Discrete: these controls can be in one of several predefined states, which are specified by an integer enumeration
- Analog: these controls have a numeric (integer) value which can be set by the user
- Float analog: same as analog, but the value can be a floating-point number.

A control can be in one of the two modes: automatic and manual. In automatic mode, the control state or value is chosen automatically and the user can only read it. In manual mode, the user directly specifies the state or value for the control. Not all controls support automatic mode; a fan PWM control can be one example of a control supporting automatic mode.

8.3.1 Examples

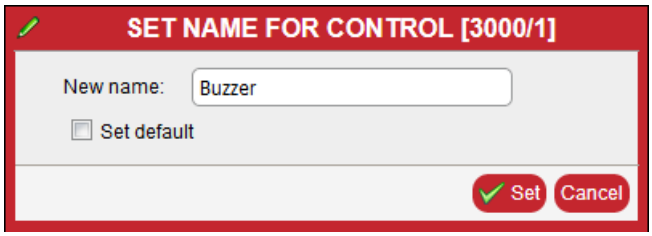
With the Web interface, controls and sensors are shown together in the list of instruments in the tree (left) pane below the corresponding resource. To manipulate a control, select it in the tree pane; the control management pane will be shown on the right:



The information shown on the screen about the control includes its number, name, description, type, output type, default mode, default state, actual mode and actual state or value.

To change the actual state/value of the control, enter the new value in the field “Actual value” or toggle the state in the field “Actual state” and press the “Set” button. The new value or state will be set.

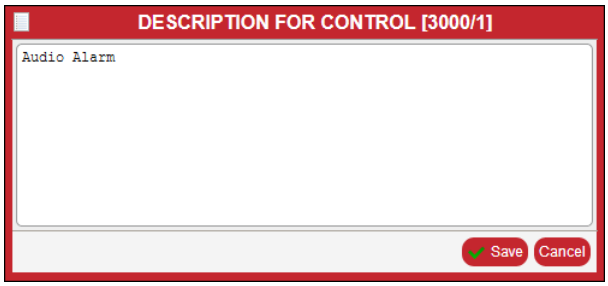
To change the name of the control, press the button “Change name”. The dialog asking for the new control name will appear:



Enter the new name for the control in the text box and press the “Set” button. The control name will be changed.

To restore the default name of the control, check the “Set default” checkbox and press the “Set” button.

To change the human-readable description of the control, press the “Description” button. The dialog asking for the new control description will appear:



Enter the new description for the control in the text box and press the “Save” button. The “Description” attribute is used in BACnet applications.

8.4 Events

Events represent the method for an HPI system to notify the environment about state and configuration changes in it. In Guardian Management Gateway, the subset of the whole HPI set of events is supported.

8.4.1 Event categories

Events, that a Guardian Management Gateway can generate, can be split in several categories:

Resource events

- Resource events are sent when:
  - Resource added: Event is sent when a new resource is added to the system (hot-inserted)
  - Resource removed: Event is sent when a resource is removed from the system (hot-extracted)
  - Resource updated: Event is sent when the population of instruments (sensors, controls, inventory) changes for a given resource

Sensor Events

- Sensor events are sent when:
  - For numeric sensors, when a sensor value crosses one of its thresholds; depending on the event enable mask, events can be sent when the sensor value goes beyond a threshold, returns back or both
  - For discrete sensor, when the sensor changes its state; also the event enable mask determines for which state changes the event is generated

Software Events

- Software events are sent by software when certain actions are initiated by the user or other software-related conditions occur; for example:
  - When a user logs in or logs out
  - When the Guardian Management Gateway connects to a wireless LAN or disconnects from a wireless LAN
  - When a specific server, which is being monitored, becomes reachable or unreachable
  - When the device establishes a MQTT connection with an IoT cloud or closes such a connection.

Upgrade Events

- Upgrade events are sent when a firmware upgrade takes place and indicate different phases of the upgrade process.

8.4.2 Event parameters

Events are data packets that have standardized format. Besides the type of the event, they carry parameters which vary depending on the event type.

All events include the following parameters:

- Event type
- Timestamp (when the event happened)
- Severity (can be one of Critical, Major, Minor, Informational or OK).

Resource Events

Resource events carry the resource ID as the only additional parameter; this resource ID identifies the resource that has been added, removed or changed its instrument population.

Sensor events

For sensor events, the following additional parameters are provided:

- Resource ID and sensor number, for the sensor that originated the event
- Sensor type (e.g. temperature, voltage, humidity, etc.)
- Event category (one specific event category is threshold crossing)
- Is the event condition asserted or deasserted?
- For threshold-crossing events on numeric sensors:
  - Which threshold has been crossed?
  - The sensor value that triggered the event
  - The value of the threshold that has been crossed
- For sensor state change events on discrete sensors:
  - A single state being asserted or deasserted that triggered the event
  - The current state mask of the sensor

Software events

Software events contain the following additional parameters: the specific event type (e.g. “user logs in”) and a text string that describes the event, in a human-readable form.

Upgrade events

Upgrade events carry the enumeration that identifies the current stage of the upgrade process, as the only significant additional parameter.

8.4.3 Event processing

After being generated, the event passes through event filters that may initiate certain actions based on the event type and values of event parameters.
Configuration of event filters and actions is discussed in the section 18 Events and Actions.

Finally, the event is stored in the System Event Log on the Guardian Management Gateway where it can be examined later.

8.5 Inventory

An inventory contains information about a resource in a special structured format. The format used for Guardian Management Gateway is the IPMI FRU Information format, described in [2]. Information is represented in several standard sections, followed by a number of OEM-specific records of variable length.

The following standard sections, defined in the IPMI FRU Information format, are used with Guardian Management Gateway resources:

- Board information area. This section contains information about the hardware aspects of the resource, including date and time of manufacturing, manufacturer name, board product name, part number and serial number
- Product information area. This section contains information about the general aspects of the resource, or about the resource as a separate product; it includes manufacturer name, product name, product version, part number, serial number and optional asset tag.

The chassis information area and the internal use area, also defined in the IPMI FRU Information format, are not used in the Guardian Management Gateway.

The following OEM-specific variable-length records are used with Guardian Management Gateway resources (the OEM is nVent/Schroff for all records):

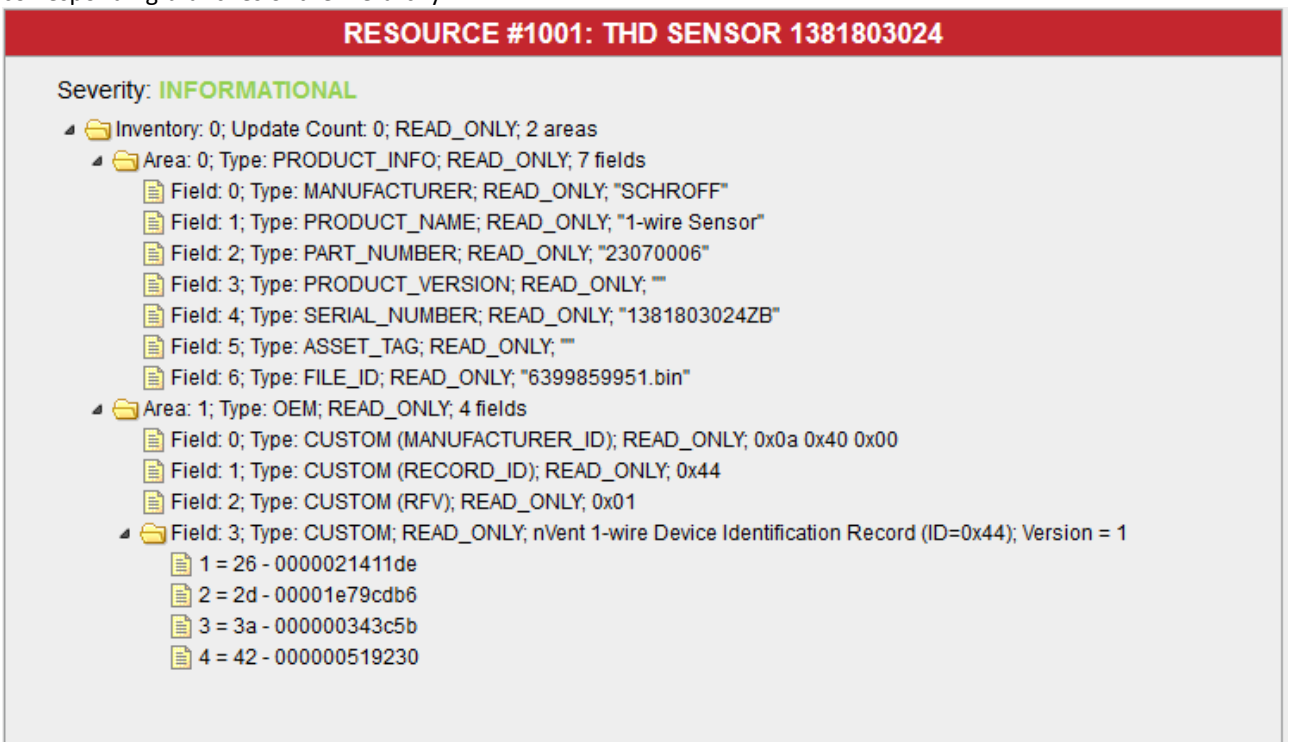
- Guardian Management Gateway configuration record; this record describes general configuration of power-related aspects of the Guardian Management Gateway
- LCD calibration parameters record; this record contains calibration parameters of the LCD screen
- Sensor device identification record; this record identifies the components of a specific Schroff sensor device.

For the specific types of Guardian Management Gateway resources, inventory contains the following areas and records:

- Managed sensors resource (0): the corresponding inventory describes the Guardian Management Gateway as a whole. It contains the board information area, the product information area and the Guardian Management Gateway configuration record
- Schroff environmental sensor resources（1000 - 1999）: the inventory contains the product information area and the sensor device identification record (no board information area is included because of the limited inventory size on these devices)
- MCB（3000）: the inventory contains the board information area, the product information area and the LCD calibration parameters record

From the user perspective, inventory is always read-only; only read access is provided to it.

With the Web interface, a user can view the inventory on the global resource, sensor devices and on the MCB. To do that, point to the corresponding item in the tree (left) pane; the inventory will be shown in the hierarchical format in the pane in the middle of the screen. Use arrows to the left of the hierarchical items to open and close the corresponding branches of the hierarchy.



9 Managed Sensors

Managed sensors allow a user to designate a subset of the whole set of sensors in the Guardian Management Gateway (which may be quite large), work with this subset and apply additional management actions to it.

The sensors in this subset are mapped to resource 0 ("Managed Sensors") and are allocated available sensor numbers on that resource. They are still available at their original resource number and sensor number; sensor numbers on resource 0 are just aliases.

One possible use of managed sensors is to collect most important sensors in one place to be able to manage them efficiently (since a Guardian Management Gateway, depending on configuration, may have hundreds and even thousands of different sensors).

9.1 Features of managed sensors

Sensors on resource 0 can have the following additional attributes compared to regular sensors:

- User-defined sensor name, as an arbitrary text string
- Description, as an arbitrary text string
- User-defined sensor type and subtype, as arbitrary text strings (in addition to normal HPI sensor type that is defined for all sensors)
- Location attributes in the form of X, Y and Z coordinates, in the form of arbitrary text strings (for the Z coordinate, a numeric representation can be chosen).

The attributes listed above are opaque for the Guardian Management Gateway but are kept persistent across reboots. These attributes are associated with the sensor representation on resource 0 (that is, they are not visible if the sensor is accessed by its original resource number and sensor number).

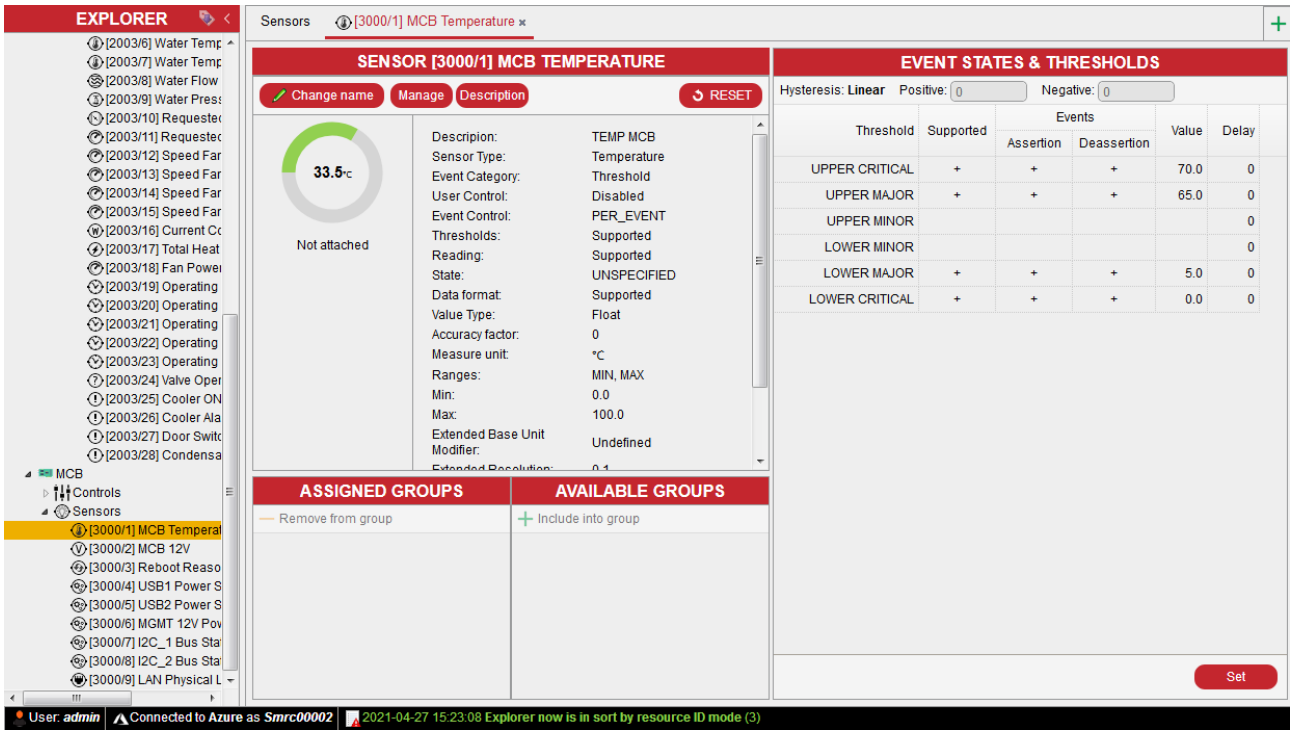
In addition, sensor logging applies to managed sensors. Sensor logging involves periodic polling of managed sensors and calculating some statistics for the sensor values over time.

9.2 Attaching and detaching managed sensors

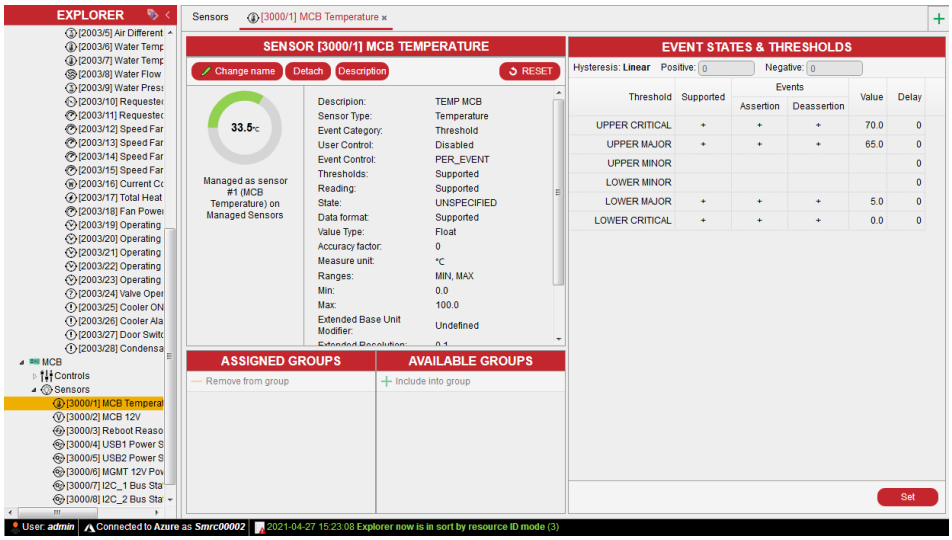
To make sensor a managed sensor, a user should first choose the actual sensor, by resource and sensor number, and then map (attach) it to resource *0*. Sensor mapping is persistent across reboots. If a sensor belongs to a hot-swappable resource, the mapping is also automatically restored when the resource is hot-inserted (if the resource was previously hot-extracted or was not present during the initial start).

With the Web interface, attaching and detaching can be performed in the following way:

To create a managed sensor, find the actual sensor in the tree pane on the left side on the screen and click on it. The sensor information pane contains information whether the sensor is managed and, if the sensor is not attached yet, the “Manage” button.



Press the “Manage” button to attach the sensor. The sensor information pane changes to reflect that the sensor is now managed and shows the managed sensor number; the “Manage” button becomes unavailable:



The list of managed sensors is shown in the tree pane on the left side of the screen, under “Managed Sensors”:

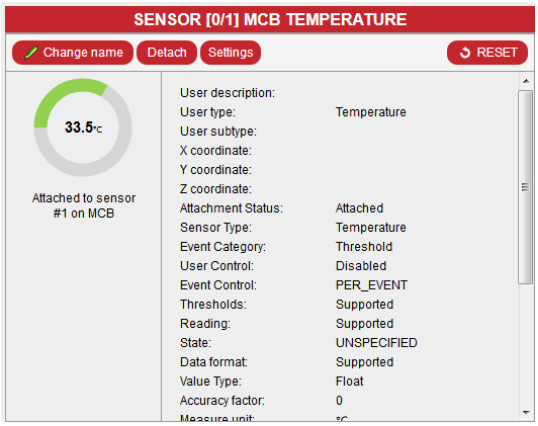
In the sensor information pane for a managed sensor, the title line shows information about the original sensor.

To delete a managed sensor, press the “Detach” button in the title line of the sensor information pane, either for the managed sensor on resource *0*, or for the actual sensor. The corresponding managed sensor on resource *0* will disappear.

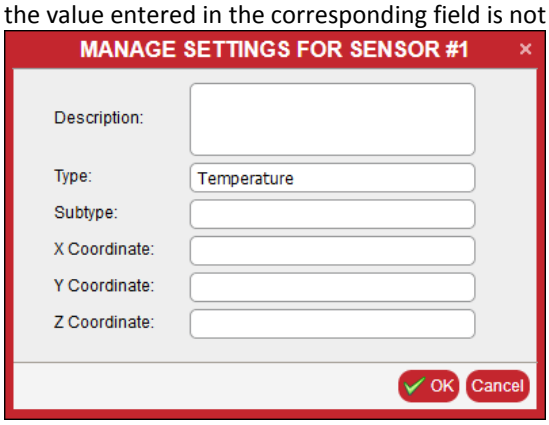
Unlike CLI, in the Web interface there is no way to attach a sensor to a specific managed sensor number on resource *0*.

9.3 Managing attributes of managed sensors

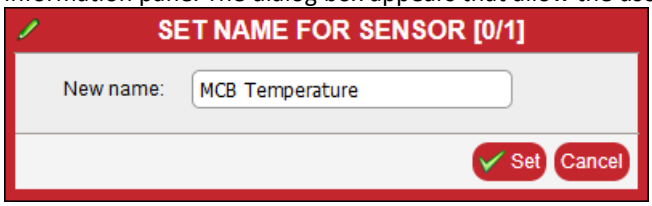
To view and modify managed sensor attributes (except the user-defined sensor name) with the Web interface, press the “Settings” button in the title bar of the sensor information pane.



The “Manage settings” dialog appears. The user can view and edit the text of sensor attributes and press the “OK” button to save the values. Web interface automatically removes the numeric restriction for the Z coordinate value, if the value entered in the corresponding field is not numeric.



To change the managed sensor name with the Web interface, use the same mechanism as with a regular sensor: choose the target sensor in the left tree pane and press the “Change name” button in the title bar of the sensor information pane. The dialog box appears that allow the user to edit the sensor name and save changes.



The “Set default” operation is not supported for managed sensors.

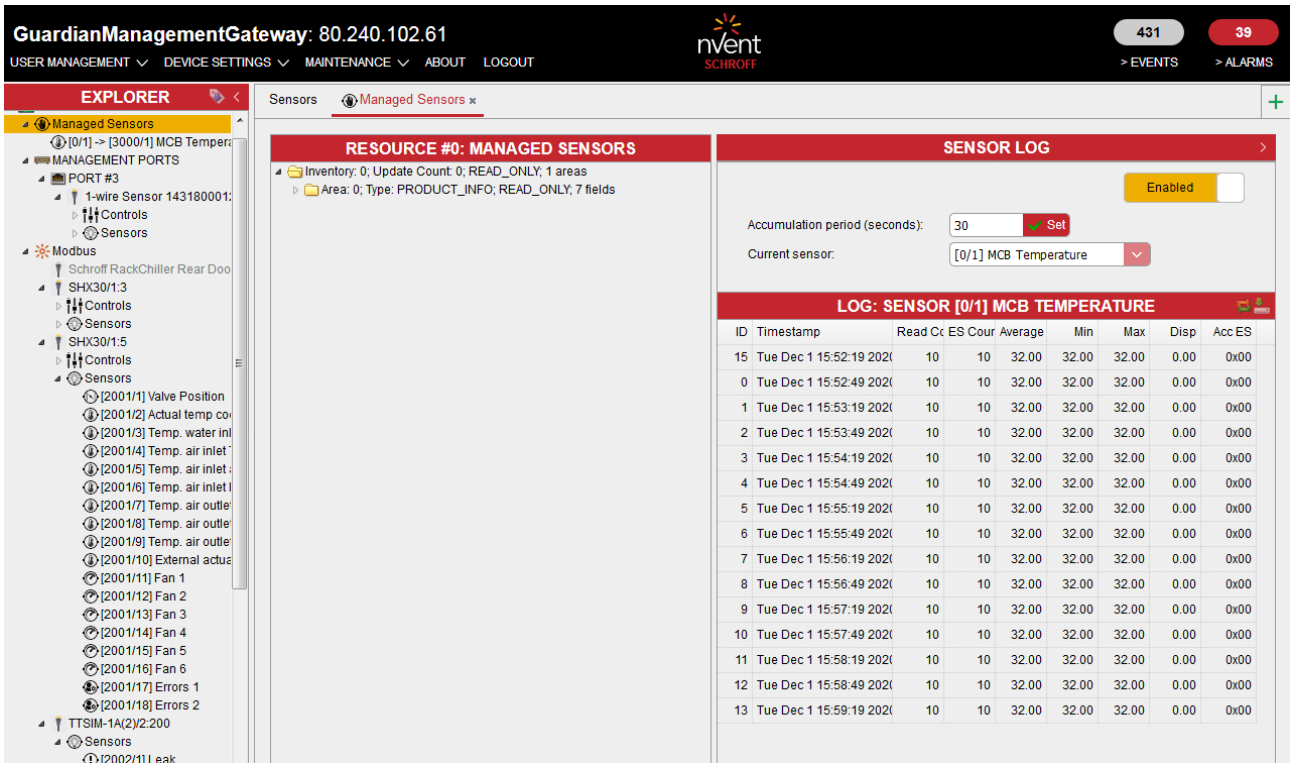
9.4 Logging for managed sensors

The logging facility for managed sensors implements periodic polling and accumulation of sensor values. It works as follows:

- The logging period is defined, during which managed sensor values are accumulated; the typical duration of this period is about 30 seconds and this value is configurable for all sensors
- During this period, the sensors are periodically polled; the typical period is about 3 seconds but can be configured separately for each sensor (this is the configuration parameter “Polling period” which is defined for all sensors, not just managed sensors)
- After the period is finished, the following values are calculated for each managed sensor:
    - number of polls during the period
    - number of polls in which the sensor reported a value (the sensor could return the condition “sensor reading unavailable” during some polls)
    - average sensor value for the period
    - minimal sensor value for the period
    - maximal sensor value for the period
    - dispersion of the sensor value during the period
    - accumulated event state mask during the period (it includes all sensor states that were detected during the period)
- These values are stored in an entry of a ring buffer; there is a separate ring buffer for each managed sensor and the number of entries in each buffer is fixed to 16 entries. When all entries are filled in, the buffer wraps around.

The logging facility can be enabled and disabled by the user.

To get access to the managed sensor log facility with the Web interface, choose the “Managed Sensors” in the left tree pane; the “Sensor Log” pane will be on the right side of the screen. Choose the sensor from “Current Sensor” combo-box to see the sensor log for a specific managed sensor. Also, the controls for changing the accumulation period and enabling/disabling sensor log exist on this page.

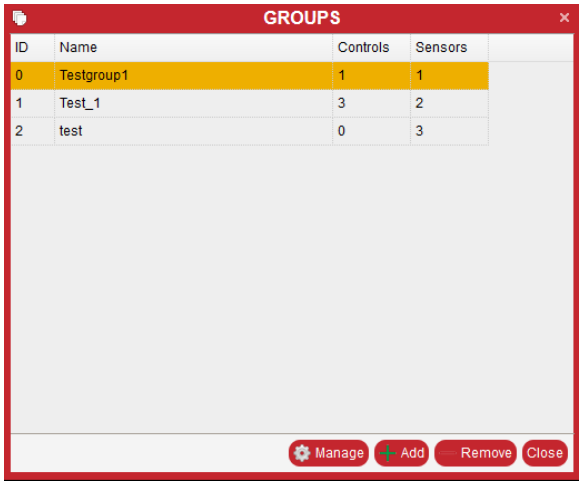


10 Group Operations on Controls and Sensors

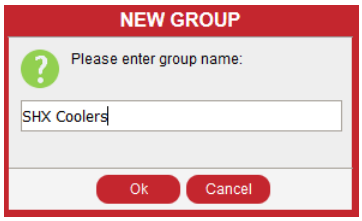
It is possible to group controls and sensors into larger entities and perform group operations on them.
For controls, the group operation is setting all controls in the group to the same state. For example, for digital controls this can involve setting all controls to the state *ON* or to the state *OFF*. For analog or discrete controls, all controls are assigned the same value.
For group control operations, it is possible to specify the order of processing of controls and, for each control, a delay after processing that control.
A group control operation can be executed synchronously or asynchronously. For synchronous execution, the caller waits until the operation is complete. However, since a group control operation may take a long time, asynchronous execution is also supported. In that case, the caller does not wait until the execution is complete, but gets control back immediately after execution is started and can then poll for the progress of the execution.
For sensors, the group operation can be setting thresholds to all sensors in the group to the same values, or calculating some aggregate value over the values of sensors comprising the group. The supported aggregates include sum, average, dispersion, minimum and maximum values, and others. For example, if homogeneous sensors from several resources are grouped together, it is possible to calculate an average value from the values of these sensors.
To facilitate group operations, a special named entity (a group) is created by a user, and then, sensors and/or controls are added to that entity. The order of adding controls to the group corresponds to the order in which controls are processed for a group operation. The order of adding sensors to the group is not significant.
It is possible to have separate groups for working with sensors and controls or use one group to work with several sensors and several controls. Operations with sensors and controls in one group are handled independently.
The following operations with groups are supported:

- Create a new group and assign a name to it
- Delete an existing group by name
- Get the list of existing groups
- Add a control to the group, specify the delay after processing this control
- Delete a control from the group, by control number or by position
- List all controls and sensors in the group
- Assign a state to all controls in the group, synchronously
- Assign a state to all controls in the group, asynchronously
- Show progress of an asynchronous group control operation
- Cancel an asynchronous group control operation
- Add a sensor to the group
- Delete a sensor from the group, by sensor number or by position
- Set thresholds to certain values to all sensors in the group
- Calculate an aggregate over all sensors in the group.

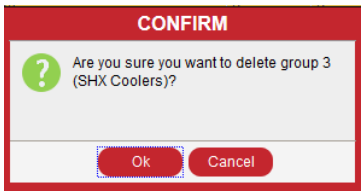
From the Web interface, to get the list of the groups , invoke the dialog box with the menu items “Maintenance” -> “Sensor/Control Groups”. The dialog box “Groups” allows the user to create a new group (the button “Add”), delete an existing group (the button “Remove”), edit a group or perform a group operation on it (the button “Manage”).



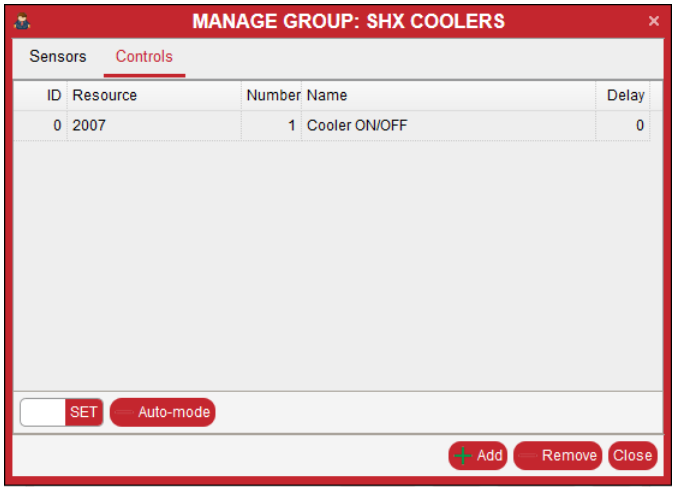
To create a new group, press the “Add” button; a “New group” dialog appears that asks the user for the name of the new group that should be unique:



To delete a group, press the “Remove” button; a confirmation dialog appears that asks the user to confirm the deletion of the specific group:

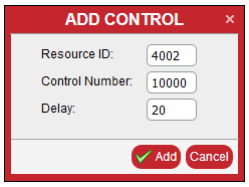


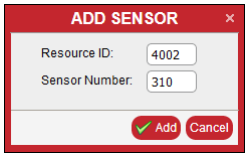
To edit a group or perform a group operation on it, press the “Manage” button. The “Manage group” window appears that contains two tabs: “Sensors” and “Controls”.



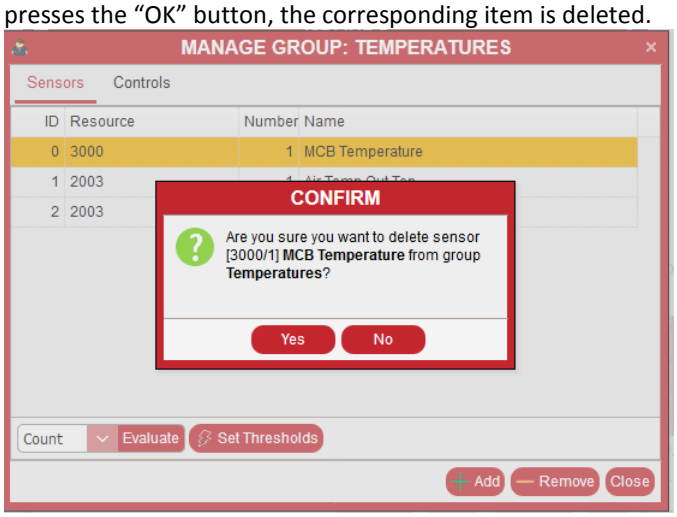
On each of the tabs, there is an “Add” button that allows a user to add the corresponding instrument to the group, and the “Remove” button that allows the user to remove the selected instrument from the group.

When an “Add” button is pressed, a “new instrument” dialog shows up that asks the user for the identification and parameters of the new instrument. For a control, these are the resource ID, the control number and the mandatory delay after setting the control state (in milliseconds). For a sensor, these are the resource ID and the sensor number.

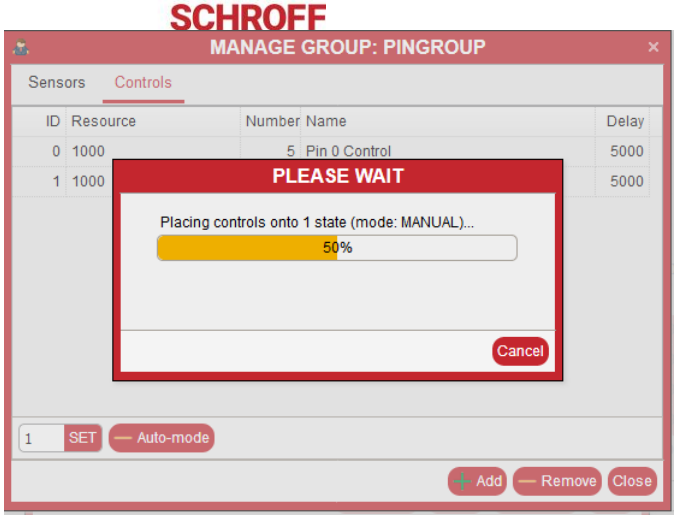




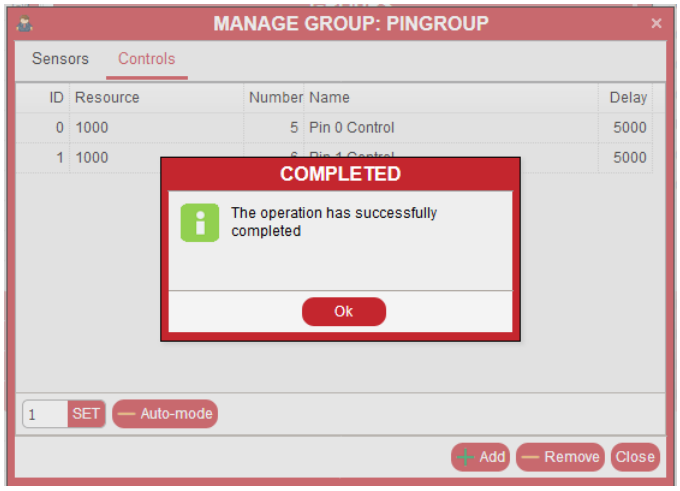
When a “Remove” button is pressed, and a group item is selected, a confirmation dialog shows up, and if the user presses the “OK” button, the corresponding item is deleted.



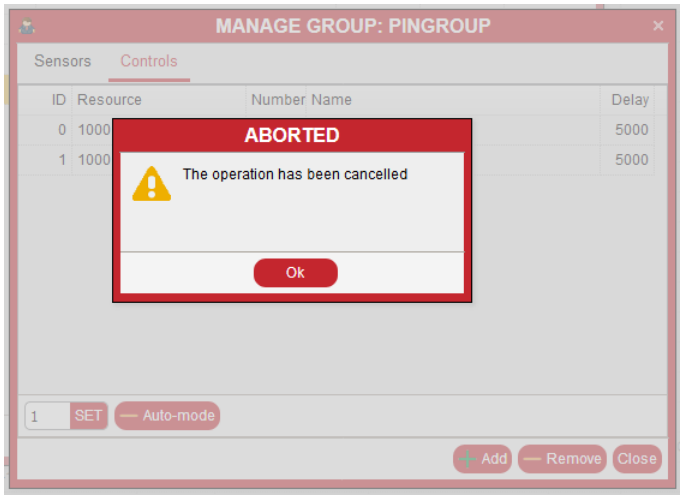
To set all controls in the group to the same state, select the “Controls” tab, enter the numeric state in the edit field in the left-bottom corner, and press the “SET” button.
For digital sensors, use the number *0* for the “Off” state, *1* for the “On” state, *2* for the “Pulse Off” state, and *3* for the “Pulse On” state.
To set all controls in the group to Automatic mode, press the “Auto-mode” button.
In both cases, an asynchronous operation is started and a progress bar indicating the progress of the operation is shown. While the operation is in progress, it can be cancelled by pressing the “Cancel” button.



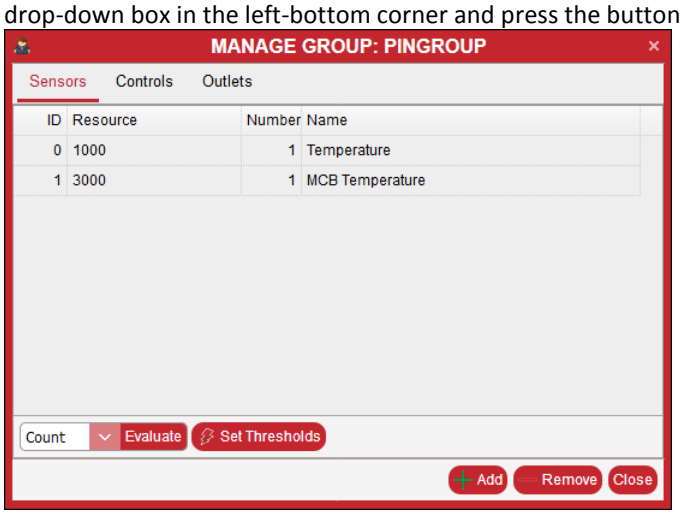
After the operation is completed, a message box indicating the successful completion is shown. Press the “OK” button to dismiss it:



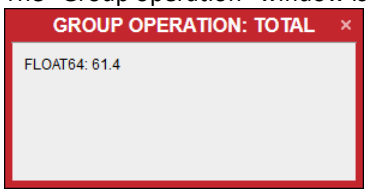
If the user presses the “Cancel” button, the operation is cancelled and the corresponding message box is shown; also pressing the “OK” button will dismiss it.



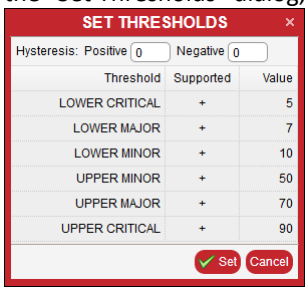
To initiate an aggregate operation on all sensors in a group, select the “Sensors” tab, choose the operation in the drop-down box in the left-bottom corner and press the button “Evaluate”.



The “Group operation” window is generated that shows the result:



To set a threshold (or hysteresis) value for all sensors in a group press the button “Set threshold”, fill in the fields in the “Set Thresholds” dialog, and press the “Set” button.



11 Users, Roles and Privileges

A list of valid users exists for a Guardian Management Gateway. A valid user can log in to some upper Guardian Management Gateway interface: Web, CLI (with SSH) or SNMP. The list of Guardian Management Gateway users is integrated with the list of users in the underlying Linux.
With respect to user permissions, the role-based model is used.
Each user is associated with one or more roles (e.g. “administrator”, “normal user” or “power user”). Each role has a set of privileges associated with it. A user possesses a certain privilege if at least one role this user is associated with, has this privilege.
For each user, the following attributes are defined:

- User name (used as a logon name)
- Password (not stored explicitly, but only as a hash)
- Full user name
- User phone number
- User e-mail address
- “User enabled” flag
- The list of roles associated with the user
- Preferred measurement units
- Web session preferences
- SSH public key
- SNMPv3 settings
- Preferred interface language (English, German, French)

External users are the users not listed in the list of valid users but that can be authenticated using external means (e.g. via LDAP). For such users, the entry in the user list is created during first logon. The role “ExternalUserRole” is used and other attributes are defined as empty strings. These attributes can later be redefined by an administrator.
For each role, the following attributes are defined:

- Role name (used in the user attributes)
- Role description (arbitrary text)
- Privilege mask (a bit mask where bit 1 means the corresponding privilege is included).

The following privileges are currently defined (this set of privileges may be extended in future versions of the firmware):

	PRIVILEGE	DESCRIPTION (WEB)
1	Common user	The default privilege. View the network configuration, the network services configuration, the Event Filters and Periodic Rules, The Reachability table, the Alarm Table, the Sensor User Types, own “User Preferences”, the “About” information and the communication error log. Write own “User Preferences”: language, measurement units, web session parameters.
2	Administrator	The maximum privilege.
3	Change authentication settings	Not used.
4	Change date/time settings	Write Date/Time settings.
5	Change EMD configuration	Not used.
5	Change event settings	Create/delete/edit a filter, a periodic rule, a named action list. Delete an alarm. Execute or verify an expression.
6	Change external sensor configuration	“Write” access on sensor’s detail page: thresholds, state severities, state masks, sensor name, “Assign Sensor Type”. Reset a sensor to its default settings. Write “Z Coordinate”, “Transient Alarm Severity” on “Device Settings” page. “Remove resource”, “Set Resource Severity”, and “Description” for resources sensors and controls. Discovery of Modbus devices.
7	Change SHX and other Modbus devices configuration	Write Modbus configuration. Write to Modbus controls.
8	Change network	Write the network service’s configuration: IPv4, IPv6, HTTP, HTTPS, SSH, SMTP,

	configuration	Telnet, NTP, IoT, BACnet, RedFish. Write Asset Tag, Location and Device Description attributes, Device Name. Write the Reachability table.
9	Change own password	Change own password, SSH public keys.
10	Change security settings	Write the LDAP configuration, firewall configuration, role-based firewall configuration. SSL certificate creation, SSL certificate installation. Login restrictions, Restricted service agreement. Write to Login Settings and Password Requirements.
11	Change SNMP settings	Write the SNMP configuration: Read/Write community string, SysLocation, SysName, SysContact, default trap destination.
12	Change user settings	Create, delete a user or role. Change user attributes, roles, global configuration (language, measurement units, web parameters, debug level, LCD UI settings). "Transient Alarm Severity", "Asset Tag", "Location", "Device Description" are not covered.
13	Change Webcam settings	Not used.
14	Change Power configuration	Not used in the Guardian.
15	Clear event log	Clear the SEL.
16	Firmware update	Configuration Import, Firmware Update.
17	Perform reset (warm start)	Restart, reboot.
18	View event settings	Not used
19	View event log	Show the SEL content.
20	View security settings	View Login Settings and Password Requirements, LDAP configuration, firewall configuration, role-based firewall configuration.
21	View SNMP settings	View the SNMP configuration: Read/Write community string, SysLocation, SysName, SysContact, default trap destination.
22	View user settings	View the settings of users: name, text attributes (full name, e-mail, phone), "locked" status, roles, language, SNMPv3 status and protocols. View the roles.
23	View Webcam settings	Not used.
24	View Power configuration	Not used in the Guardian.
25	Use groups	Perform group operations on sensors and controls. For example, evaluating of aggregated value of a group of sensors, or "turn off" operation of a group of Modbus devices.
26	Change group's configuration	Create/delete Sensors/Control groups. Add sensor/control to the group and delete sensor/control from the group.
27	Perform factory reset	Perform factory reset.
28	Alarm acknowledgement	Enable alarm acknowledgment.
29	Export configuration	Export configuration.

