nE2 Link for Niagara

User Manual

Quick Start-up

iSMA CONTROLLI S.p.A. - Via Carlo Levi 52, 16010 Sant'Olcese (GE) - Italy | support@ismacontrolli.com

Table of Contents

1 Introduction

The nE2 Link for Niagara is a comprehensive solution designed to enhance the Niagara Framework by enabling seamless commissioning, programming, and control of nano EDGE ENGINE devices.

The module is addressed to current and future Niagara Framework users who want to comprehensively manage, program, and integrate nano EDGE ENGINE devices directly into the Niagara 4 environment.

Using the built-in functionalities in Niagara, nE2 Link extends its capabilities to include nano EDGE ENGINE functions natively. The extension greatly extends the reach and usability of nano EDGE ENGINE devices, making it easier for users to integrate and manage building systems directly into Niagara 4, without the need for third-party tools.

The purpose of this document is to describe how to correctly install and start using nano EDGE ENGINE devices in a native Niagara environment.

> Note
> The nE2 Link is designed to enable the programming of controllers with the nano EDGE ENGINE embedded and provides all necessary functionalities for programming and commissioning the controller. However, it is important to note that the Niagara Framework includes a wide range of features, some of which are not supported by the nano EDGE ENGINE. As a result, certain functionalities of the Niagara Workbench may not be fully compatible with the nano EDGE ENGINE.
> To assist users in addressing any potential issues, iSMA CONTROLLI has developed detailed troubleshooting materials, which are available at the following link: nE2 Link.
> For further assistance or to provide suggestions for improvement, please contact iSMA CONTROLLI Support at support@ismacontrolli.com.

1.1 Revision History

Date	Manual Rev.	nE2 Link Module Ver.	Description
17 Mar 2026	1.9.0	1.9	New functionalities: • Tagging service • Haystack service • Web service • DeviceFolder component (BACnet Device Discover Manager, Modbus Device Manager) • PointFolder component (BACnet Point Manager, Modbus Point Manager, LocalIO Manager)
2 Dec 2025	1.8.0	1.8	New functionalities: • Trends service • Trend extension to Data Points
28 Aug 2025	1.7.0 (versioning merged with the module version)	1.7 (versioning merged with the supported nano EDGE ENGINE version)	New functionalities: • BACnet discover: ○ BACnet Device discover ○ BACnet Object discover • Ne2LinkService

Date	Manual Rev.	nE2 Link Module Ver.	Description
			• Forced password change at first connection • Auto-disconnect function • USB connection for device management and programming • Emergency mode Improvements: • Libraries for programming available in the Palette window
18 Jun 2025	1.2.0	1.1	New functionalities: • Configuration managers: ◦ Application Manager ◦ Data Point Manager ◦ Local IO Manager ◦ BACnet Network Manager ◦ BACnet Device Manager ◦ BACnet Point Manager ◦ Modbus Network Manager ◦ Modbus Device Manager ◦ Modbus Point Manager • Configuration Data Service • Saving logic to Niagara palette • Logs Improvements: • Software Manager update - information about not loaded libraries • Adding extensions directly from the context menu • Reordering option • Cut and paste options
30 Jan 2025	1.1.0	1.0	General availability edition
18 Jul 2024	1.0.0	1.0	First edition

2 Installation

2.1 Supported Niagara Versions

nE2 Link is dedicated to Niagara 4 and supports Niagara 4.11 and higher.

2.2 Required Modules

In order to work properly, the nE2 Link requires the following modules:

- nE2Link-rt.jar,
- nE2Link-ux.jar,
- nE2Link-wb.jar,
- docNe2Link-doc.jar.

Contact the authorized iSMA CONTROLLI distributor to get the latest modules.

2.3 Installation

2.3.1 Niagara Workbench (Recommended)

> nE2 Link is dedicated to work with Niagara Workbench.

nE2 Link can work directly on local Workbench stations. In order to use it correctly, follow the steps below:

1. Close the Workbench.
2. Copy the `nE2link-rt.jar`, `nE2link-ux.jar`, `nE2link-wb.jar`, and `docNe2Link-doc.jar` files to the Niagara modules directory. Default Path: `C:\Niagara\Niagara-4.x.x\modules`.
3. Reopen the Workbench.
4. Connect to the local Platform using Workbench and restart the station.

2.3.2 Niagara Controller

> In exceptional situations, it is possible to use nE2 Link directly on a Niagara controller with a limitation of opening maximum 2 connections at a time using nE2deviceExt in Niagara controllers. **For effective and seamless work, it is recommended to use nE2 Link with Niagara Workbench.**

It is possible to use the nE2 Link directly on a Niagara controller, such as MAC36 or JACE.

> **Note:** Before proceeding, make sure the module is correctly installed in the local Workbench.

1. In Workbench, connect to the controller's Platform.
2. Open the Software Manager.
3. Locate the `nE2link-rt` and `nE2link-ux` modules in the list and select them.
4. Click the `Install` button at the bottom of the window.
5. Click the `Commit` button at the bottom of the window.

6. The modules will be installed on the station, and the list will reload when the process is complete.
7. Verify the `nE2link` modules are marked as “Up to Date”.
8. Restart the Niagara station.

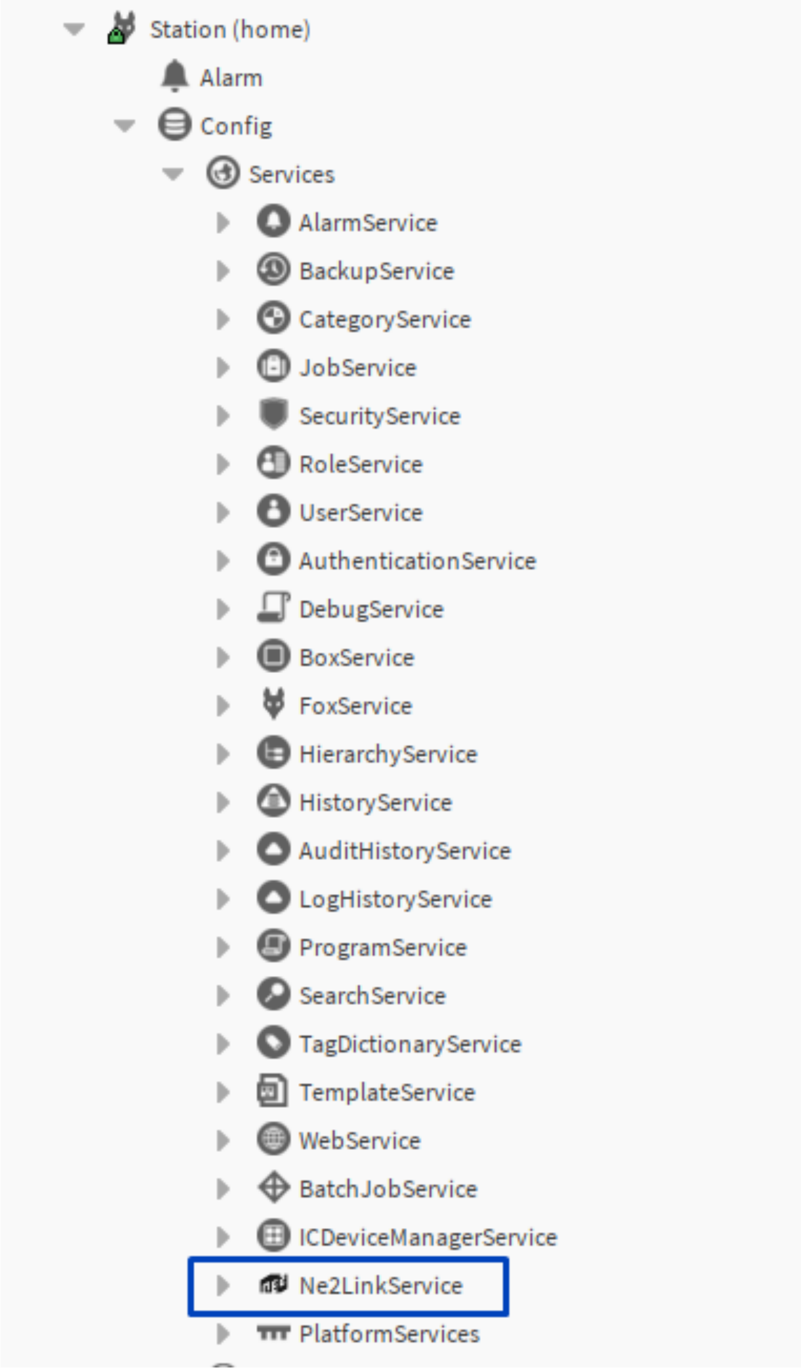
By following these steps, the nE2 Link modules will be successfully installed, enabling to leverage the full capabilities of the nano EDGE ENGINE devices within the Niagara 4 environment.

3 Ne2LinkService

The Ne2LinkService service allows for managing existing nE2deviceExt added to the supported networks. Device extensions are automatically discovered and visible in the service. The service allows to establish a connection with a USB device directly through the service.

3.1 Installation

There are two ways the Ne2LinkService is added to the station Services, automatically and manually.



- Station (home)
  - Alarm
  - Config
    - Services
      - AlarmService
      - BackupService
      - CategoryService
      - JobService
      - SecurityService
      - RoleService
      - UserService
      - AuthenticationService
      - DebugService
      - BoxService
      - FoxService
      - HierarchyService
      - HistoryService
      - AuditHistoryService
      - LogHistoryService
      - ProgramService
      - SearchService
      - TagDictionaryService
      - TemplateService
      - WebService
      - BatchJobService
      - ICDeviceManagerService
      - Ne2LinkService
      - PlatformServices

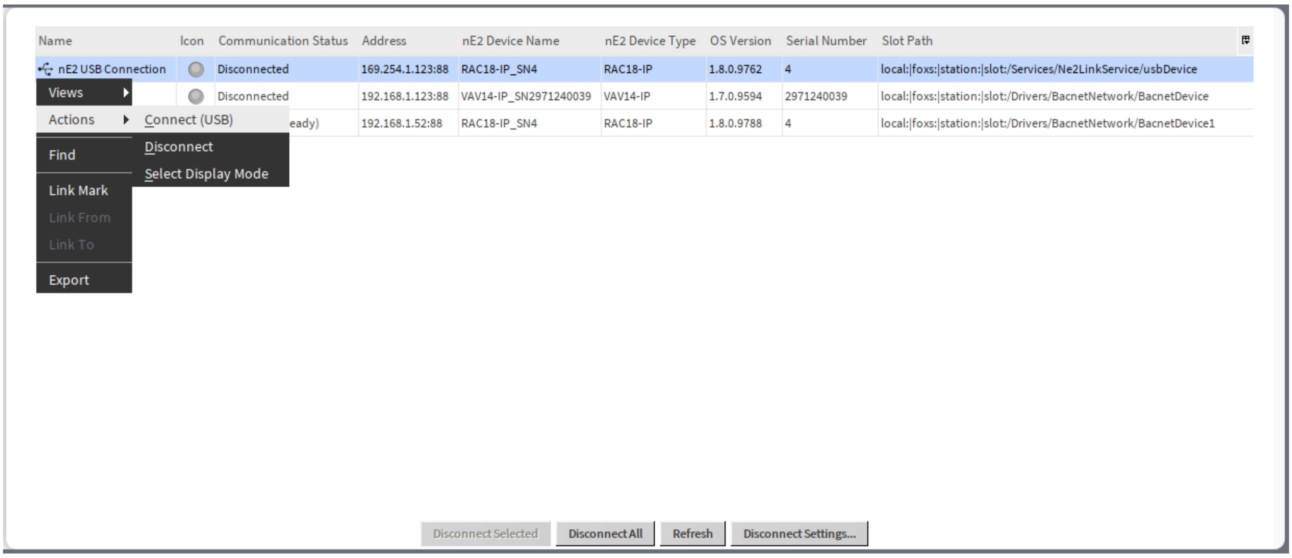
Figure 1. Ne2LinkService

It is also possible to add it manually to the Services in the station, dragging it from the nE2 Link palette.

3.2 USB Connection

From the nano EDGE ENGINE V1.7.0, it is possible to connect and program devices using the USB cable. The Ne2LinkService is a dedicated way to manage the USB connection to the device.

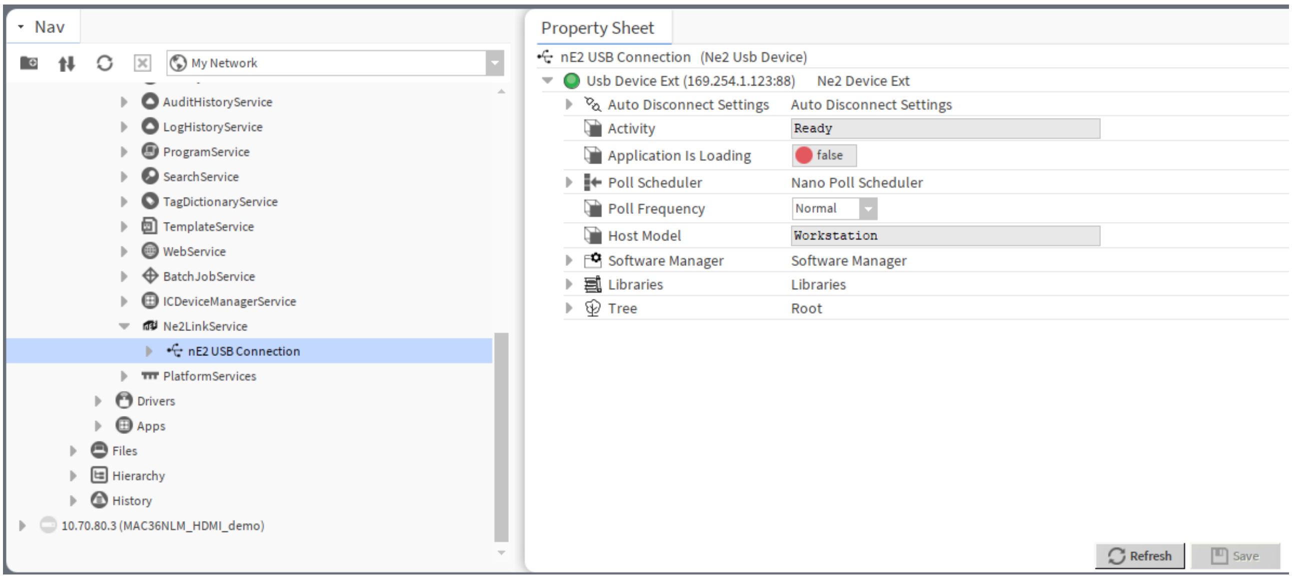
In the service's view, there is a permanent position 'nE2 USB Connection', which allows to connect to any nano EDGE ENGINE device connected to the PC using the USB cable (even if it has not been previously added as the BACnet device extension).



Name	Icon	Communication Status	Address	nE2 Device Name	nE2 Device Type	OS Version	Serial Number	Slot Path
nE2 USB Connection	●	Disconnected	169.254.1.123:88	RAC18-IP_SN4	RAC18-IP	1.8.0.9762	4	local:|foxs:|station:|slot:/Services/Ne2LinkService/usbDevice
	●	Disconnected	192.168.1.123:88	VAV14-IP_SN2971240039	VAV14-IP	1.7.0.9594	2971240039	local:|foxs:|station:|slot:/Drivers/BacnetNetwork/BacnetDevice
		eady)	192.168.1.52:88	RAC18-IP_SN4	RAC18-IP	1.8.0.9788	4	local:|foxs:|station:|slot:/Drivers/BacnetNetwork/BacnetDevice1

Figure 2. Ne2LinkService - USB connection

The USB connection is also added (automatically) as a component under the Ne2LinkService.



nE2 USB Connection (Ne2 Usb Device)

Usb Device Ext (169.254.1.123:88)	Ne2 Device Ext
Auto Disconnect Settings	Auto Disconnect Settings
Activity	Ready
Application Is Loading	● false
Poll Scheduler	Nano Poll Scheduler
Poll Frequency	Normal
Host Model	Workstation
Software Manager	Software Manager
Libraries	Libraries
Tree	Root

Figure 3. nE2 USB Connection component

The nE2 USB connection component gathers fundamental features related to the USB connection with the nano EDGE ENGINE controller:

- Auto Disconnect Settings: allows to set the time after which the device disconnects automatically, Note: Auto-disconnect parameters can be set individually for the device connected with USB but please note that setting them directly in the Ne2LinkService main view (using the Disconnect Settings button) will overwrite the individual settings.

- Activity: shows the status of the connected device,
- Application is Loading: shows if the application loading process is ongoing; if false, it means the process has been finished and the application is loaded,
- Poll Scheduler: allows to set polling speed parameters and shows polling statistics,
- Host Model: shows the host model of the connected device,
- Software Manager: shows the operating properties of the connected device:
    - iFnet port,
    - IP address,
    - last connected,
    - MAC address,
    - maximum number of Data Points,
    - device name,
    - model,
    - OS version,
    - ping,
    - serial number,
    - information about not loaded libraries (if any);
- Libraries: shows libraries installed on the connected device and their versions,
- Tree: shows the workspace tree of the connected device.

3.2.1 Establishing USB Connection

USB Cable

For the USB connection, please use either the USB-C/USB-C cable or the USB-C/USB-A type. Please note that nano EDGE ENGINE controllers, which can be connected in the Ne2LinkService, are equipped with the USB-C port on their end.

The cable has to support both power and communication to establish a proper connection with the nano EDGE ENGINE controller. Once connected, the controller should power up directly, if it is not already powered with 24 V AC/DC source.

To establish the USB connection, go either to the Ne2Service main view (double click the service) or to the nE2 USB Connection component in the tree. Open the context menu and select the Connect (USB) option.

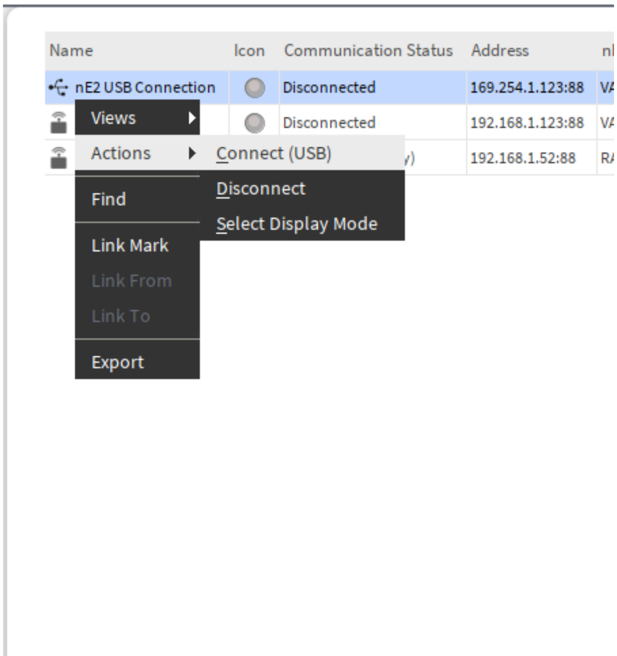


Figure 4. Connect (USB) option in the Ne2LinkService main view

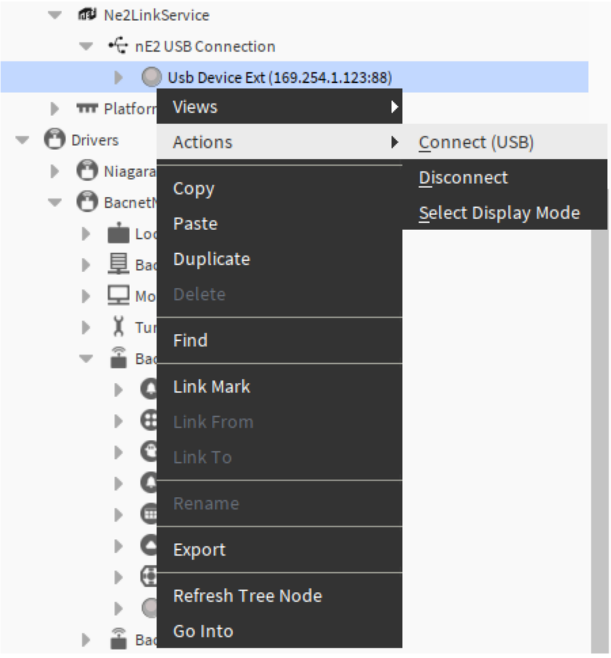
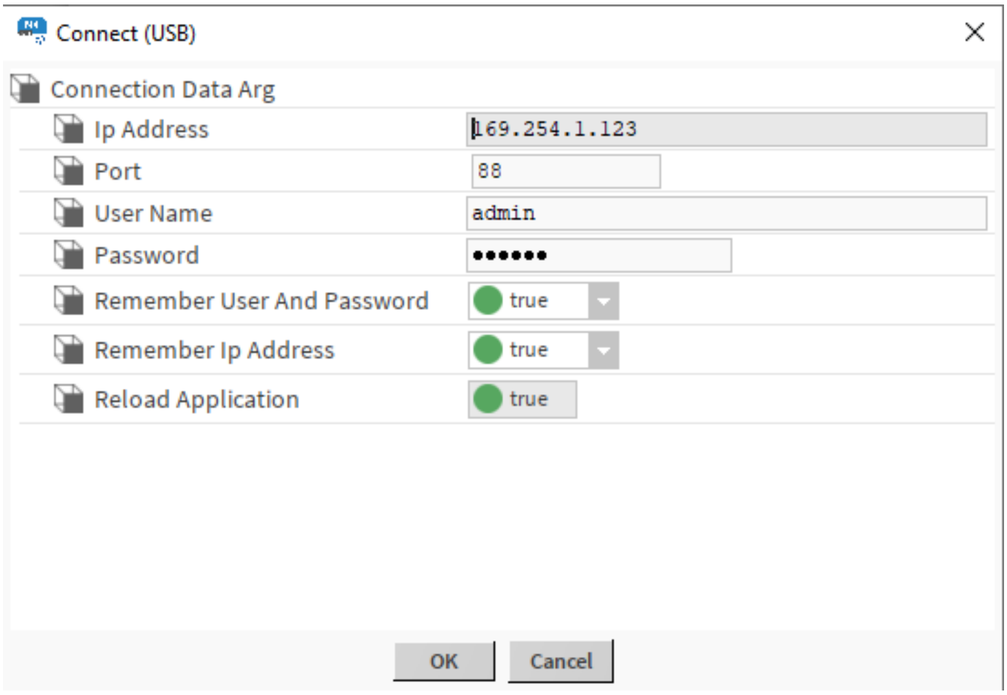


Figure 5. Connect (USB) option in the nE2 USB Connection component

The Connect (USB) action brings up a pop-up window:


Connect (USB)

Connection Data Arg

Ip Address	169.254.1.123
Port	88
User Name	admin
Password	••••••
Remember User And Password	● true
Remember Ip Address	● true
Reload Application	● true

Figure 6. USB connection dialog window

Here, enter the controller's user name and password. The pop-up shows also the following parameters:

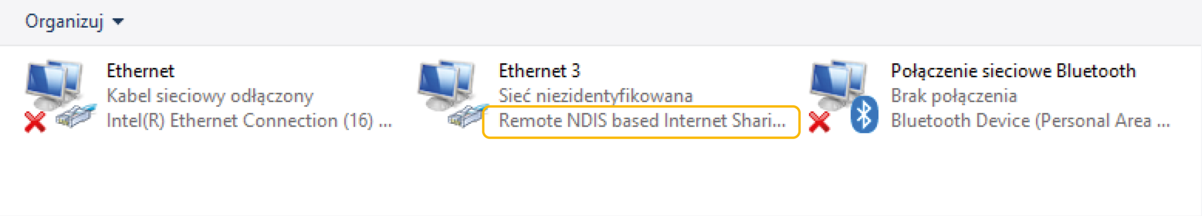
- IP Address: shows the fixed IP address used for the USB connection (169.254.1.123),
- Port: shows the iFnet port number and allows to change it, if different,
- User Name: allows to enter the controller's user name,
- Password: allows to enter the controller's password,
- Remember User and Password: allows to set the remembering credentials parameter (if true, the credentials will be saved for the next logging in),

- Remember Ip Address: allows to set the remembering IP address parameter (the USB connection always uses the fixed IP address, which is remembered by default),
- Reload Application: shows if the reload application action will be executed upon connection (true by default).

Note

If the connection cannot be established, it is advisable to check if the PC recognized the connection as the `Remote NDIS based Internet Sharing Device` .

To check the connection, go to the Control panel>Network Connections:


Organizuj ▾
Ethernet
Kabel sieciowy odłączony
Intel(R) Ethernet Connection (16) ...
Ethernet 3
Sieć niezidentyfikowana
Remote NDIS based Internet Shari...
Połączenie sieciowe Bluetooth
Brak połączenia
Bluetooth Device (Personal Area ...

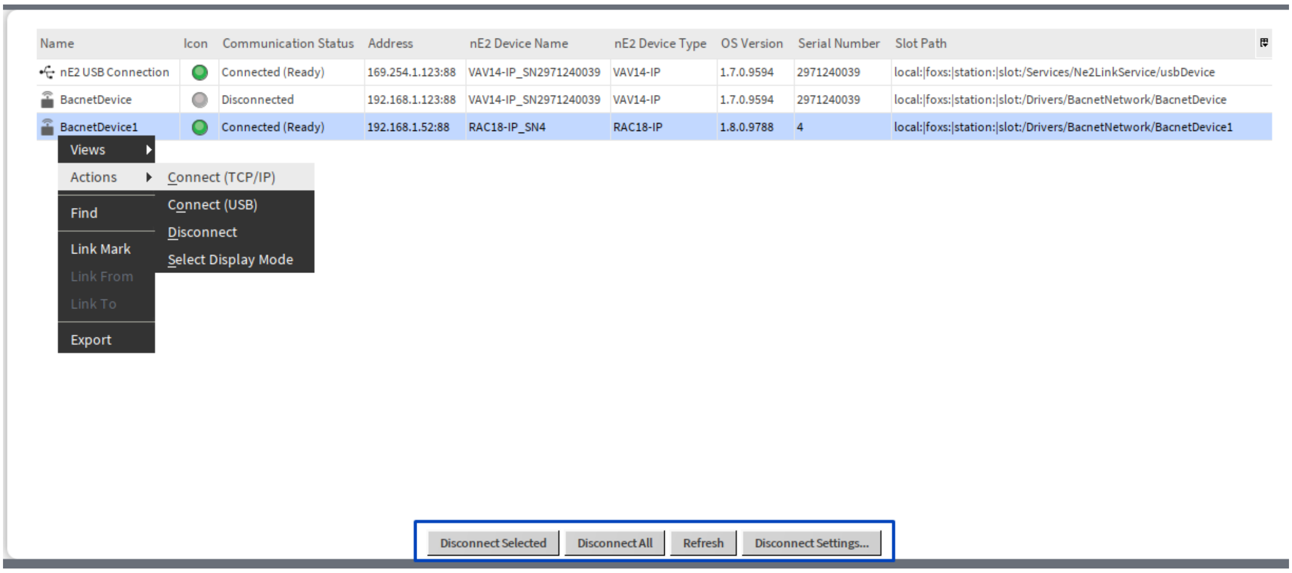
3.3 Managing Added Devices

Managing nano EDGE ENGINE devices added as BACnet device extensions or connected with USB is most convenient from the main view of the Ne2LinkService. The view lists all nEE devices added in the station making it easy to locate them and manage TCP/IP and USB connections.

The Ne2LinkService dedicated view is opened on double-click on the service in the tree. The view shows the following data about the devices:

- name,
- status icon,
- communication status,
- IP address,
- nE2 device name (with a serial number),
- nE2 device type,
- OS version on the device,
- serial number,
- slot path.

Each device can be managed either by action buttons on the bottom of the view or individually from the context menu.



Name	Icon	Communication Status	Address	nE2 Device Name	nE2 Device Type	OS Version	Serial Number	Slot Path
nE2 USB Connection	●	Connected (Ready)	169.254.1.123:88	VAV14-IP_SN2971240039	VAV14-IP	1.7.0.9594	2971240039	local:|foxs:|station:|slot:/Services/Ne2LinkService/usbDevice
BacnetDevice	●	Disconnected	192.168.1.123:88	VAV14-IP_SN2971240039	VAV14-IP	1.7.0.9594	2971240039	local:|foxs:|station:|slot:/Drivers/BacnetNetwork/BacnetDevice
BacnetDevice1	●	Connected (Ready)	192.168.1.52:88	RAC18-IP_SN4	RAC18-IP	1.8.0.9788	4	local:|foxs:|station:|slot:/Drivers/BacnetNetwork/BacnetDevice1

Figure 7. Ne2LinkService

From the context menu, it is possible to connect or disconnect the device. Also, there are options to copy/paste/duplicate/delete/rename the selected device.

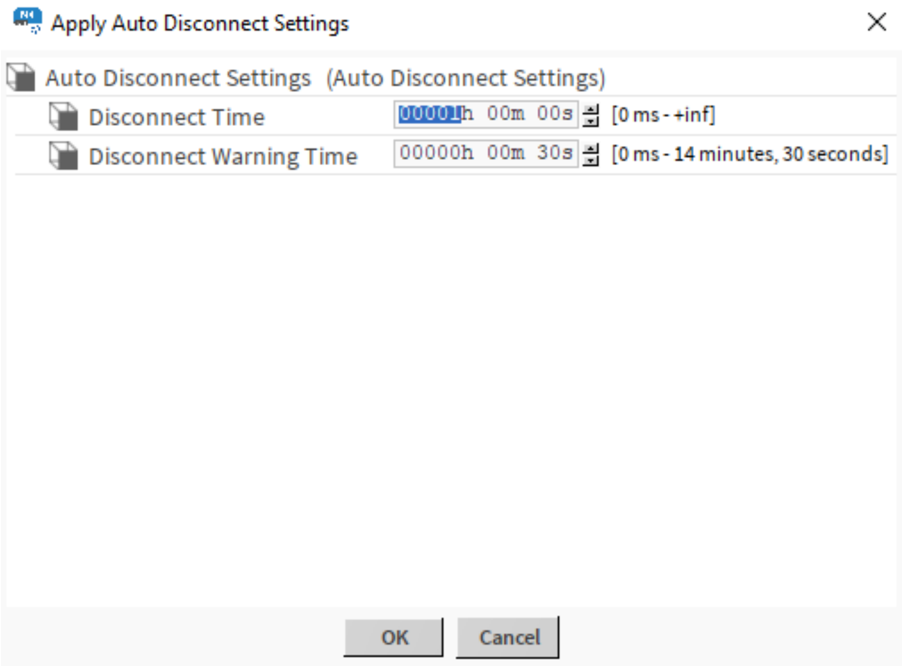
The Ne2LinkService has four action buttons which allow to perform the following:

- Disconnect Selected: disconnects selected device(s),
- Disconnect All: disconnects all connected devices,
- Refresh: refreshes the view (devices added/removed while the view has been displayed will appear/disappear from the list),
- Disconnect Settings: shows setting for an automatic disconnect mechanism (applies to all connected devices).

For user convenience, it is recommended to disconnect devices after commissioning or programming. The devices can be disconnected manually or by the auto-disconnect function, which helps unnecessary consumption of the resources, while the service also manages inactive connections (they are not lost after disconnecting).

3.3.1 Auto-disconnect

Auto-disconnect function disconnects the device automatically after the set time runs off. Auto-disconnect settings can be set in the pop-up dialog window invoked by the Disconnect Settings button:



Apply Auto Disconnect Settings

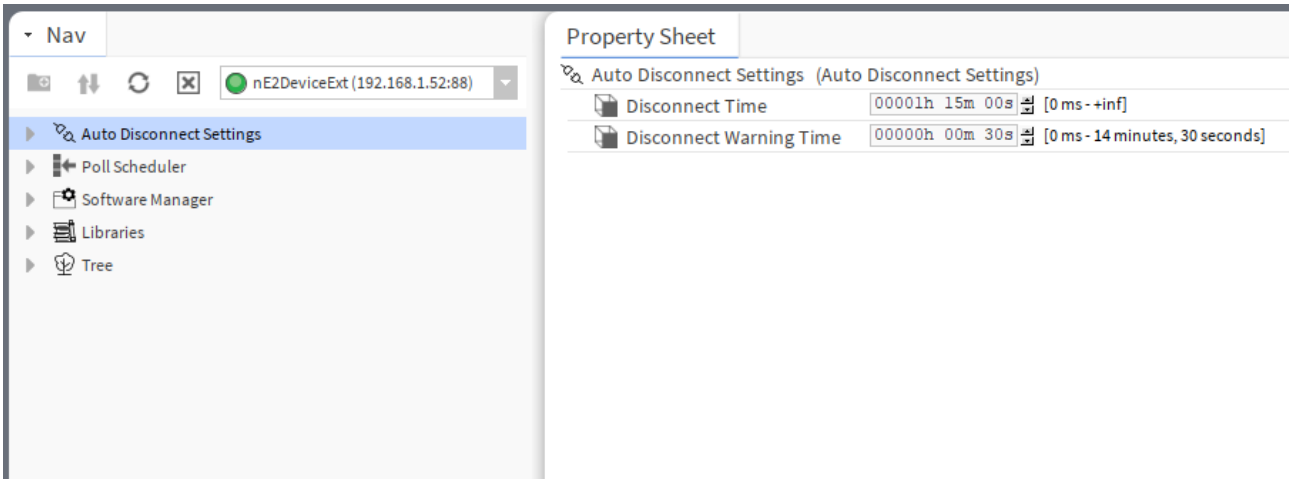
Auto Disconnect Settings (Auto Disconnect Settings)

Disconnect Time 00001h 00m 00s [0 ms - +inf]

Disconnect Warning Time 00000h 00m 30s [0 ms - 14 minutes, 30 seconds]

Figure 8. Auto-disconnect setting dialog window

The same settings are available under double-click on the Auto Disconnect Settings option in the `nE2DeviceExt` tree:



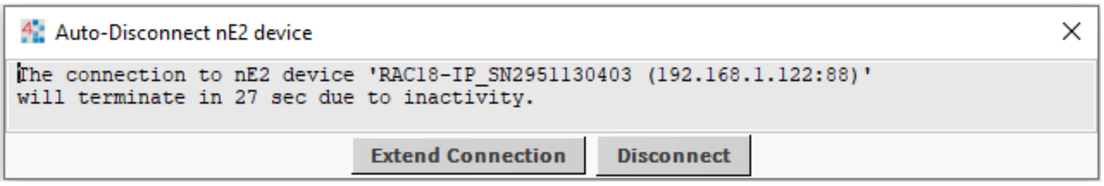
Auto Disconnect Settings (Auto Disconnect Settings)

Disconnect Time 00001h 15m 00s [0 ms - +inf]

Disconnect Warning Time 00000h 00m 30s [0 ms - 14 minutes, 30 seconds]

Figure 9. Auto-disconnect settings

Once the disconnect time passes, the auto log off notice is displayed:



Auto-Disconnect nE2 device

```
The connection to nE2 device 'RAC18-IP_SN2951130403 (192.168.1.122:88)'
will terminate in 27 sec due to inactivity.
```

Figure 10. Auto log off notice

It is also possible to set the auto-disconnect warning displayed before the devices get disconnect in time set in the Disconnect Warning Time slot. To disable the auto-disconnect function, set the disconnect time to 0 s.

4 Configuration

4.1 Adding nE2DeviceExt in Niagara

(a) In Workbench, navigate to the `nE2Link` module in the Palette window, search and open the `nE2Link` module.

The module palette contains the Programming and Services folders.

The `nE2DeviceExt` is a network device extension located in the Programming folder. The nE2DeviceExt functions as a device extension inside Niagara networks, the BACnetNetwork or ModbusTcpNetwork, it must be dropped under the proper network device.

In the Services folder, the nE2LinkService is included. The service allows to manage the existing connections to nE2DeviceExt devices. It has to be located under Services in the station.

(b) Locate the `nE2DeviceExt` extension within the `Programming` folder.

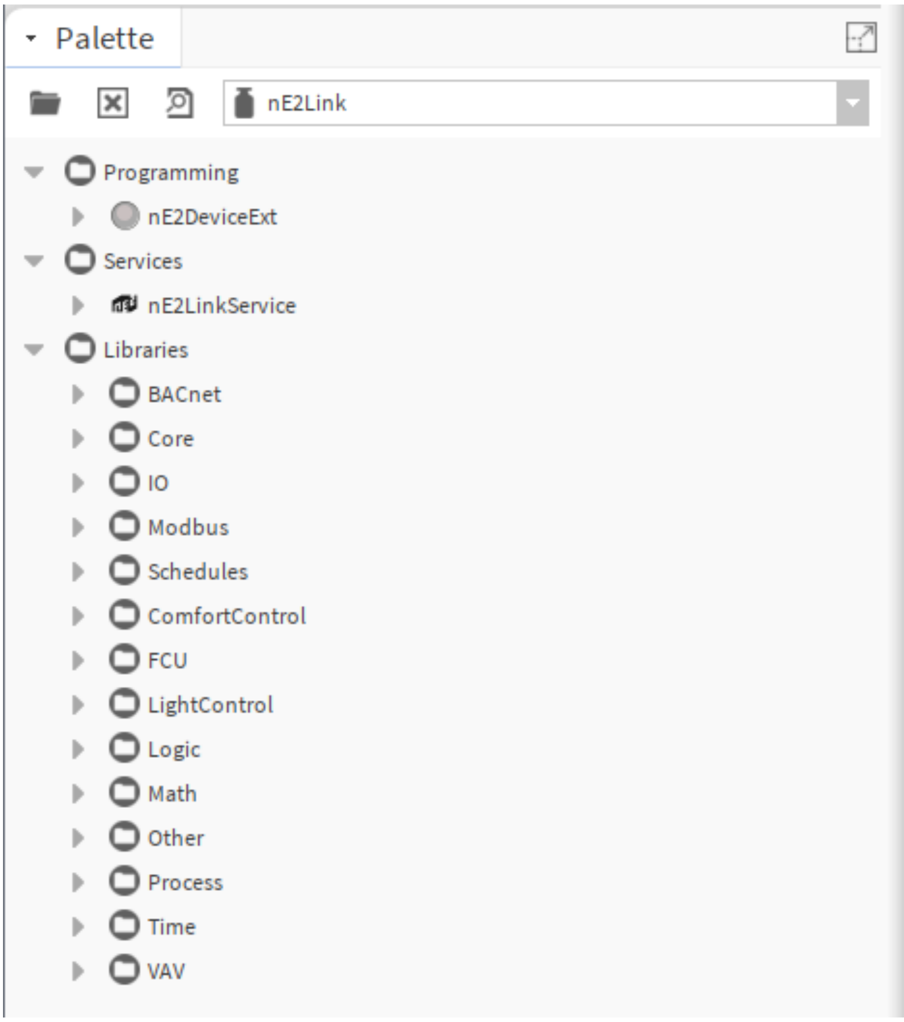


Figure 11. nE2Device extension in the nE2Link module

The nE2DeviceExt extension consist of:

- **Auto Disconnect Settings:** allows to set the autodisconnect parameters;

- Poll Scheduler: manages communication between the Niagara Framework and the nE2 controller;
- Software Manager: allows for managing libraries on the controller.
- Libraries: by default, the folder is empty and requires a real-time connection to upload the libraries available on the device. Once connected, the device's library will be populated with data from the device and load all the libraries available on the nano EDGE ENGINE device.

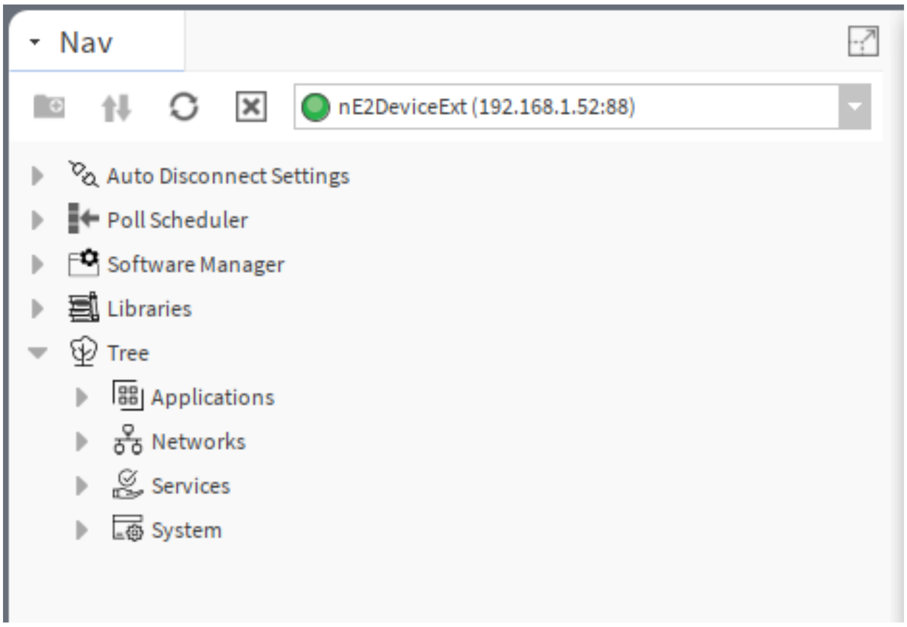


Figure 12. nE2DeviceExt contents

(c) Add nE2DeviceExt to BACnet Device:

- Make sure that the BACnet network is set up in the station.

> Note: In nano EDGE ENGINE devices, such as the VAV14-IP or RAC18-IP, the native BACnet support guarantees that it can be discovered on the BACnet IP network out of the box.

- Drag the `nE2DeviceExt` extension from the `nE2Link` palette and drop it onto the BACnet device in the Niagara station.

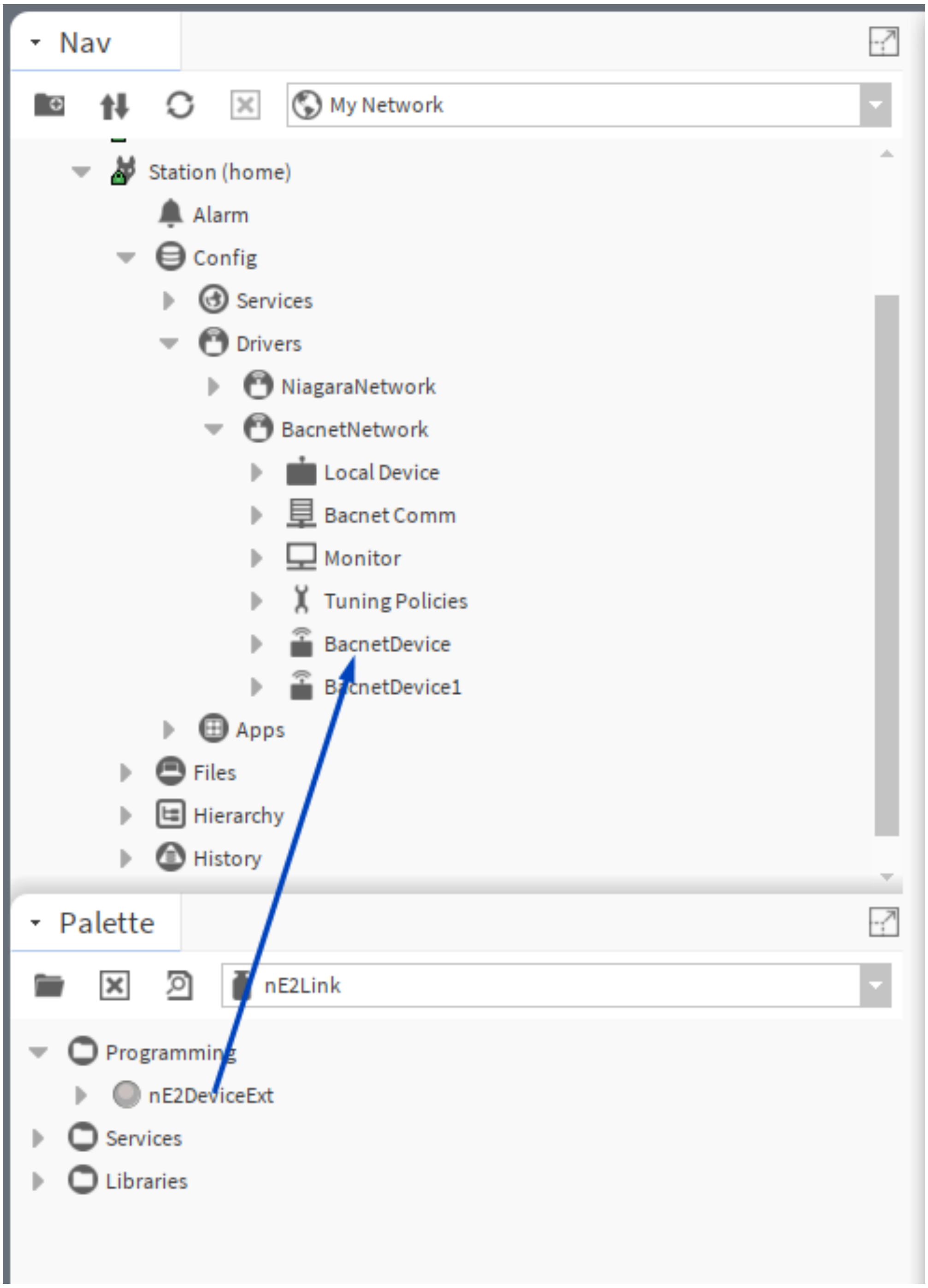


Figure 13. Adding nE2Device Ext to the BACnet device

> Ne2LinkService
> At this step, the Ne2LinkService is automatically added to the station Services.

4.2 Connection and Logging in to the nano EDGE ENGINE Device

4.2.1 Establishing a Connection

(a) Once the extension is added to the device, right-click on the `nE2DeviceExt`, go to Actions → Connect. The following options are available here:

- Connect (TCP/IP): uses a standard Ethernet cable connection,

- Connect (USB): uses a USB cable connection,
- Disconnect: disconnects the device.

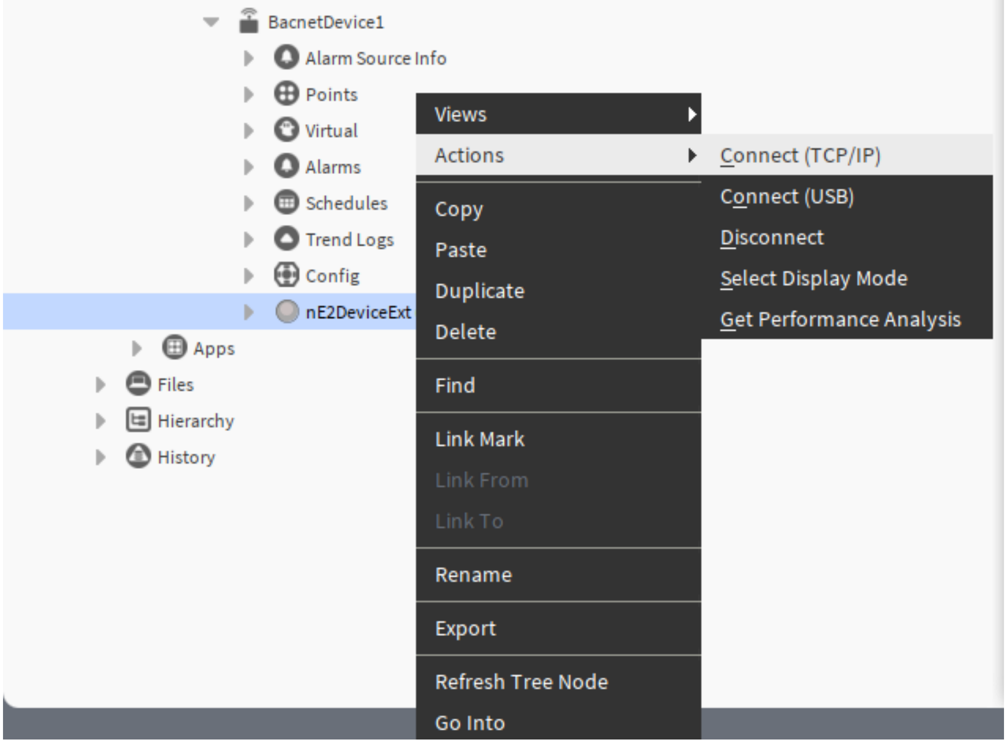


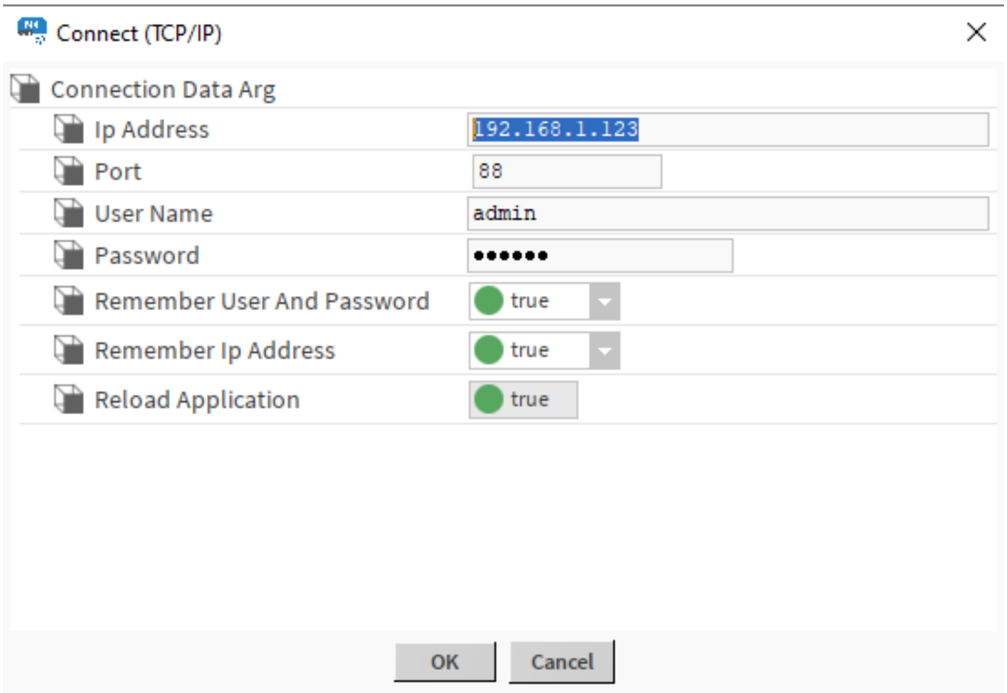
Figure 14. Connect option

(b) A pop-up connection window will open.

For TCP/IP connection:

- IP Address: the nano EDGE ENGINE device address;
- Port: iFnet port (by default, 88);
- User Name: nano EDGE ENGINE username (by default, admin);
- Password: nano EDGE ENGINE user password (by default, admin);
- Remember User And Password: allows to change if the credentials are saved for the future connections;
- Remember Ip Address: allows to change if the IP address is saved for the future connections;
- Reload Application: shows if the application is automatically reloaded upon logging in.

> Note: Password must be changed after the first connection to the device, see the First Connection section below.



Connect (TCP/IP)

Connection Data Arg

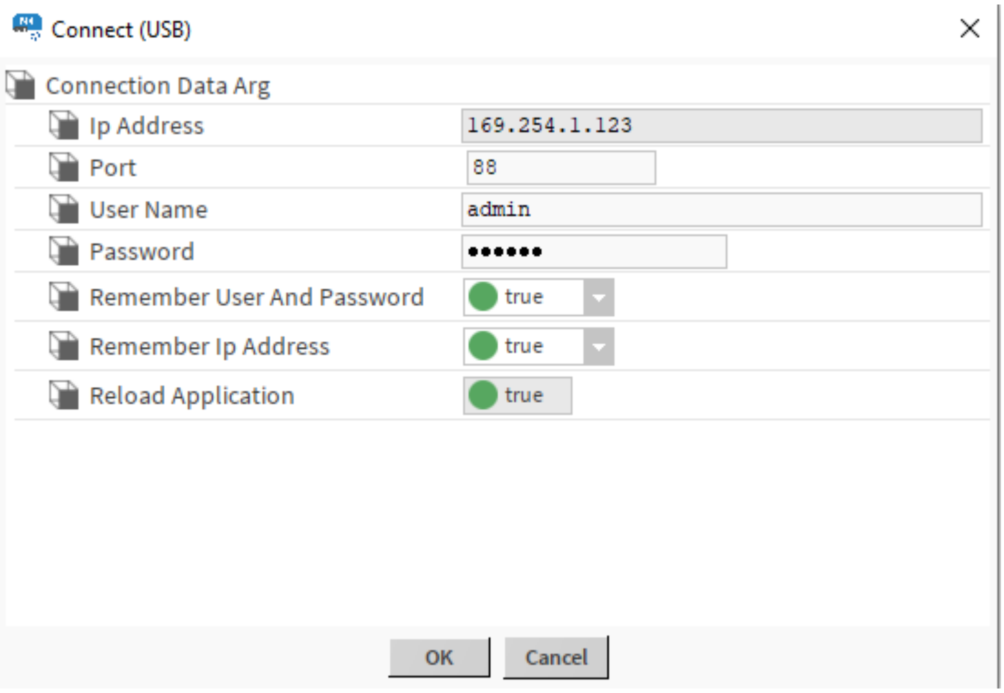
Ip Address	192.168.1.123
Port	88
User Name	admin
Password	••••••
Remember User And Password	● true
Remember Ip Address	● true
Reload Application	● true

Figure 15. TCP/IP connection pop-up

For USB connection:

- IP Address: fixed IP address for the USB connection with the nano EDGE ENGINE device (169.254.1.123);
- Port: iFnet port (by default, 88);
- User Name: nano EDGE ENGINE username (by default, admin);
- Password: nano EDGE ENGINE user password (by default, admin);
- Remember User And Password: allows to change if the credentials are saved for the future connections;
- Remember Ip Address: allows to change if the IP address is saved for the future connections;
- Reload Application: shows if the application is automatically reloaded upon logging in.

Note: Password must be changed after the first connection to the device, see the First Connection section below.



Connect (USB)

Connection Data Arg

Ip Address	169.254.1.123
Port	88
User Name	admin
Password	••••••
Remember User And Password	● true
Remember Ip Address	● true
Reload Application	● true

Figure 16. USB connection pop-up

(c) Click OK to establish connection with the device.

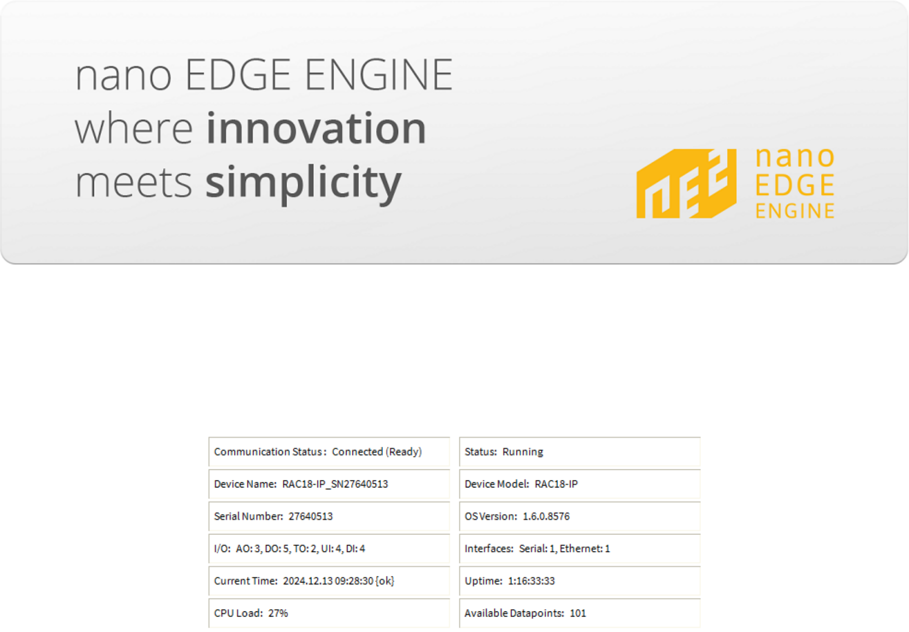
LED Progress Indicator

The `nE2DeviceExt` icon includes a status LED indicator that visually represents the device's connection state to ensure easy monitoring for users:

- gray LED ●: the device is disconnected,
- orange LED ●: the device is connecting,
- yellow LED ●: the device is connected and the application is loading,
- green LED ●: the device is successfully connected and the application has finished loading.

These color-coded LED statuses provide clear feedback to the customer about the current state of the device.

Once the connection with the device has been established (the green LED indicator is displayed), the following start screen is displayed:



Communication Status : Connected (Ready)	Status: Running
Device Name: RAC18-IP_SN27640513	Device Model: RAC18-IP
Serial Number: 27640513	OS Version: 1.6.0.8576
I/O: AO: 3, DO: 5, TO: 2, UI: 4, DI: 4	Interfaces: Serial: 1, Ethernet: 1
Current Time: 2024.12.13 09:28:30 {ok}	Uptime: 1:16:33:33
CPU Load: 27%	Available Datapoints: 101

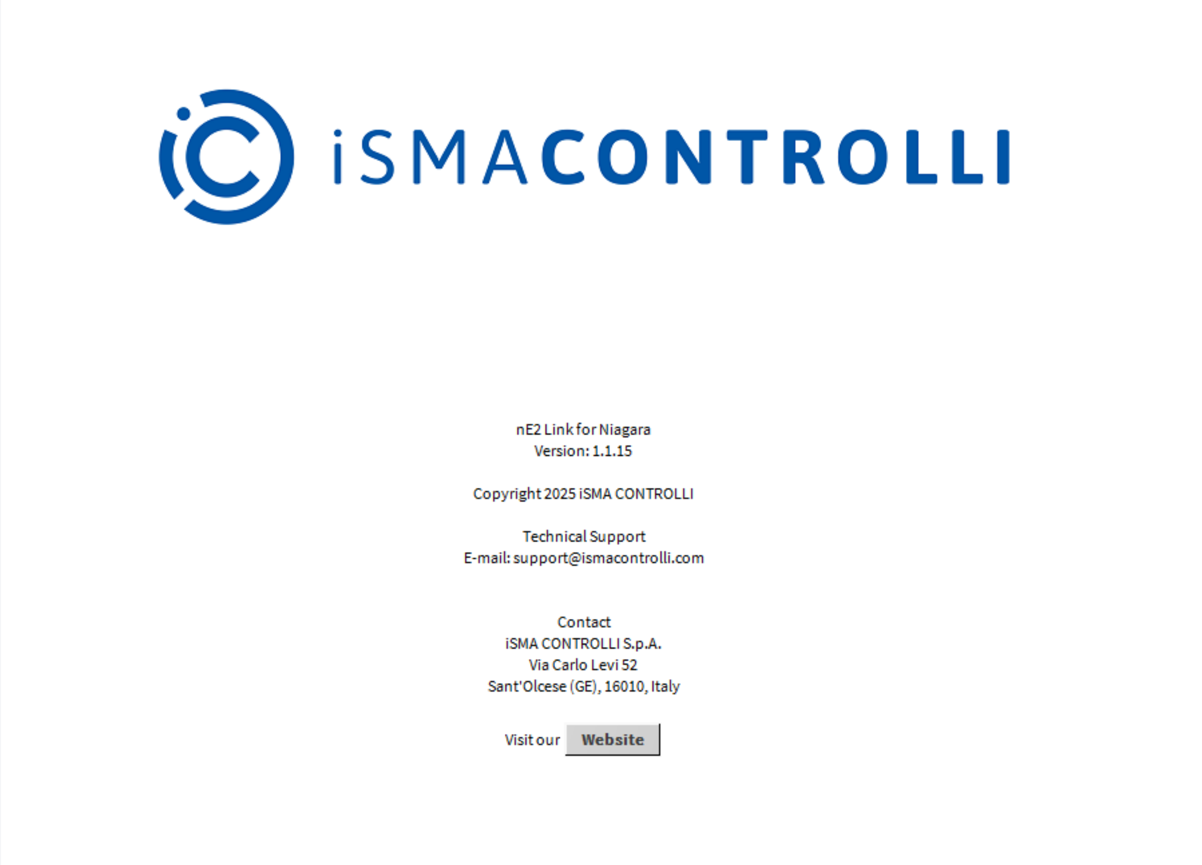
Figure 17. nE2 Link start screen

The start screen shows the following information:

- communication status,
- device status,
- device name,
- device model,
- serial number,
- OS version,
- list of I/Os,
- interfaces,
- current time,
- uptime,
- CPU load,
- available Data Points.

Note

If the connection is established for the first time or the extension gets disconnected, the following home screen is displayed:



nE2 Link for Niagara
Version: 1.1.15

Copyright 2025 iSMA CONTROLLI

Technical Support
E-mail: support@ismacontrolli.com

Contact
iSMA CONTROLLI S.p.A.
Via Carlo Levi 52
Sant'Olcese (GE), 16010, Italy

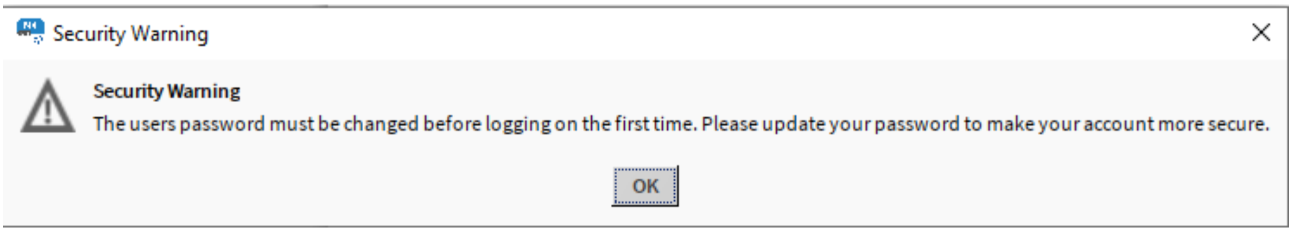
The screen provides information such as:

- version of the module;
- copyrights;
- support;
- contact information.

4.2.2 First Connection

When the connection is established correctly, the `nE2DeviceExt` icon will go from gray to green.

During the first connection to the device using a default password, a message will be displayed requesting to change the password.



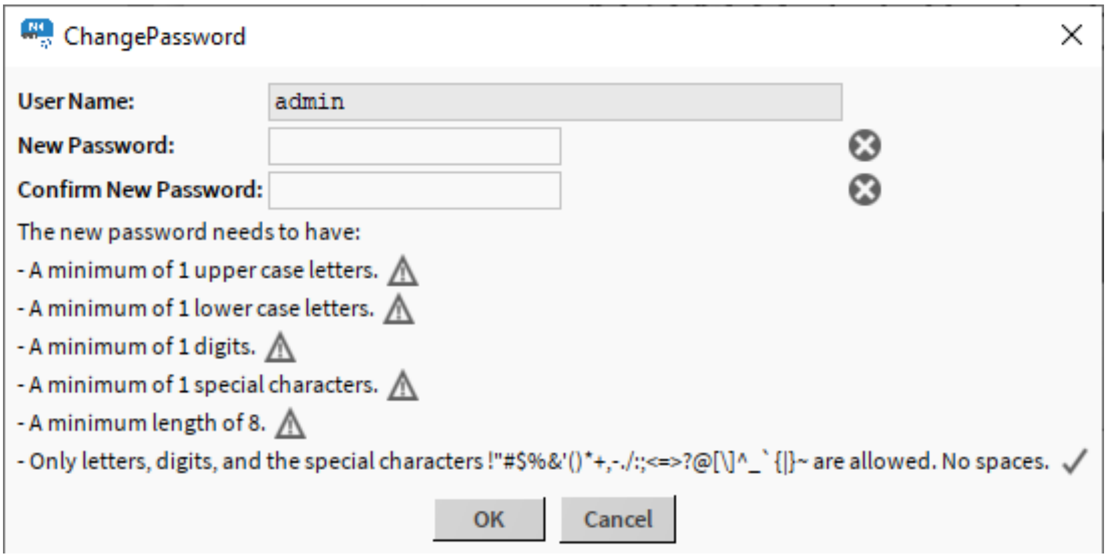
Security Warning

The users password must be changed before logging on the first time. Please update your password to make your account more secure.

Figure 18. Change password prompt

Changing password is obligatory when first connecting to the device!

After confirming the security warning, a change password dialog window appears:



ChangePassword

User Name: admin

New Password: ⊗

Confirm New Password: ⊗

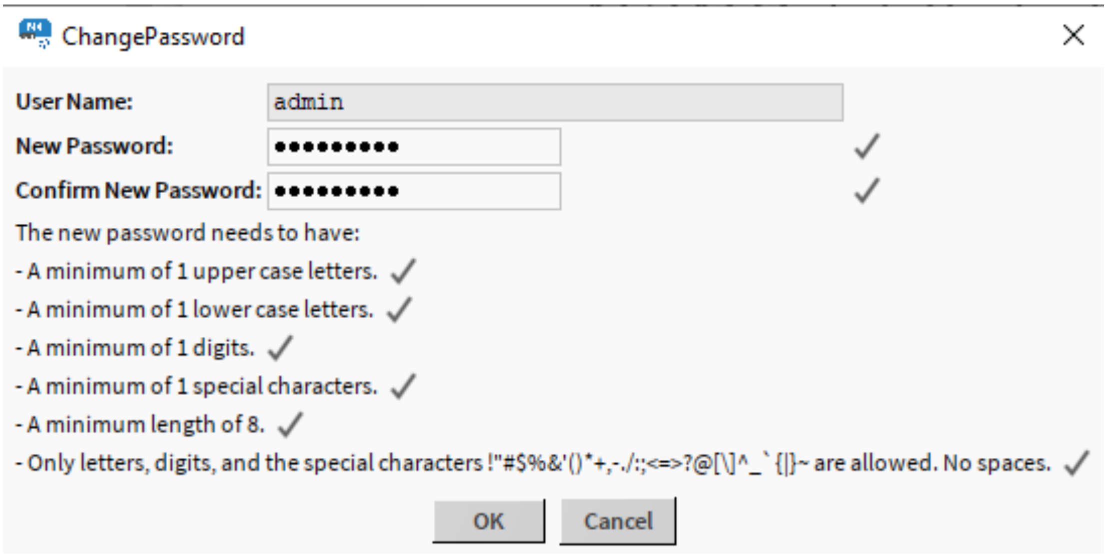
The new password needs to have:

- A minimum of 1 upper case letters. ⚠
- A minimum of 1 lower case letters. ⚠
- A minimum of 1 digits. ⚠
- A minimum of 1 special characters. ⚠
- A minimum length of 8. ⚠
- Only letters, digits, and the special characters !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~ are allowed. No spaces. ✓

Figure 19. ChangePassword dialog window

Enter the new password in the New Password and Confirm New Password fields. Make sure the entered password is identical in both fields. A proper password has to fulfill the following conditions:

- include at least 1 upper case letter,
- include at least 1 lower case letter,
- include at least 1 digit,
- include at least 1 special character,
- be at least 8 characters long,
- no spaces.