Some privileges are not used in the current version of the Guardian, and may be used in future versions. Some privileges are reserved for other products based on the SGP and are not applicable to the Guardian.
If a "write" privilege is associated with a role, it is convenient to add the correspondent "view" privilege to the role. For example, if the "Change user settings" privilege associated with the role, then the "View user setting" privilege should be associated with the role, too
On the first start of a Guardian Management Gateway, or after a factory reset, the following default list of users and roles exists on the Guardian Management Gateway:

- User "admin", password "admin", enabled, list of roles consists of one role "AdministratorRole".
- User "user", password "user", disabled, list of roles consists of one role "UserRole".
- User "guest", password "guest", disabled, list of roles consists of one role "ReadOnlyUserRole".
- Role "AdministratorRole": includes all the privileges.
- Role "UserRole": includes the following privileges: Change Date/Time, Change Event Setup, Change External Sensor Configuration, Change LHX Configuration, Change Network Configuration, Change Own Password, Clear Event Log, Firmware Update, Perform Reset, View Event Setup, View Event Log, View Security Settings,

View SNMP Settings, View User Settings, Use Groups, Change Group Configuration, Acknowledge Alarms, Export Configuration.

- Role “ReadOnlyUserRole”: includes the following privileges: View Event Setup, View Event Log, View Security Settings, View SNMP Settings, View User Settings, Use Groups, Acknowledge Alarms, Export Configuration.
- Role “ExternalUserRole”: includes the following privileges: Perform Reset, View Event Setup, View Event Log, View Security Settings, View SNMP Settings, View User Settings, Use Groups, Acknowledge Alarms, Export Configuration. This role is assigned by default to new external users.

This configuration of roles and users can be subsequently changed by a user having the Change User Settings privilege.

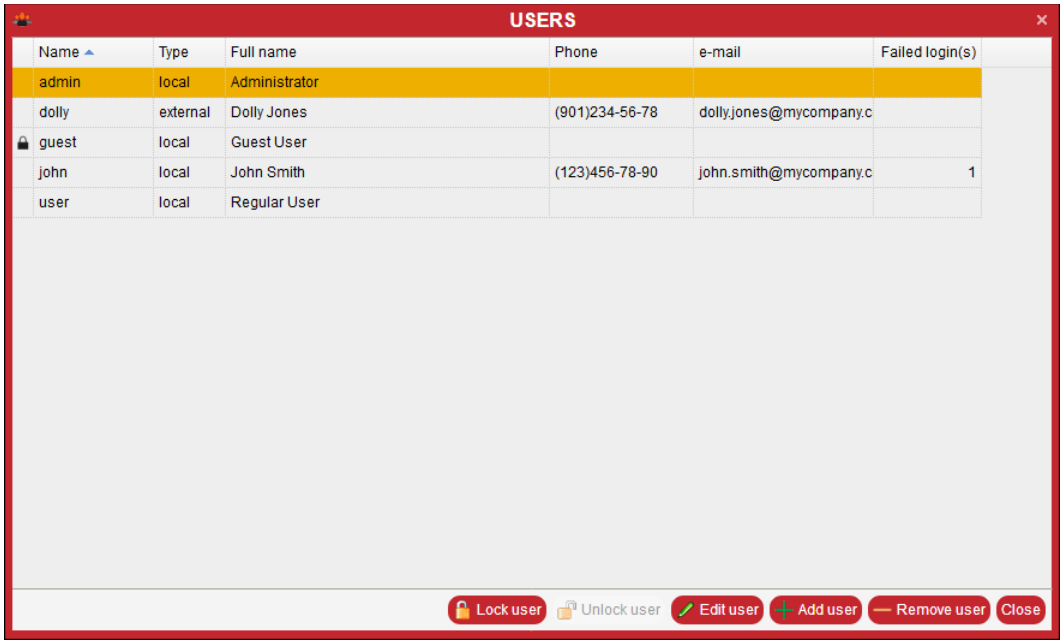
The following management operations are defined for the users:

- Get user information by index. The returned information includes all user attributes except the password and the list of roles. This operation allows enumerate existing users.
- Get user information by user name
- Create a user, specifying user name, password, attributes and the list of roles
- Delete a user by user name
- Change user attributes by user name
- Change user password (by user name or for the current user)
- Set user SSH public key by user name
- Get SNMPv3 attributes for a user by user name
- Set SNMPv3 attributes for a user by user name
- Get list of roles for a user by user name
- Set list of roles for a user by user name
- Get the mask of privileges that user has, by user name
- Get role information by index. The returned information includes role name, description and the privilege mask. This operation allows enumerate existing roles.
- Get role information by role name
- Create a role
- Delete a role by role name
- Change role information by role name
- Add privileges to the existing role, by role name
- Remove privileges from the existing role, by role name.

11.1 Create a new user

	Before you create a new user, check the privileges of the existing roles. If the privileges of the existing roles are not suitable for the new user, first create a new role.

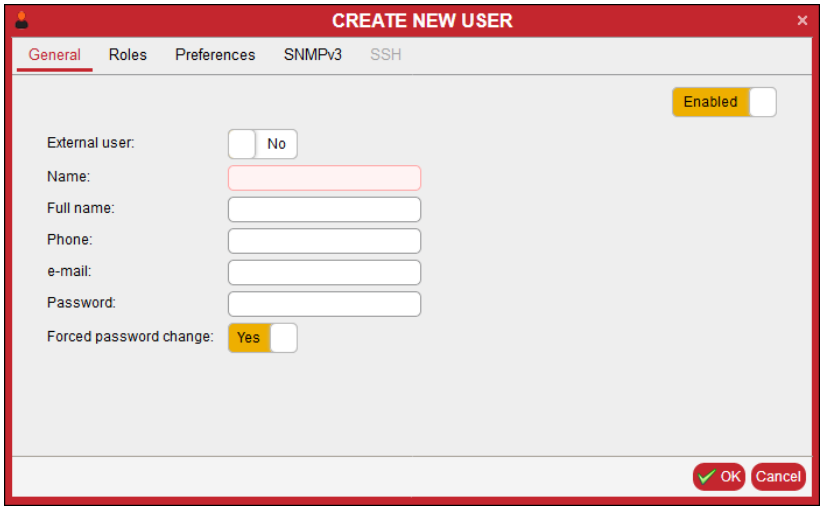
From the Web interface, to create a new user, invoke the dialog box with the menu items “User Management” -> “Users” and press the button “Add user”.

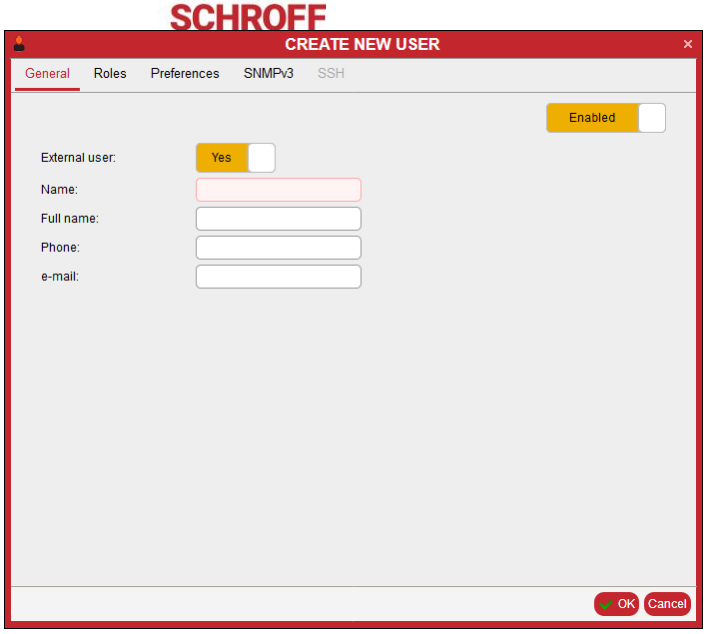


There are four tabs in the “Create new user” window: “General”, “Roles”, “Preferences”, “SNMPv3”.

The field “Name” is mandatory; the field “Password” is mandatory if the field “External user” is set to “No”. If the field “External user” is set to “Yes”, the field “Password” is not available. Other fields are optional on the “General” tab.

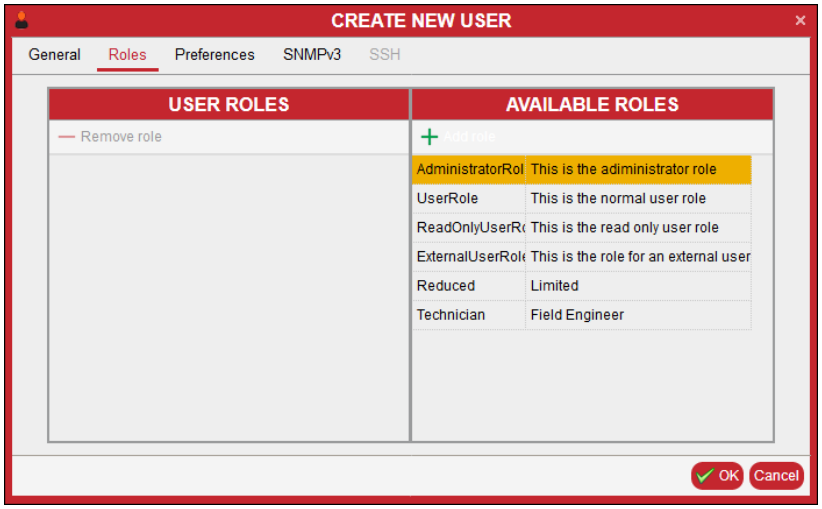
	The **Name** shall only consist of lowercase letters without spaces. The **Password** must have sufficient complexity, namely, it should be 8 characters or longer, include at least one lowercase letter, one uppercase letter, one digit and one special character.




CREATE NEW USER
General
Roles
Preferences
SNMPv3
SSH
Enabled
External user:
Yes
Name:
Full name:
Phone:
e-mail:
OK
Cancel

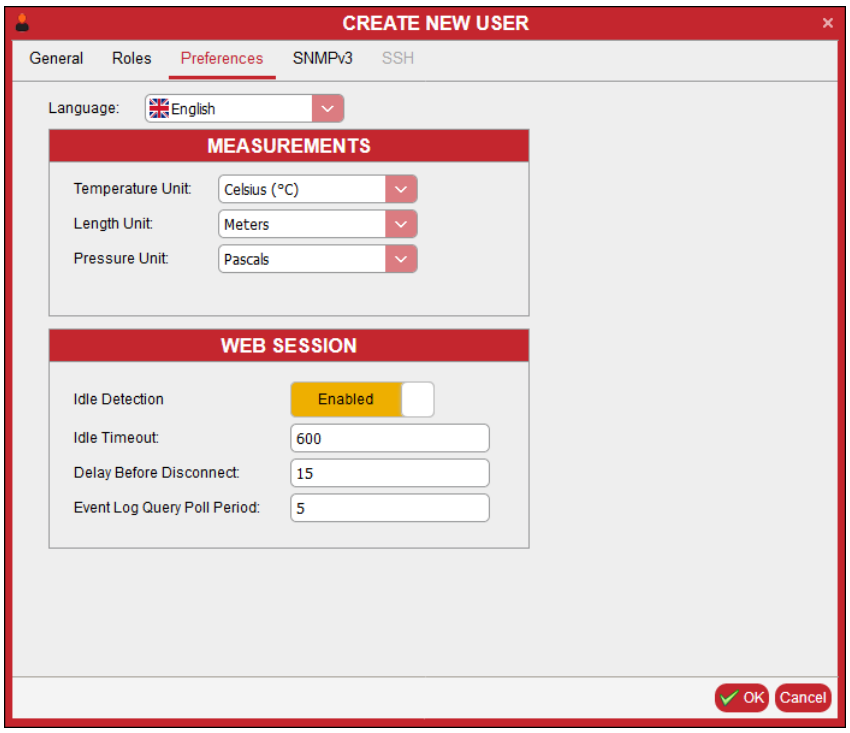
11.2 Set roles for a new user

To set roles for a new user use the “Roles” tab. There is no default role.

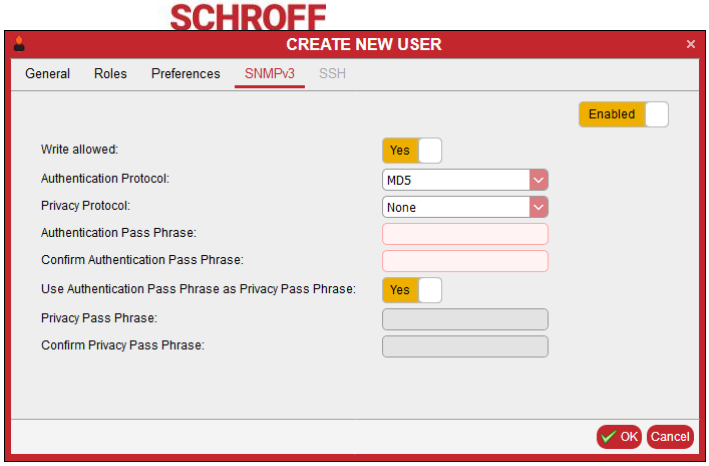


11.3 Set preferred measurement units

To set preferred measurement units for a new user use the “Preferences” tab. The default preferred measurement units are suggested.



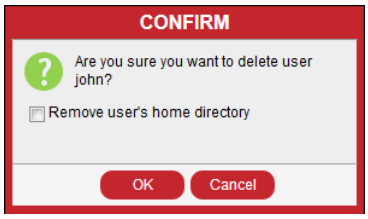
Optionally, SNMPv3 attributes may be set for a new user in the “SNMPv3” tab if the SNMP service is enabled.



When all the attributes of a new user are set, press the “OK” button and a new user will be created.

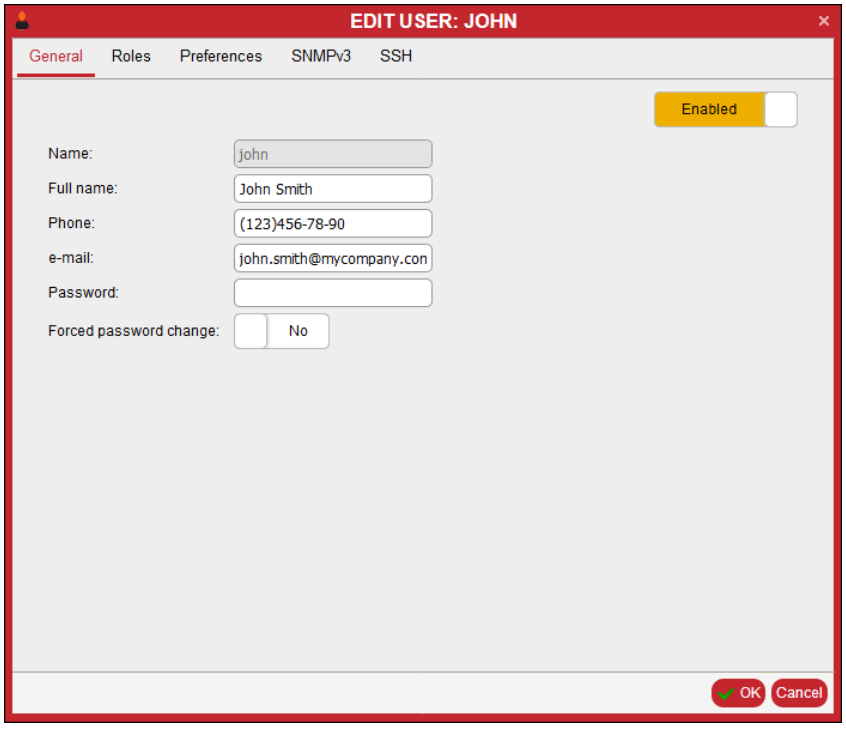
11.4 Delete an existing user

From the Web interface, to delete an existing user, invoke the dialog box with the menu items “User Management” -> “Users”, move the cursor to the correspondent line and press the button “Remove user”. The “Confirm” window with two buttons, “OK” and “Cancel”, is generated. If the “Remove user’s home directory” checkbox is checked, the user’s home directory is deleted. In the other (default) case, the directory remains intact.



11.5 Edit an existing user

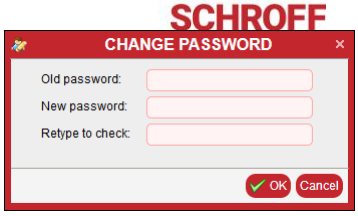
From the Web interface, to edit an existing user, invoke the dialog box with the menu items “User Management” -> “Users”, move the cursor to the correspondent line and press the button “Edit user”. There are 5 tabs in the “Edit user” window: “General”, “Roles”, “Preferences”, “SNMPv3”, “SSH”. The field “Password” is mandatory, other fields are optional. The field “Password” and the tab “SSH” are not available for an external user.



11.6 Lock a user

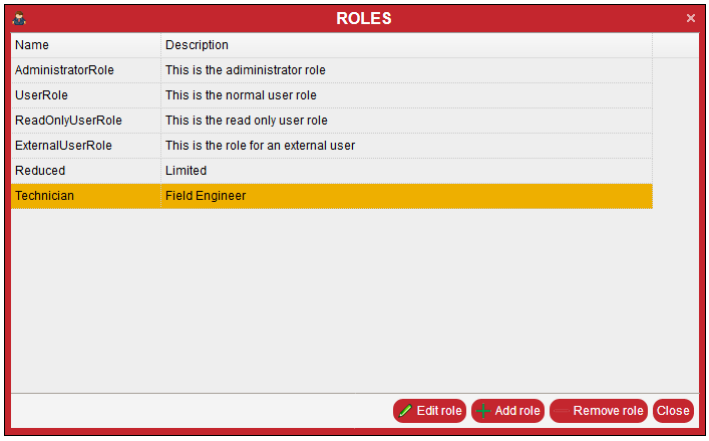
To lock a user, invoke the dialog box with the menu items “User Management” -> “Users”, move the cursor to the correspondent line and press the button “Lock user.” To unlock a user, invoke the dialog box with the menu items “User Management” -> “Users”, move the cursor to the correspondent line and press the button “Unlock user.”

To change own password, invoke the dialog box with the menu items “User Management” -> “Change Password.”


CHANGE PASSWORD
Old password:
New password:
Retype to check:
OK
Cancel

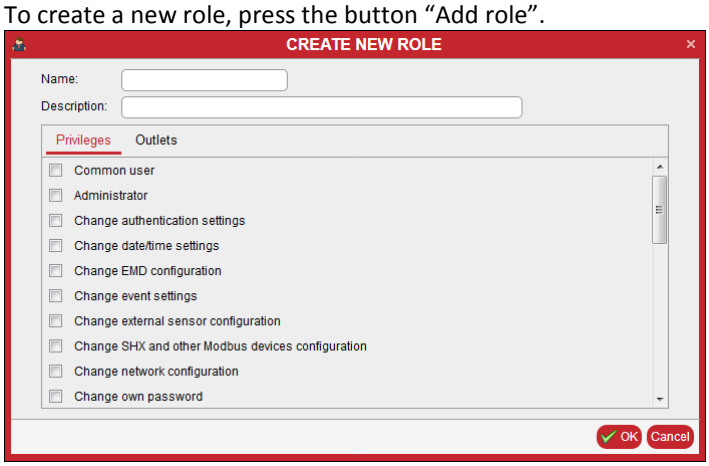
11.7 Get the list of the roles

From the Web interface, to get the list of the roles, invoke the dialog box with the menu items “User Management” -> “Roles”.



11.8 Create a new role

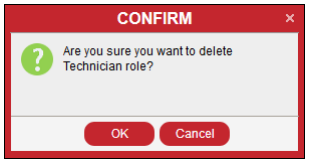
To create a new role, press the button “Add role”.



Then, fill the fields “Name” and “Description”. For every privilege there is a corresponding checkbox in the window. Check boxes that correspond to the set of privileges of the role and press the button “OK”.

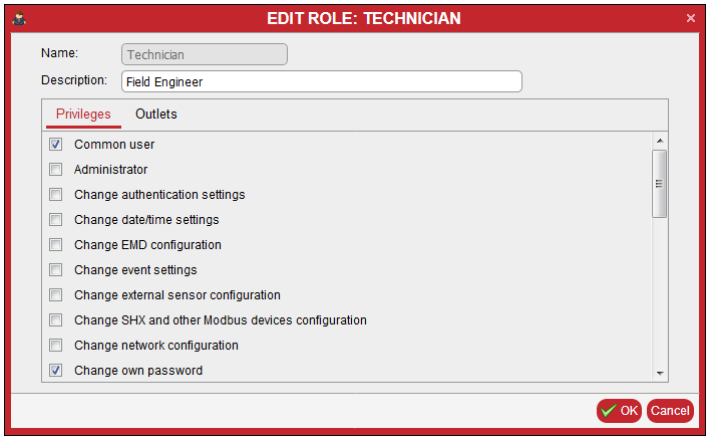
11.9 Delete an existing role

To delete an existing role, move the cursor over the corresponding line and press the button “Remove role”. The “Confirm” window with two buttons, “OK” and “Cancel”, is generated.



11.10 Edit an existing role

To edit an existing role, move the cursor over the corresponding line and press the button “Edit role”.



Change the description and the set of privileges of the role and press the button “OK”.

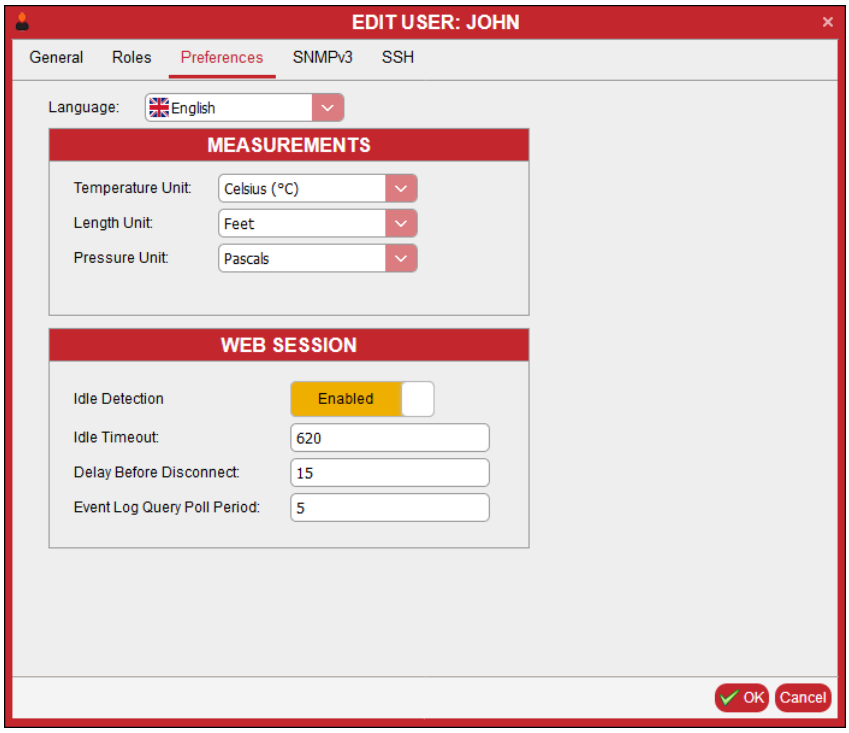
11.11 Preferred Measurement Units

Information about preferred measurement units includes the following:

- Temperature units: Celsius or Fahrenheit degrees
- Length units: Metric (meters) or English (feet)
- Pressure units: Pascal or PSI.

For a new user, measurement units are inherited from global parameters.

From the Web interface, to view and set measurement units for the current user, invoke the dialog box with the menu items “User Management” -> “User Preferences”. For an arbitrary user, choose the user from the list of users (invoked with menu items “User Management” -> “Users”), press the button “Edit user” and choose the tab “Preferences”. In both cases, the same dialog box appears, it gives access to both measurement units and Web session preferences.



11.12 Web Session Preferences

For the Web session preferences, the following information is included:

- Whether idle detection is enabled: *TRUE* or *FALSE*
- Idle detection timeout, in seconds
- Delay before disconnecting the session after idle state is detected (and the corresponding warning is shown), in seconds
- The period to poll the System Event Log, in seconds.

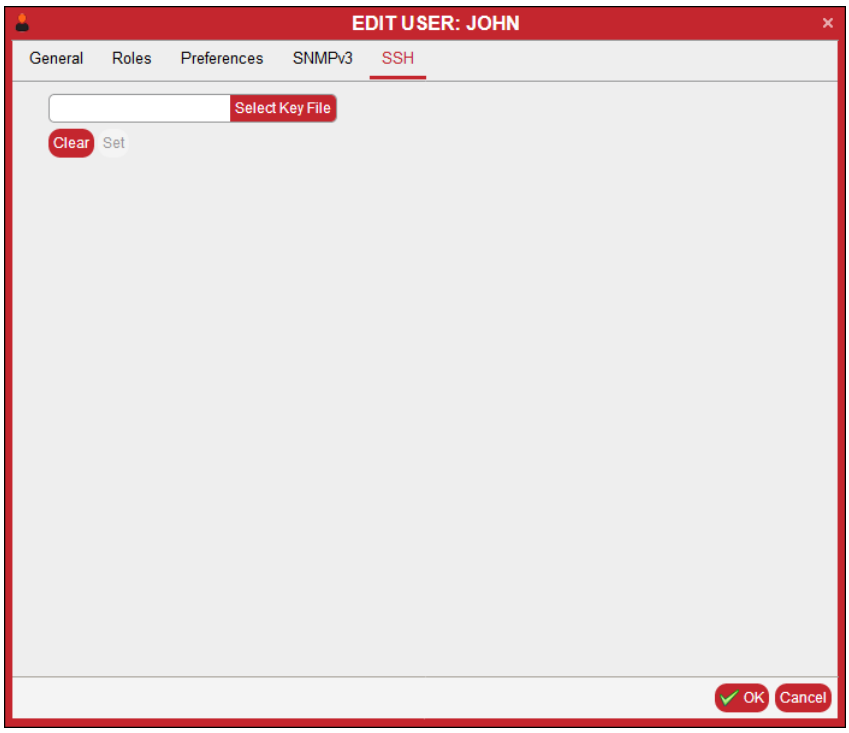
For a new user, Web session parameters are inherited from global parameters.

In the Web interface, Web session preferences appear in the same dialog box as the preferred measurement units (see the previous section).

11.13 SSH public key

SSH supports user authentication by the private/public key pair, without need for entering a password. In that case, a pair of keys is generated for the user on some system, and the public key is stored in a user-specific location `~/.ssh/authorized_keys` on the Guardian Management Gateway file system. During authentication, the SSH client verifies the match of the user's private key (stored on the client) with the user's public key, stored on the Guardian Management Gateway. If the match is successful, the user is authenticated.

To store an SSH public key for the current user on the Guardian Management Gateway with the Web interface, invoke the corresponding dialog via menu: "User Management" -> "Change User SSH Key". To do the same for an arbitrary local user, choose the user via "User Management" -> "Users", press the "Edit User" button and choose the "SSH" tab. In both cases, the dialog looks the same:



The key should be present in a local file; press the "Choose Key File" button to choose the file, and then press the "Set" button to store the key on the Guardian Management Gateway. Press the "Clear" button to delete all public keys stored on the Guardian Management Gateway for the given user.

11.14 SNMPv3 User Settings

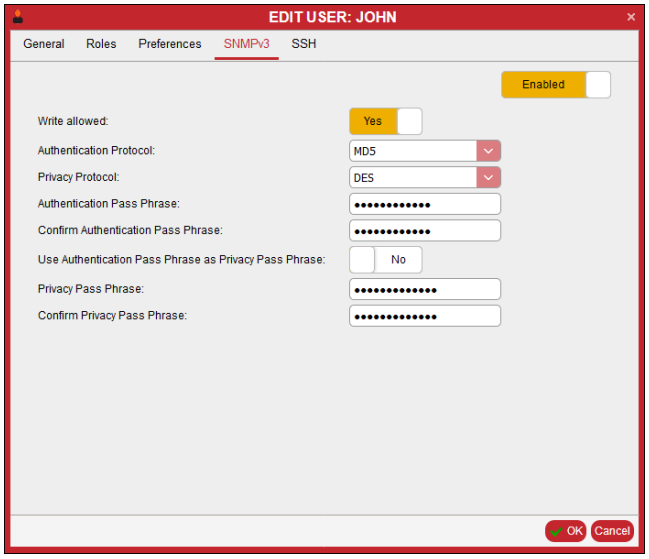
SNMP protocol version 3 implements mandatory user authentication. Therefore, if an external SNMP client communicates with the Guardian Management Gateway using version 3 of the protocol, it should first authenticate itself on the Guardian Management Gateway with some user identity. SNMPv3 user identities on the Guardian Management Gateway match regular user identities; for each user, SNMPv3 identity attributes can be specified. Also, it is possible to configure the SNMP server on the Guardian Management Gateway so that only version 3 of the SNMP protocol is supported; in that case, user authentication is mandatory for any SNMP client.

The following SNMPv3 related settings exist for a user:

- Whether SNMPv3 access is enabled for the user
- Whether the user has read/write (or read-only) access in SNMP terms
- The protocol used for authentication (MD5 or SHA1)
- The protocol user for encryption (None, DES, or AES128)
- The passphrase used for authentication
- The passphrase used for encryption

By default, a new or built-in user is not SNMPv3 enabled; an administrator needs to explicitly set SNMPv3-related parameters for each new user, and for built-in users if necessary.

In the Web interface, SNMPv3 user settings can be viewed and changed on the “SNMPv3” tab of the “Edit User” tabbed dialog. This dialog can be invoked by choosing menu items “User Management” -> “Users”, choosing the user and pressing the button “Edit user”.



After editing is complete, press the “OK” button to save the SNMPv3 settings for the user.

12 Device Management

This section describes managing global user interface preferences (measurement units, Web interface properties, language), managing other global attributes, viewing device make, model and version and managing time attributes (date, time and time zone).

12.1 Global User Interface Preferences and Other Global Attributes

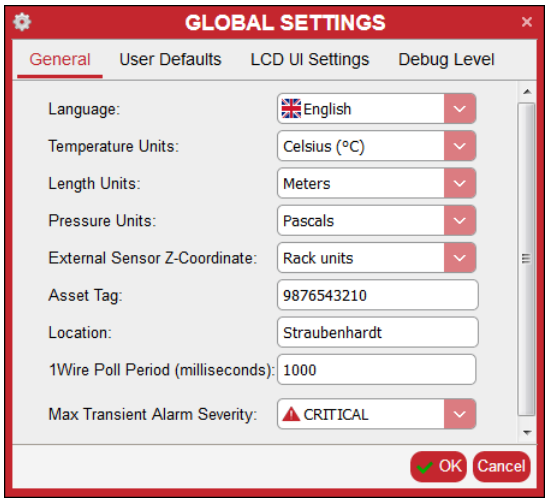
Global user interface preferences are used as defaults assigned to a new user when a new user is created. Also, they are used in the contexts where no logged in user exists (e.g. on the LCD interface to the Guardian Management Gateway). They the following items:

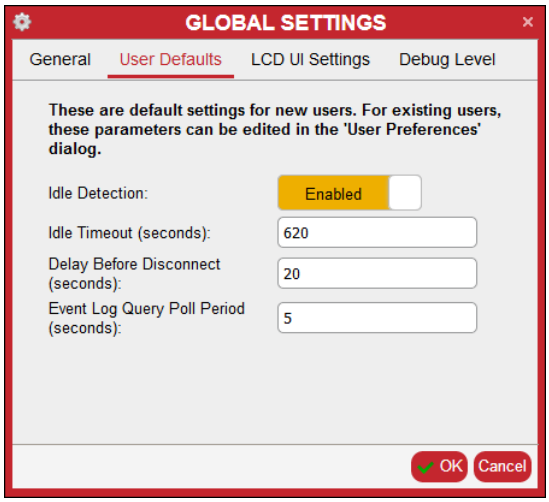
- Measurement units:
  - Temperature units: Celsius or Fahrenheit degrees
  - Length units: meters or feet
  - Pressure units: Pascal or PSI.
- Web interface preferences:
  - Whether idle detection is enabled: *TRUE* or *FALSE*
  - Idle detection timeout, in seconds
  - Delay before disconnecting the session after idle state is detected (and the corresponding warning is shown), in seconds
  - The period to poll the System Event Log, in seconds.
- Interface language (English, German, French)
- LCD User Interface flags (an opaque integer number). The definition of these flags:
  - Bit 1 (mask 2): alarm acknowledgment enabled
  - Bit 2 (mask 4): firmware upgrade enabled
  - Bit 3 (mask 8): configuration loading enabled.

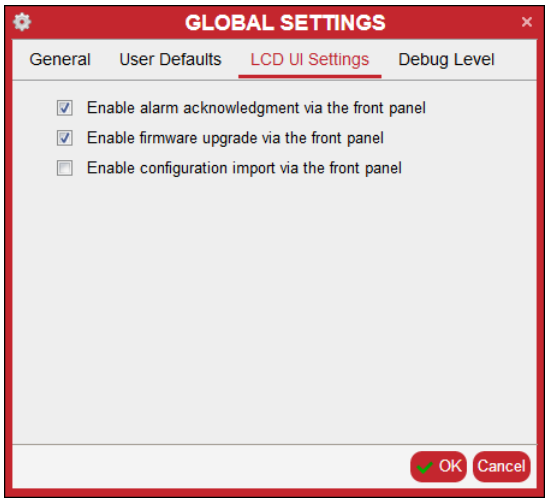
Other global attributes include the following items:

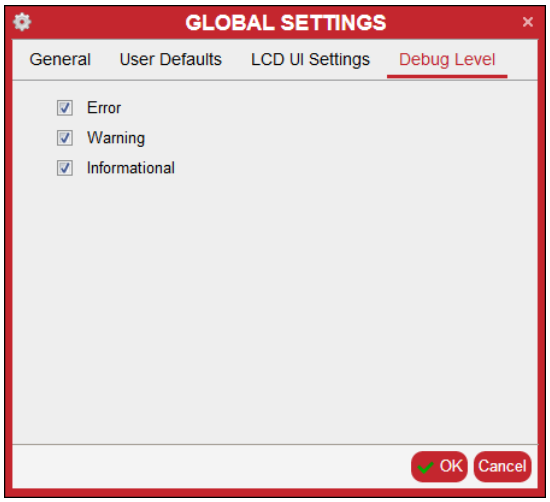
- Debug level: the bit mask that indicates the verbosity of log messages that are posted in the system log file */var/log/messages*, the bits have the following meaning:
  - Bit 0 (mask 1): error level
  - Bit 1 (mask 2): warning level
  - Bit 2 (mask 4): informational level
  - Bit 3 (mask 8): verbose level.
- Z-coordinate type for managed sensors: a two-state flag, can have values “Rack Units” or “Arbitrary text”
- Maximum Transient Alarm Severity: the severity level can be one of “Critical”, “Major”, “Minor”, “Informational” or “OK”. If this value is set to a level less than “Critical”, then more severe alarms are not automatically deleted from the Alarm table when the alarm condition goes away. They stay in the Alarm table until manually deleted by the user or until a restart of the Guardian Management Gateway.

To manage global user interface preferences and other global attributes with the Web interface, use the dialog box “Global Settings” that can be invoked with the menu command “Device Settings” -> “Settings”. This dialog box contains 4 tabs: “General”, “User Defaults”, “LCD UI Settings”, “Debug Level” and allows the user to view and change attributes of both kinds described above:







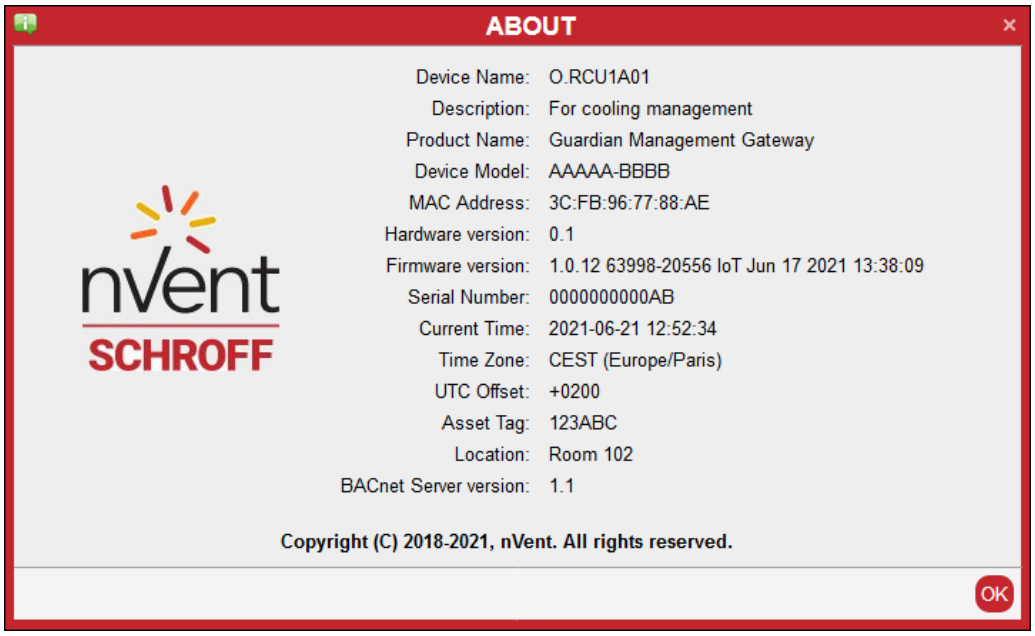


12.2 Device attributes, date and time

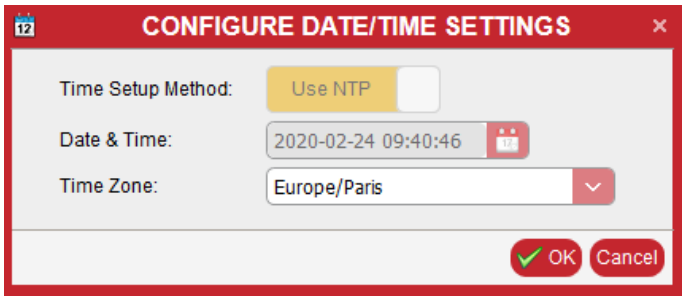
The device attributes include:

- Device name
- Device model
- Device description (it is used in BACnet applications)
- Serial number of the device
- Hardware version
- Software (firmware) version: it includes the application version, Schroff version of the image
- MAC address
- BACnet server version.

In the Web interface, device attributes and the current time are shown in the “About” dialog, which is invoked with the menu command “About”:

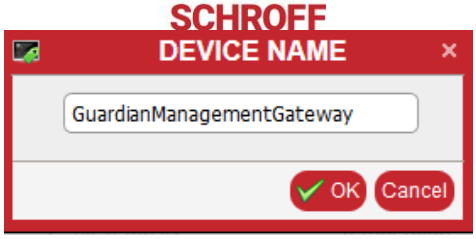


To change the current time and time zone with the Web interface, use the “Configure Date/Time Settings” dialog, which is invoked with the menu command “Device Settings” -> “Date/Time”.

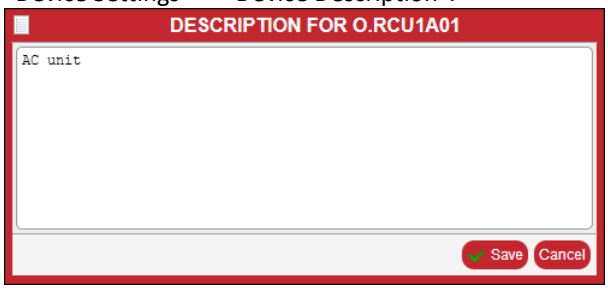


In this dialog, the user can set the date, time (if it is set manually), and choose the time zone. In the picture above, date and time are set via NTP.

To change the device name, use the “Device Name” dialog, which is invoked with the menu command “Device Settings” -> “Device Name”. Spaces are not allowed in the device name.



To change the device description, use the “Device Description” dialog, which is invoked with the menu command “Device Settings” -> “Device Description”.



To change the “Asset Tag” attribute and the “Location” attribute, use the “Global Settings” dialog, which is invoked with the menu command “Device Settings” -> “Settings” (see section 12.1).

13 Network Configuration

Network configuration consists of the following parts:

- Network adapter configuration – applicable to wired Ethernet adapters (except for getting MAC address which applies to all network adapters)
- IPv4 address, subnet mask and default IPv4 gateway assignment – applies to each supported network adapter separately
- List of IPv6 addresses, subnet mask and default IPv6 gateway assignment – applies to each supported network adapter separately
- IPv4 and IPv6 DNS server configuration – applies to the whole system
- Additional DNS attributes: host name and DNS domain search path – apply to the whole system
- List of DHCPv4 and DHCPv6 rejected servers - applies to the whole system

To view and edit network configuration with the Web interface, use subcommands of the menu command “Device Settings” -> “Network”. These subcommands include “Interface Settings”, “IPv4 Settings”, “IPv6 Settings”, “DNS Settings”.

13.1 Network adapter configuration

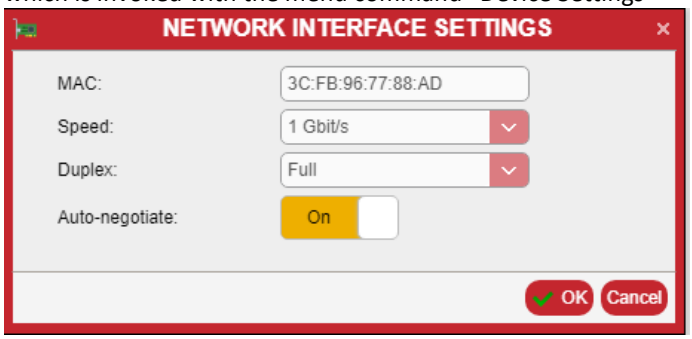
For each network adapter, the following low-level attributes can be configured:

- MAC address (read-only)
- Interface mode and speed (applicable on `eth0`):
  - Auto-negotiate flag: true or false
  - Duplex mode: half-duplex or full-duplex
  - Speed: 10 Mbit/s or 1000 Mbit/s (higher speeds are set as 1000 Mbit/s).

If the auto-negotiate flag is set to true, interface speed and duplex mode are set up automatically from the transmission media; manual settings are ignored.

By default, auto-negotiation on `eth0` is turned on, so duplex mode and speed are automatically assigned.

To view and edit the low-level attributes for `eth0` with the Web interface use the “Network Interface Settings” dialog, which is invoked with the menu command “Device Settings” -> “Network”-> “Interface Settings”.



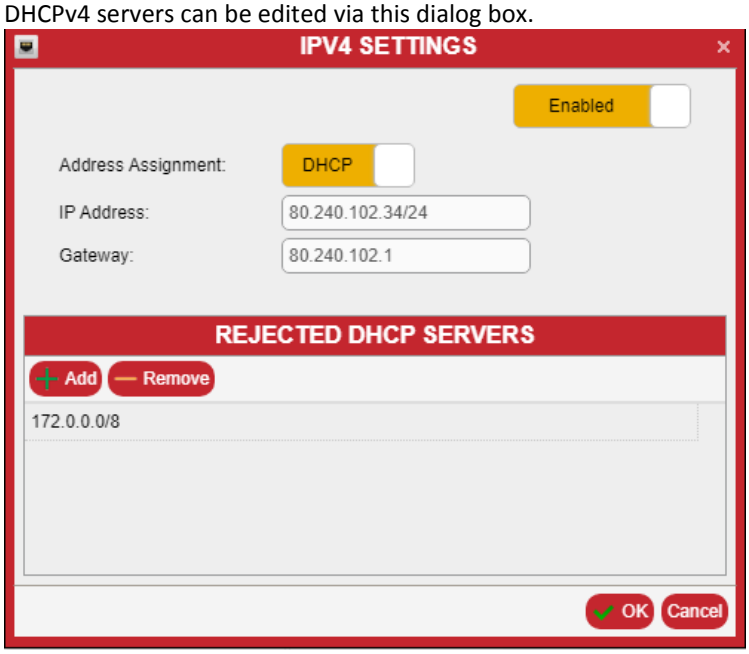
13.2 IPv4 configuration

For each network adapter, the following IPv4 configuration attributes can be configured:

- IPv4 address of the Guardian Management Gateway
- Subnet mask
- Default IPv4 gateway address
- Address assignment type (static or DHCP).

By default, the IPv4 address, subnet mask and the default gateway address are assigned automatically from DHCP. The user should set the assignment type to static to assign these attributes manually.

To view and edit the IP4 configuration attributes for `eth0` interface with the Web interface use the “IPv4 Settings” dialog, which is invoked with the menu command “Device Settings” -> “Network”-> “IPv4 Settings”. The list of rejected DHCPv4 servers can be edited via this dialog box.



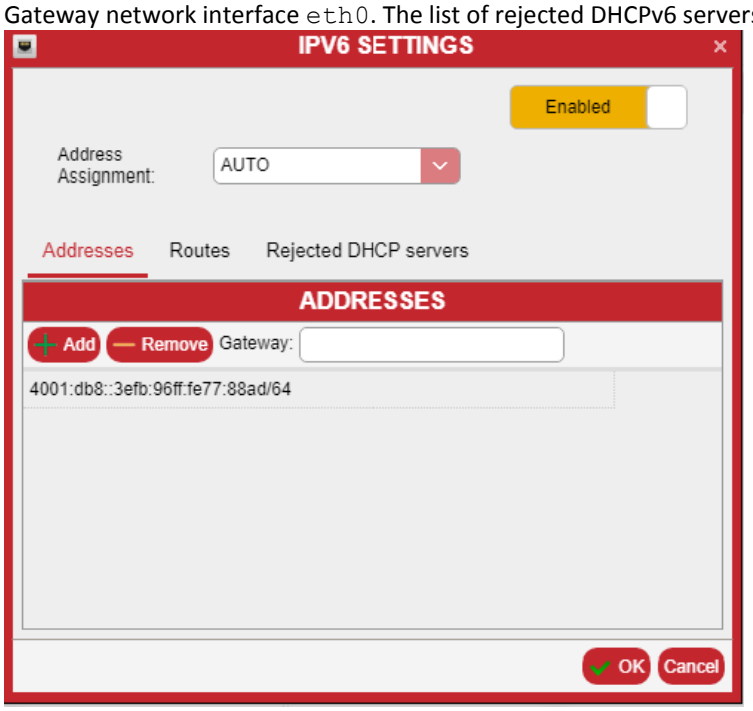
13.3 IPv6 configuration

For each network adapter, the following IPv6 configuration attributes can be configured:

- List of IPv6 addresses with subnet prefixes
- Default IPv6 gateway address.

By default, no IPv6 addresses are configured for a network adapter; an IPv6 address with the link scope is usually auto-configured for each network adapter by the system.

To view and edit the IP6 configuration attributes for `eth0` interface with the Web interface use the “IPv6 Settings” dialog, which is invoked with the menu command “Device Settings” -> “Network”-> “IPv6 Settings”. Besides the IPv6 default gateway, this dialog box also shows the list of currently assigned IPv6 addresses for the Guardian Management Gateway network interface `eth0`. The list of rejected DHCPv6 servers can be edited via this dialog box.



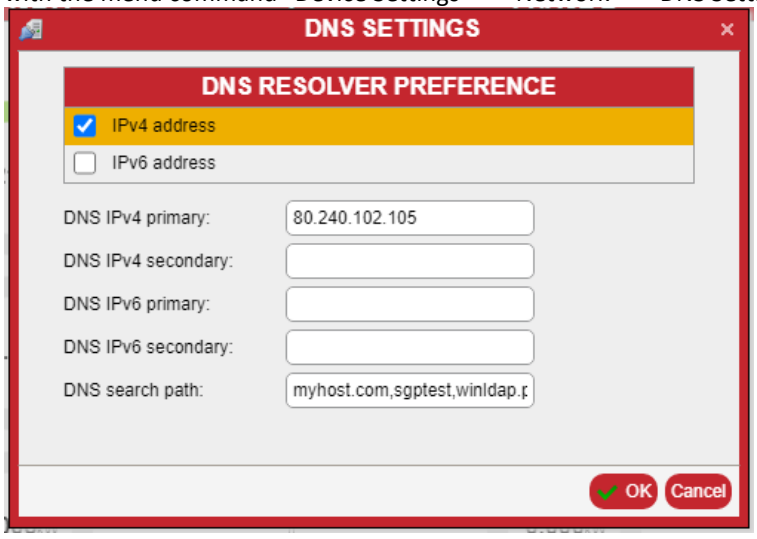
13.4 DNS server configuration

These configuration attributes specify the location of the DNS server; they are system-wide, but server addresses are defined separately for IPv4 and IPv6 protocols and a choice can be made between them:

- DNS resolver preference flag: which DNS settings to prefer, IPv4 or IPv6?
- IPv4 address of the primary DNS server
- IPv4 address of the secondary DNS server
- IPv6 address of the primary DNS server
- IPv6 address of the secondary DNS server.

By default, in the case of automatic IPv4 address assignment, IPv4 information takes preference and DNS server addresses are provided by the corresponding DHCPv4 server.

To view and edit the DNS server configuration with the Web interface use the “DNS Settings” dialog, which is invoked with the menu command “Device Settings” -> “Network”-> “DNS Settings”.



13.5 Additional configurable DNS attributes

These attributes are system-wide and include the following:

- Guardian Management Gateway host name
- DNS domain search path.

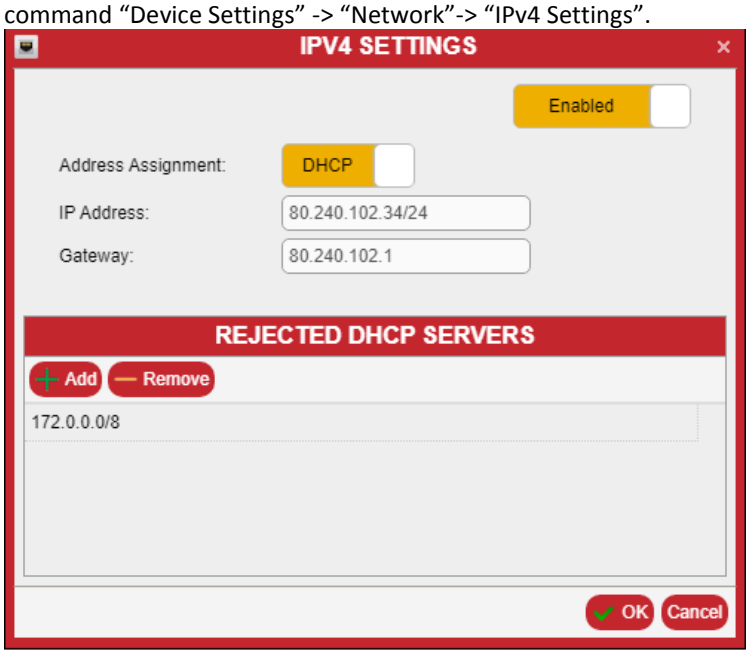
By default, these attributes are not set, but the DHCP server can provide the host name.

The host name can be changed in the “Device Name” dialog, which is invoked with the menu command “Device Settings” -> “Device Name”.

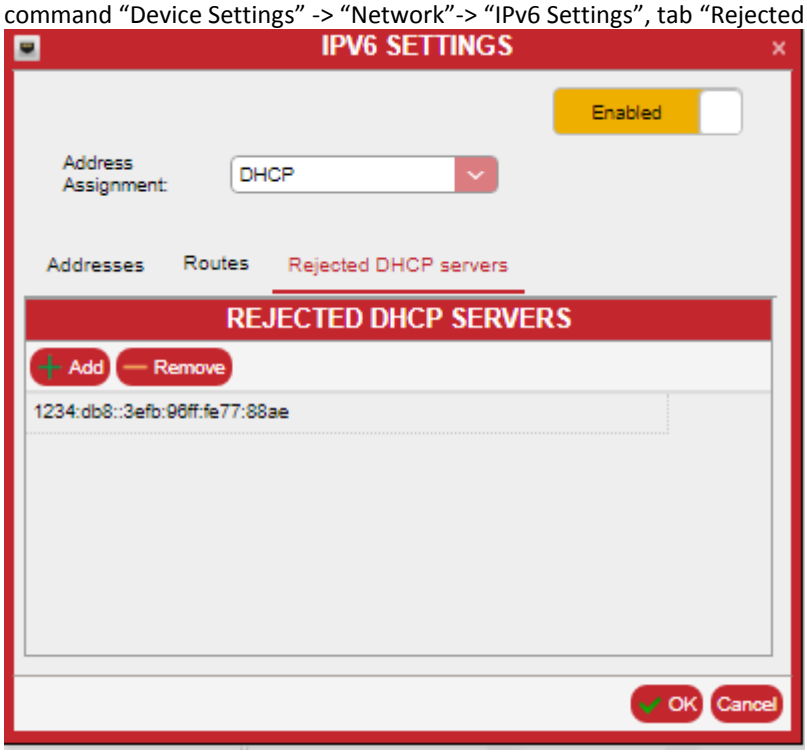
13.6 List of rejected DHCP servers

In some cases, when DHCP is used, it may be necessary to avoid accepting configuration from certain DHCP servers, which are available in the local network. It is possible to configure the list of rejected DHCP server addresses (both for DHCPv4 and for DHCPv6); the Guardian Management Gateway will not accept configuration parameters from these servers.
Two address lists can be configured: the list of IPv4 addresses for DHCPv4 and the list of IPv6 addresses for DHCPv6.

The list of rejected DHCPv4 servers can be edited via the “IPv4 Settings” dialog box, which is invoked with the menu command “Device Settings” -> “Network”-> “IPv4 Settings”.

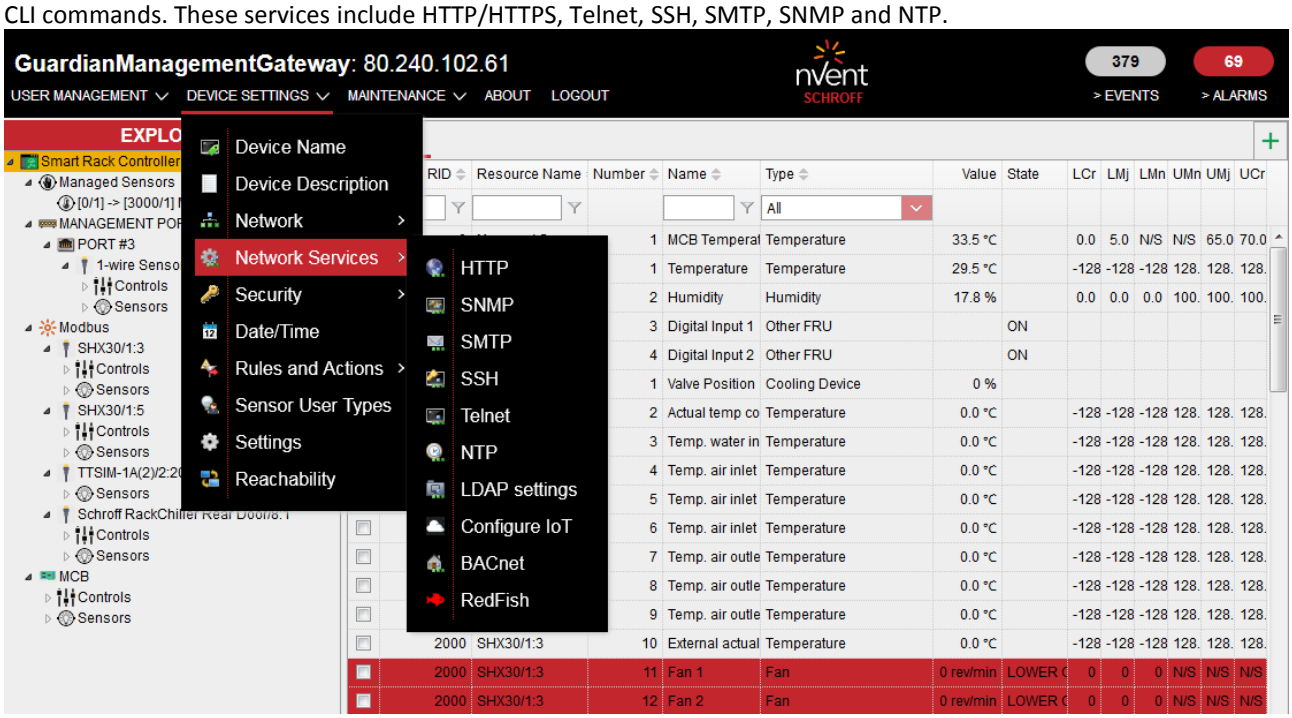


The list of rejected DHCPv6 servers can be edited via the “IPv6 Settings” dialog box, which is invoked with the menu command “Device Settings” -> “Network”-> “IPv6 Settings”, tab “Rejected DHCP Servers”.



14 Network Service Configuration

The user via the web interface can configure several network services provided by Guardian Management Gateway or CLI commands. These services include HTTP/HTTPS, Telnet, SSH, SMTP, SNMP and NTP.



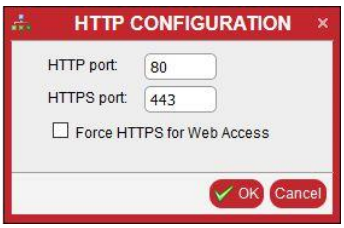
14.1 HTTP/HTTPS configuration

HTTP and HTTPS network services provide Web interface to the Guardian Management Gateway. On the Guardian Management Gateway, the web server program `lighttpd` provides these services. HTTPS, unlike HTTP, provides secure access to the Guardian Management Gateway over the Web interface, the corresponding traffic is encrypted.

For HTTP and HTTPS, the user can configure the following parameters:
HTTP port
HTTPS port
Enforce HTTPS (a logical flag, if `TRUE`, only secure access is allowed to the Guardian Management Gateway).

Default settings:
HTTP port = 80
HTTPS port = 443
Enforce HTTPS = False

To change the HTTP/HTTPS configuration, invoke the menu command “Device Settings” -> “Network Services” -> “HTTP”. The “HTTP Configuration” dialog appears.



14.2 SNMP Configuration

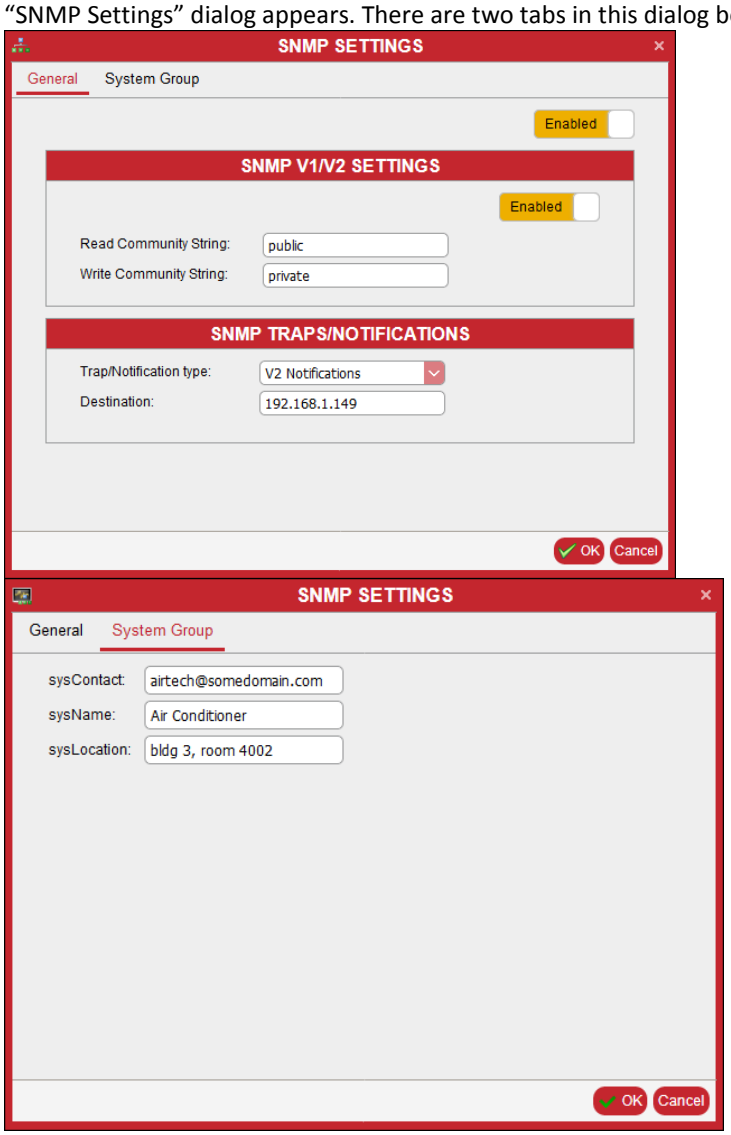
SNMP service provides SNMP interface to the Guardian Management Gateway. On the Guardian Management Gateway, the SNMP server program *snmpd* provides this service.
For SNMP, the user can configure the following parameters:

- Whether SNMP service is enabled (*TRUE*/*FALSE*)
- Whether SNMP v1/v2 legacy protocols are enabled; if false, only secure SNMPv3 protocol can be used to communicate to the Guardian Management Gateway over the SNMP interface
- Read community string
- Write community string
- The “Sys Name” string
- The “Sys Contact” string
- The “Sys Location” string
- IP address of the SNMP trap destination system
- Whether to use SNMPv2 format for SNMP traps (if *FALSE*, SNMPv1 format is used).

Default settings:

- SNMP service is enabled = *TRUE*
- SNMP v1/v2 legacy protocols are enabled = *TRUE*
- Read community string = *public*
- Write community string = *private*
- The “Sys Name” string = “*GuardianManagementGateway*”
- The “Sys Contact” string = *ipdu-support@nVent.com*
- The “Sys Location” string = *Unknown*
- IP address of the SNMP trap destination system = not specified
- Whether to use SNMPv2 format for SNMP traps = *TRUE*.

To change the SNMP parameters, invoke the menu command “Device Settings” -> “Network Services” -> “SNMP”. The “SNMP Settings” dialog appears. There are two tabs in this dialog box: “General” and “System Group”.



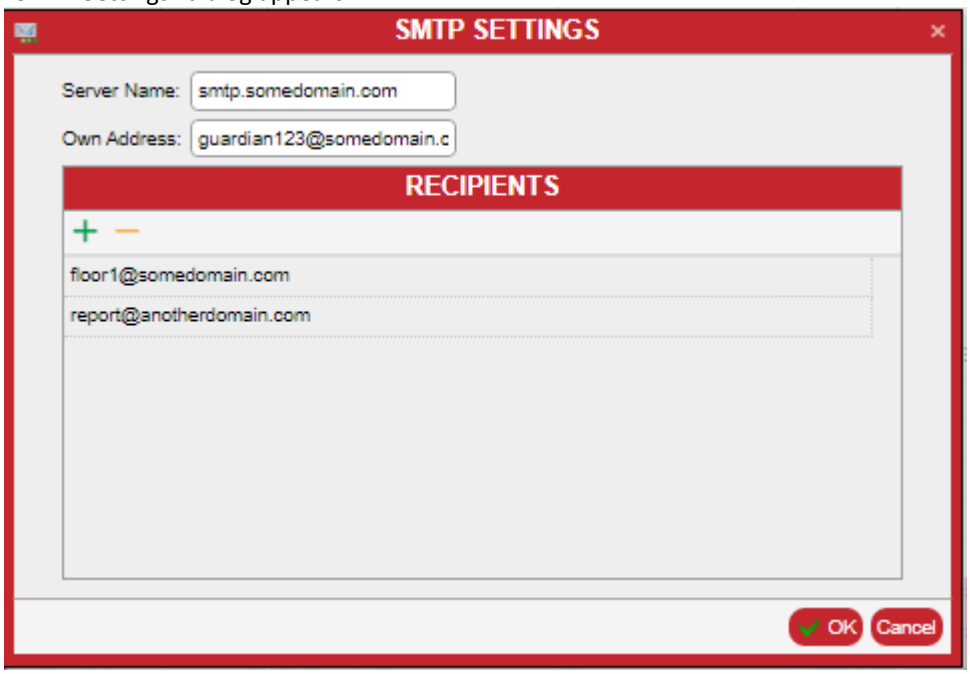
14.3 SMTP Configuration

SMTP service allows sending e-mail. On the Guardian Management Gateway, there is the SMTP client that can connect to an external SMTP server and send e-mail messages. Sending e-mail is one of the actions that can be specified in event filtering. In that case, the message body is constructed on the base of the event just received.
The user can configure the following SMTP parameters:

- Guardian Management Gateway own e-mail address
- Name or IP address of the SMTP server
- The default list of recipient e-mail addresses (comma-separated).

These parameters are specified once for all event filters; other parameters, like the e-mail subject line and the actual list of recipients, are specified as parameters for a specific action in a specific event filter.
By default, these parameters are not specified (empty strings).

To change the SMTP parameters, invoke the menu command “Device Settings” -> “Network Services” -> “SMTP”. The “SMTP Settings” dialog appears.



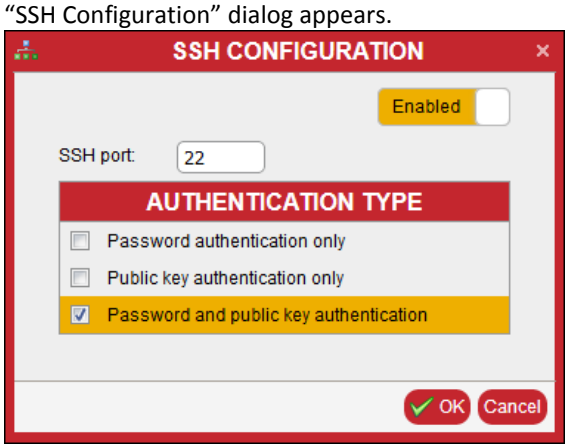
14.4 SSH Configuration

SSH service allows secure terminal access to an Guardian Management Gateway; SSH traffic is encrypted in transit. SSH protocol is the preferred instrument for terminal access to an Guardian Management Gateway. On the Guardian Management Gateway, SSH service is provided by the `sshd` daemon.
For SSH, the user can configure the following parameters:

- Whether SSH service is enabled
- SSH port
- Supported SSH authorization methods: by password, by public key or both.

By default, SSH service is enabled on port 22, with both authorization methods (password and public key) supported.

To change the SSH configuration, invoke the menu command “Device Settings” -> “Network Services” -> “SSH”. The “SSH Configuration” dialog appears.



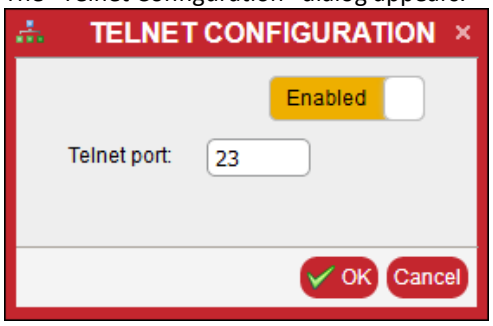
14.5 Telnet Configuration

Telnet service allows terminal access to an Guardian Management Gateway. Telnet protocol is not secure, so SSH protocol is the preferred instrument for terminal access to an Guardian Management Gateway. On the Guardian Management Gateway, Telnet service is provided by the `telnetd` daemon.
For Telnet, the user can configure the following parameters:

- Whether Telnet service is enabled
- Telnet port.

By default, Telnet port is 23 and this service is disabled.

To change the Telnet configuration, invoke the menu command “Device Settings” -> “Network Services” -> “Telnet”. The “Telnet Configuration” dialog appears.



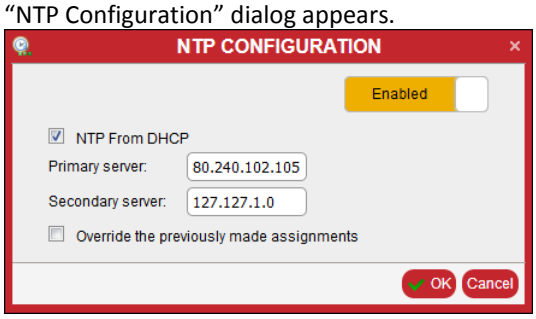
14.6 NTP Configuration

NTP service allows time synchronization with external servers. On the Guardian Management Gateway, the NTP client exists that is able to connect to an external NTP server and obtain current time from it.
The user can configure the following NTP parameters:

- Whether NTP client functionality is enabled (if *FALSE*, system time must be manually set by the administrator)
- Whether NTP client should obtain NTP server addresses via DHCP (if *FALSE*, NTP server addresses must be set manually)
- Name or IP address of the primary NTP server (if obtaining via DHCP is disabled)
- Name or IP address of the secondary NTP server (if obtaining via DHCP is disabled).

By default, NTP client is enabled and obtaining NTP server addresses via DHCP is enabled; primary and secondary NTP server addresses are not set.

To change the NTP configuration, invoke the menu command “Device Settings” -> “Network Services” -> “NTP”. The “NTP Configuration” dialog appears.



15 LDAP Configuration

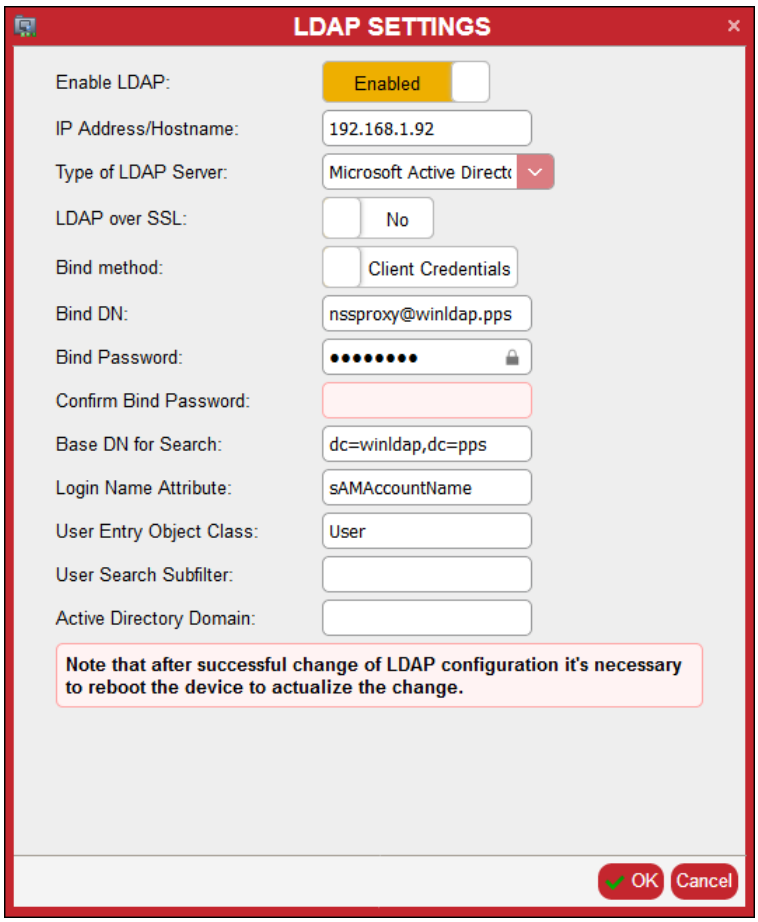
Guardian Management Gateway supports authenticating users via LDAP. If this method is used, user records are stored on a remote server and user authentication on the Guardian Management Gateway involves communication with that server. The Guardian Management Gateway itself may not have information about the user that intends to log in; if this is the case and remote authentication is successful, this user is considered an “external user” and user information record with default attributes is created on the Guardian Management Gateway for that user; in particular, that user is assigned the predefined role “ExternalUserRole”, that defines its privileges with respect to the Guardian Management Gateway.

LDAP configuration parameters, that a user can view and set, include the following:

- Whether logging in via LDAP is enabled (*TRUE*/*FALSE*)
- LDAP server name (URI)
- Server type (OpenLDAP or ActiveDirectory, other parameters may need to be set differently depending on the server type)
- Whether to use SSL for connection to the LDAP server
- SSL port number (if SSL is used)
- SSL certificate for the server
- Whether the client uses an anonymous bind to the LDAP server to authenticate a user
- Distinguished name used for binding (only if anonymous bind is not used)
- Password used for binding (only if anonymous bind is not used)
- Distinguished name used as search base
- Login name attribute (normally “sAMAccountName” for ActiveDirectory servers or empty for OpenLDAP servers)
- User entry object class (normally “User”)
- User search subfilter
- Extra configuration options (as a sequence of strings in the format `<option> <value>`, separated by the newline characters)

By default, LDAP-based logins are disabled and all other parameters are undefined.
When LDAP is enabled by a user, all LDAP configuration parameters should also be supplied by the user in the same command or dialog box. When LDAP is disabled by a user, all other configuration parameters become undefined and need not be specified.

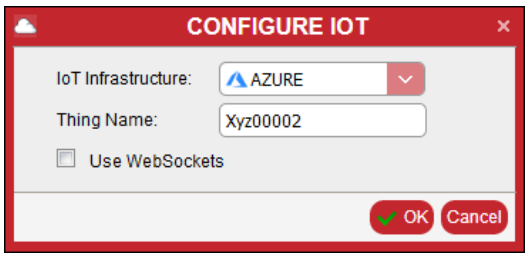
In the Web interface, LDAP is configured via a dialog box that is accessible via menu items “Device Settings” -> “Network Services” -> “LDAP Settings”. This dialog box allows the user to specify all LDAP configuration parameters. For the SSL certificate, local path to the certificate should be specified; the certificate will be downloaded to the fixed place in the Guardian Management Gateway file system.



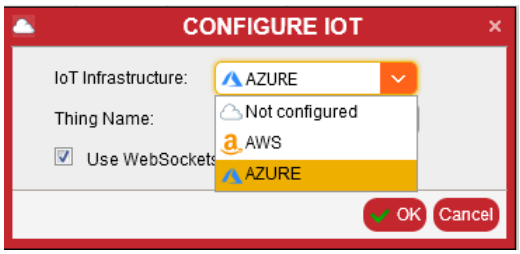
16 IoT

In the Web interface, IoT is configured via a dialog box that is accessible via menu items “Device Settings” -> “Network Services” -> “Configure IoT”.

This dialog box allows the user to obtain and change the IoT configuration.



Currently, the Guardian Management Gateway supports MS Azure IoT and AWS IoT/AWS Greengrass IoT. The WebSockets are not supported for AWS Greengrass IoT.

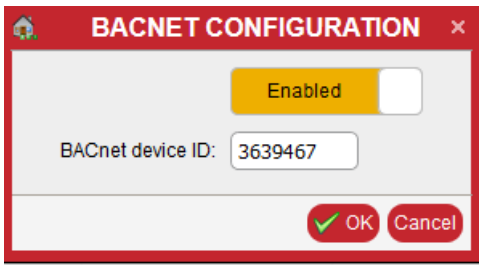


The status of the IoT connection is reported in the status bar at the bottom of the page.

17 BACnet

In the Web interface, BACnet is configured via a dialog box that is accessible via menu items “Device Settings” -> “Network Services” -> “BACnet”.

This dialog box allows the user to enable BACnet and change the BACnet device ID.



By default, this service is disabled. The default BACnet device ID depends of the MAC address of the Guardian Management Gateway device.

17.1 BACnet Overview

HPI objects are mapped to BACnet objects as follows:

HPI	MAPPING	BACNET
Guardian Management Gateway	← →	Device Object
Resource	← →	Structured View objects
Sensor	← →	Analog Input, Binary Input and Multi-State Input objects
Control	← →	Analog Output, Binary Output and Multi-state Output objects
HPI Event	← →	BACnet Event
HPI Alarm Table	← →	Get Alarm Summary service
HPI Event Log	← →	BACnet Event Log object

- A single Guardian Management Gateway device is mapped to the Device object. The Device object instance ID is by default based on the MAC address of the device but can be changed by the user.
- HPI resources are mapped to Structured View objects.
- HPI sensors are mapped to Analog Input, Binary Input and Multi-State Input objects; the corresponding object belongs to the Structured View object which corresponds to the resource – owner of the sensor.
- HPI controls are mapped to Analog Output and Binary Output objects; the corresponding object belongs to the Structured View object which corresponds to the resource – owner of the control.
- HPI events are mapped to BACnet events and are forwarded to BACnet clients via the subscriptions in the Notification Class object.
- HPI Alarm Table is exposed to BACnet via the services Get Alarm Summary, Acknowledge Alarm.
- HPI Event Log is mapped to the BACnet Event Log object .

In addition, our BACnet server exposes Calendar, Schedule and Trend Log objects. These objects are not directly mapped to HPI objects, they are indirectly associated with them via object references. These references are embedded in them and refer to other BACnet objects, which are normally directly mapped to HPI objects.

17.2 Device Object

A single device object exists for the Guardian Management Gateway. This object describes global properties of the device. Object instance for the device object should be unique among all BACnet device objects in the network. By default, it is based on the MAC address of the Guardian Management Gateway, but it can be changed by the user via CLI or web interface.