ChangePassword

User Name: admin

New Password: ●●●●●●●●● ✓

Confirm New Password: ●●●●●●●●● ✓

The new password needs to have:

- A minimum of 1 upper case letters. ✓
- A minimum of 1 lower case letters. ✓
- A minimum of 1 digits. ✓
- A minimum of 1 special characters. ✓
- A minimum length of 8. ✓
- Only letters, digits, and the special characters !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~ are allowed. No spaces. ✓

Figure 20. New password fulfilling all requirements

Confirm new password with OK.

4.2.3 Changing Password

Changing password is possible also at every moment, not only during the first connection. To change the password in other circumstances:

- expand the System container;
- expand Users;

- right-click the admin user;
- go to Actions → ChangePassword.

New password has to meet the same requirements as the password changed during the first connection.

To learn more about the nano EDGE ENGINE architecture, please refer to the nano EDGE ENGINE Programming user manual.

4.3 Emergency Mode

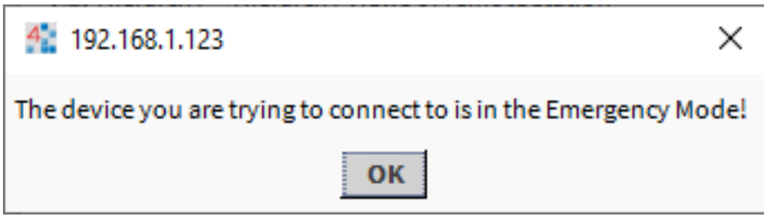
The system and application(s) of the nano EDGE ENGINE controllers are stored on an SD card. If the SD card is not detected in the device or the device detects frequent reboots (at least 5 times in 6 minutes), which prevent correct operation, the device enters an emergency mode.

What Causes the Emergency Mode?

- No SD card is detected in the device.
- The diagnostic process reveals error in I/Os.
- Storage limit is exceeded.
- Required files are missing during a start-up of the device.
- Libraries or files are corrupted.

4.3.1 Operation in Emergency Mode

Upon connecting to the device in the emergency mode, the following notice pops up:

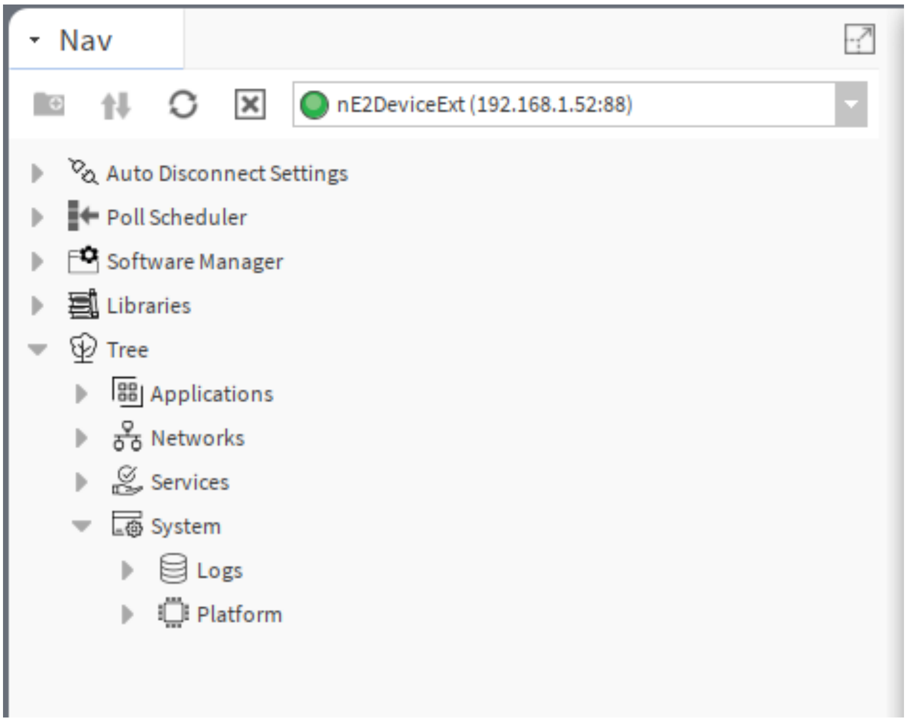


The device you are trying to connect to is in the Emergency Mode!

Figure 21. Emergency mode notice

In the emergency mode, the device operation is limited:

- libraries are not loaded;
- the SD card configuration is not loaded;
- only the System container with limited options (only Logs and Platform components) is displayed in the tree;



Nav

nE2DeviceExt (192.168.1.52:88)

- Auto Disconnect Settings
- Poll Scheduler
- Software Manager
- Libraries
- Tree
  - Applications
  - Networks
  - Services
  - System
    - Logs
    - Platform

Figure 22. Device tree in the emergency mode

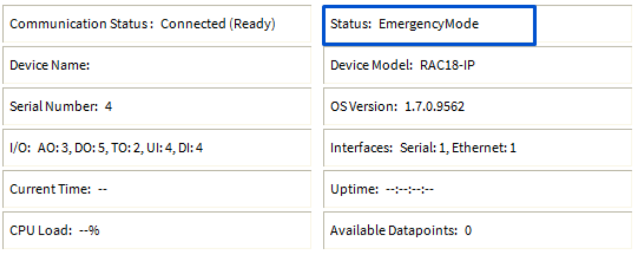
- the ALM LED is lit continuously;
- the iFnet runs with an IP/port taken from a flash storage;

Note: The flash storage must be synchronized to configuration slots when available.

- no authorization or credentials are taken from the flash storage (like IP/port).

nano EDGE ENGINE
where **innovation**
meets **simplicity**

nano
EDGE
ENGINE



Communication Status: Connected (Ready)	Status: EmergencyMode
Device Name:	Device Model: RAC18-IP
Serial Number: 4	OS Version: 1.7.0.9562
I/O: AO: 3, DO: 5, TO: 2, UI: 4, DI: 4	Interfaces: Serial: 1, Ethernet: 1
Current Time: --	Uptime: --:--:--:--
CPU Load: --%	Available Datapoints: 0

Figure 23. Start screen for a connected device in emergency mode

Possible Actions

When the device enters the emergency mode, take one of a few possible actions:

1. read logs from the SD card if available;
2. reboot;
3. restore to defaults (using the Restore in the System context menu): remove files from the SD card (if available and formatted) excluding only files with IP, port, and credentials (libraries must be also removed);

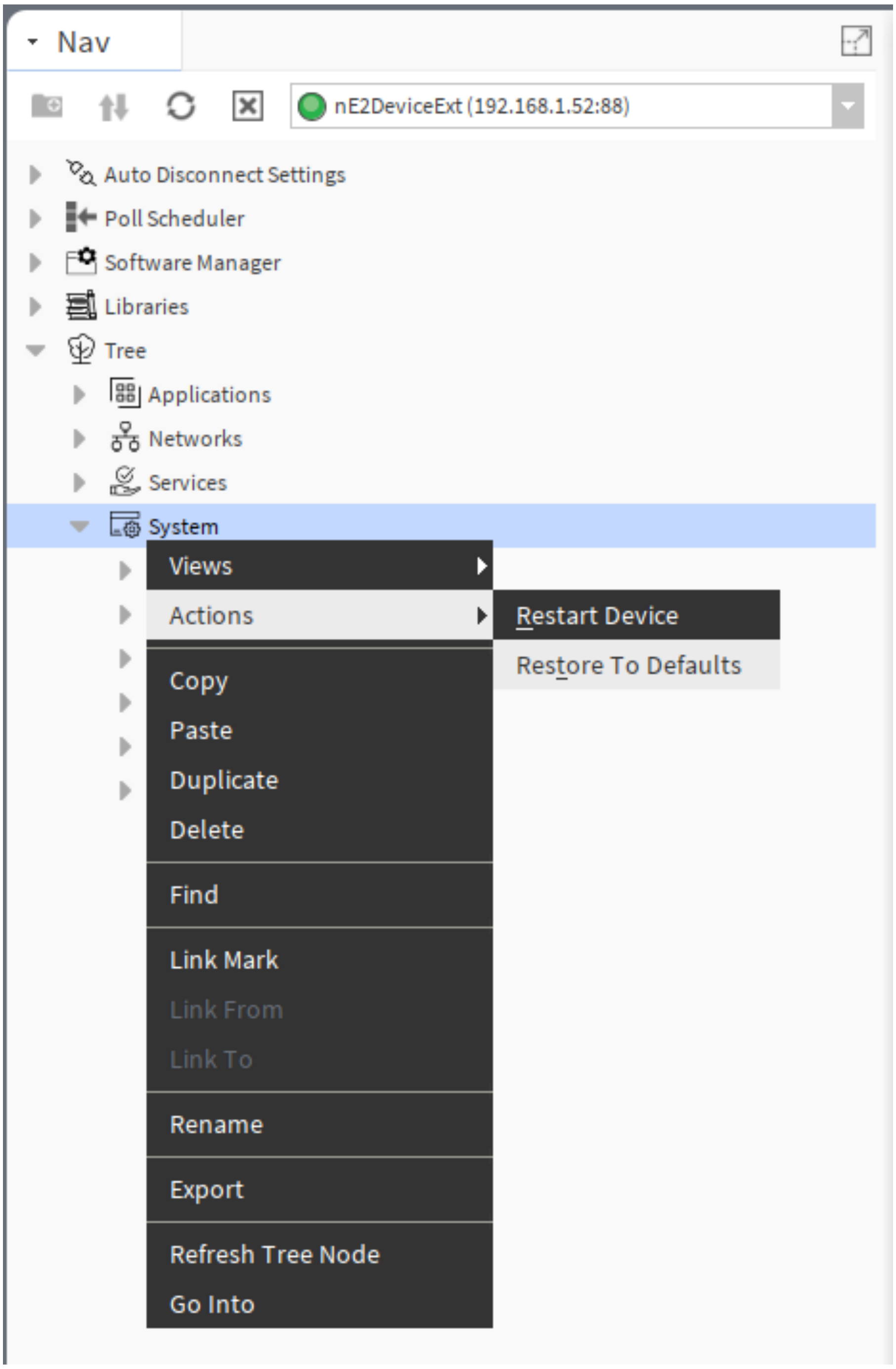


Figure 24. RestoreToDefaults action

4. restore to factory defaults (restoring with S1 6th DIP switch): format the SD card (if available), restore default credentials, IP, mask, gateway, iFnet port.

Warning - Factory Default Deletes Application

The process of bringing back factory default settings **erases the application from the controller**. In such a case, it is required to restore the application from the available backup.

For detailed information on how to restore the backup, please see: Default Communication Settings and Credentials.

4.4 IP Network Configuration

4.4.1 Change IP Network Settings

The IP address and other network settings are part of the Ethernet configuration in the Platform component in the System container.

To learn more about the System container, please refer to the nano EDGE ENGINE Programming user manual.

To change the network Settings:

- navigate to the Platform component in the System container;
- double-click on the Ethernet1 component.

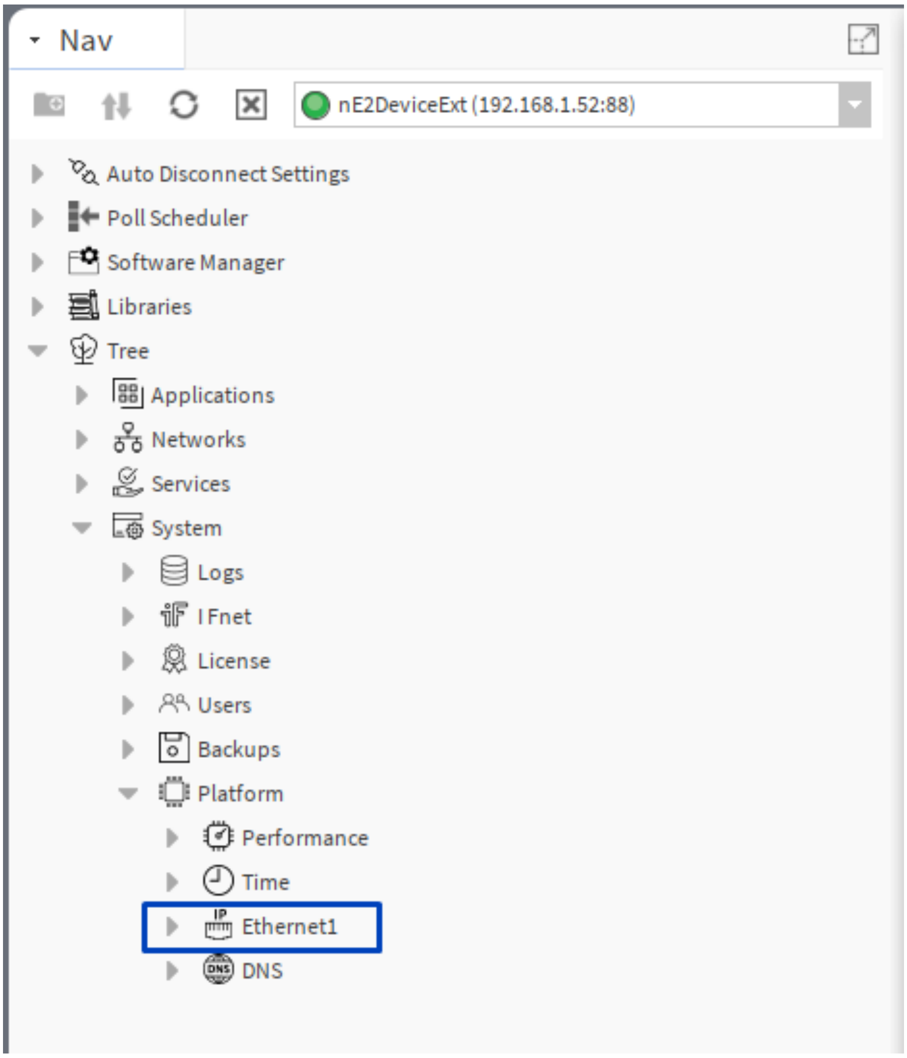
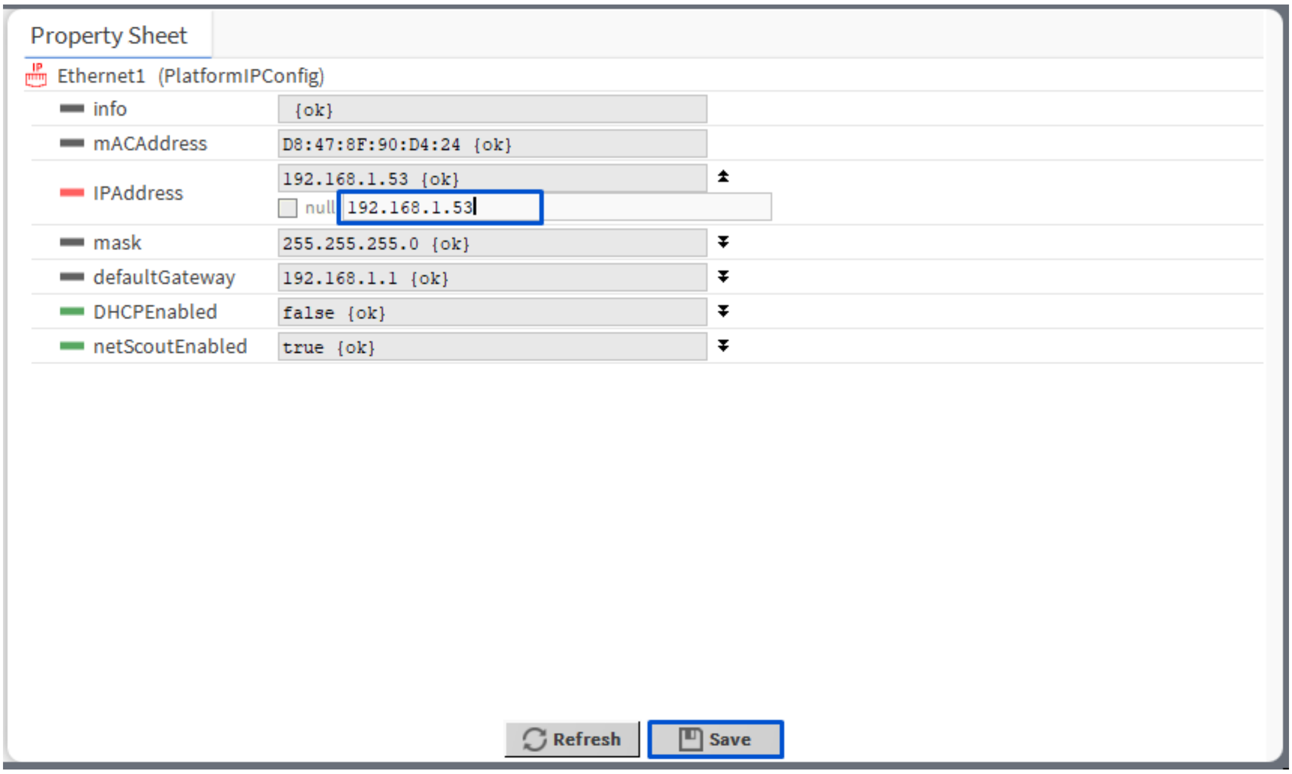


Figure 25. The Ethernet1 component in the tree

The Ethernet1 properties sheet will open on the main screen.

- Expand the IPAddress slot and type the new IP address, mask and gateway, or enable the DHCP mode.
- Confirm the new configuration with the Save button.


Property Sheet

Ethernet1 (PlatformIPConfig)

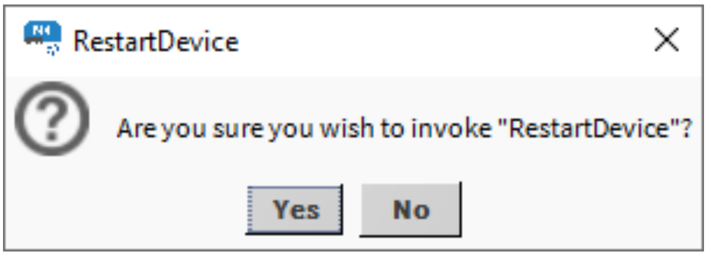
info	{ok}
mACAddress	D8:47:8F:90:D4:24 {ok}
IPAddress	192.168.1.53 {ok} □ null 192.168.1.53
mask	255.255.255.0 {ok}
defaultGateway	192.168.1.1 {ok}
DHCPEnabled	false {ok}
netScoutEnabled	true {ok}

Figure 26. Ethernet1 properties

To learn more about Ethernet1, please refer to the nano EDGE ENGINE Programming user manual.

- After changing the device address, right-click on Ethernet1 and go to Actions -> RestartDevice;

The device will be rebooted after confirming the RestartDevice prompt:


RestartDevice

Are you sure you wish to invoke "RestartDevice"?

Figure 27. RestartDevcie prompt

- Reconnect to the device by changing the IP address in the nE2DeviceExt using the Connect action.

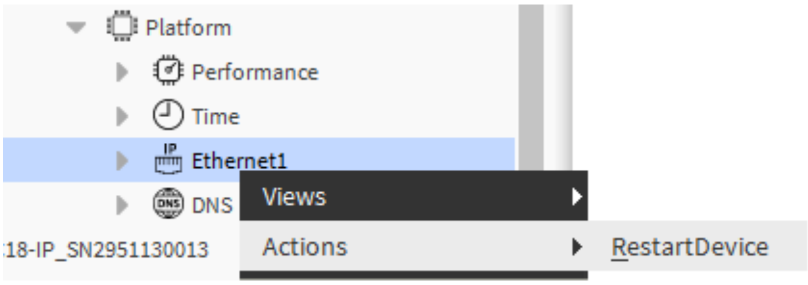
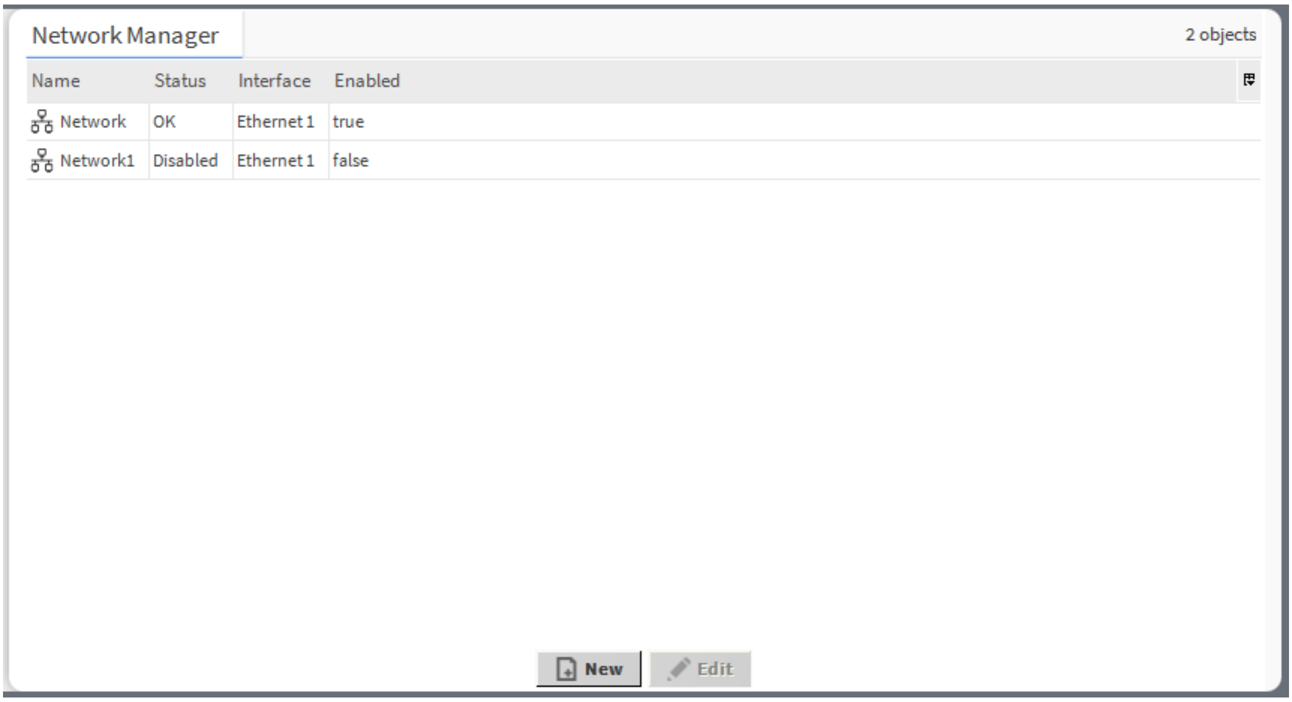


Figure 28. RestartDevice action in the Ethernet1 component

4.4.2 BACnet Network Manager

The Network Manager view is available for the BACnet component. It lists all BACnet networks configured on the device's ports. The Network Manager view shows the statuses, ports (which the network is configured on), and enabled or disabled states of the network.

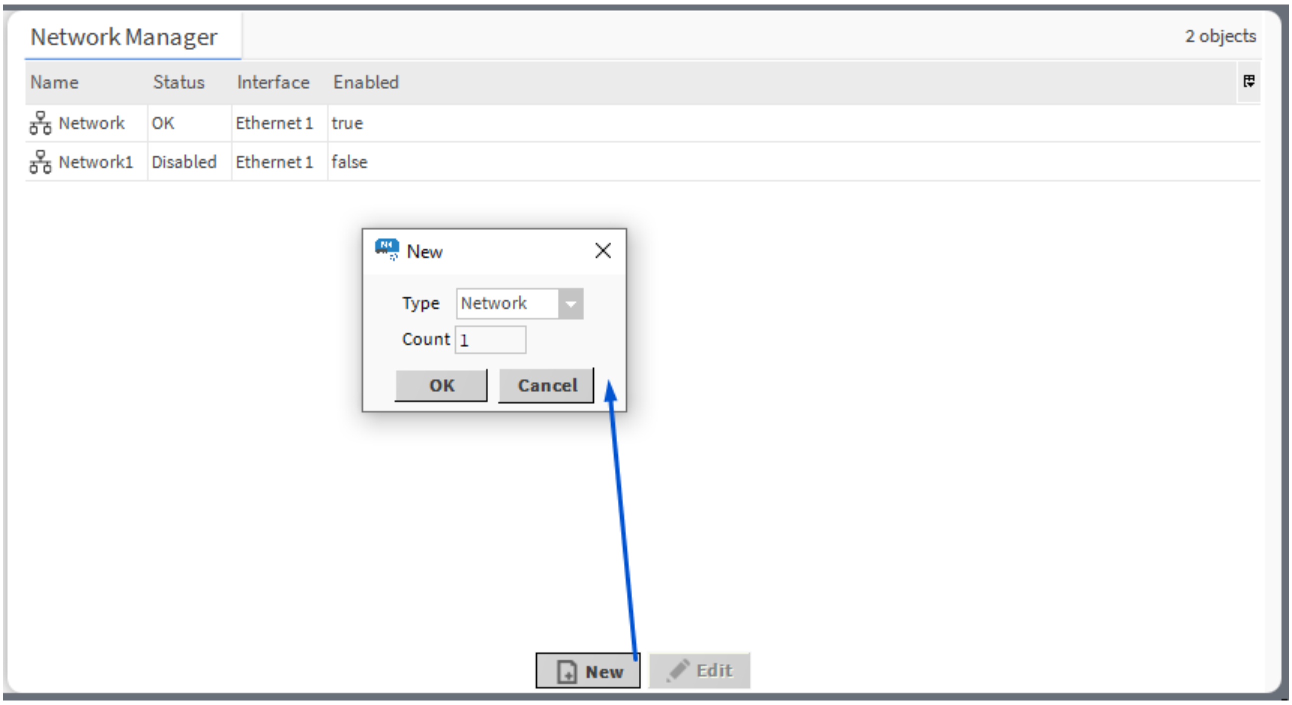

Network Manager — 2 objects

Name	Status	Interface	Enabled
Network	OK	Ethernet 1	true
Network1	Disabled	Ethernet 1	false

Figure 29. BACnet Network Manager

In the BACnet Network Manager, it is possible to:

- add BACnet network component:



Network Manager — 2 objects

Name	Status	Interface	Enabled
Network	OK	Ethernet 1	true
Network1	Disabled	Ethernet 1	false

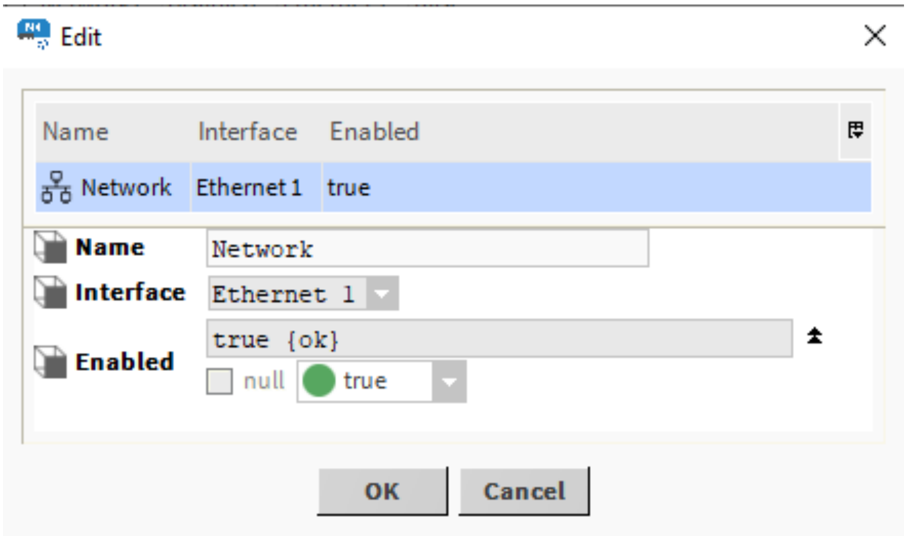
New

Type: Network

Count: 1

Figure 30. Adding new BACnet network

- edit the BACnet network's name, interface and enable/disable the component:



Edit

Name	Interface	Enabled
Network	Ethernet 1	true

Name: Network

Interface: Ethernet 1

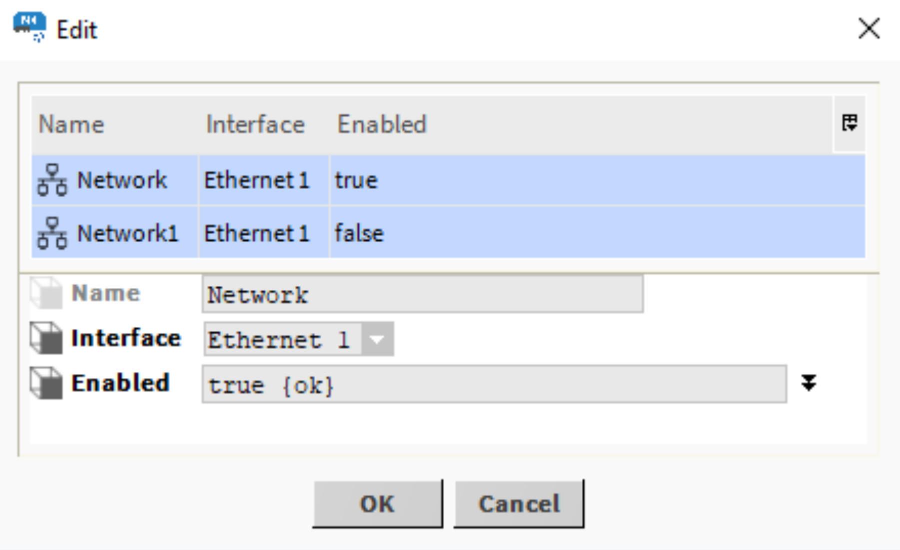
Enabled: true {ok}

□ null ● true

Figure 31. Editing pop-up

Note

Editing is possible for more than one network at a time. If multiple networks are edited, the same new value is written to common slots, so individual slots, such as Name, cannot be edited in this manner.



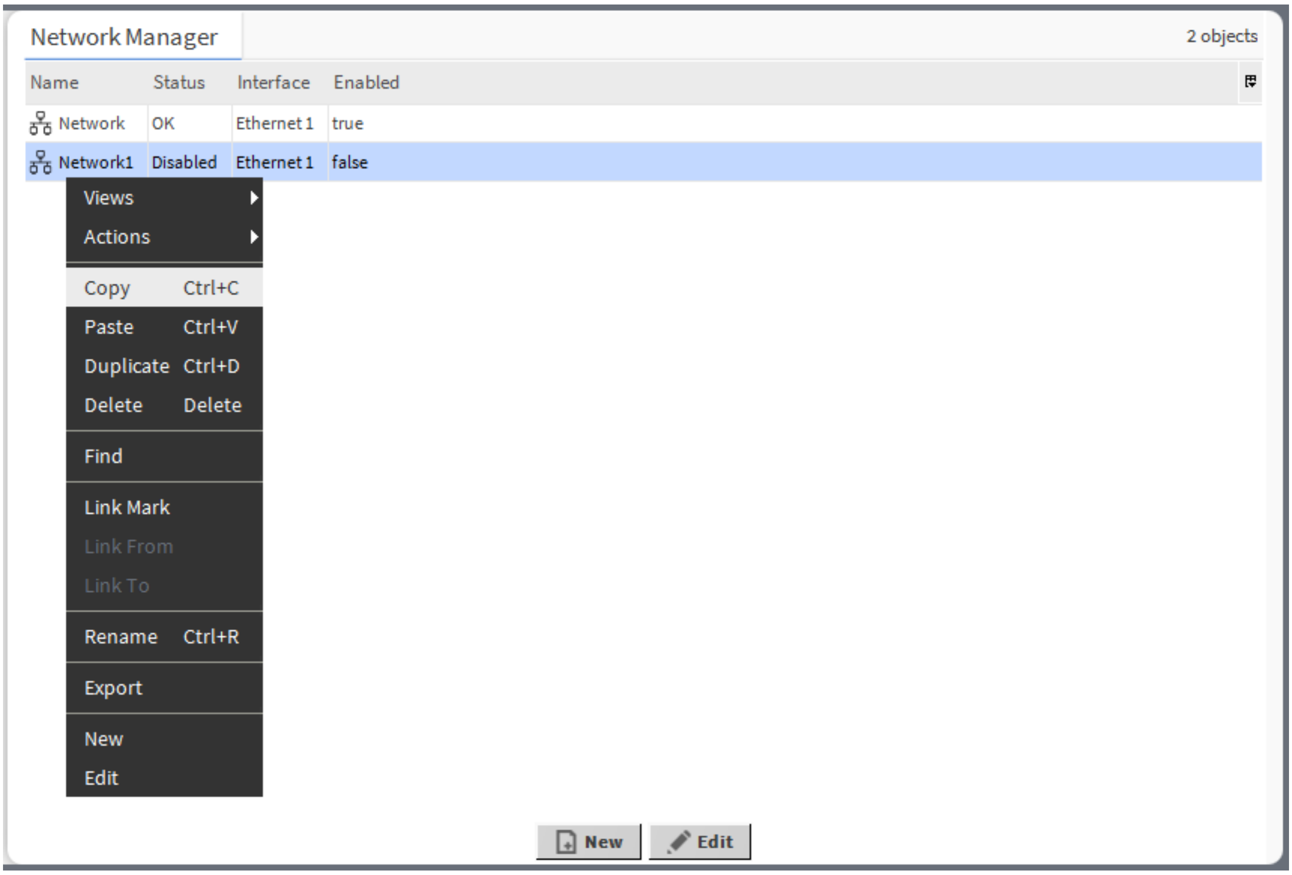
Edit

Name	Interface	Enabled
Network	Ethernet 1	true
Network1	Ethernet 1	false

Name: Network
Interface: Ethernet 1
Enabled: true {ok}

Figure 32. Editing multiple BACnet networks

- copy/duplicate/remove BACnet network components:



Network Manager

2 objects

Name	Status	Interface	Enabled
Network	OK	Ethernet 1	true
Network1	Disabled	Ethernet 1	false

Figure 33. Context menu options for BACnet network

Opening BACnet Network Manager

The BACnet Network Manager view is accessible from the context menu of the BACnet component. It is also automatically opened if the BACnet component is double-clicked in the nav tree window.

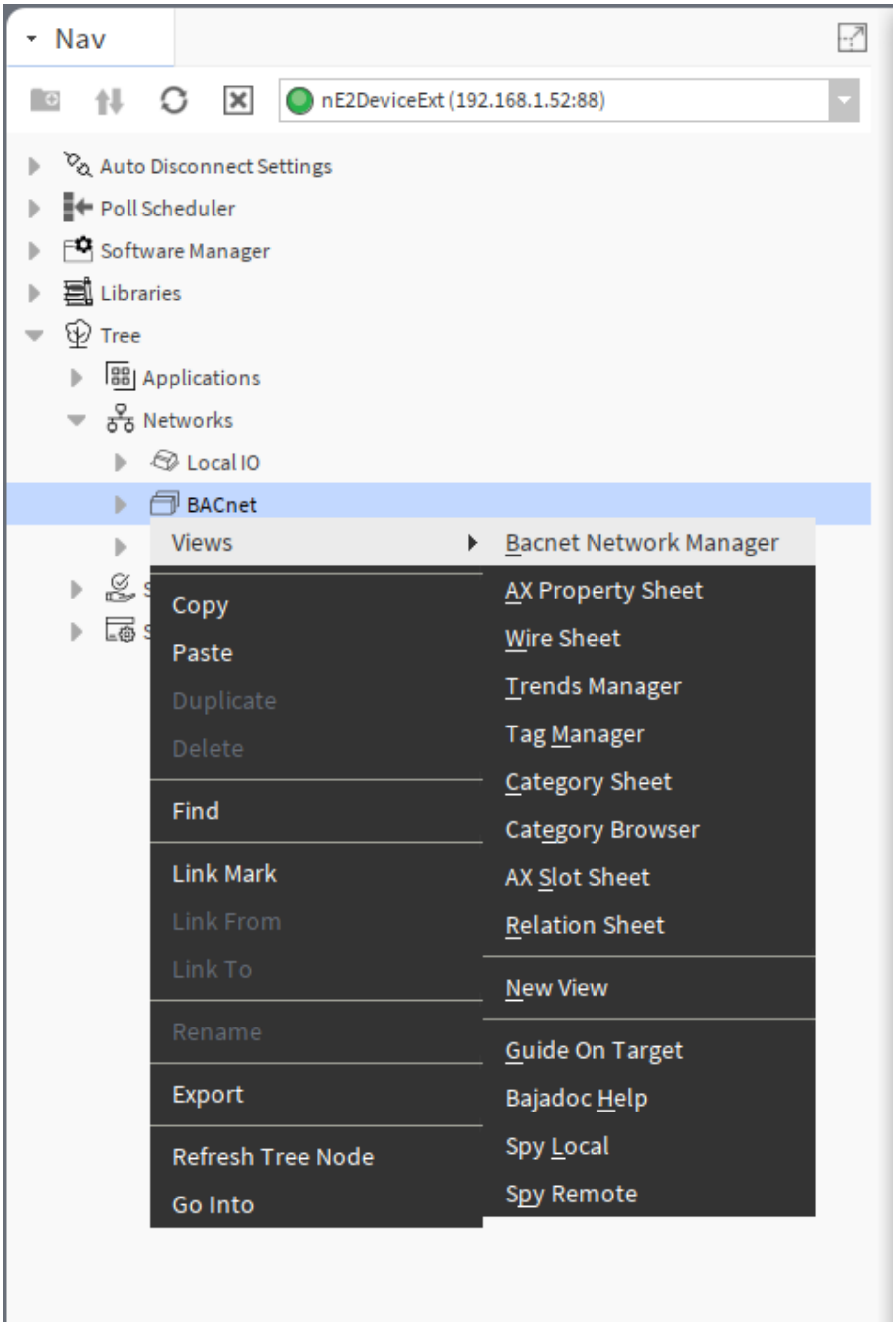


Figure 34. Accessing the BACnet Network Manager

4.4.3 BACnet Device Discover Manager

The Device Discover Manager view is available for the BACnet Network component. It lists all BACnet devices added to the network. The Device Discover Manager view shows the names, statuses, device names and ID, and enabled or disabled states of the device. The special Exts columns provides quick access to the Point Manager (with all points of the relevant device).

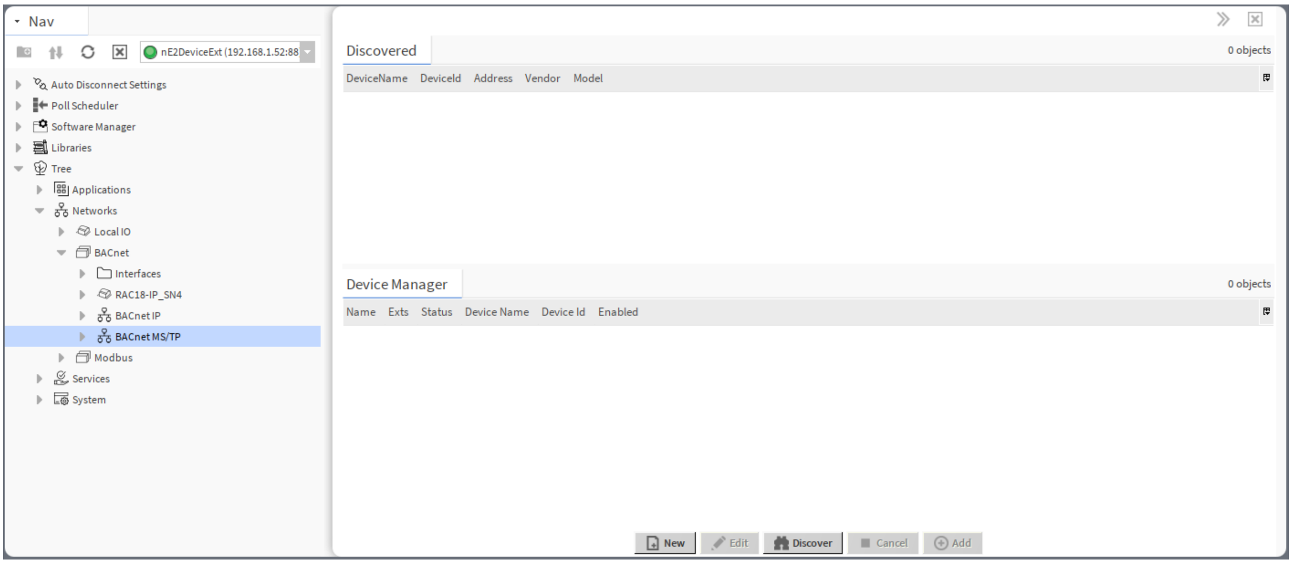
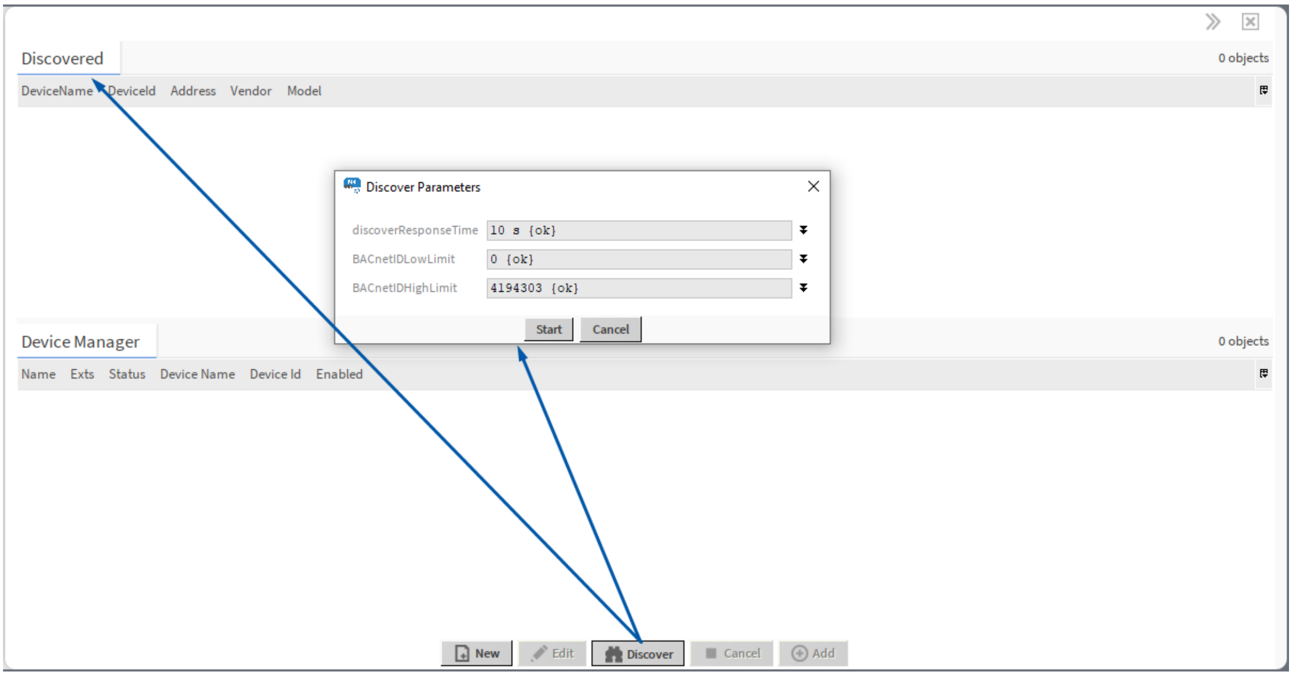


Figure 35. BACnet Device Discover Manager

In the BACnet Device Manager, it is possible to:

- discover BACnet device(s) available on the network:

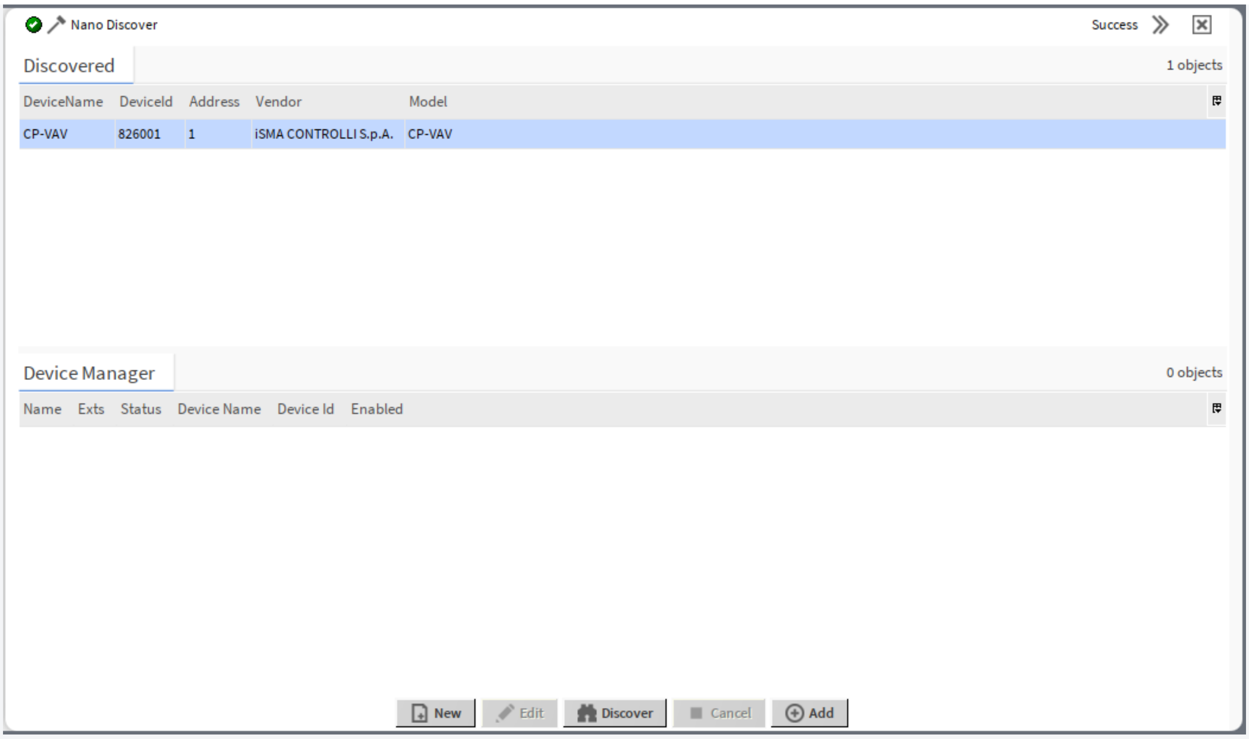


discoverResponseTime	10 s {ok}
BACnetIDLowLimit	0 {ok}
BACnetIDHighLimit	4194303 {ok}

Figure 36. Discovering devices on the BACnet network

To discover devices available on the BACnet network, use the Discover button. In the pop-up, set the discover parameters:

- discover response time,
- BACnet ID low limit,
- BACnet ID high limit.



Nano Discover

Success

Discovered

1 objects

DeviceName	DeviceId	Address	Vendor	Model
CP-VAV	826001	1	iSMA CONTROLLI S.p.A.	CP-VAV

Device Manager

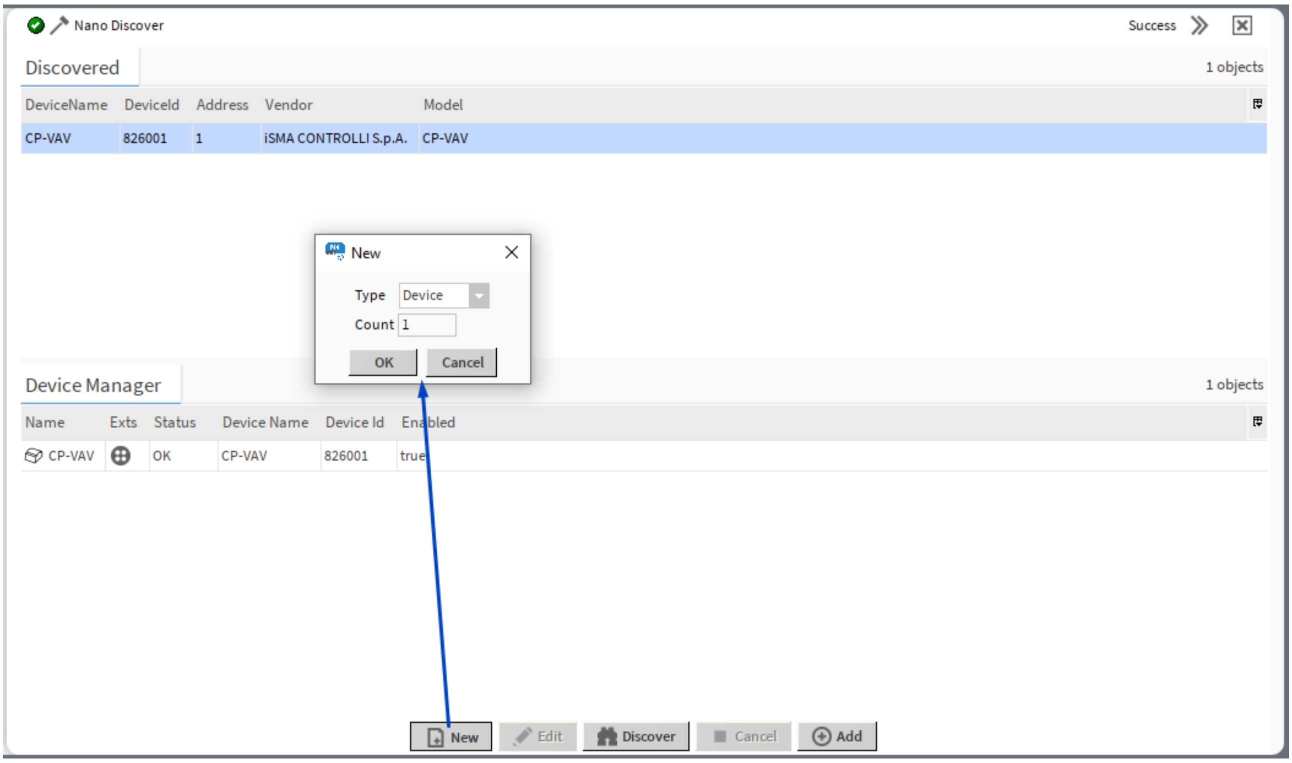
0 objects

Name	Exts	Status	Device Name	Device Id	Enabled

Figure 37. Device discovered

Once the device is discovered on the network, it appears in the Discovered window. To manage the device, add it to the Device Manager section (drag and drop or use the Add button - if the list includes a lot of devices to add, first, use the Select All option from the context menu).

- manually add the BACnet Device component:



Nano Discover

Success

Discovered

1 objects

DeviceName	DeviceId	Address	Vendor	Model
CP-VAV	826001	1	iSMA CONTROLLI S.p.A.	CP-VAV

New

Type: Device

Count: 1

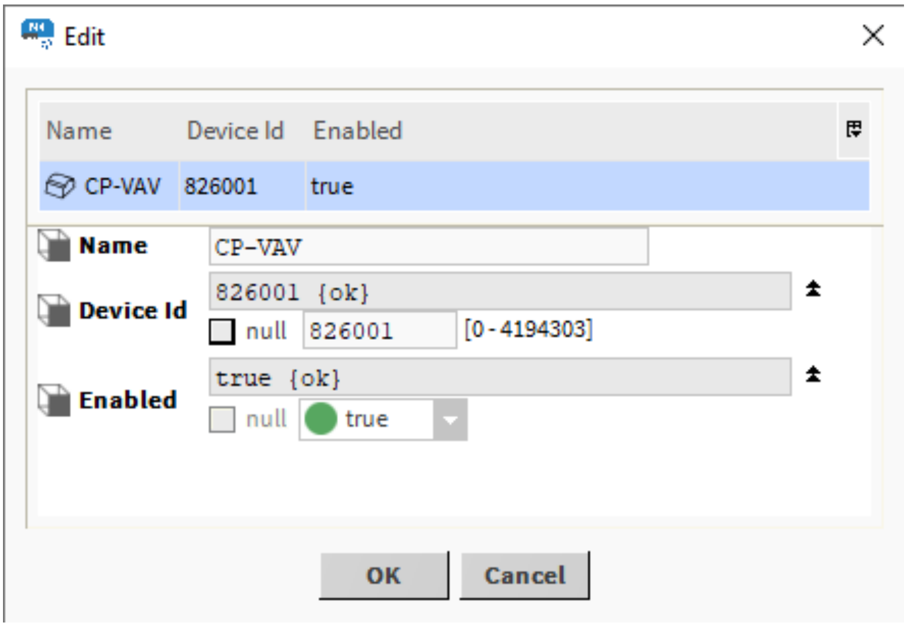
Device Manager

1 objects

Name	Exts	Status	Device Name	Device Id	Enabled
CP-VAV	⊕	OK	CP-VAV	826001	true

Figure 38. Adding new device in the Device Discover Manager

- edit the BACnet device’s name, device ID and enable/disable the component:



Name	Device Id	Enabled
CP-VAV	826001	true

Name: CP-VAV

Device Id: 826001 {ok}

□ null 826001 [0 - 4194303]

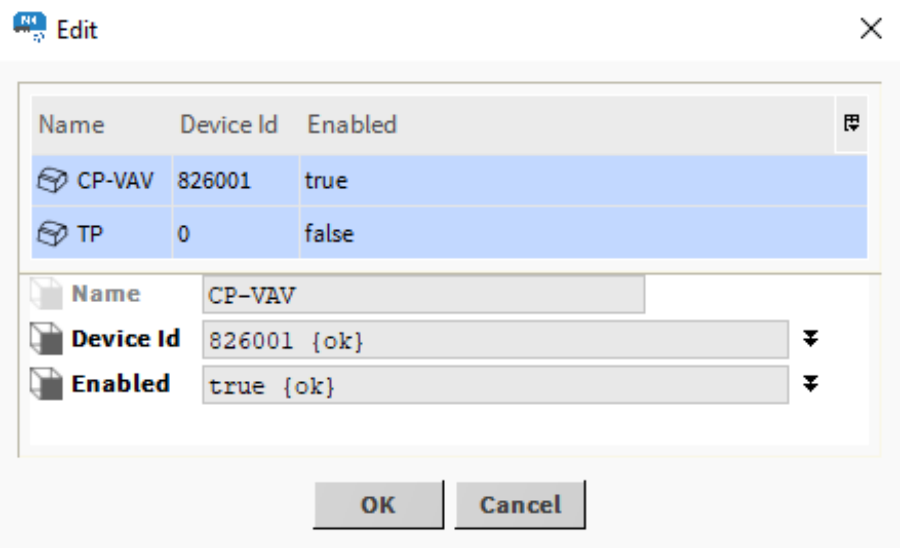
Enabled: true {ok}

□ null ● true

Figure 39. Editing pop-up

Note

Editing is possible for more than one device at a time. If multiple devices are edited, the same new value is written to common slots, so individual slots, such as Name, cannot be edited in this manner.



Name	Device Id	Enabled
CP-VAV	826001	true
TP	0	false

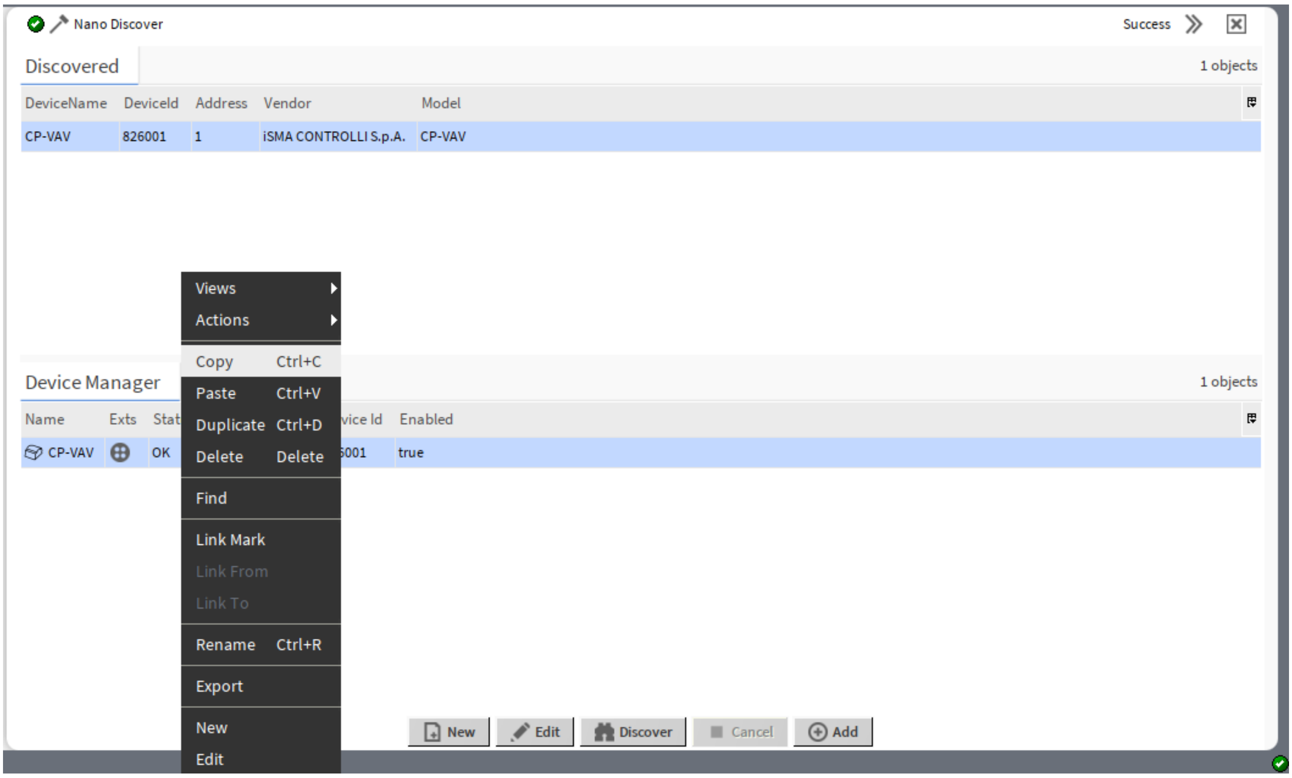
Name: CP-VAV

Device Id: 826001 {ok}

Enabled: true {ok}

Figure 40. Editing multiple devices

- copy/paste/duplicate/delete BACnet device components:



DeviceName	DeviceId	Address	Vendor	Model
CP-VAV	826001	1	iSMA CONTROLLI S.p.A.	CP-VAV

Name	Exts	Stat	vice Id	Enabled
CP-VAV	⊕	OK	5001	true

Figure 41. Context menu options for the BACnet device

Opening BACnet Device Manager

The BACnet Device Manager view is accessible from the context menu of the BACnet Network component. It is also automatically opened if the BACnet Network component is double-clicked in the nav tree window.

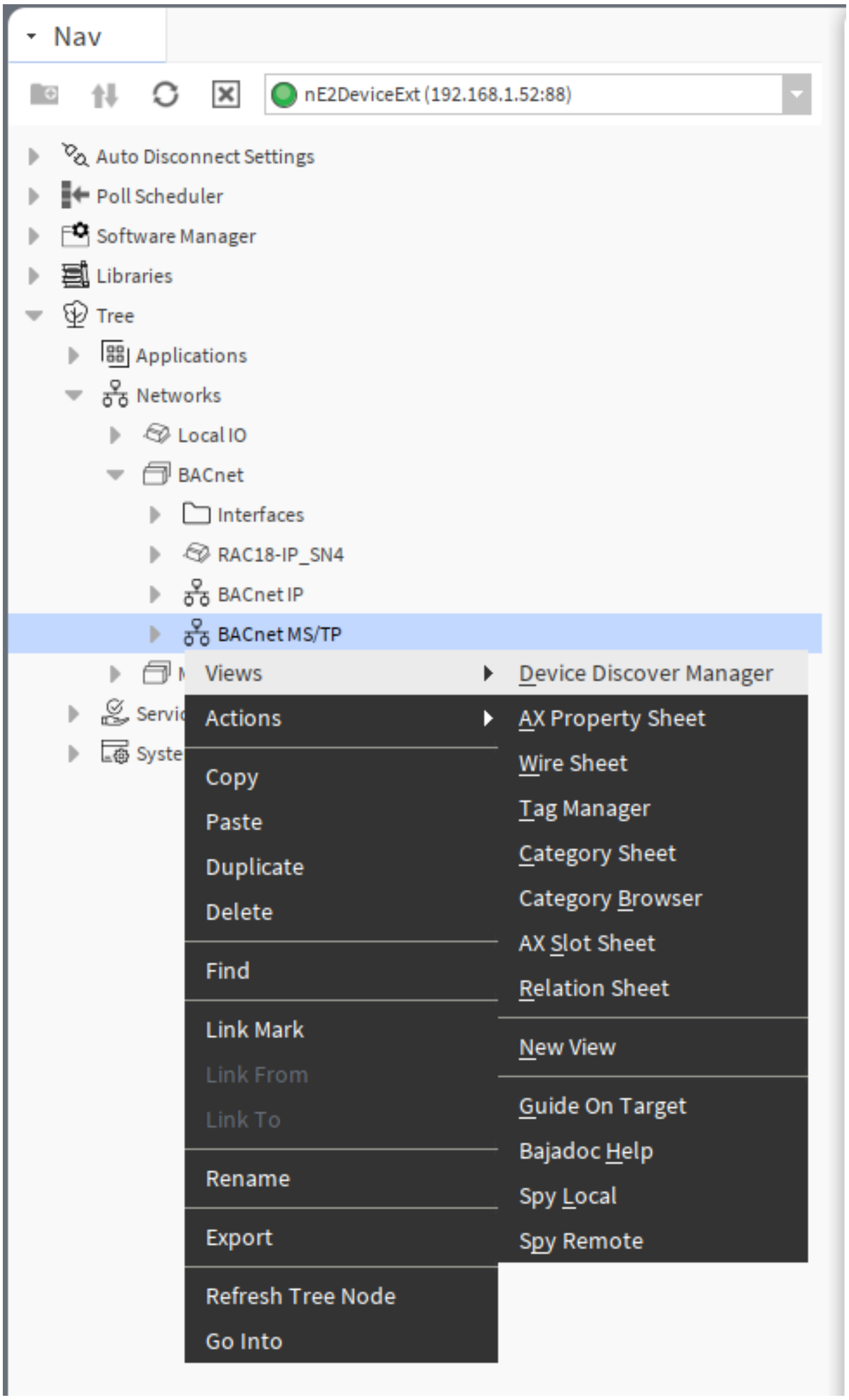
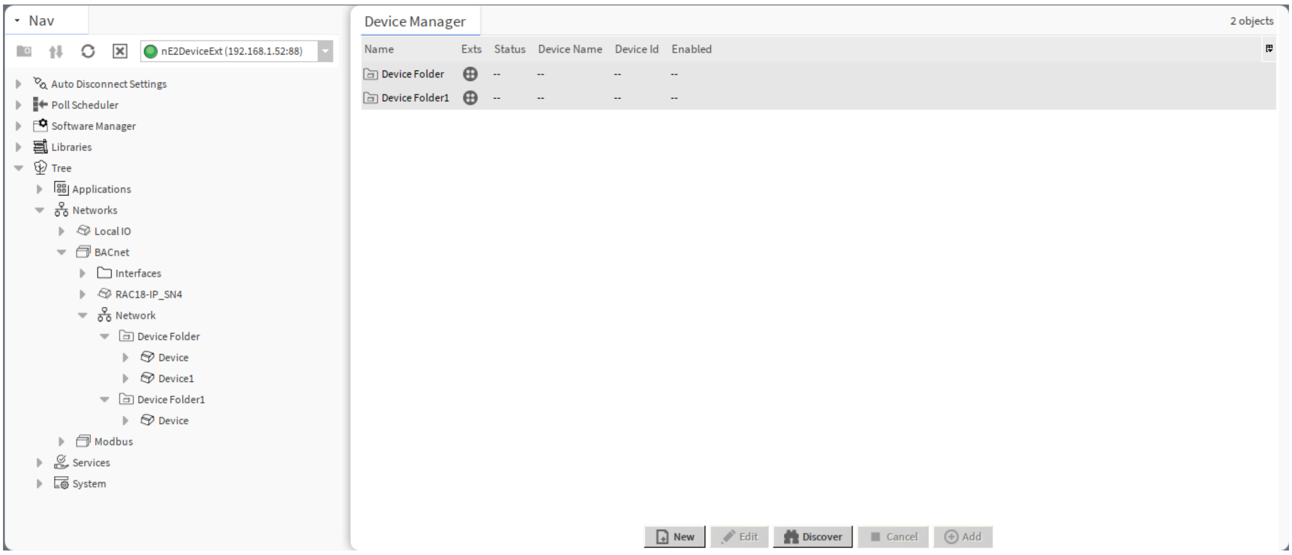


Figure 42. Accessing the BACnet Device Discover Manager

DeviceFolder

The DeviceFolder component is a device grouping component, which can be added under the BACnet network. It allows to organize devices in the network on the nav tree and maintains the grouping in the Device Manager view. It is clickable in the Device Manager allowing to display only devices grouped in one DeviceFolder component.



Name	Exts	Status	Device Name	Device Id	Enabled
Device Folder	⊕	--	--	--	--
Device Folder1	⊕	--	--	--	--

Figure 43. DeviceFolder

The DeviceFolder component has no slots nor actions. It is available in the Core library.

4.4.4 BACnet Point Manager

The BACnet Point Manager view is available for each device added to the BACnet network. It allows to discover new BACnet points available for the devices and lists all BACnet points added to the Device component (AnalogPoint/BinaryPoint/MultistatePoint and AnalogCustomPoint/BinaryCustomPoint/StringCustomPoint).

The BACnet Point Manager view is divided into 2 sections: Discovered and BACnet Point Manager. In the Discovered section, the view shows the points' name, object Id, object type, type and value. In the BACnet Point Manager section, the view shows the points' Out slot value, status, object name and ID, polling mode, and enabled or disabled state.