Properties:

PROPERTY NAME	ACCESS	PROPERTY SOURCE
Object Type	RO	8 = “Device”
Object Instance	RO	Unique number, based on MAC address by default (but can be redefined by the user)
Object name	RO	Device name
Description	RW	Device description, arbitrary text up to 256 characters
Apdu Timeout	RO	3000
Application Software Version	RO	“1.1”
Database Revision	RO	1
Daylight Saving Status	RO	Based on the current timezone
Device Address Binding	RO	Empty list
Firmware Revision	RO	Guardian Management Gateway firmware version

Local Date	RO	Current system date
Local Time	RO	Current system time
Location	RW	Based on the current timezone
Max Apdu Length Accepted	RO	1476
Model Name	RO	Model name from the inventory
Number Of Apdu Retries	RO	3
Protocol Object Types Supported	RO	The bit string – the mask of supported object types
Protocol Version	RO	1
Protocol Revision	RO	12
Protocol Services Supported	RO	The bit string – the mask of supported protocol services
Segmentation Supported	RO	0:Both (segmentation is supported for both Transmit and Receive)
System Status	RO	"Operational"
Utc Offset	RO	Based on the current timezone
Vendor Identifier	RO	1094 ("Nvent Thermal Management")
Vendor Name	RO	Manufacturer name from the inventory

17.3 Structured View

HPI Resources are mapped to Structured View objects. All sensors and controls which belong to the resource are mapped to the Subordinate List for this Structured View.
Properties:

PROPERTY NAME	ACCESS	PROPERTY SOURCE
Object Type	RO	29 = "Structured View"
Object Instance	RO	Small integer number, assigned sequentially
Object name	RO	Resource name
Description	RW	Resource description, arbitrary text up to 256 characters
Subordinate List	RO	List of object identifiers for the mapped sensors/controls which belong to this resource

17.4 Analog Input

HPI analog sensors (sensors that have numeric values) are mapped to Analog Input objects.
Properties:

PROPERTY NAME	ACCESS	PROPERTY SOURCE
Object Type	RO	0 = “Analog Input”
Object Instance	RO	Small integer number, assigned sequentially
Object name	RO	Sensor name
Description	RW	Sensor description, arbitrary text up to 256 characters
Present Value	RO	Current numeric value of the sensor
Event State	RO	“Normal” if the sensor value is within thresholds “Offnormal” if the sensor value is beyond thresholds
Out of service	RO	FALSE
Units	RO	Sensor units (in BACnet encoding)
Reliability	RO	“No Fault Detected”
Status Flags	RO	Flag “In Alarm” is set if the sensor value is beyond thresholds; otherwise no flags are set
Notification Class	RW	Handled internally inside the BACnet server; contains the instance of the notification class object used for notifications from this object
COV Increment	RW	Handled internally inside the BACnet server; default value is 1

17.5 Binary Input

HPI discrete sensors with two states are mapped to Binary Input objects.
Properties:

PROPERTY NAME	ACCESS	PROPERTY SOURCE
Object Type	RO	3 = “Binary Input”
Object Instance	RO	Small integer number, assigned sequentially
Object name	RO	Sensor name
Description	RW	Sensor description, arbitrary text up to 256 characters
Polarity	RO	“Normal”
Present Value	RO	0 if the sensor is in the first state; 1 if the sensor is in the second state
Event State	RO	“Offnormal” if the severity of the current sensor state is MINOR, MAJOR or CRITICAL “Normal” otherwise
Out of service	RO	FALSE
Units	RO	Empty (discrete sensors do not have units)
Reliability	RO	“No Fault Detected”
Status Flags	RO	Flag “In Alarm” is set if the severity of the current sensor state is MINOR, MAJOR or CRITICAL; otherwise no flags are set
Inactive Text	RO	Name of the first state of the sensor; “OFF” if undefined
Active Text	RO	Name of the second state of the sensor; “ON” if undefined
Notification Class	RW	Handled internally inside the BACnet server; contains the instance of the notification class object used for notifications from this object
COV Increment	RW	Handled internally inside the BACnet server; default value is 1

17.6 Multi-State Input

HPI discrete sensors with more than two states are mapped to Multi-State Input objects.
Properties:

PROPERTY NAME	ACCESS	PROPERTY SOURCE
Object Type	RO	13 = “Multi-State Input”
Object Instance	RO	Small integer number, assigned sequentially
Object name	RO	Sensor name
Description	RW	Sensor description, arbitrary text up to 256 characters
Number of States	RO	Number of states defined for the sensor
Present Value	RO	Bit string of sensor states; each bit = 1 if the state is asserted and 0 if the state is not asserted
Event State	RO	“Offnormal” if the severity of any of the asserted states is MINOR, MAJOR or CRITICAL “Normal” otherwise
Out of service	RO	FALSE
Units	RO	Empty (discrete sensors do not have units)
Reliability	RO	“No Fault Detected”
Status Flags	RO	Flag “In Alarm” is set if the severity of any of the asserted states is MINOR, MAJOR or CRITICAL; otherwise no flags are set
State Text	RO	Array of sensor state names (as strings)
Notification Class	RW	Handled internally inside the BACnet server; contains the instance of the notification class object used for notifications from this object
COV Increment	RW	Handled internally inside the BACnet server; default value is 1

17.7 Analog Output

HPI analog controls and HPI discrete controls without explicit state names are mapped to Analog Output objects.
Properties:

PROPERTY NAME	ACCESS	PROPERTY SOURCE
Object Type	RO	1 = “Analog Output”
Object Instance	RO	Small integer number, assigned sequentially
Object name	RO	Control name
Description	RW	Control description, arbitrary text up to 256 characters
Present Value	RW	Current numeric value of the control, can be set
Priority Array	RO	The array of priorities and last values assigned to the control at that priority (according to the regular BACnet semantics)
Relinquish Default	RO	The default value for the control (specified in the static attributes of the control)
Event State	RO	“Normal”
Out of service	RO	FALSE
Units	RO	Control units if information about units is available (in BACnet encoding)
Status Flags	RO	No flags are set
Notification Class	RW	Handled internally inside the BACnet server; contains the instance of the notification class object used for notifications from this object
COV Increment	RW	Handled internally inside the BACnet server; default value is 1

17.8 Binary Output

HPI digital controls are mapped to Binary Output objects.
Properties:

PROPERTY NAME	ACCESS	PROPERTY SOURCE
Object Type	RO	4 = “Binary Output”
Object Instance	RO	Small integer number, assigned sequentially
Object name	RO	Control name
Description	RW	Control description, arbitrary text up to 256 characters
Polarity	RO	“Normal”
Present Value	RW	Control value: 0 for the “Off” state, 1 for the “On” state, can be set
Priority Array	RO	The array of priorities and last values assigned to the control at that priority (according to the regular BACnet semantics)
Relinquish Default	RO	The default value for the control (specified in the static attributes of the control)
Event State	RO	“Normal”
Out of service	RO	FALSE
Units	RO	None (no units are defined for digital controls)
Status Flags	RO	No flags are set
Inactive Text	RO	“OFF”
Active Text	RO	“ON”
Notification Class	RW	Handled internally inside the BACnet server; contains the instance of the notification class object used for notifications from this object
COV Increment	RW	Handled internally inside the BACnet server; default value is 1

17.9 Multi-State Output

HPI discrete controls with explicit state names are mapped to Multi-State Output objects.
Properties:

PROPERTY NAME	ACCESS	PROPERTY SOURCE
Object Type	RO	14 = “Multi-State Output”
Object Instance	RO	Small integer number, assigned sequentially
Object name	RO	Control name
Description	RW	Control description, arbitrary text up to 256

		characters
Number of States	RO	Number of explicit states defined for the control
Present Value	RW	Control value current state as an unsigned number, can be set
Priority Array	RO	The array of priorities and last values assigned to the control at that priority (according to the regular BACnet semantics)
Relinquish Default	RO	The default value for the control (specified in the static attributes of the control)
Event State	RO	"Normal"
Out of service	RO	FALSE
Units	RO	None (no units are defined for discrete controls)
Status Flags	RO	No flags are set
State Text	RO	Array of state names (as strings)
Notification Class	RW	Handled internally inside the BACnet server; contains the instance of the notification class object used for notifications from this object
COV Increment	RW	Handled internally inside the BACnet server; default value is 1

17.10 Calendar

Calendar objects exist exclusively inside the BACnet server component of the Guardian Management Gateway firmware and are not mapped directly to any HPI object. Some of their properties are writable by BACnet clients. There are N Calendar objects for each Guardian Management Gateway instance (N is a configuration parameter)
Properties:

PROPERTY NAME	ACCESS	PROPERTY SOURCE
Object Type	RO	6 = "Calendar"
Object Instance	RO	Sequential number of the given Calendar object, 0 to $N-1$
Object name	RW	Object name, "Calendar_<n>" by default but can be changed by a client
Description	RW	Object description, arbitrary text up to 256 characters
Present Value	RO	Boolean value; TRUE if current date matches the calendar, FALSE otherwise
Date List	RW	The list of dates which comprise the calendar. Each component of the list can be in one of the three formats: - Specific date or date pattern - Range of dates - Month/week-of-month/day-of-week specification

17.11 Schedule

Schedule objects exist exclusively inside the BACnet server component of the Guardian Management Gateway firmware and are not mapped directly to any HPI object. Some of their properties are writable by BACnet clients. There are N Schedule objects for each Guardian Management Gateway instance (N is a configuration parameter)

Properties:

PROPERTY NAME	ACCESS	PROPERTY SOURCE
Object Type	RO	17 = “Schedule”
Object Instance	RO	Sequential number of the given Calendar object, 0 to $N-1$
Object name	RW	Object name, “Schedule_<n>” by default but can be changed by a client
Description	RW	Object description, arbitrary text up to 256 characters
Present Value	RO	The value of Any type; equals to the value from the currently active schedule item, or to the default value if no schedule item is currently active
Effective Period	RW	The date range, in which the schedule is effective
Schedule Default	RW	The default value; it is assigned to the present value when no item of the schedule is active.
List of Object Property References	RW	The list of object property references; the present value of the schedule object is assigned to all referenced properties, when it changes.
Priority for Writing	RW	The priority for writing the present value to the referenced objects, 1 to 16
Reliability	RO	“No Fault Detected”
Status Flags	RO	No flags are set
Out of service	RO	FALSE
Weekly Schedule	RW	The array of 7 BACnetDailySchedule objects, one for each day of the week
Exception Schedule	RW	The list of exceptions from the regular schedule, each list entry is a BACnetSpecialEvent object.

17.12 Notification Class

Notification Class objects are not mapped directly to any HPI object but are used to dispatch notifications from BACnet objects mapped to HPI objects. Each of these mapped objects has the property “Notification Class” which contains the object instance of the Notification Class used. That property is writable by BACnet clients.
A BACnet client can subscribe to notifications from a specific notification class by writing to the property “Recipient List” of the corresponding Notification Class object.
There are N Notification Class objects for each Guardian Management Gateway instance (N is a configuration parameter)
Properties:

PROPERTY NAME	ACCESS	PROPERTY SOURCE
Object Type	RO	15 = “Notification Class”
Object Instance	RO	Sequential number of the given Calendar object, 0 to $N-1$
Object name	RW	Object name, “Notification_Class_<n>” by default but can be changed by a client
Description	RW	Object description, arbitrary text up to 256 characters
Notification Class	RO	The object instance of this object
Priority	RW	The array of 3 priorities, for the “Transition To Offnormal”, “Transition To Fault” and

		"Transition To Normal" events, respectively
Ack Required	RW	Boolean value, indicates if acknowledgement is required for notifications
Recipient List	RW	The list of subscribers for notifications from this notification class

17.13 Trend Log

Trend Log objects exist exclusively inside the BACnet server component of the Guardian Management Gateway firmware and are not mapped directly to any HPI object. Some of their properties are writable by BACnet clients. There are N Trend Log objects for each Guardian Management Gateway instance (N is a configuration parameter)
Properties:

PROPERTY NAME	ACCESS	PROPERTY SOURCE
Object Type	RO	20 = "Trend Log"
Object Instance	RO	Sequential number of the given Calendar object, 0 to $N-1$
Object name	RW	Object name, "Trend_Log_<n>" by default but can be changed by a client
Description	RW	Object description, arbitrary text up to 256 characters
Enable	RW	This property, of Boolean type, indicates and controls whether (TRUE) or not (FALSE) logging of events and collected data is enabled. Logging occurs if and only if Enable is TRUE, Local_Time is on or after Start_Time, and Local_Time is before Stop_Time.
Stop When Full	RW	This property, of Boolean type, specifies whether (TRUE) or not (FALSE) logging should cease when the log buffer is full.
Buffer Size	RO	This property specifies the maximum number of log records the log buffer may hold.
Log Buffer	RO	The array of log entries; use the Read Range service to access them.
Record Count	RW	Number of log records currently placed to the log buffer. A write of the value zero to this property clears the log.
Total Record Count	RO	The total number of records added to the trend log, since its creation.
Event State	RO	The object does not support event reporting, the value of this property is always "Normal".
Logging Type	RW	Specifies whether the Trend Log object collects log records using polling (0) or triggered (2) acquisition.
Status Flags	RO	No flags are set
Start Time	RW	Specifies the date and time at or after which logging is enabled.
Stop Time	RW	Specifies the date and time at or after which logging is disabled.
Log Device Object Property	RW	Specifies the Device Identifier, Object Identifier and Property Identifier of the

		property to be trendlogged.
Log Interval	RW	Specifies the periodic interval in hundredths of seconds for which the referenced property is to be logged when Logging_Type has the value “Polling” (0). Implementation specific granularity of this property is 100 (1 second).
Align Intervals	RW	This property, of type Boolean, specifies whether (TRUE) or not (FALSE) clock-aligned periodic logging is enabled. If clock-aligned periodic logging is enabled and the value of Log_Interval is a factor of (i.e. it divides into without a remainder) a second, minute, hour or day, then the beginning of the period specified for logging shall be aligned to the second, minute, hour or day, respectively.
Interval Offset	RW	Specifies the offset in hundredths of seconds from the beginning of the period specified for logging until the actual acquisition of a log record begins. The offset used shall be the value of Interval_Offset modulo the value of Log_Interval; i.e. if Interval_Offset has the value 31 and Log_Interval is 30, the offset used shall be 1. Interval_Offset has no effect if Align_Intervals is FALSE.
Trigger	RW	Causes the Trend Log object to acquire a log record whenever the value of this property is changed from FALSE to TRUE. It remains TRUE while the Trend Log object is acquiring the data items for a log record. When all data items have been collected or it has been determined that all outstanding data requests will not be fulfilled, the Trend Log object resets the value to FALSE.

17.14 Mapping HPI events to BACnet events

The following HPI events are mapped to BACnet events:

- Sensor events (both from analog and discrete sensors)
- Software events (auditing, logins, logouts)

Events are dispatched to BACnet clients via subscriptions in a Notification Class object. There are N Notification Class objects for each Guardian Management Gateway instance (N is a configuration parameter), with Object Instances = $0..N-1$. The implementation of these objects is provided by the `bacnet-stack` library, its properties are described in the section 17.12.
The instance number of the Notification Class used to dispatch notifications from a given object is specified by the value of the writable object property "Notification Class" (default value of this property is 0).
To start receiving events, the client should create a subscription in the corresponding Notification Class object.
Event structure fields are specified below separately for each event type.
Sensor events from analog sensors (note that both assertion and deassertion events are sent for threshold crossing):

Field Name	Field Value
Process Identifier	PID of the BACnet server process
Notification Class	0
Object Identifier	Object identifier for the corresponding Analog Input object
Timestamp	Current date and time
Event Type	"Out of Range"
Notify Type	"Alarm" for threshold-crossing assertion events "Event" otherwise
Ack Required	FALSE
From State	"Off Normal" if some thresholds were crossed before the event, "Normal" otherwise
To State	"Off Normal" if some thresholds are crossed after the event, "Normal" otherwise
Exceeding Value	The current value of the sensor
Exceeded Limit	The value of the threshold which has been crossed
Status Flags	Flag "In Alarm" is set if and only if "To State" = "Off Normal"; other flags are cleared
Text	A text string indicating in human-readable form, which threshold is exceeded, and including the sensor value and the threshold value

Sensor events from discrete sensors with two states:

Field Name	Field Value
Process Identifier	PID of the BACnet server process
Notification Class	0
Object Identifier	Object identifier for the corresponding Binary Input object
Timestamp	Current date and time
Event Type	"Change of State"
Notify Type	"Alarm" for assertion events with event severity = MINOR, MAJOR or CRITICAL "Event" otherwise
Ack Required	FALSE
From State	"Off Normal" if sensor severity state before the event was MINOR, MAJOR or CRITICAL, "Normal" otherwise
To State	"Off Normal" if sensor severity state after the event is MINOR, MAJOR or CRITICAL, "Normal" otherwise
New State	The index of the new sensor state (BINARY_INACTIVE for the first state, BINARY_ACTIVE for the second state)
Status Flags	Flag "In Alarm" is set if and only if "To State" = "Off Normal"; other flags are

	cleared
Text	Description of the state transition in human-readable form

Sensor events from discrete sensors with more than two states:

FIELD NAME	FIELD VALUE
Process Identifier	PID of the BACnet server process
Notification Class	0
Object Identifier	Object identifier for the corresponding Multi-State Input object
Timestamp	Current date and time
Event Type	“Change of Bit String”
Notify Type	“Alarm” for assertion events with event severity = MINOR, MAJOR or CRITICAL “Event” otherwise
Ack Required	FALSE
From State	“Off Normal” if sensor severity state before the event was MINOR, MAJOR or CRITICAL, “Normal” otherwise
To State	“Off Normal” if sensor severity state after the event is MINOR, MAJOR or CRITICAL, “Normal” otherwise
Referenced Bit String	The current sensor state mask as a bit string
Status Flags	Flag “In Alarm” is set if and only if “To State” = “Off Normal”; other flags are cleared
Text	Description of the state transition in human-readable form

Software events:

FIELD NAME	FIELD VALUE
Process Identifier	PID of the BACnet server process
Notification Class	0
Object Identifier	Object identifier for the Device object
Timestamp	Current date and time
Event Type	“Change of State”
Notify Type	“Event”
Ack Required	FALSE
From State	“Normal”
To State	“Normal”
New State	“Normal”
Status Flags	All flags are cleared
Text	The text portion the original event

17.15 Mapping alarms

The BACnet services Get Alarm Summary and Alarm Acknowledge are implemented by the server and provide direct access to the Guardian Management Gateway alarm table.
In the Get Alarm Summary output, all alarms from one sensor are consolidated into a single BACnet alarm with the alarm state “Transition to Off Normal”, originating from the corresponding BACnet object. The list of consolidated alarms is then provided to the client.
A BACnet alarm is considered unacknowledged, if at least one alarm of those which were consolidated, was unacknowledged.
The fields of the alarm data structure given to the client is shown in the following table.
Alarm fields.

FIELD NAME	FIELD VALUE
Object Identifier	Object identifier for the corresponding Analog Input, Binary Input or Multi-State Input object

Alarm State	“Transition to Off Normal”
Acknowledged Transitions	For the “Transition to Off Normal”: TRUE if all Guardian Management Gateway alarms consolidated into this BACnet alarm are acknowledged, FALSE otherwise For all other transitions: TRUE

The service Alarm Acknowledge is given the following parameters:

- object identifier for the object-originator of the alarm
- the alarm state -must be “Transition to Off Normal”
- the timestamp of the last Get Alarm Summary call.

The server acknowledges all alarms in the Guardian Management Gateway alarm table, associated with the sensor that is mapped to the specified object. However, before doing that, it verifies the timestamp to make sure than no alarm originated after the specified timestamp. If this is not the case, then the client does not have the latest information about alarms, and the request is rejected.

17.16 Mapping the system event log

The Guardian Management Gateway system event log maps straightforwardly to the BACnet event log. The following properties of the BACnet event log object are supported:

Property Name	Access	Property Source
Object Type	RO	25 = “Event Log”
Object Instance	RO	0
Object name	RO	“Event Log”
Description	RO	“Event Log”
Event State	RO	“Normal”
Reliability	RO	“No Fault Detected”
Status Flags	RO	No flags are set
Enable	RW	The “enabled” flag from Guardian Management Gateway SEL Info. Can be turned on and off from BACnet
Stop When Full	RO	TRUE if OverflowAction == DROP in Guardian Management Gateway SEL Info, FALSE otherwise (normally it is FALSE).
Buffer Size	RO	System event log capacity, in records
Record Count	RW	The number of records in the system event log. Can be set to *0* to clear the event log.
Total Record Count	RO	The total number of records added to the system event log, since its creation
Log Buffer	RO	The array of log entries; use the Read Range service to access them

The event log object exposes the event log entries in the property Log Buffer in the form of an array. To access specific event log entries, the client should use the Read Range service. The only supported Read Range request type is “By Position”.

Each event log entry reported to the client consists of the following fields:

Field Name	Field Value
Timestamp	Timestamp from the corresponding Guardian Management Gateway event log entry
Log Datum	Has type “notification”; contains the event that comprises the corresponding Guardian Management Gateway event log entry, translated to the ConfirmedEventNotification object. in accordance with section 17.14

17.17 Mapping the Reinitialize Device service

This service allows a BACnet client to reinitialize the server. The Guardian Management Gateway BACnet server partially implements this service, mapping it to Guardian Management Gateway reboot and restart operations.

This service requires the client to supply a password for the operation. This password is verified by the server and the operation is rejected if the password does not match. For the Guardian Management Gateway BACnet server, this password should be the Guardian Management Gateway password of the user “admin”.
Service operations are mapped according to the following table:

OPERATION	GUARDIAN MANAGEMENT GATEWAY ACTION
Cold Restart	Reboot operation – the Guardian Management Gateway is rebooted
Warm Restart	Restart operation – the Guardian Management Gateway software is restarted
Start Backup	Not implemented, an error is returned
End Backup	Not implemented, an error is returned
Start Restore	Not implemented, an error is returned
End Restore	Not implemented, an error is returned
Abort Restore	Not implemented, an error is returned

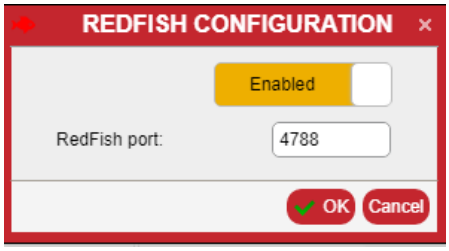
17.18 Supported BACnet services (protocol commands)

The following table lists the BACnet services (protocol commands) which are supported by the Guardian Management Gateway BACnet server (as a responder).

BACNET SERVICE	CONFIRMED/UNCONFIRMED	SUPPORT STATUS
Who Is	Unconfirmed	Supported
Who Has	Unconfirmed	Supported
I Am	Unconfirmed	Supported
Read Property	Confirmed	Supported
Read Property Multiple	Confirmed	Supported
Read Range	Confirmed	Supported
Write Property	Confirmed	Supported
Write Property Multiple	Confirmed	Supported
Reinitialize Device	Confirmed	Partially supported
UTC Time Synchronization	Unconfirmed	Accepted but no action
Time Synchronization	Unconfirmed	Accepted but no action
Device Communication Control	Confirmed	Partially supported
Acknowledge Alarm	Confirmed	Supported
Get Alarm Summary	Confirmed	Supported

18 RedFish Configuration

In the Web interface, RedFish is configured via a dialog box that is accessible via menu items “Device Settings” -> “Network Services” -> “RedFish”.



By default, Redfish port is 4788 and this service is disabled.

19 Security

In this section, the following facilities are described:

- Firewall
- Role-based firewall
- Login restrictions and password policy
- SSL certificate management
- Restricted Service Agreement

19.1 Firewall

This group of settings specifies Linux firewall rules. Firewall settings are separate for the IPv4 and IPv6 protocols, but have the same structure.

The global firewall settings include:

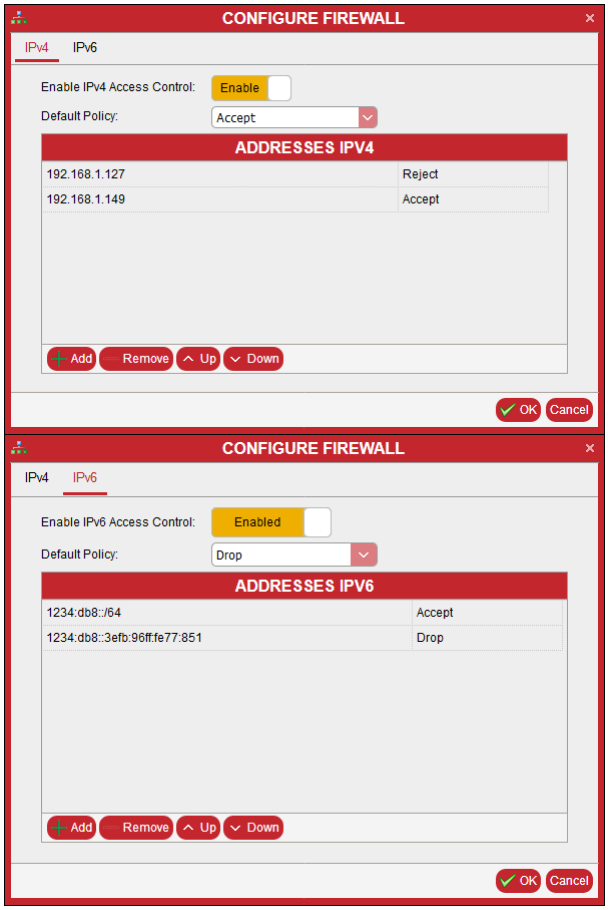
- Enable firewall: true if the firewall is enabled for the given protocol, false otherwise
- Default firewall policy for incoming packets: *ACCEPT* or *DROP* packets

Each rule defines a network address (a host or subnet address) and the policy that applies to the packets originating from this address. The policy can be *ACCEPT*, *REJECT* or *DROP*. The order of rules is significant: for each incoming packet, the rules are examined in that order and the first matching rule determines the policy to be applied. If no rules match, the default policy is applied.

The following operations are defined for the firewall (separately for IPv4 and IPv6):

- Get “firewall enabled” flag
- Set “firewall enabled” flag
- Get default firewall policy (*ACCEPT* or *DROP*)
- Set default firewall policy (*ACCEPT* or *DROP*)
- Get firewall rule by index (index starts from *0*); the information returned includes the network address and policy
- Add firewall rule to the end of the list; the network address and the policy are specified
- Insert firewall rule by index (the new rule is inserted before the rule with index *index*); the network address and the policy are specified
- Modify firewall rule by index (the new rule replaces the rule with index *index*); the network address and the policy are specified
- Delete firewall rule by index

In the Web interface, the global firewall is configured via a dialog box that is accessible via menu items “Device Settings”->”Security”->”Firewall”. There are two tabs in this dialog box: “IPv4” and “IPv6”.


CONFIGURE FIREWALL
IPv4
IPv6
Enable IPv4 Access Control:
Enable
Default Policy:
Accept
ADDRESSES IPV4
192.168.1.127
Reject
192.168.1.149
Accept
Add
Remove
Up
Down
OK
Cancel
CONFIGURE FIREWALL
IPv4
IPv6
Enable IPv6 Access Control:
Enabled
Default Policy:
Drop
ADDRESSES IPV6
1234:db8::/64
Accept
1234:db8::3efb:96ff:fe77:851
Drop
Add
Remove
Up
Down
OK
Cancel

19.2 Login restrictions and password policy

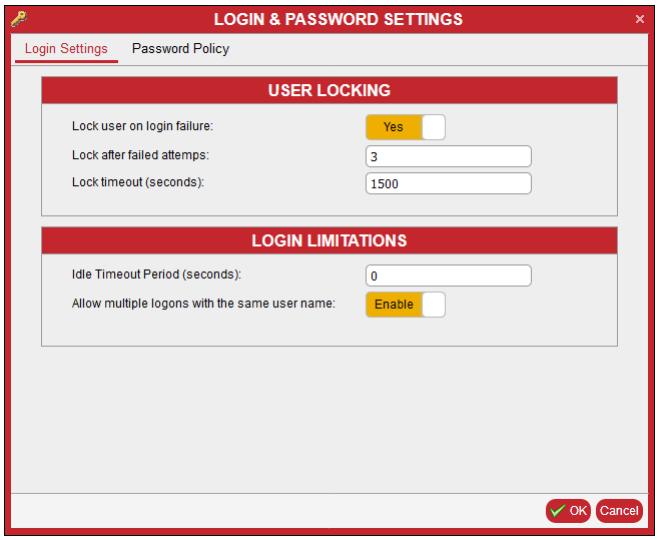
This set of options specifies requirements to password complexity, password aging and login security. The following options exist:

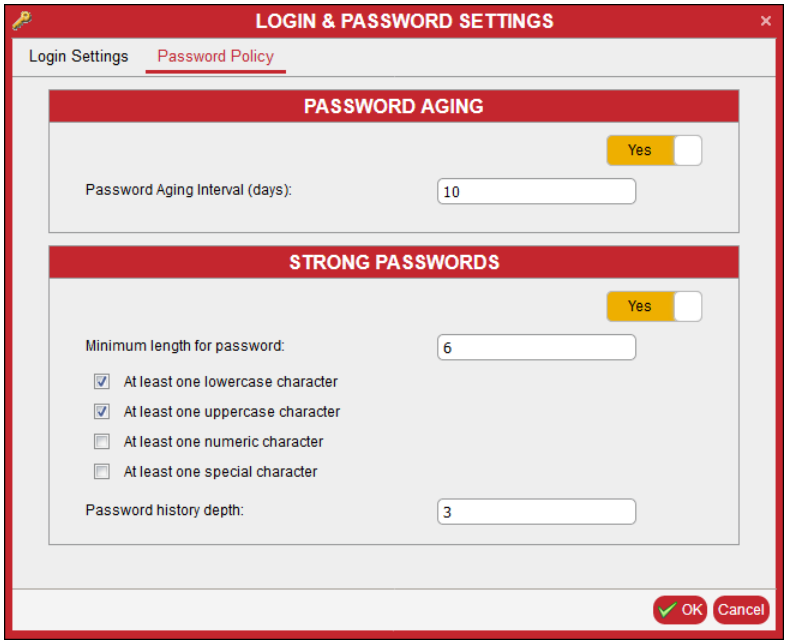
- AllowMultipleLogons – *TRUE* if multiple logons with the same user name are allowed, *FALSE* otherwise
- LockAfterFailedAttempts – lock a user (prevent from login) after this number of failed logon attempts, for a certain time
- LockTime – the number of seconds for which the user is locked
- IdleTimeout – the number of seconds; if a user is inactive for this number of seconds, he/she is logged off automatically. Value *0* turns off this feature.
- PasswordAging – *TRUE* if password aging is enabled (logon passwords expire after some time and need to be changed after that)
- PasswordAgingInterval – the password aging interval, in days
- PasswordHistoryDepth – the system refuses to assign a new password that matches one of the most recent passwords for the user; this parameter specifies how many most recent passwords the system remembers. Value *0* turns off this feature.
- StrongPasswords – *TRUE* if strong passwords are enforced (the properties of strong passwords are given by subsequent options), *FALSE* otherwise
- MinStrongPasswordLength – minimum length of a strong password, in characters
- AtLeastOneLcCharacter – *TRUE* if a strong password must contain at least one lowercase character, *FALSE* otherwise
- AtLeastOneUcCharacter – *TRUE* if a strong password must contain at least one uppercase character, *FALSE* otherwise
- AtLeastOneNumCharacter – *TRUE* if a strong password must contain at least one numeric character, *FALSE* otherwise
- AtLeastOneSpecCharacter – *TRUE* if a strong password must contain at least one special (punctuation) character, *FALSE* otherwise.

The following operations are defined for the login restrictions and password policy:

- Get login restrictions and password policy (all options)
- Set login restrictions and password policy (all options)
- Check if the specified user is currently locked out
- Unlock the specified user

In the Web interface, the login restrictions and password policy are configured via a dialog box that is accessible via menu items “Device Settings” -> “Security” -> “Login Settings & Password Policy”. There are two tabs in this dialog box: “Login Settings” and “Password Policy”.





19.3 Role-based firewall

This group of settings specifies rules for the role-based firewall. This firewall allows or denies logins for specific users from specific IP address ranges. Firewall settings are separate for the IPv4 and IPv6 protocols, but have the same structure.

The global role-based firewall settings include:

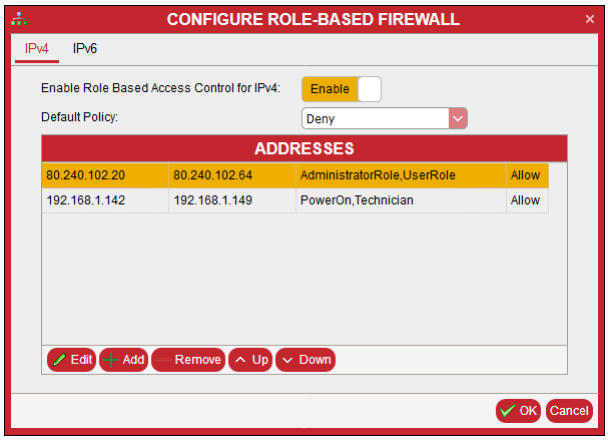
- Enable role-based firewall: true if the role-based firewall is enabled for the given protocol, false otherwise
- Default role-based firewall policy: *ALLOW* or *DENY* login

Each rule defines a range of network addresses (IPv4 or IPv6 addresses), the list of roles and the policy that applies to the login attempt of a user belonging to one of the specified roles, from an IP address belonging to the specified range. The policy can be *ALLOW* or *DENY*. The order of rules is significant: for each login attempt, the rules are examined in that order and the first matching rule determines the policy to be applied. If no rules match, the default policy is applied.

The following operations are defined for the role-based firewall (separately for IPv4 and IPv6):

- Get “role-based firewall enabled” flag
- Set “role-based firewall enabled” flag
- Get default role-based firewall policy (*ALLOW* or *DENY*)
- Set default role-based firewall policy (*ALLOW* or *DENY*)
- Get role-based firewall rule by index (index starts from *0*); the information returned includes the starting and ending IP address, the list of roles and the policy
- Add role-based firewall rule to the end of the list; the starting and ending IP address, the list of roles and the policy are specified
- Insert role-based firewall rule by index (the new rule is inserted before the rule with index *index*); the starting and ending IP address, the list of roles and the policy are specified
- Modify role-based firewall rule by index (the new rule replaces the rule with index *index*); the starting and ending IP address, the list of roles and the policy are specified
- Delete role-based firewall rule by index

In the Web interface, the global role-based firewall is configured via a dialog box that is accessible via menu items “Device Settings”->”Security”->”Role-Based Firewall”.



19.4 SSL Certificate Management

An SSL certificate is a file that is needed for secure HTTP (HTTPS) access to the Guardian Management Gateway; this file is issued by some certificate authority and confirms the identity of a specific HTTPS server, in our case, this is the identity of the Guardian Management Gateway. This file should be installed in a specific location on the Guardian Management Gateway, then it becomes an active certificate and can participate in the secure communication.

By default, the Guardian Management Gateway uses a self-signed certificate, which is generated automatically when the network configuration is changed (refer to 19.4.1). However, it is the user's responsibility to install a certificate that is specific to the company and domain/host name used by each customer (if secure HTTP communication with the Guardian Management Gateway is needed).

There are three kinds of certificates and certificate-related objects that Guardian Management Gateway software can deal with:

- A certificate signed by a certificate authority (CA). This certificate must come from outside, and can be downloaded on the Guardian Management Gateway, stored there and installed as the active certificate
- A self-signed certificate. This certificate is generated on the Guardian Management Gateway and can be stored there and installed as the active certificate. When the active certificate is a self-signed one, Web browsers normally issue a warning when establishing an HTTPS session with the target server; the Web user must acknowledge the security risks to continue the communication
- A certificate sign request (CSR). This is the file that is generated on the Guardian Management Gateway and must be sent to a certificate authority to obtain a valid certificate. This file contains necessary information about the Guardian Management Gateway, its location and ownership.

The set of operations that deal with certificates is different between CLI and Web interface. This is because CLI is executed locally on the Guardian Management Gateway, while the Web client is remote to the Guardian Management Gateway.

With the CLI interface, the following certificate-related operations are supported:

- Generate a self-signed certificate or a certificate sign request. The resulting file is stored in the specified location on the Guardian Management Gateway (by default, this is the user's home directory) or can be copied to a remote SCP location
- Show the list of existing certificates and CSRs in the specified directory (by default, in the user's home directory)
- Show the details of the specified certificate or CSR file
- Show the details of the active certificate
- Delete the specified local certificate or CSR file
- Copy a certificate or a CSR file to a remote SCP location or from a remote SCP location
- Install the specified certificate (a local file or a remote file accessible via SCP) as the active certificate.

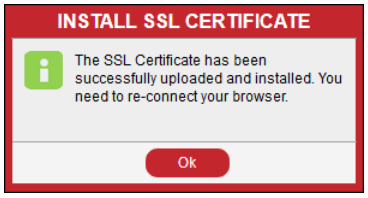
With the Web interface, the following certificate-related operations are supported:

- Copy a certificate from the client system to the Guardian Management Gateway and install it as the active certificate
- Generate a certificate sign request on the Guardian Management Gateway and copy it to the client system
- Generate a self-signed certificate on the Guardian Management Gateway and install it as the active certificate
- Show the details of the active certificate.

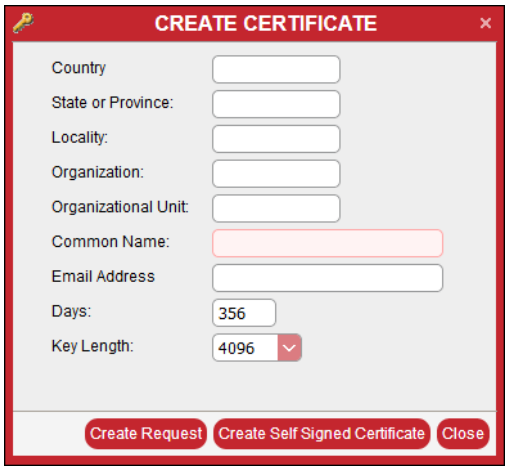
In the Web interface, certificate-related operations are implemented as follows:

To copy a certificate from the client system to the Guardian Management Gateway and install it, use the dialog "Install SSL Certificate", which is invoked with the menu command "Device Settings" -> "Security" -> "SSL Certificate".

The user should select the local file with the certificate and start the upload by pressing the “OK” button. When the certificate is successfully uploaded, the following window is generated.



To create a new SSL certificate or a CSR use the menu command “Device Settings” -> “Security” -> “Create SSL Certificate”. There are two buttons in the window “Create certificate”: “Create Request and “Create Self Signed Certificate”. A two-letter country code (ISO 3166-1 alpha-2 standard) or three-letter country code (ISO 3166-1 alpha-3 standard) should be written in the “Country” field.



19.4.1 **Default SSL Certificate**

When a user-specific certificate is not installed, the Guardian Management Gateway uses a default self-signed certificate, which is generated automatically when the network configuration is changed. This certificate is bound to the domain name assigned to the Guardian Management Gateway device, and also to its IP address (as an alternative name). To avoid security warnings, this certificate can be downloaded and added to the list of trusted certificates in the browser. However, for maximum security, it is highly recommended that customers generate and install their own certificates (preferably signed by a certification authority) that use the correct company name, country, and other fields.

19.5 Restricted Service Agreement

A restricted service agreement (a special security banner) can be shown to a user during the logon, both in CLI and Web interface. In addition, the restricted service agreement can be enforced, which means that the user should explicitly acknowledge it in order to be able to log in.

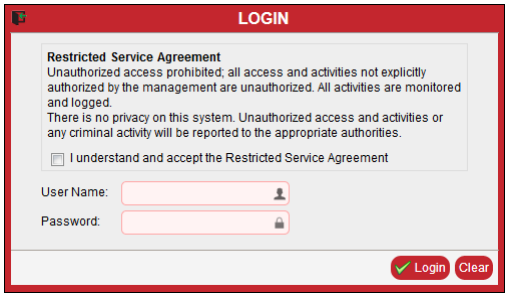
The following attributes are specified for the restricted service agreement:

- The text of restricted service agreement
- Enforce flag (*TRUE*/*FALSE*)

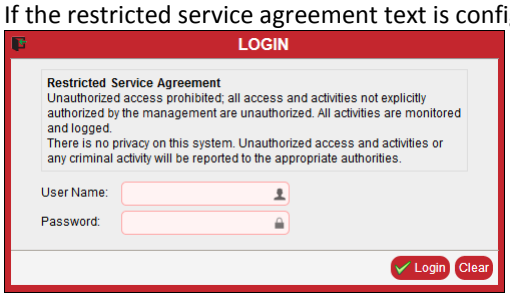
If the restricted service agreement text is configured and it is enforced, the following dialog will be shown during CLI logon:

```
SGP Command Line Interpreter
RESTRICTED SERVICE AGREEMENT

---------- ------- ---------

Unauthorized access to this system is prohibited; all access and activities not
explicitly authorized by management are unauthorized. All activities are
monitored and logged.


Do you accept the restricted service agreement (y/n)? y

Connection from 80.240.102.63 as testuser
Current language: English
CLI{testuser}>
```

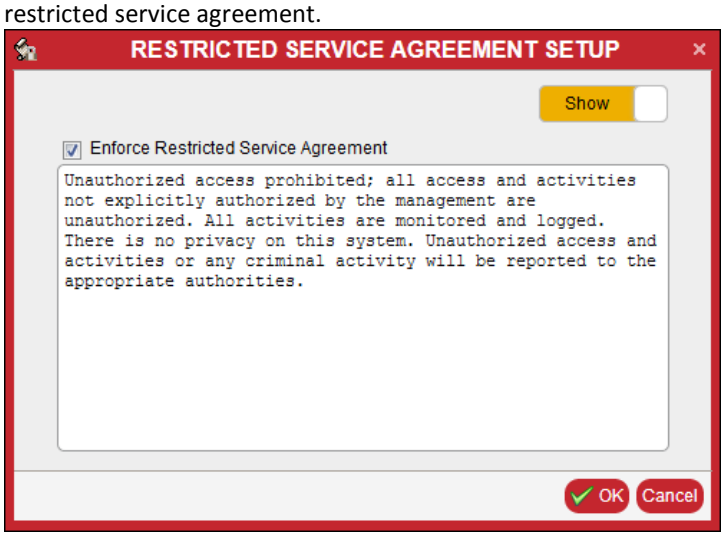
In the Web interface, the following dialog is shown:



If the restricted service agreement text is configured and it is not enforced, the following dialog is shown:



To configure the restricted service agreement with the Web interface, use the menu command “Device Settings” -> “Security” -> “Restricted Service Agreement Banner”. If the switch button “Show/Not used” is set to “Show”, the restricted service agreement is shown at every logon. The “Restricted Service Agreement Setup” window contains the checkbox “Enforce Restricted Service Agreement”. It corresponds to the enforce flag. The text area in the window contains the full text of the restricted service agreement. A user with sufficient privileges can edit the text of the restricted service agreement.



20 Events and Actions

Events are used to notify about state changes in various Guardian Management Gateway subsystems. All generated events are stored in the event log, most events are generated by sensors.

- Threshold-based sensors generate events when thresholds are crossed
- Discrete sensors generate events when sensor state changes

An important feature is that it's possible to define event filters and periodic rules.

- Event filters allow to send messages, SNMP notifications and perform device control functions in response to certain events.
- Periodic rules allow for automatic device control based on sensor reading values, control states and alarms.

To generate an event, the assertion/deassertion event must be enabled!

Example:

EVENT STATES & THRESHOLDS

Hysteresis: **Linear** Positive: 0 Negative: 0

Threshold	Supported	Events		Value
		Assertion	Deassertion	
UPPER CRITICAL	+	+		70.0
UPPER MAJOR	+	+	+	60.0

When assertion and deassertion is enabled, an event is generated for the threshold crossings in both directions. In this example for a temperature sensor, an event is generated when the temperature exceeds 60 degrees, and the next event is generated when the temperature exceeds 70 degrees. When the temperature falls below 70 degrees no event is generated because deassertion for the Upper Critical threshold is not enabled.

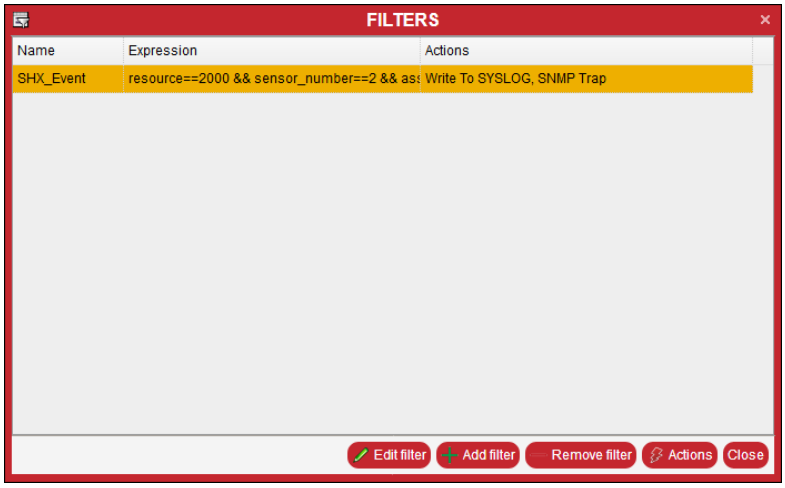
20.1 Event Filters

Event filters allow the user to trigger specified actions such as: send messages, SNMP notifications and perform device control functions to events. Each filter consists of a rule defined by an expression and one or more actions. When an event is generated, the filter expression is evaluated and, if the result is non-zero, the actions belonging to this filter are executed.

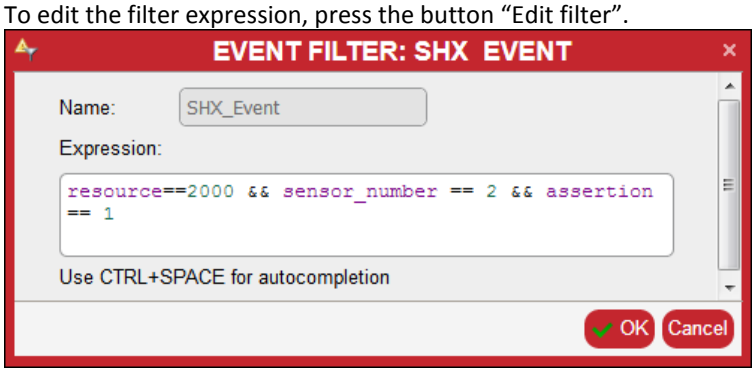
If the filter list consists of several filters, at an event the entire list is walked through and all filter expression are evaluated and the resp. actions are executed.

Expressions are evaluated in units defined by the global settings.

The window below can be accessed by selecting the menu item “Device Settings” -> “Rules and Actions”-> “Event Rules”.



To add a new filter, press the button “Add filter”. Visual Expression Builder features can be invoked for event rules (filter expressions) via the CTRL+Space key combination.
To edit the filter expression, press the button “Edit filter”.



20.2 Actions

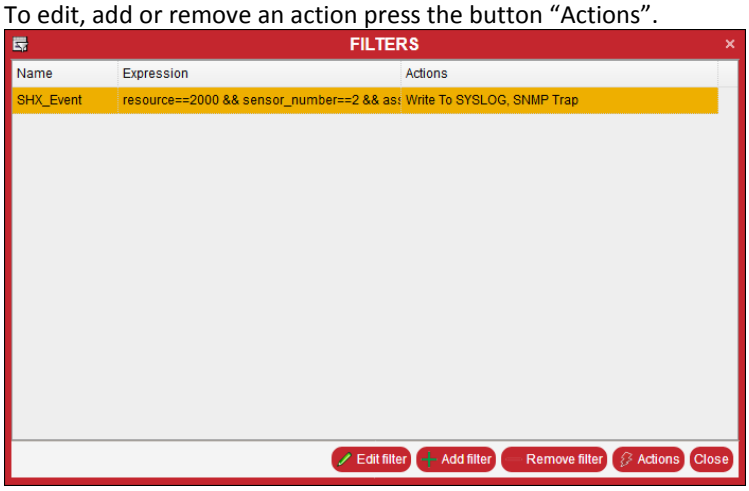
Each action in a filter (or in a periodic rule) has a “disposition” parameter. The following dispositions are defined:

- “Always” – the action is always executed.
- “If successful” – the action is executed only if execution of the previous action in the list was successful.
- “If unsuccessful” – the action is executed only if execution of the previous action in the list was unsuccessful.

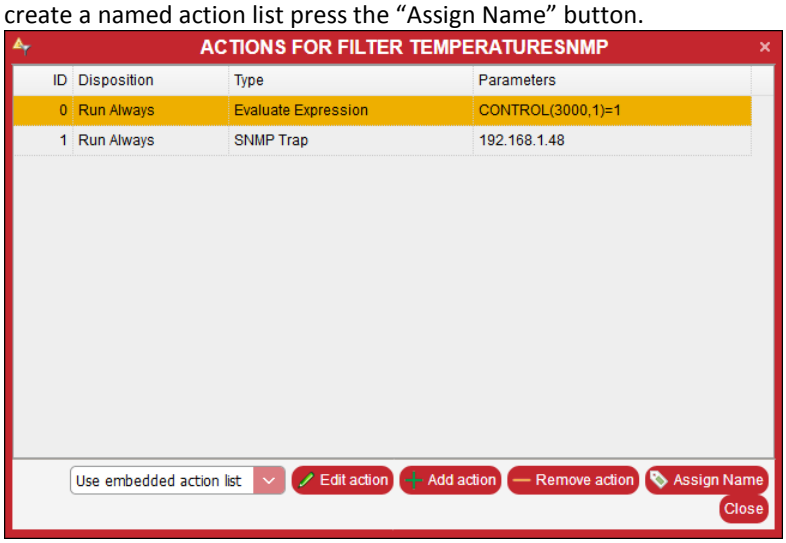
There are several types of actions, they include:

- “Expression” – evaluate an expression (likely with a side effect, e.g. assign a value to a control).
- “Command” – run a CLI command on the Guardian Management Gateway (only an administrator user can add actions of this type).
- “Syslog” – log information about the event into the Linux system log on the Guardian Management Gateway.
- “Send mail” – send an e-mail with the information about the event, via the preconfigured SMTP server. The list of recipients and the subject are the parameters of the action.
- “SNMP Trap” – send an SNMP trap (notification) to the previously specified target IP address.
- “Turn Cooling On” – turn on environment cooling, using a previously specified SHX cooling device.
- “Turn Cooling Off” – turn off environment cooling, using a previously specified SHX cooling device.
- “Max Cooling” – set maximum environment cooling, using a previously specified SHX cooling device.
- “Publish MQTT” - If the Guardian Management Gateway is IoT-enabled, for each event, a MQTT message with the information about the event is sent to IoT cloud.

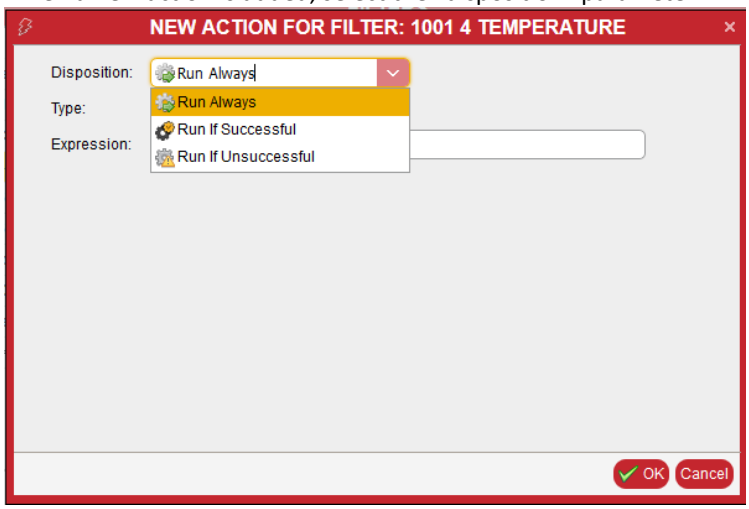
To edit, add or remove an action press the button “Actions”.



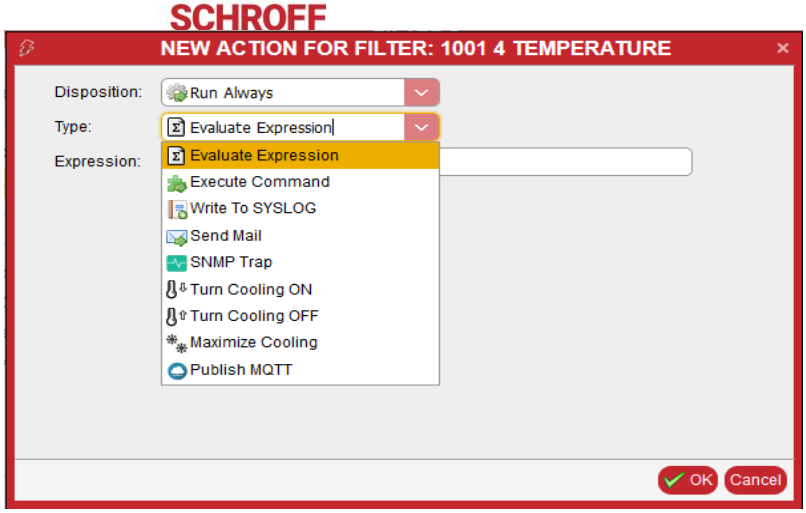
To add a new action press the button “Add action”. To edit an action select the entry in the action list and press the button “Edit action”. To remove an action, select the entry in the action list and press the button “Remove action”. To create a named action list press the “Assign Name” button.



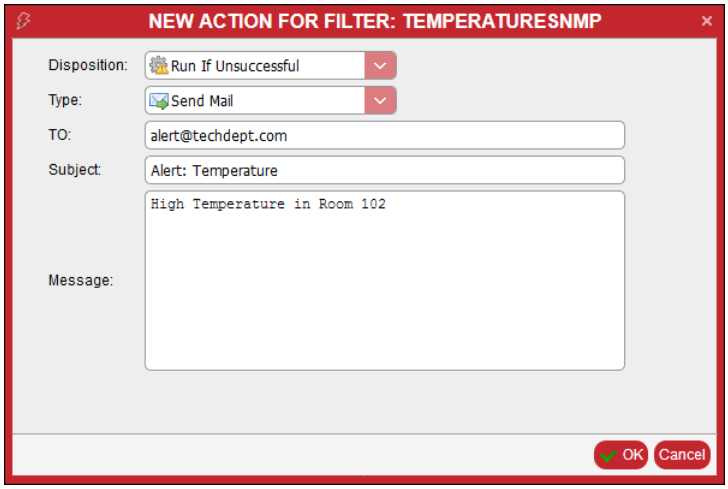
When a new action is added, select the “disposition” parameter:



Then, select the action type:



Then, fill fields that are specific for the selected type. An example for “Send Mail” type is below:



The following operations exist for event filters and actions in event filters:

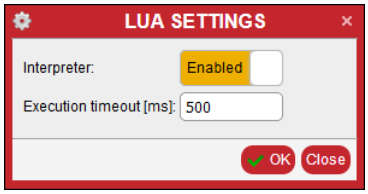
- Create an event filter, specifying its name and filter expression
- Delete an event filter by name
- Enumerate existing event filters
- Get event filter expression by event filter name
- Add an action to the event filter
- Enumerate actions for the given event filter
- Update a specific action for the given event filter
- Remove a specific action from the given event filter
- Assign a named action list to the given event filter
- Remove a named action list from the given event filter.

20.3 Lua interpreter

To create, edit, upload, download and delete Lua script files select the “Device Settings” -> “Rules and Actions” -> “Lua Interpreter” menu item.

When a Lua script file is saved or uploaded, the syntax of the file is checked for correctness.

To change the Lua interpreter settings, press the “Configure” button and the dialog window “Lua settings” is generated.



The official web site of the Lua language is http://www.lua.org.

20.4 Expressions

Guardian Management Gateway allows expressions to be specified as event filtering criteria and as event actions. Also an expression can be directly evaluated by the CLI command `expression`. These expressions conform to certain syntax, similar to the syntax of arithmetic and logical expressions in the C and Java programming languages.
For event filtering, the expression is evaluated and the result determines whether the event passes the filter (the event passes if the result is not `0`).
For actions, the expression is evaluated and the result is ignored. The expression normally has some side effects (e.g. assignment to some variable or control).
For the CLI command `expression`, the expression is evaluated and the result is printed on the CLI console.

20.4.1 Value Types

The result of an expression evaluation is a value, which has a type. A value type can be "integer number", "real number" or "string". Boolean values are represented as integer numbers, `1` represents `TRUE`, `0` represents `FALSE`. Both integer and real numbers have 64 bits in size.

20.4.2 Expression Structure

The expression consists of terms connected with operators. Terms include special names, variables, integer, real and string constants, sensor and control designators, Lua script invocations.

20.4.3 Special Names

Special names designate certain fields in the event that is currently subject to filtering. In the action expressions, these special names designate the fields of the event on which the action is invoked.
In alarm subexpressions (parameters to the functions `alarm_exists` and `alarm_count`) the special names designate the fields of the alarm table entries.
Special names are defined in the following tables, they are case insensitive (i.e. `sensor_number`, `Sensor_Number`, or `SENSOR_NUMBER` variants can be used).

Table 1: Event-related special names

NAME	TYPE	DESCRIPTION
assertion	Integer	`0` for deassertion events, `1` for assertion events
event_category	Integer	The event category, according to HPI definition (e.g. `1` for threshold events, `2` for usage state events, etc.)
resource	Integer	The resource ID which sourced the event. This value is `-1` if not applicable.
is_fumi	Integer	`1` if the event is an HPI FUMI (upgrade-related) event, `0` otherwise
is_sensor	Integer	`1` if the event is originated by a sensor, `0` otherwise
sensor_number	Integer	The number of the sensor that originated the event. This value is `-1` if not applicable
managed_sensor	Integer	The number of the managed sensor that corresponds to the physical sensor that originated the event. This value is -1 if not applicable
sensor_state	Integer	The single sensor state that, asserted or deasserted, caused the event; it is represented as a bit mask with a single bit set. This value is `0` if not applicable
sensor_type	Integer	The type of the sensor that originated the event, according to HPI definition (e.g. `1` for Temperature sensors, `2` for Voltage sensors, etc). A string value is returned in the contexts that allow string values (e.g "Temperature", "Voltage")
severity	Integer	Event severity according to HPI definition (`0` for Critical, `1` for Major, `2` for Minor, `3` for Informational, `4` for OK). A string value is returned in the contexts that allow string values: "Critical", "Major, "Minor", "Informational" and "OK".
upper_critical	boolean	True if the event indicates crossing of the corresponding threshold (including deassertion events), false otherwise.

Name	Type	Description
upper_major	boolean	True if the event indicates crossing of the corresponding threshold (including deassertion events), false otherwise.
upper_minor	boolean	True if the event indicates crossing of the corresponding threshold (including deassertion events), false otherwise.
lower_critical	boolean	True if the event indicates crossing of the corresponding threshold (including deassertion events), false otherwise.
lower_major	boolean	True if the event indicates crossing of the corresponding threshold (including deassertion events), false otherwise.
lower_minor	boolean	True if the event indicates crossing of the corresponding threshold (including deassertion events), false otherwise.

Table 2: Alarm-related special names

Name	Type	Description
acknowledged	Integer	*1* if the alarm is acknowledged, *0* otherwise
event_category	Integer	The event category of the alarm, according to HPI definition (e.g. *1* for threshold events, *2* for usage state events, etc.)
resource	Integer	The resource ID which sourced the event that caused the alarm. This value is *−1* if not applicable.
is_sensor	Integer	*1* if the alarm is caused by a sensor event, *0* otherwise
sensor_number	Integer	The number of the sensor that originated the event. This value is *−1* if not applicable
sensor_state	Integer	The single sensor state that, asserted or deasserted, caused the alarm; it is represented as a bit mask with a single bit set. This value is *0* if not applicable
sensor_type	Integer	The type of the sensor that originated the event, according to HPI definition (e.g. *1* for Temperature sensors, *2* for Voltage sensors, etc.). A string value is returned in the contexts that allow string values (e.g. “Temperature”, “Voltage”)
severity	Integer	Alarm severity according to HPI definition (*0* for Critical, *1* for Major, *2* for Minor, *3* for Informational, *4* for OK). A string value is returned in the contexts that allow string values: “Critical”, “Major, “Minor”, “Informational” and “OK”.

20.4.4 **Variables**

Variable names start with *$* and further consist of alphanumeric characters. They are case insensitive (e.g. *$var1*, *$Var1* and *$VAR1* designate the same variable). The variables are global variables that are created when they are first referenced; the variable value is integer *0* at this point. Values of variables can be integer numbers, real numbers or strings. The type is associated with the value, not with the variable.
Variable values set in one expression are preserved after the evaluation of this expression is complete and can later be used in other expressions.

20.4.5 **Sensor items**

A sensor item has the following syntax:

```
sensor-item ::= "SENSOR" "(" resource-id "," sensor-number ")" [ "." sensor-item-tail ]
sensor-item-tail ::= "UCR" | "UMJ" | "UMN" | "LCR" | "LMJ" | "LMN" | "FAILED" | "INITIAL_UPDATE" | state-number
```

The value of a sensor item is calculated as follows:
If `sensor-item-tail` is omitted, the value is the numeric sensor value.
If a `sensor-item-tail` is present, the value is of the item is *1* or *0*, depending on whether the sensor is in the specified state. The states `UCR`, `UMJ`, `UMN`, `LCR`, `LMJ`, `LMN` indicate whether the sensor is beyond the corresponding threshold. The state `FAILED` indicates whether the sensor reading has failed. The state `INITIAL_UPDATE` indicates

whether the numeric value of the sensor is not available. A state number indicates whether the corresponding state is set in the sensor event state mask.

Table 3: Aliases for threshold names in expressions

THRESHOLD NAME	NAME IN EXPRESSION	ALIAS	ALIAS
Upper Critical	UCR	UCRIT	UNR
Upper Major	UMJ	UMAJ	UC
Upper Minor	UMN	UMIN	UNC
Lower Critical	LCR	LCRIT	LNR
Lower Major	LMJ	LMAJ	LC
Lower Minor	LMN	LMIN	LNC

20.4.6 **Control items**

A control item has the following syntax:

```
control-item ::= "CONTROL" "(" resource-id "," control-number ")"
```

The value of a control item is numeric; it's the result of the "get" operation applied to the corresponding control. For digital controls, the result is *1* if the control is in the *ON* state and *0* if it is in the *OFF* state.

Control items can be targets of an assignment operation. Assigning a value to a control means setting the control to this value (and to "manual" mode in the HPI sense). For digital controls, assigning *1* sets the control to the *ON* state, assigning *0* sets the control to the *OFF* state.

20.4.7 **Constants**

Integer and real constants have usual representation (e.g. *25*, *2.5*). String constants are enclosed in double quotes (e.g. "string"). The value of a constant is this constant.

20.4.8 **Lua script invocation**

A Lua script invocation has the following syntax:

```
lua_item ::= "lua" "(" filename ")";
```

The *filename* term is a filename with *lua* extension. A Lua script file is to be either created or uploaded via the "Lua Scripts" dialog, which can be accessed by selecting the "Device Settings" -> "Rules and Actions" -> "Lua Interpreter" menu item.

20.4.9 **Operators**

The following table lists all operators, with their arity and priority for binary operators:

Table 4: Operators

OPERATOR	ARITY	PRIORITY	DEFINITION
!	1		*NOT operator*. The operand must be numeric. The result is `1` if applied to `0` and `0` if applied to any non-zero value.
~	1		*Complement operator*. The operand must be numeric. The result is a bit-wise complement of the operand.
−	1		*Negation operator*. The operand must be numeric. The result is the operand subtracted from `0`.
*	2	1	*Multiplication*. The operands must be numeric. The result is the product of the operands. If one of the operands is a real number, the result is a real number.
/	2	1	*Division*. The operands must be numeric. If one of the operands is a real number, the result is a real number, otherwise the operation is integer division.
%	2	1	*Remainder*. The operands must be integer. The result is the remainder of division of the first operand by the second operand.
+	2	2	*Addition*. For numeric operands, the result is the sum of the operands. If one of the operands is a real number, the result is a real number. This operation is also applicable to string values and yields their concatenation.
−	2	2	*Subtraction*. The operands must be numeric. The result is the difference of the operands. If one of the operands is a real number, the result is a real number.
<<	2	3	*Left shift*. The operands must be numeric. The result is the result of the left shift of the first operand by the number of bits specified by the second operand.
>>	2	3	*Right shift*. The operands must be numeric. The result is the result of the right shift of the first operand by the number of bits specified by the second operand.
==	2	4	*Equal*. Compares the two operands for equality and yields `1` if they are equal and `0` if they are not equal. String values can also be compared; if one of the operands is a string value, and the other is not, the other value is converted to a string.
!=	2	4	*Not Equal*. Compares the two operands for inequality and yields `1` if they are not equal and `0` if they are equal. String values can also be compared; if one of the operands is a string value, and the other is not, the other value is converted to a string.
<	2	4	*Less*. Compares the two operands and yields `1` if the first operand is less than the second operand and `0` otherwise. String values can also be compared; if one of the operands is a string value, and the other is not, the other value is converted to a string.
>	2	4	*Greater*. Compares the two operands and yields `1` if the first operand is greater than the second operand and `0` otherwise. String values can also be compared; if one of the operands is a string value, and the other is not, the other value is converted to a string.
<=	2	4	*Less or Equal*. Compares the two operands and yields `1` if the first operand is less or equal than the second operand and `0` otherwise. String values can also be compared; if one of the operands is a string value, and the other is not, the other value is converted to a string.

Operator	Arity	Priority	Definition
>=	2	4	*Greater or Equal*. Compares the two operands and yields `1` if the first operand is greater or equal than the second operand and `0` otherwise. String values can also be compared; if one of the operands is a string value, and the other is not, the other value is converted to a string.
&	2	5	*Bitwise AND*. The operands must be numeric. The result is the result of the bitwise AND of the two operands. The type of the result is integer.
|	2	6	*Bitwise OR*. The operands must be numeric. The result is the result of the bitwise OR of the two operands. The type of the result is integer.
^	2	6	*Bitwise XOR*. The operands must be numeric. The result is the result of the bitwise XOR of the two operands. The type of the result is integer.
&&	2	7	*Logical short-circuit AND*. The operator evaluates the first operand, and if the result is 0, it yields 0 and does not evaluate the second operand. Otherwise, it evaluates the second operand and returns the resulting value as the result of the whole operation.
||	2	8	*Logical short-circuit OR*. The operator evaluates the first operand, and if the result is not `0`, it yields this result as the result of the whole operation and does not evaluate the second operand. Otherwise (if the first operand evaluates to `0`), it evaluates the second operand and returns the resulting value as the result of the whole operation.
=	2	9	*Assignment*. This operator is right-associative. The first operand (assignment target) must be a variable or a control item. The operator evaluates the second operand and assigns the resulting value to the assignment target and yields it as the result of the operation (allowing chained assignments)
?: IF...THEN...ELSE	3	10	This is the *conditional operator*, can be represented in the form `a ? b : c` or `IF a THEN b ELSE c`. First `a` is evaluated, then depending on the value of `a` (non-zero or `0`), subexpression `b` or `c` respectively is evaluated and the corresponding value is returned.
,	2	11	*Comma operator*. The first operand is evaluated, the value is thrown away, and then the second operand is evaluated and its value is the value of the whole operation.

20.4.10 Alarm-related functions

Alarm-related functions `alarm_exists` and `alarm_count` return information about the presence of certain entries in the alarm table. Both functions take one argument, which is a predicate expression evaluated over all alarm table entries. Special names in this expression refer to the fields in the alarm table entry.

The function `alarm_exists` returns `TRUE` if the predicate returns `TRUE` for at least one entry, `FALSE` otherwise.

The function `alarm_count` returns the number of alarm table entries for which the predicate returns `TRUE`. For example, `alarm_count(1)` returns the total number of entries in the alarm table.

20.4.11 Aggregate functions

These functions implement aggregate operations on groups. Values of all sensors in the group are evaluated and aggregated according to the specific function. All these functions have a single parameter that should be a group name. The functions, their return types and their semantics are listed in the table below:

Table 5: Aggregate functions

Function name	Type	Description
count	Integer	The number of sensors that return valid readings.
total	Real	Sum of readings of all sensors in the group.
minimum	Real	The minimal reading among all sensors in the group.
maximum	Real	The maximal reading among all sensors in the group.

Function name	Type	Description
average	Real	The average reading among all sensors in the group (total() divided by count())
square_total	Real	Sum of squares of readings of all sensors in the group.
dispersion	Real	The dispersion of readings of all sensors in the group.
state_count	Integer	The number of sensors in the group that return state mask.
state_and	Integer	The aggregate AND of state masks for all sensors in the group.
state_or	Integer	The aggregate OR of state masks for all sensors in the group.
state_xor	Integer	The aggregate XOR of state masks for all sensors in the group.

20.4.12 Floating-point functions

Table 6: Floating-point functions

Function name	Type	Description
asfloat32	Real	This function evaluates its argument, treating it as a 32-bit representation of a floating-point number (C type float), returns this number.
asfloat64	Real	This function evaluates its argument, treating it as a 64-bit representation of a floating-point number (C type double), returns this number.

These functions are needed for Modbus JSON drivers, but may be used in other contexts as well.

20.4.13 min() and max()

Function name	number of arguments	Description
min	2	Compares the two arguments and yields the first argument if the first argument is less than the second one, and the second argument otherwise.
max	2	Compares the two arguments and yields the first argument if the first argument is greater than the second one, and the second argument otherwise.

20.4.14 Proportional-integrative-derivative (PID) control algorithm

The proportional-integrative-derivative (PID) control algorithm is implemented by the `PID()` function. This function may be used in expressions. This algorithm might be used for cooling management or other environmental management algorithms.

The format of the call is the following:

```
result = PID(setpoint, input, $error, $integral, $deriv, Kp, Ki, Kd[, dt=1.0])
```

All arguments are real and the result is real:
`setpoint` - the desired setpoint value, may be a value of a control (e.g. on SHX30 the temperature setpoint is control #2, e.g. `CONTROL(2000,2)`)
`input` - the input value for the controlled characteristic, usually a sensor value (e.g. `SENSOR(2000,2)` for the actual temperature in the cooler, or can be a combination of several sensor values)
`$error` - a variable that contains intermediate error value, must be a variable name (start with `$`)
`$integral` - a variable that contains intermediate integral value, must be a variable name (start with `$`)
`$deriv`- a variable that contains intermediate derivative component value, must be a variable name (start with `$`)
`Kp` - the proportional coefficient (a number or expression)
`Ki` - the integral coefficient (a number or expression)
`Kd` - the derivative coefficient (a number or expression)
`dt` – the loop interval time ((a number or expression)), it is an optional argument, the default value is *1.0*.

The result can be assigned to some control to set the value that controls the process. The function is intended to be used in periodic expressions.

Example of a periodic expression:
`control(2000,3) = 30 + pid( 0, sensor(2000,8), $a, $b, $c, 0.3, 0.001, 0.01)`

sets the value of the fan speed control on SHX30 trying to make the value of sensor 8 "Temp. air outlet average" as close to *0* as possible.

The coefficients are: *Kp=0.3*, *Ki=0.001*, *Kd=0.01*.

Variables *$a*, *$b*, *$c* store the historical data and change after each call of the `PID()`. These variables are lost after a restart (or reboot) of the Guardian Management Gateway.

20.5 Examples for event filtering expressions

The following expression can be used for an event filtering.
`resource==1000 && sensor_number==1 && assertion==1`
If an assertion event is generated by sensor #1 at resource 1000, the event passes the filter, → an action is triggered.

- The special names *resource*, *sensor_number*, *assertion* are defined in Table 1.
- The operators *==*, *&&* are defined in Table 4.

Sensor 1 at resource 1000 is usually a temperature sensor on a 1-wire sensor box.

The following expression can be used for an event filtering.
`SENSOR(3000,2).UCR && sensor_type==2`
An event passes the filter if the type of the sensor that originated the event is "Voltage" and the sensor #2 at resource 3000 is beyond the Upper Critical threshold. The special name *sensor_type* is defined in Table 1. The operators *==*, *&&* are defined in Table 4.
Sensor 2 at resource 3000 is normally an MCB 12V sensor.

The following expression makes use of an alarm-related function.
`SENSOR(3000,1)> 25. && alarm_exists(is_sensor && resource == 4002 && acknowledged == 0)`
The expression evaluates to *TRUE* if the sensor reading of sensor #1 at resource 3000 exceeds 25.0 and there is an unacknowledged alarm caused by a sensor event, and the resource ID that sourced the sensor event is 4002. The special names *is_sensor*, *resource*, *acknowledged* are defined in Table 2. The operators *==*, *&&*, *>* are defined in Table 4.
Sensor 1 at resource 3000 is normally an MCB Temperature sensor.

The following expression makes use of an aggregate function.
`IF total("SHX COOLER") > 70 THEN CONTROL(3000,1)=1 ELSE CONTROL(3000,1)=0`
The expression evaluates the sum of sensor readings of all the sensors in the group 'SHX COOLER' and compares it to *70*. It is supposed that the group contains at least one sensor. If the comparison holds *TRUE*, the control #1 at resource 3000 is set the *ON* state. If the comparison holds *FALSE*, the control #1 at resource 3000 is set the *OFF* state. The aggregate function *total()* is defined in Table 5. Its only argument is a group name. In this specific case the group name should be put into the quotes ("") since it contains a whitespace character. The operators *>*, *IF..THEN..ELSE* are defined in Table 4.

20.6 Periodic rules

Periodic rules are objects, similar to event filters but intended to run certain actions periodically (instead of as a reaction to an event). Periodic rules can be used to implement environment management algorithms (e.g. cooling management) on a Guardian Management Gateway. Similar to event filters, each periodic rule consists of a predicate expression and one or more actions. Periodically (the value of the period is specified when the periodic rule is created), the periodic rule is invoked: that is, the predicate expression is evaluated and, if the result is non-zero, the corresponding actions are executed. Expressions are evaluated in units defined by the global settings.
Periodic rules share their name space with event filters (there can be no event filter and periodic rule with the same name), and action management for them uses the same commands as for event filters.
The following operations exist for periodic rules:

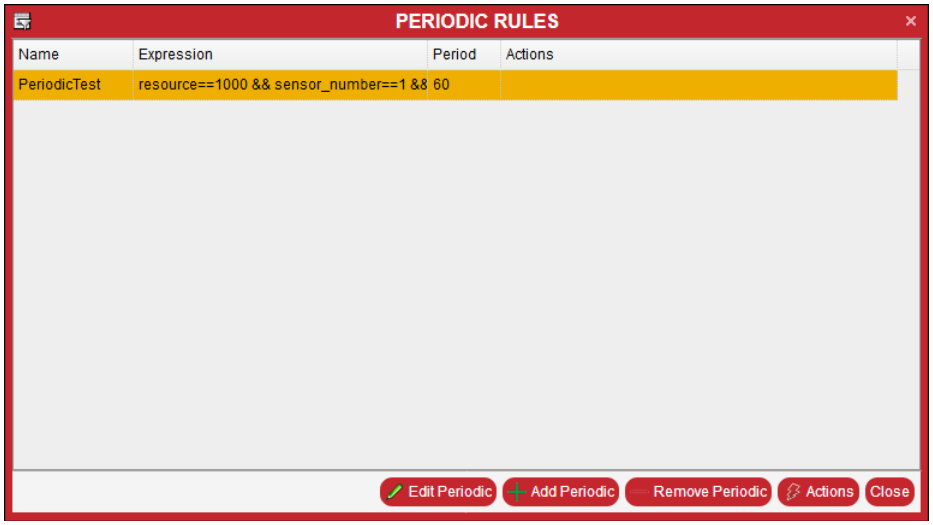
- Create a periodic rule, specifying its name, predicate expression and invocation period (in seconds)
- Delete a periodic rule by name
- Enumerate existing periodic rules
- Get predicate expression by periodic rule name
- Add an action to the periodic rule
- Enumerate actions for the given periodic rule
- Update a specific action for the given periodic rule
- Remove a specific action from the given periodic rule
- Assign a named action list to the given periodic rule
- Remove a named action list from the given periodic rule.

To manage periodic rules in CLI, use commands `periodic` and `action`.
For the command `periodic`, use its subcommands as follows:

- Use the command `periodic add` to create a new periodic rule, specify the periodic rule name, the predicate expression and the period in seconds
- Use the command `periodic delete` to delete a periodic rule by name
- Use the command `periodic list` to see the list of defined periodic rules
- Use the command `periodic show` to see information about a specific periodic rule by its name.

Use the command `action` to manage the action list for a specific periodic rule; the usage scenarios are the same as for the event filters.

This window below can be accessed by selecting the menu item “Device Settings” -> “Rules and Actions” -> “Periodic Rules”.

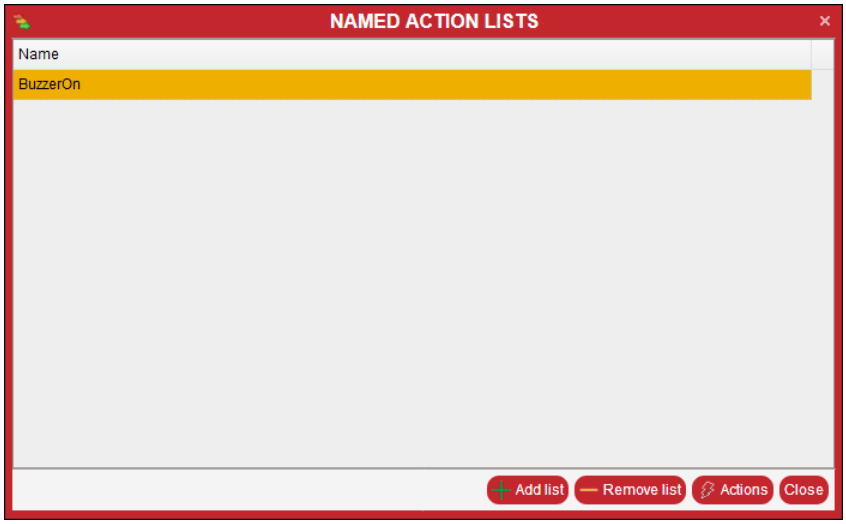


To add a new periodic rule, press the button “Add Periodic”. Visual Expression Builder features can be invoked for predicate expressions via the CTRL+Space key combination.
To edit the predicate expression and the value of the period, press the button “Edit Periodic”.
To edit the action list of the periodic rule press the button “Actions”.

Action lists for periodic rules are managed in the same way as action lists for filters (see section 20.2).

20.7 Named action lists

To facilitate event filter and periodic rule operations, a special named entity (an action lists) is created by a user, and then, actions are added to that entity. The order of adding actions to the list corresponds to the order in which actions are executed for an event filter or periodic rule operation.
Named action lists can be useful if several filters and periodic rules have the same action list. Once created, a named action list can be used multiple times. Changes made to this list are propagated to the event filters and the periodic rules to which this list is assigned.
This window below can be accessed by selecting the menu item “Device Settings” -> “Rules and Actions” -> “Action Lists”.



To create a new named list, press the “Add list” button. To delete a named list, move the cursor on the entry and press the “Remove list” button. To edit a named list, move the cursor on the entry and press the “Actions” button.

20.8 Examples of event filter and periodic rule setup

20.8.1 Example 1: Sending an e-mail for an event

This example shows how to send an e-mail message if there is a fault in the 12V voltage circuit on the MCB.
The sensor #6 “MGMT 12V Power Status” on resource 3000 “MCB” is a discrete sensor with three states: “STATE OK”, “STATE FAULT”, “STATE OFF”. In the case of a fault, the state will be “STATE FAULT”. The filter expression `resource==3000 && sensor_number==6 && sensor_state==2` is *TRUE* only for events generated by sensor #6 on resource 3000 when entering the state “STATE FAULT”. The `sensor_state` parameter is a bit mask with a single bit set. “STATE OK” corresponds to `1`, “STATE FAULT” to `2` and “STATE OFF” to `4`.

Let's create the event filter "VoltageFault".

The action list for this event filter consists of only one item:
Disposition: Run Always
Type: Send Mail
Additional fields for this action type include the recipient's e-mail address ("TO" field), the e-mail subject ("Subject" field) and the text of the message ("Message" field). The detailed human-readable information on the event will be appended to the text of the message. Let the recipient's mail address be report_floor1@myaddress.com, the e-mail subject be "Room 102: Guardian Gateway Alert", and the text of the message be "Voltage Fault".

The SMTP server should be pre-configured by selecting the menu item "Device Settings" -> "Network Services" -> "SMTP" or by the *srvconf smtp* command. The detailed information on the SMTP configuration is presented in section 14.3. In particular, the SMTP configuration defines the sender's address in the e-mail message.

The following CLI commands set up the filter:
1. Create the event filter.

```
CLI{admin}>filter add VoltageFault "resource==3000 && sensor_number==6 && sensor_state==2"
Operation completed successfully
```

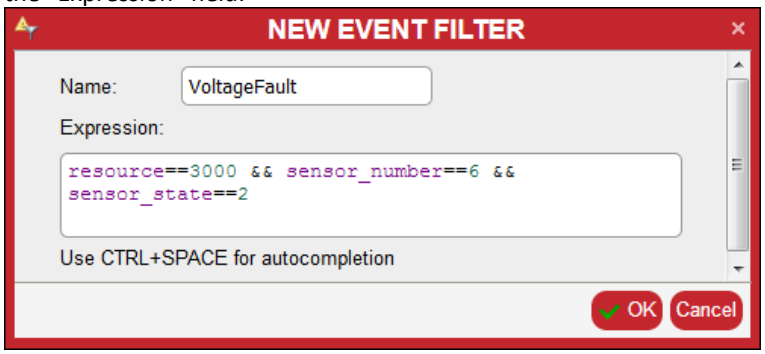
2. Add the action to the action list of the event filter.

```
CLI{admin}>action add VoltageFault always sendmail "report@myaddress.com\nRoom 102: Guardian Gateway Alert\nVoltage Fault"
Added as action 0
```

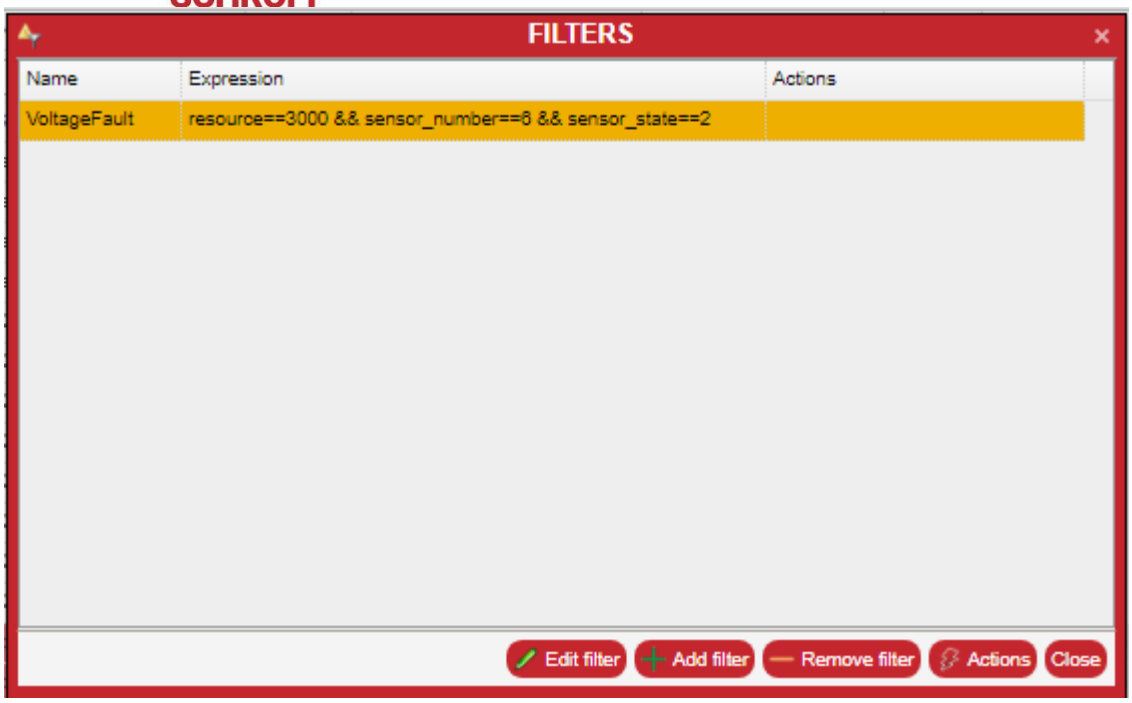
Verify the result:

```
CLI{admin}> filter show VoltageFault
Filter "VoltageFault": "resource==3000 && sensor_number==6 && sensor_state==2"
Action list:
    0: Always: Send Mail: "report@myaddress.com\nRoom 102: Guardian Gateway
Alert\nVoltage Fault"
```

Now let's set up the event filter via the Web Interface.
1. Create the event filter: Select "Device Setting" -> "Rules and Actions" -> "Event Rules" menu item. The "Filters" window is generated. Press the "Add filter" button. The "New Event Filter" window is generated. Enter the value "VoltageFault" into the "Name" field and the value "resource==3000 && sensor_number==6 && sensor_state==2" to the "Expression" field.