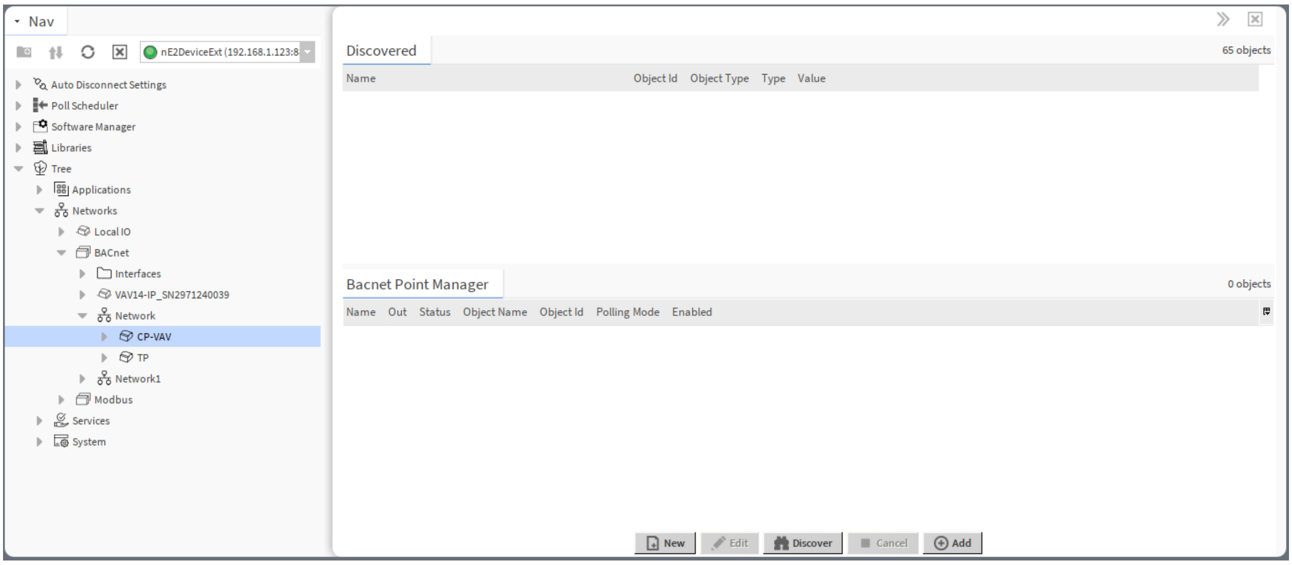
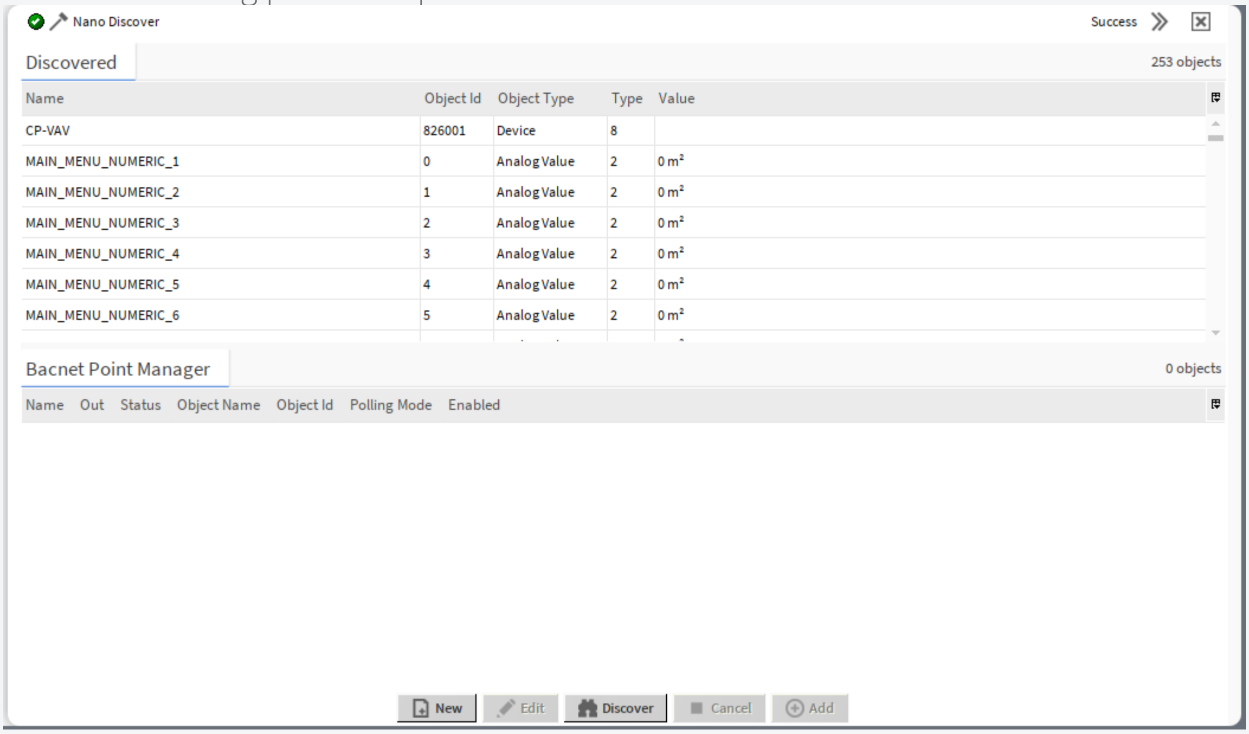


Figure 44. BACnet Point Manager

In the BACnet Point Manager, it is possible to:

- discover BACnet points of the given BACnet device:

To discover points of the given BACnet device, use the Discover button. The progress of the discovering process is presenter on a bar.



Discovered — 253 objects

Name	Object Id	Object Type	Type	Value
CP-VAV	826001	Device	8	
MAIN_MENU_NUMERIC_1	0	Analog Value	2	0 m^2
MAIN_MENU_NUMERIC_2	1	Analog Value	2	0 m^2
MAIN_MENU_NUMERIC_3	2	Analog Value	2	0 m^2
MAIN_MENU_NUMERIC_4	3	Analog Value	2	0 m^2
MAIN_MENU_NUMERIC_5	4	Analog Value	2	0 m^2
MAIN_MENU_NUMERIC_6	5	Analog Value	2	0 m^2

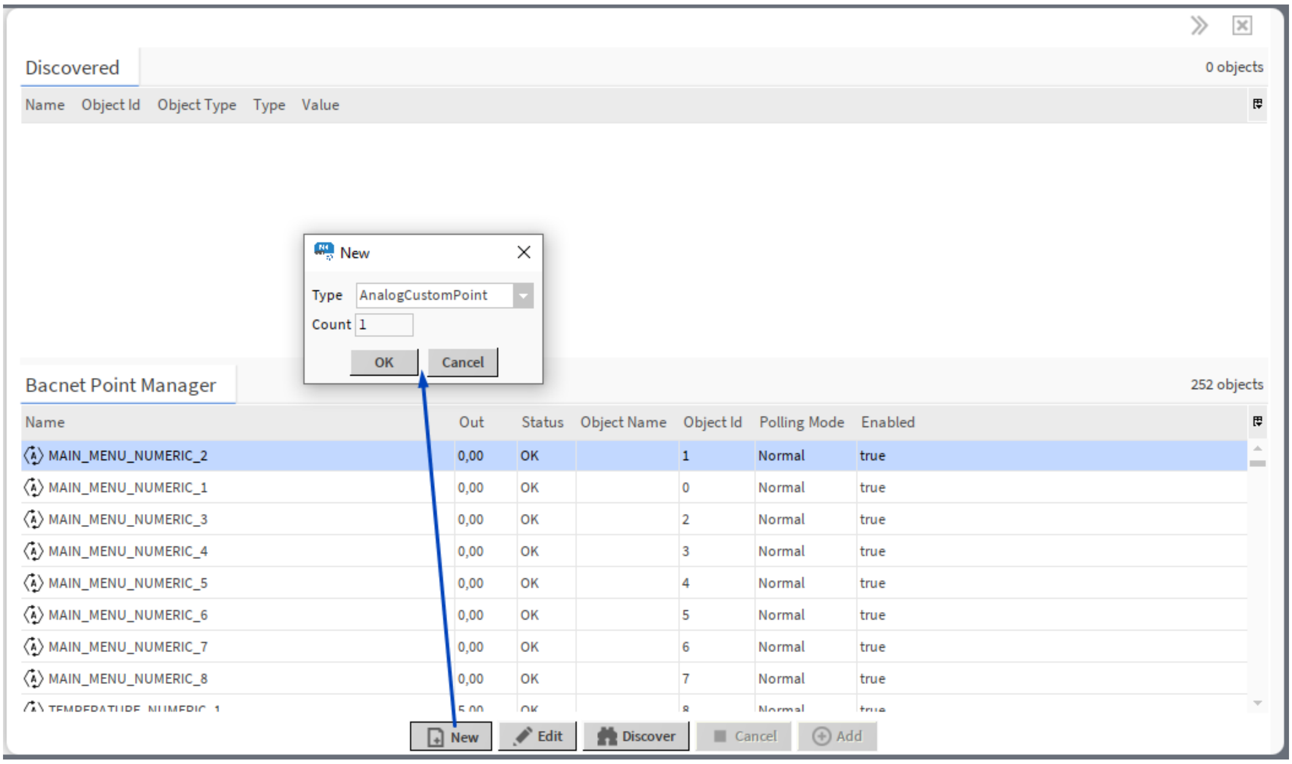
Bacnet Point Manager — 0 objects

Name	Out	Status	Object Name	Object Id	Polling Mode	Enabled

Figure 45. Points discovered

Once points are discovered for the device, they appear in the Discovered window. To manage the points, add them to the Point Manager section (drag and drop or use the Add button - if the list includes a lot of points to add, first, use the Select All option from the context menu).

- manually add BACnet points:



Discovered — 0 objects

Name	Object Id	Object Type	Type	Value

New

Type: AnalogCustomPoint

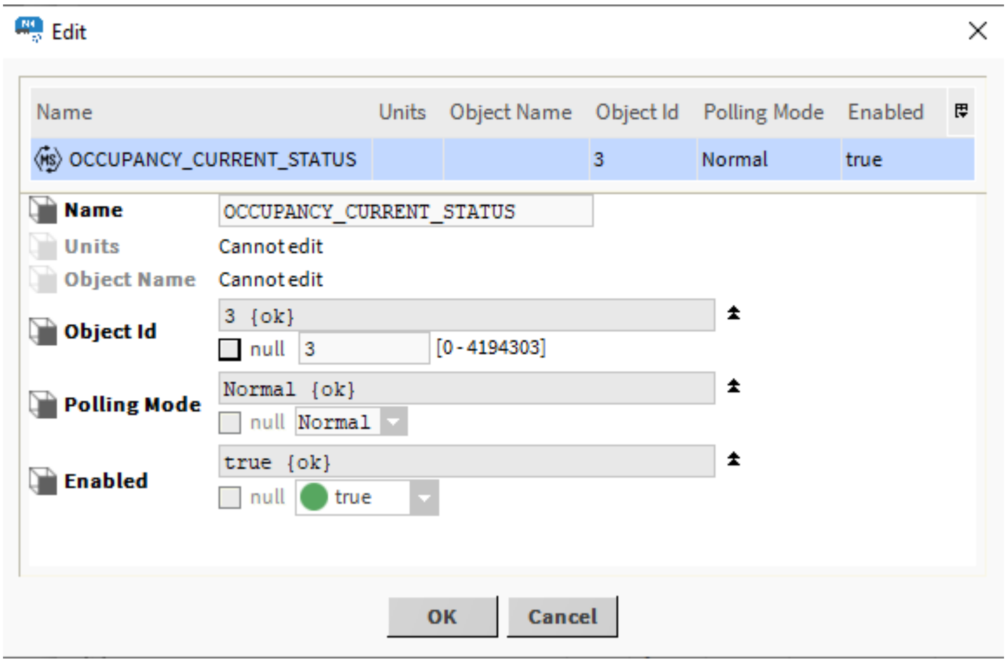
Count: 1

Bacnet Point Manager — 252 objects

Name	Out	Status	Object Name	Object Id	Polling Mode	Enabled
MAIN_MENU_NUMERIC_2	0,00	OK		1	Normal	true
MAIN_MENU_NUMERIC_1	0,00	OK		0	Normal	true
MAIN_MENU_NUMERIC_3	0,00	OK		2	Normal	true
MAIN_MENU_NUMERIC_4	0,00	OK		3	Normal	true
MAIN_MENU_NUMERIC_5	0,00	OK		4	Normal	true
MAIN_MENU_NUMERIC_6	0,00	OK		5	Normal	true
MAIN_MENU_NUMERIC_7	0,00	OK		6	Normal	true
MAIN_MENU_NUMERIC_8	0,00	OK		7	Normal	true
TEMPERATURE_NUMERIC_1	5,00	OK		8	Normal	true

Figure 46. Adding BACnet point

- edit the BACnet point's name, units, address and enable/disable the component:



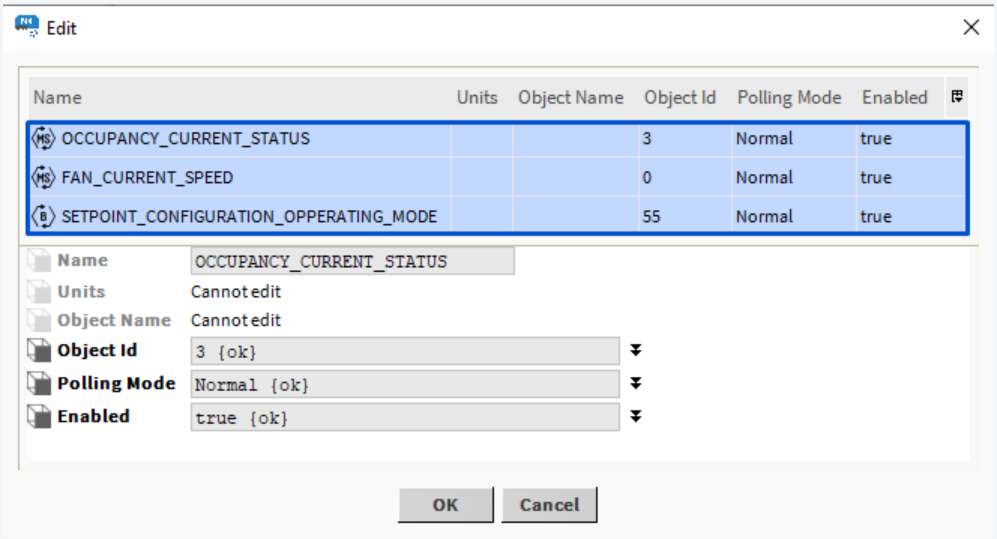
Edit

Name	Units	Object Name	Object Id	Polling Mode	Enabled
OCCUPANCY_CURRENT_STATUS			3	Normal	true

Name: OCCUPANCY_CURRENT_STATUS
Units: Cannot edit
Object Name: Cannot edit
Object Id: 3 {ok}
□ null 3 [0 - 4194303]
Polling Mode: Normal {ok}
□ null Normal
Enabled: true {ok}
□ null ● true

Figure 47. Editing pop-up

Note
Editing is possible for more than one point at a time. If multiple points are edited, the same new value is written to common slots, so individual slots, such as Name, cannot be edited in this manner.



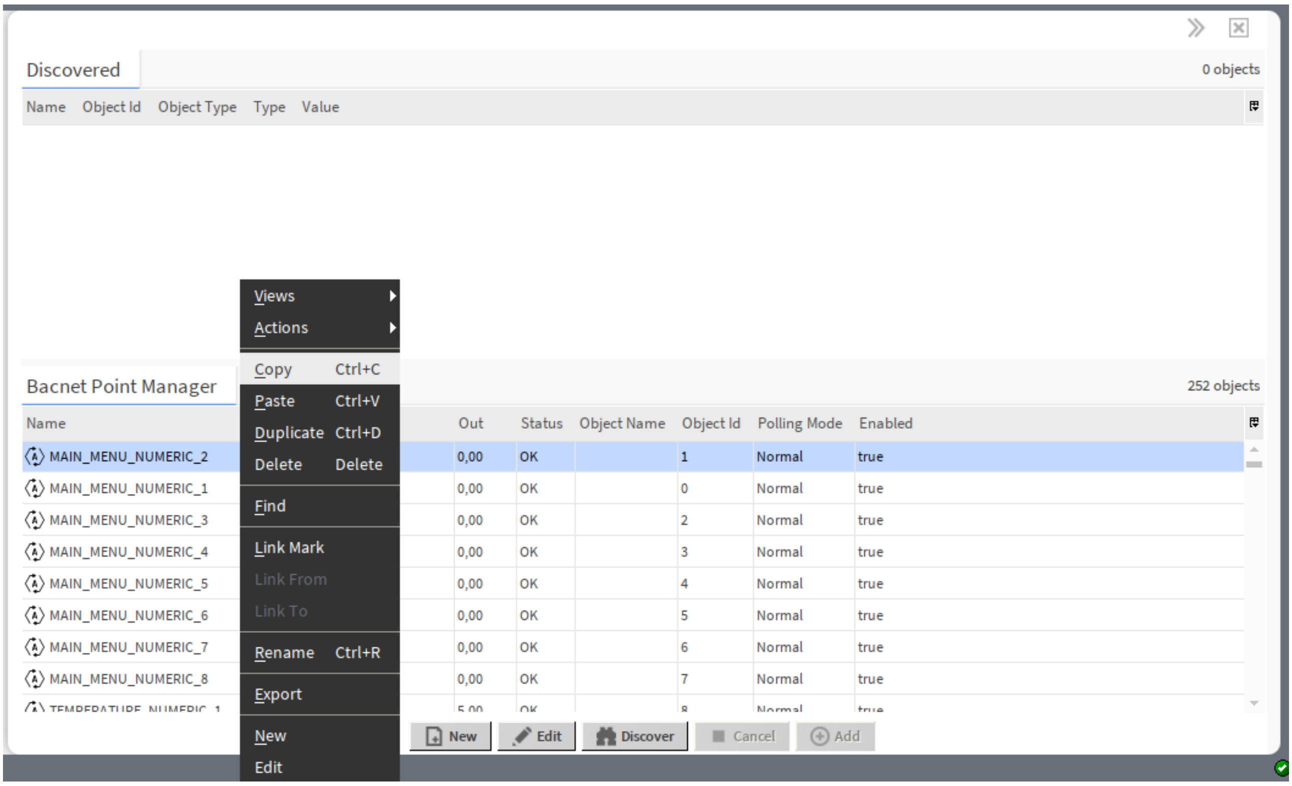
Edit

Name	Units	Object Name	Object Id	Polling Mode	Enabled
OCCUPANCY_CURRENT_STATUS			3	Normal	true
FAN_CURRENT_SPEED			0	Normal	true
SETPOINT_CONFIGURATION_OPPERATING_MODE			55	Normal	true

Name: OCCUPANCY_CURRENT_STATUS
Units: Cannot edit
Object Name: Cannot edit
Object Id: 3 {ok}
Polling Mode: Normal {ok}
Enabled: true {ok}

Figure 48. Editing of multiple points

- copy/paste/duplicate/delete BACnet points:



Name	Object Id	Object Type	Type	Value

Name	Out	Status	Object Name	Object Id	Polling Mode	Enabled
MAIN_MENU_NUMERIC_2	0,00	OK		1	Normal	true
MAIN_MENU_NUMERIC_1	0,00	OK		0	Normal	true
MAIN_MENU_NUMERIC_3	0,00	OK		2	Normal	true
MAIN_MENU_NUMERIC_4	0,00	OK		3	Normal	true
MAIN_MENU_NUMERIC_5	0,00	OK		4	Normal	true
MAIN_MENU_NUMERIC_6	0,00	OK		5	Normal	true
MAIN_MENU_NUMERIC_7	0,00	OK		6	Normal	true
MAIN_MENU_NUMERIC_8	0,00	OK		7	Normal	true
TEMPERATURE_NUMERIC_1	5,00	OK		8	Normal	true

Figure 49. Context menu options for the BACnet point

Opening the BACnet Point Manager

The BACnet Point Manager view is accessible from the context menu of the Device component. It is also automatically opened if the Device component is double-clicked in the nav tree window.

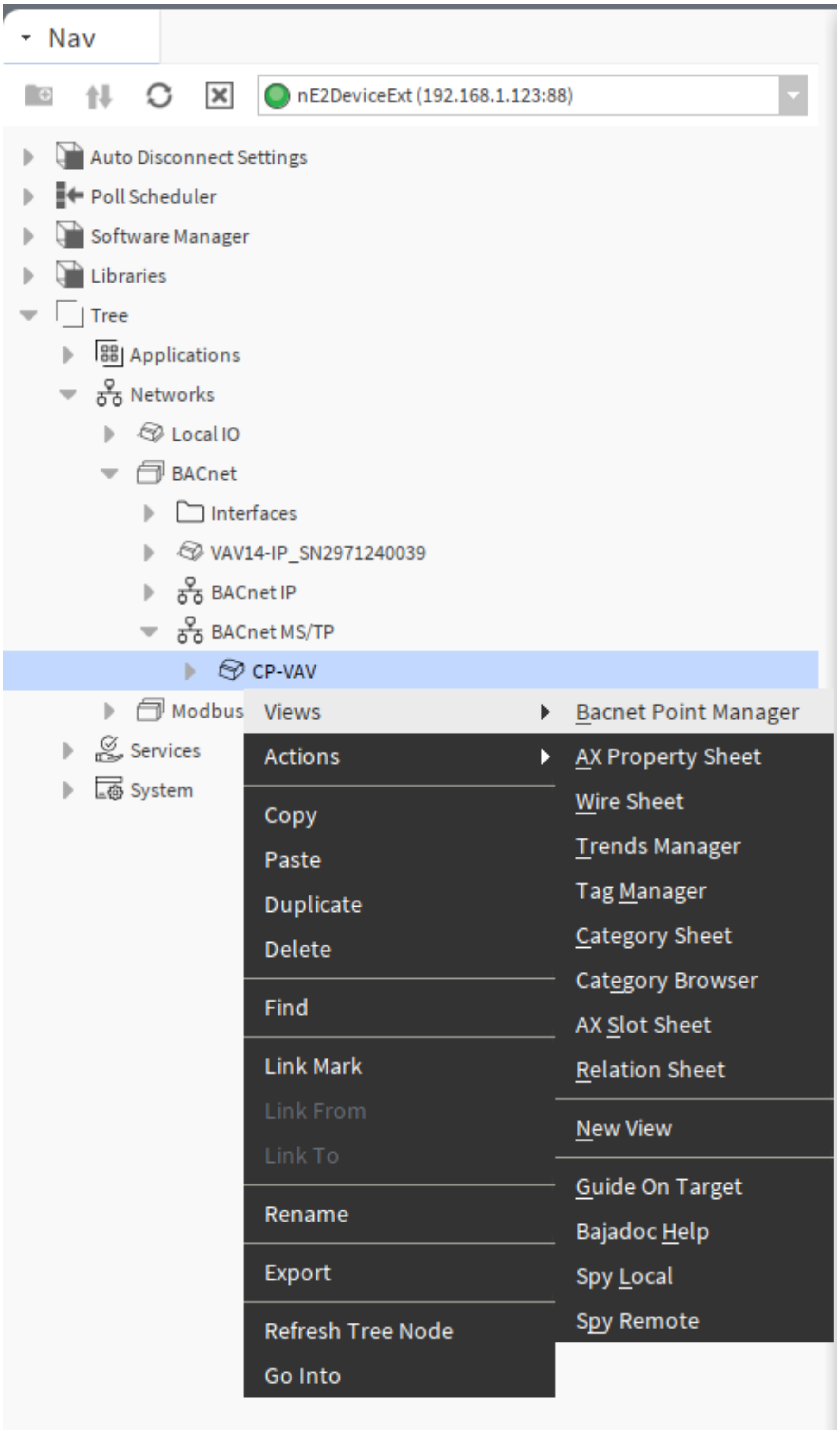
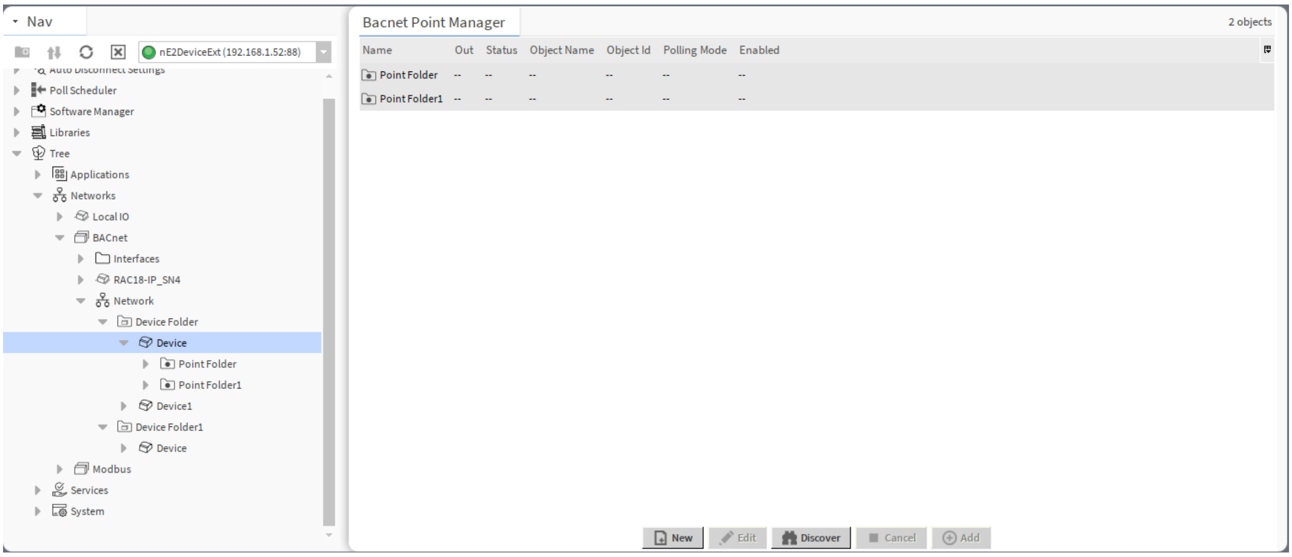


Figure 50. Accessing the BACnet Point Manager from the context menu

PointFolder

The PointFolder component is a point grouping component, which can be added under the BACnet device. It allows to organize points of the device on the nav tree and maintains the grouping in the Point Manager view. It is clickable in the Point Manager allowing to display only points grouped in one PointFolder component.



Bacnet Point Manager

2 objects

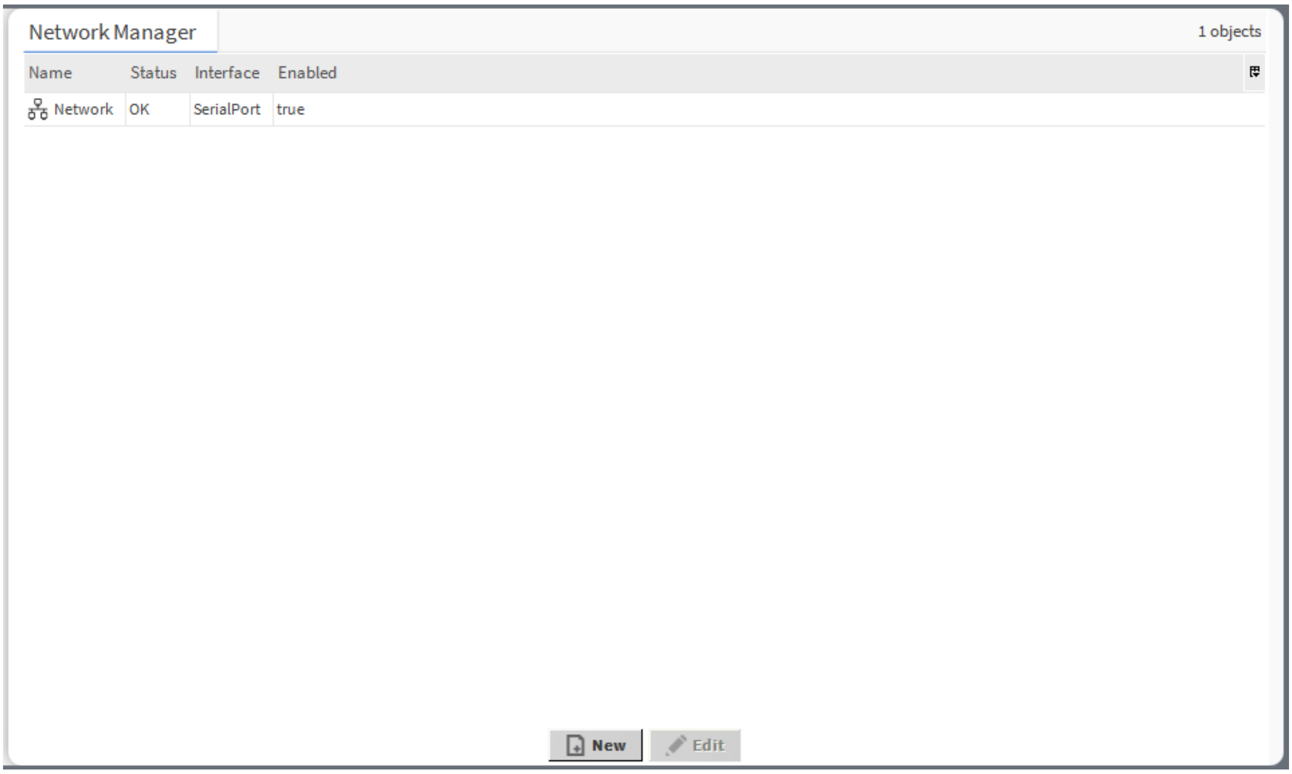
Name	Out	Status	Object Name	Object Id	Polling Mode	Enabled
Point Folder	--	--	--	--	--	--
Point Folder1	--	--	--	--	--	--

Figure 51. PointFolder

The PointFolder component has no slots nor actions. The PointFolder is available in the Core library.

4.4.5 Modbus Network Manager

The Network Manager view is available for the Modbus component. It lists all Modbus networks configured on the device's ports. The Network Manager view shows the statuses, ports (which the network is configured on), and enabled or disabled states of the network.



Network Manager

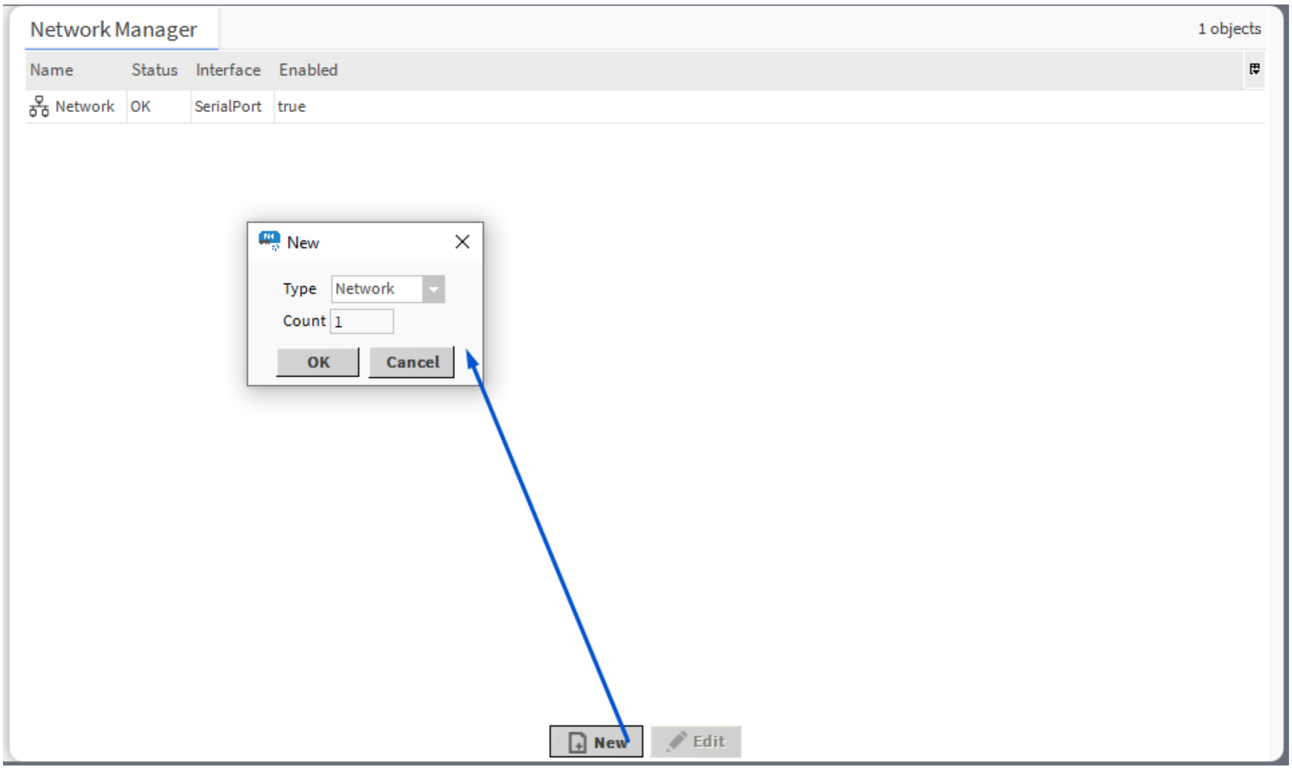
1 objects

Name	Status	Interface	Enabled
Network	OK	SerialPort	true

Figure 52. Modbus Network Manager

In the Modbus Network Manager, it is possible to:

- add Modbus network component:



Network Manager — 1 objects

Name	Status	Interface	Enabled
Network	OK	SerialPort	true

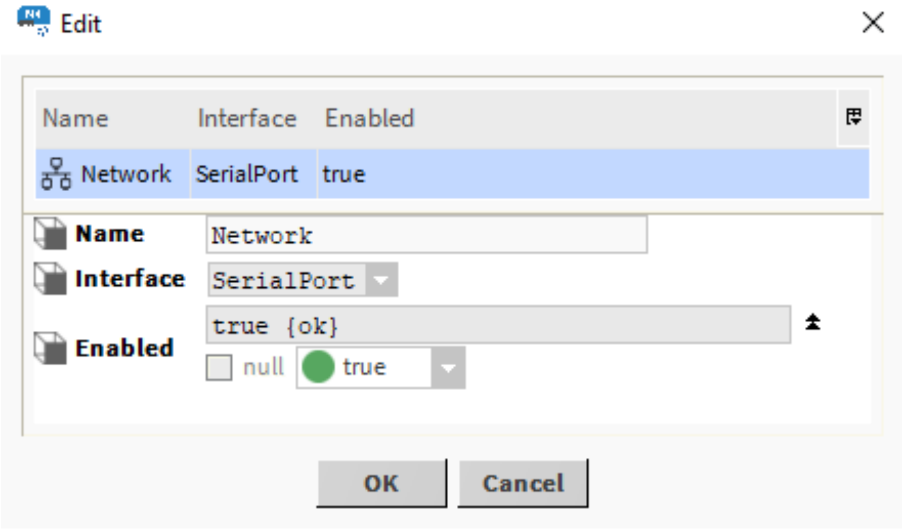
New

Type: Network

Count: 1

Figure 53. Adding new Modbus network

- edit the Modbus network's name, interface and enable/disable the component:



Edit

Name	Interface	Enabled
Network	SerialPort	true

Name: Network

Interface: SerialPort

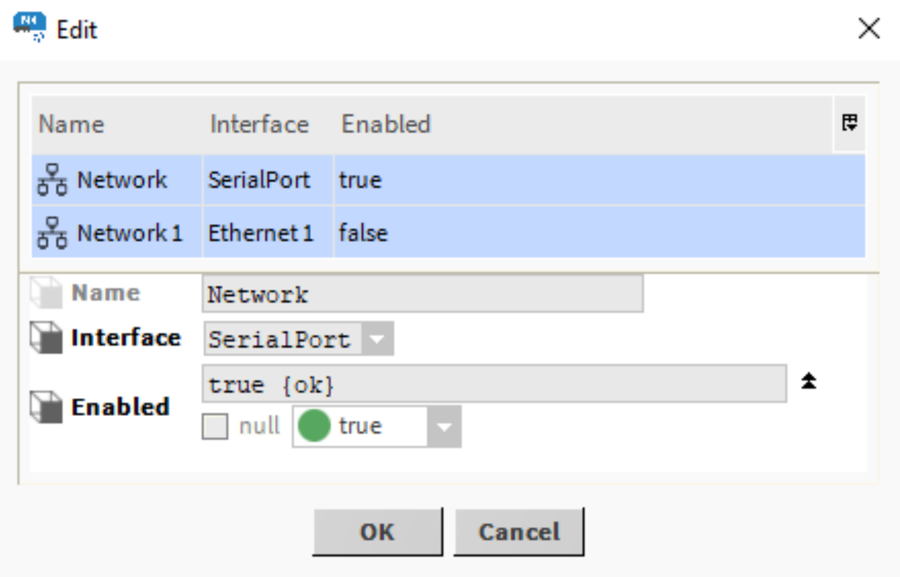
Enabled: true {ok}

□ null ● true

Figure 54. Editing pop-up

Note

Editing is possible for more than one network at a time. If multiple networks are edited, the same new value is written to common slots, so individual slots, such as Name, cannot be edited in this manner.



Name	Interface	Enabled
Network	SerialPort	true
Network 1	Ethernet 1	false

Name: Network

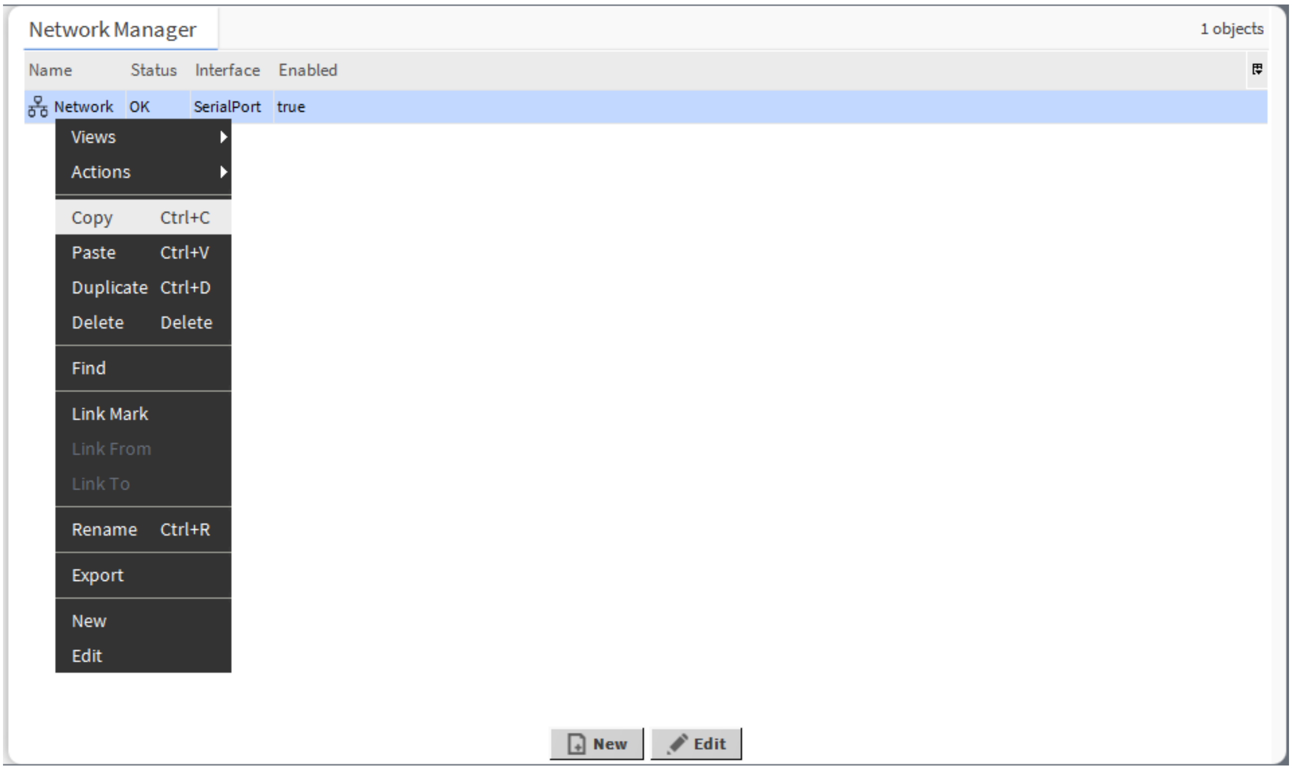
Interface: SerialPort

Enabled: true {ok}

□ null ● true

Figure 55. Editing multiple Modbus networks

- copy/duplicate/remove Modbus network components:



Name	Status	Interface	Enabled
Network	OK	SerialPort	true

Figure 56. Context menu options for Modbus network

Opening Modbus Network Manager

The Modbus Network Manager view is accessible from the context menu of the Modbus component. It is also automatically opened if the Modbus component is double-clicked in the nav tree window.

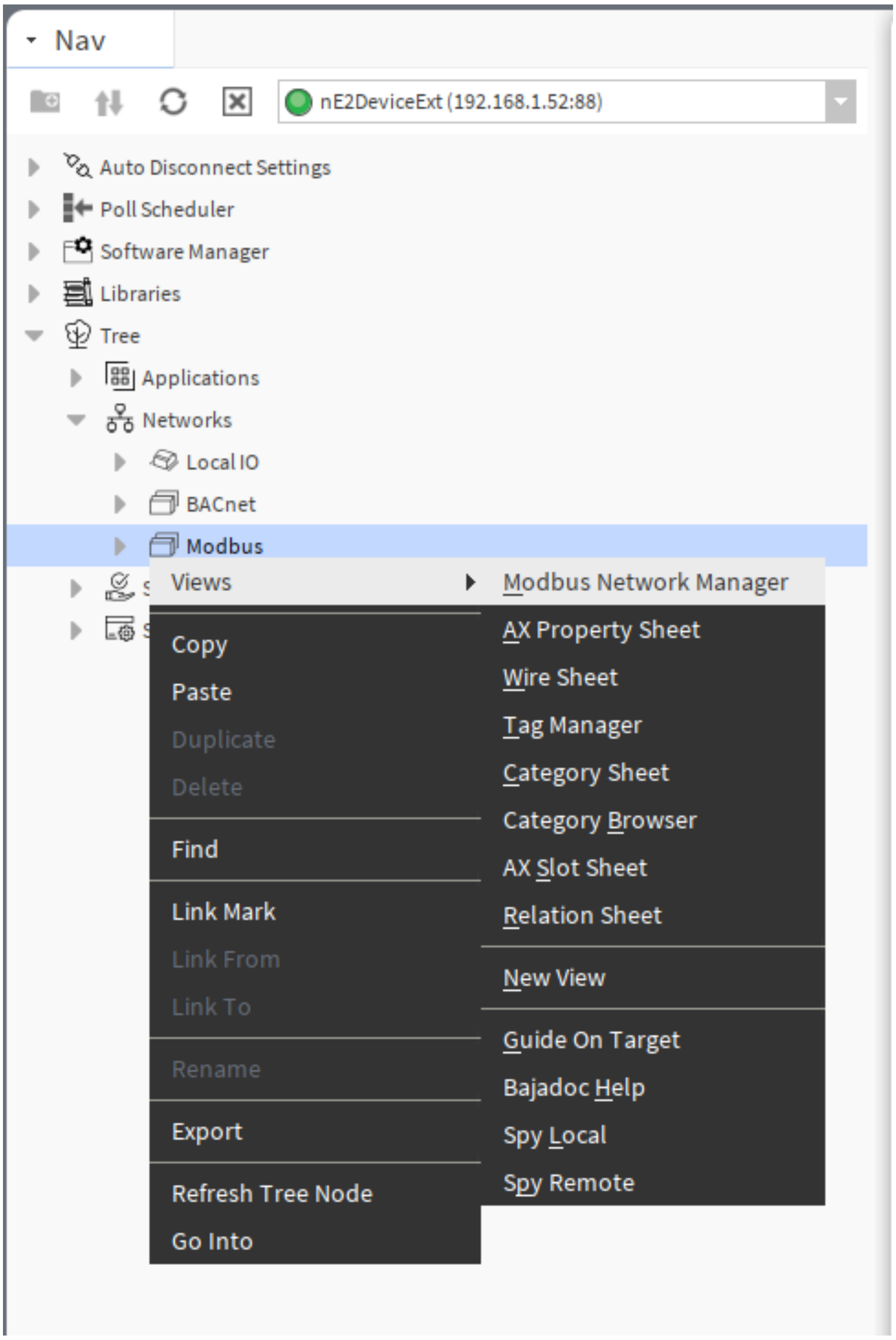
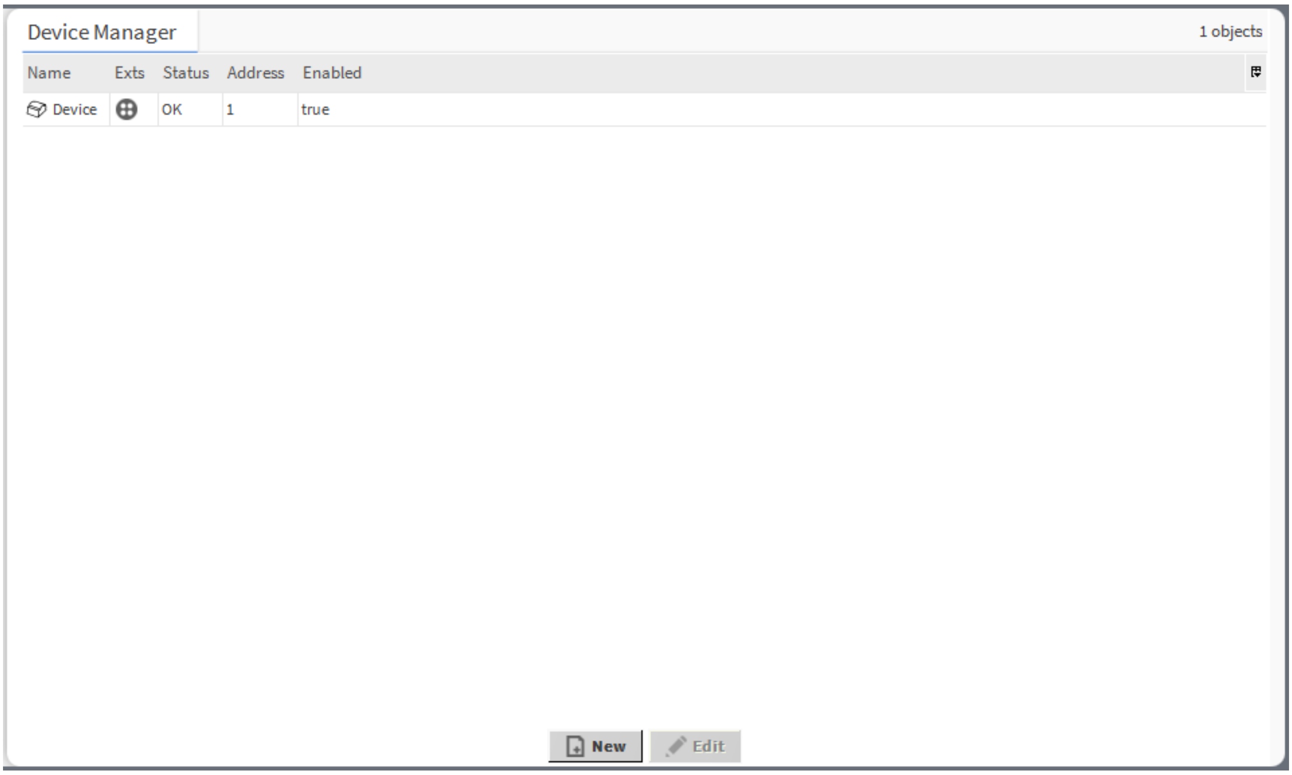


Figure 57. Accessing the Modbus Network Manager

4.4.6 Modbus Device Manager

The Device Manager view is available for the Modbus Network component. It lists all Modbus devices added to the network. The Device Manager view shows the names, statues, device names and ID, and enabled or disabled states of the device. The special Exts columns provides quick access to the Point Manager (with all points of the relevant device).



Device Manager

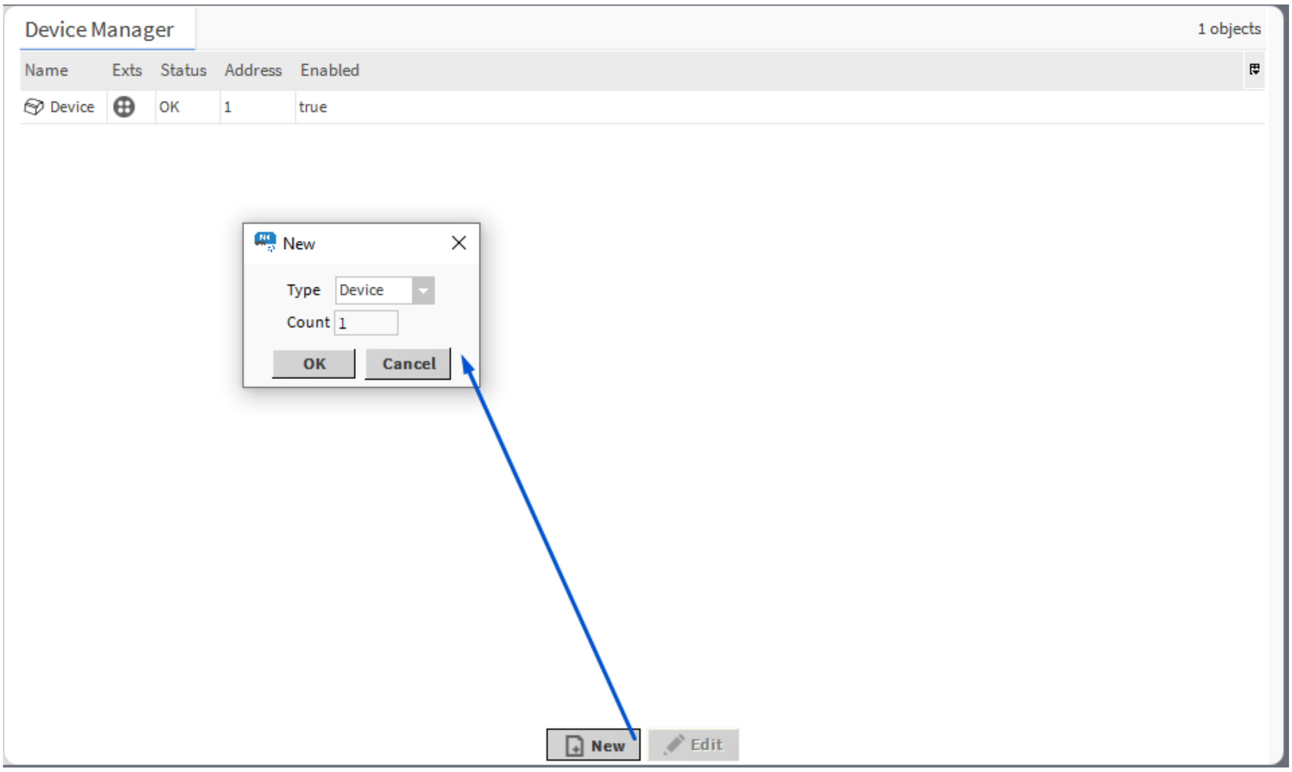
1 objects

Name	Exts	Status	Address	Enabled
Device		OK	1	true

Figure 58. Modbus Device Manager

In the Modbus Device Manager, it is possible to:

- add Modbus Device component:



Device Manager

1 objects

Name	Exts	Status	Address	Enabled
Device		OK	1	true

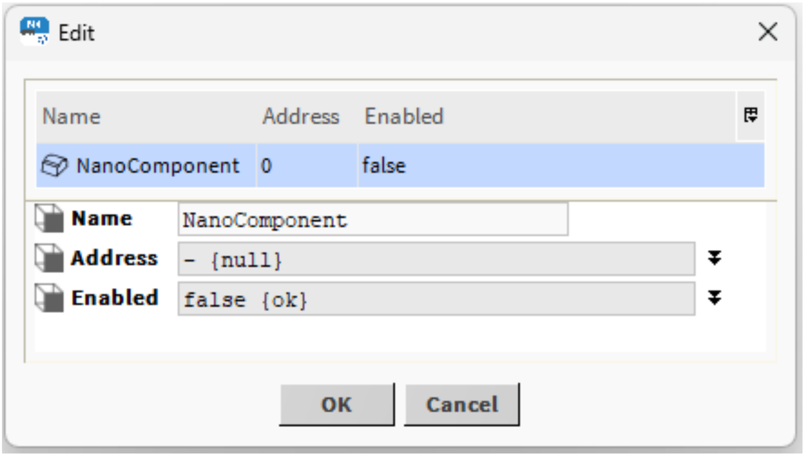
New

Type: Device

Count: 1

Figure 59. Adding new Modbus device

- edit the Modbus device's name:



Edit

Name	Address	Enabled
NanoComponent	0	false

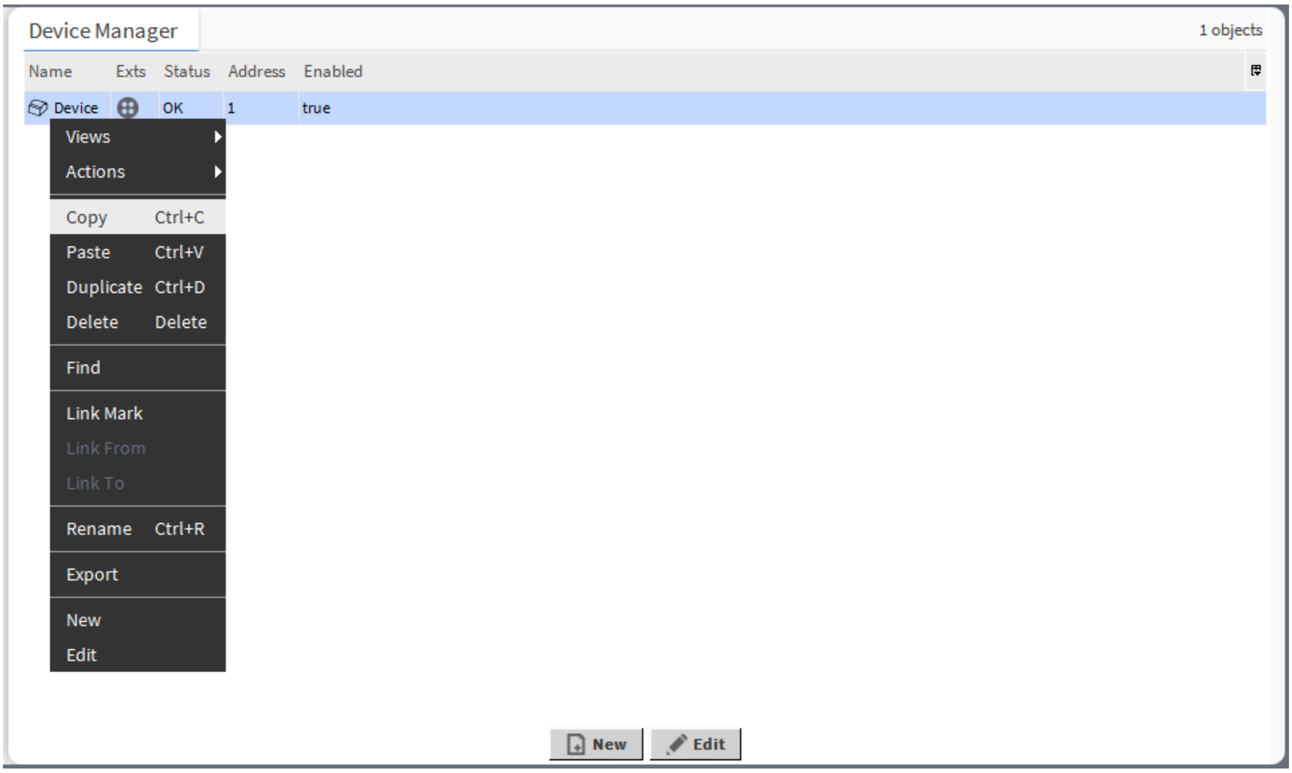
Name: NanoComponent

Address: - {null}

Enabled: false {ok}

Figure 60. Editing pop-up

- copy/duplicate/remove Modbus device components:



Device Manager — 1 objects

Name	Exts	Status	Address	Enabled
Device		OK	1	true

Figure 61. Context menu options for Modbus device

Opening Modbus Device Manager

The Modbus Device Manager view is accessible from the context menu of the Modbus Network component. It is also automatically opened if the Modbus Network component is double-clicked in the nav tree window.

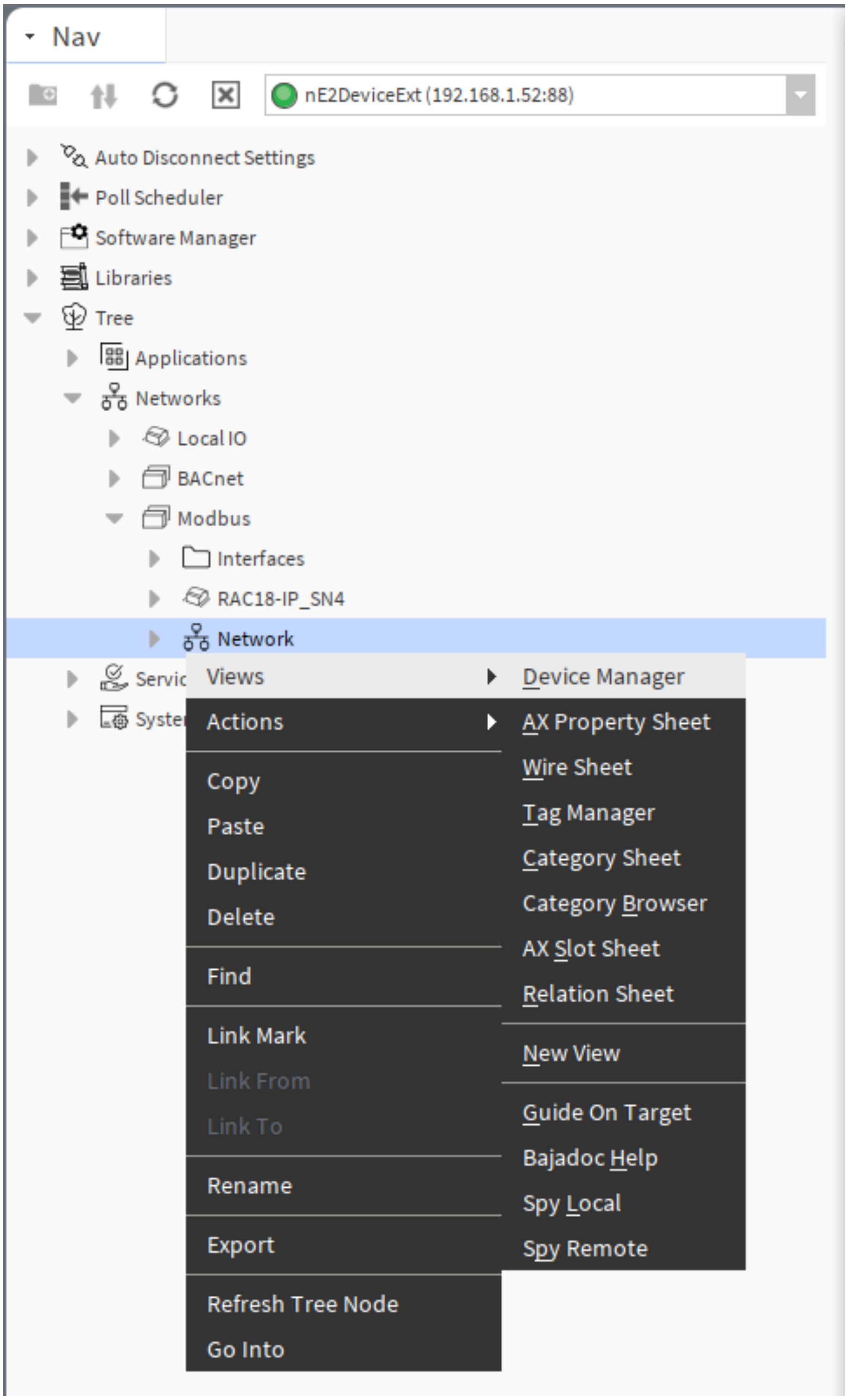
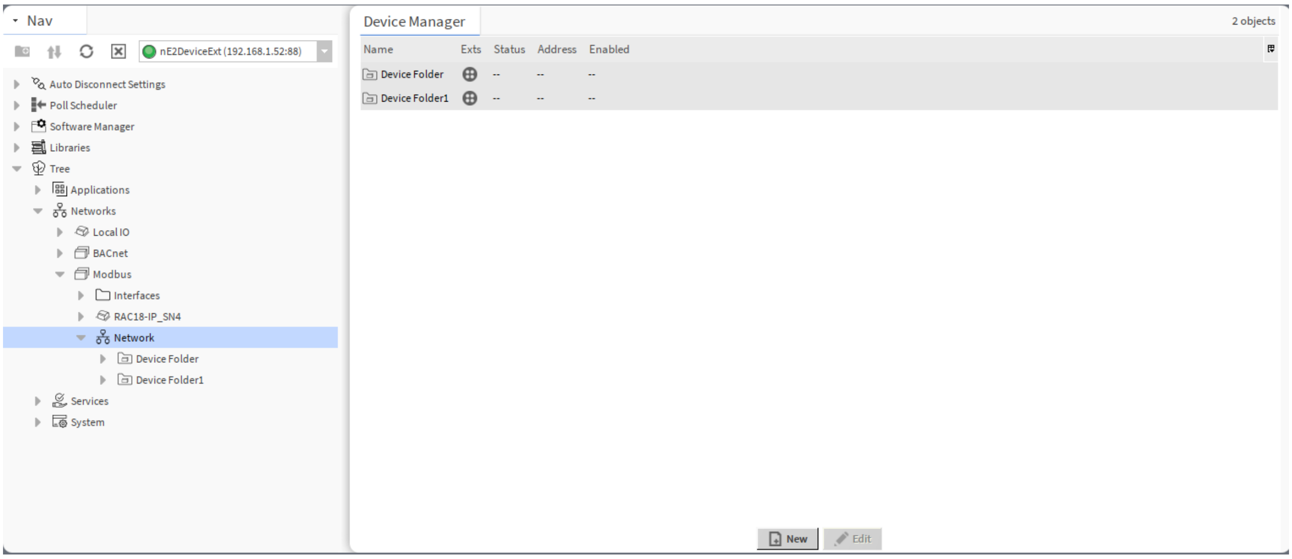


Figure 62. Accessing the Modbus Device Manager

DeviceFolder

The DeviceFolder component is a device grouping component, which can be added under the Modbus network. It allows to organize devices in the network on the nav tree and maintains the grouping in the Device Manager view. It is clickable in the Device Manager allowing to display only devices grouped in one DeviceFolder component.



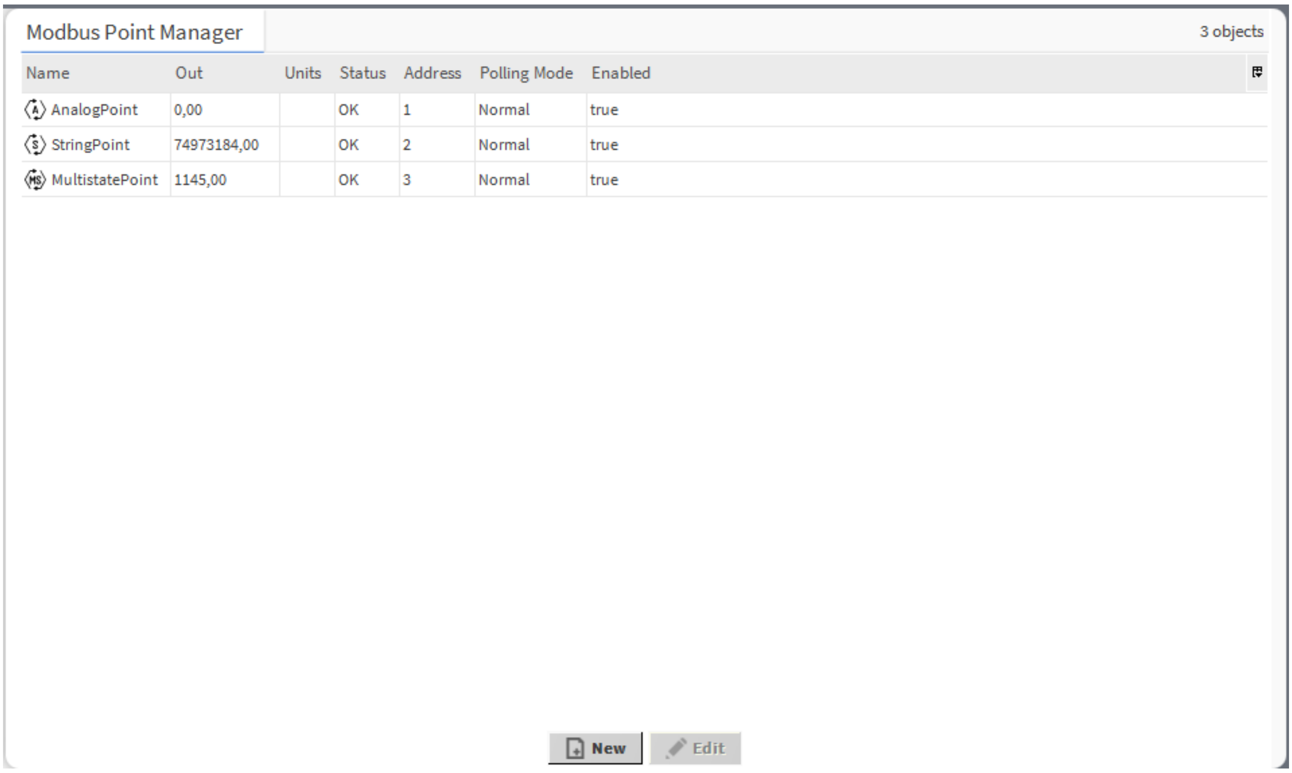
Name	Exts	Status	Address	Enabled
Device Folder	⊕	--	--	--
Device Folder1	⊕	--	--	--

Figure 63. DeviceFolder

The DeviceFolder component has no slots nor actions. The DeviceFolder is available in the Core library.

4.4.7 Modbus Point Manager

The Modbus Point Manager view is available for each device added to the Modbus network. It lists all Modbus points added to the Device component, and shows their Out slot value, status, object name and ID, polling mode, and enabled or disabled state.

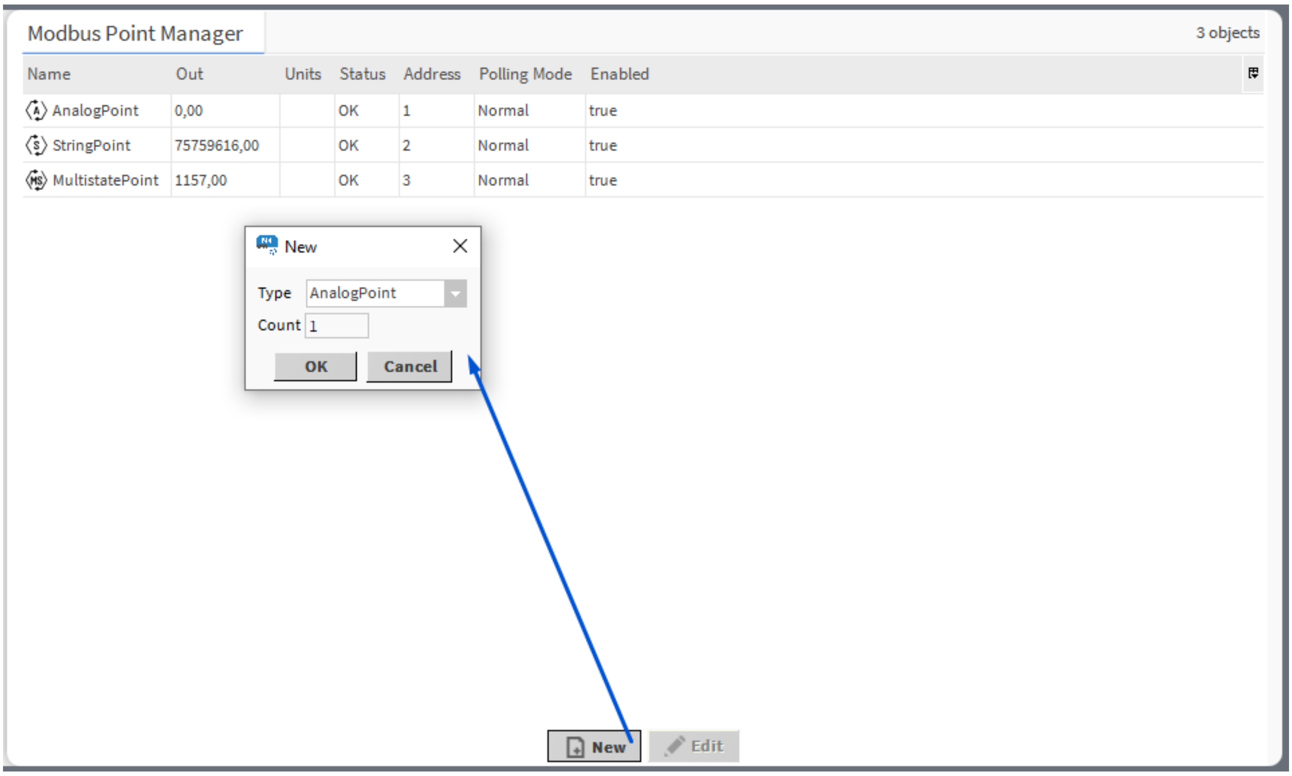


Name	Out	Units	Status	Address	Polling Mode	Enabled
AnalogPoint	0,00		OK	1	Normal	true
StringPoint	74973184,00		OK	2	Normal	true
MultistatePoint	1145,00		OK	3	Normal	true

Figure 64. Modbus Point Manager

In the Modbus Point Manager, it is possible to:

- add Modbus points:



Modbus Point Manager

3 objects

Name	Out	Units	Status	Address	Polling Mode	Enabled
AnalogPoint	0,00		OK	1	Normal	true
StringPoint	75759616,00		OK	2	Normal	true
MultistatePoint	1157,00		OK	3	Normal	true

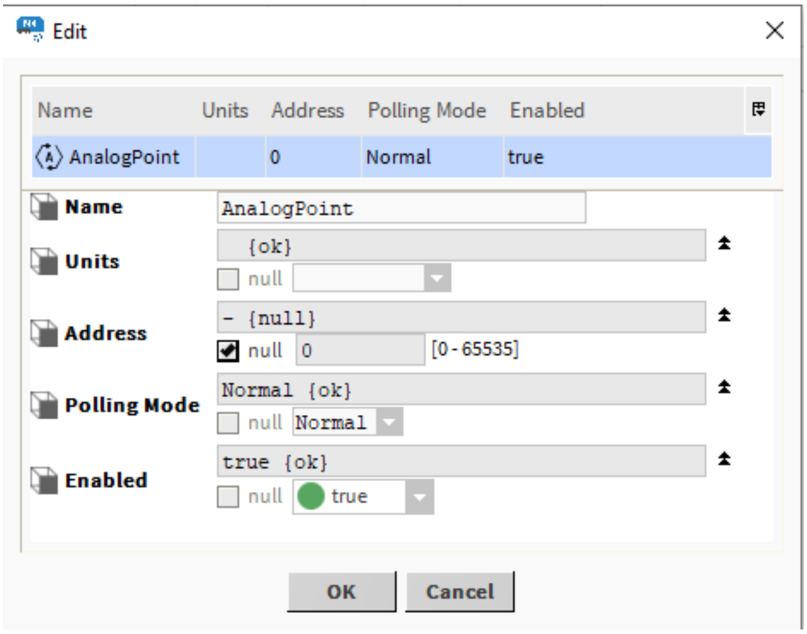
New

Type: AnalogPoint

Count: 1

Figure 65. Adding new Modbus point

- edit the Modbus point’s name, units, address and enable/disable the component:



Edit

Name	Units	Address	Polling Mode	Enabled
AnalogPoint		0	Normal	true

Name: AnalogPoint

Units: {ok}
□ null

Address: - {null}
☑ null 0 [0 - 65535]

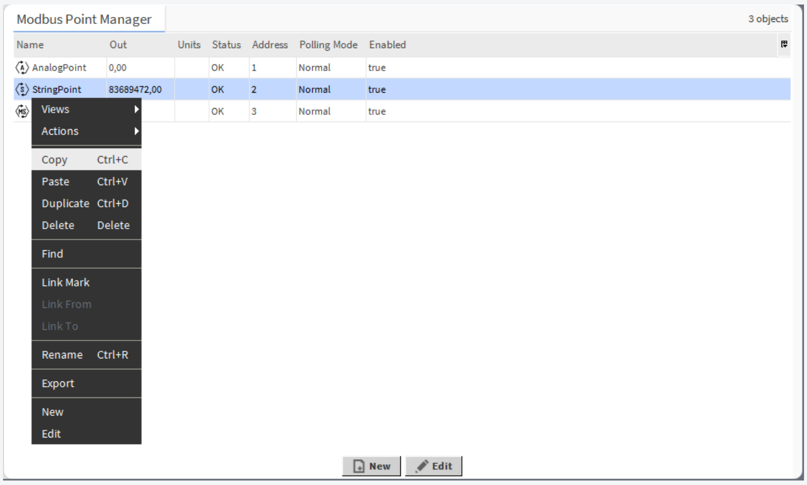
Polling Mode: Normal {ok}
□ null Normal

Enabled: true {ok}
□ null ● true

Figure 66. Editing pop-up

Note

Editing is possible for more than one point at a time. If multiple points are edited, the same new value is written to common slots, so individual slots, such as Name, cannot be edited in this manner.



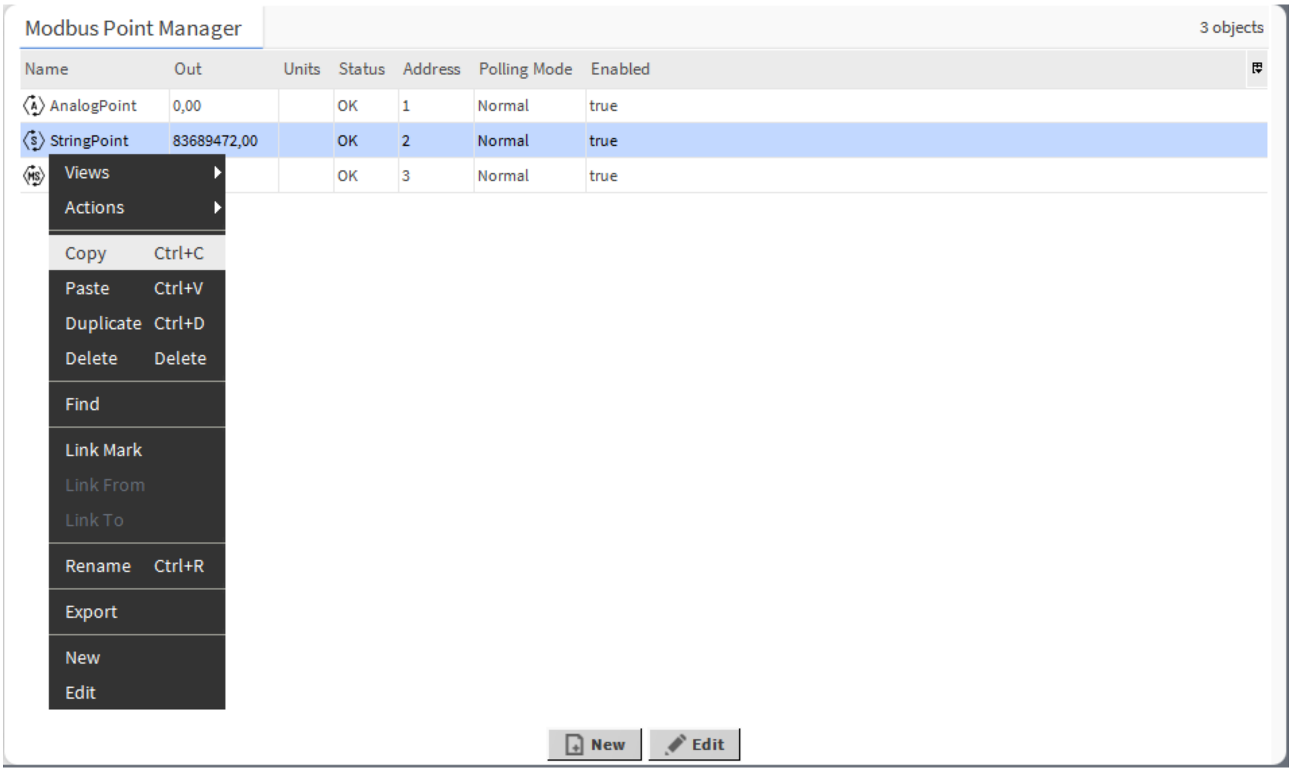
Modbus Point Manager

3 objects

Name	Out	Units	Status	Address	Polling Mode	Enabled
AnalogPoint	0,00		OK	1	Normal	true
StringPoint	83689472,00		OK	2	Normal	true
			OK	3	Normal	true

Figure 67. Editing multiple Modbus points

- copy/duplicate/remove Modbus points:



Modbus Point Manager

3 objects

Name	Out	Units	Status	Address	Polling Mode	Enabled
AnalogPoint	0,00		OK	1	Normal	true
StringPoint	83689472,00		OK	2	Normal	true
			OK	3	Normal	true

Figure 68. Context menu for Modbus points

Opening the Modbus Point Manager

The Modbus Point Manager view is accessible from the context menu of the Device component. It is also automatically opened if the Device component is double-clicked in the nav tree window.

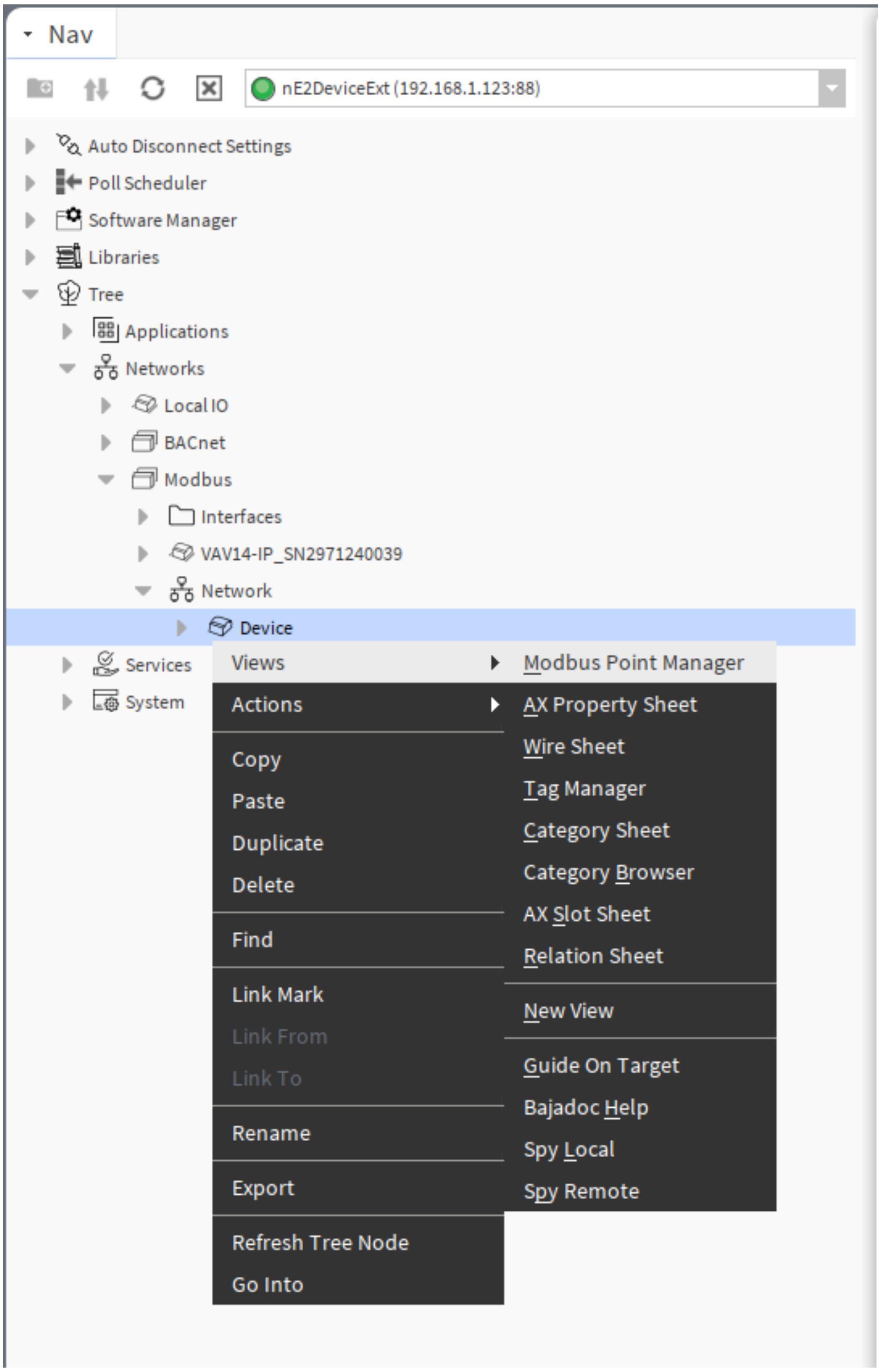
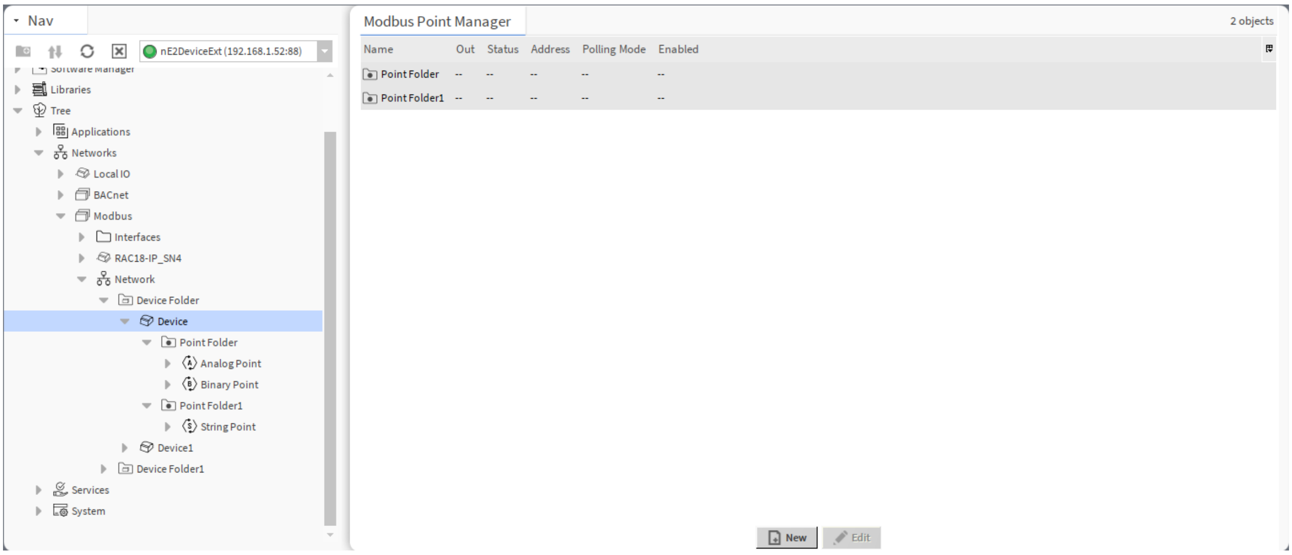


Figure 69. Accessing the Modbus Point Manager from the context menu

PointFolder

The PointFolder component is a point grouping component, which can be added under the Modbus device. It allows to organize points of the device on the nav tree and maintains the grouping in the Point Manager view. It is clickable in the Point Manager allowing to display only points grouped in one PointFolder component.



Modbus Point Manager

2 objects

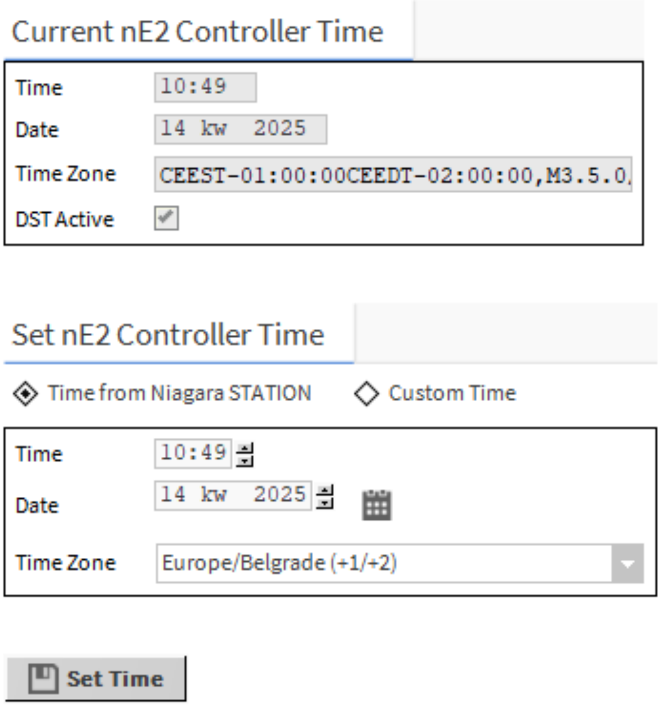
Name	Out	Status	Address	Polling Mode	Enabled
Point Folder	--	--	--	--	--
Point Folder1	--	--	--	--	--

Figure 70. PointFolder

The PointFolder component has no slots nor actions. The PointFolder is available in the Core library.

4.5 Time Settings

The time setting function allows users to configure the time settings of the controller directly from the Niagara station or to set a custom time. This feature is a part of the Platform component.



Current nE2 Controller Time

Time	10:49
Date	14 kw 2025
Time Zone	CEEST-01:00:00CEEDT-02:00:00,M3.5.0,
DST Active	☑

Set nE2 Controller Time

◈ Time from Niagara STATION ◇ Custom Time

Time	10:49
Date	14 kw 2025
Time Zone	Europe/Belgrade (+1/+2)

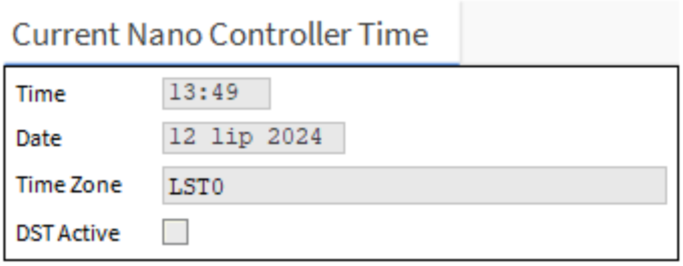
Figure 71. Time settings

To set the time:

- navigate to System>Platform;
- double-click the Time component.

The dialog window will display the following:

- **Current Nano Controller Time:** shows the currently set time, date, and time zone as well as indicates whether the Daylight Saving Time is currently active;

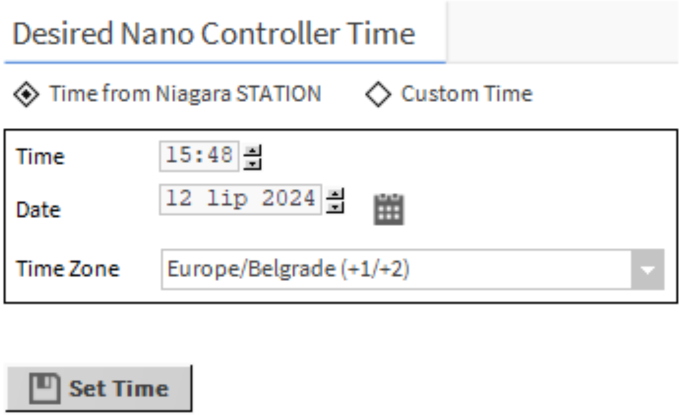


Current Nano Controller Time

Time	13:49
Date	12 lip 2024
Time Zone	LST0
DST Active	□

Figure 72. Current nano EDGE ENGINE device time

- **Desired Nano Controller Time:** allows the user to set the time on the controller directly from the Niagara station or to set a custom time.



Desired Nano Controller Time

◈ Time from Niagara STATION ◇ Custom Time

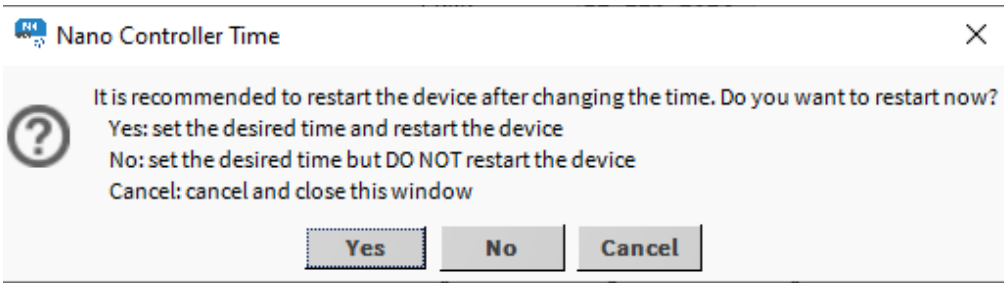
Time	15:48
Date	12 lip 2024
Time Zone	Europe/Belgrade (+1/+2)

To set the nano EDGE ENGINE device time based on the Niagara station time:

- confirm that the Time from Niagara Station option is selected;

> In this configuration, the displayed time, date, and time zone are in read-only mode.

- click Set Time to configure the time on the nano EDGE ENGINE device as in the station;
- a pop-up asking to restart the device will be displayed;
- click Yes to confirm, the device will be restarted;



Nano Controller Time

It is recommended to restart the device after changing the time. Do you want to restart now?
Yes: set the desired time and restart the device
No: set the desired time but DO NOT restart the device
Cancel: cancel and close this window

Figure 73. Set the time from the station dialog window

- reconnect with the device using the Connect action in the nE2DeviceExt.

To set a custom time in the controller:

- select the Custom Time option;
- the Time Setting dialog window can now be edited: set the time, date, and time zone;

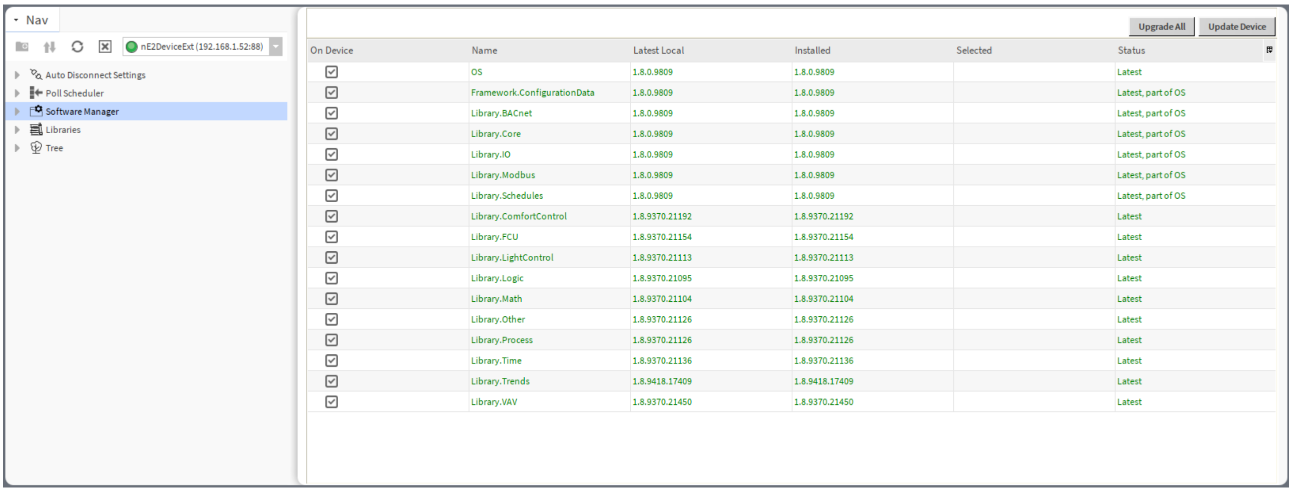
- click Set Time to confirm;
- a pop-up asking to restart the device will be displayed, click Yes to confirm. The device will be restarted;
- reconnect with the device using the Connect action in the nE2DeviceExt.

To learn more about time settings and configurations, please refer to the nano EDGE ENGINE Programming user manual.

4.6 Software Manager

The Software Manager is synchronized with the Workbench or the Niagara controller shared folder of the station. By default, the Software Manager displays the default nano EDGE ENGINE libraries supported by the version of the module. All third party libraries must be added to the shared folder.

To navigate, double click the Software Manager component in the device tree.



On Device	Name	Latest Local	Installed	Selected	Status
☑	OS	1.8.0.9809	1.8.0.9809		Latest
☑	Framework.ConfigurationData	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.BACnet	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.Core	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.IO	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.Modbus	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.Schedules	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.ComfortControl	1.8.9370.21192	1.8.9370.21192		Latest
☑	Library.FCU	1.8.9370.21154	1.8.9370.21154		Latest
☑	Library.LightControl	1.8.9370.21113	1.8.9370.21113		Latest
☑	Library.Logic	1.8.9370.21095	1.8.9370.21095		Latest
☑	Library.Math	1.8.9370.21104	1.8.9370.21104		Latest
☑	Library.Other	1.8.9370.21126	1.8.9370.21126		Latest
☑	Library.Process	1.8.9370.21126	1.8.9370.21126		Latest
☑	Library.Time	1.8.9370.21136	1.8.9370.21136		Latest
☑	Library.Trends	1.8.9418.17409	1.8.9418.17409		Latest
☑	Library.VAV	1.8.9370.21450	1.8.9370.21450		Latest

Figure 74. Software Manager view

4.6.1 Using Software Manager

The Software Manager view lists the OS and libraries available locally. The view highlights each row (OS or libraries) according to its status:

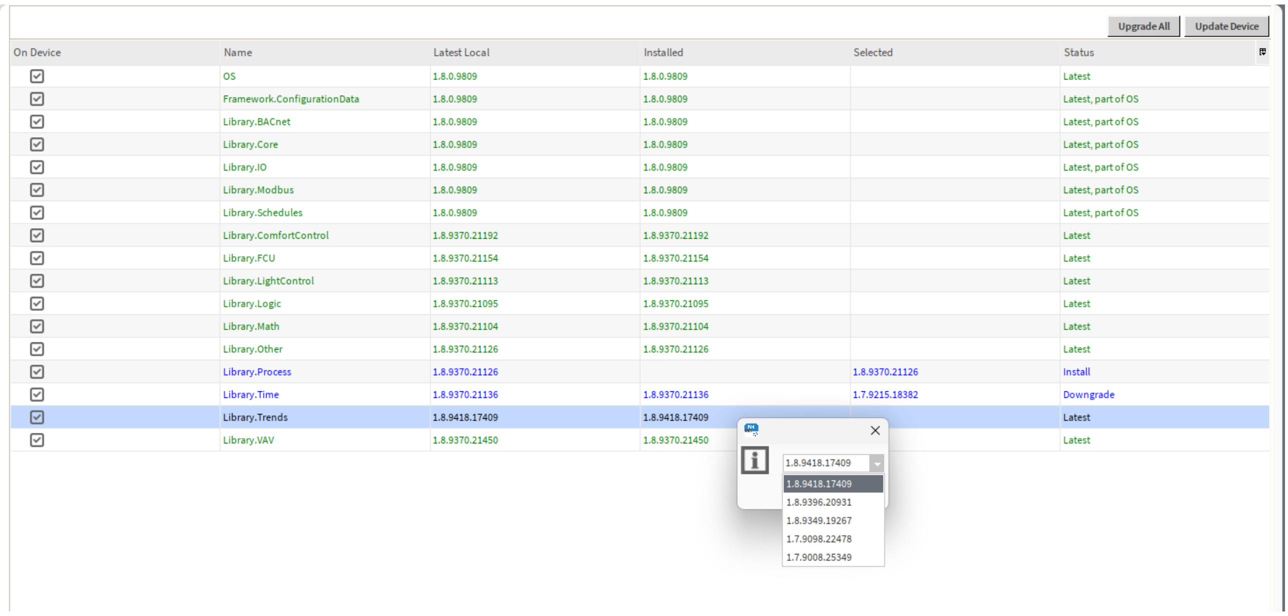
- green: the element is up to date, and requires no action;
- orange: the element is out of date, and can be updated;
- blue: the action is about to be taken on the element;
- red: the library could not be loaded (the library installed on the device has a version, which is not supported by the OS).

The Software Manager table contains the following columns:

- On Device: indicates, whether a given element is already installed on the device.
- Name: shows the name of the element.
- Latest Local: shows the latest version available locally to be installed on the device.
- Installed: shows the version of the element installed on the device.
- Selected: opens a drop-down list with all versions available locally for a selected element.
- Status: indicates, which action is to be performed on the element, once a specific version has been selected in the Action column.

  - Available information: Latest, Out of Date, Upgrade, Downgrade, Install, Uninstall, none (the selected version is the same as the one installed on the device).

In order to upgrade or downgrade the selected element, choose the desired version of the element in the Selected column, and press the Update Device option (highlighted in blue in the right upper corner of the Software Manager). This option executes all actions indicated in the Status column.



On Device	Name	Latest Local	Installed	Selected	Status
☑	OS	1.8.0.9809	1.8.0.9809		Latest
☑	Framework.ConfigurationData	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.BACnet	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.Core	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.IO	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.Modbus	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.Schedules	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.ComfortControl	1.8.9370.21192	1.8.9370.21192		Latest
☑	Library.FCU	1.8.9370.21154	1.8.9370.21154		Latest
☑	Library.LightControl	1.8.9370.21113	1.8.9370.21113		Latest
☑	Library.Logic	1.8.9370.21095	1.8.9370.21095		Latest
☑	Library.Math	1.8.9370.21104	1.8.9370.21104		Latest
☑	Library.Other	1.8.9370.21126	1.8.9370.21126		Latest
☑	Library.Process	1.8.9370.21126		1.8.9370.21126	Install
☑	Library.Time	1.8.9370.21136	1.8.9370.21136	1.7.9215.18382	Downgrade
☑	Library.Trends	1.8.9418.17409	1.8.9418.17409		Latest
☑	Library.VAV	1.8.9370.21450	1.8.9370.21450		Latest

Figure 75. Selecting a library's version

Unless the user intends to manually select the versions to be installed, there is also the option to automatically select all newest versions for all out of date elements using the Upgrade All button.

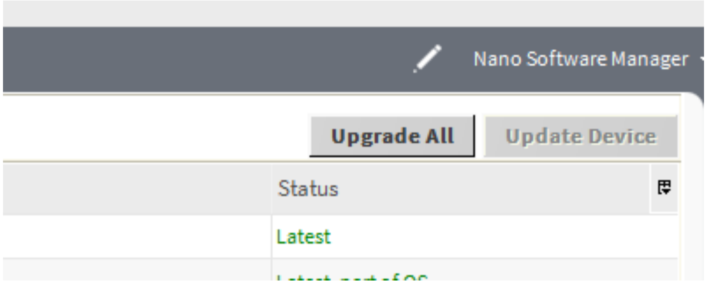


Figure 76. Upgrade All button

Regardless of the fact whether the user intends to add or remove the library available in the Software Manager, each operation requires performing three steps:

- check the box in the On Device column to install the library, or uncheck it to uninstall the library;
- provided the library is to be installed on the device, check its preferred version in the Selected column–by default, the newest version available locally is indicated to be installed;
- once selection of all libraries to be installed or uninstalled on the device is complete, hit the Update Device command.

Warning!
The OS cannot be removed from the device; it is preinstalled on the device's SD card, and the only operations, which can be performed on this element, are upgrading or downgrading it.

4.6.2 Uploading New Libraries

With the nE2Link, it is possible to upload libraries and/or OS files to the controller, which were not pre-loaded in the nE2DeviceExt. New libraries have to be saved on a local PC and then copied to the nanoEdgeEngine folder on the station:

- save new libraries in a folder selected location on the local PC;
- copy the proper nano EDGE ENGINE libraries and/or OS files from the local PC by navigating to the proper location on My Host (local PC); (it is possible to copy the whole folder with proper contents but please make sure then that the folder is named 'nanoEdgeEngine');

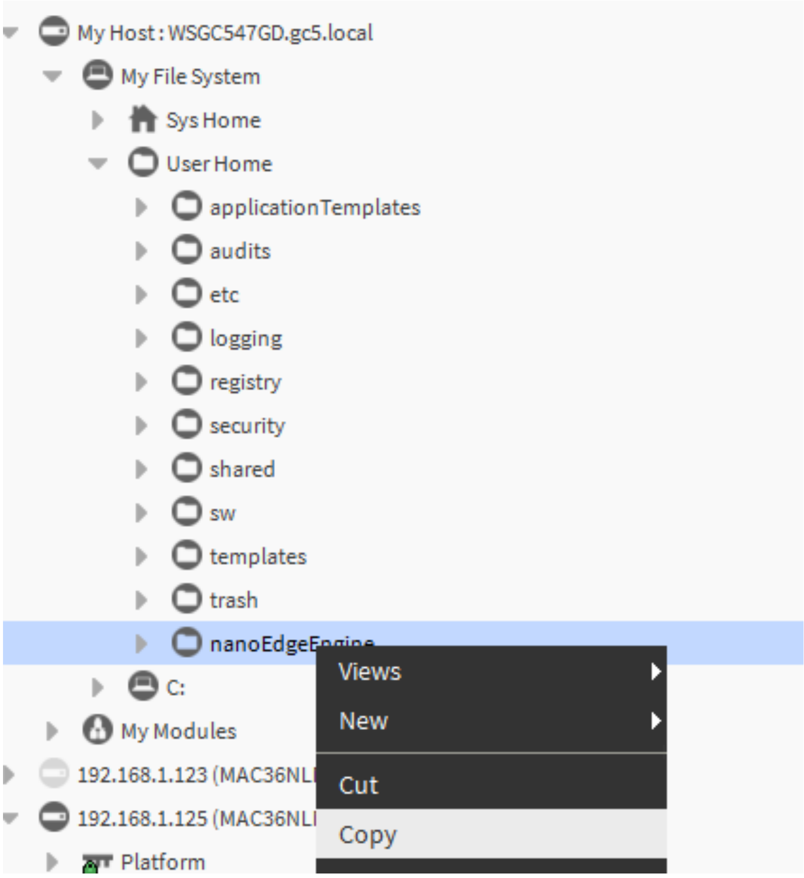


Figure 77. Copying the nanoEdgeEngine folder

- navigate to Station → Files → nE2Link →nanoEdgeEngine folder;

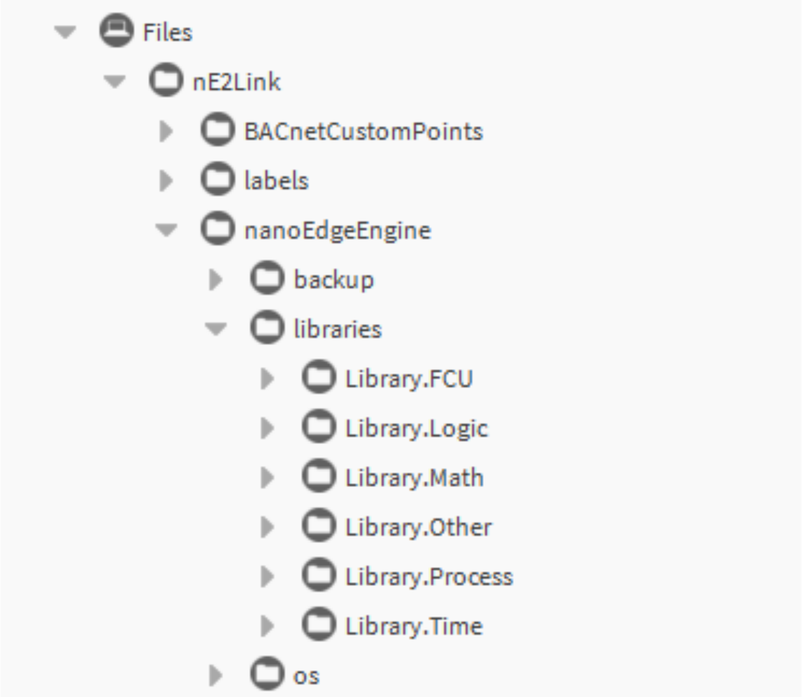


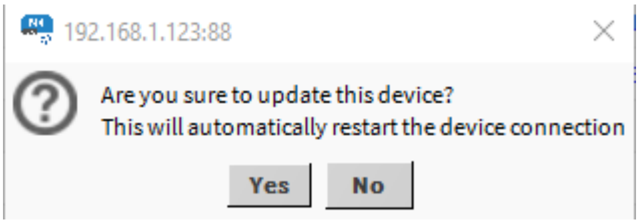
Figure 78. Pasted folder

- paste it under the Files container in the local station.

Note: Please make sure to preserve a proper structure of subfolders: new libraries files have to be stored in the libraries subfolder, OS files in the os subfolder, and backup files in the backup subfolder.

Once the libraries are added, they will become visible in the Software Manager.

- Select the libraries and OS version to be installed, upgraded or downgraded on the controller, or select the Upgrade all option;
- once all necessary software is selected, click Update device;
- a pop-up will be shown asking to confirm the action. Click Yes to load the new OS and libraries.

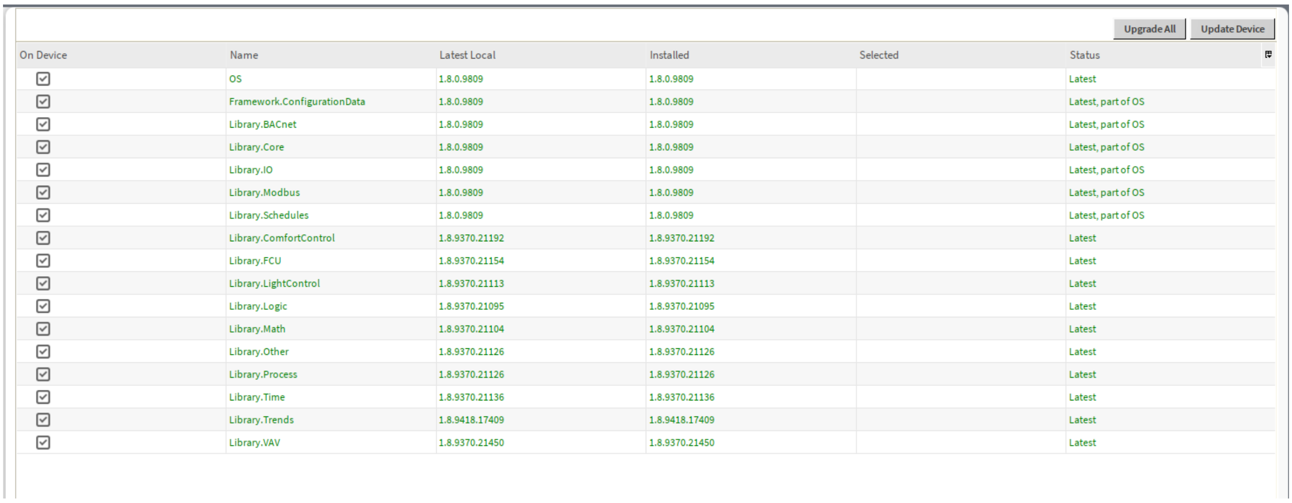

192.168.1.123:88

Are you sure to update this device?
This will automatically restart the device connection

Figure 79. Update device prompt

The device will be restarted automatically.

- Once the device has restarted successfully, right-click on the nE2DeviceExt and connect to the device;
- after reconnection, confirm that selected software was successfully installed on the device.

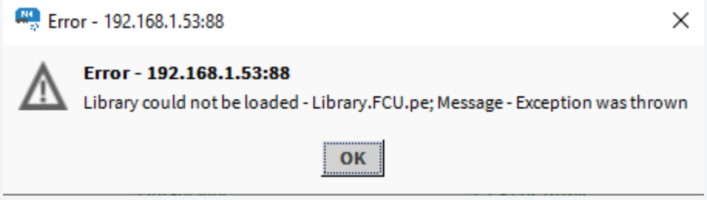


On Device	Name	Latest Local	Installed	Selected	Status
☑	OS	1.8.0.9809	1.8.0.9809		Latest
☑	Framework.ConfigurationData	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.BACnet	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.Core	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.IO	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.Modbus	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.Schedules	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.ComfortControl	1.8.9370.21192	1.8.9370.21192		Latest
☑	Library.FCU	1.8.9370.21154	1.8.9370.21154		Latest
☑	Library.LightControl	1.8.9370.21113	1.8.9370.21113		Latest
☑	Library.Logic	1.8.9370.21095	1.8.9370.21095		Latest
☑	Library.Math	1.8.9370.21104	1.8.9370.21104		Latest
☑	Library.Other	1.8.9370.21126	1.8.9370.21126		Latest
☑	Library.Process	1.8.9370.21126	1.8.9370.21126		Latest
☑	Library.Time	1.8.9370.21136	1.8.9370.21136		Latest
☑	Library.Trends	1.8.9418.17409	1.8.9418.17409		Latest
☑	Library.VAV	1.8.9370.21450	1.8.9370.21450		Latest

Figure 80. Updated Software Manager view

Library Not Loaded

The Software Manager has a mechanism informing the user about the libraries compatibility. For a proper operation, libraries versions have to be supported by the OS. If the library installed on the device has a version, which is not supported by the OS, Software Manager displays an error prompt and marks the library as 'not loaded'. In such case, it is required to upgrade the library.

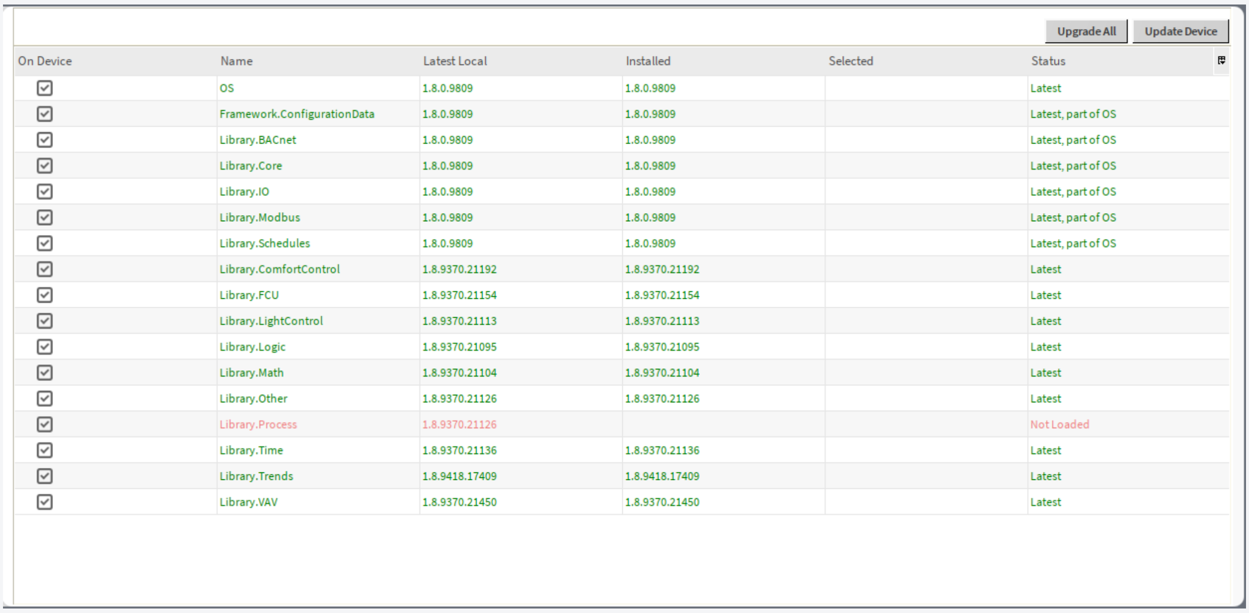


Error - 192.168.1.53:88

Error - 192.168.1.53:88

Library could not be loaded - Library.FCU.pe; Message - Exception was thrown

Figure 81. Error prompt

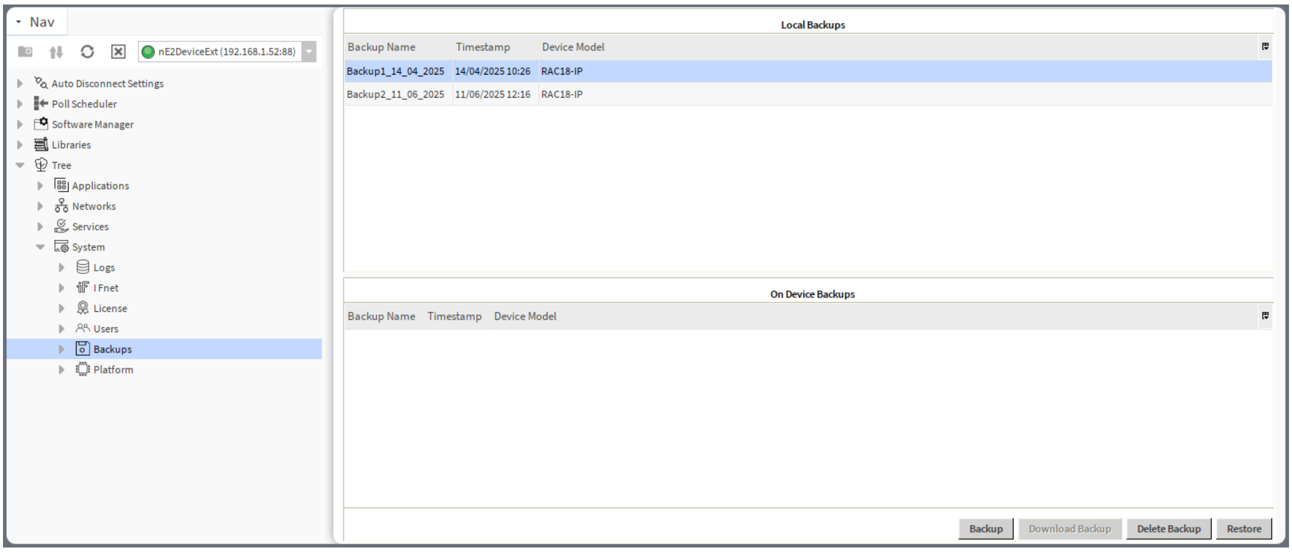


On Device	Name	Latest Local	Installed	Selected	Status
☑	OS	1.8.0.9809	1.8.0.9809		Latest
☑	Framework.ConfigurationData	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.BACnet	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.Core	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.IO	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.Modbus	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.Schedules	1.8.0.9809	1.8.0.9809		Latest, part of OS
☑	Library.ComfortControl	1.8.9370.21192	1.8.9370.21192		Latest
☑	Library.FCU	1.8.9370.21154	1.8.9370.21154		Latest
☑	Library.LightControl	1.8.9370.21113	1.8.9370.21113		Latest
☑	Library.Logic	1.8.9370.21095	1.8.9370.21095		Latest
☑	Library.Math	1.8.9370.21104	1.8.9370.21104		Latest
☑	Library.Other	1.8.9370.21126	1.8.9370.21126		Latest
☑	Library.Process	1.8.9370.21126			Not Loaded
☑	Library.Time	1.8.9370.21136	1.8.9370.21136		Latest
☑	Library.Trends	1.8.9418.17409	1.8.9418.17409		Latest
☑	Library.VAV	1.8.9370.21450	1.8.9370.21450		Latest

Figure 82. Library not loaded

4.7 Backups

nE2DeviceExt allows users to invoke the device's backup function. The local backups are saved directly into the station's shared folder.



Local Backups

Backup Name	Timestamp	Device Model
Backup1_14_04_2025	14/04/2025 10:26	RAC18-IP
Backup2_11_06_2025	11/06/2025 12:16	RAC18-IP

On Device Backups

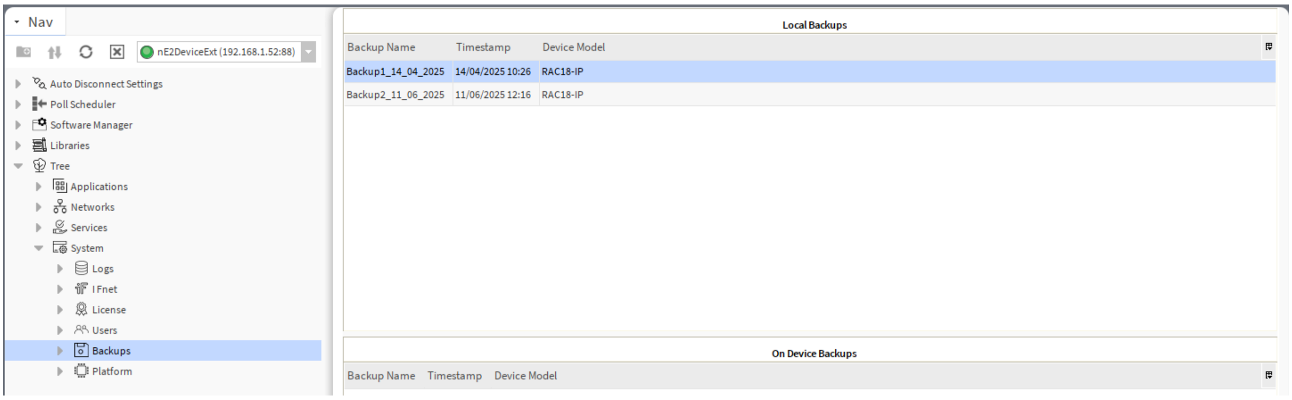
Backup Name	Timestamp	Device Model

Figure 83. Backup Manager view

To perform a backup, go to the Backup Manager of the device, in the Backups component in the System container. The backup can be restored to the device.

Note: nano EDGE ENGINE controllers allow to store one backup directly in the local device memory. Backups can be downloaded and stored in the Station Files folder.

All local backups are stored in the local Niagara station. To access backups go to Station → Files → nE2Link→ nanoEdgeEngine ->backup → *BackupName*. Backups can be imported or exported from this location manually and will become visible in the Backup Manager view.



Local Backups

Backup Name	Timestamp	Device Model
Backup1_14_04_2025	14/04/2025 10:26	RAC18-IP
Backup2_11_06_2025	11/06/2025 12:16	RAC18-IP

On Device Backups

Backup Name	Timestamp	Device Model

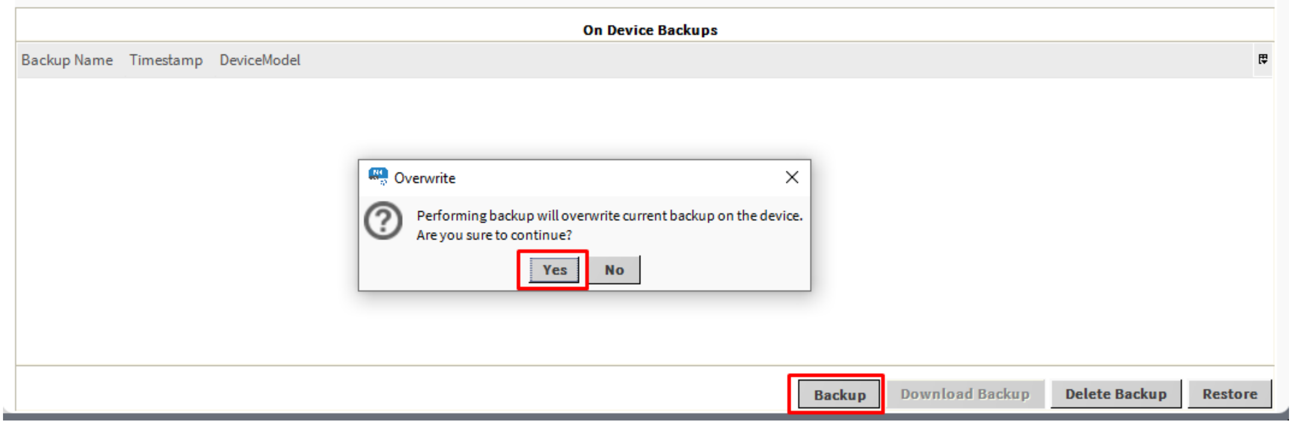
Figure 84. Local station backup stored in the station's Files

To learn more about the Backups, refer to the nano EDGE ENGINE Programming user manual.

4.7.1 Performing Backup

- Click the Backup button to invoke creating a backup.

Warning!
If there is any existing backup on the device, performing the backup action will overwrite it.



On Device Backups

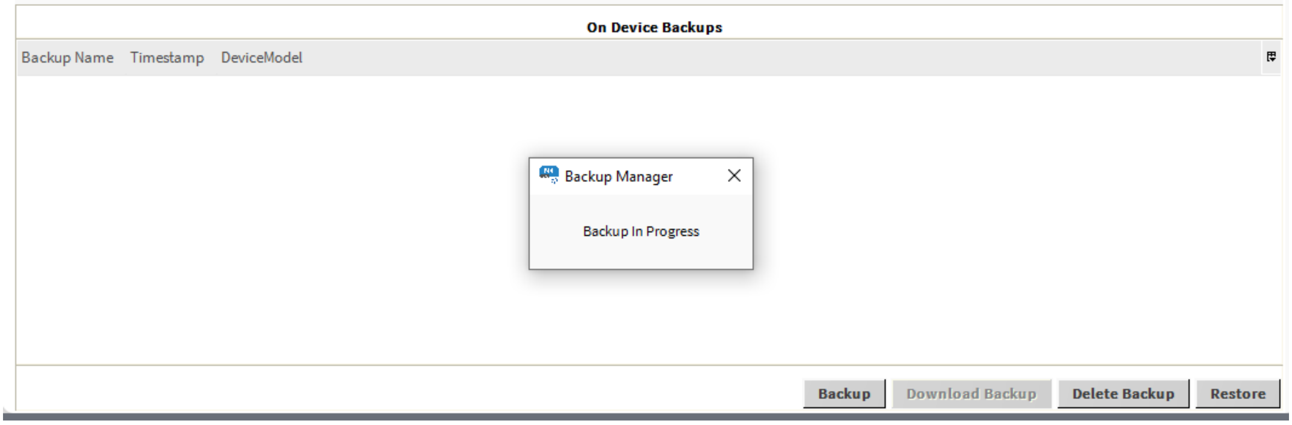
Backup Name | Timestamp | DeviceModel

Overwrite

Performing backup will overwrite current backup on the device. Are you sure to continue?

Figure 85. Pop-up informing about the risk of overwriting the existing backup on the device

Once the backup action is confirmed, the device will perform the backup. This process can take up to a few minutes. Wait for the process to finish.



On Device Backups

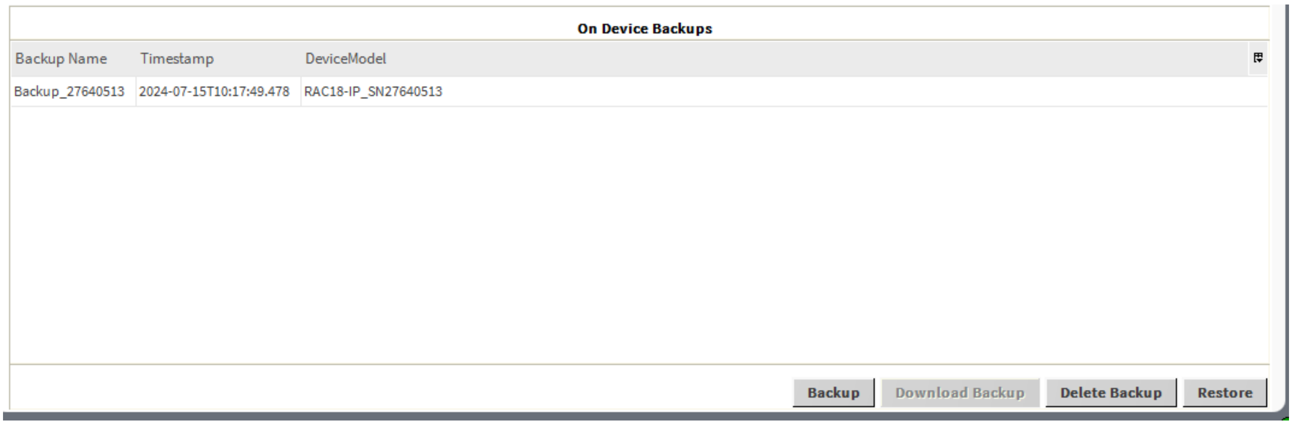
Backup Name | Timestamp | DeviceModel

Backup Manager

Backup In Progress

Figure 86. Pop-up informing about the backup progress

Once the process is completed, the backup will be visible in the On Device Backups table in the Backup Manager view.



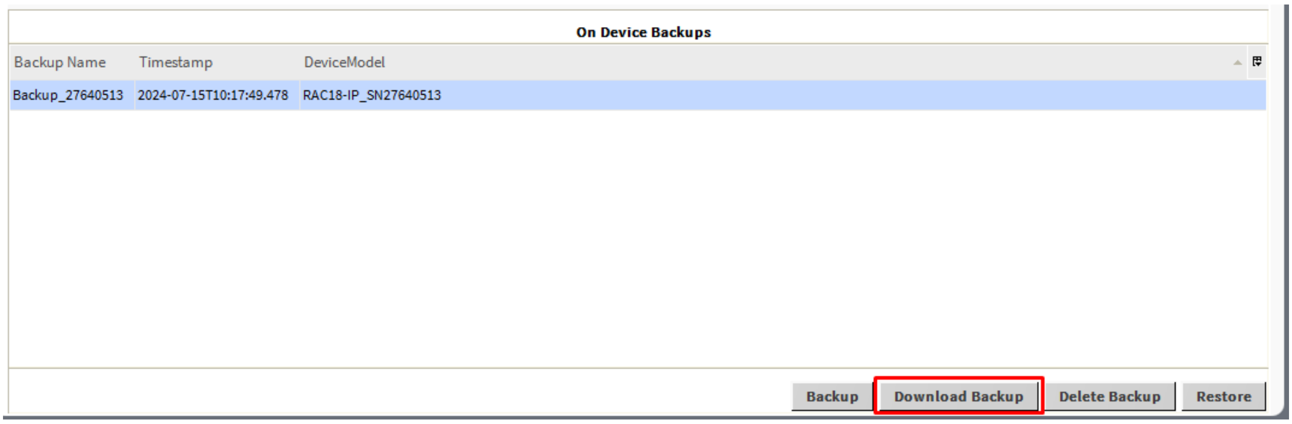
On Device Backups

Backup Name	Timestamp	DeviceModel
Backup_27640513	2024-07-15T10:17:49.478	RAC18-IP_SN27640513

Figure 87. On Device Backups

4.7.2 Downloading Backup

- Click the Download Backup button to download the backup from the device to the local station.

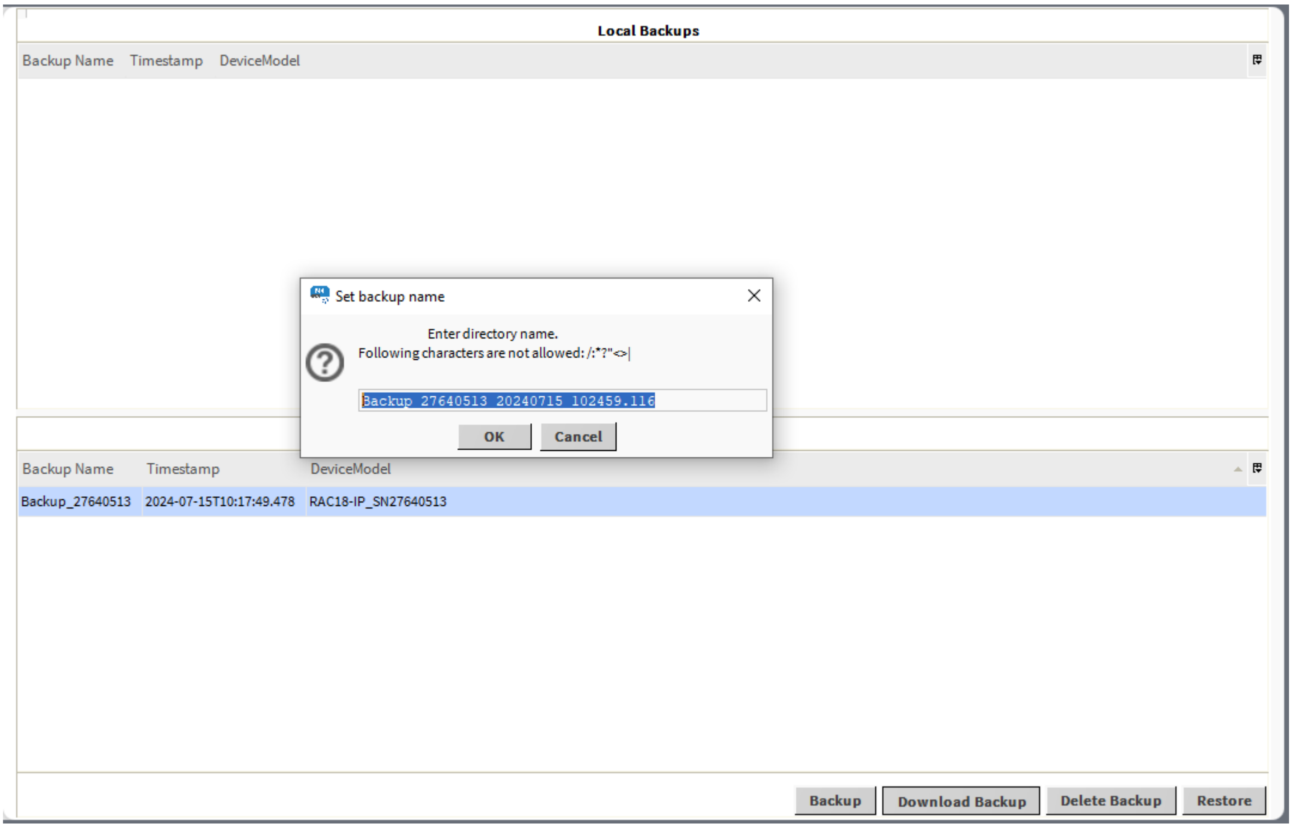


On Device Backups

Backup Name	Timestamp	DeviceModel
Backup_27640513	2024-07-15T10:17:49.478	RAC18-IP_SN27640513

Figure 88. Download Backup button

A pop-up will appear. Set the Backup name or keep the default name. Click Ok to Confirm once the backup name is set. The backup will be downloaded.



Local Backups

Backup Name	Timestamp	DeviceModel

Set backup name

Enter directory name.
Following characters are not allowed: /:*?"<>|

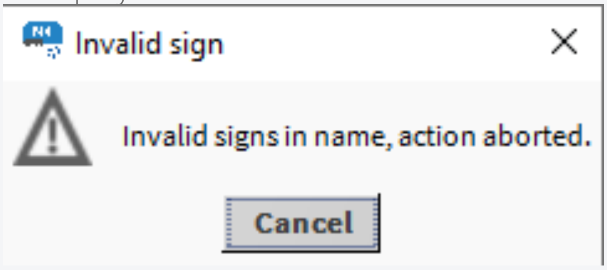
Backup_27640513_20240715_102459.116

Backup Name	Timestamp	DeviceModel
Backup_27640513	2024-07-15T10:17:49.478	RAC18-IP_SN27640513

Figure 89. Changing backup name

Note

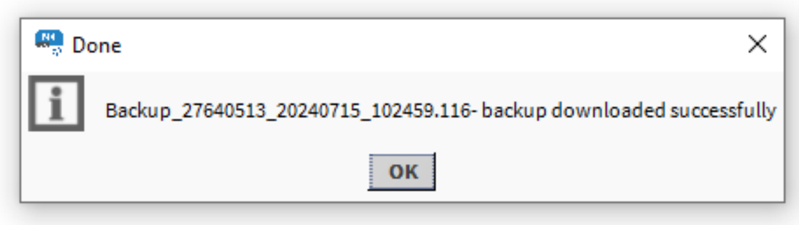
The Set backup name pop-up informs about characters that are not allowed to be included in a backup's name. If one of them is, the action will be aborted and the following prompt will be displayed:


Invalid sign

Invalid signs in name, action aborted.

Figure 90. Invalid characters included in the backup's name

A pop-up will appear informing about the completed download process.

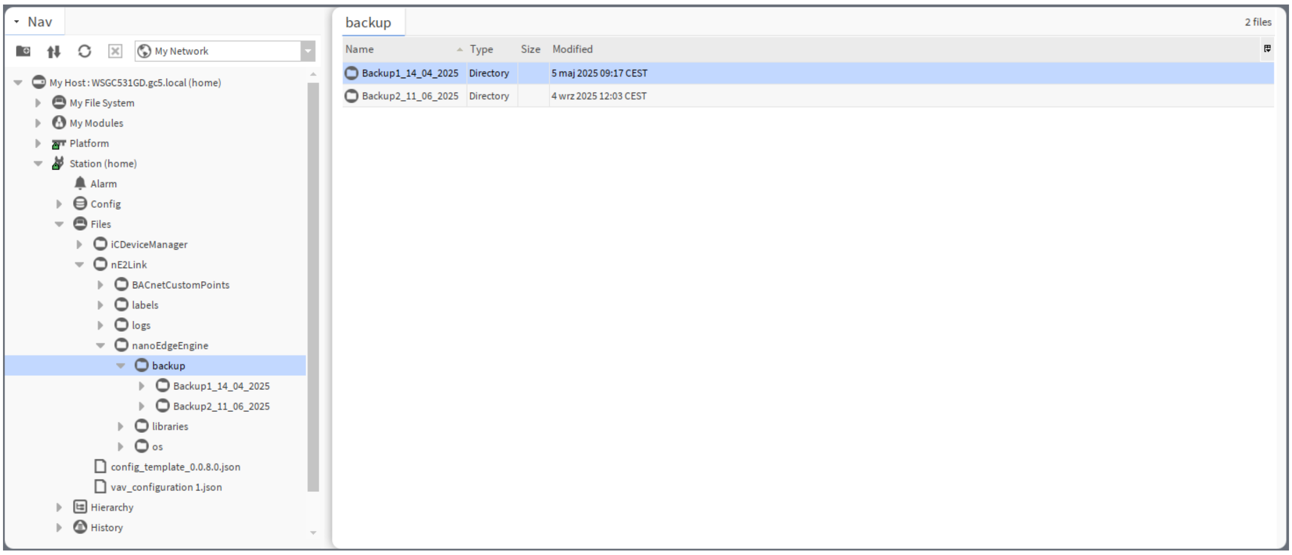

Done

Backup_27640513_20240715_102459.116- backup downloaded successfully

Figure 91. Successful backup confirmation

Once the process is completed, the backup will be visible in the Local Backups table in the Backup Manager view.