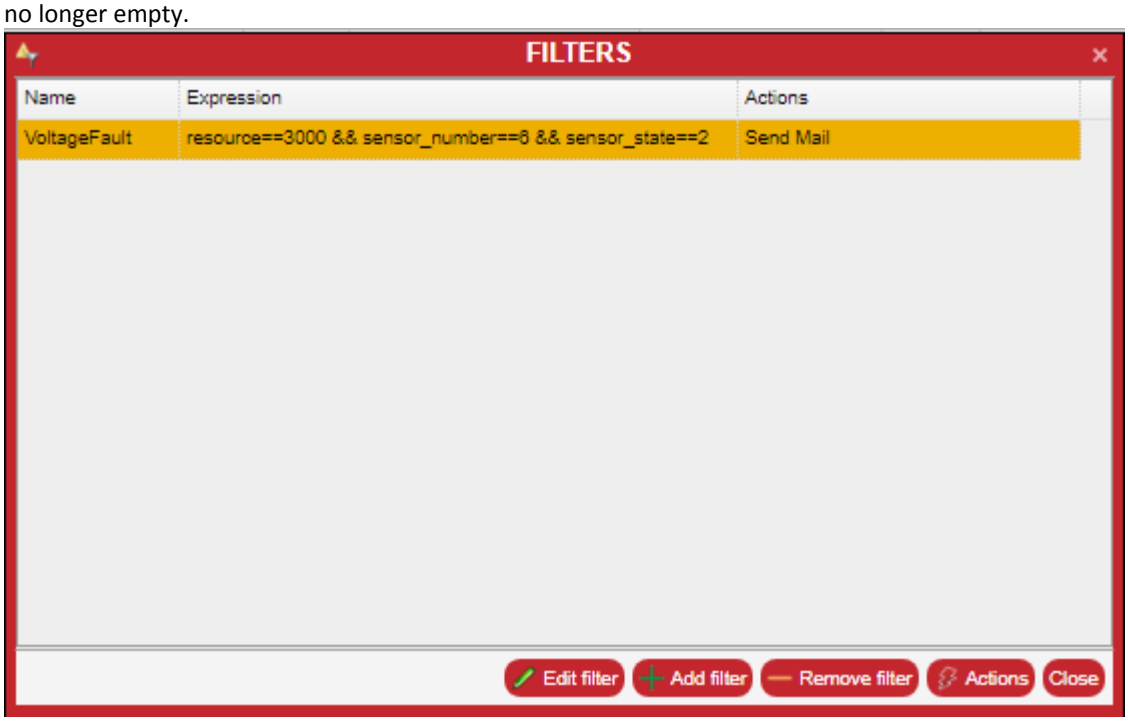
Press the "OK" button. The newly created filter is presented on the "Filters" window. The "Name" and the "Expression" cells are filled, but the "Actions" cell is empty.



2. Add the action to the action list of the event filter. Select the “VoltageFault” filter on the “Filters” window. Press the “Actions” button. The window “Actions for filter VoltageFault” is generated. Press the “Add action” button. The “New action for filter: VoltageFault” is generated. Select the “Run Always” item in the “Disposition” drop-down list. Select the “Send Mail” in the “Type” drop-down list. Enter the value “report_floor1@myaddress.com” into the “TO” field, enter value “Room 102: Guardian Gateway Alert” into the ‘Subject” field, enter the value “Voltage Fault” into the “Message” field.

Press the “OK” button. The newly created action is presented on the “Actions for filter VoltageFault” window.

Press the “Close” button on the “Actions for filter VoltageFault” window. The “Actions” cell on the “Filters” window is no longer empty.



It contains “Send Mail” text. Press the “Close” button on the “Filters” window.

20.8.2 **Example 2: Sending an SNMP trap for an event**

This example shows how to send an SNMP trap when a temperature sensor crosses a threshold. This goal can be achieved by using one filter. Let IP address `192.168.1.48` be the destination of an SNMP trap.

Let’s create the event filter “TemperatureSNMP”.

Below there are several examples of filter expressions that may be useful in the processing of temperature events.

The filter expression `resource==1000 && sensor_number==1 && assertion==1` is *TRUE* only for assertion events generated by sensor #1 on resource 1000. It ignores all the deassertion events generated by this sensor as well as all the events generated by other sensors. The sensor #1 on resource 1000 is normally a temperature sensor on a 1-Wire resource. In this filter expression the sensor type is not explicitly specified.

The filter expression `sensor_type==1 && assertion==1` is *TRUE* only for an assertion event generated by a temperature sensor. It ignores all the deassertion events generated by temperature sensors as well as all the events generated by non-temperature sensors.

The filter expression `sensor_type==1 && assertion==1 && upper_critical && resource==2000` is *TRUE* only for an assertion Upper-Critical event (crossing of the Upper Major threshold) generated by a temperature sensor on resource 2000. Resource 2000 is normally a Modbus device that may be populated with several temperature sensors.

The action list for this event filter consists of only one item:
Disposition: Run Always
Type: SNMP Trap
Destination: `192.168.1.48`

The following CLI commands set up the filter:
1. Create the event filter.

```
CLI{admin}>filter add TemperatureSNMP "sensor_type==1 && assertion==1"
Operation completed successfully
```

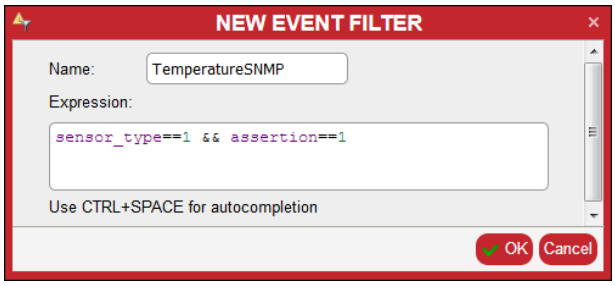
2. Add the action to the action list of the event filter.

```
CLI{admin}>action add TemperatureSNMP always snmptrap 192.168.1.48
Added as action 0
```

Verify the result:

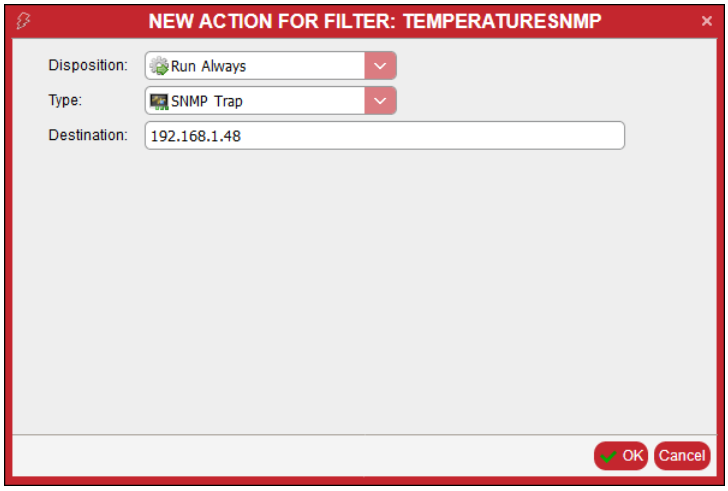
```
CLI{admin}> filter show TemperatureSNMP
Filter "TemperatureSNMP": "sensor_type==1 && assertion==1"
Action list:
    0: Always: SNMP Trap: "192.168.1.48"
```

Now let's set up the event filter via the Web Interface.
1. Create the event filter: Select "Device Setting" -> "Rules and Actions" -> "Event Rules" menu item. The "Filters" window is generated. Press the "Add filter" button. The "New Event Filter" window is generated. Enter the value "TemperatureSNMP" into the "Name" field and the value "sensor_type==1 && assertion==1" to the "Expression" field.

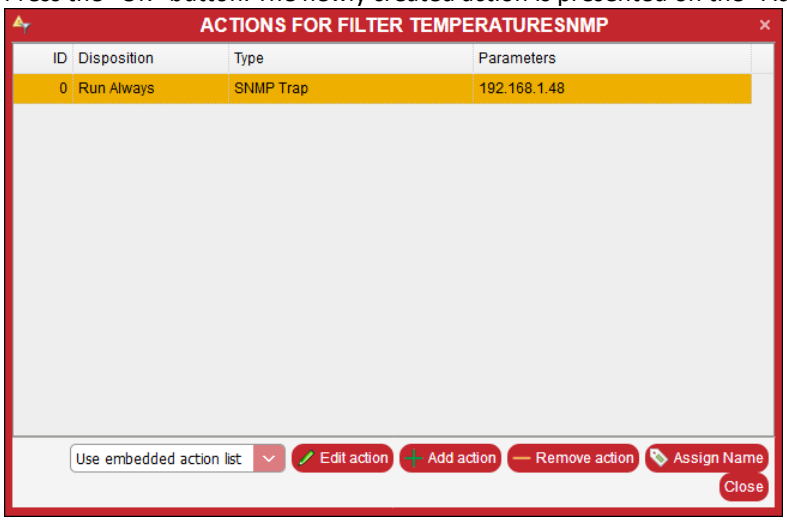


Press the "OK" button. The newly created filter is presented on the "Filters" window. The "Name" and the "Expression" cells are filled, but the "Actions" cell is empty.

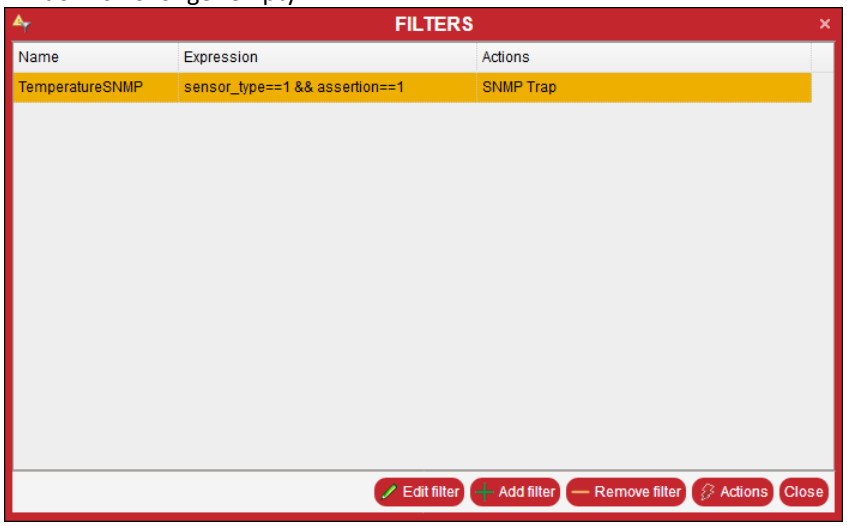
2. Add the action to the action list of the event filter: Select the “TemperatureSNMP” filter on the “Filters” window. Press the “Actions” button. The window “Actions for filter TemperatureSNMP” is generated. Press the “Add action” button. The “New action for filter: TemperatureSNMP” is generated. Select the “Run Always” item in the “Disposition” drop-down list. Select the “SNMP Trap” in the “Type” drop-down list. Enter the value “192.168.1.48” into the “Destination” field.



Press the “OK” button. The newly created action is presented on the “Actions for filter TemperatureSNMP” window.



Press the “Close” button on the “Actions for filter TemperatureSNMP” window. The “Actions” cell on the “Filters” window is no longer empty.



It contains “SNMP Trap” text. Press the “Close” button on the “Filters” window.

20.8.3 Example 3: Using periodic rules to track presence of alarms in the system

This example shows how to turn on the buzzer (Control #1 on Resource 3000, Output: Audible) when a CRITICAL Alarm is generated and to turn off the buzzer when all the Critical Alarms are either acknowledged or deleted. This goal can be achieved by using one filter and one periodic rule.

Let's create the event filter "TurnOnBuzzer" with the filter expression
`alarm_exists(severity == 0 && acknowledged == 0)`

This expression evaluates to *TRUE* when there is an unacknowledged Critical alarm.
Then set up the action list for the event filter. It contains only one item:
Disposition: Run Always
Type: Evaluate Expression
Expression: `CONTROL(3000,1)=1`

Let's create the periodic rule "TurnOffBuzzer" with the expression
`CONTROL(3000,1)==1 && !alarm_exists(severity == 0 && acknowledged == 0)`
Let set the period to 60 seconds. Every 60 seconds the expression is evaluated. It is *TRUE* if the buzzer is turned on and there is no unacknowledged Critical alarm.
Then set up the action list for the periodic rule. It contains only one item:
Disposition: Run Always
Type: Evaluate Expression
Expression: `CONTROL(3000,1)=0`

The buzzer will be turned on immediately after a Critical alarm is generated. The buzzer will be turned off within 1 minute after all the Critical alarms are either acknowledged or gone.

The following CLI commands set up the filter and the periodic rule:
1. Create the event filter.

```
CLI{admin}>filter add TurnOnBuzzer "alarm_exists(severity == 0 && acknowledged == 0)"
Operation completed successfully
```

2. Add the action to the action list of the event filter.

```
CLI{admin}>action add TurnOnBuzzer always expression "CONTROL(3000,1)=1"
Added as action 0
```

3. Create the periodic rule.

```
CLI{admin}>periodic add TurnOffBuzzer "CONTROL(3000,1)==1 && !alarm_exists(severity == 0 && acknowledged == 0)" 60
Operation completed successfully
```

4. Add the action to the action list of the periodic rule.

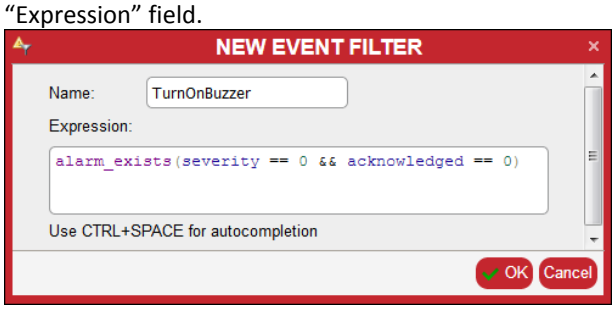
```
CLI{admin}>action add TurnOffBuzzer always expression "CONTROL(3000,1)=0"
Added as action 0
```

Verify the result:

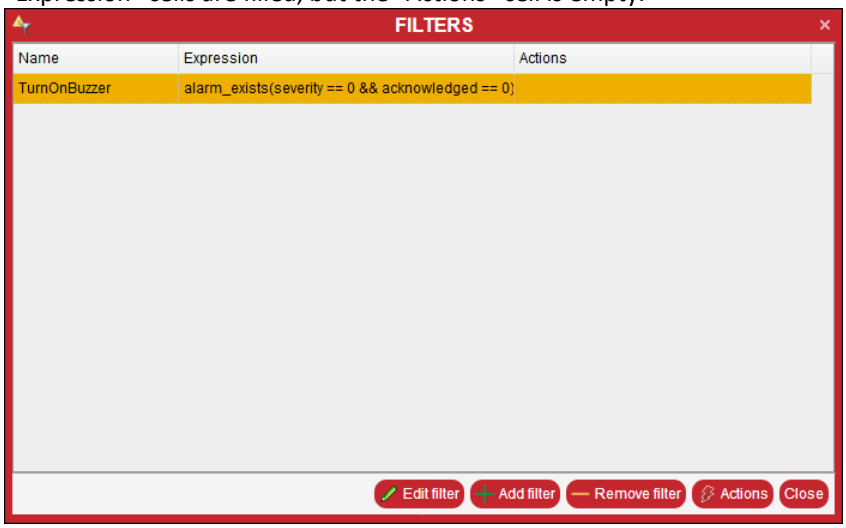
```
CLI{admin}> filter show TurnOnBuzzer
Filter "TurnOnBuzzer": "alarm_exists(severity == 0 && acknowledged == 0)"
Action list:
    0: Always: Expression: "CONTROL(3000,1)=1"

CLI{admin}> periodic show TurnOffBuzzer
Periodic expression "TurnOffBuzzer": "CONTROL(3000,1)==1 && !alarm_exists(severity == 0 && acknowledged == 0)"; Period: 60 sec
Action list:
    0: Always: Expression: "CONTROL(3000,1)=0"
```

Now let's set up the event filter and the periodic rule via the Web Interface.

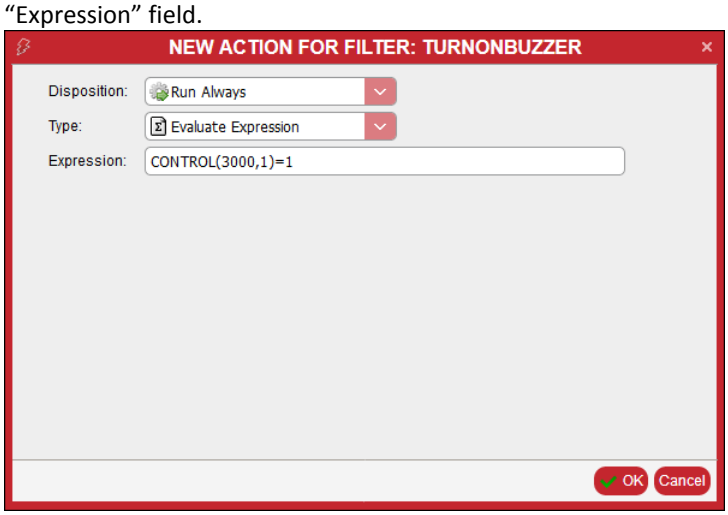
1. Create the event filter: Select "Device Setting" -> "Rules and Actions" -> "Event Rules" menu item. The "Filters" window is generated. Press the "Add filter" button. The "New Event Filter" window is generated. Enter the value "TurnOnBuzzer" into the "Name" field and the value "alarm_exists(severity == 0 && acknowledged == 0)" to the "Expression" field.



Press the "OK" button. The newly created filter is presented on the "Filters" window. The "Name" and the "Expression" cells are filled, but the "Actions" cell is empty.

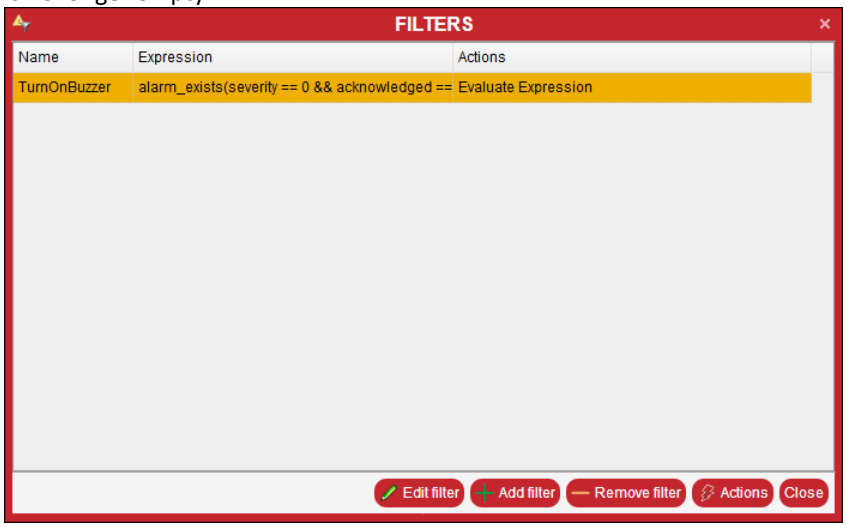


2. Add the action to the action list of the event filter: Select the "TurnOnBuzzer" filter on the "Filters" window. Press the "Actions" button. The window "Actions" for filter TurnOnBuzzer" is generated. Press the "Add action" button. The "New action for filter: TurnOnBuzzer" is generated. Select the "Run Always" item in the "Disposition" drop-down list. Select the "Evaluate Expression" in the "Type" drop-down list. Enter the value "CONTROL(3000,1)=1" into the "Expression" field.



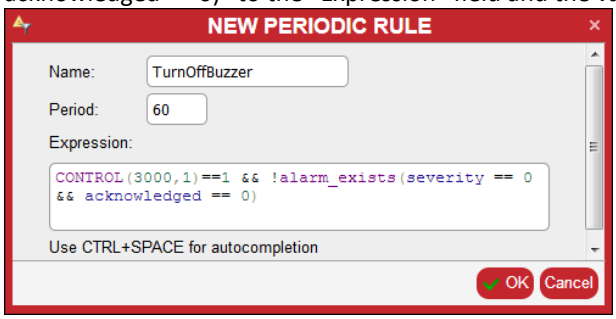
Press the "OK" button. The newly created action is presented on the "Actions for filter TurnOnBuzzer" window.

Press the “Close” button on the “Actions for filter TurnOnBuzzer” window. The “Actions” cell on the “Filters” window is no longer empty.



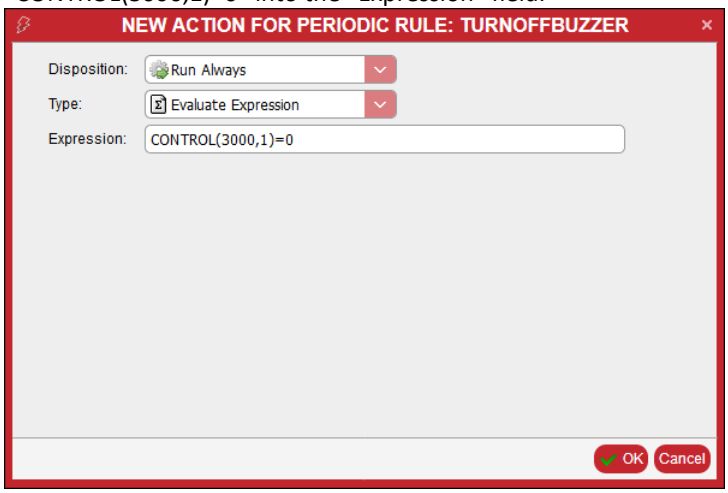
It contains “Evaluate Expression” text. Press the “Close” button on the “Filters” window.

3. Create the periodic rule: Select “Device Setting” -> “Rules and Actions” -> “Periodic Rules” menu item. The “Periodic Rules” window is generated. Press the “Add Periodic” button. The “New periodic rule” window is generated. Enter the value “TurnOffBuzzer” into the “Name” field and the value “CONTROL(3000,1)==1 && !alarm_exists(severity == 0 && acknowledged == 0)” to the “Expression” field and the value “60” to the “Period” field.

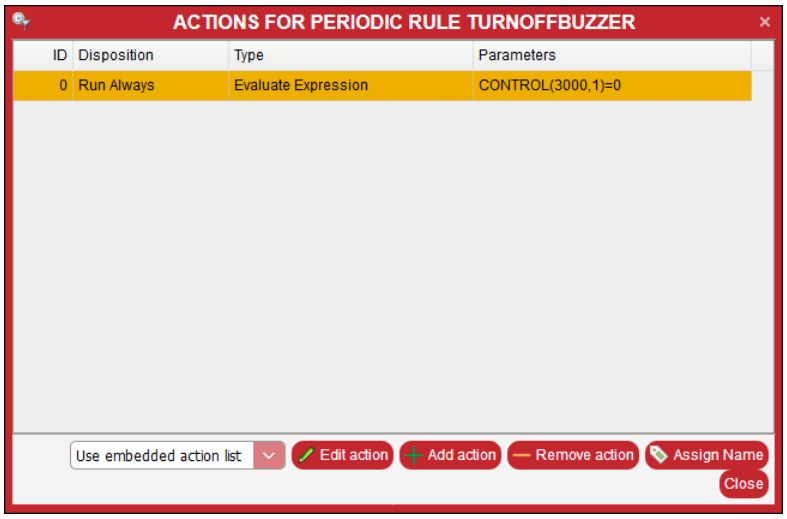


Press the “OK” button. The newly created periodic rule is presented on the “Periodic Rules” window. The “Name”, the “Expression” and the “Period” cells are filled, but the “Actions” cell is empty.

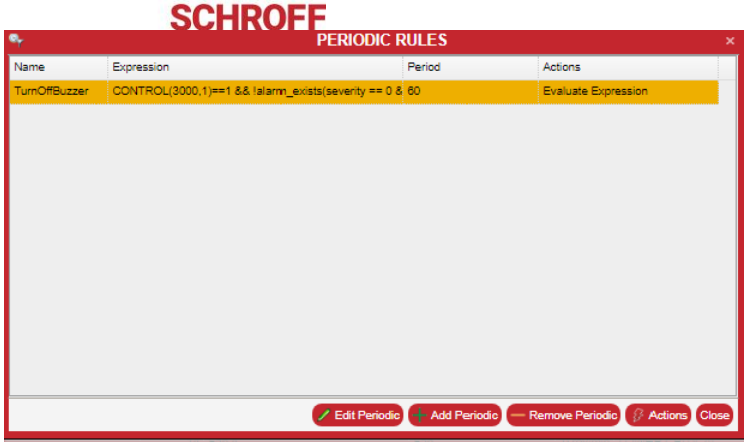
4. Add the action to the action list of the periodic rule: Select the “TurnOffBuzzer” periodic rule on the “Periodic Rules” window. Press the “Actions” button. The window “Actions for periodic rule TurnOffBuzzer” is generated. Press the “Add action” button. The “New action for periodic rule: TurnOffBuzzer” is generated. Select the “Run Always” item in the “Disposition” drop-down list. Select the “Evaluate Expression” in the “Type” drop-down list. Enter the value “CONTROL(3000,1)=0” into the “Expression” field.



Press the “OK” button. The newly created action is presented on the “Actions for periodic rule TurnOffBuzzer” window.



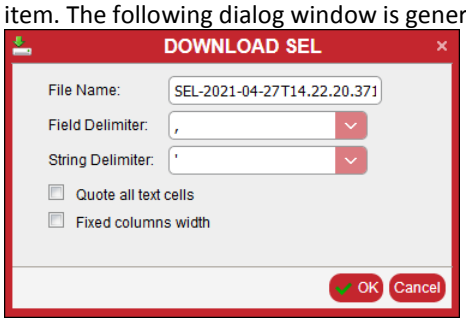
Press the “Close” button on the “Actions for periodic rule TurnOffBuzzer” window. The “Actions” cell on the “Perodic Rules” window is no longer empty.



It contains “Evaluate Expression” text. Press the “Close" button on the “Periodic Rules” window.

21 System Log

To download the system log of the device in the CSV format select the “Maintenance”->”Download System Log” menu item. The following dialog window is generated (see section 22).



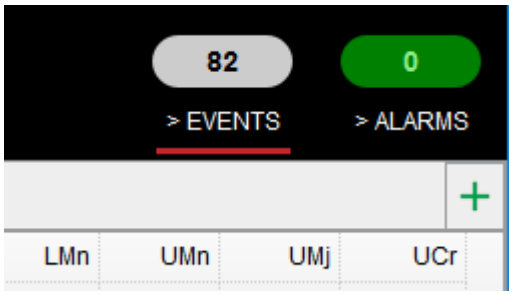
Set values in the fields and press the “OK” button. The system log file will be downloaded to the local machine.

22 Event log (SEL)

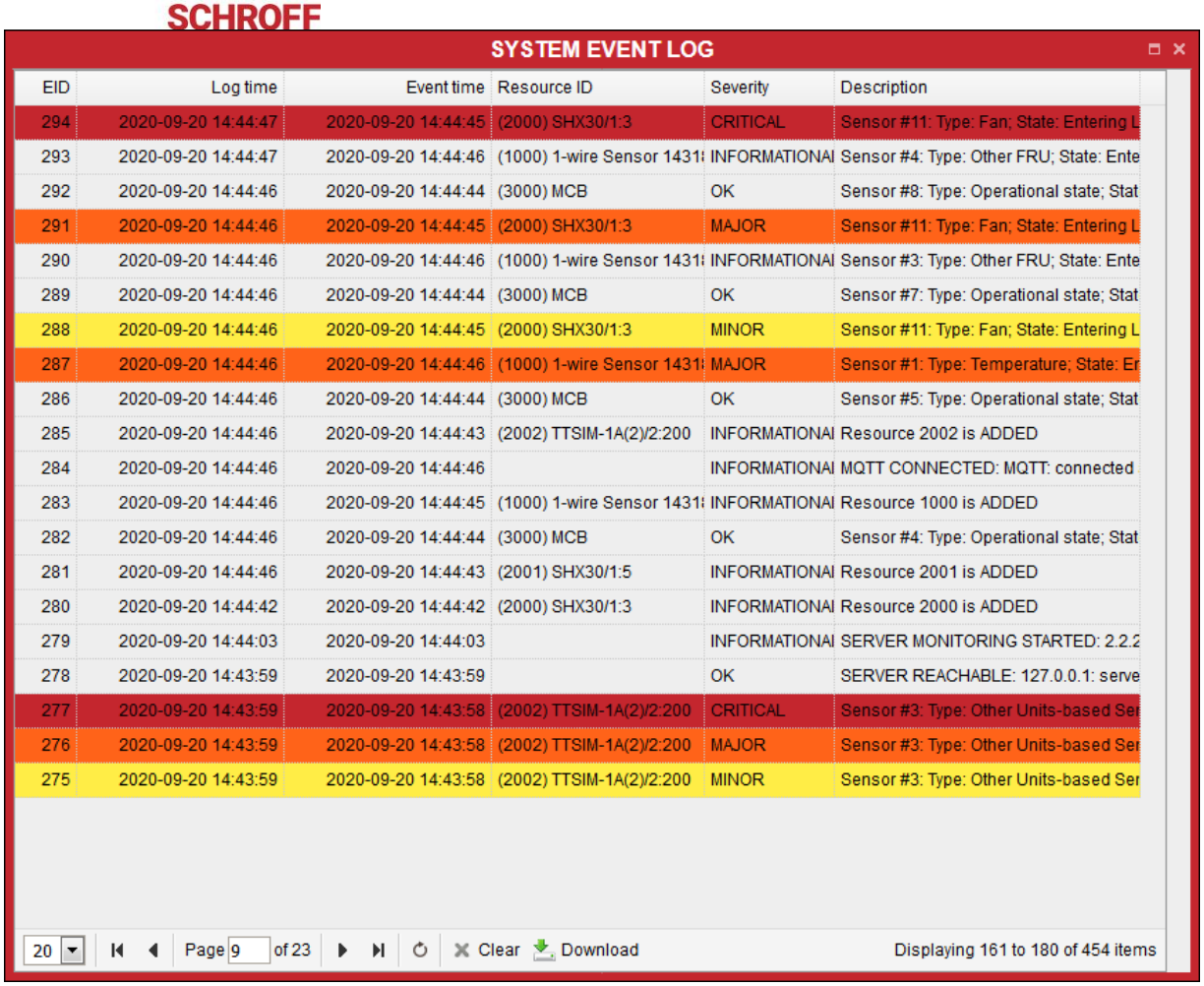
The event log is maintained on the Guardian Management Gateway in the format of HPI System Event Log (SEL). All generated events are stored in the event log. Event log storage capacity is 10000 events by default. When the event log reaches its capacity, the oldest events become deleted. Also, a user can clear the event log at any moment. Other than that, the event log is read-only for the user.
The following operations are defined for the system event log:

- Get information about the event log as a whole
- Enumerate entries in the event log
- Clear the event log
- Download the event log.

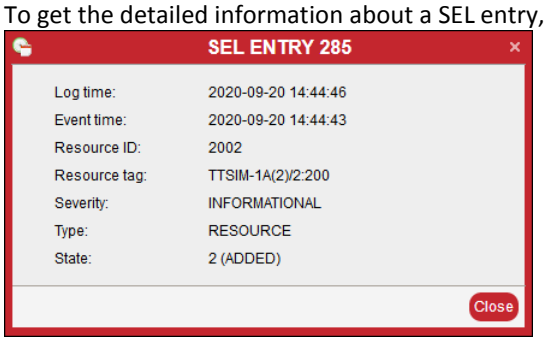
In the Web interface, the event log is represented by the “System Event Log” window. This window can be accessed by selecting the menu item “Maintenance” -> “System Event Log” or by pressing the SEL indicator on the status bar.



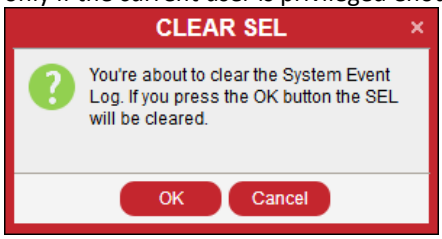
The “System Event Log” window shows event log entries page by page. Log entries are colored according to its severity: yellow for Minor, orange for Major, red for Critical. Log entries with OK and Informational severity levels are not colored. Control items at the bottom of the window allow the user to navigate to the previous page and to the next page, navigate to the beginning or to the end of the event log, change the number of items per page, refresh the view, clear the event log or download the event log in the CSV format.



To get the detailed information about a SEL entry, double-click on this entry. The “SEL Entry <id>” is generated.

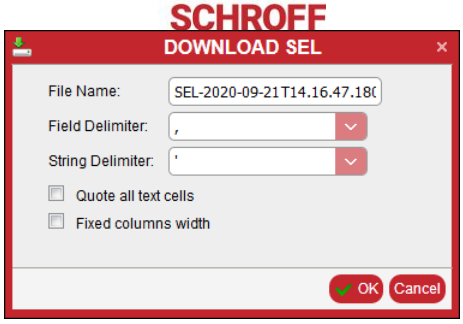


To clear the event log (erase all entries), press the “Clear” button at the bottom of the window (this button is visible only if the current user is privileged enough to clear the event log). The confirmation dialog will appear:



Press the OK button to confirm the intention to clear the event log.

To download the SEL in the CSV format press the “Download” button at the bottom of the window. The “Download SEL” window is generated. It contains the “Field Delimeter” and “String Delimiter” dropdown lists, the “Quote all text cells” and “Fixed columns width” checkboxes”. The fields define the CSV format.



Press the “OK” button and the SEL will be saved on the local machine. The file name can be set in the “File Name” field.

23 Alarm Table

Guardian Management Gateway maintains the HPI Alarm table, which is the table of active alarms and represents an aggregated view on any anomalies in the current state of the Guardian Management Gateway. There is only one alarm table on the Guardian Management Gateway. Each alarm is caused by the corresponding alarm condition; alarm conditions are typically associated with sensors. Guardian Management Gateway currently supports only sensor-based alarm conditions.

For a threshold-based sensor, an alarm condition occurs when the sensor reading goes beyond a threshold, and ceases to exist when the sensor reading goes back. If multiple thresholds are crossed at once, multiple alarm conditions are generated.

For a discrete sensor, an alarm condition occurs when the sensor goes into a state with the severity Minor, Major or Critical, and disappears when the sensor leaves this state.

Alarms are associated with events; an alarm is added to the alarm table in response to the event that indicates that the corresponding alarm condition has appeared.

An alarm can be automatically removed from the alarm table is response to the event that indicates that the corresponding alarm condition has disappeared. Or, an alarm can be “sticky” and stay in the alarm table until it is deleted manually by a user. This behavior depends on the global parameter “maximum transient alarm severity”. Alarms with the severity greater than the value of this parameter stay permanently in the alarm table; alarms with the severity less or equal than value of this parameter are automatically removed when the corresponding alarm condition goes away. By default, the value of this parameter is set to “Critical”, which means that all alarms are transient, but it can be changed by a user. For example, if this parameter is set to “Minor”, then alarms with severity “Critical” and “Major” will stay in the alarm table permanently, while alarms with the severity “Minor” and below will be transient.

An alarm can be acknowledged by a user, meaning that the user has recognized the presence of this alarm. Initially an alarm is unacknowledged. Acknowledged and unacknowledged alarms are shown differently in CLI and Web interfaces.

A user can manually delete an alarm from the alarm table. Transient alarms can be deleted by the user even while the alarm condition is active.

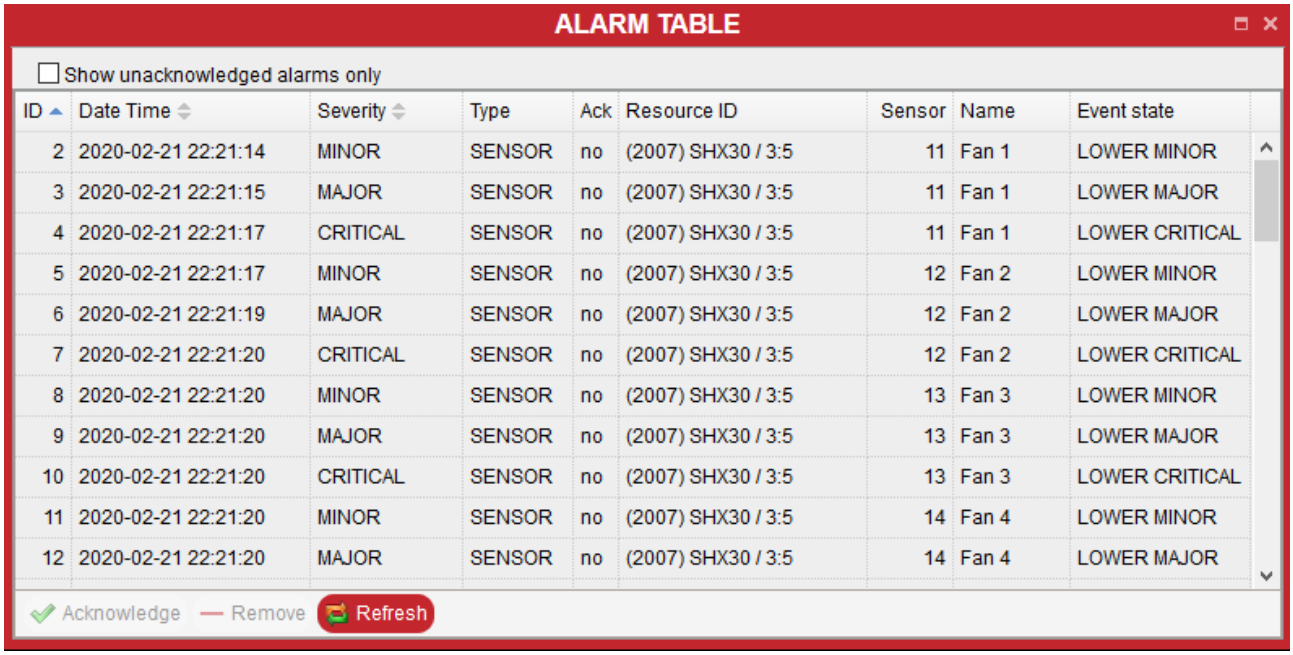
For each alarm table entry (active alarm), the following information is available:

- Alarm ID – the index of the alarm in the table
- Timestamp - when the alarm was created
- Alarm severity – can be Minor, Major or Critical; corresponds to the severity of the event that caused the alarm
- Acknowledged state (`yes` or `no`)
- Alarm condition; for sensor-based alarms, the alarm condition contains the following fields:
    - Entity path of the entity related to the alarm condition
    - Resource number
    - Sensor number
    - Event state (sensor state) that caused the alarm condition

A user can perform the following operations with the alarm table and specific alarms in it:

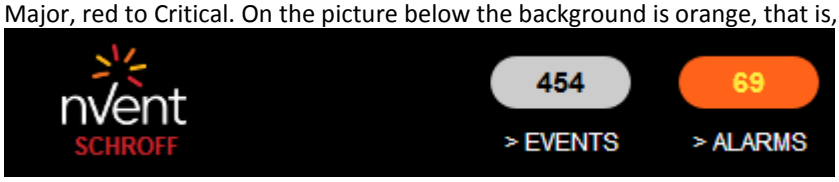
- View the alarm table as a whole
- View information about a specific alarm
- Acknowledge a specific alarm
- Delete a specific alarm from the alarm table

In the Web interface, the alarm table is represented by a separate table-like window which is invoked by the menu command “Maintenance” -> “Alarm Table”. Each line in the table represents one alarm.



ID	Date Time	Severity	Type	Ack	Resource ID	Sensor	Name	Event state
2	2020-02-21 22:21:14	MINOR	SENSOR	no	(2007) SHX30 / 3:5	11	Fan 1	LOWER MINOR
3	2020-02-21 22:21:15	MAJOR	SENSOR	no	(2007) SHX30 / 3:5	11	Fan 1	LOWER MAJOR
4	2020-02-21 22:21:17	CRITICAL	SENSOR	no	(2007) SHX30 / 3:5	11	Fan 1	LOWER CRITICAL
5	2020-02-21 22:21:17	MINOR	SENSOR	no	(2007) SHX30 / 3:5	12	Fan 2	LOWER MINOR
6	2020-02-21 22:21:19	MAJOR	SENSOR	no	(2007) SHX30 / 3:5	12	Fan 2	LOWER MAJOR
7	2020-02-21 22:21:20	CRITICAL	SENSOR	no	(2007) SHX30 / 3:5	12	Fan 2	LOWER CRITICAL
8	2020-02-21 22:21:20	MINOR	SENSOR	no	(2007) SHX30 / 3:5	13	Fan 3	LOWER MINOR
9	2020-02-21 22:21:20	MAJOR	SENSOR	no	(2007) SHX30 / 3:5	13	Fan 3	LOWER MAJOR
10	2020-02-21 22:21:20	CRITICAL	SENSOR	no	(2007) SHX30 / 3:5	13	Fan 3	LOWER CRITICAL
11	2020-02-21 22:21:20	MINOR	SENSOR	no	(2007) SHX30 / 3:5	14	Fan 4	LOWER MINOR
12	2020-02-21 22:21:20	MAJOR	SENSOR	no	(2007) SHX30 / 3:5	14	Fan 4	LOWER MAJOR

To acknowledge an alarm, select it in the table window and press the “Acknowledge” button at the bottom of the window.
To delete an alarm, select it in the table window and press the “Remove” button at the bottom of the window.
If the “Show unacknowledged alarms only” checkbox is checked, acknowledged alarms are not shown in the “Alarm Table” window.
To refresh current view, press the “Refresh” button at the bottom of the window.
On the top black bar a total count of the entries of the Alarm Table is shown. The color of the background corresponds to the maximal severity level of unacknowledged entries of the Alarm Table: yellow corresponds to Minor, orange to Major, red to Critical. On the picture below the background is orange, that is,



there is at least one unacknowledged alarm with Major severity, and alarms with Critical severity are either not present or acknowledged.