All local backups are stored in the local Niagara station. To access backups go to Station → Files → nE2Link → nanoEdgeEngine ->backup → *BackupName.* Backups can be imported or exported from this location manually and will become visible in the Backup Manager view.



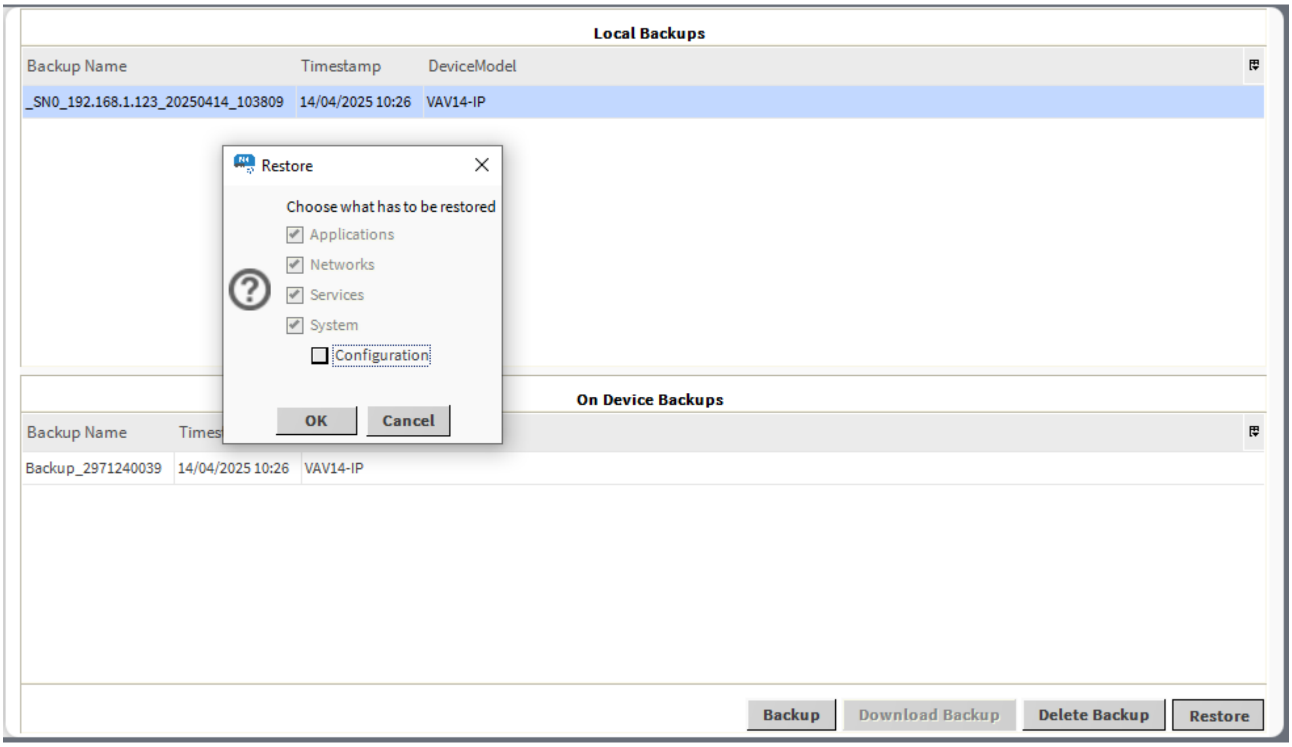
Name	Type	Size	Modified
Backup1_14_04_2025	Directory		5 maj 2025 09:17 CEST
Backup2_11_06_2025	Directory		4 wrz 2025 12:03 CEST

Figure 92. Backups stored in the station files

4.7.3 Restoring Backup

The backup can be restored to the device. To perform the restore function, select the backup to be restored to the device and click the Restore button.

A pop-up will appear with available containers that should be restored to the device. Select the proper configuration and confirm with the OK button.



Local Backups

Backup Name	Timestamp	DeviceModel
_SN0_192.168.1.123_20250414_103809	14/04/2025 10:26	VAV14-IP

Restore

Choose what has to be restored

- ☑ Applications
- ☑ Networks
- ☑ Services
- ☑ System
  - □ Configuration

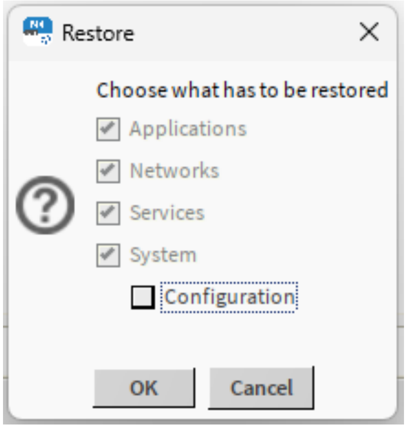
On Device Backups

Backup Name	Times	
Backup_2971240039	14/04/2025 10:26	VAV14-IP

Figure 93. Available restore options will appear after clicking on the Restore button

Warning!
Restoring backup will overwrite the existing application loaded on the device.

A pop-up window will appear to confirm restoring of the backup. Click Yes to start the restoring process.



Restore

Choose what has to be restored

- ☑ Applications
- ☑ Networks
- ☑ Services
- ☑ System
  - □ Configuration

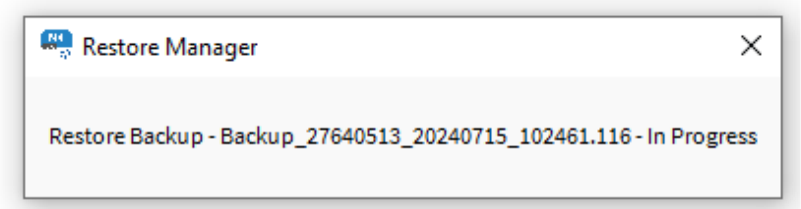
Figure 94. Restore pop-up

While restoring a backup, it is possible to decide whether the configuration settings are restored from the backup too, or left as set on the device. By default, the option to restore configuration settings from a backup is unchecked.

The configuration settings include:

- iFnet port,
- IP address,
- mask,
- default gateway,
- DHCP enabled,
- device name,
- DNS1,
- DNS2,
- BACnet LocalDevice ID.

Please wait until the end of the process a pop-up will inform the user about the process in progress.

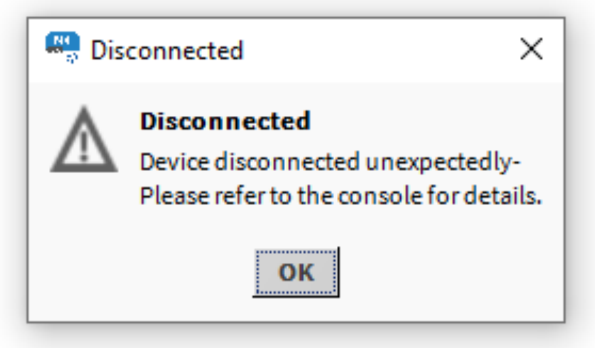


Restore Manager

Restore Backup - Backup_27640513_20240715_102461.116 - In Progress

Figure 95. Restoring in progress

Once the process is finished, a pop-up will appear informing that the device has been disconnected. Right-click on the nE2DeviceExt, go to Actions → Connect to reconnect with the device.



Disconnected

Disconnected

Device disconnected unexpectedly-
Please refer to the console for details.

Figure 96. Device disconnected after restoring backup

5 Programming

5.1 nano EDGE ENGINE Libraries

nano EDGE ENGINE enables real-time device programming using components from libraries installed on the device. Application programming on the controller is done in the Applications container. The nano EDGE ENGINE controller can run multiple applications in different time cycles, running simultaneously.

> To learn more about the Applications, please refer to the nano EDGE ENGINE Programming user manual.

To start programming, make sure that the required libraries are installed on the device.

> To learn more about the nano EDGE ENGINE libraries and components, please refer to the nano EDGE ENGINE Programming user manual.

5.1.1 Installing Libraries

Software Manager is the tool designed to manage (install, uninstall, upgrade, downgrade) nano EDGE ENGINE OS and libraries. To start programming applications, first, go to the Software Manager and make sure that the OS is installed in the required version and all required libraries are installed in relevant versions.

5.2 Programming

Programming of the nano EDGE ENGINE devices is typically carried out by dragging and dropping components from the installed libraries (either directly from the nE2 Link module or from the Libraries folder in the device tree, see sections below) to the Applications container. It is possible to drop the components directly to the device tree or onto the wire sheet (in this case, the wire sheet must be opened for the Application or Equipment component).

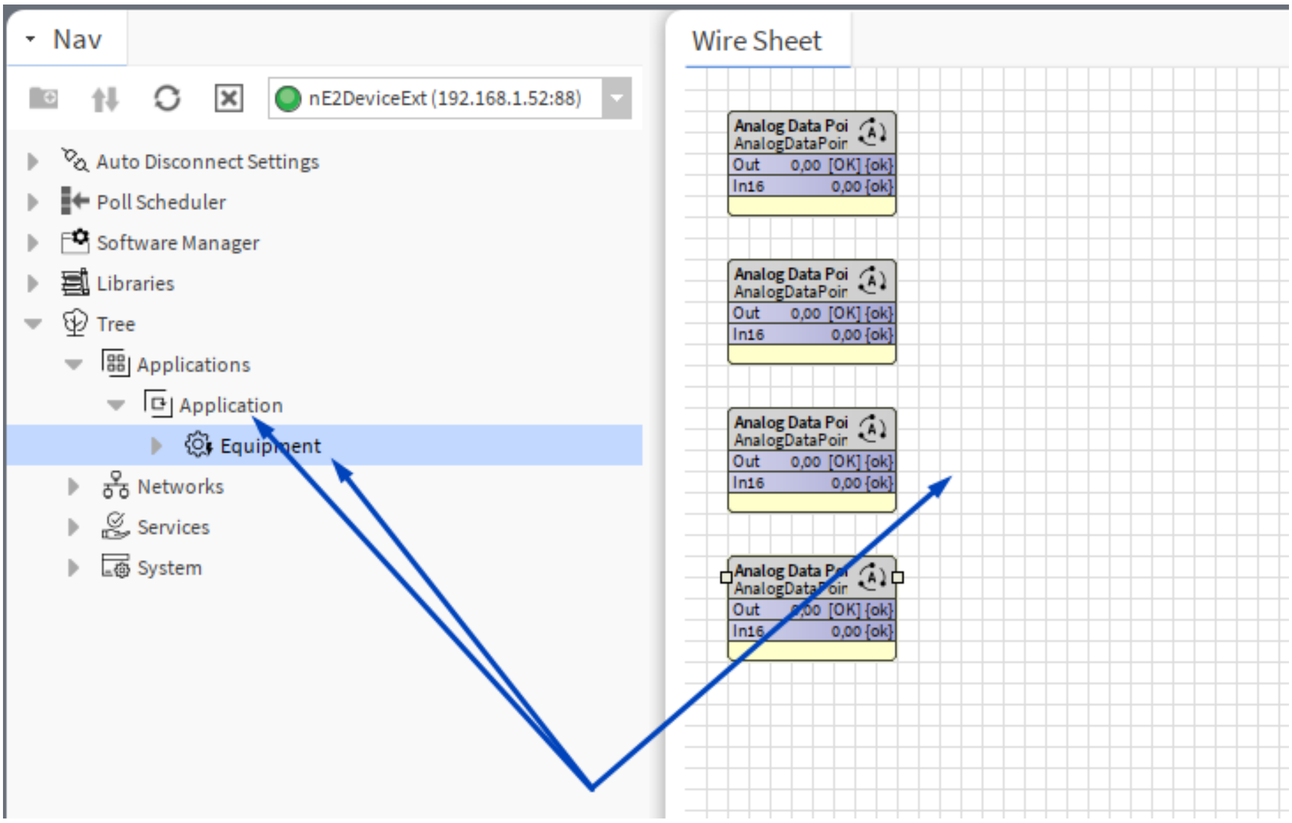


Figure 97. Example locations to drop a component

Basic components (Data Points, schedules, calendar, and folder-type components) can be added by right-clicking on the Application/Equipment or directly on the wire sheet.

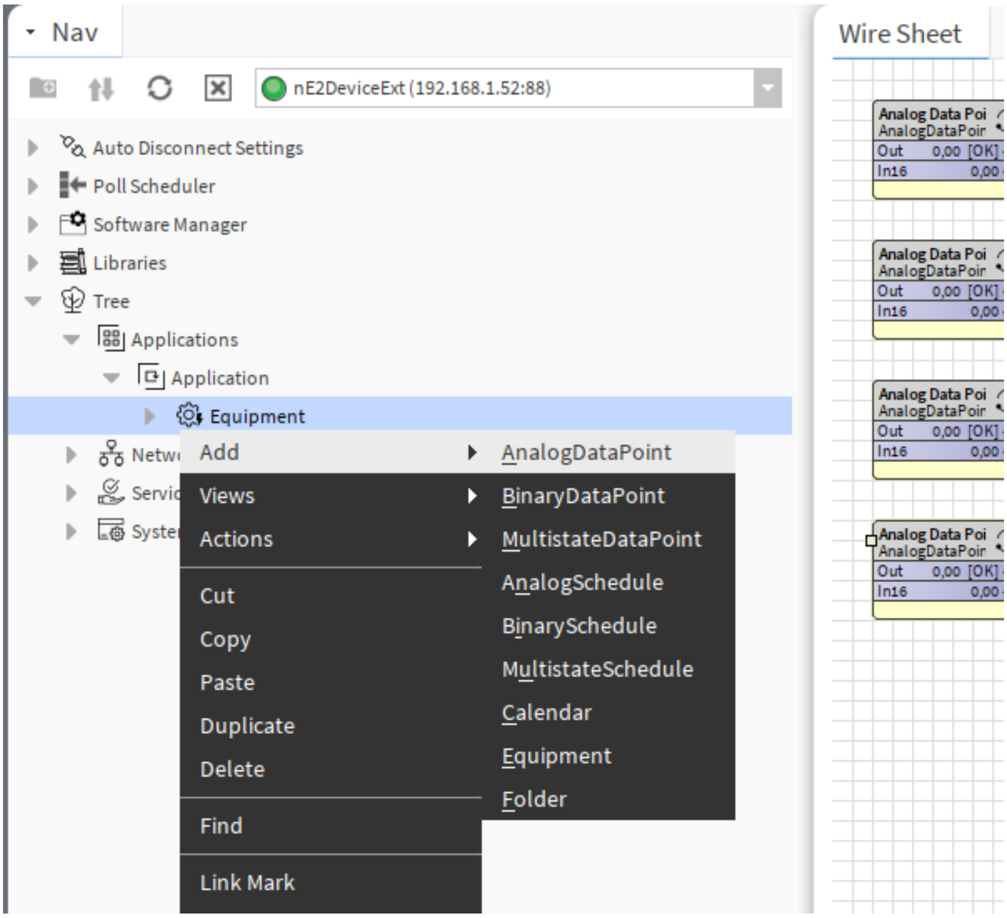


Figure 98. Context menu for adding basic components in the device tree

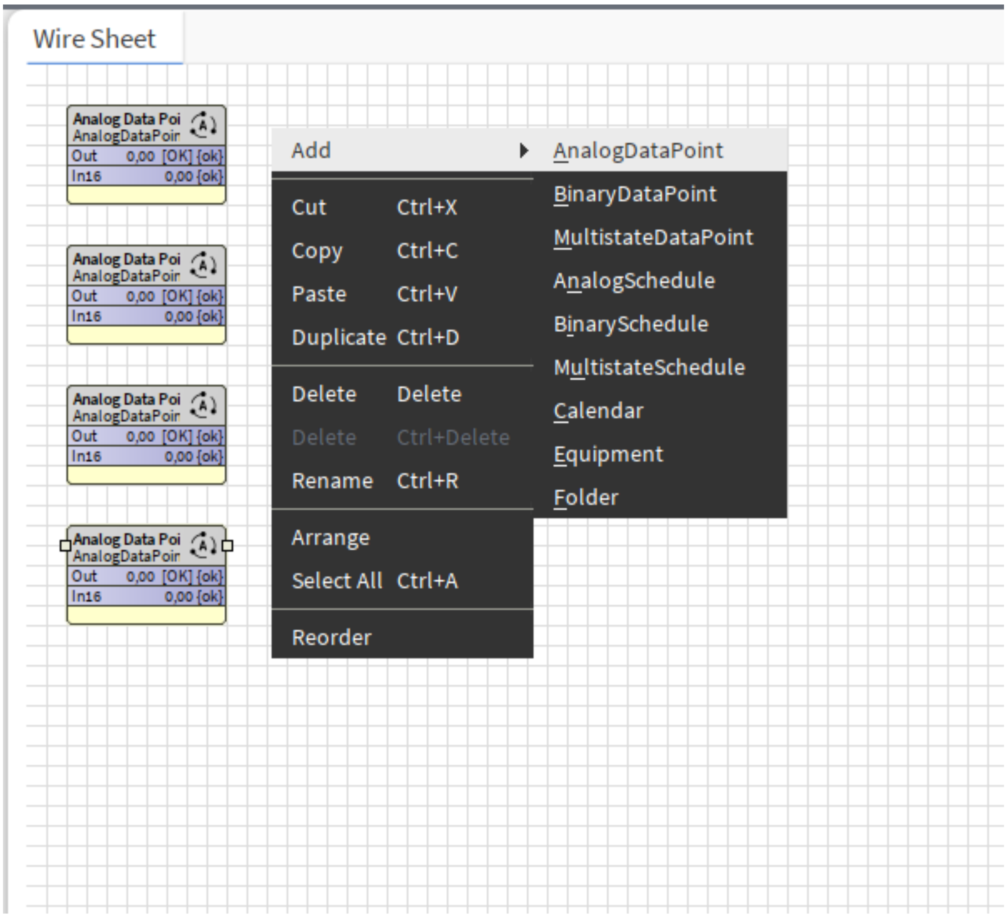


Figure 99. Context menu for adding basic components on the wire sheet

Note
All the Data Points required for the application programming are stored in the Library.Core. To learn more about the Core library and its components, please refer to the nano EDGE ENGINE Programming user manual.

5.2.1 Application Structure

The nano EDGE Engine organizes the application structure using a clear, hierarchical model designed for scalable, logical, and intuitive system configuration. The architecture is designed to support the tagging functionality for efficient data modelling.

Applying tags in the nano EDGE ENGINE is based on a semantic approach that ensures consistent data structure which is easily usable by the nanoWebUI™ and by third-party systems. Tags are applied at the Equipment and Data Point levels, where Equipment serves as the logical container defining what is being controlled, and Data Points represent the measured or commanded values associated with that equipment. This structured model ensures that tagged data is immediately usable by platforms capable of communicating through standardized tag-based HTTP APIs, e.g., Haystack.

Note: Tags can be only applied to the Equipment components and Data Points. Other component types are not supported.

It is therefore recommended (however, not mandatory) to use the following structure when creating applications:

- Applications container
    - Application component
        - Equipment component
            - Data Point(s)
            - other components
        - Equipment component
            - Data Point(s)
            - other components

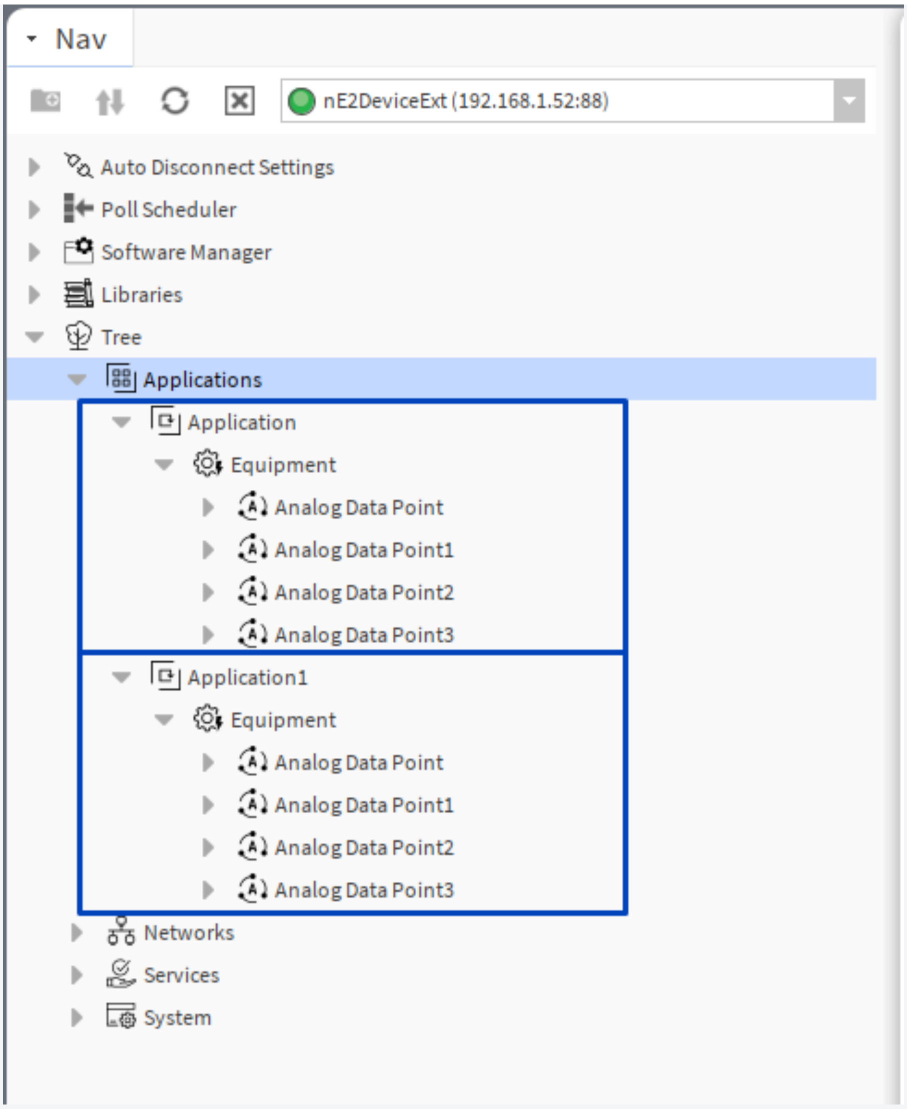


Figure 100. Recommended application structure

To find out more about the tagging functionality for data modelling and nanoWebUI™ interface, please refer to the Tagging, Haystack, and Web services.

5.2.2 Programming with Libraries in nE2 Link Module

From the V1.7 and up, the nE2 Link module provides nano EDGE ENGINE libraries to install on the device (using the Software Manager).

Important - compatible versions
In order to use libraries available directly in the nE2 Link module, it is required to make sure that the controller is upgraded to the latest version supported by the nE2 Link. To check it, go to the Software Manager and see if the OS or any library is marked as outdated or available to install. If so, perform an upgrade and/or installation of required elements.
If - for any reason - the OS/library versions are not the same as the nE2 Link version and they cannot be upgraded because of any system requirements, it is recommended to use the components coming from the device tree (see the Programming from Device Libraries section below).

In the Palette of the nE2Link module there is the Libraries folder with all nano EDGE ENGINE libraries available to install on the device:

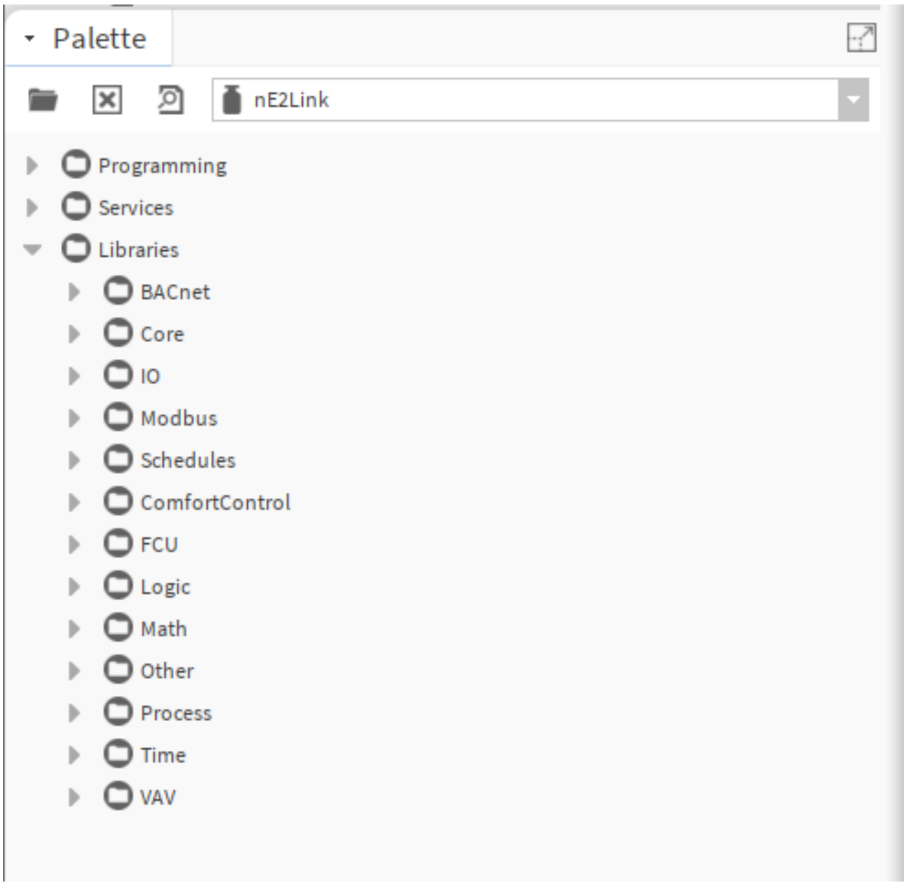


Figure 101. Libraries available in the palette

Programming of the nano EDGE ENGINE applications is typically carried out by dragging and dropping components in the Applications container, under the Application component.

Components can be dragged and dropped directly from the Palette folder.

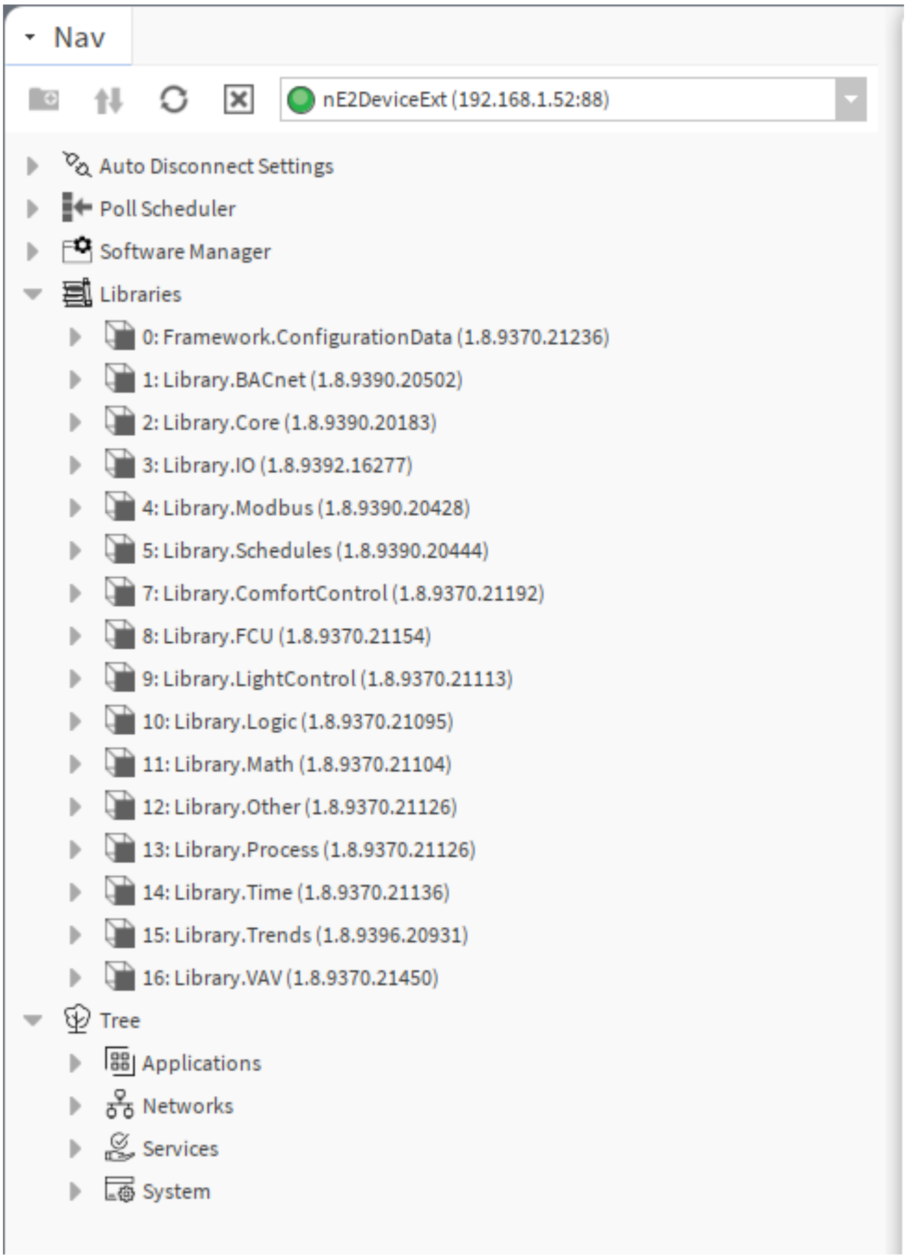
Important
When adding components from the Palette window, all libraries and components will be visible on the list, but only components from the libraries installed on the device can be effectively added to the application.

5.2.3 Programming with Device Libraries

The user can program the nano EDGE ENGINE device using installed libraries and components found in the Libraries folder.

The libraries on the device act as a palette of components that can be dropped into the device logic. If the user does not have a dedicated module with nano EDGE ENGINE libraries, it is possible to use those installed on the device.

Libraries installed on the device are accessible in the device tree:



Nav

nE2DeviceExt (192.168.1.52:88)

- Auto Disconnect Settings
- Poll Scheduler
- Software Manager
- Libraries
  - 0: Framework.ConfigurationData (1.8.9370.21236)
  - 1: Library.BACnet (1.8.9390.20502)
  - 2: Library.Core (1.8.9390.20183)
  - 3: Library.IO (1.8.9392.16277)
  - 4: Library.Modbus (1.8.9390.20428)
  - 5: Library.Schedules (1.8.9390.20444)
  - 7: Library.ComfortControl (1.8.9370.21192)
  - 8: Library.FCU (1.8.9370.21154)
  - 9: Library.LightControl (1.8.9370.21113)
  - 10: Library.Logic (1.8.9370.21095)
  - 11: Library.Math (1.8.9370.21104)
  - 12: Library.Other (1.8.9370.21126)
  - 13: Library.Process (1.8.9370.21126)
  - 14: Library.Time (1.8.9370.21136)
  - 15: Library.Trends (1.8.9396.20931)
  - 16: Library.VAV (1.8.9370.21450)
- Tree
  - Applications
  - Networks
  - Services
  - System

Figure 102. Libraries available in the device tree

The libraries on the device act as a palette of components that can be dropped into the device logic. To enhance the user experience, it is recommended to open an additional nav view for the Libraries view:

- in Workbench, go to Window → Sidebar → Nav. Select Nav.

A new Nav view will appear on the left bottom side of the Workbench view. Navigate to the nE2DeviceExt, right-click on the Libraries folder, and select Go Into.

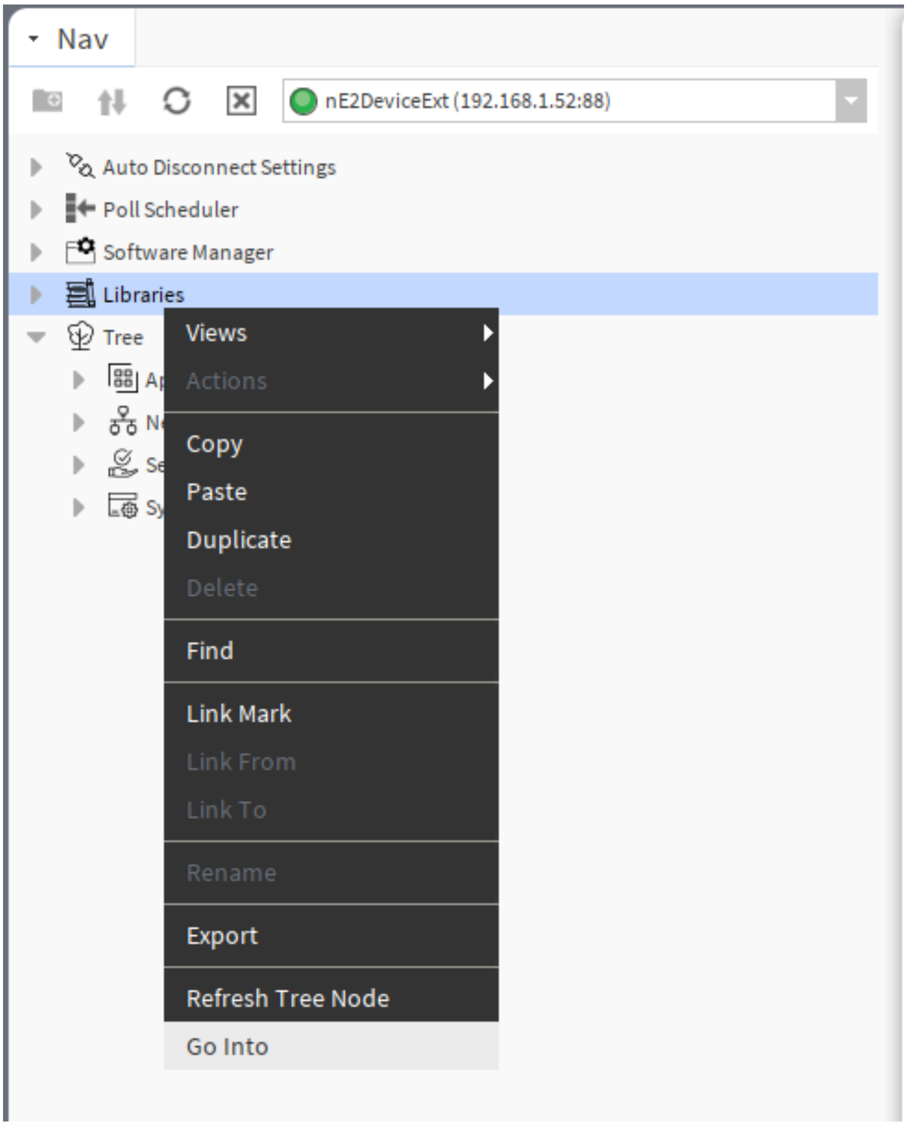
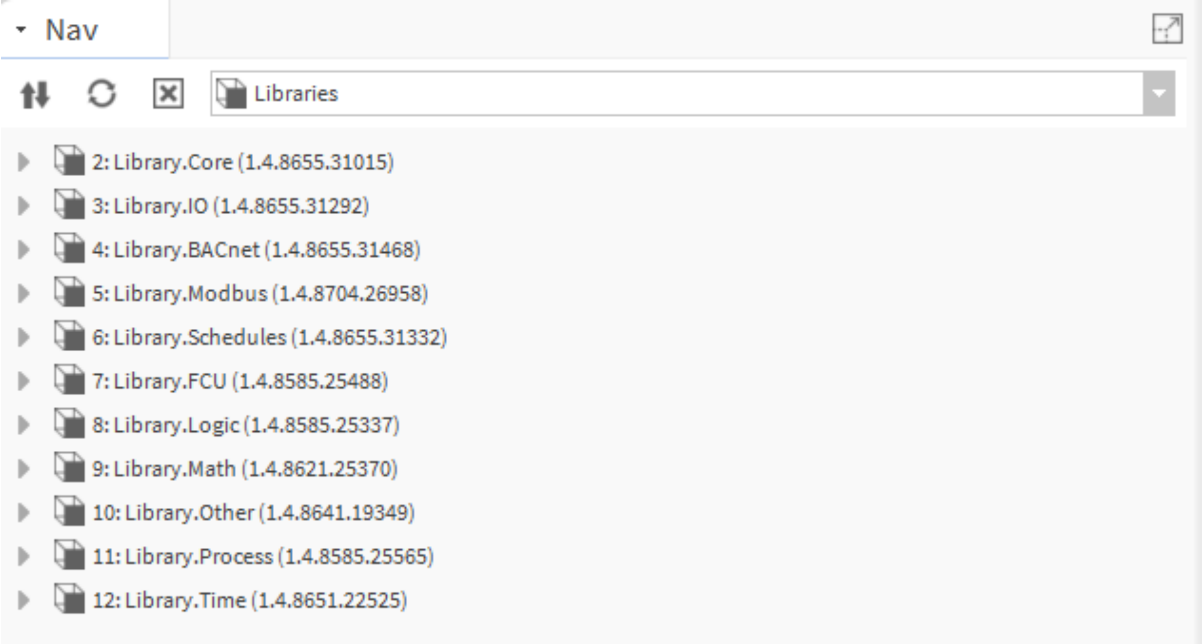


Figure 103. Go into libraries

An on-device libraries list will appear.



2: Library.Core (1.4.8655.31015)
3: Library.IO (1.4.8655.31292)
4: Library.BACnet (1.4.8655.31468)
5: Library.Modbus (1.4.8704.26958)
6: Library.Schedules (1.4.8655.31332)
7: Library.FCU (1.4.8585.25488)
8: Library.Logic (1.4.8585.25337)
9: Library.Math (1.4.8621.25370)
10: Library.Other (1.4.8641.19349)
11: Library.Process (1.4.8585.25565)
12: Library.Time (1.4.8651.22525)

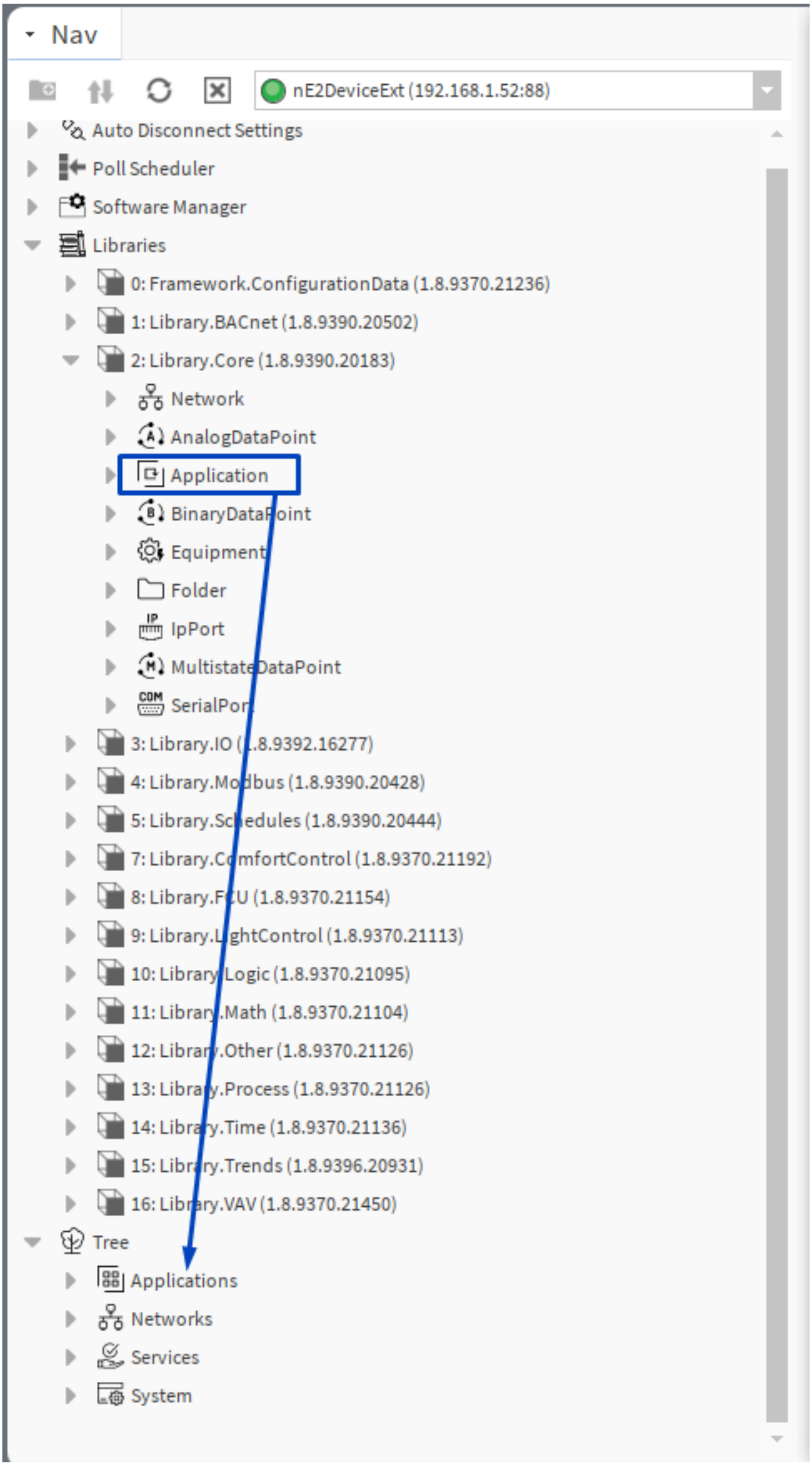
Figure 104. On-device libraries

5.3 Application

The Applications container allows to add multiple Application components for building independent user applications, which are cycle-driven and may work simultaneously.

The user may define the application purpose (heating, lighting, etc.) and a cycle time of algorithms operation (cycles may differ between applications).

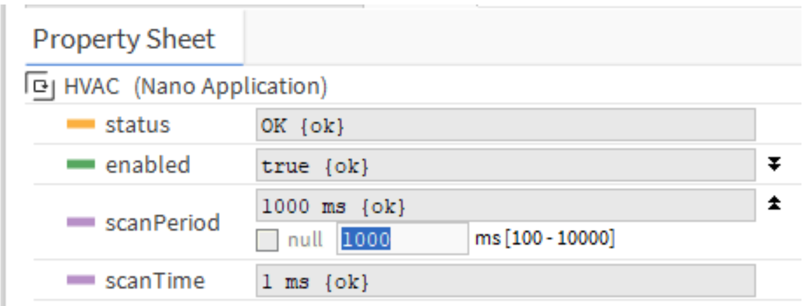
To create a first application, drag and drop the Application component from Library.Core to the Applications container and name it as appropriate.



Nav
nE2DeviceExt (192.168.1.52:88)
Auto Disconnect Settings
Poll Scheduler
Software Manager
Libraries
0: Framework.ConfigurationData (1.8.9370.21236)
1: Library.BACnet (1.8.9390.20502)
2: Library.Core (1.8.9390.20183)
Network
AnalogDataPoint
Application
BinaryDataPoint
Equipment
Folder
IpPort
MultistateDataPoint
SerialPort
3: Library.IO (1.8.9392.16277)
4: Library.Modbus (1.8.9390.20428)
5: Library.Schedules (1.8.9390.20444)
7: Library.ComfortControl (1.8.9370.21192)
8: Library.FCU (1.8.9370.21154)
9: Library.LightControl (1.8.9370.21113)
10: Library.Logic (1.8.9370.21095)
11: Library.Math (1.8.9370.21104)
12: Library.Other (1.8.9370.21126)
13: Library.Process (1.8.9370.21126)
14: Library.Time (1.8.9370.21136)
15: Library.Trends (1.8.9396.20931)
16: Library.VAV (1.8.9370.21450)
Tree
Applications
Networks
Services
System

Figure 105. Drag and drop the Application component to Applications

Go to the Application AX Property Sheet. Configure the scanPeriod slot and click Save.



Property Sheet

HVAC (Nano Application)

status	OK {ok}
enabled	true {ok}
scanPeriod	1000 ms {ok} □ null 1000 ms [100 - 10000]
scanTime	1 ms {ok}

Figure 106. Setting the application's scan period

To learn more about the Application component, please refer to the nano EDGE ENGINE Programming user manual.

5.3.1 Folder

The Folder component is a grouping component, which can be used to gather other components to help organize the tree. The Folder component can be used both in the Applications and Networks containers, however, it cannot be added directly to the container. The Folder component can be freely renamed to facilitate categorization of components included within.

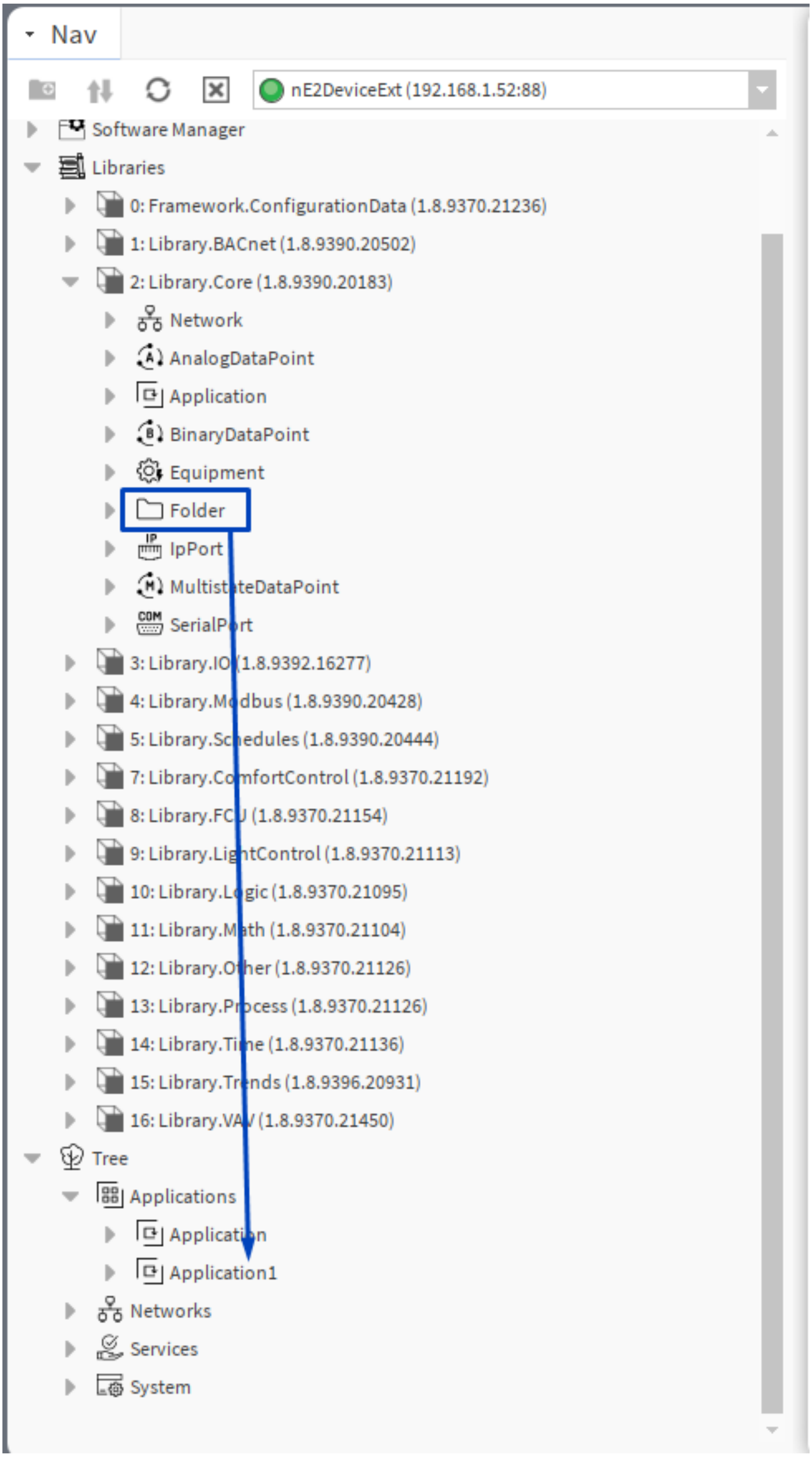

Nav
nE2DeviceExt (192.168.1.52:88)
Software Manager
Libraries
0: Framework.ConfigurationData (1.8.9370.21236)
1: Library.BACnet (1.8.9390.20502)
2: Library.Core (1.8.9390.20183)
Network
AnalogDataPoint
Application
BinaryDataPoint
Equipment
Folder
IpPort
MultistateDataPoint
SerialPort
3: Library.IO (1.8.9392.16277)
4: Library.Modbus (1.8.9390.20428)
5: Library.Schedules (1.8.9390.20444)
7: Library.ComfortControl (1.8.9370.21192)
8: Library.FCU (1.8.9370.21154)
9: Library.LightControl (1.8.9370.21113)
10: Library.Logic (1.8.9370.21095)
11: Library.Math (1.8.9370.21104)
12: Library.Other (1.8.9370.21126)
13: Library.Process (1.8.9370.21126)
14: Library.Time (1.8.9370.21136)
15: Library.Trends (1.8.9396.20931)
16: Library.VAV (1.8.9370.21450)
Tree
Applications
Application
Application1
Networks
Services
System

Figure 107. Drag and drop Folder to the Application

It is also possible to add the Folder component directly from the context menu:

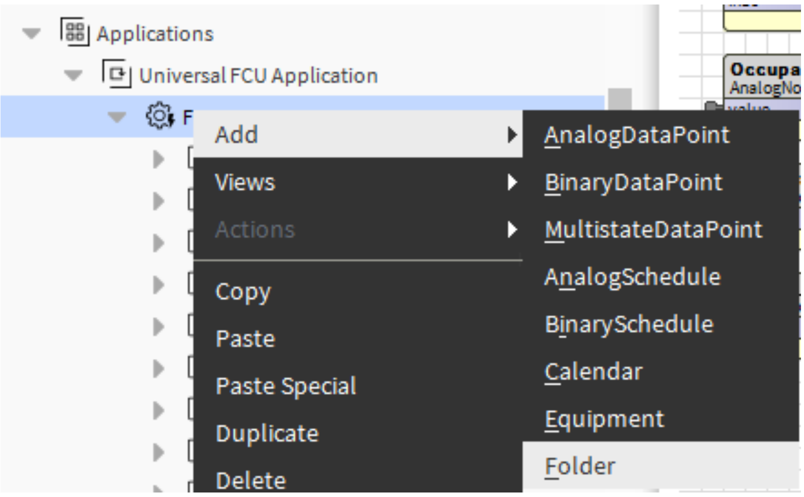
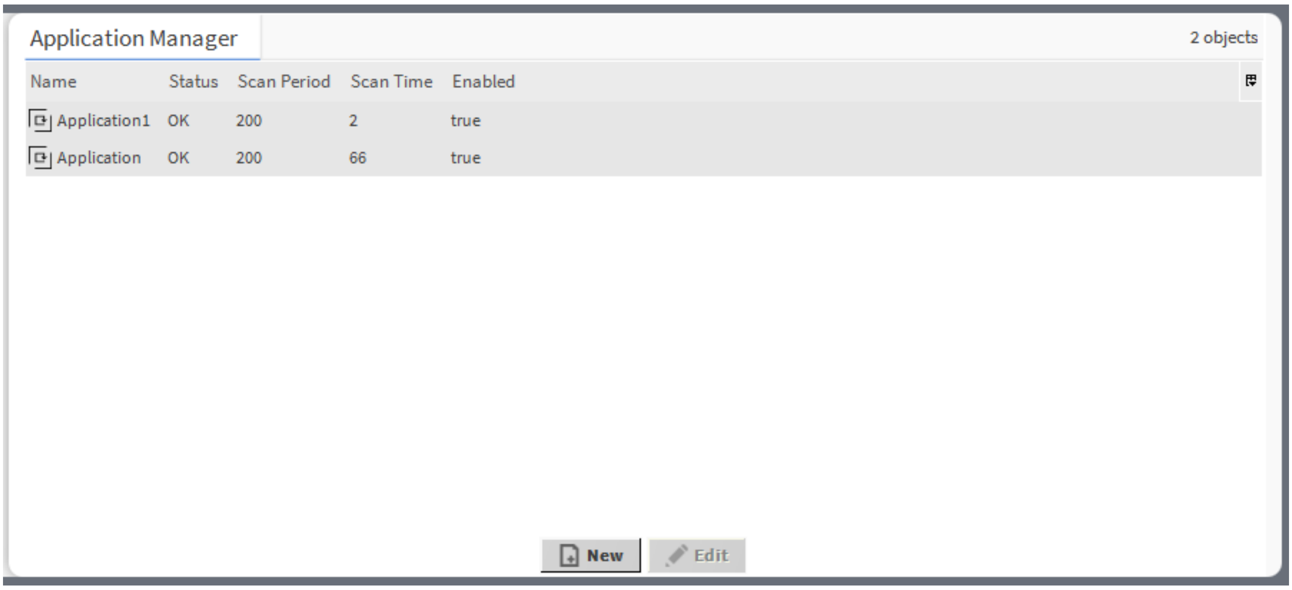


Figure 108. Adding the Folder component from the context menu

To learn more about the Folder component, please refer to the nano EDGE ENGINE Programming user manual.

5.3.2 Application Manager

The Application Manager is a special view that allows to manage the Application components added to the Applications container.


Application Manager

2 objects

Name	Status	Scan Period	Scan Time	Enabled
Application1	OK	200	2	true
Application	OK	200	66	true

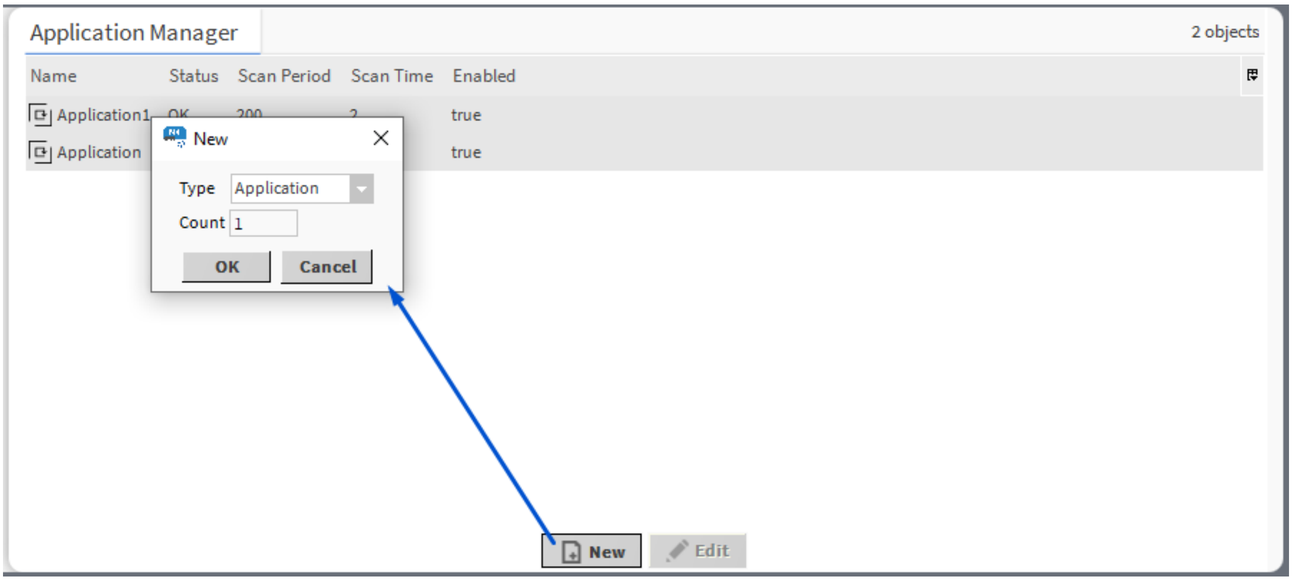
Figure 109. Application Mananger

The Application Manager lists all the Application components used on the device. The view shows the following fields:

- name of the application;
- status;
- scan period;
- scan time;
- enabled or disabled status.

In the Application Manager, it is possible to:

- add Application components:



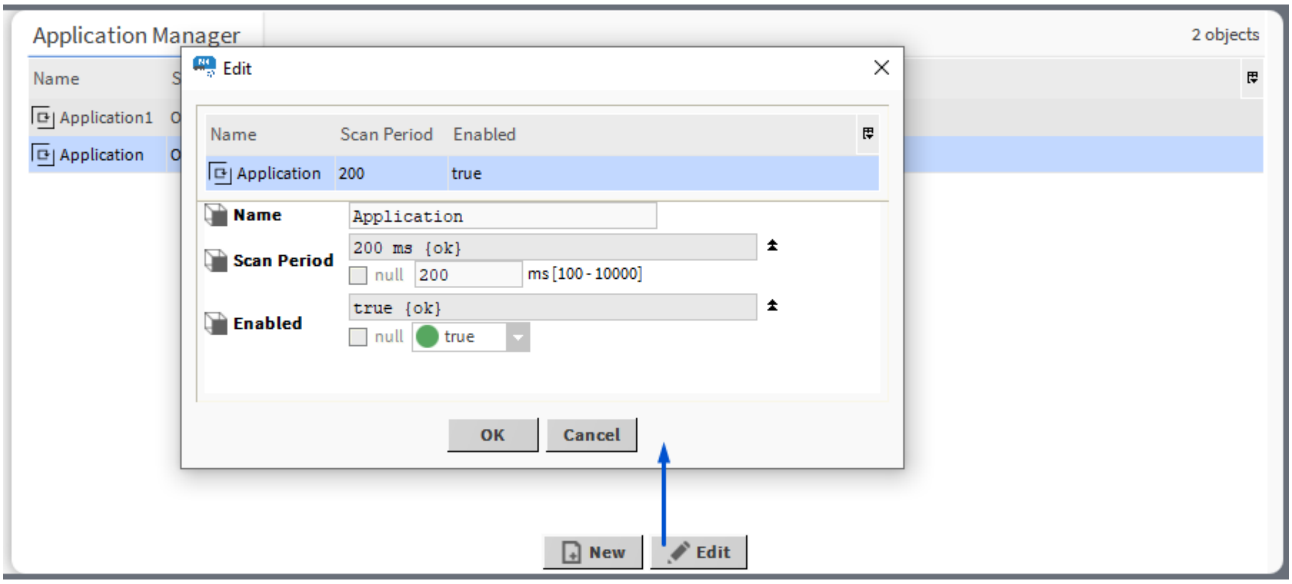
Name	Status	Scan Period	Scan Time	Enabled
Application1	OK	200	2	true
Application				true

Type: Application

Count: 1

Figure 110. Adding new Application component in the Application Manager

- edit the Application's name and scan period and enable/disable the component:



Name	Scan Period	Enabled
Application	200	true

Name: Application

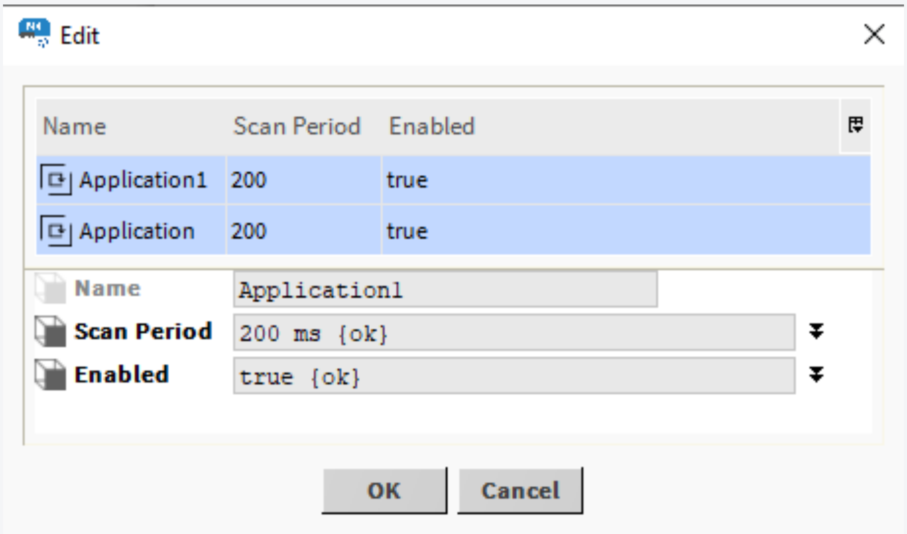
Scan Period: 200 ms {ok}
□ null 200 ms [100 - 10000]

Enabled: true {ok}
□ null ● true

Figure 111. Edit pop-up

Note

Editing is possible for more than one component at a time. If multiple components are edited, the same new value is written to common slots, so individual slots, such as Name, cannot be edited in this manner.

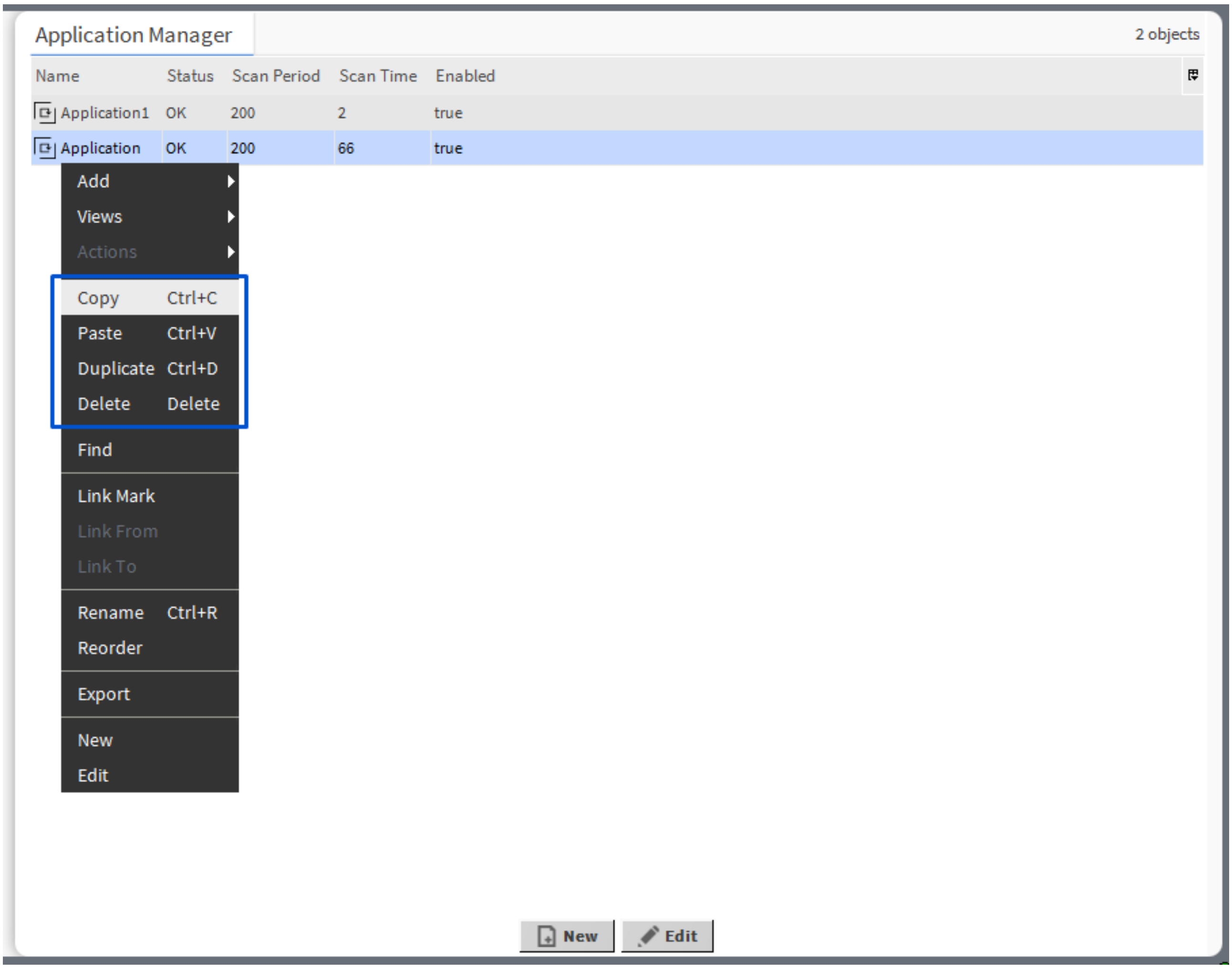


Name	Scan Period	Enabled
Application1	200	true
Application	200	true

Name: Application1
Scan Period: 200 ms {ok}
Enabled: true {ok}

Figure 112. Editing of multiple components

- copy/duplicate/remove Application components:



Application Manager — 2 objects

Name	Status	Scan Period	Scan Time	Enabled
Application1	OK	200	2	true
Application	OK	200	66	true

Figure 113. Context menu options for the Application

Opening the Application Manager

The Application Manager view is accessible in the context menu of the Applications container.

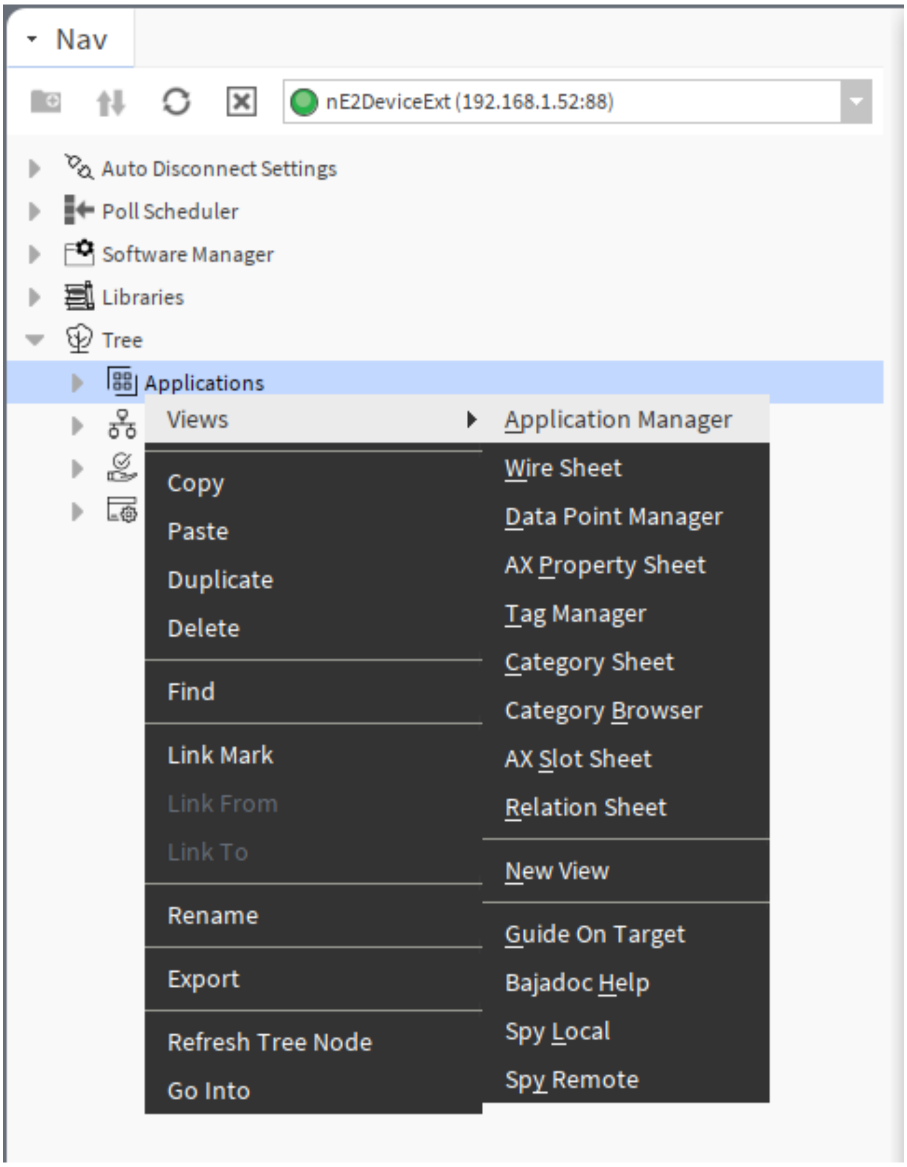


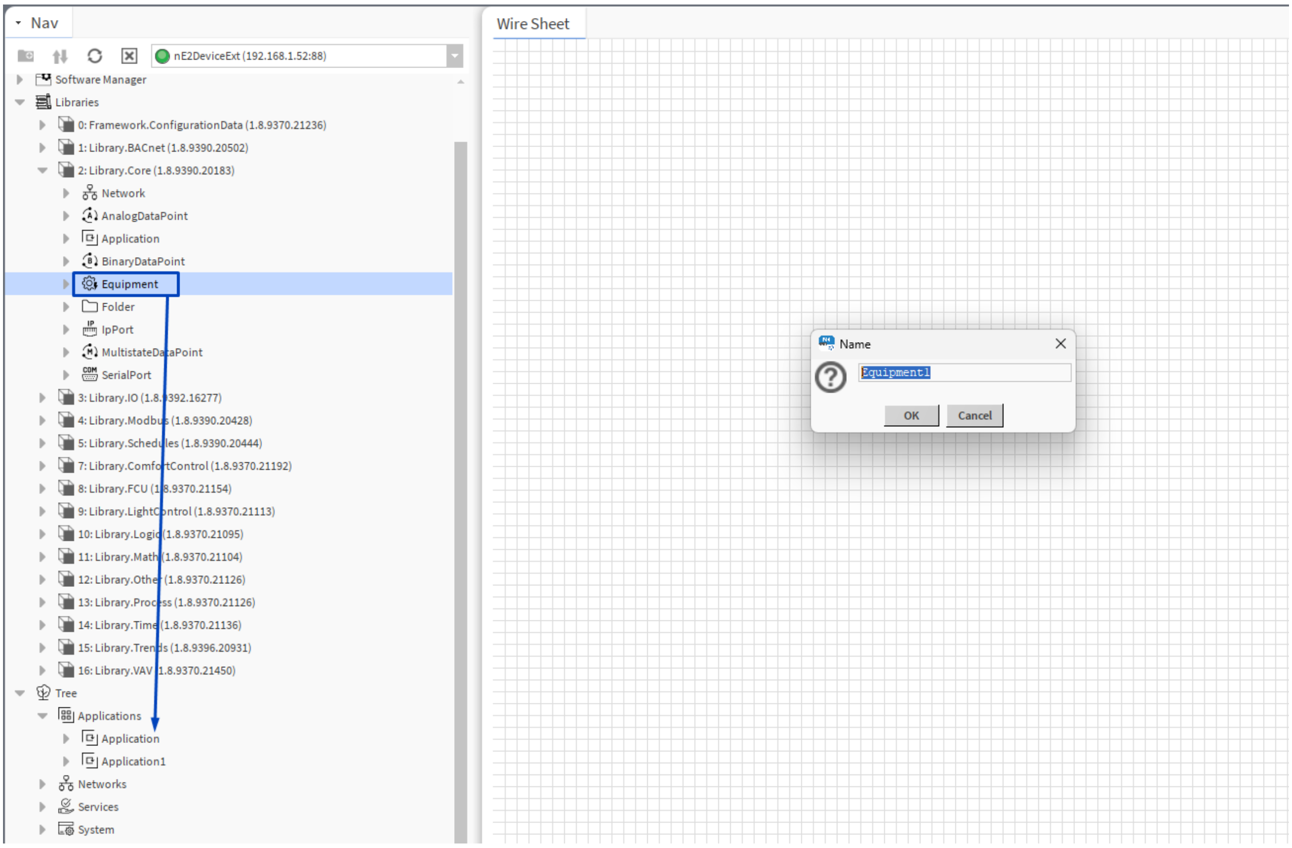
Figure 114. Accessing the Application Manager

The Application Manager view is also automatically opened if the Applications container is double-clicked in the nav tree window.

5.4 Equipment

Double-click the Application component (below, renamed as HVAC, as a reference to the purpose of the application), the wire sheet opens. Drag and drop the Equipment component to the wire sheet, and name the component as appropriate (here, FCU).

> The Equipment component is a grouping folder-type component, which can be used to gather other components, regarding specific equipment included in the Application, to help organize the Tree. It can be freely renamed to facilitate categorization of components included within.


Nav
nE2DeviceExt (192.168.1.52:88)
Software Manager
Libraries
0: Framework.ConfigurationData (1.8.9370.21236)
1: Library.BACnet (1.8.9390.20502)
2: Library.Core (1.8.9390.20183)
Network
AnalogDataPoint
Application
BinaryDataPoint
Equipment
Folder
IpPort
MultistateDataPoint
SerialPort
3: Library.IO (1.8.9392.16277)
4: Library.Modbus (1.8.9390.20428)
5: Library.Schedules (1.8.9390.20444)
7: Library.ComfortControl (1.8.9370.21192)
8: Library.FCU (1.8.9370.21154)
9: Library.LightControl (1.8.9370.21113)
10: Library.Logic (1.8.9370.21095)
11: Library.Math (1.8.9370.21104)
12: Library.Other (1.8.9370.21126)
13: Library.Process (1.8.9370.21126)
14: Library.Time (1.8.9370.21136)
15: Library.Trends (1.8.9396.20931)
16: Library.VAV (1.8.9370.21450)
Tree
Applications
Application
Application1
Networks
Services
System
Wire Sheet
OK
Cancel

Name: Equipment1

Figure 115. Add (drag and drop) and rename the Equipment component

It is also possible to add the Equipment component directly from the context menu of the Application component:

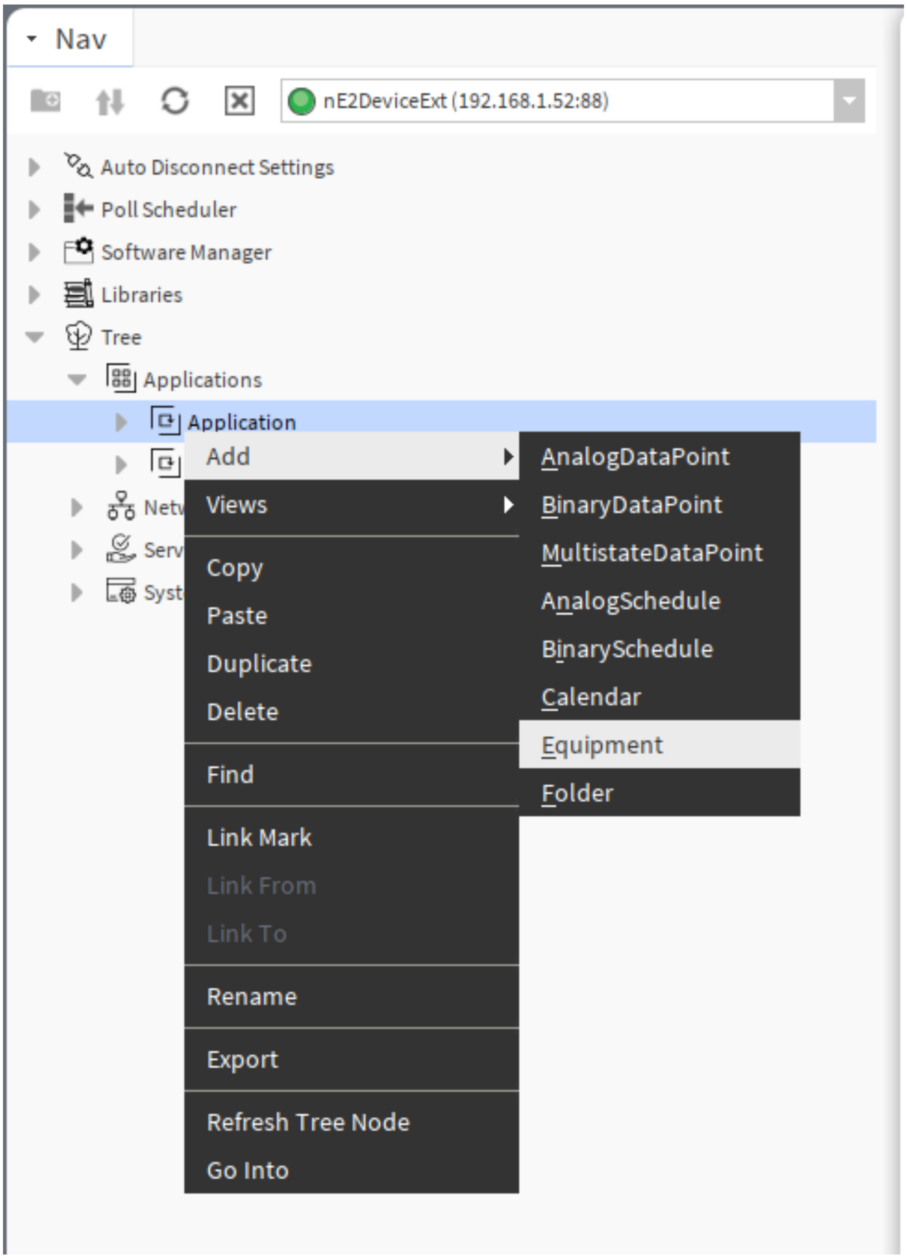


Figure 116. Adding the Equipment component from the context menu

To learn more about the Equipment component, please refer to the nano EDGE ENGINE Programming user manual.

5.5 Data Points

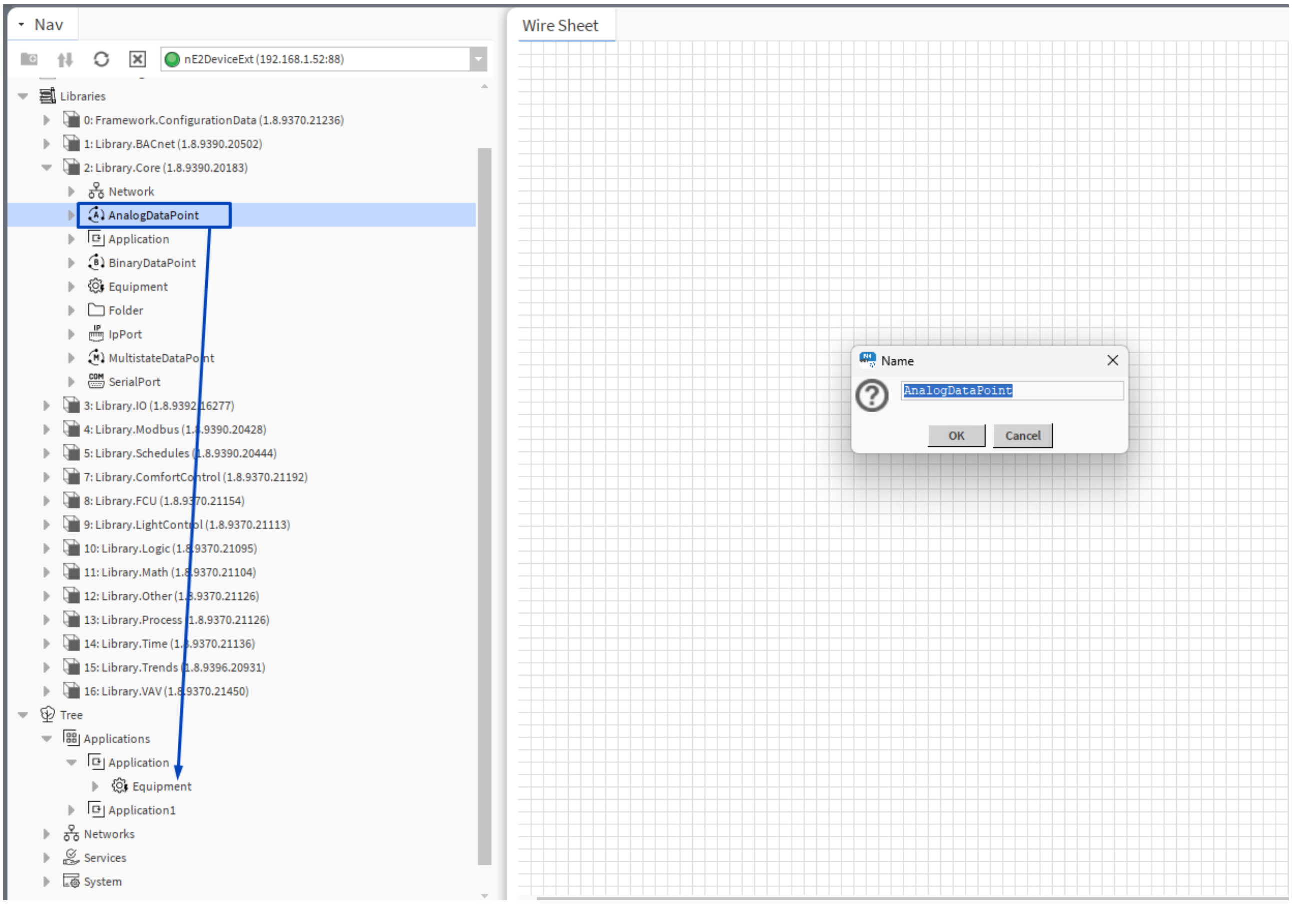
Data Points are universal components that represent a value in the application logic. The available Data Points:

- AnalogDataPoint with native BACnetAnalogPoint and ModbusAnalogPoint extensions;
- BinaryDataPoint with native BACnetBinaryPoint and ModbusBinaryPoint extensions;
- MultistateDataPoint with native BACnetMultistatePoint and ModbusMultistatePoint extensions.

To learn more about Data Points, please refer to the nano EDGE ENGINE Programming user manual.

5.5.1 Adding Data Points

To add a Data Point to the application, drag and drop the relevant component (AnalogDataPoint, BinaryDataPoint, or MultistateDataPoint) from the Library.Core to the application.


Nav
nE2DeviceExt (192.168.1.52:88)
Libraries
0: Framework.ConfigurationData (1.8.9370.21236)
1: Library.BACnet (1.8.9390.20502)
2: Library.Core (1.8.9390.20183)
Network
AnalogDataPoint
Application
BinaryDataPoint
Equipment
Folder
IpPort
MultistateDataPoint
SerialPort
3: Library.IO (1.8.9392.16277)
4: Library.Modbus (1.8.9390.20428)
5: Library.Schedules (1.8.9390.20444)
7: Library.ComfortControl (1.8.9370.21192)
8: Library.FCU (1.8.9370.21154)
9: Library.LightControl (1.8.9370.21113)
10: Library.Logic (1.8.9370.21095)
11: Library.Math (1.8.9370.21104)
12: Library.Other (1.8.9370.21126)
13: Library.Process (1.8.9370.21126)
14: Library.Time (1.8.9370.21136)
15: Library.Trends (1.8.9396.20931)
16: Library.VAV (1.8.9370.21450)
Tree
Applications
Application
Equipment
Application1
Networks
Services
System
Wire Sheet
Name
OK
Cancel

Name: AnalogDataPoint

Figure 117. Add (drag and drop) and rename Data Points

It is also possible to add Data Points directly from the context menu of the Application component:

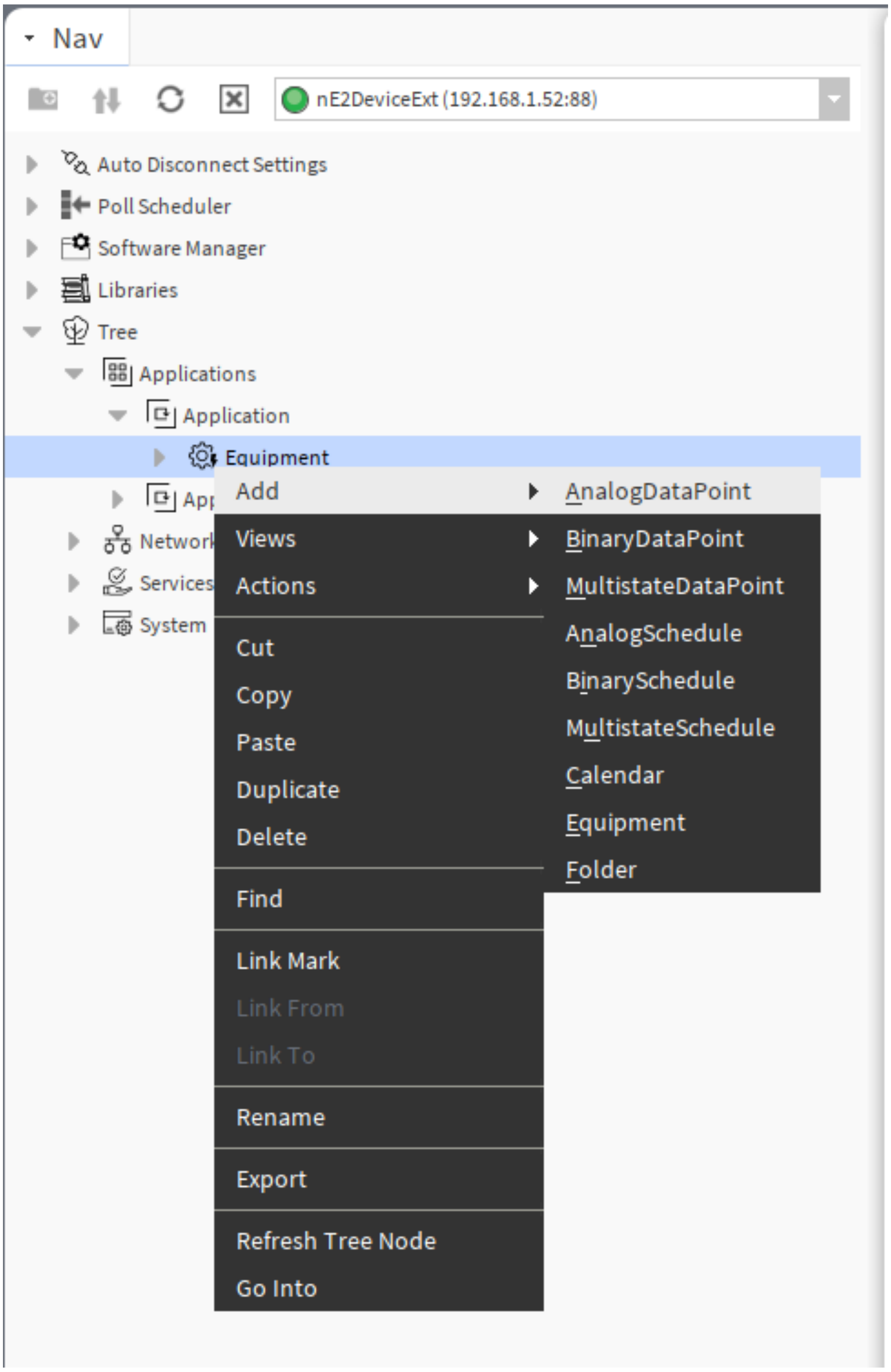


Figure 118. Adding Data Points directly from the context menu

Each Data Point has 3 actions available from the Actions menu. Actions are related with the type of the Data Point and its extensions.

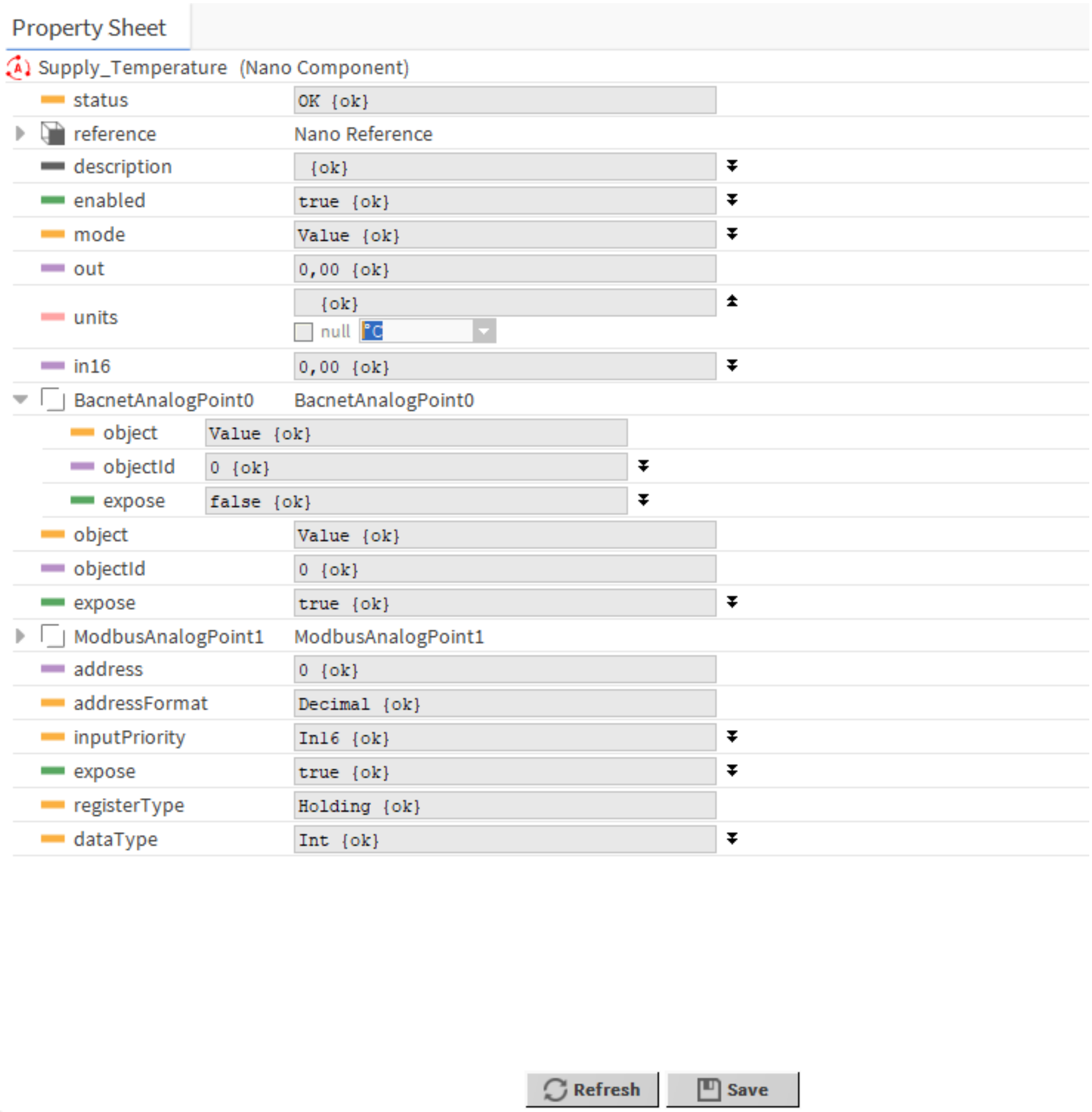
- Set: allows entering a value to set the In16 slot;
- SetId: sets a BACnet object Id of the Data Point (exposed in the BACnetPoint extension);
- SetAddress: sets a Modbus address of the Data Point (exposed in the ModbusPoint extension).

All slots and options are described in the nano EDGE ENGINE Programming user manual: AnalogDataPoint, BinaryDataPoint, MultistateDataPoint.

5.5.2 Data Points Configuration

To configure the added Data Point, go to its Property Sheet. Standard Data Point slots will be visible. The Property Sheet view allows to configure the following parameters:

- mode;
- units;
- extensions;
- other points available for the selected component.



Property Sheet

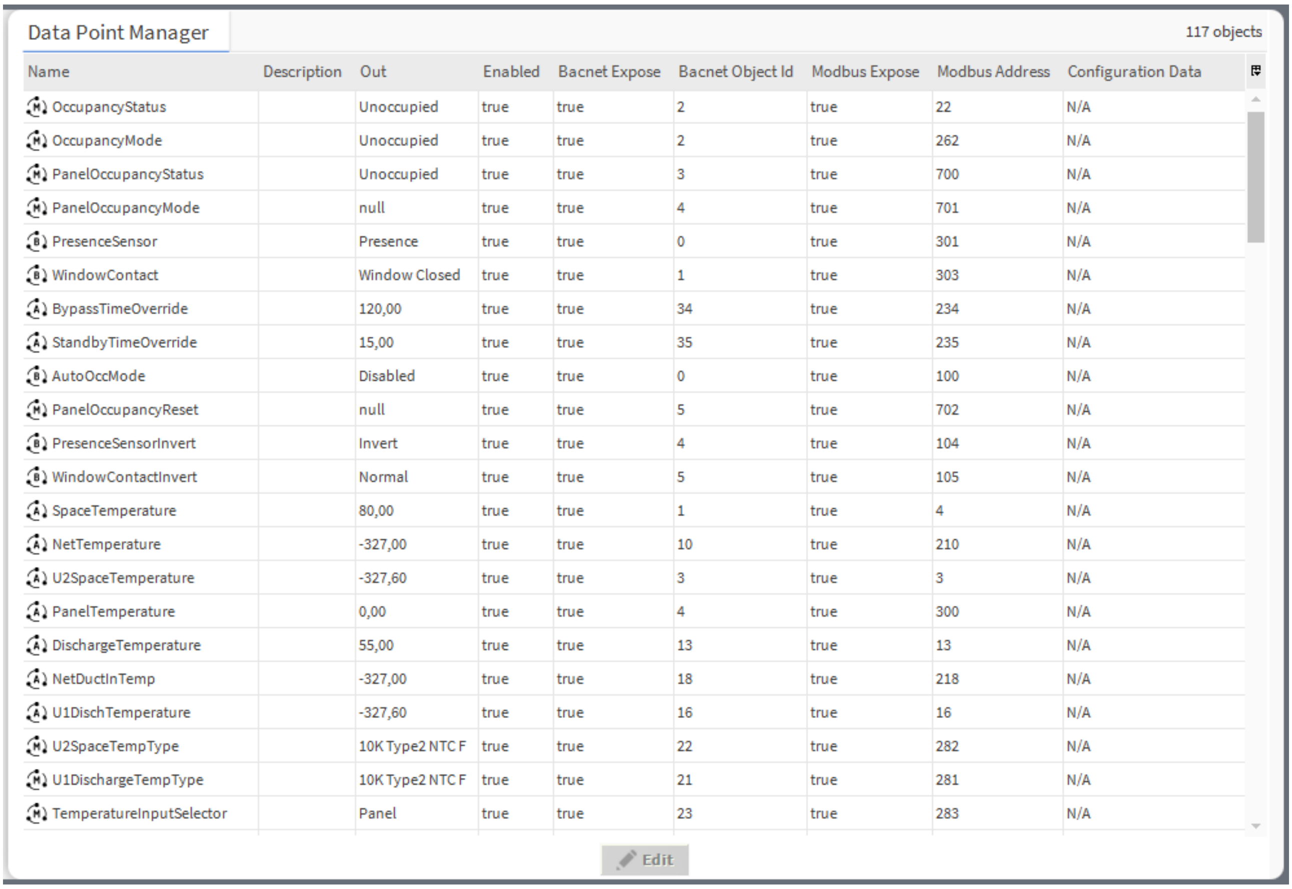
Supply_Temperature (Nano Component)

status	OK {ok}
reference	Nano Reference
description	{ok}
enabled	true {ok}
mode	Value {ok}
out	0,00 {ok}
units	{ok} □ null °C
in16	0,00 {ok}
BacnetAnalogPoint0	BacnetAnalogPoint0
object	Value {ok}
objectId	0 {ok}
expose	false {ok}
object	Value {ok}
objectId	0 {ok}
expose	true {ok}
ModbusAnalogPoint1	ModbusAnalogPoint1
address	0 {ok}
addressFormat	Decimal {ok}
inputPriority	In16 {ok}
expose	true {ok}
registerType	Holding {ok}
dataType	Int {ok}

Figure 119. Data Point's Property Sheet view

5.5.3 Data Point Manager

The Data Point Manager is a special view that allows to manage the Data Points available within the nano EDGE ENGINE license.



Data Point Manager

117 objects

Name	Description	Out	Enabled	Bacnet Expose	Bacnet Object Id	Modbus Expose	Modbus Address	Configuration Data
OccupancyStatus		Unoccupied	true	true	2	true	22	N/A
OccupancyMode		Unoccupied	true	true	2	true	262	N/A
PanelOccupancyStatus		Unoccupied	true	true	3	true	700	N/A
PanelOccupancyMode		null	true	true	4	true	701	N/A
PresenceSensor		Presence	true	true	0	true	301	N/A
WindowContact		Window Closed	true	true	1	true	303	N/A
BypassTimeOverride		120,00	true	true	34	true	234	N/A
StandbyTimeOverride		15,00	true	true	35	true	235	N/A
AutoOccMode		Disabled	true	true	0	true	100	N/A
PanelOccupancyReset		null	true	true	5	true	702	N/A
PresenceSensorInvert		Invert	true	true	4	true	104	N/A
WindowContactInvert		Normal	true	true	5	true	105	N/A
SpaceTemperature		80,00	true	true	1	true	4	N/A
NetTemperature		-327,00	true	true	10	true	210	N/A
U2SpaceTemperature		-327,60	true	true	3	true	3	N/A
PanelTemperature		0,00	true	true	4	true	300	N/A
DischargeTemperature		55,00	true	true	13	true	13	N/A
NetDuctInTemp		-327,00	true	true	18	true	218	N/A
U1DischTemperature		-327,60	true	true	16	true	16	N/A
U2SpaceTempType		10K Type2 NTC F	true	true	22	true	282	N/A
U1DischargeTempType		10K Type2 NTC F	true	true	21	true	281	N/A
TemperatureInputSelector		Panel	true	true	23	true	283	N/A

Figure 120. Data Point Manager

The Data Point Manager lists all the Data Points used in applications saved on the device. The view shows the following fields:

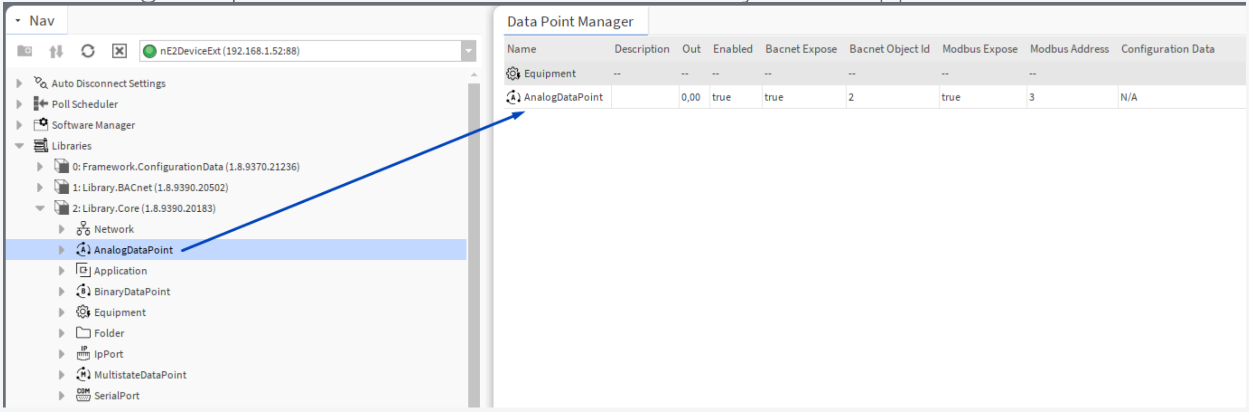
- name of the Data Point;
- description;
- value on the Out slot;
- enabled or disabled status;
- exposed on BACnet status;
- BACnet object Id;
- exposed on Modbus status;
- Modbus address;
- Configuration Data extension status.

In the Data Point Manager, it is possible to:

- add Data Point components,

Methods of adding Data Points

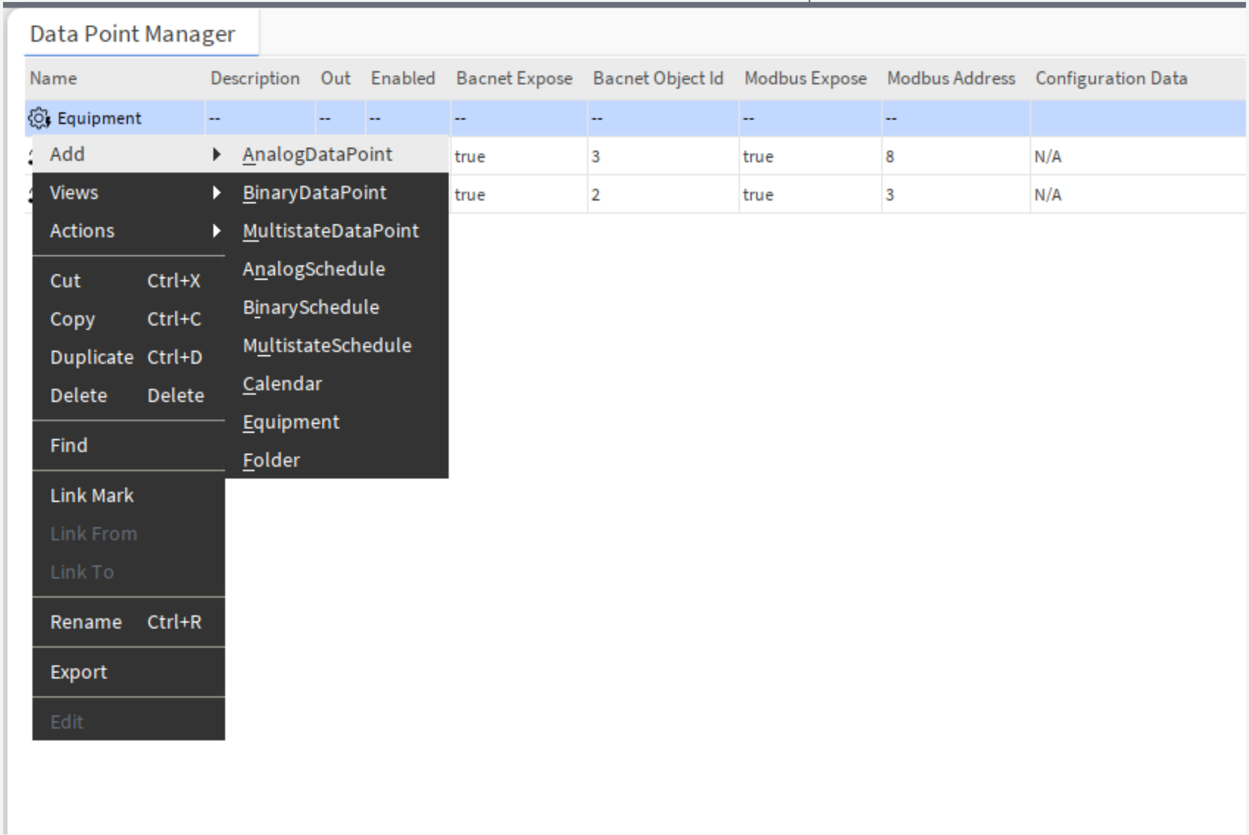
Use a drag&drop method to add Data Points directly to the Application:



Name	Description	Out	Enabled	Bacnet Expose	Bacnet Object Id	Modbus Expose	Modbus Address	Configuration Data
Equipment	--	--	--	--	--	--	--	
AnalogDataPoint		0,00	true	true	2	true	3	N/A

Figure 121. Adding Data Point - drag and drop method

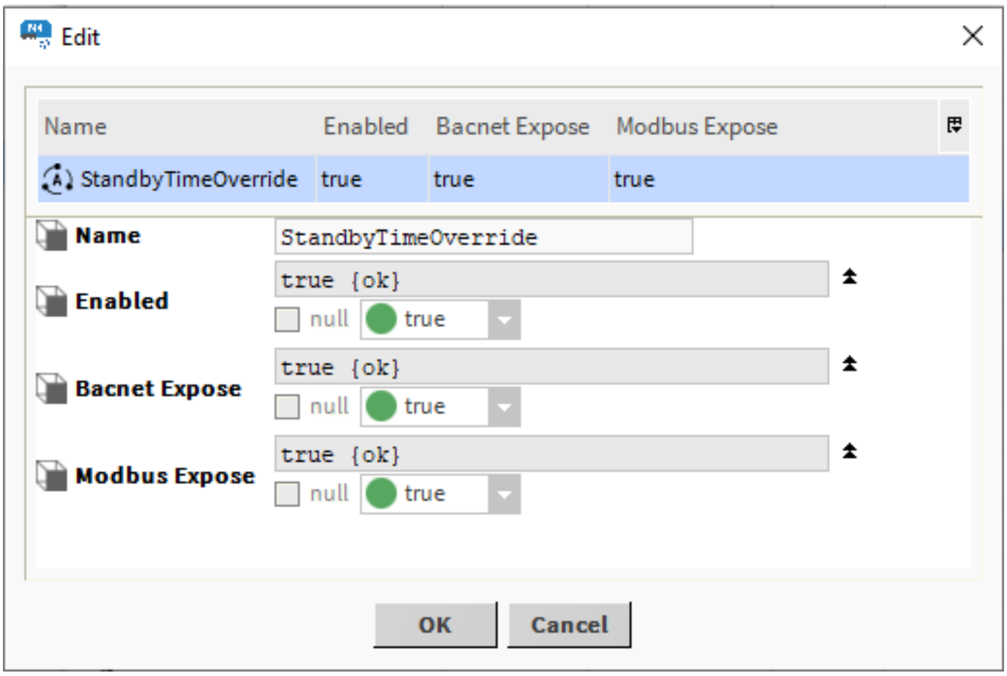
Use a context menu to add Data Points as **children** components:



Name	Description	Out	Enabled	Bacnet Expose	Bacnet Object Id	Modbus Expose	Modbus Address	Configuration Data
Equipment	--	--	--	--	--	--	--	
				true	3	true	8	N/A
				true	2	true	3	N/A

Figure 122. Adding Data Point as a child component to the Equipment component

- edit the Data Point's name and BACnet/Modbus exposition and enable/disable the component:



Name	Enabled	Bacnet Expose	Modbus Expose
StandbyTimeOverride	true	true	true

Name: StandbyTimeOverride

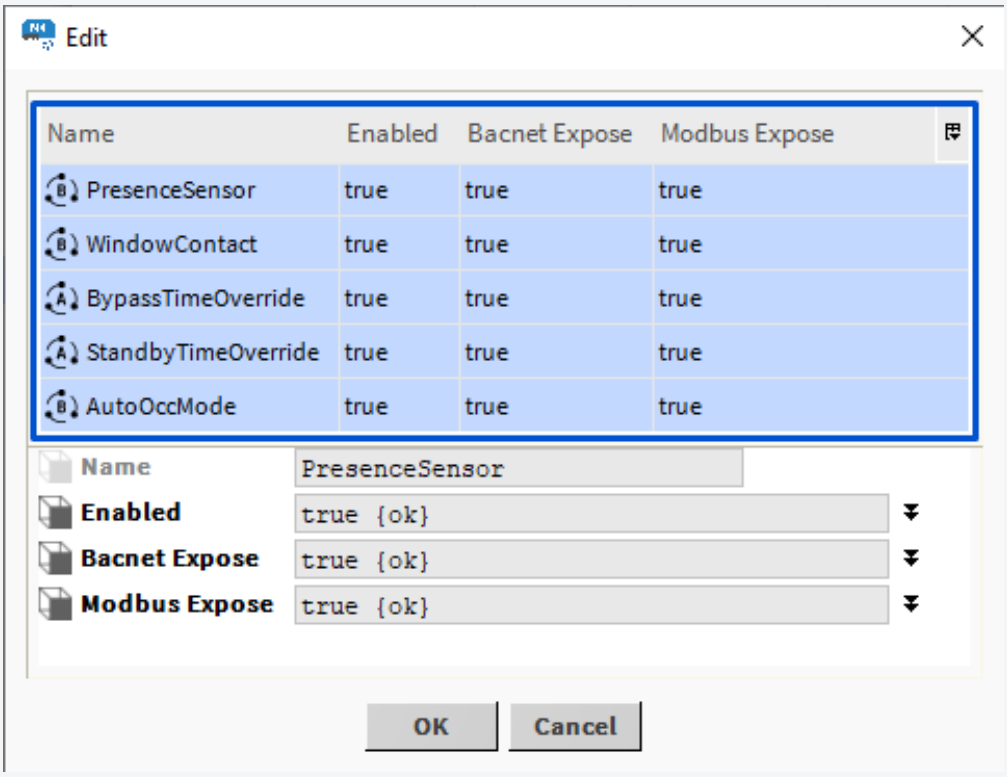
Enabled: true {ok} — □ null ● true

Bacnet Expose: true {ok} — □ null ● true

Modbus Expose: true {ok} — □ null ● true

Figure 123. Edit pop-up

Note
Editing is possible for more than one Data Point at a time. If multiple Data Points are edited, the same new value is written to common slots, so individual slots, such as Name, cannot be edited in this manner.



Name	Enabled	Bacnet Expose	Modbus Expose
PresenceSensor	true	true	true
WindowContact	true	true	true
BypassTimeOverride	true	true	true
StandbyTimeOverride	true	true	true
AutoOccMode	true	true	true

Name: PresenceSensor

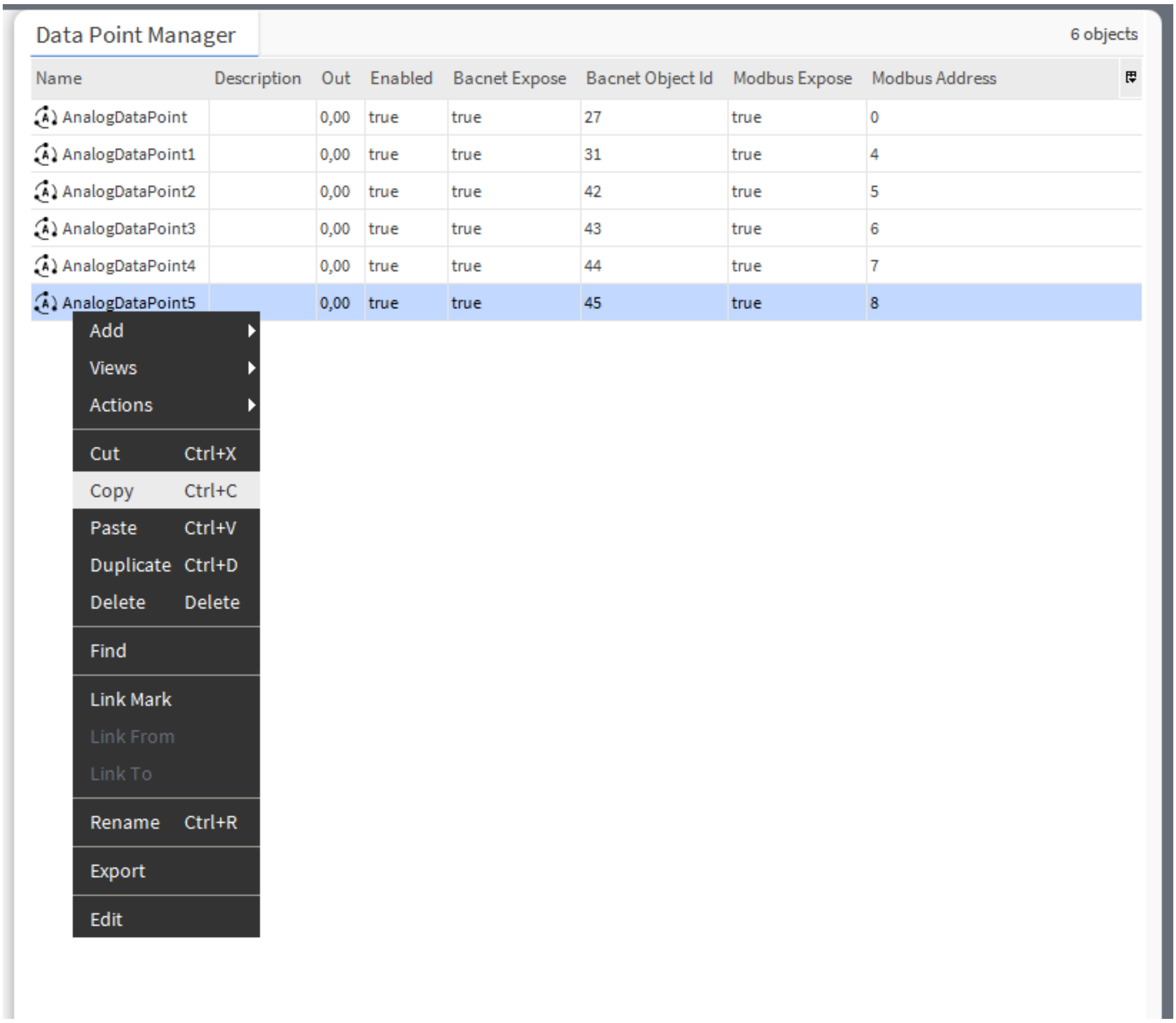
Enabled: true {ok}

Bacnet Expose: true {ok}

Modbus Expose: true {ok}

Figure 124. Editing of multiple Data Points

- copy/cut/duplicate/remove Data Point components:



Data Point Manager

6 objects

Name	Description	Out	Enabled	Bacnet Expose	Bacnet Object Id	Modbus Expose	Modbus Address
AnalogDataPoint		0,00	true	true	27	true	0
AnalogDataPoint1		0,00	true	true	31	true	4
AnalogDataPoint2		0,00	true	true	42	true	5
AnalogDataPoint3		0,00	true	true	43	true	6
AnalogDataPoint4		0,00	true	true	44	true	7
AnalogDataPoint5		0,00	true	true	45	true	8

Figure 125. Context menu options for the Data Point

Opening the Data Point Manager

The Data Point Manager view is accessible from two locations:

- in the context menu of the Applications container;
- in the context menu of the Application component;
- in the context menu of the Equipment container.

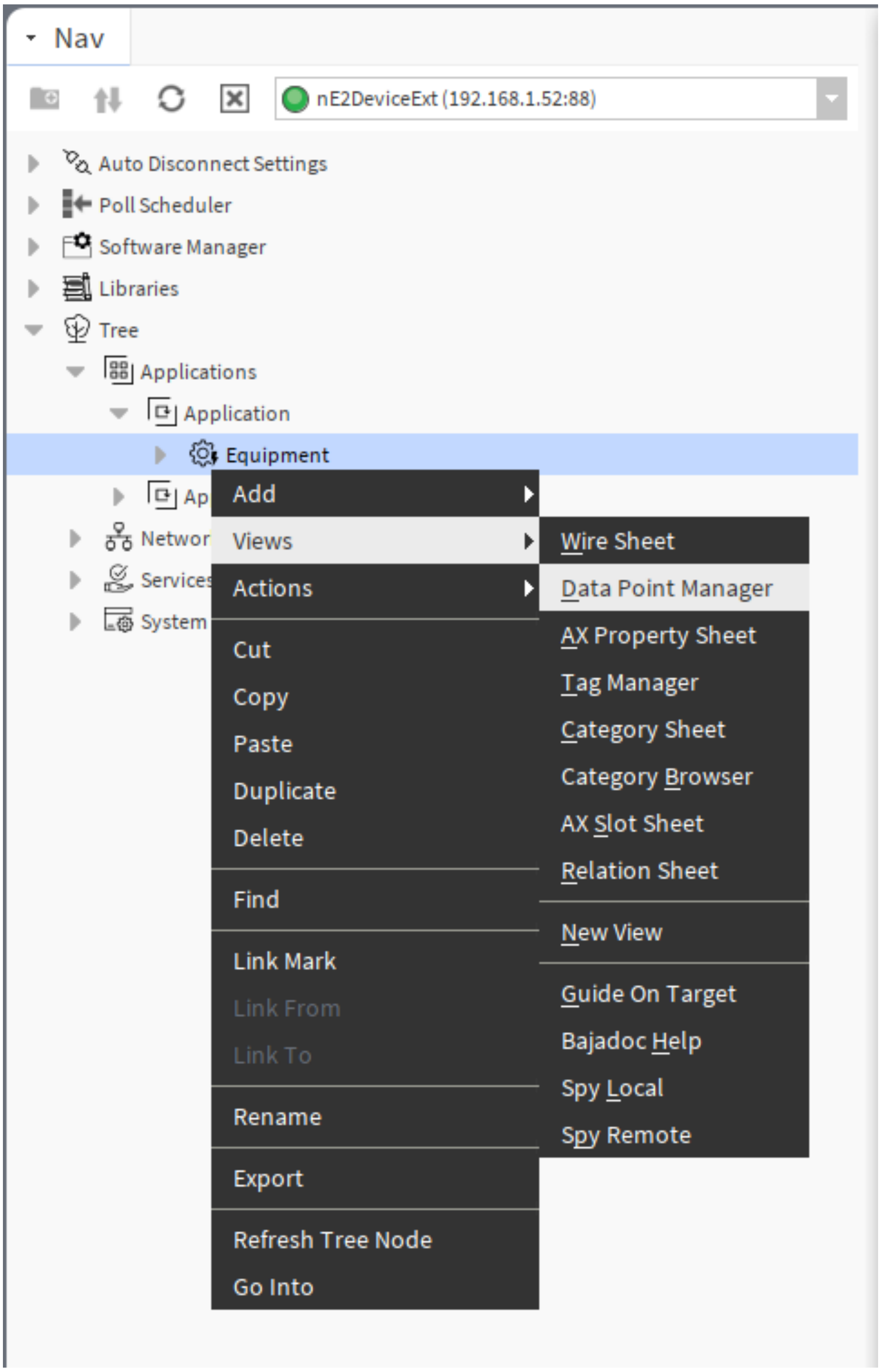


Figure 126. Accessing the Data Point Manager in the context menu

Licensing

The license for the new generation of iSMA CONTROLLI controllers driven by the nano EDGE ENGINE is constructed against the number of Data Points: each device based on the nano EDGE ENGINE is granted a specified number of license points (Data Points in this case), which can be used within applications. Therefore, the licensing system is only of quantitative, not functional, character–only the real number of Data Points in applications is taken into account, regardless of how many communication protocols are used to expose them, or how many network points are controlled. With the nano EDGE ENGINE-generation devices it is possible to create as big an application (or applications) as the number of licensed Data Points. No elements in the Networks, Services, or System containers are subject to license limitations, other than Data Points in the Applications container.

Note: In order to check the number of license points, please refer to the License in the device.

5.6 Extensions to Data Points

Data Points can have their functionality enhanced by extensions. For example, the AnalogDataPoint is originally equipped with the BACnetAnalogPoint and ModbusAnalogPoint extensions (these cannot be added or removed), but other extensions, which offer different functionalities, can be added or removed as necessary.

Extensions are added from the context menu, select the Add Extension option; add the extension from the list of available options.

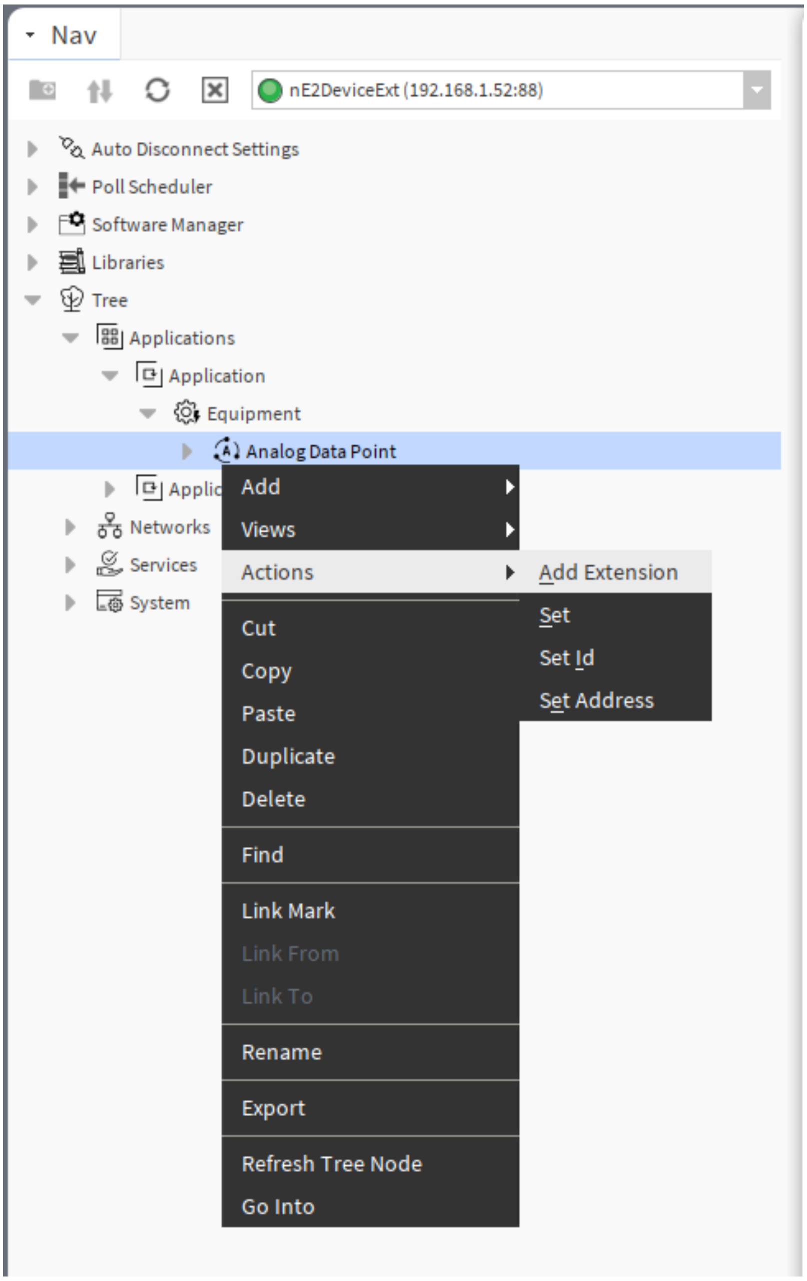


Figure 127. Adding extension in the workspace tree

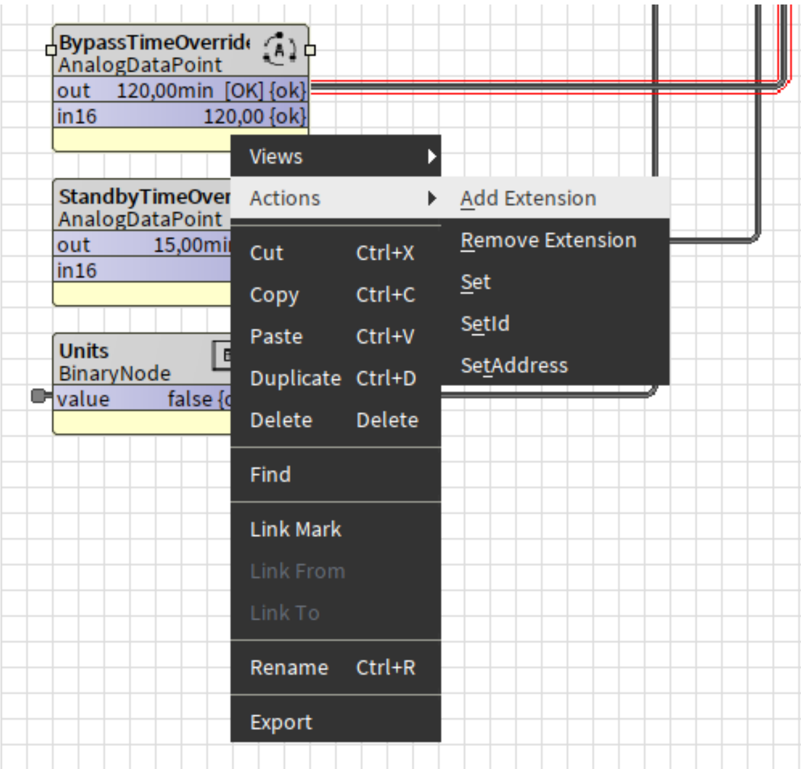


Figure 128. Adding extension in the wire sheet

5.6.1 BACnetPoint

The BACnetAnalogPoint/BACnetBinaryPoint/BACnetMultistatePoint extension expands the Data Point's functionality giving it an option to expose it to the BACnet IP network as an Analog/Binary/Multistate Value object, and, otherwise, it allows to manually hide it from the network yet preserving its function in the application. It also transfers information to the BACnet IP network about the Data Point's status. The extension is native (cannot be removed), and is visible along with the regular slots and actions of the Data Point as a separate, integral part in the property sheet view.

The extension has the following slots:

- Object: a read-only slot showing a type of BACnet object attributed to the extension;
- ObjectID: a BACnet object ID, which is automatically numbered from 0 up;
- Expose: enables the Data Point to be recognized within the BACnet IP network;
    - Available settings: true (exposed), false (hidden).

5.6.2 ModbusPoint

The ModbusAnalogPoint/ModbusBinaryPoint/ModbusMultistatePoint extension expands the Data Point's functionality giving it an option to expose it to the Modbus TCP/IP network as a Modbus point, and, otherwise, it allows to manually hide it from the network yet preserving its function in the application. It also transfers information to the Modbus TCP/IP network about the Data Point's status. The extension is native (cannot be removed), and is visible along with the regular slots and actions of the Data Point as a separate, integral part in the property sheet view.

The extension has the following slots:

- **Address:** a read-only slot showing a Modbus register, which the Data Point is exposed on;
- **Address Format:** a read-only slot showing a register address format;
    - Available information: decimal, Modbus, HEX;
- **Input Priority:** allows to select the input number in the Data Point, which the value from the register is synchronized on;
- **Expose:** enables the Data Point to be recognized within the Modbus TCP/IP network;
    - Available settings: true (exposed), false (hidden);
- **Register:** a read-only slot showing the type of the register used;
    - Available information: holding register;
- **Data Type:** allows to select a value data type;
    - Available settings: integer (default), signed integer, long, signed long, float, double.

5.6.3 Trend

The Trend extension allows to save values of the Data Point for the historical analysis purposes. When added and enabled, the extension records values in a selected sampling mode and saves them in a database. The data saved from the Trend extension can be managed in the Trends service (available in the Services container).

It is not possible to add more than one Trend extension to the Data Point.

The extension is added from the context menu of the Data Point.

Note

For the Trends extension to operate properly, please make sure that the Trends library is installed on the device.

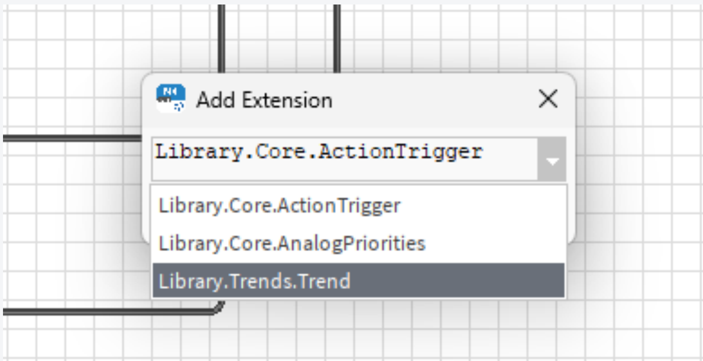
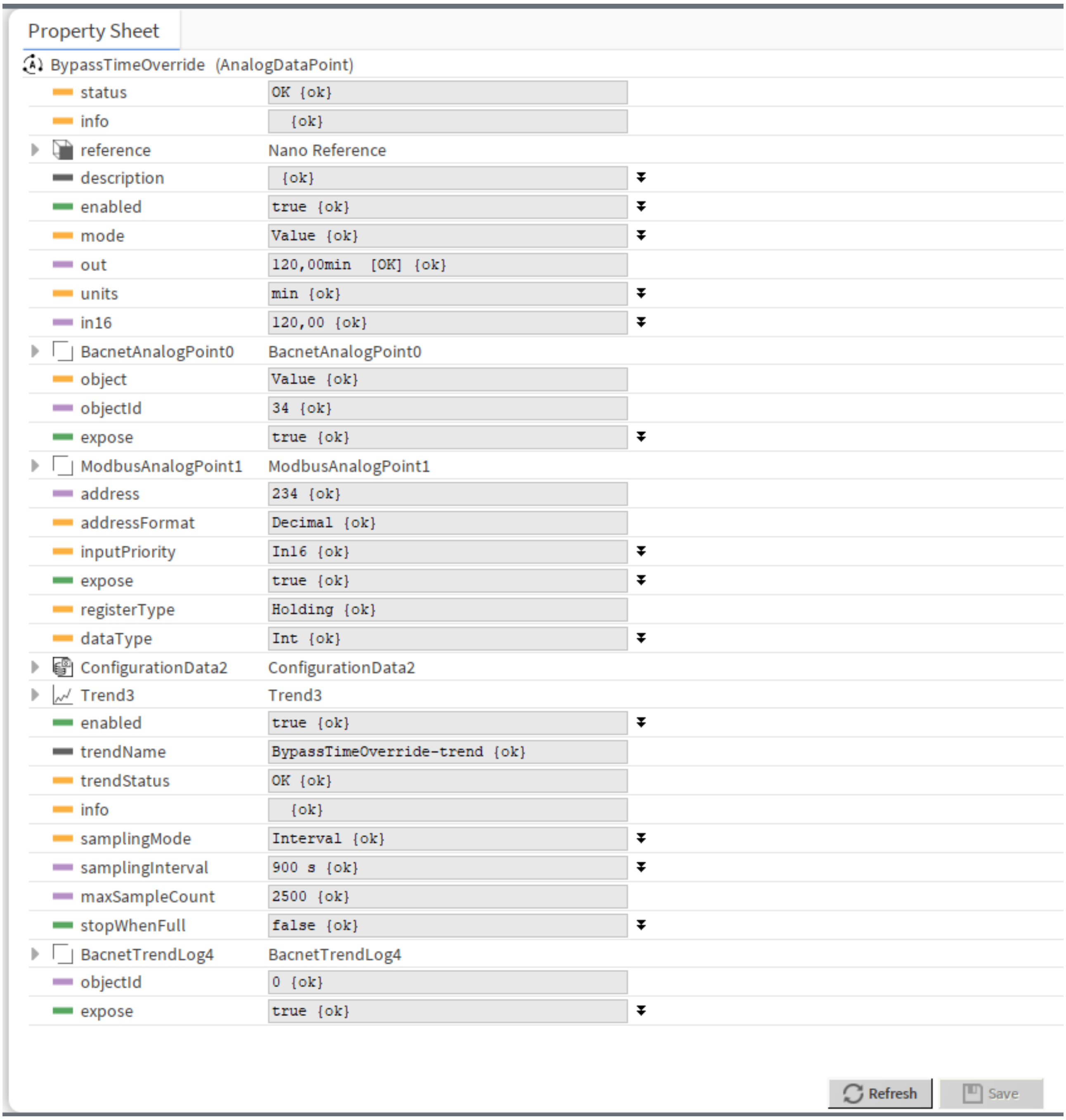

Add Extension
Library.Core.ActionTrigger
Library.Core.ActionTrigger
Library.Core.AnalogPriorities
Library.Trends.Trend

Figure 129. Adding Trends extension

The Trends library is by default added in the nano EDGE ENGINE V1.8 OS (and is not compatible with earlier versions) and the Trends Service is then automatically displayed in the Services container once the library is correctly installed on the device.



BypassTimeOverride (AnalogDataPoint)

status	OK {ok}
info	{ok}
reference	Nano Reference
description	{ok}
enabled	true {ok}
mode	Value {ok}
out	120,00min [OK] {ok}
units	min {ok}
in16	120,00 {ok}
BacnetAnalogPoint0	BacnetAnalogPoint0
object	Value {ok}
objectId	34 {ok}
expose	true {ok}
ModbusAnalogPoint1	ModbusAnalogPoint1
address	234 {ok}
addressFormat	Decimal {ok}
inputPriority	In16 {ok}
expose	true {ok}
registerType	Holding {ok}
dataType	Int {ok}
ConfigurationData2	ConfigurationData2
Trend3	Trend3
enabled	true {ok}
trendName	BypassTimeOverride-trend {ok}
trendStatus	OK {ok}
info	{ok}
samplingMode	Interval {ok}
samplingInterval	900 s {ok}
maxSampleCount	2500 {ok}
stopWhenFull	false {ok}
BacnetTrendLog4	BacnetTrendLog4
objectId	0 {ok}
expose	true {ok}

Figure 130. Trend service extension added to the AnalogDataPoint

The Trend extension has the following slots:

- Enabled: allows to enable or disable the trend;
- Trend Name: shows the name of the trend;
- Trend Status: informs of the extension's status;
    - Available information:
        - Disabled (the Data Point, Trend extension, or Trends service is disabled),
        - Unlicensed (the Data Point is unlicensed),
        - Stopped (the number of saved samples is equal to or greater than Max Sample Count and the Stop When Full slot is set to true),
        - OK;
- Info: provides a detailed information about the Disabled status of the component;

    - Available information:
        - Trend extension disabled: the Enabled slot in the Trend extension is set to false,
        - Parent (Data Point) disabled: the Enabled slot in the Data Point is set to false,
        - Parent (Data Point) unlicensed: the Data Point, which the Trend extension is added to, is unlicensed,
        - Trends Service disabled: the Enabled slot in the Trends service is set to false;

From OS V1.8, the Info is a numerical slot, which displays a correlated text information:

Numerical value	Displayed information
0	No information displayed in the Info slot
1	Trend extension disabled
2	Parent (Data Point) disabled
3	Parent (Data Point) unlicensed
4	Trends Service disabled

- Sampling Mode: allows to set a sampling mode for saving data;
    - Available settings:
        - COV (triggers saving the sample when the Data Point's Out slot value changes and this change is bigger than the Deadband slot's value),
        - Interval (triggers saving the sample when time reaches the value of Sampling Interval slot's value),
        - COV Interval (triggers saving the sample in both the above cases);
- Sampling Interval: (available only if interval or COV interval mode is selected) time value, which triggers saving trend data;
- Deadband: (available only if COV or COV interval mode is selected, applicable only in the AnalogDataPoint) allows to set a value for a non-responsive zone; the trend data will only be saved if the COV exceeds the deadband value;
- Max Sample Count: shows a number of maximum (2500) samples that can be saved in the database for the trend;
- Stop When Full: allows to stop saving trend data if the Max Sample Count number is reached;
    - Available settings: false (trend overwrites the oldest samples), true (trend stops executing).