24 MCB Instruments

The Master Control Board (MCB) of the Guardian Management Gateway hosts the single-board computer that runs Guardian Management Gateway firmware. Also it hosts several hardware entities that are exposed to the user as sensor and controls. The MCB itself is exposed as the resource 3000.

There are following sensors and controls on the MCB resource:

- Sensor “MCB Temperature” (#1): reports the temperature measured on the MCB, in degrees C
- Sensor “MCB 12V” (#2): reports the 12V voltage on the MCB, in volts
- Sensor “Reboot Reason” (#3): the discrete sensor reports the reason of the last reboot of the Guardian Management Gateway (see details below)
- Sensor “USB1 Power Status” (#4): the discrete sensor that reports the power fault state of USB 1 interface (see below for the state meaning for this and subsequent sensors)
- Sensor “USB2 Power Status” (#5): the discrete sensor that reports the power fault state of USB 2 interface
- Sensor “MGMT 12V Power Status” (#6): the discrete sensor that reports the fault state of the external +12V power line
- Sensor “I2C_1 Bus Status” (#7): the discrete sensor that reports the fault state of the I^2C bus #1
- Sensor “I2C_2 Bus Status” (#8): the discrete sensor that reports the fault state of the I^2C bus #2
- Sensor “LAN Physical Link” (#9): the discrete sensor that reports the state of the LAN physical link.
- Control “Buzzer” (#1): a digital control that controls a buzzer located on the MCB, set to *ON* to turn the buzzer on, set to *OFF* to turn it off
- Control “USB1 Power Fault Reset” (#2), a digital control, set to *Pulse ON* to reset the power fault state of the USB 1 interface
- Control “USB2 Power Fault Reset” (#3), a digital control, set to *Pulse ON* to reset the power fault state of the USB 2 interface
- Control “MGMT 12V Power Fault Reset” (#4), a digital control, set to *Pulse ON* to reset the power fault state of the external +12V interface

For the “Reboot Reason” sensor, the states have the following meaning:

- State 0 (State Mask 1): the last boot was a power-on
- State 1 (State Mask 2): the last reboot was caused by a watchdog timer
- State 2 (State Mask 4): the last reboot was caused by software (e.g. a CLI command or Web interface action)
- State 3 (State Mask 8): the last reboot was caused by hardware reset
- State 4 (State Mask 0x10): the last reboot was caused by a firmware upgrade.
- State 5 (State Mask 0x20): the last reboot was caused by a crash.

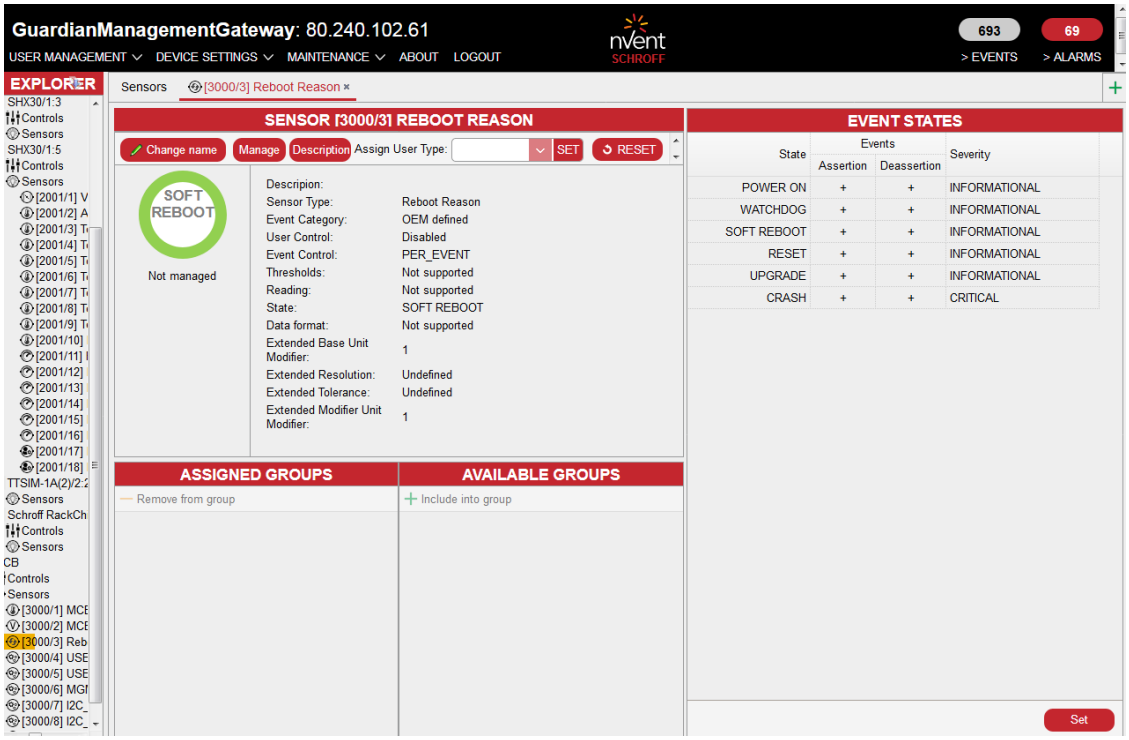
For the status sensors, the states have the following meaning:

- State 0 (State Mask 1): no fault
- State 1 (State Mask 2): a fault is present
- State 2 (State Mask 4): the corresponding subsystem is turned off

For the LAN state sensors, the states have the following meaning:

- State 0 (State Mask 1): no LAN physical link
- State 1 (State Mask 2): the LAN physical link is present

The inventory #0 is present on the MCB resource. This inventory contains a standard FRU information about the MCB (manufacturer, product name, serial number, etc.) and the LCD Calibration Parameters record in the nVent OEM format. It is read-only and stored in the EEPROM physically located on the MCB.



25 Restart, Reboot and Factory Reset

There are three types of restart applicable to a Schroff Guardian Management Gateway:

1. Restart is a termination and relaunch of the application (`smrc`) that runs on the MCB CPU and manages the Guardian Management Gateway functionality. The operating system (Linux) running on that CPU is not affected. This is the fastest type of restart
2. Reboot means reboot of the operating system running on the MCB CPU. After the restart of the operating system, the managing application is started automatically. A reboot takes longer than a restart, because the operating system gets involved. A reboot can be caused by a hardware reset or by a software command
3. Factory reset involves clearing of all configuration data on the Guardian Management Gateway and return to factory default settings. A factory reset involves a reboot.

Reboot (hardware reset) and factory reset can be initiated from the front panel, by pressing hardware buttons; there are two buttons, “Reset” and “Recovery” which are recessed to prevent them from being accidentally pressed. A sharp object like a tip of a pen is needed to press them.
To initiate a hardware reset from the front panel, press the “Reset” button.
To initiate a factory reset from the front panel, press and hold the “Recovery” button, then press the “Reset” button and then release the “Recovery” button.
To initiate a restart, reboot or factory reset with the Web interface, use the menu commands “Maintenance” -> “Restart”, “Maintenance” -> “Reboot” and “Maintenance” -> “Factory Reset”, respectively. In all three cases, a confirmation dialog is shown to prevent accidental invocation of the command.

For example, in the case of reboot, the following dialog will be shown:

26 Firmware Upgrade

Updated firmware images are periodically released by nVent and made accessible to customers.

Upgrading Guardian Management Gateway firmware is done in the following way:

1. A new firmware image is downloaded on the Guardian Management Gateway.
2. The firmware is installed to the flash partition.
3. The Guardian Management Gateway is rebooted to activate the new firmware.

Each firmware image contains everything needed for Guardian Management Gateway operation: the U-Boot, the Linux operating system kernel and the root file system that hosts system utilities and applications.

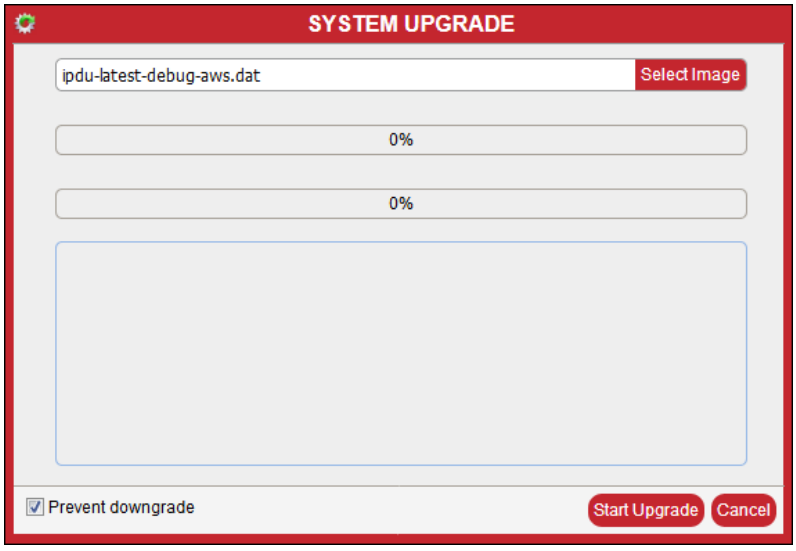
The image is protected by a digital signature (SHA256 digest) to ensure its integrity and authenticity. The signature is added to the image when it is created. When the installation of a firmware upgrade image is requested, the image signature is verified against a public key stored on the Guardian Management Gateway file system. If the signature is absent or is not valid, the image is rejected.

If the signature is valid, the following conditions are guaranteed to be met:

4. The image is not corrupted (since otherwise the signature would no longer match the calculated digest).
5. The image comes from nVent (since no one else has our private key, which is necessary to generate a valid signature).

Firmware upgrade does not affect the Guardian Management Gateway configuration (i.e. does not reset settings previously made).

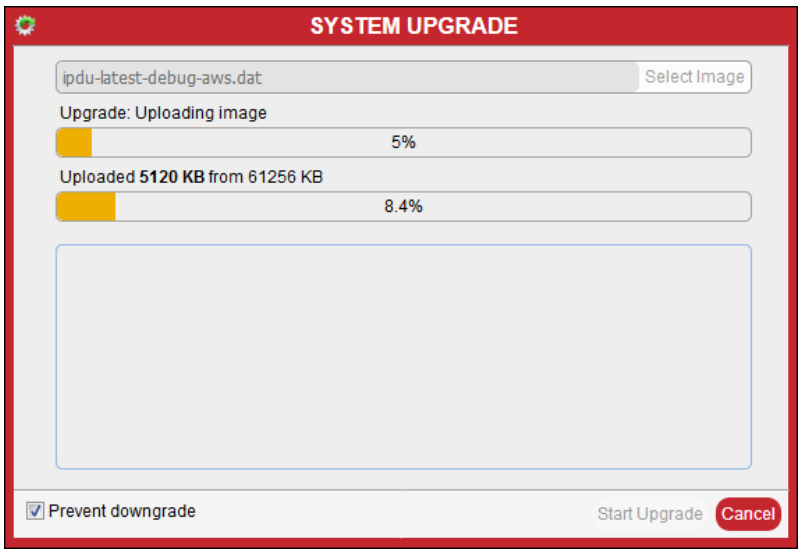
To perform firmware upgrade via the Web interface, use the “System Upgrade” dialog invoked with the menu command “Maintenance” -> “Upgrade”.



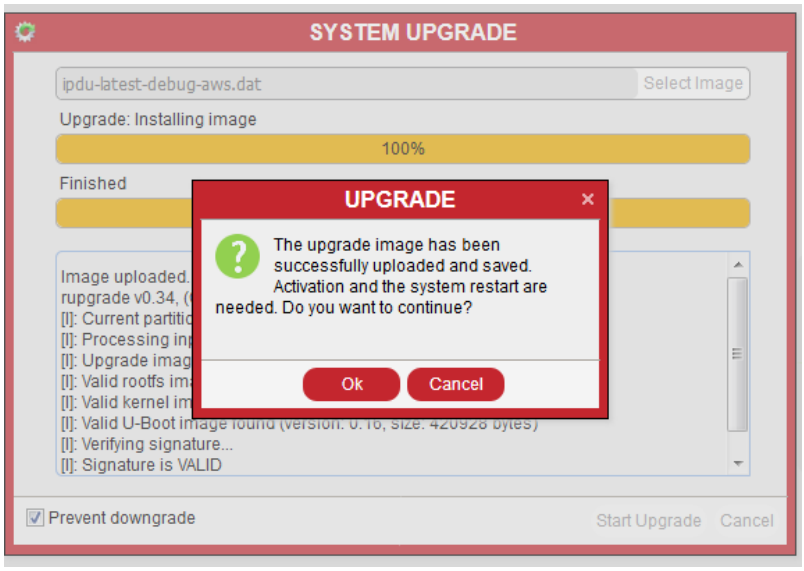
In this dialog, the user chooses the upgrade image located on the local (client) file system. This file is downloaded to the Guardian Management Gateway (into the temporary directory), then installed in the flash partition and then the Guardian Management Gateway is rebooted to activate the new firmware.

Check the check box “Prevent downgrade” (it is checked by default) to disallow downgrade (installation of images with the version less or equal to the current version); the meaning of this check box is opposite to the meaning of the option $-f$ for the CLI.

After the image file is chosen, press the button “Start Upgrade” to start image download and installation. The installation progress is reflected in the progress bars (the first progress bar corresponds to the whole firmware upgrade procedure, the second progress bar reflects the progress of a specific firmware upgrade stage). Press “Cancel” during this phase to cancel the upgrade.

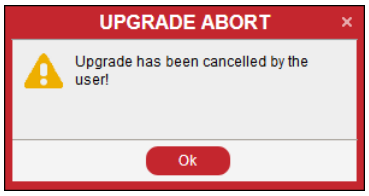


After the image is downloaded and installed in the flash partition, the user is presented with the dialog, asking to confirm activation of the new image:



If the user agrees to the request, then the Guardian Management Gateway is rebooted and the new image is activated.

If the user declines the activation request, the upgrade is cancelled and the image installation is rolled back. The window below informs the user:



The currently running firmware will continue to run, even after future reboots.

27 Saving and Loading Configuration

Guardian Management Gateway configuration includes data items and values that are persistent across system reboots and the `smrc` application restarts. It is stored in the flash file system as a collection of JSON files, each file storing data for a specific component of the configuration.

In addition, configuration is periodically archived and stored in the Guardian Management Gateway EEPROM. This is to facilitate hot swapping of management controller boards (MCBs) between Guardian Management Gateways. In the case of a hot swap, the configuration stays with the Guardian Management Gateway and can be obtained and applied on the newly inserted MCB.

Configuration consists of following components:

- Global settings
- Network configuration settings
- Host name
- Network service configuration settings
- List of users and their properties
- List of roles and their properties
- SNMPv3 user settings
- Security settings (firewalls and login restrictions)
- SSL certificate for the HTTP server
- LDAP settings
- Rules for events handling with corresponding actions
- User-defined resource names
- Configuration of physical sensors (with user-defined sensor names)
- Configuration of controls (user names assigned to controls)
- List of sensor/control groups, their contents and properties
- List of managed sensors and their properties
- Server reachability settings
- Resource map for 1-wire devices
- Resource map for Modbus devices

For a user, the following actions are available:

- Save configuration in an archive and download it to an external server or copy to the USB stick inserted into the USB 2 port
- Load and apply configuration from an archive from an external server or from the USB stick inserted into the USB 2 port
- View the list of available configuration archives on the USB stick inserted into the USB 2 port.

A configuration archive saved on one Guardian Management Gateway can then be loaded on another Guardian Management Gateway in order to duplicate configuration from one Guardian Management Gateway to another. It is also possible to apply this operation to multiple Guardian Management Gateways in turn if their configuration should be similar. The components that are different between these Guardian Management Gateways should not be included into the configuration archive during the save operation. This configuration transfer could be done with the use of the USB stick, or of a remote location via the Web interface.

In addition, the scenario of MCB hot replacement should be considered. In this case, configuration is loaded from the EEPROM belonging to the Guardian Management Gateway and is applied on the newly inserted MCB.

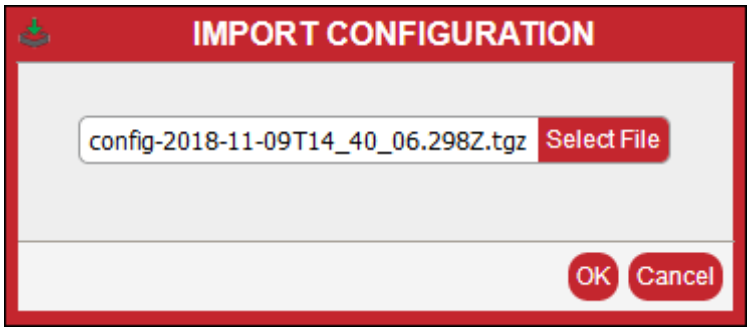
27.1 Saving current configuration

With the Web interface, the configuration archive can be downloaded from the Guardian Management Gateway to the client system. To do that, invoke the menu command “Maintenance” -> “Export configuration”. The dialog appears, in which the user can choose the components to be saved:

After the desired set of components is chosen, the configuration archive is created on the Guardian Management Gateway and then downloaded to the client system (normally to the system “Downloads” directory).

27.2 Loading configuration

In the Web interface, a configuration archive can be uploaded from the client system to the Guardian Management Gateway and the configuration will be applied. To do that, invoke the menu command “Maintenance” -> “Import configuration”. The dialog “Import Configuration” appears, in which the user can choose the configuration archive to upload:



After the user chooses the target file and presses the OK button, the configuration archive is transferred to the Guardian Management Gateway, the *smrc* application is restarted and the new configuration is applied.

28 Using SNMP

The Guardian Management Gateway supports a Simple Network Management Protocol (SNMP) interface to that allows accessing configuration, control variables and sensor readings. The following groups of variables are supported by this interface:

- System Configuration
- Physical Sensors
- Managed Sensors, including sensor log
- Controls
- Schroff SHX Devices
- Server reachability table
- System Event Log

According to SNMP rules, the variables from these groups are represented via a hierarchical data model, each variable identified via an object identifier (OID). These object identifiers have a common root OID:
`iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).nvent(16394).products(2).smartGatewayPlatform(3).sgp(1)`
16394 is a unique private enterprise number for nVent, Schroff GmbH (formerly Pentair Technical Products, referenced as nVent in this document), obtained from IANA. In the remainder of this section, the root OID is denoted as *<ROOTOID>*.
The structure of the branches of the SNMP variables tree is described in the following subsections.
The definition of SNMP variables provided by the Guardian Management Gateway is contained in a Management Information Base (MIB) file *SGP-MIB.txt*. This file should be installed on the management system (the client system, that interacts with the Guardian Management Gateway over the network). It depends on the SNMP client software how the MIB file should be installed on the management system; usually this file must be placed in a special location on the management system or compiled with a MIB compiler. If the MIB is not installed on the client system, SNMP communication with the Guardian Management Gateway is still possible; however symbolic names for the OIDs are not available and OIDs should be used in numeric form.
It should be mentioned that access to some SNMP variables may require communication with physical devices or EEPROM data read operations to be invoked. In some cases such operations (e.g. accessing Controls) may take a rather long time. It is recommended to set the SNMP client timeout to 15 seconds. For example, to retrieve the entire Guardian Management Gateway tree (i.e. everything starting with *<ROOTOID>*) from server with IP address 192.168.0.1 via SNMPv1, run

```
$ snmpwalk -v1 -c public -t 15 192.168.0.1 1.3.6.1.4.1.16394.2.3.1
```

or, assuming SGP-MIB is installed at SNMP client system, run

```
$ snmpwalk -v1 -c public -t 15 192.168.0.1 SGP-MIB::sgp
```

Examples in this chapter refer to OIDs in numeric form e.g. *<ROOTOID>.1.1.4.0* which means the variable can be accessed via the following command:

```
$ snmpwalk -v1 -c public -t 15 192.168.0.1 SGP-MIB::sgp.1.1.4.0
```

28.1 Guardian Management Gateway specific data types

The Management Information Base (MIB) file defines several additional data types like *SensorType*, *SensorUnit*, *SensorCategory*, *EventType*, *SeverityType*, *ControlType* and *ControlOutput* that are used in variable description below. To avoid text duplication, this chapter describes such data types in the tables below.

Table 7: *SensorType* values

Value	Description		Value	Description
1	Temperature		26	Other FRU
2	Voltage		27	Cable Interconnect
3	Current		28	Terminator
4	Fan (Tachometer)		29	System Boot Initiated
5	Physical Security		30	Boot Error
6	Platform Violation		31	OS Boot
7	Processor		32	OS Critical Stop
8	Power Supply		33	Slot Connector
9	Power Unit		34	System ACPI Power State
10	Cooling Device		35	Reserved
11	Other Units Based Sensor		36	Platform Alert
12	Memory		37	Entity Presence
13	Drive Slot		38	Monitor ASIC IC
14	Post Memory Resize		39	LAN
15	System FW Progress		40	Management Subsystem Health
16	Event Logging Disabled		41	Battery
17	Reserved		42	System Audit
18	System Event		43	Version Change
19	Critical Interrupt		160	Operational
20	Button		192	OEM Sensor
21	Module Board		65537	Comm Channel Link State
22	Microcontroller Coprocessor		65538	Management Bus State
23	Add In Card		65539	Comm Channel Bus State
24	Chassis		65540	Config Data
25	Chipset		65541	Power Budget

Table 8: *SensorUnit* values

Value	Description		Value	Description
-1	Unspecified		46	Ft-Lb
1	Degrees C		47	Oz-In
2	Degrees F		48	Gauss
3	Degrees K		49	Gilberts
4	Volts		50	Henry
5	Amps		51	Millihenry
6	Watts		52	Farad
7	Joules		53	Microfarad
8	Coulombs		54	Ohms
9	VA		55	Siemens
10	Nits		56	Mole
11	Lumen		57	Becquerel
12	Lux		58	Ppm
13	Candela		59	reserved
14	Kpa		60	Decibels
15	Psi		61	Dba
16	Newton		62	Dbc
17	Cfm		63	Gray

Value	Description		Value	Description
18	Rpm		64	Sievert
19	Hz		65	Color Temp Degrees K
20	Microseconds		66	Bits
21	Milliseconds		67	Kilobits
22	Seconds		68	Megabits
23	Minutes		69	Gigabits
24	Hours		70	Bytes
25	Days		71	Kilobytes
26	Weeks		72	Megabytes
27	Mil		73	Gigabytes
28	Inches		74	Words
29	Feet		75	DWords
30	Cubic Inches		76	QWords
31	Cubic Feet		77	Lines
32	mm		78	Hits
33	cm		79	Misses
34	m		80	Retries
35	Cubic cm		81	Resets
36	Cubic m		82	Overruns
37	Liters		83	Underruns
38	Fluid Ounce		84	Collisions
39	Radians		85	Packets
40	Steradians		86	Messages
41	Revolutions		87	Characters
42	Cycles		88	errors
43	Gravities		89	Correctable Errors
44	Ounces		90	Uncorrectable Errors
45	Pounds			

Table 9: *SensorCategory* values

Value	Description		Value	Description
-1	Unspecified		7	Severity
1	Threshold		8	Presence
2	Usage		9	Enable
3	State		10	Availability
4	Predicted Fail		11	Redundancy
5	Limit		126	Sensor Specific
6	Performance		127	Generic

Table 10: *EventType* values

Value	Description
1	Resource
2	Domain
3	Sensor
4	Sensor Enable Change
5	Hot Swap
6	Watchdog
7	HPI SW
8	OEM
9	User
10	DIMI
11	DIMI Update

Value	Description
12	FUMI

Table 11: *SeverityType* values

Value	Description
1	Critical
2	Major
3	Minor
4	Informational
5	OK
241	Debug
255	All

Table 12: *ControlType* values

Value	Description
1	Digital
2	Discrete
3	Analog
4	Stream
5	Text
193	OEM

Table 13: *ControlOutput* values

Value	Description		Value	Description
1	Generic		10	LCD Display
2	LED		11	OEM
3	Fan Speed		12	Generic Address
4	Dry Contact Closure		13	IP Address
5	Power Supply Inhibit		14	Resource ID
6	Audible		15	Power Budget
7	Front Panel Lockout		16	Activate
8	Power Interlock		17	Reset
9	Power State			

28.2 Configuration MIB variables

The variables defined in this section contain information about the Guardian Management Gateway configuration, including configuration of system, controls, sensors, managed sensors, managed sensor log. Currently, most of the configuration variables are read-only but in future, the number of read-write variables may be increased, to improve management capabilities via the SNMP interface.

Basic system configuration variables have the following OID, where *<var>* is the variable index:

<ROOTOID>.1.1.<var>.0

Table 14: Basic system configuration indices

Variable	Index	Type	Access mode	Description
shxCount	1	INTEGER	Read-only	The number of SHX units (RackChiller included) supported.
unitName	2	STRING	Read-write	System host name.
hardwareVersion	3	STRING	Read-only	Hardware version of the main board.
firmwareVersion	4	STRING	Read-only	System firmware version.
utcOffset	5	STRING	Read-only	UTC offset of the system time.
resourceCount	6	INTEGER	Read-only	The number of resources in the system.

Variable	Index	Type	Access mode	Description
totalSensorCount	7	INTEGER	Read-only	The number of external (physical) sensors.
managedSensorCount	8	INTEGER	Read-only	The number of managed sensors.
externalSensorsZCoordinateUnits	9	INTEGER	Read-only	External Sensor Z Coordinate units: Freeform Text (T) or Rack Units (U).
serverCount	10	INTEGER	Read-only	The number of entries in serverReachabilityTable.
model	11	STRING	Read-only	The device model name.
cascadedDeviceConnected	12	INTEGER (TruthValue)	Read-only	Reserved for future use
unitsTemperature	14	STRING	Read-only	The global temperature measurement units: Celsius or Fahrenheit.
unitsLength	15	STRING	Read-only	The global length measurement units: Meters or Feet.
unitsPressure	16	STRING	Read-only	The global pressure measurement units: PSI or Pascals.

For example, to retrieve the system firmware version, use the following OID:

<ROOTOID>.1.1.4.0

```
snmpwalk -v1 -c private 80.240.102.34 SGP-MIB::unitConfiguration
SGP-MIB::shxCount.0 = INTEGER: 0
SGP-MIB::unitName.0 = STRING: Sgp000001
SGP-MIB::hardwareVersion.0 = STRING: 0.1
SGP-MIB::firmwareVersion.0 = STRING: 1.0.13 63998-20557 IoT
Aug 19 2021
18:28:15
SGP-MIB::utcOffset.0 = STRING: +0000
SGP-MIB::resourceCount.0 = INTEGER: 6
…..
```

Also, there is the networkConfigurationTable table in this section that contains parameters of the system network interfaces that have the following OIDs, where *<var>* is the variable index from the table below and *<entry>* is the entry number:

<ROOTOID>.1.1.13.1.<var>.<entry>

Table 15: Network interface table variables

Variable	Index	Type	Access mode	Description
networkInterfaceId	1	INTEGER	Read-only	Index of the network interface, equal to *<entry>*
networkInterfaceName	2	STRING	Read-only	Network interface name, e.g “eth0” or “wlan0”.
networkInterfaceMacAddress	3	STRING	Read-only	MAC Address.
networkInterfaceIPv4UseDHCP	4	INTEGER (TruthValue)	Read-only	Indicates whether IPv4 DHCP used: true (*1*) or false (*2*).
networkInterfaceIpV4Address	5	STRING	Read-only	IPv4 address with the number of significant bits in the network mask, e.g. “192.16.1.35/24".
networkInterfaceIpV4Gateway	6	STRING	Read-only	IPv4 gateway address.
networkInterfaceIPv6UseDHCP	7	INTEGER (TruthValue)	Read-only	Indicates whether IPv6 DHCP used: true (*1*) or false (*2*).
networkInterfaceIpV6Addresses	8	STRING	Read-only	IPv6 address with scope.

The following command retrieves information on network interfaces at the Guardian Management Gateway.

```
snmpwalk -v1 -c private 192.168.0.1 SGP-MIB::networkConfigurationTable
SGP-MIB::networkInterfaceName.1 = STRING: lo
SGP-MIB::networkInterfaceName.2 = STRING: eth0
SGP-MIB::networkInterfaceName.3 = STRING: eth1
SGP-MIB::networkInterfaceName.4 = STRING: sit0
SGP-MIB::networkInterfaceMacAddress.1 = STRING: 00:00:00:00:00:00
….
```

The shxConfiguration sub-branch contains details on Side Heat eXchangers (SHX) in the system in three tables: shxConfigurationTable, shxSensorCountTable and shxSensorConfigurationTable.

The shxConfigurationTable exposes SHX device parameters that have the following OIDs, where *<var>* is the variable index described below and *<resource>* is the resource ID of the SHX device.

<ROOTOID>.1.2.1.1.<var>.<resource>

Table 16: SHX Configuration table variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
shxResourceId	1	INTEGER	Read-only	Resource ID of the SHX device, equal to *<resource>*. Resource IDs for SHX devices are in the range *2000* to *2999*.
shxOperationalState	2	INTEGER	Read-write	The operational state of the SHX controller: disconnected(*−1*), offline(*2*) or online(*1*). To switch SHX power state while controller is connected, set shxOperationalState to *2* (offline) or *1* (online).
shxValvePosition	3	INTEGER	Read-only	The current opening state of the water valve (in percentages from 0 to 100).
shxCoolerTempSetpoint	4	INTEGER	Read-write	The setpoint for the desired temperature.
shxFanPerformanceSetpoint	5	INTEGER	Read-write	The fan performance setpoint, in percents
shxMaximumCooling	6	INTEGER (TruthValue)	Read-write	Indicates whether maximum cooling is requested (*1*) or not (*2*). To request maximum cooling, set shxMaximumCoolingState to *1* (true).
shxAlertState	7	INTEGER (TruthValue)	Read-write	Indicates whether SHX controller is in alert state (*1*) or not (*2*). To acknowledge alert status, set shxAlertState to *2* (false).
shxModel	8	STRING	Read-only	The model identifier of an SHX controller
shxFirmwareVersion	9	STRING	Read-only	The firmware version of an SHX controller

For example, to retrieve firmware versions of SHX devices, use the following OID:

<ROOTOID>.1.2.1.1.9

The shxSensorCountTable exposes the number of sensors of SHX devices that have the following OIDs, where *<resource>* is the resource ID of SHX device.

<ROOTOID>.1.2.2.1.2.<resource>

The shxSensorConfigurationTable exposes sensor parameters of SHX devices that have the following OIDs, where *<var>* is the variable index described below, *<resource>* is the resource ID of SHX device and *<sensor>* is the sensor number.

<ROOTOID>.1.2.3.1.<var>.<resource>.<sensor>

Table 17: SHX Sensor Configuration table variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
shxSensorId	1	INTEGER	Read-only	Sensor ID of the SHX device, equal to *<sensor>*.
shxInterface	2	INTEGER	Read-only	SHX sensor interface number.
shxAddress	3	INTEGER	Read-only	SHX sensor device address.
shxSensorName	4	STRING	Read-only	User-defined name of the sensor (e.g. Fan Speed 1).
shxSensorType	5	SensorType	Read-only	The sensor type. This data type is described in Table 7.
shxSensorCategory	6	SensorCategory	Read-only	The sensor category. This data type is described in Table 9.
shxSensorEnableControl	7	INTEGER (TruthValue)	Read- only	Indicates whether sensor control is enabled(*1*) or disabled(*2*).
shxSensorEventControl	8	INTEGER	Read-only	The sensor event control: Per-Event(*1*), Read-Only Masks(*2*) or Read-Only(*3*).
shxSensorAssertEventMask	9	STRING	Read-only	Bitmask of allowed Assertion events from the sensor, e.g. *0x003F*.
shxSensorDeassertEventMask	10	STRING	Read-only	Bitmask of allowed Deassertion events from the sensor, e.g. *0x003F*.
shxSensorIsReadingSupported	11	INTEGER (TruthValue)	Read-only	Indicates whether sensor reading is supported(*1*) or not supported(*2*).
shxSensorBaseUnit	12	SensorUnit	Read-only	The base units (this data type is described in Table 8). This parameter does not apply to discrete sensors.
shxSensorModifierUnit	13	SensorUnit	Read-only	The sensor modifier unit (this data type is described in Table 8 in the section 28.1).
shxSensorModifierUse	14	INTEGER	Read-only	A sensor modifier unit use: Basic Over Modifier(1), Basic Times Modifier(2) or None(-1).
shxSensorPercentage	15	INTEGER (TruthValue)	Read-only	Indicated whether the sensor reading is returned in percents (1) or not (2).
shxSensorAccuracy	16	FLOAT64	Read-only	The sensor accuracy.
shxSensorResolution	17	FLOAT64	Read-only	The sensor resolution.
shxSensorTolerance	18	FLOAT64	Read-only	The sensor tolerance.
shxSensorMaximum	19	FLOAT64	Read-only	The largest possible value. This parameter does not apply to discrete sensors.
shxSensorMinimum	20	FLOAT64	Read-only	The smallest possible value. This parameter does not apply to discrete sensors.
shxSensorThresholdsIsAccessible	21	INTEGER (TruthValue)	Read-only	Indicates whether sensor thresholds are accessible(*1*) or not (*2*).
shxSensorLowerCriticalThreshold	22	FLOAT64	Read- write	The lower critical threshold. This parameter does not apply to discrete sensors.
shxSensorLowerMajorThreshold	23	FLOAT64	Read- write	The lower major threshold. This parameter does not apply to discrete sensors.

Variable	Index	Type	Access Mode	Description
shxSensorLowerMinorThreshold	24	FLOAT64	Read- write	The lower minor threshold. This parameter does not apply to discrete sensors.
shxSensorUpperCriticalThreshold	25	FLOAT64	Read- write	The upper critical threshold. This parameter does not apply to discrete sensors.
shxSensorUpperMajorThreshold	26	FLOAT64	Read- write	The upper major threshold. This parameter does not apply to discrete sensors.
shxSensorUpperMinorThreshold	27	FLOAT64	Read- write	The upper minor threshold. This parameter does not apply to discrete sensors.
shxSensorPositiveHysteresis	28	FLOAT64	Read- write	The positive hysteresis used for deassertions. This parameter does not apply to discrete sensors.
shxSensorNegativeHysteresis	29	FLOAT64	Read- write	The negative hysteresis used for deassertions. This parameter does not apply to discrete sensors.
shxSensorPollInterval	30	INTEGER	Read- write	The sensor polling interval in milliseconds.
shxSensorAssertionDelayCount	31	INTEGER	Read- write	The delay measured in samples before a state is asserted. If the value is zero, then the state is asserted as soon as it is detected; if it is non-zero, say n, then the assertion condition must exist for *n+1* consecutive samples before the corresponding assertion event is reported.

For example, to retrieve all SHX sensor names, use the following OID:

<ROOTOID>.1.2.3.1.4

The managedSensorConfigurationTable exposes sensor parameters of managed sensors (i.e. virtual replicas of physical sensors located at resource *0*) that have the following OIDs, where *<var>* is the variable index described below and *<msensor>* is the managed sensor number.

<ROOTOID>.1.3.1.<var>.<msensor>

Table 18: Managed sensor configuration table variables

Variable	Index	Type	Access Mode	Description
managedSensorId	1	INTEGER	Read-only	Managed sensor ID, equal to *<msensor>*.
managedSensorType	2	SensorType	Read-only	The sensor type. This data type is described in the previous chapter.
managedSensorName	3	STRING	Read- write	The user-defined name of the sensor (e.g. Fan Speed 1); defaults to the original physical sensor name if not changed by the user.
managedSensorDescription	4	STRING	Read- write	The user-defined description of the sensor.
managedSensorXCoordinate	5	STRING	Read- write	The X coordinate of the sensor location.
managedSensorYCoordinate	6	STRING	Read- write	The Y coordinate of the sensor location.
managedSensorZCoordinate	7	STRING	Read- write	The Z coordinate of the sensor location.

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
managedSensorSubtype	8	STRING	Read- write	Type of measurement in case the sensor type is discrete.
managedSensorCategory	9	SensorCategory	Read-only	The sensor category. This data type is described in Table 9.
managedSensorEnableControl	10	INTEGER (TruthValue)	Read- only	Indicates whether sensor control is enabled(*1*) or disabled(*2*).
managedSensorEventControl	11	INTEGER	Read-only	The sensor event control: Per-Event(*1*), Read-Only Masks(*2*) or Read-Only(*3*).
managedSensorAssertEventMask	12	STRING	Read-only	Bitmask of allowed Assertion events from the sensor, e.g. *0x003F*.
managedSensorDeassertEventMask	13	STRING	Read-only	Bitmask of allowed Deassertion events from the sensor, e.g. *0x003*.
managedSensorIsReadingSupported	14	INTEGER (TruthValue)	Read-only	Indicates whether sensor reading is supported(*1*) or not supported(*2*).
managedSensorBaseUnit	15	SensorUnit	Read-only	The base units (this data type is described in Table 8). This parameter does not apply to discrete sensors.
managedSensorModifierUnit	16	SensorUnit	Read-only	The sensor modifier unit (data type is described in Table 8).
managedSensorModifierUse	17	INTEGER	Read-only	A sensor modifier unit use: Basic Over Modifier(*1*), Basic Times Modifier(*2*) or None(*−1*).
managedSensorPercentage	18	INTEGER (TruthValue)	Read-only	Indicated whether the sensor reading is returned in percents (*1*) or not (*2*).
managedSensorAccuracy	19	FLOAT64	Read-only	The sensor accuracy: how close (in percents) the measurement is to the actual value. This parameter does not apply to discrete sensors.
managedSensorResolution	20	FLOAT64	Read-only	The sensor resolution: the minimum difference between any two measured values. This parameter does not apply to discrete sensors.
managedSensorTolerance	21	FLOAT64	Read-only	The sensor tolerance: the difference between a sensor value and the actual value. This parameter does not apply to discrete sensors.
managedSensorMaximum	22	FLOAT64	Read-only	The biggest possible value. This parameter does not apply to discrete sensors.
managedSensorMinimum	23	FLOAT64	Read-only	The smallest possible value. This parameter does not apply to discrete sensors.
managedSensorThresholdsIsAccessible	24	INTEGER (TruthValue)	Read-only	Indicates whether sensor thresholds are accessible (*1*) or not (*2*).

Variable	Index	Type	Access mode	Description
managedSensorLowerCriticalThreshold	25	FLOAT64	Read- write	The lower critical threshold. This parameter does not apply to discrete sensors.
managedSensorLowerMajorThreshold	26	FLOAT64	Read- write	The lower major threshold. This parameter does not apply to discrete sensors.
managedSensorLowerMinorThreshold	27	FLOAT64	Read- write	The lower minor threshold. This parameter does not apply to discrete sensors.
managedSensorUpperCriticalThreshold	28	FLOAT64	Read- write	The upper critical threshold. This parameter does not apply to discrete sensors.
managedSensorUpperMajorThreshold	29	FLOAT64	Read- write	The upper major threshold. This parameter does not apply to discrete sensors.
managedSensorUpperMinorThreshold	30	FLOAT64	Read- write	The upper minor threshold. This parameter does not apply to discrete sensors.
managedSensorPositiveHysteresis	31	FLOAT64	Read- write	The positive hysteresis used for deassertions. This parameter does not apply to discrete sensors.
managedSensorNegativeHysteresis	32	FLOAT64	Read- write	The negative hysteresis used for deassertions. This parameter does not apply to discrete sensors.
managedSensorPollInterval	33	INTEGER	Read- write	The sensor polling interval in milliseconds.
managedSensorAssertionDelayCount	34	INTEGER	Read- write	The delay measured in samples before a state is asserted. If the value is zero, then the state is asserted as soon as it is detected; if it is non-zero, say n, then the assertion condition must exist for *n+1* consecutive samples before the corresponding assertion event is reported.
managedSensorResourceId	35	INTEGER	Read-only	The resource number of the original physical sensor.
managedSensorExternalSensorNumber	36	INTEGER	Read-only	The sensor number of the original physical sensor.

For example, to retrieve user-defined descriptions of all managed sensors, use the following OID:

<ROOTOID>.1.3.1.4

The controlConfigurationTable exposes parameters of controls that have the following OIDs, where *<var>* is the variable index described below, *<resource>* is the resource ID and *<control>* is the control number.

<ROOTOID>.1.4.1.<var>.<resource>.<control>

Table 19: Control configuration table variables

Variable	Index	Type	Access mode	Description
ctrlResourceId	1	INTEGER	Read-only	The resource number of the control, equal to *<resource>*.
ctrlId	2	INTEGER	Read-only	The control number, equal to *<control>*.
ctrlType	3	ControlType	Read-only	The control type. This data type is described in the Table 12.