BACnet Trend Log

When the Trend extension is added to the Data Point, it automatically adds another extension in the Data Points view: the **BACnet Trend Log extension**. This extension allows to manage the trend's BACnet exposition to the network and adds a BACnet ID if exposed.

The BACnet Trend Log extension has the following slots:

- **Object Id:** shows an automatically assigned BACnet object ID number if the trend is exposed to the BACnet network;
- **Expose:** allows to enable or disable the trend's exposition to the BACnet network; by default, the trends is exposed to the network.

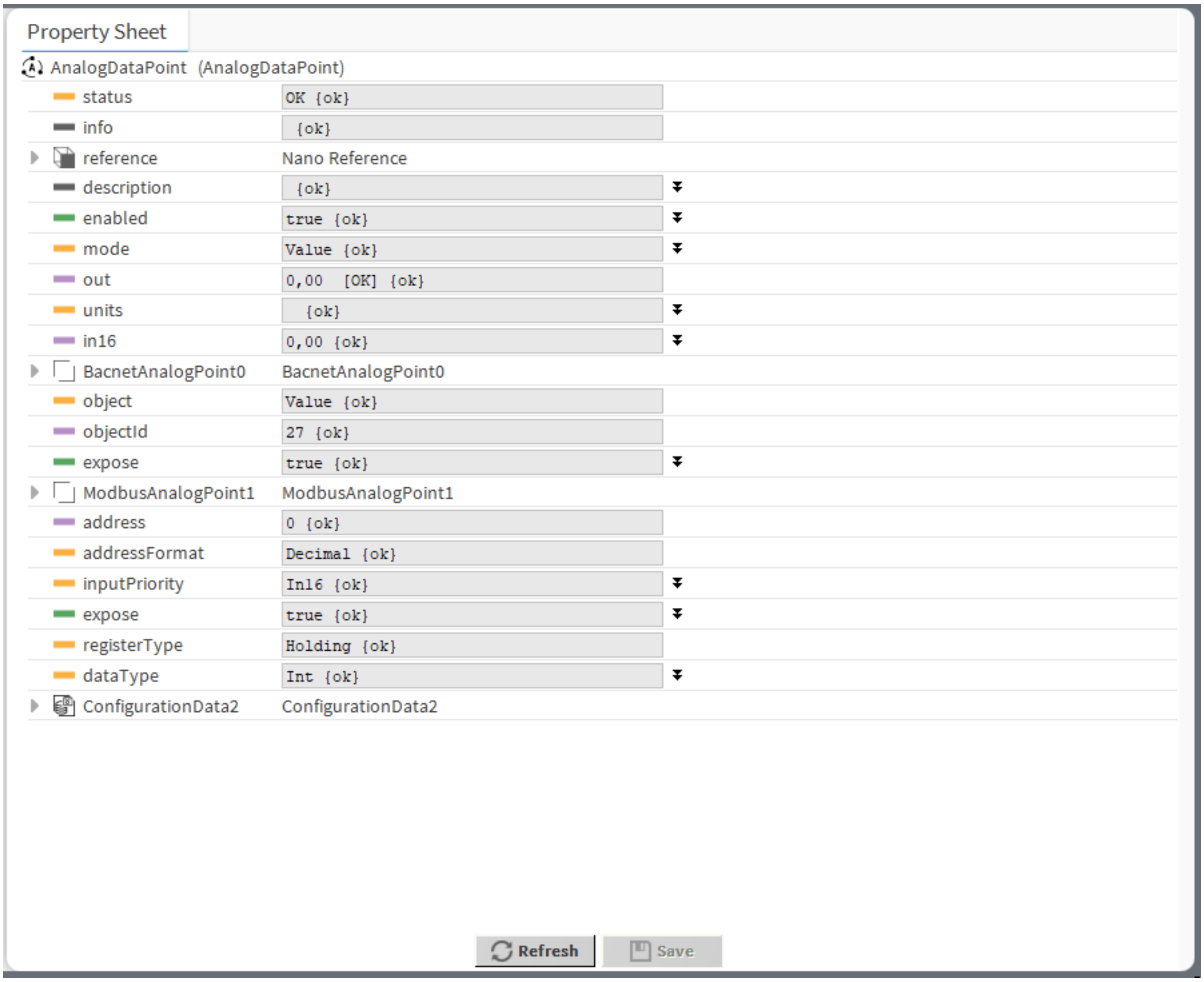
The BACnet Trend Log extension has the following action:

- **Set Trend Id:** allows to set the object ID value for the trend exposed to the BACnet network.

Note: The trend's exposition can also by enabled/disabled directly in the Trends service view, editing the Expose on Bacnet field.

5.6.4 ConfigurationData

The Configuration Data extension has no slots. Its functionality is fully achieved by adding it to the Data Point. It is automatically enabled and allows the Configuration Data service to save and upload slots values of the Data Point.



Property Sheet

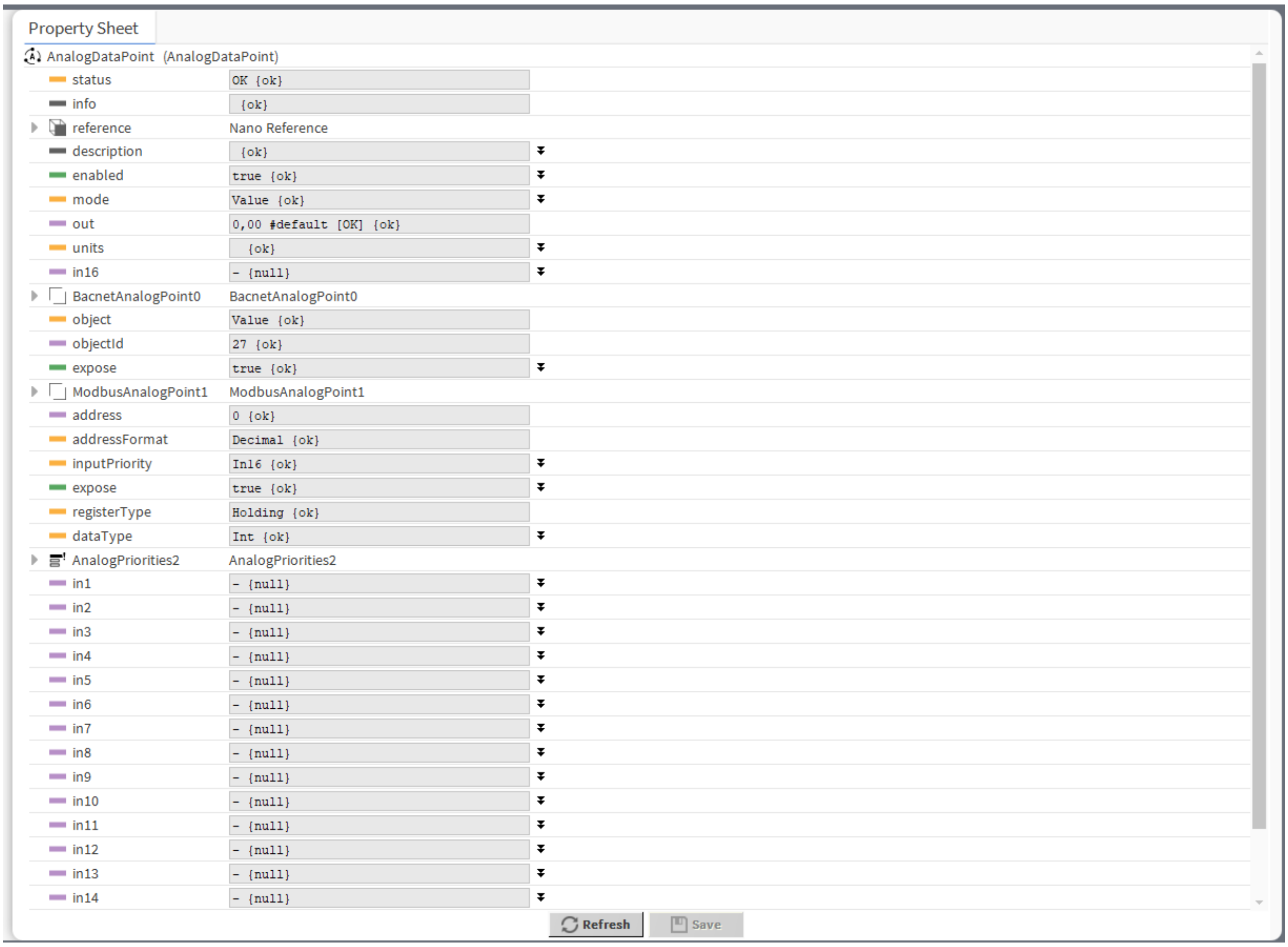
AnalogDataPoint (AnalogDataPoint)

status	OK {ok}
info	{ok}
reference	Nano Reference
description	{ok}
enabled	true {ok}
mode	Value {ok}
out	0,00 [OK] {ok}
units	{ok}
in16	0,00 {ok}
BacnetAnalogPoint0	BacnetAnalogPoint0
object	Value {ok}
objectId	27 {ok}
expose	true {ok}
ModbusAnalogPoint1	ModbusAnalogPoint1
address	0 {ok}
addressFormat	Decimal {ok}
inputPriority	In16 {ok}
expose	true {ok}
registerType	Holding {ok}
dataType	Int {ok}
ConfigurationData2	ConfigurationData2

Figure 131. ConfigurationData extension

5.6.5 Priorities

The AnalogPriorities/BinaryPriorities/MultistatePriorities extension adds fifteen writable input slots and the default (lowest) priority slot to the Data Point. The extension includes the Priority slot indicating, which slot is transferring value to the Out slot. The Priorites extension adds In1–In15 slots and the Default slot, which is the lowest, 17th priority. The extension also introduces new actions to the Data Point: EmergencyOverride, EmergencyAuto, Override, and OverrideAuto.



AnalogDataPoint (AnalogDataPoint)

status	OK {ok}
info	{ok}
reference	Nano Reference
description	{ok}
enabled	true {ok}
mode	Value {ok}
out	0,00 #default [OK] {ok}
units	{ok}
in16	- {null}
BacnetAnalogPoint0	BacnetAnalogPoint0
object	Value {ok}
objectId	27 {ok}
expose	true {ok}
ModbusAnalogPoint1	ModbusAnalogPoint1
address	0 {ok}
addressFormat	Decimal {ok}
inputPriority	In16 {ok}
expose	true {ok}
registerType	Holding {ok}
dataType	Int {ok}
AnalogPriorities2	AnalogPriorities2
in1	- {null}
in2	- {null}
in3	- {null}
in4	- {null}
in5	- {null}
in6	- {null}
in7	- {null}
in8	- {null}
in9	- {null}
in10	- {null}
in11	- {null}
in12	- {null}
in13	- {null}
in14	- {null}

Figure 132. Priorities extension

The extension has the following slots:

- In1-In15: input slots providing values to the Out slot (from 1 to 16, the highest priority is In1); only the highest priority value is provided to the Out slot, the rest is dismissed. All input slots are linkable. In the extended mode, the In1 and In8 slots have actions available for overriding their values.

Note: By default, only the In16 is displayed in the Wire Sheet. In case any other input slot receives a value via link, is it displayed in the Wire Sheet along with the In16. Only the null input, which is a lack of value, allows the higher priority input to be dismissed–zero (0) is still a value that will be provided to the Out slot.

- Default: the 17th, lowest priority input slot; allows to introduce a default value to the Data Point in case there are no links providing values from other components. If the value to the Data Point is provided by the Reference link, then the Default value is automatically dismissed (the Reference link cannot be directed to the 17th priority, only from the 16th up).

Note: According to BACnet requirements, the Default slot value can never be null; if no other value is set on the slot, it is zero (0).

- Priority: shows, which slot is currently providing the value to the Out slot.

The Data Point has the following actions available in the Priorities extension:

- EmergencyOverride: enables entering an analog value to the In1 slot;

- EmergencyAuto: sets the null value to the In1 slot (cancels the EmergencyOverride action);
- Override: enables entering an analog value to the In8 slot;
- OverrideAuto: sets the null value to the In8 slot (cancels the Override action).

Note: If the link is connected to the slot that may be affected by an action, the value coming from the link connection has priority over the manually evoked action.

5.6.6 ActionTrigger

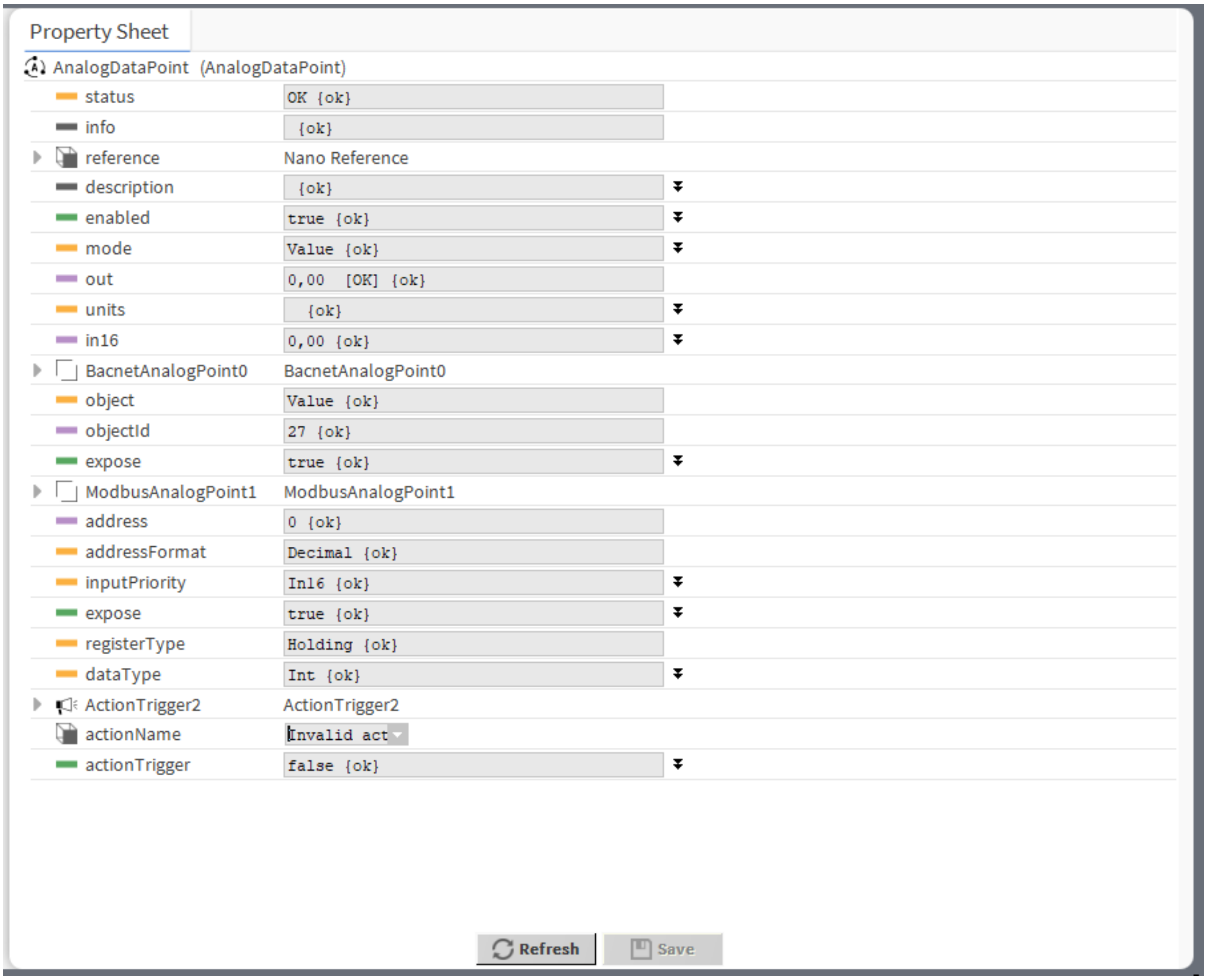
The ActionTrigger extension is designed to invoke any action that is available for the component. The extension triggers an action selected in the Action Name on the rising edge of the Action Trigger slot. If the action has parameters to set, the parameter is taken from a relevant slot automatically added to the extension (Analog Value/Binary Value/ String Value).

It is possible to add more than one ActionTrigger extension to the component (for example, one for each action in the component).

The extension is added from the context menu of the component.

The ActionTrigger extension has the following slots:

- Action Name: allows to select an action to invoke;
- Action Trigger: triggers an action selected in the Action Name slot;
- Action Analog Value/Action Binary Value/Action String Value: a slot added automatically to the extension if an action selected in the Action Name slot has any specific parameters to set (depending on the type of action and its parameters, the relevant type of value is matched).



Property Sheet

AnalogDataPoint (AnalogDataPoint)

status	OK {ok}
info	{ok}
reference	Nano Reference
description	{ok}
enabled	true {ok}
mode	Value {ok}
out	0,00 [OK] {ok}
units	{ok}
in16	0,00 {ok}
BacnetAnalogPoint0	BacnetAnalogPoint0
object	Value {ok}
objectId	27 {ok}
expose	true {ok}
ModbusAnalogPoint1	ModbusAnalogPoint1
address	0 {ok}
addressFormat	Decimal {ok}
inputPriority	In16 {ok}
expose	true {ok}
registerType	Holding {ok}
dataType	Int {ok}
ActionTrigger2	ActionTrigger2
actionName	Invalid act
actionTrigger	false {ok}

Figure 133. ActionTrigger extension

5.7 Services

The Services provide a space for additional services developed to enhance the device's functionalities. Services may be added to the device and then used within applications. They are designed to provide additional functionalities to the basic algorithms included in applications, allowing the device to communicate with systems superior to building automation systems.

- Configuration Data Service
- Trends Service
- Tagging Service
- Web Service
- Haystack Service

5.7.1 Configuration Data Service

The Configuration Data is a service designed to save configuration settings of Data Points for the purpose of restoring them if changed or lost. The service is executed by adding the Configuration Data extension to Data Points:

- AnalogDataPoint;
- BinaryDataPoint;

- MultistateDataPoint.

The service functions as a backup mechanism for device-specific configurations – it can bring back saved values of Data Points within a single device. To transfer applications between devices, use Backups.

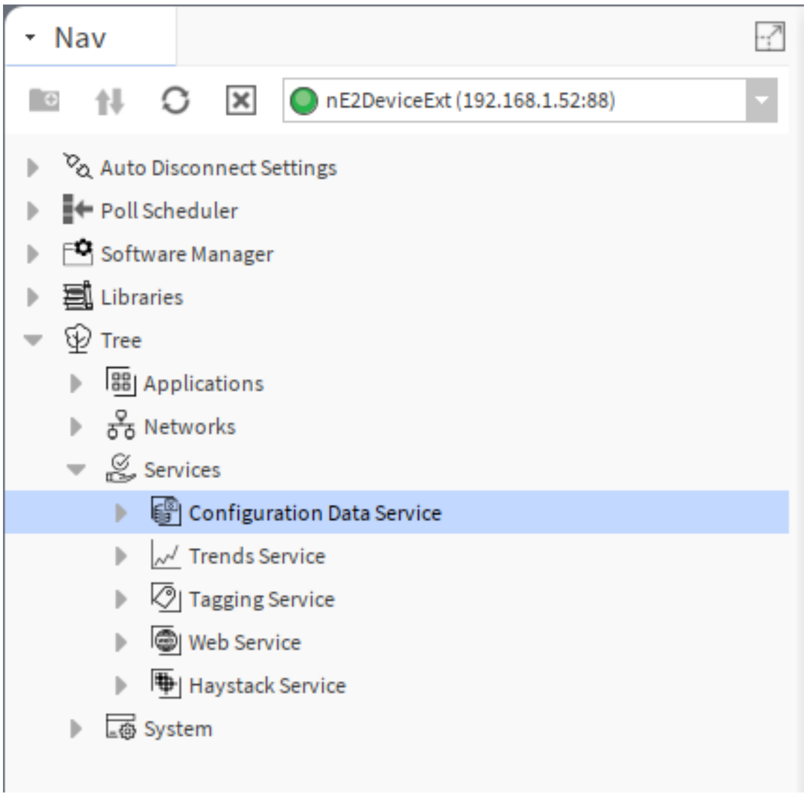
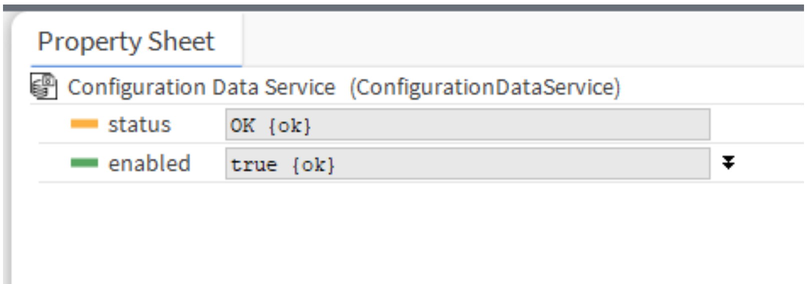


Figure 134. Configuration Data service in the nav tree


Property Sheet

Configuration Data Service (ConfigurationDataService)

status	OK {ok}
enabled	true {ok}

Figure 135. Configuration data service slots

The Configuration Data service has the following slots:

- **Status:** indicates the current status of the component. If the component works properly, its status is OK; however, it changes accordingly when values in other slots are adjusted.
    - Available information: disabled (the Enabled slot is set to false), OK;
- **Enabled:** change of the slot's value enables or disables the component.

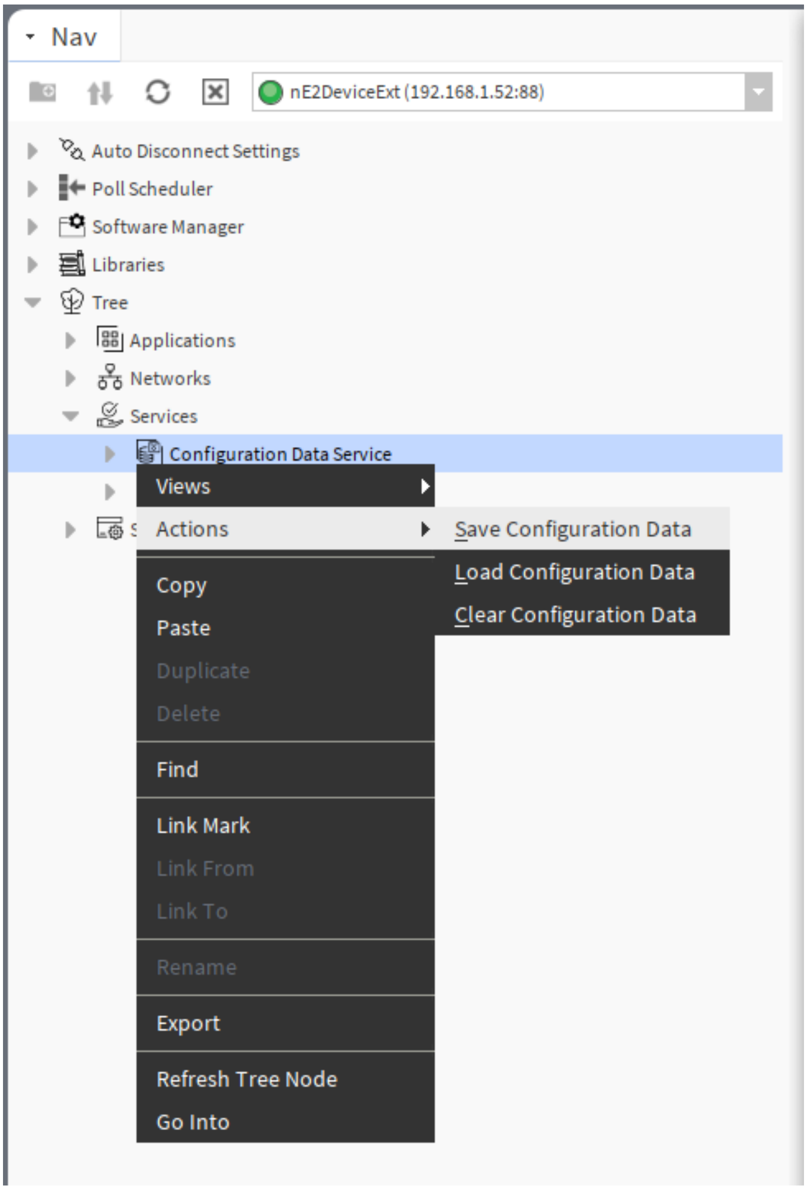
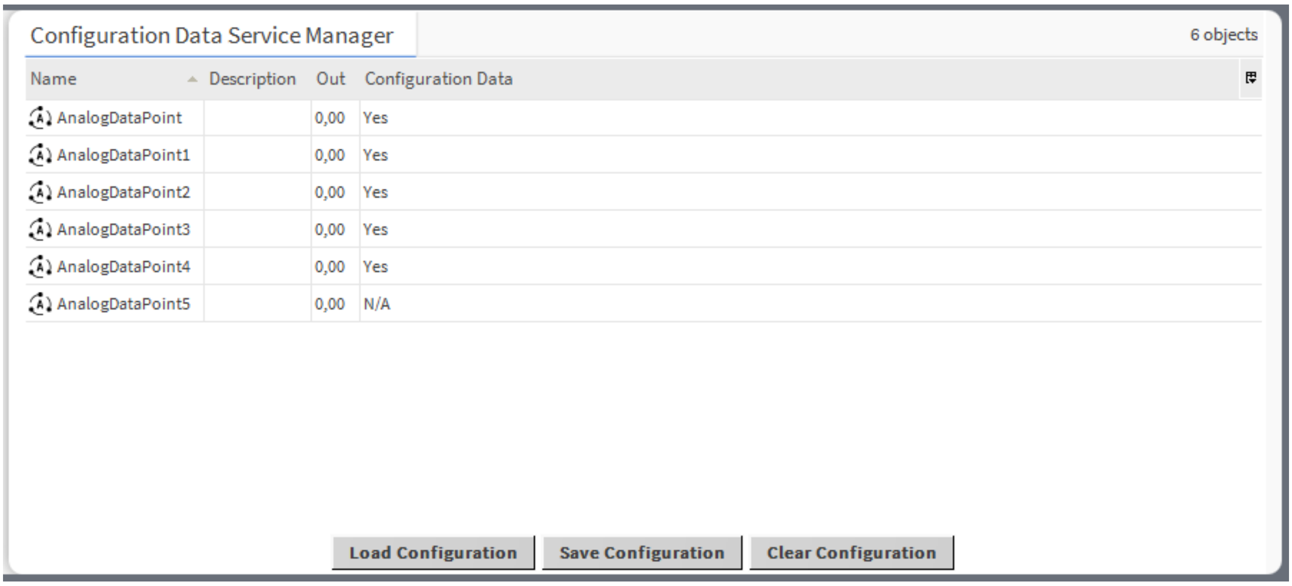


Figure 136. Configuration Data service actions

Configuration Data Service Manager



Configuration Data Service Manager

6 objects

Name	Description	Out	Configuration Data
AnalogDataPoint		0,00	Yes
AnalogDataPoint1		0,00	Yes
AnalogDataPoint2		0,00	Yes
AnalogDataPoint3		0,00	Yes
AnalogDataPoint4		0,00	Yes
AnalogDataPoint5		0,00	N/A

Figure 137. Configuration Data service manager

The Configuration Data service view is a simple table view showing which Data Points have the Configuration Data extension added, along with their Description and Out slots.

The Data Points marked N/A in the view are Data Points added to applications but without the Configuration Data extensions, as the service collects data only from Data Points with added extension. The view shows Data Points from all applications executed in the Applications container (if gathered under the Equipment component, double-click it to show Data Points). Additionally, it allows to export the gathered data with the Export option in the context menu.

The Configuration Data service has the following actions:

Note
Actions are executed for all applicable Data Points at once.

- Save Configuration Data: saves the slots values of Data Points with added Configuration Data extension to the controller's memory;
- Load Configuration Data: uploads the saved slots values to Data Points with added Configuration Data extension;

Note: The data can be loaded only to Data Points, which had the extension added at the point of saving the values.

- Clear Configuration Data: erases the saved slots values of Data Points with added Configuration Data extension.

Warning!
Remember that restoring default settings on the controller by the 6th DIP switch clears the values saved in the Configuration Data service too.

Configuration Data Extension

The Configuration Data extension has no slots. Its functionality is fully achieved by adding it to the Data Point. It is automatically enabled and allows the Configuration Data service to save and upload slots values of the Data Point.

5.7.2 Trends Service

The Trends service provides a simplified management of one or many Trend and BACnetTrendLog extensions to Data Points. It allows to manage historical data coming from Data Points and enable or disable saving these data from a specific Data Point.

Trends are executed by adding a dedicated extension to Data Points:

- Trend extension to the AnalogDataPoint;
- Trend extension to the BinaryDataPoint;
- Trend extension to the MultistateDataPoint.

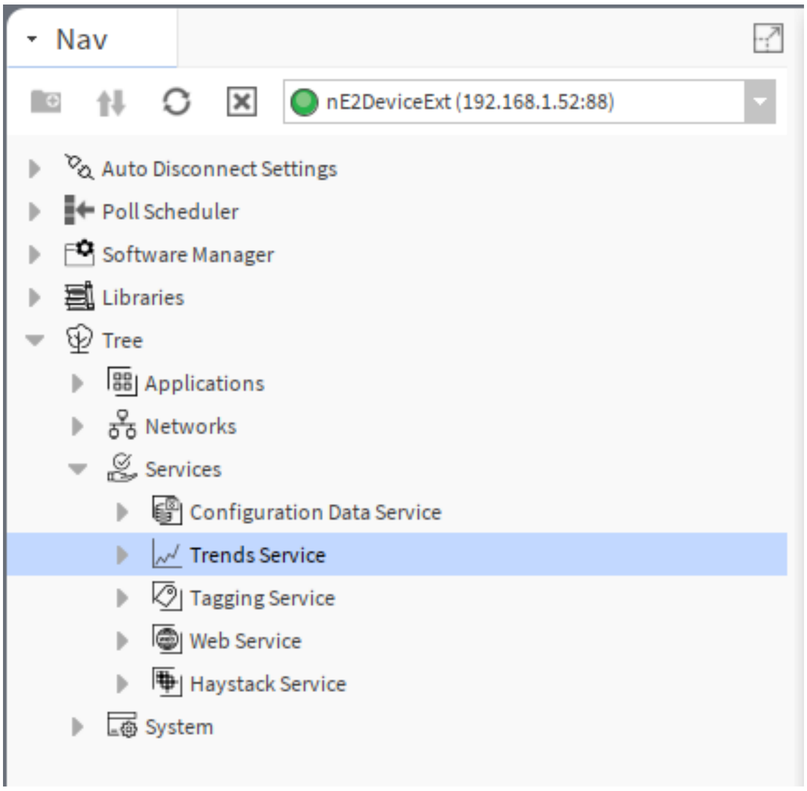
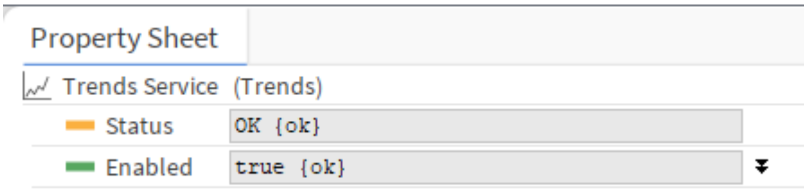


Figure 138. Trends service in the nav tree



Trends Service (Trends)

Status	OK {ok}
Enabled	true {ok}

Figure 139. Trends service slots

The Trends service has the following slots:

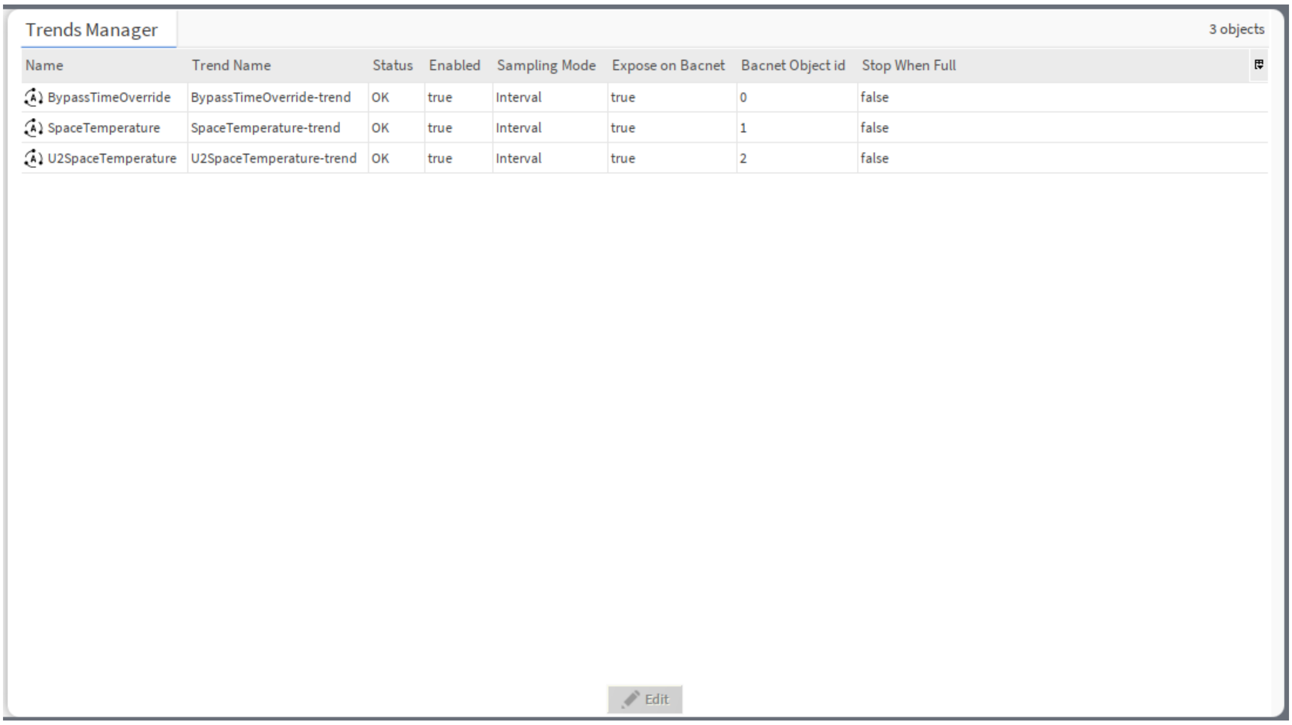
- **Status:** indicates the current status of the component. If the component works properly, its status is OK; however, it changes accordingly when values in other slots are adjusted.
    - Available information: disabled (the Enabled slot is set to false), OK;

- **Enabled:** change of the slot's value enables or disables the component.

The Trends Service is automatically available in the Services container once the Trends library is installed on the device. The Trends library is a default part of the nano EDGE ENGINE from the OS V1.8 and is not compatible with previous versions.

Trends Manager

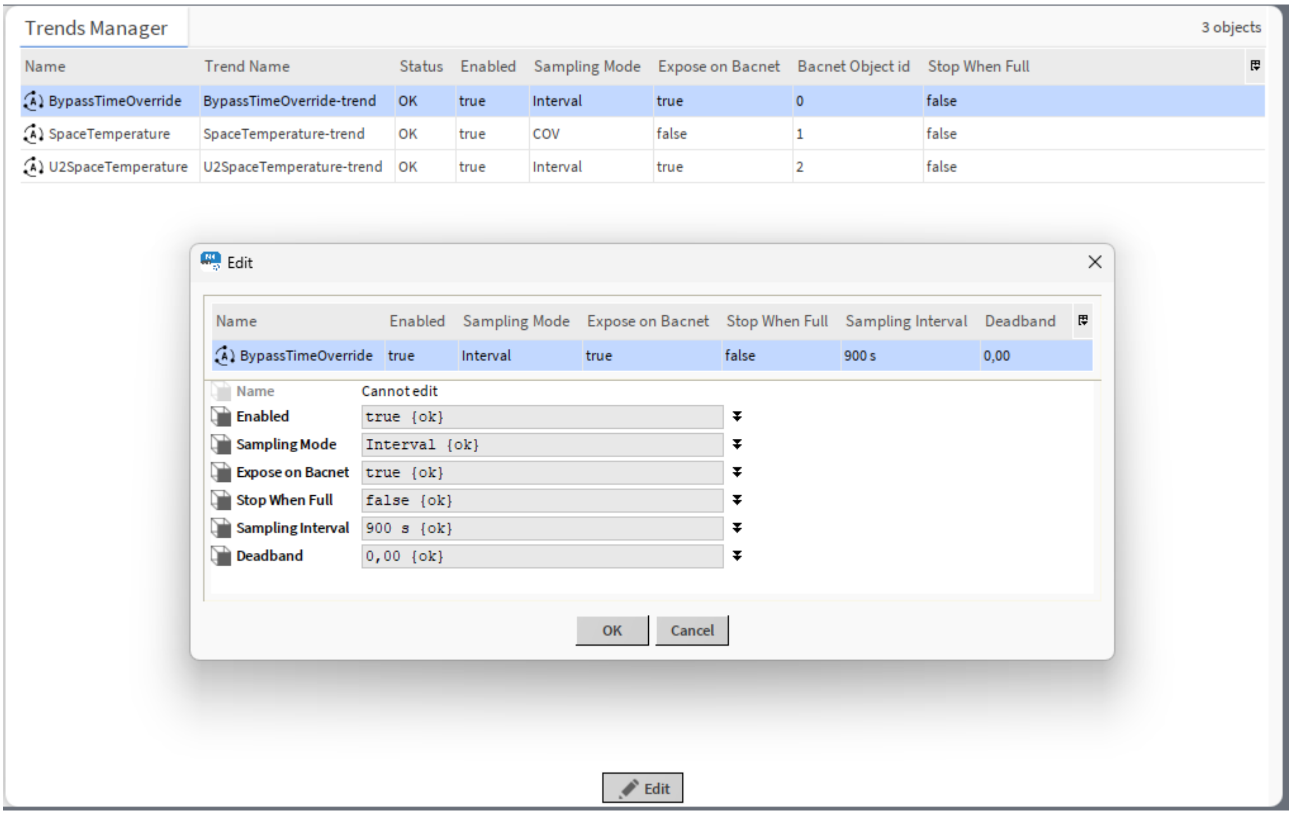
The Trends Manager is a simple view containing the list of all trend extensions added to Data Points.



Name	Trend Name	Status	Enabled	Sampling Mode	Expose on Bacnet	Bacnet Object id	Stop When Full
BypassTimeOverride	BypassTimeOverride-trend	OK	true	Interval	true	0	false
SpaceTemperature	SpaceTemperature-trend	OK	true	Interval	true	1	false
U2SpaceTemperature	U2SpaceTemperature-trend	OK	true	Interval	true	2	false

Figure 140. Trends Manager

The view allows to see all relevant information about the trend extensions, edit or clear trends data.



Trends Manager

3 objects

Name	Trend Name	Status	Enabled	Sampling Mode	Expose on Bacnet	Bacnet Object id	Stop When Full
BypassTimeOverride	BypassTimeOverride-trend	OK	true	Interval	true	0	false
SpaceTemperature	SpaceTemperature-trend	OK	true	COV	false	1	false
U2SpaceTemperature	U2SpaceTemperature-trend	OK	true	Interval	true	2	false

Edit

Name	Enabled	Sampling Mode	Expose on Bacnet	Stop When Full	Sampling Interval	Deadband
BypassTimeOverride	true	Interval	true	false	900 s	0,00

Name	Cannot edit
Enabled	true {ok}
Sampling Mode	Interval {ok}
Expose on Bacnet	true {ok}
Stop When Full	false {ok}
Sampling Interval	900 s {ok}
Deadband	0,00 {ok}

Figure 141. Editing options in the Trends Manager

The Trends Manager has the following columns:

- Name: shows the name of the Data Point that contains the trend extension;
- Trend Name: shows the name of the trend set in the Trend extension;
- Status: shows the status of the trend extension;

(the below data are editable in a pop-up window available upon selecting the trend(s) from the list and using the Edit button or from the trend's context menu):

- Enabled: shows if the trends extension is enabled in the Data Point;
- Sampling Mode: informs about the way the samples are gathered;
- Expose On BACnet: informs if the trend extension is automatically exposed on the BACnet network;
- BACnet Object ID: shows the assigned BACnet ID, if exposed;
- Stop When Full: informs if the stop-when-full setting is enabled in the extension.

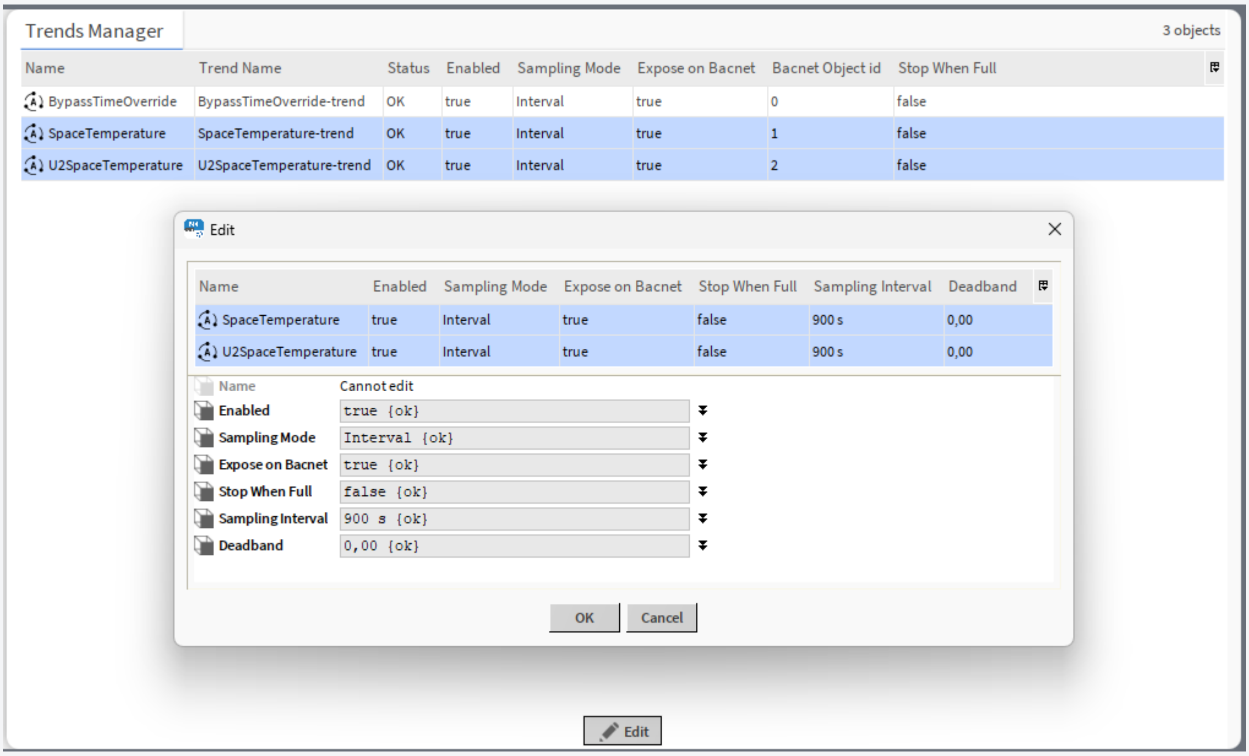
Additional parameters available to configure in the pop-up window:

- Sampling Interval: (applicable to interval or COV interval mode) time value, which triggers saving trend data;
- Deadband: (available only if COV or COV interval mode is selected, applicable only in the AnalogDataPoint) allows to set a value for a non-responsive zone; the trend data will only be saved if the COV exceeds the deadband value;

Please note that all the above data is available and (some) editable in the trends extension's slots in the relevant Data Point.

Multiediting

In the editing pop-up, it is possible to edit parameters for more than one trend at a time. To mark more than one trend for editing, use Shift or Ctrl keys. Then, press the Edit button.



Name	Trend Name	Status	Enabled	Sampling Mode	Expose on Bacnet	Bacnet Object id	Stop When Full
BypassTimeOverride	BypassTimeOverride-trend	OK	true	Interval	true	0	false
SpaceTemperature	SpaceTemperature-trend	OK	true	Interval	true	1	false
U2SpaceTemperature	U2SpaceTemperature-trend	OK	true	Interval	true	2	false

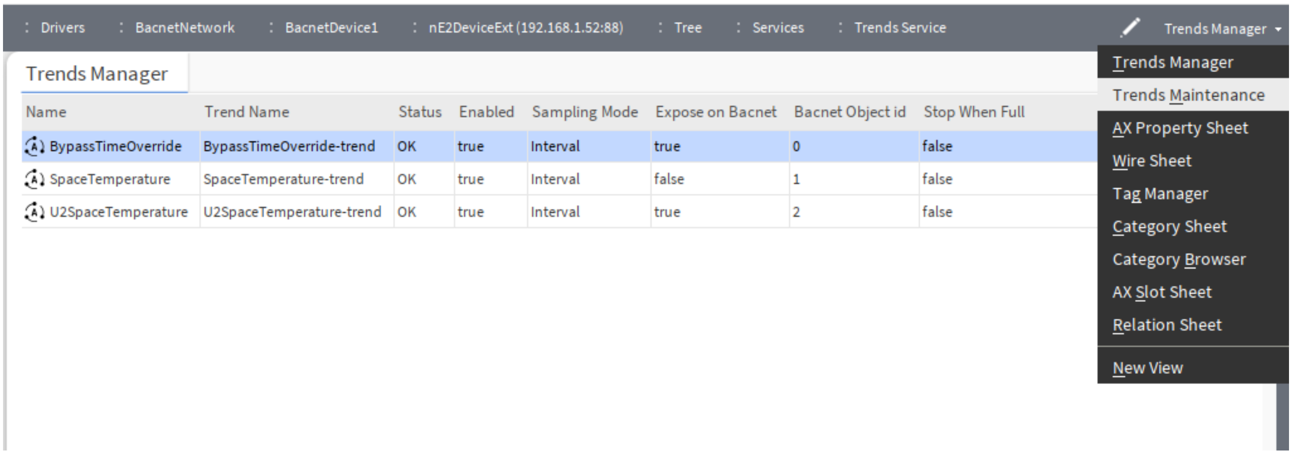
Name	Enabled	Sampling Mode	Expose on Bacnet	Stop When Full	Sampling Interval	Deadband
SpaceTemperature	true	Interval	true	false	900 s	0,00
U2SpaceTemperature	true	Interval	true	false	900 s	0,00

Name	Cannot edit
Enabled	true {ok}
Sampling Mode	Interval {ok}
Expose on Bacnet	true {ok}
Stop When Full	false {ok}
Sampling Interval	900 s {ok}
Deadband	0,00 {ok}

Figure 142. Multiediting

Trends Maintenance

In the top bar menu in the Trends Manager, another trends view is available, the Trends Maintenance.



Name	Trend Name	Status	Enabled	Sampling Mode	Expose on Bacnet	Bacnet Object id	Stop When Full
BypassTimeOverride	BypassTimeOverride-trend	OK	true	Interval	true	0	false
SpaceTemperature	SpaceTemperature-trend	OK	true	Interval	false	1	false
U2SpaceTemperature	U2SpaceTemperature-trend	OK	true	Interval	true	2	false

Figure 143. Accessing Trends Maintenance view

The Trends Maintenance view allows to clear data from the selected trend(s) or trends data up to a selected point in time.

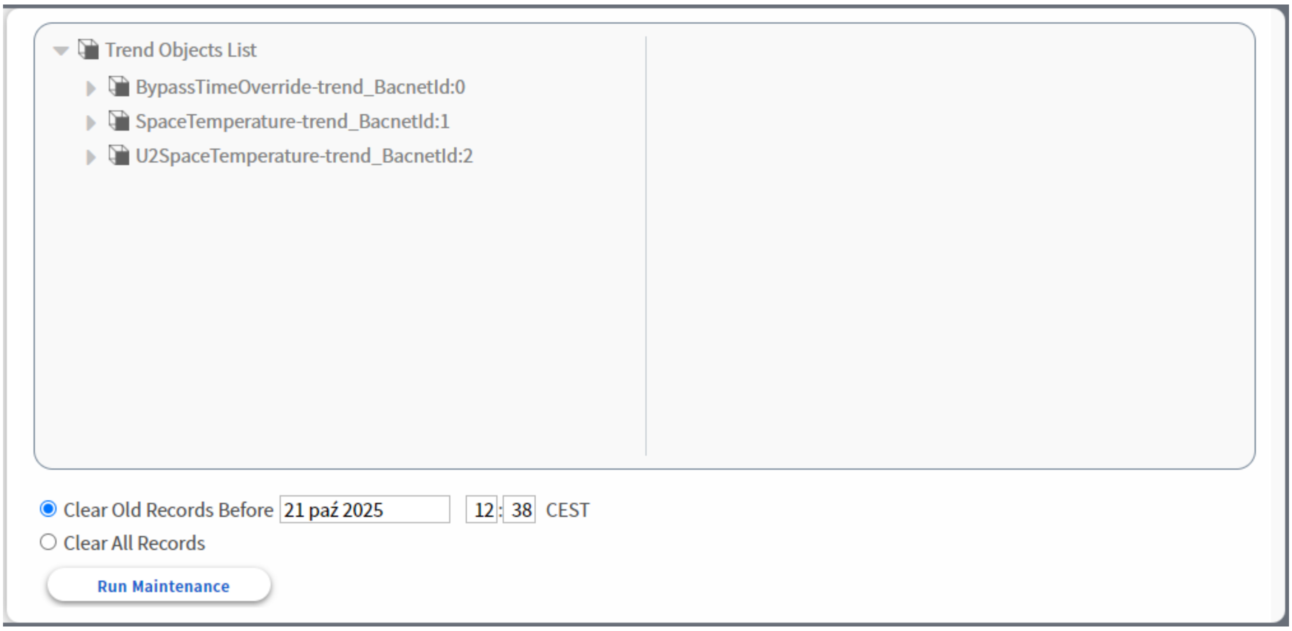

Trend Objects List
BypassTimeOverride-trend_BacnetId:0
SpaceTemperature-trend_BacnetId:1
U2SpaceTemperature-trend_BacnetId:2
● Clear Old Records Before 21 paź 2025 12 : 38 CEST
○ Clear All Records
Run Maintenance

Figure 144. Trends maintenance view

Note

In the Trends Maintenance view, each trend's name is followed by the BACnet ID assigned to the trend to help identify Data Points which are named the same but located in different applications:

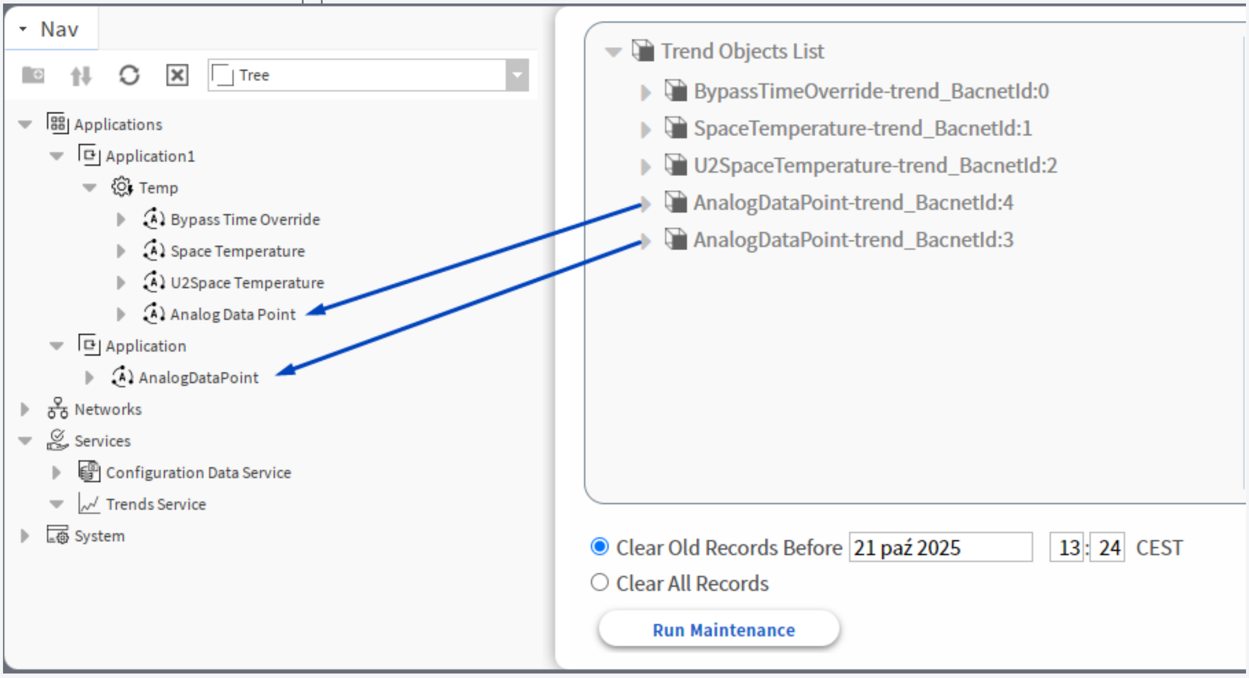

Nav
Tree
Applications
Application1
Temp
Bypass Time Override
Space Temperature
U2Space Temperature
Analog Data Point
Application
AnalogDataPoint
Networks
Services
Configuration Data Service
Trends Service
System
Trend Objects List
BypassTimeOverride-trend_BacnetId:0
SpaceTemperature-trend_BacnetId:1
U2SpaceTemperature-trend_BacnetId:2
AnalogDataPoint-trend_BacnetId:4
AnalogDataPoint-trend_BacnetId:3
● Clear Old Records Before 21 paź 2025 13 : 24 CEST
○ Clear All Records
Run Maintenance

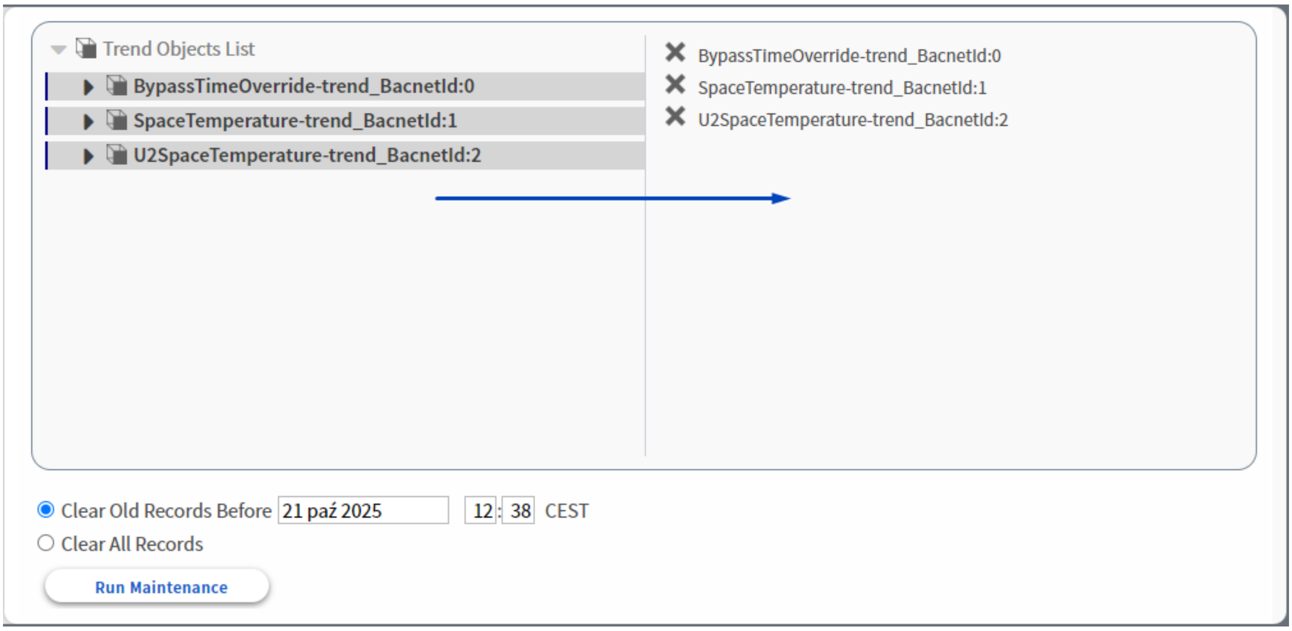
Figure 145. Identification of Data Points' trends from different applications

To clear trends data:

- First, select which trends to remove data from:
  - double-click a trend in the left-side window,
  - drag and drop a trend from the left-side window to the right-side window.

To add more than one trend to the right-side window, use Ctrl while selecting trends in the left-side window.

Selected trends will appear in the right-side window and further operations will **apply only** to these trends.


Trend Objects List
BypassTimeOverride-trend_BacnetId:0
SpaceTemperature-trend_BacnetId:1
U2SpaceTemperature-trend_BacnetId:2
BypassTimeOverride-trend_BacnetId:0
SpaceTemperature-trend_BacnetId:1
U2SpaceTemperature-trend_BacnetId:2
Run Maintenance

● Clear Old Records Before 21 paź 2025 12 : 38 CEST

○ Clear All Records

Figure 146. Selected trends

To remove a trend from the right-side window, use the cross (X) button:

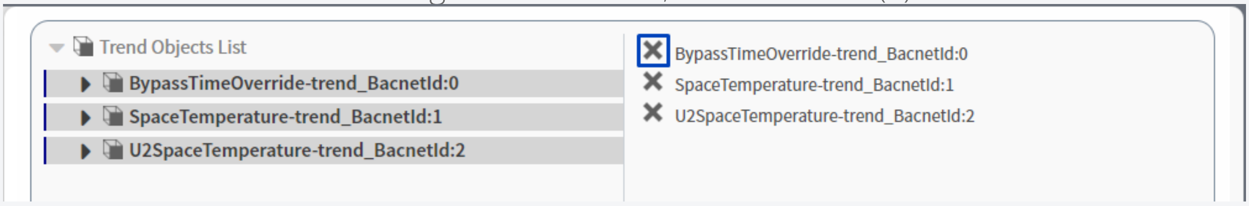
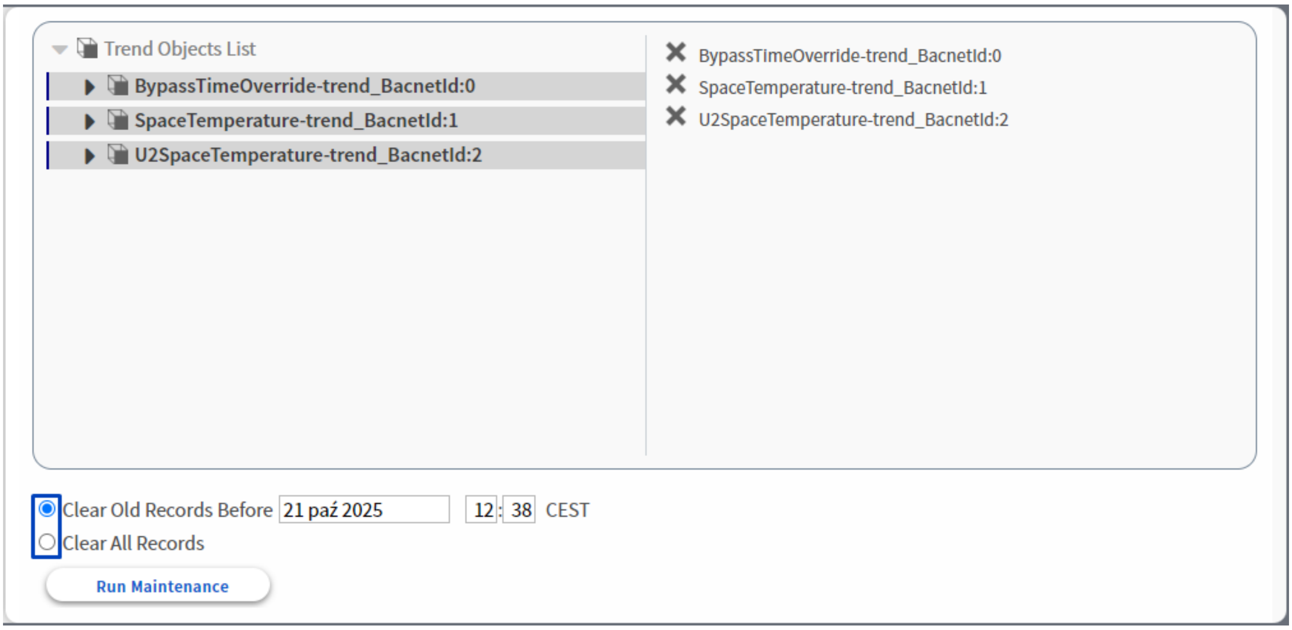

Trend Objects List
BypassTimeOverride-trend_BacnetId:0
SpaceTemperature-trend_BacnetId:1
U2SpaceTemperature-trend_BacnetId:2
BypassTimeOverride-trend_BacnetId:0
SpaceTemperature-trend_BacnetId:1
U2SpaceTemperature-trend_BacnetId:2

Figure 147. Removing a trend from the executable list

- Select whether:
  - all trends data should be erased (second bullet) or
  - just data until a given point in time (first bullet, required to fill in date and time before which all trends data will be removed).

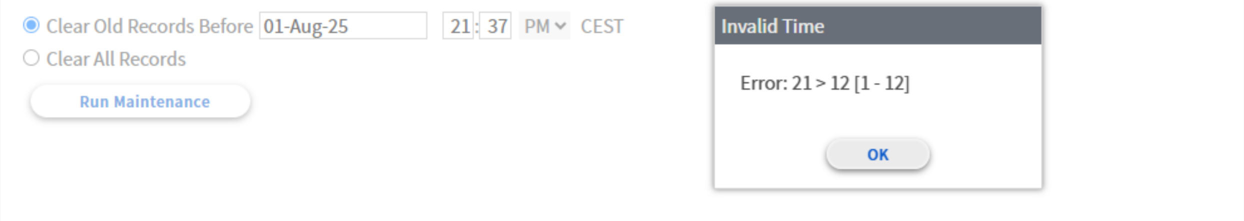

Trend Objects List
BypassTimeOverride-trend_BacnetId:0
SpaceTemperature-trend_BacnetId:1
U2SpaceTemperature-trend_BacnetId:2
BypassTimeOverride-trend_BacnetId:0
SpaceTemperature-trend_BacnetId:1
U2SpaceTemperature-trend_BacnetId:2
Run Maintenance

● Clear Old Records Before 21 paź 2025 12 : 38 CEST

○ Clear All Records

Figure 148. Selecting actions for trends maintenance

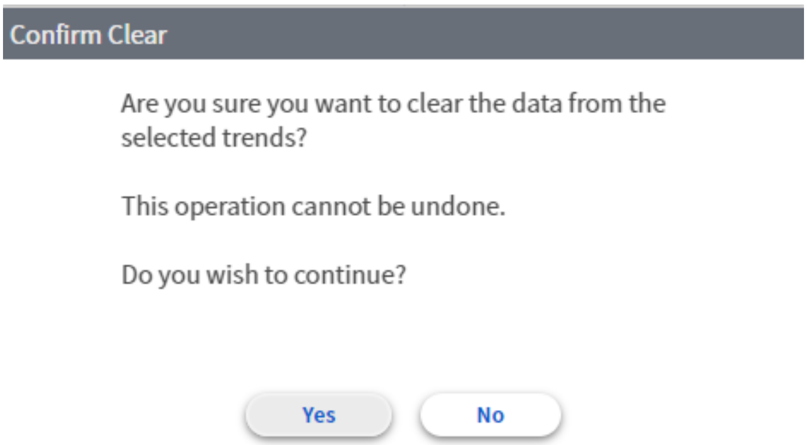
Note
The time format, when setting the time up to which the trends data will be deleted, is dependent on the computer time settings, it is either 12H AM/PM format or 24H format. If the time format is set to 12H and the time entered is in 24H time format, an error will be communicated:



Error: 21 > 12 [1 - 12]

Figure 149. Time format error

- Use the Run Maintenance button to erase data from selected trends. Removing trends data has to be confirmed in a pop-up window:



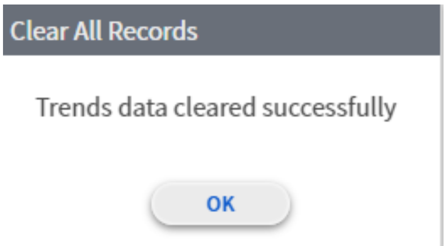
Are you sure you want to clear the data from the selected trends?

This operation cannot be undone.

Do you wish to continue?

Figure 150. Data removal confirmation pop-up

Successful removing of trends data is confirmed by a pop-up.



Trends data cleared successfully

Figure 151. Successful clearing of trends data

5.7.3 Tagging Service

The Tagging service provides centralized management of semantic tags for Data Points and equipment, ensuring consistent and standardized metadata assignment across the local control system. It supports both automatic and manual tagging mechanisms, allowing automatic tag generation to be enabled or disabled, as well as manual tag assignment based on available tag dictionaries.

Tags in BMS

Tags are labels or key-value pairs that provide metadata to identify, categorize, and organize data from devices and systems like HVAC, lighting, or security. They enable interoperability by providing a standardized way to understand data from different sources, which improves data quality, facilitates automation, simplifies integration, and allows for more efficient data analysis.

Installation

The Tagging service requires a Tagging library and is automatically available in the Services container once the Tagging library is installed on the device. The Tagging library is a default part of the nano EDGE ENGINE from the OS V1.9 and is not compatible with earlier versions.