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
ctrlOutputType	4	ControlOutput	Read-only	The control output type. This data type is described in the Table 13.
ctrlMaximumValue	5	INTEGER	Read-only	The maximum value of the control.
ctrlMinimumValue	6	INTEGER	Read-only	The minimum value of the control.
ctrlDefaultValue	7	INTEGER	Read-only	The default value of the control.
ctrlDefaultMode	8	INTEGER	Read-only	The default mode of the control: automatic(*1*), manual(*2*) or unavailable(*−1*).
ctrlDefaultModeReadOnly	9	INTEGER (TruthValue)	Read-only	Indicates whether the default control mode is read-only(*1*) or not(*2*).
ctrlWriteOnly	10	INTEGER (TruthValue)	Read-only	Indicates whether the control is write-only(*1*) or not(*2*).
ctrlOem	11	INTEGER	Read-only	An OEM specific value in the control definition.
ctrlName	12	STRING	Read-write	The name of the control (e.g. Fan Speed setpoint 1)

For example, to retrieve control types of all controls in the system, use the following OID:

<ROOTOID>.1.4.1.3

The logConfiguration sub-branch exposes sensor log parameters with the following OIDs, where *<var>* is the variable index described in the table below:

<ROOTOID>.1.5.<var>.0

Table 20: Log configuration indices

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
logDataRetrieval	1	INTEGER (TruthValue)	Read- write	Indicates if log data retrieval is enabled(*1*) or disabled(*2*).
logMeasurementPeriod	2	INTEGER	Read- write	Data sample collection periodicity in seconds.
logSize	3	INTEGER	Read-only	The number of entries in the sensor log.

For example, to retrieve the current sensor log size, use the following OID:

<ROOTOID>.1.5.3.0

The externalSensorConfigurationTable table exposes means to configure external (physical) sensors. This table is indexed with the resource ID and sensor number. Variables from this table have the following OIDs, where *<var>* is the variable index from the table below:

<ROOTOID>.1.6.1.<var>.<resource>.<sensor>

Table 21: External Sensor Configuration table variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
externalResourceId	1	INTEGER	Read-only	The resource number of the physical sensor.
externalSensorId	2	INTEGER	Read-only	The sensor number, unique for each sensor within a resource.
externalResourceName	3	STRING	Read- write	The name of the resource.
externalSensorName	4	STRING	Read- write	The name of the sensor (e.g. Fan Speed 1).
externalSensorType	5	SensorType	Read-only	The sensor type. This data type is described in the Table 7
externalSensorCategory	6	SensorCategory	Read-only	The sensor category. This data type is described in the Table 9.
externalSensorEnableControl	7	INTEGER (TruthValue)	Read-only	Indicates whether sensor control is enabled(*1*) or disabled(*2*).

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
externalSensorEventControl	8	INTEGER	Read-only	The sensor event control: Per-Event(1), Read-Only Masks(2) or Read-Only(3).
externalSensorAssertEventMask	9	STRING	Read-only	Bitmask of allowed Assertion events from the sensor, e.g. 0x003F.
externalSensorDeassertEventMask	10	STRING	Read-only	Bitmask of allowed Deassertion events from the sensor, e.g. 0x003F.
externalSensorIsReadingSupported	11	INTEGER (TruthValue)	Read-only	Indicates whether sensor reading is supported(1) or not supported(2).
externalSensorBaseUnit	12	SensorUnit	Read-only	The base units (data type is described in the Table 8). This parameter does not apply to discrete sensors.
externalSensorModifierUnit	13	SensorUnit	Read-only	The sensor modifier unit (data type is described in the Table 8).
externalSensorModifierUse	14	INTEGER	Read-only	A sensor modifier unit use: Basic Over Modifier(1), Basic Times Modifier(2) or None(–1).
externalSensorPercentage	15	INTEGER (TruthValue)	Read-only	Indicated whether the sensor reading is returned in percents (1) or not (2).
externalSensorAccuracy	16	FLOAT64	Read-only	The accuracy: how close (in percents) the measurement is to the actual value. This parameter does not apply to discrete sensors.
externalSensorResolution	17	FLOAT64	Read-only	The resolution: the minimum difference between any two measured values. This parameter does not apply to discrete sensors.
externalSensorTolerance	18	FLOAT64	Read-only	The tolerance: the difference between a sensor value and the actual value. This parameter does not apply to discrete sensors.
externalSensorMaximum	19	FLOAT64	Read-only	The biggest possible value. This parameter does not apply to discrete sensors.
externalSensorMinimum	20	FLOAT64	Read-only	The smallest possible value. This parameter does not apply to discrete sensors.
externalSensorThresholdsIsAccessible	21	INTEGER (TruthValue)	Read-only	Indicates whether sensor thresholds are accessible (1) or not (2).
externalSensorLowerCriticalThreshold	22	FLOAT64	Read-write	The lower critical threshold. This parameter does not apply to discrete sensors.

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
externalSensorLowerMajorThreshold	23	FLOAT64	Read-write	The lower major threshold. This parameter does not apply to discrete sensors.
externalSensorLowerMinorThreshold	24	FLOAT64	Read-write	The lower minor threshold. This parameter does not apply to discrete sensors.
externalSensorUpperCriticalThreshold	25	FLOAT64	Read-write	The upper critical threshold. This parameter does not apply to discrete sensors.
externalSensorUpperMajorThreshold	26	FLOAT64	Read-write	The upper major threshold. This parameter does not apply to discrete sensors.
externalSensorUpperMinorThreshold	27	FLOAT64	Read-write	The upper minor threshold. This parameter does not apply to discrete sensors.
externalSensorPositiveHysteresis	28	INTEGER	Read- write	The positive hysteresis used for deassertions. This parameter does not apply to discrete sensors.
externalSensorNegativeHysteresis	29	INTEGER	Read- write	The negative hysteresis used for deassertions. This parameter does not apply to discrete sensors.
externalSensorPollInterval	30	INTEGER	Read- write	The sensor polling interval in milliseconds.
externalSensorAssertionDelayCount	31	INTEGER	Read- write	The delay measured in samples before a state is asserted. If the value is zero, then the state is asserted as soon as it is detected; if it is non-zero, say n, then the assertion condition must exist for *n+1* consecutive samples before the corresponding assertion event is reported.
externalSensorIsManaged	32	INTEGER (TruthValue)	Read-write	Indicates if the sensor is managed (*1*), or not (*2*). Set to *1* to manage this sensor, set to *2* to unmanage it.
externalSensorManagedNumber	33	INTEGER	Read-only	The sensor number of the corresponding managed sensor on resource *0* or *-1* if the sensor is not managed.

For example, to retrieve names of all physical sensors, use the following OID:

<ROOTOID>.1.6.1.5

```
snmpwalk -v1 -c private 80.240.102.34 SGP-MIB::externalSensorConfigurationTable
SGP-MIB::externalResourceId.1000.1 = INTEGER: 1000
SGP-MIB::externalResourceId.1000.2 = INTEGER: 1000
SGP-MIB::externalResourceId.1000.3 = INTEGER: 1000
….
SGP-MIB::externalSensorAssertionDelayCount.4002.3718 = Gauge32: 0
SGP-MIB::externalSensorAssertionDelayCount.4002.3719 = Gauge32: 0
SGP-MIB::externalSensorIsManaged.1000.1 = INTEGER: true(1)
SGP-MIB::externalSensorIsManaged.1000.2 = INTEGER: true(1)
```

```
SGP-MIB::externalSensorIsManaged.1000.3 = INTEGER: true(1)
SGP-MIB::externalSensorIsManaged.1000.4 = INTEGER: false(2)
SGP-MIB::externalSensorIsManaged.1000.5 = INTEGER: false(2)
SGP-MIB::externalSensorIsManaged.1001.1 = INTEGER: false(2)
SGP-MIB::externalSensorIsManaged.1001.2 = INTEGER: false(2)
…
```

28.3 Log MIB variables

The log branch exposes sensor log for managed sensors. This branch contains log properties variables and two tables for log timestamps and for managed sensor states. The logProperties variables have the following OIDs:

<ROOTOID>.2.1.<var>

Table 22: Log properties variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
logOldestId	1	INTEGER	Read-only	Index of the oldest sample in the log.
logNewestId	2	INTEGER	Read-only	Index of the newest sample in the log.

The logTimeStampTable contains timestamps or each reading sample. By default, the log contains 16 samples. This table has the following OID, where *<var>* is the variable index and *<msensor>* is the managed sensor number:

<ROOTOID>.2.2.1.<var>.<entry>

Table 23: Log timestamp table variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
logEntryIdx	1	INTEGER	Read-only	Log entry index, equal to <entry>.
logEntryTimeStamp	2	STRING	Read-only	The time when the data was collected. It is measured in seconds relative to January 1, 1970 (midnight UTC/GMT), i.e. a value of *0* indicates January 1, 1970 (midnight UTC/GMT)

The logManagedSensorTable table contains reading samples for managed sensors. The entries of this table have the following OID, where *<var>* is the variable index described in the table below and *<msensor>* is the managed sensor number and *<entry>* is the number of a specific log entry for the sensor:

<ROOTOID>.2.3.1.<var>.<msensor>.<entry>

Table 24: Log of managed sensors table variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
logManagedSensorDataAvailable	1	INTEGER (TruthValue)	Read-only	Indicates data availability for this sensor: *1* if available, *2* otherwise.
logManagedSensorReadingCount	2	INTEGER	Read-only	The count of successfully obtained sensor readings during the period.
logManagedSensorEventStateCount	3	INTEGER	Read-only	The count of successfully obtained sensor even state words during the period.
logManagedSensorAvgValue	4	FLOAT64	Read-only	The average value across sensor readings for the period.
logManagedSensorMinValue	5	FLOAT64	Read-only	The minimum value across sensor readings for the period
logManagedSensorMaxValue	6	FLOAT64	Read-only	The maximum value across sensor readings for the period
logManagedSensorDispValue	7	FLOAT64	Read-only	The dispersion across sensor readings for the period.
logManagedSensorAccState	8	INTEGER	Read-only	The accumulated event state (the logical OR of all sensor event states obtained during the period).

For example, to retrieve average values for logged managed sensor readings, use the following OID:

<ROOTOID>.2.3.1.4

The following command retrieves average readings of managed sensor 1 (a temperature sensor).

```
snmpwalk -v1 -c private 192.168.0.1 SGP-MIB::logManagedSensorAvgValue.1
SGP-MIB::logManagedSensorAvgValue.1.1 = Opaque: Float: 29.282292
SGP-MIB::logManagedSensorAvgValue.1.2 = Opaque: Float: 29.314583
…
```

28.4 Measurements MIB variables

The measurements branch represents all sensor reading in Guardian Management Gateway, including managed sensors i.e. virtual replicas of physical sensors attached to resource *0*. This branch contains two tables for managed and physical (external) sensors. The measurementsManagedSensorTable table has the following OID, where *<var>* is the variable index and *<msensor>* is the managed sensor number:

<ROOTOID>.3.1.1.<var>.<msensor>

Table 25: Managed sensor table variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
measurementsManagedSensorIsAvailable	1	INTEGER (TruthValue)	Read-only	Indicates data availability for the sensor during this measurement period: *1* if available, *2* otherwise.
measurementsManagedSensorState	2	INTEGER	Read-only	The current event state mask for the sensor.
measurementsManagedSensorValue	3	FLOAT64	Read-only	The sensor reading. This parameter does not apply to discrete sensors
measurementsManagedSensorTimeStamp	4	STRING	Read-only	The sensor reading timestamp.

For example, to retrieve readings of all managed sensors, use the following OID:

<ROOTOID>.3.1.1.3

The following command retrieves readings of managed sensors (all three are temperature sensors).

```
snmpwalk -v1 -c private 192.168.0.1 SGP-MIB::measurementsManagedSensorValue
SGP-MIB::measurementsManagedSensorValue.1 = Opaque: Float: 29.300000
SGP-MIB::measurementsManagedSensorValue.2 = Opaque: Float: 29.300000
SGP-MIB::measurementsManagedSensorValue.3 = Opaque: Float: 15.300000
```

The measurementsExternalSensorTable table is indexed by resource ID and control number. The entries of this table have the following OID, where *<var>* is the variable index:

<ROOTOID>.3.2.1.<var>.<resource>.<sensor>

Table 26: External sensor table variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
measurementsExternalSensorIsAvailable	1	INTEGER (TruthValue)	Read-only	Indicates data availability for the sensor during this measurement period: *1* if available, *2* otherwise.
measurementsExternalSensorState	2	INTEGER	Read-only	The current sensor state.
measurementsExternalSensorValue	3	FLOAT64	Read-only	The sensor reading.
measurementsExternalSensorTimeStamp	4	STRING	Read-only	The sensor reading timestamp.

For example, to retrieve states of all physical sensors, use the following OID:

<ROOTOID>.3.2.1.2

```
snmpwalk -v1 -c private 80.240.102.34 SGP-MIB::measurementsExternalSensorTable
SGP-MIB::measurementsExternalSensorIsAvailable.1000.1 = INTEGER: true(1)
SGP-MIB::measurementsExternalSensorIsAvailable.1000.2 = INTEGER: true(1)
SGP-MIB::measurementsExternalSensorIsAvailable.1000.3 = INTEGER: true(1)
```

```
SGP-MIB::measurementsExternalSensorIsAvailable.1000.4 = INTEGER: false(2)
SGP-MIB::measurementsExternalSensorIsAvailable.1000.5 = INTEGER: false(2)
SGP-MIB::measurementsExternalSensorIsAvailable.1001.1 = INTEGER: true(1)
SGP-MIB::measurementsExternalSensorIsAvailable.1001.2 = INTEGER: true(1)
…
```

28.5 Controls MIB variables

All the controls in Guardian Management Gateway are exposed in separate SNMP branch named "controls" that contains single table controlsTable indexed by resource ID and control number. The entries of this table have the following OID, where *<var>* is the variable index:

<ROOTOID>.4.1.<var>.<resource>.<control>

Table 27: Control table variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
ctrlMode	1	INTEGER	Read-only	The actual control mode: automatic(*1*), manual(*2*) or unavailable(*-1*).
ctrlState	2	INTEGER	Read-write	The actual control state. For write-only controls: *0* in the read-mode.
ctrlCachedState	3	INTEGER	Read-write	The cached control state for slow devices. For write-only controls: *0* in the read-mode.

For example, to retrieve the actual state of all controls, use the following OID:

<ROOTOID>.4.1.1.2

The following commands turn MCB "buzzer" on then off.

```
snmpset -v1 -c private 192.168.0.1 SGP-MIB::ctrlState.3000.1 i 1
SGP-MIB::ctrlState.3000.1 = INTEGER: 1
snmpset -v1 -c private 192.168.0.1 SGP-MIB::ctrlState.3000.1 i 0
SGP-MIB::ctrlState.3000.1 = INTEGER: 0
```

28.6 serverReachability MIB variables

The server reachability variables are represented by a single table with the following OID, where *<var>* is the index of a variable in the table of reachability attributes and *<entry>* is the number of the table entry.

<ROOTOID>.5.1.<var>.<entry>

Table 28: Server reachability variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
serverId	1	INTEGER	Read-only	Table entry index, equal to *<entry>*.
serverIpAddress	2	STRING	Read-write	Host Name or IP address of the target system.
serverPingEnabled	3	INTEGER (TruthValue)	Read-write	*1* – if periodic poll of the target system via the ping command is enabled, *2* – otherwise.
serverReachable	4	INTEGER (TruthValue)	Read-only	*1* – if the target system is responding, *2* – otherwise.
serverUnreachable	5	INTEGER (TruthValue)	Read-only	*1* – if the target system is not responding, *2* – otherwise.

Normally, this table contains entries for external network servers needed for Guardian Management Gateway operations e.g. DNS, NTP and DHCP servers, so that it's easy to diagnose network issues at Guardian Management Gateway via the SNMP interface. For example, to retrieve target addresses from the server reachability table, use the following OID:

<ROOTOID>.5.1.2

The following command retrieves the entire server reachability table.

```
snmpwalk -v1 -c public 192.168.0.1 SGP-MIB::serverReachabilityTable
SGP-MIB::serverId.1 = INTEGER: 1
SGP-MIB::serverId.2 = INTEGER: 2
```

```
SGP-MIB::serverIpAddress.1 = STRING: 192.168.0.1
...
```

28.7 sel MIB variables

The sel branch provides access to the System Event Log parameters and entries and allows clearing the log.
System Event Log parameters have the following OID, where *<var>* is the variable index:

<ROOTOID>.7.<var>.0

Table 29: System Event Log variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
selEntriesCount	1	INTEGER	Read-only	The current number of entries in the SEL.
selSize	2	INTEGER	Read-only	SEL capacity (the maximum number of entries that SEL can contain).
selUpdateTimestamp	3	STRING	Read-only	The timestamp of the latest SEL update.
selCurrentTime	4	STRING	Read-only	The current SEL time.
selEnabled	5	INTEGER (TruthValue)	Read-only	Indicates if the SEL is enabled (*1*) or disabled (*2*).
selOverflowFlag	6	INTEGER (TruthValue)	Read-only	Indicates if the SEL is overflown (*1*) or not (*2*).
selOverflowAction	7	INTEGER (TruthValue)	Read-only	The overflow mode action for new entries: drop (*1*) or overwrite (*2*).
selClear	8	INTEGER (TruthValue)	Read-write	Set to *1* to cleat SEL. Value *2* means SEL clear is not in progress.

For example, to retrieve the current number of the system event log entries, use the following OID:

<ROOTOID>.7.1.0

Also, there is the selTable table in this section that contains the log entries with parameters that have the following OIDs, where *<var>* is the variable index from the table below and *<entry>* is the entry number:

<ROOTOID>.7.9.1.<var>.<entry>

Table 30: System Event Log table variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
selEntryId	1	INTEGER	Read-only	SEL entry number, equal to *<entry>*
selTimestamp	2	STRING	Read-only	Time of the entry addition into the log.
selEventType	3	EventType	Read-only	The event type (this data type is described in the Table 10).
selResourceId	4	INTEGER	Read-only	Resource ID of the event source.
selEventTimestamp	5	STRING	Read-only	Timestamp of the event generation.
selSeverity	6	SeverityType	Read-only	The event severity (this data type is described in the Table 11).
selEventSubType	7	INTEGER	Read-only	Specific event type for resource events, software events, upgrade status for FUMI events.
selSensorNum	8	INTEGER	Read-only	Sensor number of the event source.
selSensorType	9	SensorType	Read-only	The sensor type (this data type is described in the Table 7).
selEventCategory	10	EventCategory	Read-only	The event category (this data type is described in the Table 9).
selAssertionEvent	11	INTEGER (TruthValue)	Read-only	Indicates if the event is an assertion event (*1*) or a deassertion event (*2*).
selEventState	12	INTEGER	Read-only	The specific state of the sensor that triggered the event.
selTriggerReading	13	FLOAT64	Read-only	Sensor reading value that triggered the event.

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
selTriggerThreshold	14	FLOAT64	Read-only	Sensor threshold value that was crossed at the event.
selPreviousStates	15	INTEGER	Read-only	The mask of previous sensor states (before the event).
selCurrentStates	16	INTEGER	Read-only	The mask of current sensor states (after the event).
selFumiNum	17	INTEGER	Read-only	The FUMI number (for FUMI events, normally *0*).
selBankNum	18	INTEGER	Read-only	The FUMI bank number (for FUMI events, normally *0*).

For example, to retrieve event severity for all SEL entries, use the following OID:
<ROOTOID>.7.9.1.6

28.8 sgpTrap MIB variables

The SNMP Trap messages are in this section. They are defined in SGP-MIB as the sgpTrap with the following OID:
<ROOTOID>.8
Currently there is only one supported trap that contains one variable in ASCII text format (JSON format to be more specific) describing an event in system event log. This variable has the following OID:
<ROOTOID>.sgpTrap(8).sgpTextTrap(1)
Depending on network service configuration, SNMP traps can be delivered in either SNMPv1 or SNMPv2 format. Below, you can see an example of such traps collected using the *snmptrapd* utility from the Net-SNMP package. The first one is in SNMPv1 format and the second is in SNMPv2 format, describing the same event.

```
# snmptrapd -d -f -m SGP-MIB
Starting snmptrapd 5.0.6

Received 365 bytes from 192.168.0.1
0000: 30 82 01 69  02 01 00 04  06 70 75 62  6C 69 63 A4    0..i.....public.
0016: 82 01 5A 06  0E 2B 06 01  04 01 81 80  0A 02 03 01    ..Z..+..........
0032: 08 00 01 40  04 50 F0 66  22 02 01 06  02 01 63 43    ...@.P.f".....cC
0048: 01 37 30 82  01 37 30 82  01 33 06 0D  2B 06 01 04    .70..70..3..+...
0064: 01 81 80 0A  02 03 01 08  01 04 82 01  20 7B 22 45    ............ {"E
0080: 76 65 6E 74  22 3A 7B 22  53 65 6E 73  6F 72 45 76    vent":{"SensorEv
0096: 65 6E 74 22  3A 7B 22 41  73 73 65 72  74 69 6F 6E    ent":{"Assertion
0112: 22 3A 74 72  75 65 2C 22  45 76 65 6E  74 43 61 74    ":true,"EventCat
0128: 65 67 6F 72  79 22 3A 22  54 68 72 65  73 68 6F 6C    egory":"Threshol
0144: 64 22 2C 22  45 76 65 6E  74 53 74 61  74 65 22 3A    d","EventState":
0160: 22 55 70 70  65 72 4D 69  6E 6F 72 54  68 72 65 73    "UpperMinorThres
0176: 68 6F 6C 64  43 72 6F 73  73 65 64 22  2C 22 53 65    holdCrossed","Se
0192: 6E 73 6F 72  4E 75 6D 62  65 72 22 3A  31 2C 22 53    nsorNumber":1,"S
0208: 65 6E 73 6F  72 54 79 70  65 22 3A 22  54 65 6D 70    ensorType":"Temp
0224: 65 72 61 74  75 72 65 22  2C 22 54 72  69 67 67 65    erature","Trigge
0240: 72 52 65 61  64 69 6E 67  22 3A 32 37  2E 38 31 32    rReading":27.812
0256: 2C 22 54 72  69 67 67 65  72 54 68 72  65 73 68 6F    ,"TriggerThresho
0272: 6C 64 22 3A  30 2E 30 7D  2C 22 53 65  76 65 72 69    ld":0.0},"Severi
0288: 74 79 22 3A  22 4D 69 6E  6F 72 22 2C  22 53 6F 75    ty":"Minor","Sou
0304: 72 63 65 22  3A 31 30 30  30 2C 22 54  69 6D 65 73    rce":1000,"Times
0320: 74 61 6D 70  22 3A 22 32  30 31 38 2D  31 31 2D 31    tamp":"2018-11-1
0336: 33 20 31 38  3A 31 32 3A  30 32 22 2C  22 54 79 70    3 18:12:02","Typ
0352: 65 22 3A 22  53 65 6E 73  6F 72 22 7D  7D              e":"Sensor"}}

192.168.0.1: Enterprise Specific Trap (99) Uptime: 0:00:00.55, SGP-
MIB::sgpTextTrap = STRING:
"{\"Event\":{\"SensorEvent\":{\"Assertion\":true,\"EventCategory\":\"Threshold\"
,\"EventState\":\"UpperMinorThresholdCrossed\",\"SensorNumber\":1,\"SensorType\"
:\"Temperature\",\"TriggerReading\":27.812,\"TriggerThreshold\":0.0},\"Severity\
```

```
":\"Minor\",\"Source\":1000,\"Timestamp\":\"2018-11-13
18:12:02\",\"Type\":\"Sensor\"}}"

Received 394 bytes from 192.168.0.1
0000: 30 82 01 86  02 01 01 04  06 70 75 62  6C 69 63 A7    0........public.
0016: 82 01 77 02  04 30 C0 B8  C0 02 01 00  02 01 00 30    ..w..0.........0
0032: 82 01 67 30  10 06 08 2B  06 01 02 01  01 03 00 43    ..g0...+.......C
0048: 04 03 3F E3  20 30 1C 06  0A 2B 06 01  06 03 01 01    ..?. 0...+......
0064: 04 01 00 06  0E 2B 06 01  04 01 81 80  0A 02 03 01    .....+..........
0080: 08 00 01 30  82 01 33 06  0D 2B 06 01  04 01 81 80    ...0..3..+......
0096: 0A 02 03 01  08 01 04 82  01 20 7B 22  45 76 65 6E    ......... {"Even
0112: 74 22 3A 7B  22 53 65 6E  73 6F 72 45  76 65 6E 74    t":{"SensorEvent
0128: 22 3A 7B 22  41 73 73 65  72 74 69 6F  6E 22 3A 74    ":{"Assertion":t
0144: 72 75 65 2C  22 45 76 65  6E 74 43 61  74 65 67 6F    rue,"EventCatego
0160: 72 79 22 3A  22 54 68 72  65 73 68 6F  6C 64 22 2C    ry":"Threshold",
0176: 22 45 76 65  6E 74 53 74  61 74 65 22  3A 22 55 70    "EventState":"Up
0192: 70 65 72 4D  69 6E 6F 72  54 68 72 65  73 68 6F 6C    perMinorThreshol
0208: 64 43 72 6F  73 73 65 64  22 2C 22 53  65 6E 73 6F    dCrossed","Senso
0224: 72 4E 75 6D  62 65 72 22  3A 31 2C 22  53 65 6E 73    rNumber":1,"Sens
0240: 6F 72 54 79  70 65 22 3A  22 54 65 6D  70 65 72 61    orType":"Tempera
0256: 74 75 72 65  22 2C 22 54  72 69 67 67  65 72 52 65    ture","TriggerRe
0272: 61 64 69 6E  67 22 3A 32  37 2E 38 31  32 2C 22 54    ading":27.812,"T
0288: 72 69 67 67  65 72 54 68  72 65 73 68  6F 6C 64 22    riggerThreshold"
0304: 3A 30 2E 30  7D 2C 22 53  65 76 65 72  69 74 79 22    :0.0},"Severity"
0320: 3A 22 4D 69  6E 6F 72 22  2C 22 53 6F  75 72 63 65    :"Minor","Source
0336: 22 3A 31 30  30 30 2C 22  54 69 6D 65  73 74 61 6D    ":1000,"Timestam
0352: 70 22 3A 22  32 30 31 38  2D 31 31 2D  31 33 20 31    p":"2018-11-13 1
0368: 38 3A 31 32  3A 35 39 22  2C 22 54 79  70 65 22 3A    8:12:59","Type":
0384: 22 53 65 6E  73 6F 72 22  7D 7D                       "Sensor"}}

build.nvent.com  [192.168.0.1]:  Trap  SNMPv2-SMI::mib-2.1.3.0  =  Timeticks:
(54518560)  6  days,  7:26:25.60,  SNMPv2-SMI::snmpModules.1.1.4.1.0  =  OID:  SGP-
MIB::sgpNotification1,          SGP-MIB::sgpTextTrap          =          STRING:
"{\"Event\":{\"SensorEvent\":{\"Assertion\":true,\"EventCategory\":\"Threshold\"
,\"EventState\":\"UpperMinorThresholdCrossed\",\"SensorNumber\":1,\"SensorType\"
:\"Temperature\",\"TriggerReading\":27.812,\"TriggerThreshold\":0.0},\"Severity\
":\"Minor\",\"Source\":1000,\"Timestamp\":\"2018-11-13
18:12:59\",\"Type\":\"Sensor\"}}"
```

This trap contains asserted event message from temperature sensor *1* at resource *1000* saying that Upper Minor Threshold value *0* was crossed by reading *27.812* and the severity of this event is minor.

To make the Guardian Management Gateway send an SNMP trap to the test host (IP address *192.168.0.1* in the example below) it is possible to use the chain of following CLI commands, assuming there is a temperature sensor *1* at resource *1000*.

```
CLI{admin}> filter add TestFilter "resource==1000 && sensor_number==1 &&
assertion==1"
CLI{admin}> action add TestFilter always snmptrap 192.168.0.1
CLI{admin}> sensor threshold set 1000 1 umn 0
CLI{admin}> sensor threshold set 1000 1 umn 50
```

28.9 Accessing Guardian Management Gateway via SNMP

Any SNMP client implementation should be able to access the Guardian Management Gateway variables defined in SGP-MIB. One specific choice is the Net-SNMP package from: http://net-snmp.sourceforge.net/ that is a part of all popular Linux distributions. This package should be installed on the management (client) system. It provides some basic management tools. To access the SNMP server on a Guardian Management Gateway, the *snmpget*, *snmpset* and *snmpwalk* commands can be used.

To install the MIB file on the management system, follow the instructions supplied with the package e.g. for Net-SNMP the *SGP-MIB.txt* file should be placed into the */usr/share/snmp/mibs* directory or specified via command line arguments.

After that, use the *snmpget* or *snmpwalk* commands to verify access. For SNMPv1 or SNMPv2c access, the community name is either public for read-only access or private for read-write access by default. For SNMPv3 access it is necessary to add SNMPv3 user first (see CLI *user snmp* commands). For example, you can use the following command to retrieve basic Guardian Management Gateway configuration:

snmpwalk –v2c -c public <Guardian IP address> SGP-MIB::firmwareVersion

or, if MIB file is not yet installed

snmpwalk –v2c -c public <Guardian IP address> .1.3.6.1.4.1.16394.2.3.1.1.1.4

The output will be similar to the following:

```
SGP-MIB::firmwareVersion.0 = STRING: "1.0.13 63998-20557 IoT\nAug 19 2021\n14:57:54"
```

To retrieve the entire SGP-MIB variables subtree, use the following command:

snmpwalk –v2c -c public –t 15 <Guardian IP address> SGP-MIB::sgp

This command takes about 5 minutes.

SNMPv3 access command has username, password and optionally privacy string, instead of community string in SNMPv1 or SNMPv2c, so the same command looks like this:

snmpwalk -v3 -l authPriv -a SHA -u myusername -A mypassword -x DES -X myprivacy -t 15 <Guardian IP address> SGP-MIB::sgp

Here is an example of creating a managed sensor from physical sensor *1* at resource *1000*:

snmpget -v1 -c private <Guardian IP address> SGP-MIB::externalSensorIsManaged.1000.1

```
SGP-MIB::externalSensorIsManaged.1000.1 = INTEGER: false(2)
```

There is no managed sensor yet. Create it now by setting this integer variable to *1*(TRUE):

snmpset -v1 -c private <Guardian IP address> SGP-MIB::externalSensorIsManaged.1000.1 i 1

```
SGP-MIB::externalSensorIsManaged.1000.1 = INTEGER: true(1)
```

Double-check the result:

snmpwalk -v1 -c private <Guardian IP address> SGP-MIB::externalSensorIsManaged.1000

```
SGP-MIB::externalSensorIsManaged.1000.1 = INTEGER: true(1)
SGP-MIB::externalSensorIsManaged.1000.2 = INTEGER: false(2)
SGP-MIB::externalSensorIsManaged.1000.3 = INTEGER: false(2)
SGP-MIB::externalSensorIsManaged.1000.4 = INTEGER: false(2)
SGP-MIB::externalSensorIsManaged.1000.5 = INTEGER: false(2)
```

29 Front Panel Display Interface

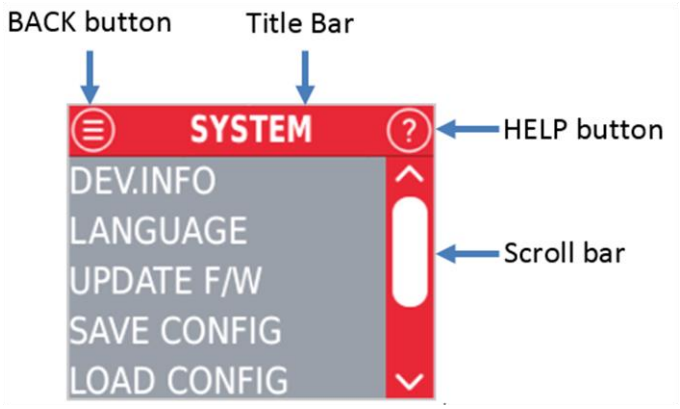
29.1 Overview

The Guardian Management Gateway implements a 2-inch color TFT display with a touch screen interface on the front panel. This display can be used to obtain basic information about the device (network configuration, serial number, etc.), to view sensor/alarm information, and to perform a number of system-level functions (such as firmware upgrades).

Note that due to physical restrictions, the front panel display supports only a subset of all Guardian Management Gateway user commands, and is less capable than the command line interface (CLI) or the Web interface. For example, it cannot be used to change configuration settings.
The Guardian Management Gateway supports automatic detection of the correct display orientation at boot time. If the device orientation changes while powered on, the orientation of the GUI on the front panel display will not change.
The front panel display GUI can be controlled with a finger using touch gestures. Tapping on a button or a menu item activates it. Tapping and dragging anywhere inside the view area can be used to scroll its contents up and down. If the display is not touched for 30 seconds, it goes blank to save power. To wake it up, tap anywhere on the screen.
The following sections describe the front panel display interface in more detail.

29.2 Main Interface Elements

The front panel display GUI is a simple menu-based interface that consists of a title bar at the top and a main viewing area in the middle:



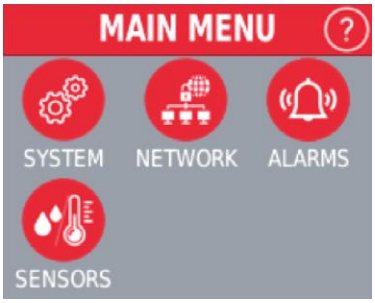
The title bar consists of the following elements, left to right:

- A BACK button, which can be used to go back to the previous view or menu at any time. Note that the BACK button is disabled when the main (i.e. top level) menu is displayed.
- A title area showing the name of the current view or menu.
- A HELP button, which can be used to display context-specific help information. The help dialog can be closed by tapping the BACK button.

The viewing area is used to display context-specific information (menus, sensor data, etc.). If the underlying text or graphics do not fit the viewing area, a vertical scroll bar is displayed on the right side of the screen. Scrolling can be performed by holding and dragging the scrollbar handle, or by tapping the arrow buttons. Alternatively, it is possible to scroll the viewing area contents by tapping and dragging anywhere inside the viewing area.

29.3 Main Menu

When the Guardian Management Gateway is powered up, an nVent/SCHROFF logo is displayed. When the boot process is complete, the nVent/SCHROFF logo is replaced by the Main Menu:



The Main Menu consists of the following items, each represented by a pictogram:

- SYSTEM: access various system-level functions, such as device information, firmware upgrade, saving/loading the configuration, etc.;
- NETWORK: show network-specific information (IP/MAC addresses, etc.);
- ALARMS: show a list of currently active alarms;
- SENSORS: show a list of resources and their corresponding sensors;

29.4 System

The System menu consists of the following items:

- Device Information
- Language
- Update Firmware
- Save Configuration
- Load Configuration
- Brightness
- Reset

Each of these functions is described in the following sub-sections.

29.4.1 Device Information

The Device Information view provides detailed information about the Guardian Management Gateway device. Specifically, the following items are displayed:

- Device name
- Model name/number
- Serial number
- Hardware version
- Firmware version
- Capabilities (*0* for Guardian Management Gateway devices).

The LCD UI Capabilities are defined at manufacturing time and cannot be changed.

29.4.2 Language

The Language view shows the list of supported languages that can be selected for the front panel interface. The currently selected language is marked with a tick symbol. To change the current language, tap on the corresponding item.
Note that this setting affects the system language, and it is saved persistently in the device configuration.

29.4.3 Update Firmware

The Update Firmware function allows the user to upgrade the currently running Guardian Management Gateway firmware by loading a firmware image from a USB Flash drive. To upgrade the Guardian Management Gateway firmware, follow the steps described below:

1. Download the latest Guardian Management Gateway firmware image from the nVent web site and copy it to a USB Flash drive
2. Insert the USB Flash drive into the USB 2 port on the Guardian Management Gateway device
3. Select the “SYSTEM”, “Update F/W” menu item
4. In the file dialog, select the firmware image file
5. A progress dialog will appear, showing the firmware upgrade progress
6. When the firmware upgrade is complete, the device will reboot automatically.

29.4.4 **Save Configuration**

The Save Configuration function is used to save the current Guardian Management Gateway configuration to a USB Flash drive inserted into the USB 2 port. The configuration is saved to a file with the following name: “config_saved_N.cfg.tgz”, where “N” is the instance number. The instance number is incremented every time a configuration is saved (to avoid file name conflicts). If the configuration has been saved successfully, the corresponding file name is displayed on the screen. After that, the USB Flash drive can be removed, and the saved configuration can be loaded into this or any other compatible Guardian Management Gateway device at a later time.

29.4.5 **Load Configuration**

The Load Configuration function is used to load a previously saved Guardian Management Gateway configuration from a USB Flash drive inserted into the USB 2 port. When this function is activated, a file selection dialog appears on the screen. To load a firmware configuration, select the corresponding file and wait for the configuration loading process to complete.

29.4.6 **Brightness**

The Brightness dialog is used to change the brightness of the front panel display. The display brightness is controlled by moving the slider left and right to decrease and increase the brightness, respectively:
The selected brightness setting is stored persistently.

29.4.7 **Reset**

The Reset dialog is used to reboot the device.

29.5 **Network**

The Network menu is used to display network configuration information, specifically:
For each configured network interface:

- IPv4 address
  - MAC address
  - Gateway address
  - IPv6 addresses
- IPv4 DNS server addresses
- IPv6 DNS server addresses
- Host name

29.6 **Alarms**

The Alarms menu is used to obtain the list of currently active alarms. For each alarm, the following information is displayed:

- Resource ID / Sensor number
- Sensor name
- Event type (threshold crossing, etc.)

For threshold crossing events, the following encoding is used:

- UCR: upper critical threshold crossing
- UMJ: upper major threshold crossing
- UMN: upper minor threshold crossing

- LCR: lower critical threshold crossing
- LMJ: lower major threshold crossing
- LMN: lower minor threshold crossing

The severity of each alarm can be easily determined by the background color of the corresponding entry, specifically:

- White: OK (lowest severity)
- Green: informational
- Yellow: minor alarm
- Orange: major alarm
- Red: critical alarm
- Gray: unknown severity

An alarm can be acknowledged by tapping on the corresponding entry and confirming the acknowledge operation.
***NOTE**: the alarm acknowledgment function is disabled in the default configuration. It can be enabled via the Web interface in the "Global Settings" dialog, which is invoked with the menu command "Device Settings" -> "Settings".*

29.7 Sensors

The Sensors menu is used to browse the device sensors (internal and external) and obtain information such as sensor readings, thresholds, etc. When selected, a list of resources is first shown on the screen. Tapping on a resource brings up a list of sensors associated with the selected resource. A color indicator to the left of each sensor item reflects the current state of the sensor, specifically:

- Green: sensor reading is within limits
- Yellow: sensor reading is outside the range specified by the minor threshold(s)
- Orange: sensor reading is outside the range specified by the major threshold(s)
- Red: sensor reading is outside the range specified by the critical threshold(s)

Tapping on a sensor item brings up a dialog showing detailed information about the selected sensor. This information is divided into three tabs with the corresponding selection buttons displayed at the top of the viewing area:

- Info: display basic sensor information (current reading, sensor name, sensor number)
- Thr: display threshold values defined for this sensor
- Hyst: display hysteresis values defined for this sensor

30 Technical Data

TECHNICAL DATA	
Height/Width/Depth	1 U / 250 mm / 1 U
Weight	270 g
Ambient Temperature	5 - 60 °C
Humidity	5 - 90% RH, non condensing
Case Material	Aluminum, powder coated
Power Supply	12 VDC, 20W
Emissions	EN 61000-6-3 including EN 55032 level B, FCC Part 15 pending
Immunity	EN 61000-6-2 (industrial environment)
Safety	EN 62368-1, UL 62368-1 pending

31 Revision history

31.1 Release 63998-20557

The differences in the functionality of the USB-A ports are described.
The dialog “Global Settings”, which is invoked via the Device Settings - > Settings menu item, is modified.
The dialog “Modbus Parameters”, which is invoked via the Maintenance -> Modbus -> Configure Modbus Parameters menu item, is modified. The parity of the Modbus interface for SHX30 is fixed.

31.2 Release 63998-20558

The *user* account has been disabled in the default configuration due to security considerations.
The *admin*, *user* and *guest* user accounts now request password change at the first successful login.
Privilege protection is added for evaluation of expressions, discovery of Modbus devices, IoT configuration, reachability.
Aliases for threshold names are introduced.
The `PID(),min(),max()` functions are described.
Support for AWS Greengrass IoT is introduced.
In the SNMP interface, the new variable *ctrlCachedState* has been implemented; it exposes the cached state of the corresponding control.
In the Web interface, Visual Expression Builder features can be invoked for event rules and periodic actions via the CTRL+Space key combination.

32 References

1. Service Availability™ Forum Hardware Platform Interface Specification, Specification SAI-HPI-B.03.02, August 4, 2009.
2. IPMI – Platform Management FRU Information Storage Definition v1.0, Document Revision 1.1, September 27, 1999.