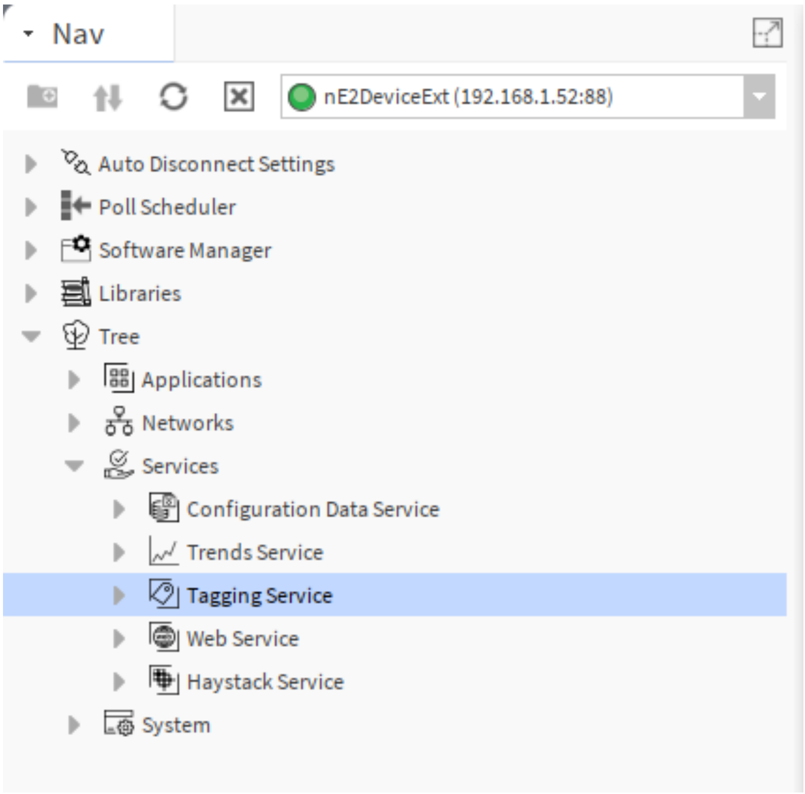
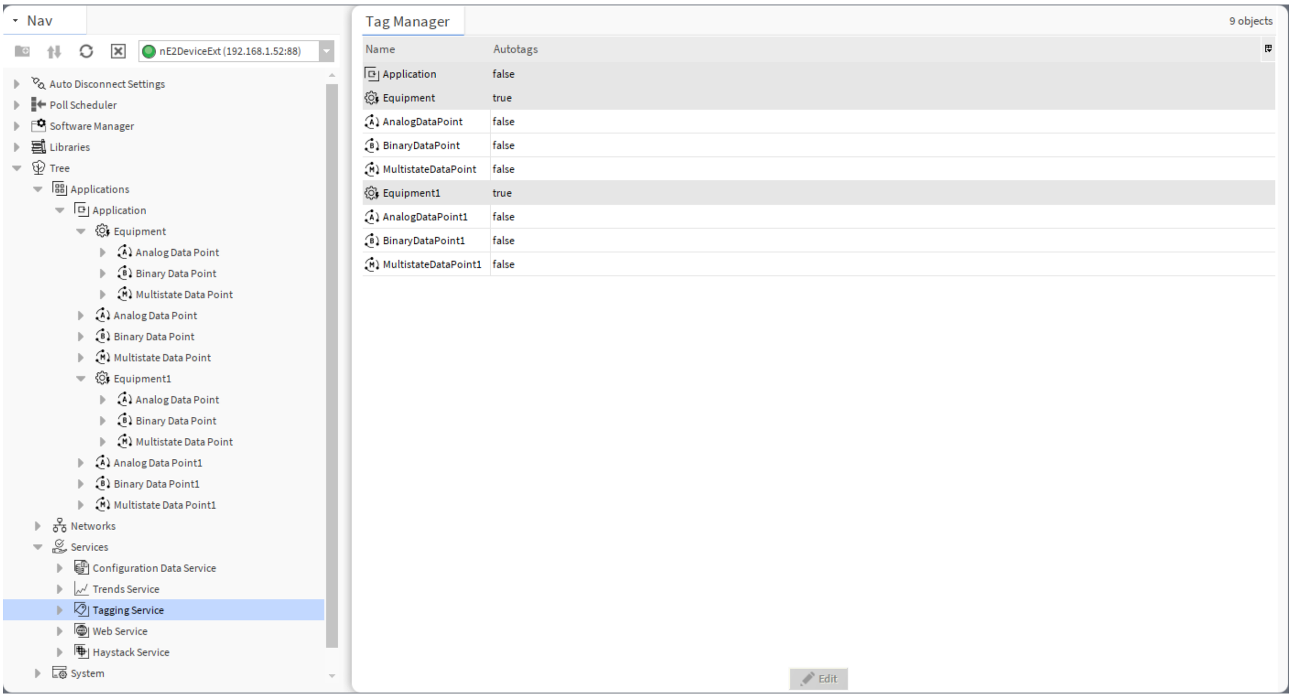


Figure 152. Tagging service in the nav tree

Default Views

Tag Manager

Tag Manager displays a list of available Applications and Equipment, along with the auto-tags status. It allows users to easily navigate between Data Points within Equipment and enables their editing and manual tag assignment using the Edit button.



Name	Autotags
Application	false
Equipment	true
AnalogDataPoint	false
BinaryDataPoint	false
MultistateDataPoint	false
Equipment1	true
AnalogDataPoint1	false
BinaryDataPoint1	false
MultistateDataPoint1	false

Figure 153. Tag Manager

The Tag Manager has the following columns:

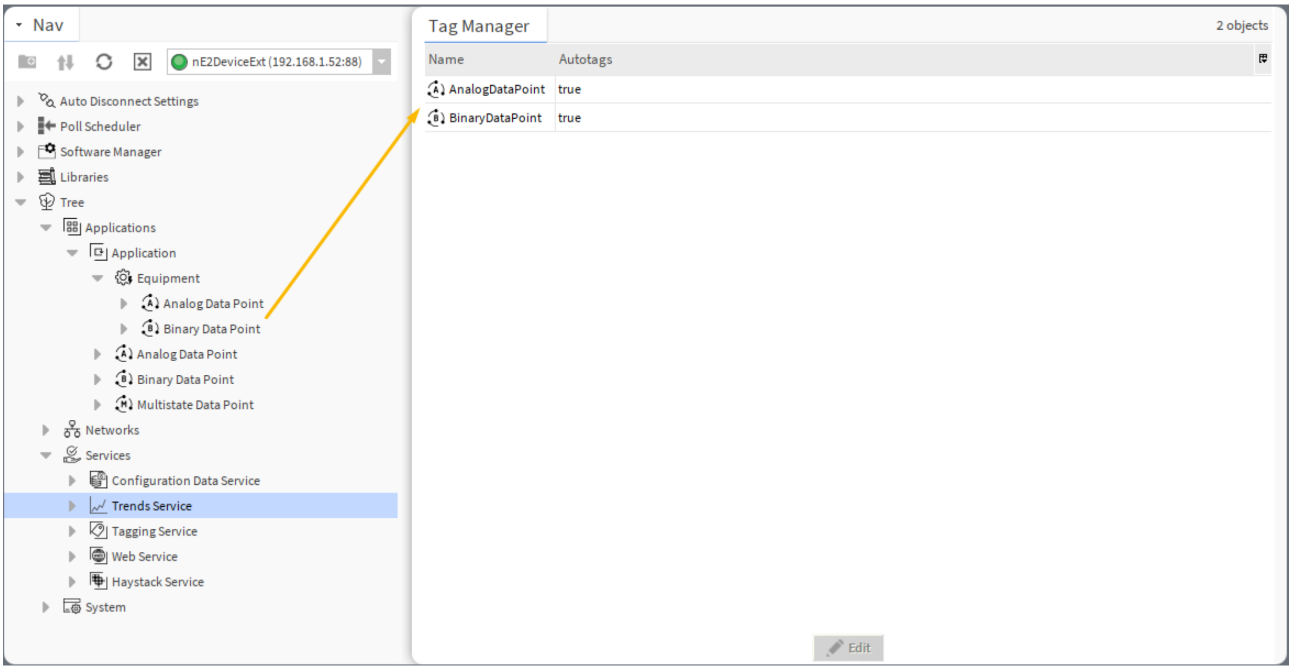
- Name: shows the name of the Equipment component or Data Point added to the device; the Equipment/Data Point hierarchy is maintained in this view;
- Autotags: informs if the auto-tag function is enabled or disabled for a given component.

The Tag Manager shows all Application and Equipment components along with Data Points.

Note

The Application components are displayed only for identification reasons, it is not possible to enable or disable auto-tagging or add tags manually to the Application component.

If Data Points are located under the Equipment component, it is required to double-click the relevant Equipment component to display its Data Points:

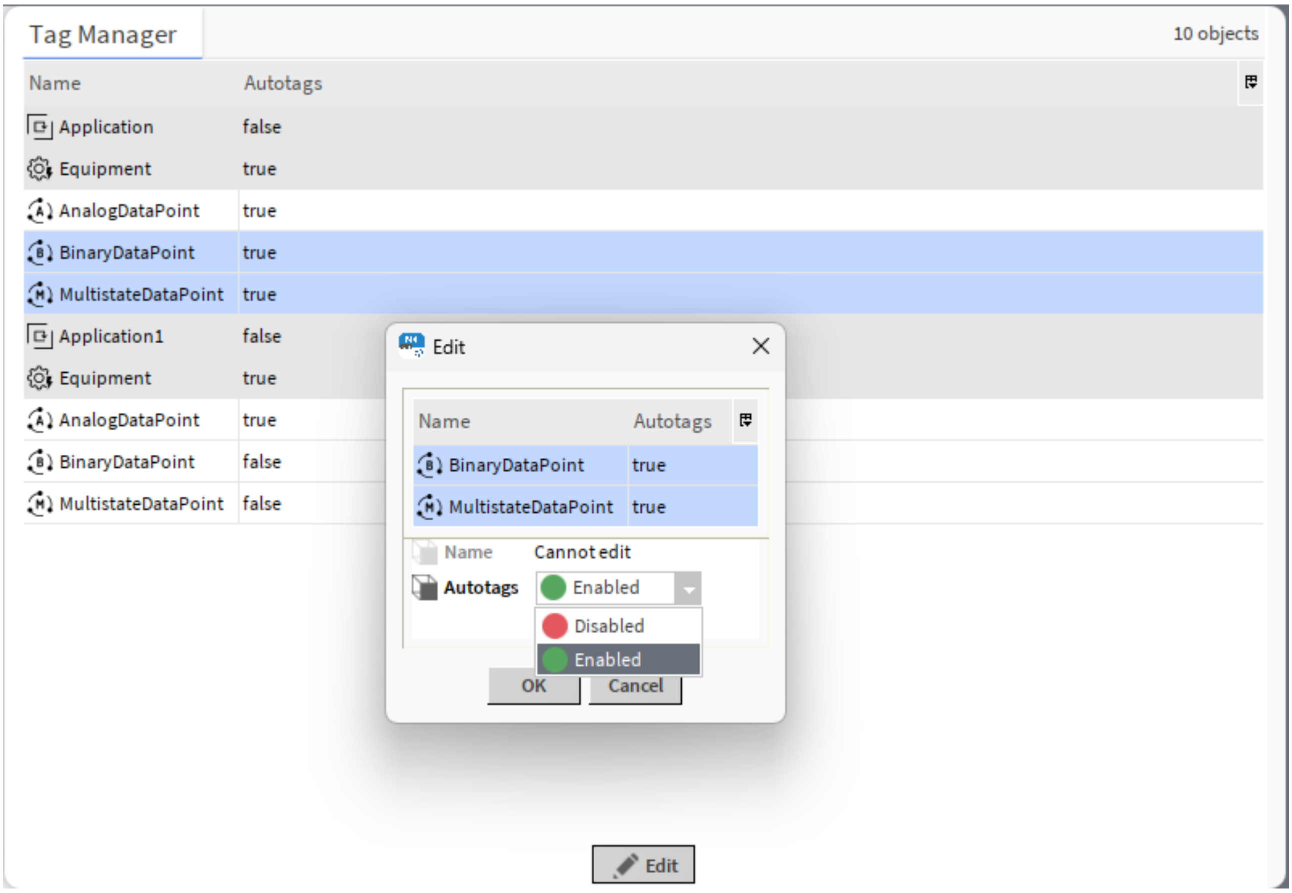


Tag Manager — 2 objects

Name	Autotags
AnalogDataPoint	true
BinaryDataPoint	true

Figure 154. Displaying Data Points under the Equipment component

The Tag Manager allows to edit auto-tagging for Data Points using the Edit button:



Tag Manager — 10 objects

Name	Autotags
Application	false
Equipment	true
AnalogDataPoint	true
BinaryDataPoint	true
MultistateDataPoint	true
Application1	false
Equipment	true
AnalogDataPoint	true
BinaryDataPoint	false
MultistateDataPoint	false

Edit

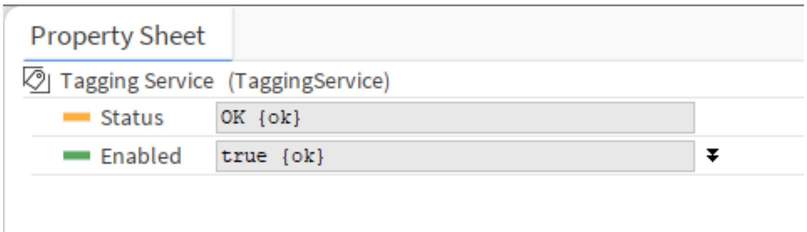
Name	Autotags
BinaryDataPoint	true
MultistateDataPoint	true

Name: Cannot edit
Autotags: Enabled

Figure 155. Editing auto-tags

It is possible to edit auto-tagging function for individual Data Point or for many Data Points at once (sets the same selected value to all marked Data Points).

AX Property Sheet



Property Sheet

Tagging Service (TaggingService)

Status	OK {ok}
Enabled	true {ok}

Figure 156. Tagging service slots

The Tagging service has the following slots, available through the AX Property Sheet view:

- Status: indicates the current status of the component. If the component works properly, its status is OK; however, it changes accordingly when values in other slots are adjusted.
    - Available information: disabled (the Enabled slot is set to false), OK;
- Enabled: change of the slot's value enables or disables the component.

Tag Dictionaries

Tag dictionaries are libraries containing lists of tags, which can be applied to Data Points and the Equipment component. Tag dictionaries are searchable and contain labels, which allow for the clear and efficient structure management, data modelling, and faster data processing by local control systems.

There are two tag dictionaries available: Haystack and Webserver.

Haystack

The list of tags in the Haystack tag dictionary corresponds with the Haystack standard: Tags in Project Haystack; however, there are some differences. The Haystack tag dictionary does not include the following tags:

Haystack Tags Exceptions

`list`, `dict`, `grid`, `scalar`, `xstr`, `choice`, `symbol`, `is`, `na`, `remove`, `of`, `quantities`, `quantityOf`, `tagOn`, `tags`, `containedBy`, `contains`, `def`, `defx`, `inputs`, `outputs`, `reciprocalOf`, `relationship`, `span`, `mlIdentificationPeriod` as well as tags of types: feature, filetype, lib, op, list, dict, grid.
These tags are not relevant for the data modelling based on Data Points and the Equipment component (in the application structure).

Webserver

The tags of the Webserver tag dictionary allow for a seamless display of tagged elements in the nanoWebUI™. The list of Webserver dictionary includes:

Tag	Auto-tag	Description
`web:application`	Yes (ADP, BDP, MDP)	Shows which application the Data Point belongs to (for the nanoWebUI order)
`web:expose`	Yes (ADP, BDP, MDP)	Defines the Data Point’s visibility on the nanoWebUI (editable)
`web:widgetType`	Yes (ADP, BDP, MDP)	Defines the widget type to display on the nanoWebUI (editable)

Tag	Auto-tag	Description
`web:order`	Yes (ADP, BDP, MDP)	Defines the order of widgets displayed on the nanoWebUI
`web:decimalPoint`	Yes (ADP)	Allows to configure the display of decimal point on the nanoWebUI (editable)
`web:step`	Yes (ADP)	Allows to configure the step value change for the nanoWebUI (editable)
`web:homepage`	No (manually added)	Sets the nanoWebUI as a web browser's homepage

Note
To learn more about the display of the tagged elements in the nanoWebUI™, please see the nanoWebUI™ description.

Applying Tags

Applying tags in the nano EDGE ENGINE is based on a semantic approach that ensures consistent data structure which is easily usable by the nanoWebUI™ and by third-party systems. Tags are applied at the Equipment and Data Point levels, where Equipment serves as the logical container defining what is being controlled, and Data Points represent the measured or commanded values associated with that equipment. This structured model ensures that tagged data is immediately usable by platforms capable of communicating through standardized tag-based HTTP APIs, e.g., Haystack.

Note: Tags can be only applied to the Equipment components and Data Points. Other component types are not supported.

It is therefore recommended (however, not mandatory) to use the following structure when creating applications:

- Applications container
    - Application component
        - Equipment component
            - Data Point(s)
            - other components
        - Equipment component
            - Data Point(s)
            - other components

A required condition for using tags is that the Tagging library is installed on the device and the Tagging service is available in the Services container. Both are a default part of the nano EDGE ENGINE V1.9 and are not compatible with previous versions.

Tags are available in two formats: auto-tagging and manually added.

Auto-tagging

The auto-tagging purpose is to automatically assign appropriate tags and markers based on various conditions. This mechanism ensures consistency and reduces manual work when creating or modifying applications.

If enabled, the auto-tagging mechanism is executed in the following situations:

- after all components are loaded during a device’s start-up,
- when a component is renamed,
- when a component is redefined (e.g., extensions/tags are added/removed),
- when a new component is added,
- when a component is moved within the Workspace tree.

The list of auto-tags includes:

Tag	Data Points applicability	Description
`hs:id`	ADP, BDP, MDP	Unique identifier of the Data Point (read-only)
`hs:cur`	ADP, BDP, MDP	Marker indicating a current value support (read-only)
`hs:curStatus`	ADP, BDP, MDP	Status of the Data Point's current value (read-only)
`hs:dis`	ADP, BDP, MDP	Display name of the component (read-only)
`hs:point`	ADP, BDP, MDP	Marker (read-only)
`hs:curVal`	ADP, BDP, MDP	Current output value (read-only)
`hs:kind`	ADP, BDP, MDP	Value kind (read-only)
`hs:equipRef`	ADP, BDP, MDP	Reference to the parent Equipment component (only if located under it) (read-only)
`web:application`	ADP, BDP, MDP	Reference to the application (for the nanoWebUI order)
`web:expose`	ADP, BDP, MDP	User-editable flag to show on the nanoWebUI (editable)
`web:widgetType`	ADP, BDP, MDP	Widget type for the nanoWebUI (editable)
`hs:writable`	ADP, BDP, MDP	Added if the Data Point’s mode is set *Output* or *Value*; removed when changed to *Input* (read-only)
`web:order`	ADP, BDP, MDP	Order for widgets on the nanoWebUI
`hs:enum`	BDP, MDP	BDP: Derived from `falseText` and `trueText` (read-only) / MDP: Derived from multistate labels (read-only)
`hs:minVal`, `hs:maxVal`	ADP	Inclusive minimum and maximum allowable value for a numeric input in the nanoWebUI-displayed component (editable)
`web:decimalPoint`, `web:step`	ADP	nanoWebUI presentation configuration (editable)
`hs:unit`	ADP	Added when the user selects a unit (read-only)

To enable or disable auto-tagging for the Data Point, either open the Edit nE2 Tags option from the Data Point’s context menu or go to the Tag Manager and use the Edit button:

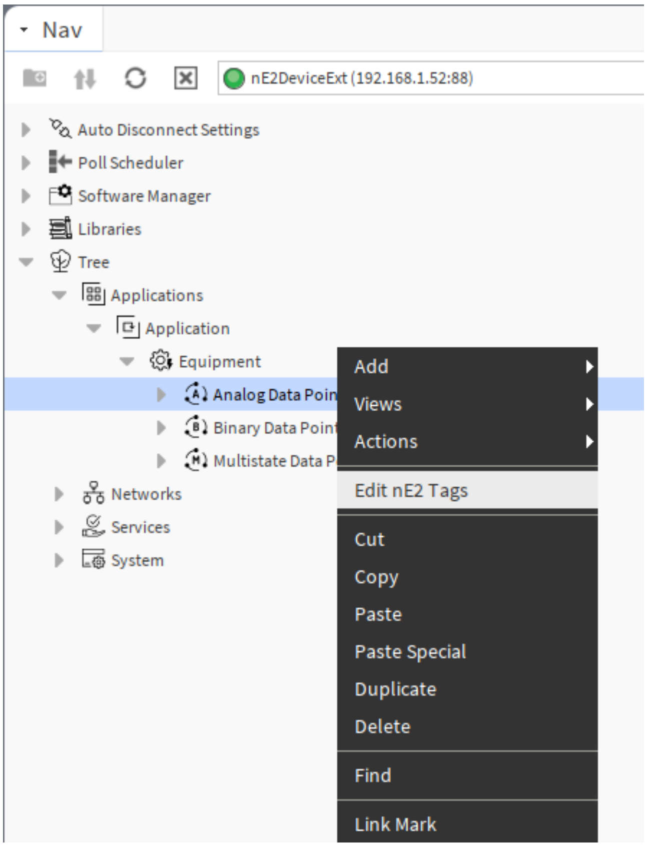
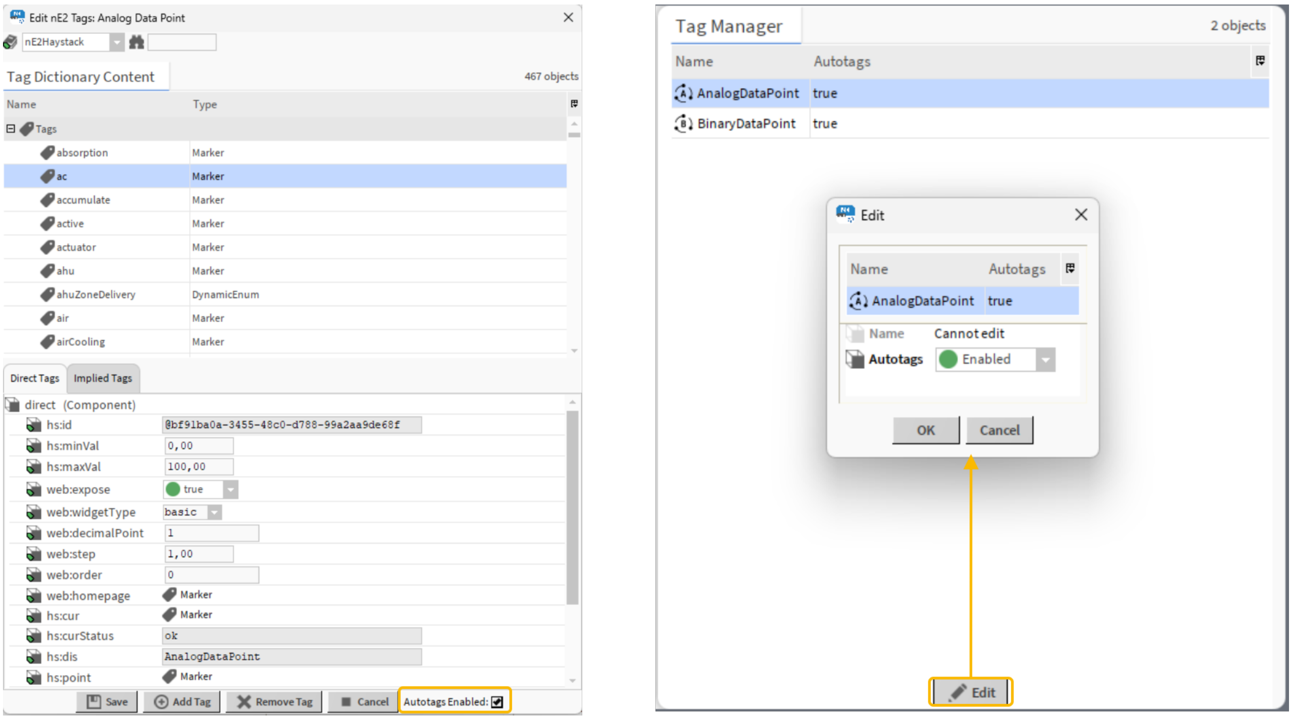


Figure 157. Edit nE2 Tags context menu option



Name	Type
Tags	
absorption	Marker
ac	Marker
accumulate	Marker
active	Marker
actuator	Marker
ahu	Marker
ahuZoneDelivery	DynamicEnum
air	Marker
airCooling	Marker

direct (Component)

hs:id	@bf91ba0a-3455-48c0-d788-99a2aa9de68f
hs:minVal	0,00
hs:maxVal	100,00
web:expose	true
web:widgetType	basic
web:decimalPoint	1
web:step	1,00
web:order	0
web:homepage	Marker
hs:cur	Marker
hs:curStatus	ok
hs:dis	AnalogDataPoint
hs:point	Marker

Autotags Enabled: ☑

Name	Autotags
AnalogDataPoint	true
BinaryDataPoint	true

Name	Autotags
AnalogDataPoint	true

Name	Cannot edit
Autotags	Enabled

Figure 158. Editing auto-tags

In the Edit nE2 Tags dialog window, check (enable) or uncheck (disable) the Autotags Enabled option. Make sure to save changes.

In the Tag Manager Edit dialog window, select Enabled or Disabled option from a drop-down list.

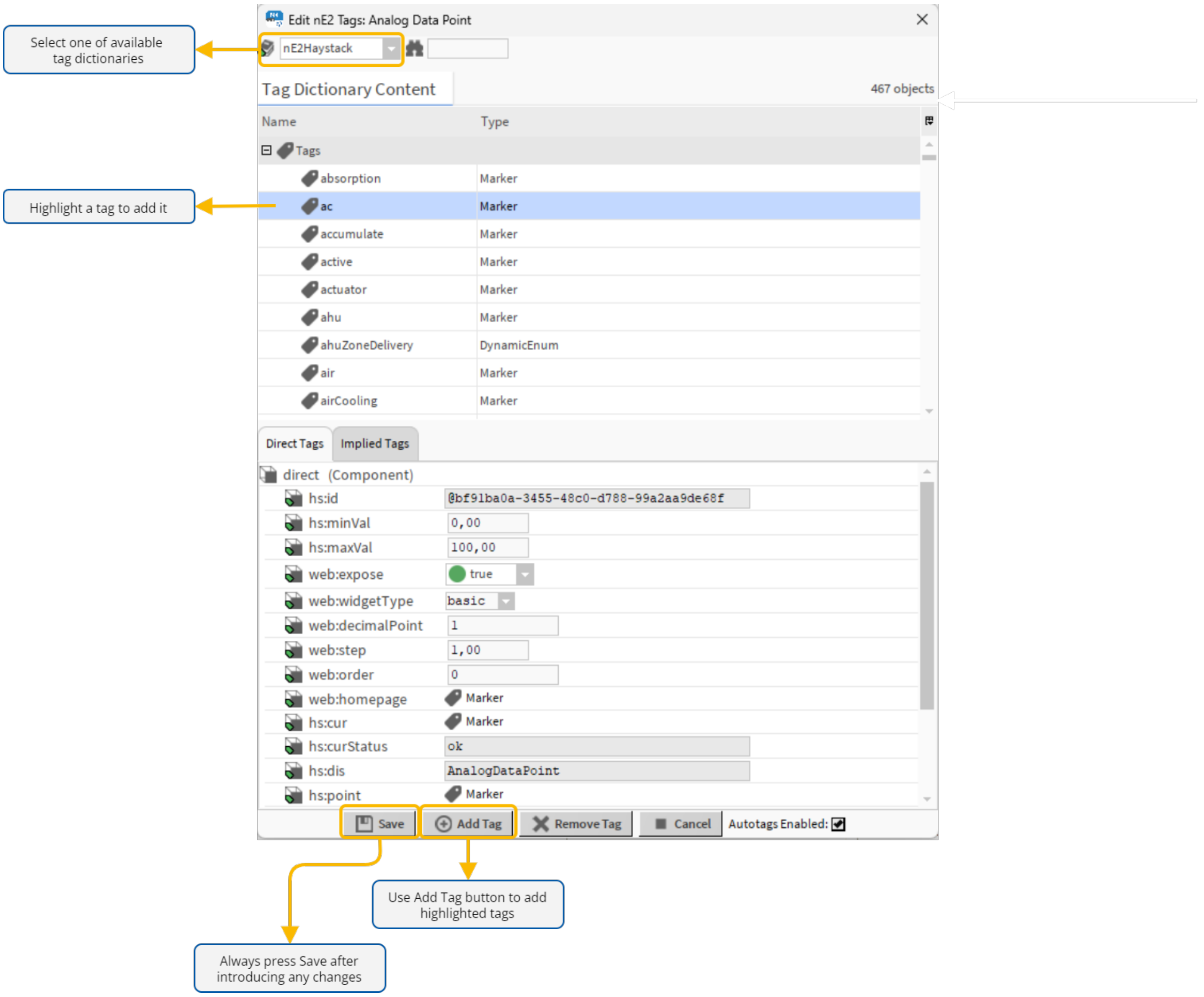
Auto-tagging in the Equipment Component

Due to implemented data modelling, auto-tags in the Equipment component are enabled by default and cannot by disabled.

Manual Adding of Tags

Outside of the auto-tagging mechanism, it is also possible to manually add tags.

Adding and editing tags are possible in the Edit nE2 Tags dialog window opened from the context menu of the Equipment component or Data Point.



Edit nE2 Tags: Analog Data Point

nE2Haystack

Tag Dictionary Content — 467 objects

Name	Type
Tags	
absorption	Marker
ac	Marker
accumulate	Marker
active	Marker
actuator	Marker
ahu	Marker
ahuZoneDelivery	DynamicEnum
air	Marker
airCooling	Marker

Direct Tags | Implied Tags

direct (Component)

hs:id	@bf91ba0a-3455-48c0-d788-99a2aa9de68f
hs:minVal	0,00
hs:maxVal	100,00
web:expose	true
web:widgetType	basic
web:decimalPoint	1
web:step	1,00
web:order	0
web:homepage	Marker
hs:cur	Marker
hs:curStatus	ok
hs:dis	AnalogDataPoint
hs:point	Marker

Autotags Enabled: ☑

Select one of available tag dictionaries

Highlight a tag to add it

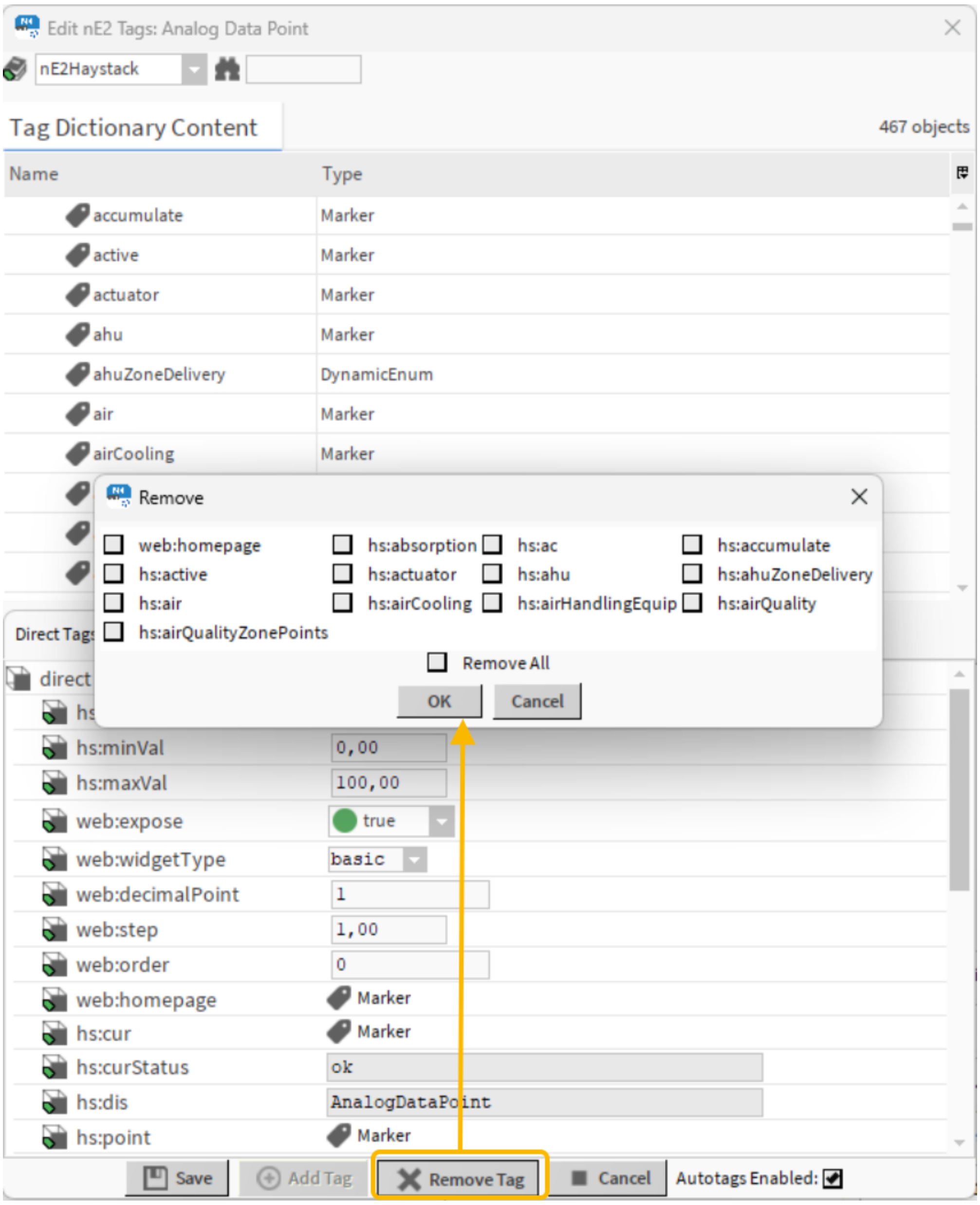
Use Add Tag button to add highlighted tags

Always press Save after introducing any changes

Figure 159. Edit nE2 Tags dialog window - options for manual adding tags

In the Edit nE2 Tags dialog window, there are the following options:

- **selecting tag dictionary:** choose from the list of available tag dictionaries to display it in the Tag Dictionary Content section; it is possible to search through the list of tags in a search field;
- **adding tags:** to add a new tag, highlight it in the Tag Dictionary Content section and confirm with the Add Tag button; it is possible to mark multiple tags by using Shift/Ctrl buttons on the keyboard;
- **removing tags:** to remove a tag, use the Remove Tag button, which opens another dialog window; check tags to remove (or Remove All option) and confirm with OK:



Name	Type
accumulate	Marker
active	Marker
actuator	Marker
ahu	Marker
ahuZoneDelivery	DynamicEnum
air	Marker
airCooling	Marker

☐ web:homepage ☐ hs:absorption ☐ hs:ac ☐ hs:accumulate
☐ hs:active ☐ hs:actuator ☐ hs:ahu ☐ hs:ahuZoneDelivery
☐ hs:air ☐ hs:airCooling ☐ hs:airHandlingEquip ☐ hs:airQuality
☐ hs:airQualityZonePoints

☐ Remove All

hs:minVal	0,00
hs:maxVal	100,00
web:expose	● true
web:widgetType	basic
web:decimalPoint	1
web:step	1,00
web:order	0
web:homepage	Marker
hs:cur	Marker
hs:curStatus	ok
hs:dis	AnalogDataPoint
hs:point	Marker

Autotags Enabled: ☑

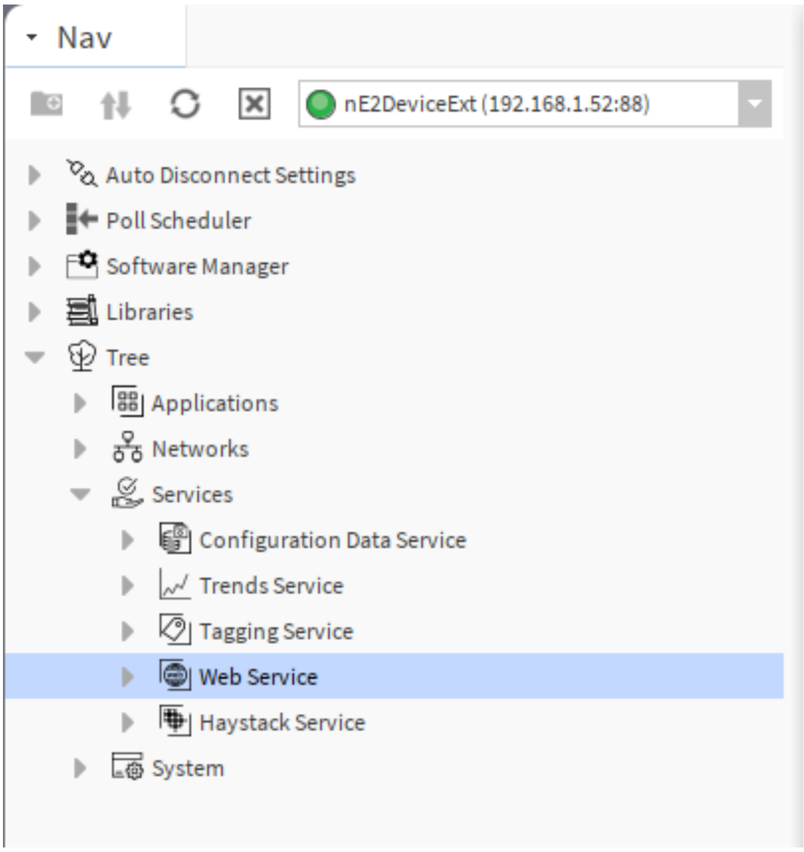
Figure 160. Remove tags dialog window

Save
Always remember to save changes made in the Edit nE2 Tags dialog window.

Direct and Implied Tags
In the bottom section of the Edit nE2 Tags dialog window, there are two tabs, Direct Tags and Implied Tags.
The Direct Tags tab includes a list of tags added from any of the tag dictionaries dedicated specifically for the nano EDGE ENGINE devices (delivered with the nE2 Link module). Depending on the tag parameters, tags in the Direct Tags tabs can be editable.
The Implied Tags tab includes a list of Niagara-derived tags, which are added automatically based on Niagara-defined rules for the given type of component. These tags cannot be edited, added, or removed, and are required for a proper integration with Niagara.

5.7.4 Web Service

The Web service enables the display of a web-based interface, nanoWebUI™, designed for the management of Data Points in applications. From the nano EDGE ENGINE V1.9 implementation of Haystack service and tagging, it allows to display tagged Data Points and Equipment components in the nanoWebUI™.

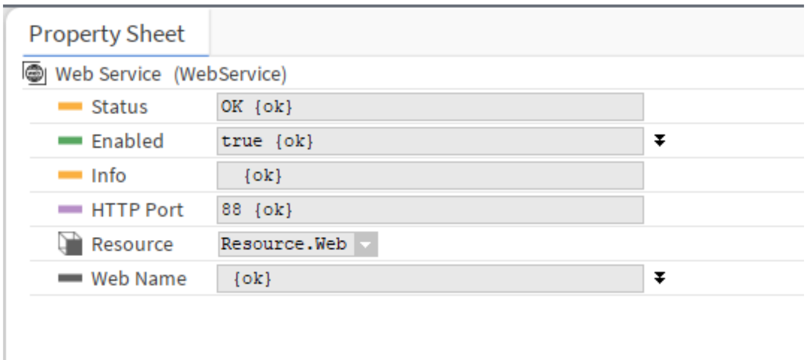


Nav

nE2DeviceExt (192.168.1.52:88)

- Auto Disconnect Settings
- Poll Scheduler
- Software Manager
- Libraries
- Tree
  - Applications
  - Networks
  - Services
    - Configuration Data Service
    - Trends Service
    - Tagging Service
    - Web Service
    - Haystack Service
  - System

Figure 161. Web service in the nav tree



Property Sheet

Web Service (WebService)

Status	OK {ok}
Enabled	true {ok}
Info	{ok}
HTTP Port	88 {ok}
Resource	Resource.Web
Web Name	{ok}

Figure 162. Web service slots

The Web service has the following slots:

- Status: indicates the current status of the component. If the component works properly, its status is OK; however, it changes accordingly when values in other slots are adjusted.
  - Available information: disabled (the Enabled slot is set to false), OK;
- Enabled: change of the slot's value enables or disables the component;
- Info: informs about a non-OK status of the component;
  - Available information:

  - Couldn’t start file server (status Error): occurs when the file server cannot not be started due to a memory shortage or other exception,
  - No resource manager (status Error): occurs when the resource file manager cannot be opened due to a memory shortage or other exception,
  - No resource (status Error): occurs when the resource file is not loaded,
  - The port change will take effect after the device restart (status OK): occurs when the HTTP port number has been changed with the Set HTTP port action;

Numerical value	Displayed information
0	No information displayed in the Info slot
1	Couldn’t start file server
2	No resource manager
3	No resource
4	The port change will take effect after the device restart

- HTTP Port: shows the set communication port number;
- Resource: allows to select a resource pack for displaying the web interface;

Note
The default resource pack for the display of the nanoWebUI™ interface is the Resource.Web file, delivered with the nano EDGE ENGINE OS V1.9.

- Web Name: allows to set an individual name for the web server, displayed in the left panel.

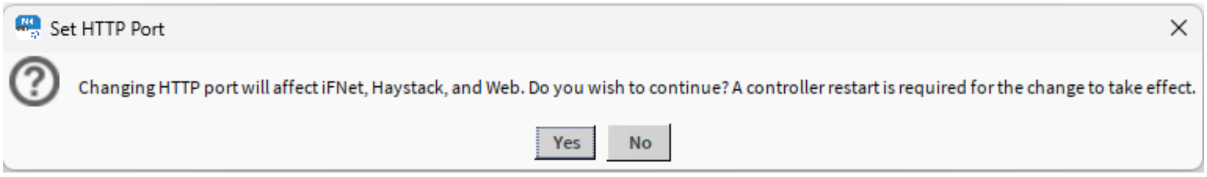
The Web service has the following action:

- Set HTTP Port: allows to change the HTTP communication port.

Note
The Set HTTP Port action is available also in the iFnet component and Haystack service.

Warning!
Changing the HTTP port affects the iFnet communication port number and will disable connecting with the controller using the current iFnet port number.

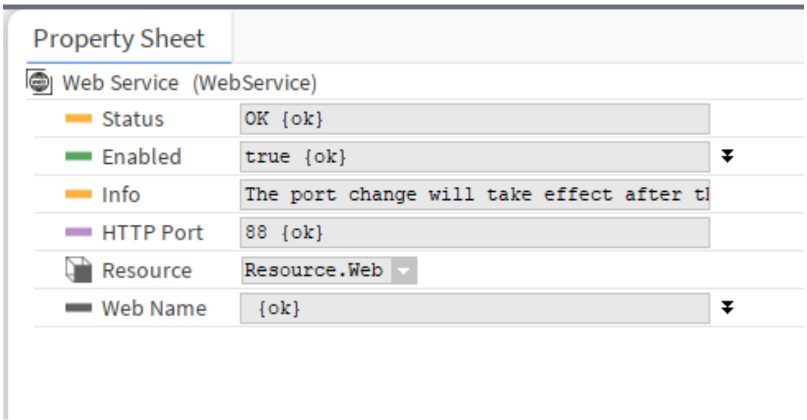
The HTTP port number can be changed by the user using the Set HTTP Port action. Changing the HTTP port number affects the communication port used to connect to the device, Haystack service, and connection to the nanoWebUI™ interface. The following notification is displayed before changing the port number:



Changing HTTP port will affect iFNet, Haystack, and Web. Do you wish to continue? A controller restart is required for the change to take effect.

Figure 163. HTTP port change notification

Once the change is introduced, it requires saving and restarting the device. Before saving, the component displays a notice in the Info slot:



Property Sheet

Web Service (WebService)

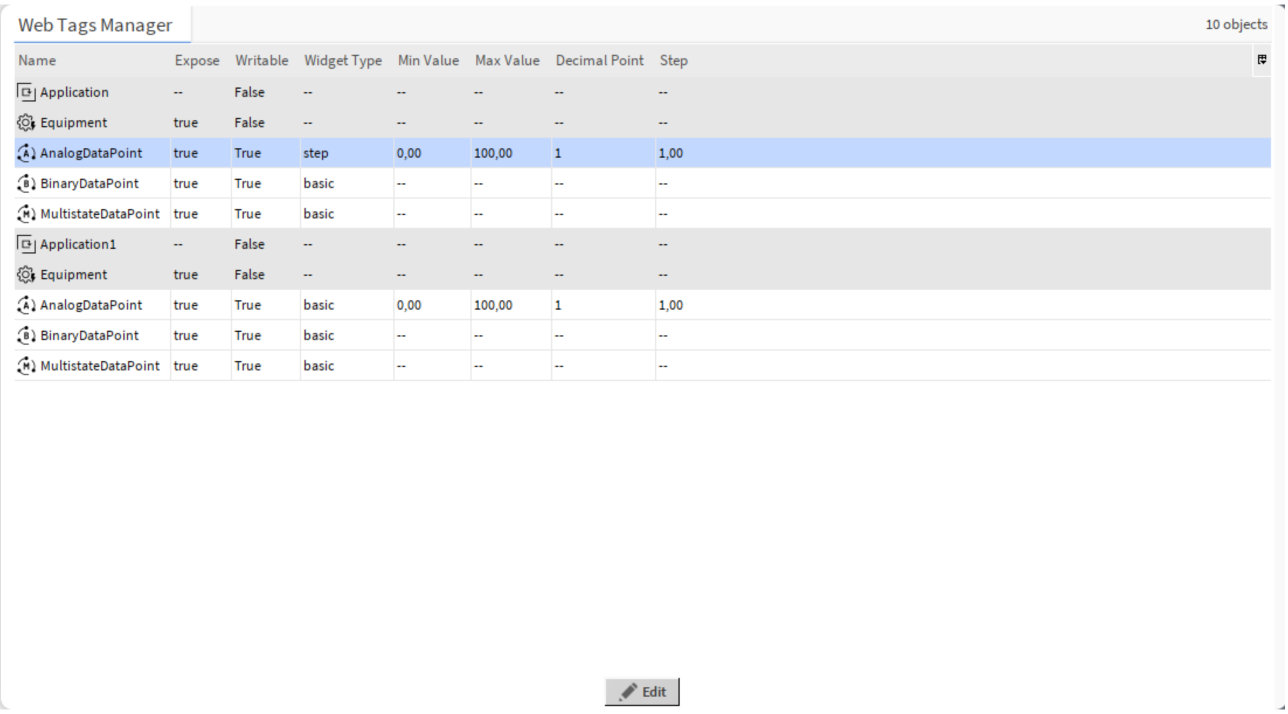
Status	OK {ok}
Enabled	true {ok}
Info	The port change will take effect after tl
HTTP Port	88 {ok}
Resource	Resource.Web
Web Name	{ok}

Figure 164. Info slot notification after HTTP port change

Web Tags Manager

The Web Tags Manager is a simple view containing the list of all Equipment components and Data Points presenting values of tags related to the nanoWebUI™ display of the point.

> Note
> The Application components is displayed in the Web Tags Manager view only for identification reasons as the component cannot be subject to auto-tagging or adding tags at all.



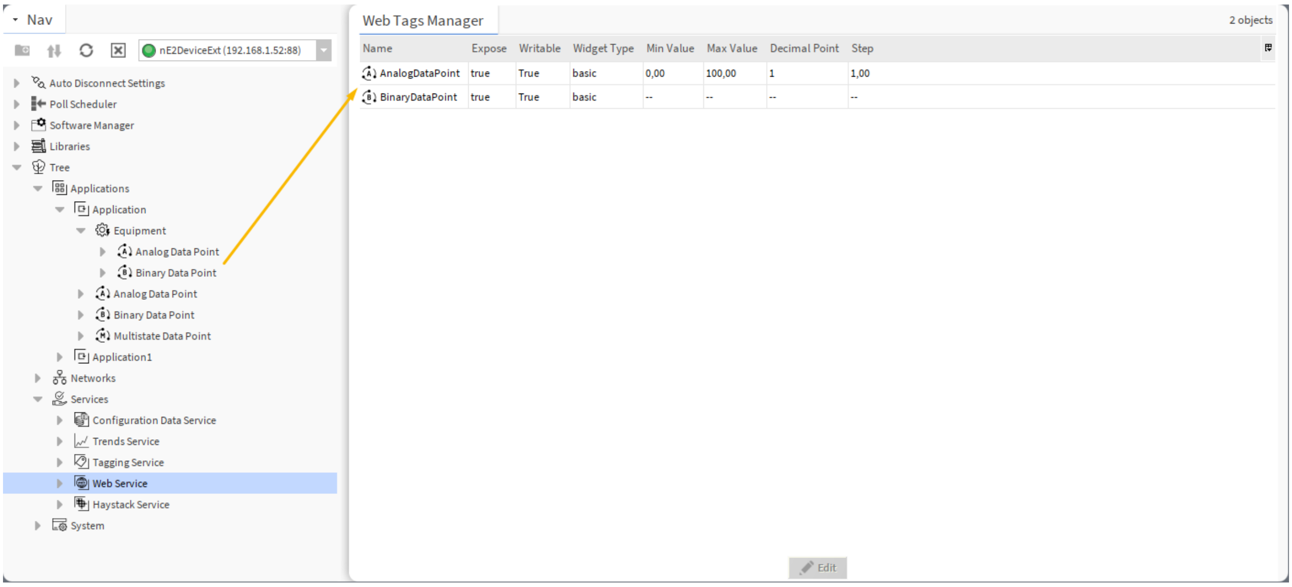
Web Tags Manager

10 objects

Name	Expose	Writable	Widget Type	Min Value	Max Value	Decimal Point	Step
Application	--	False	--	--	--	--	--
Equipment	true	False	--	--	--	--	--
AnalogDataPoint	true	True	step	0,00	100,00	1	1,00
BinaryDataPoint	true	True	basic	--	--	--	--
MultistateDataPoint	true	True	basic	--	--	--	--
Application1	--	False	--	--	--	--	--
Equipment	true	False	--	--	--	--	--
AnalogDataPoint	true	True	basic	0,00	100,00	1	1,00
BinaryDataPoint	true	True	basic	--	--	--	--
MultistateDataPoint	true	True	basic	--	--	--	--

Figure 165. Web Tags Manager view

If Data Points are located under the Equipment component, it is required to double-click the relevant Equipment component to display its Data Points:



Web Tags Manager

2 objects

Name	Expose	Writable	Widget Type	Min Value	Max Value	Decimal Point	Step
AnalogDataPoint	true	True	basic	0,00	100,00	1	1,00
BinaryDataPoint	true	True	basic	--	--	--	--

Figure 166. Displaying Data Points under the Equipment component

The Web Tags Manager view has the following columns:

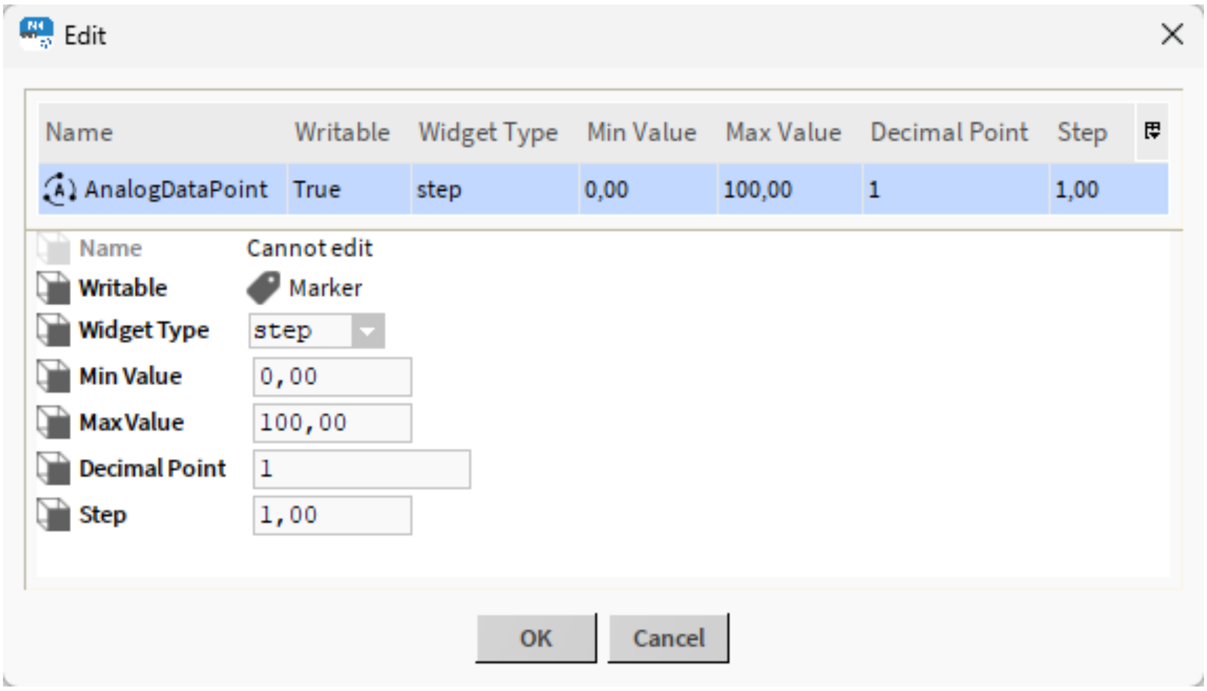
- Name: shows the name of the Application, Equipment component or Data Point added to the device; the Application/Equipment/Data Point hierarchy is maintained in this view;
- Expose: informs if the `web:expose` tag is active for the given Equipment component or Data Point;
- Writable: informs if the `hs:writable` tag is active for the given Data Point;
- Widget Type: informs which widget type is set in the `web:widgetType` tag for the given Data Point;
- Min Value: informs what minimum value is set in the `hs:minVal` tag for the given Data Point;
- Max Value: informs what maximum value is set in the `hs:maxVal` tag for the given Data Point;
- Decimal Point: informs which decimal point is set in the `web:decimalPoint` tag for the given Data Point;
- Step: informs what step value is set in the `web:step` tag for the given Data Point.

Actions

Editing

Editing of web tags parameters is available in the Edit dialog window under the Edit button:

- Edit: allows to edit fields listed in the Web Tags Manager (widget type, min. value, max. value, decimal point, and step—if applicable).



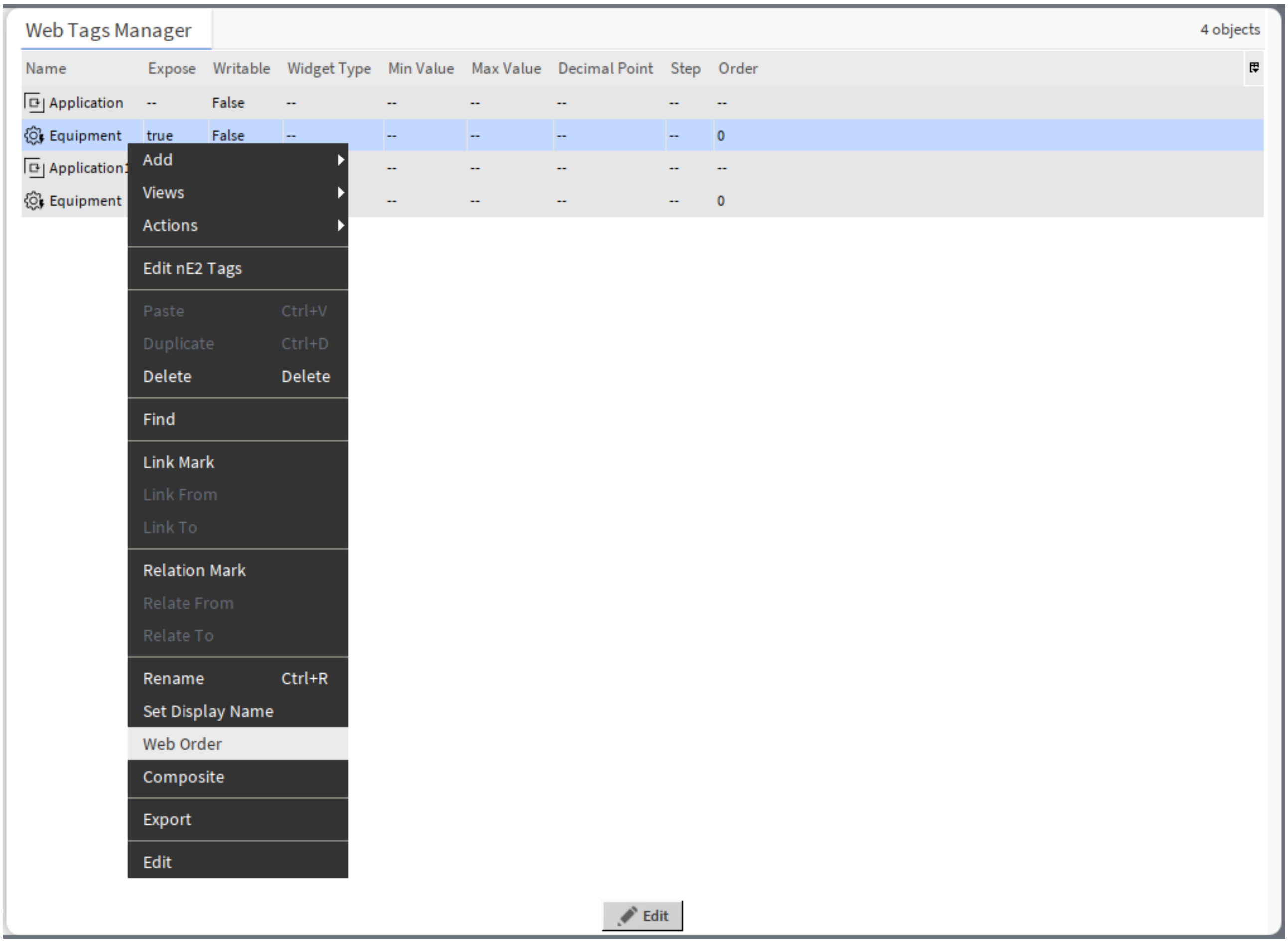
Name	Writable	Widget Type	Min Value	Max Value	Decimal Point	Step
AnalogDataPoint	True	step	0,00	100,00	1	1,00

Name Cannot edit
Writable Marker
Widget Type step
Min Value 0,00
Max Value 100,00
Decimal Point 1
Step 1,00

Figure 167. Web Tags Manager editing dialog window

Web Order

For the nanoWebUI™ display clarity and accuracy of representation of the user's requirements, in the Web tags view, it is possible to edit a web order of Data Points displayed in the nanoWebUI™.



Name	Expose	Writable	Widget Type	Min Value	Max Value	Decimal Point	Step	Order
Application	--	False	--	--	--	--	--	--
Equipment	true	False	--	--	--	--	--	0
Application1				--	--	--	--	--
Equipment				--	--	--	--	0

Figure 168. Web order option

The Web Order option is available in the context menu of the Application or Equipment components. It opens a dialog window with a list of Data Points.

Use the Move Up/Move Down buttons to determine the display order. Confirm with OK.

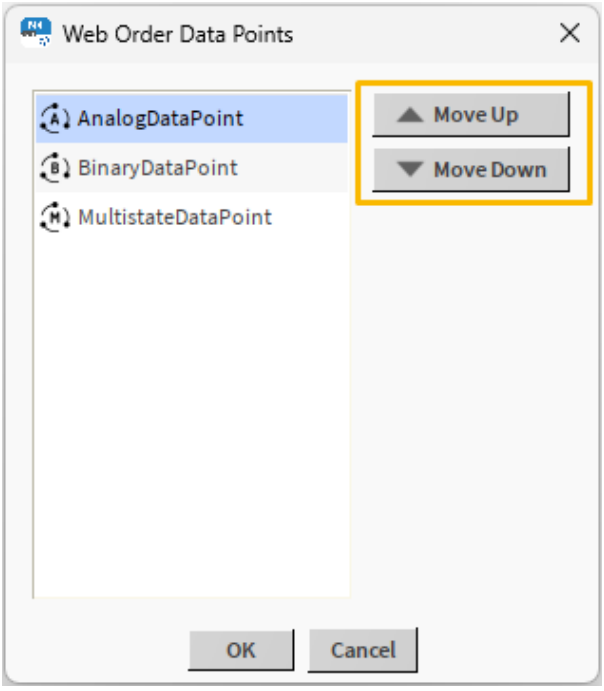


Figure 169. Web order dialog window

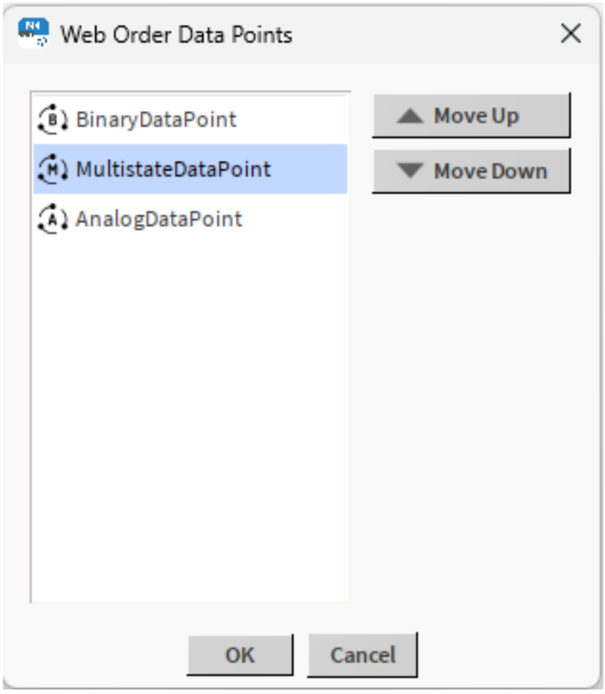


Figure 171. Changed order of Data Points

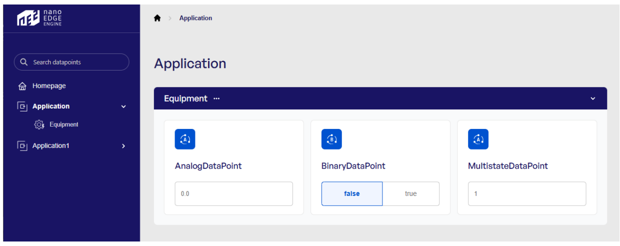


Figure 170. Original display in the nanoWebUI™

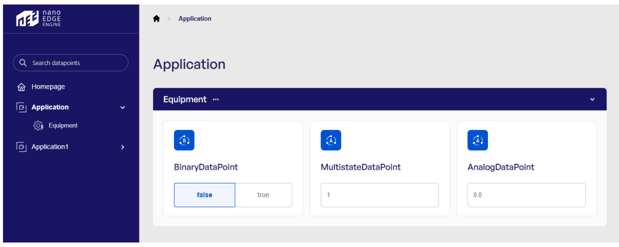


Figure 172. Changed order display in the nanoWebUI™

5.7.5 Haystack Service

The Haystack service enables the Haystack data modelling based on tagging. The Haystack service provides a Haystack dictionary of tags available to add to Data Points and the Equipment components.

The nano EDGE ENGINE implementation is based on the Haystack HTTP API, which defines a simple mechanism to exchange Haystack tagged data over HTTP. The Haystack HTTP API server executes operations: receives requests and returns responses. Operations are pluggable, which means the API can be further enhanced, and the implementation includes a pluggable authentication protocol.

The Haystack service is automatically available in the Services container once the Haystack library is installed on the device. The Haystack library is a default part of the nano EDGE ENGINE from the OS V1.9 and is not compatible with earlier versions.

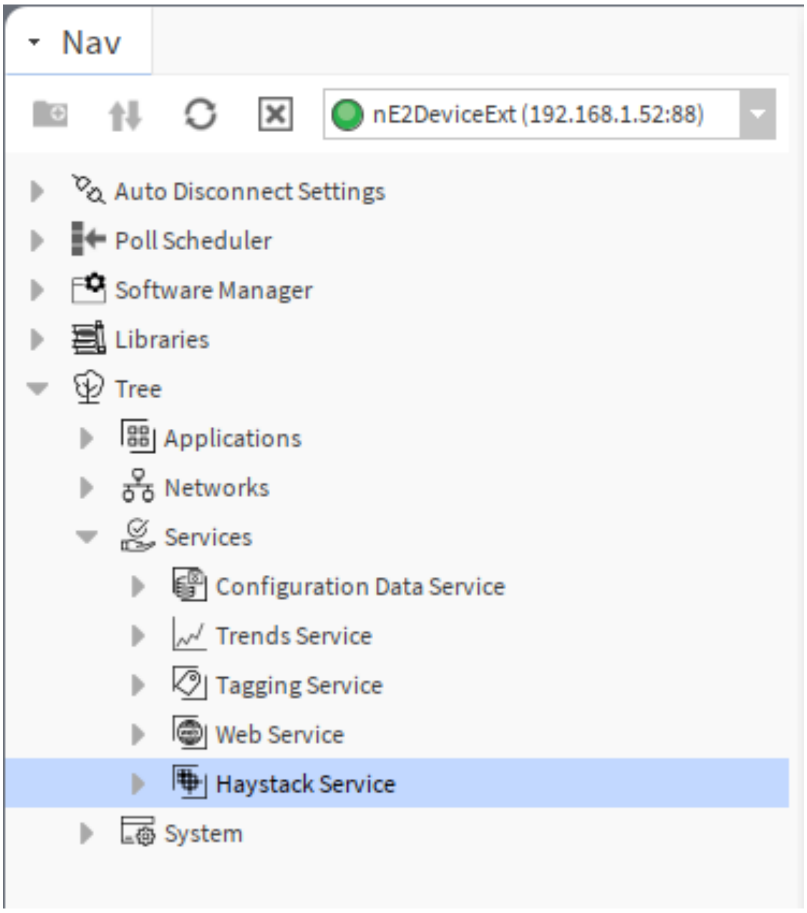
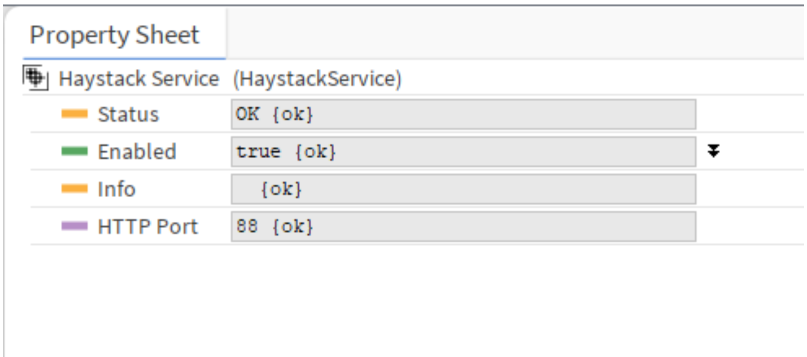


Figure 173. Haystack service in the nav tree



Property Sheet

Haystack Service (HaystackService)

Status	OK {ok}
Enabled	true {ok}
Info	{ok}
HTTP Port	88 {ok}

Figure 174. Haystack service slots

The Haystack service has the following slots:

- Status: indicates the current status of the component. If the component works properly, its status is OK; however, it changes accordingly when values in other slots are adjusted.
    - Available information: disabled (the Enabled slot is set to false), OK;
- Info: informs about a required device restart after changing the port's number;
- HTTP Port: shows the set communication port number.

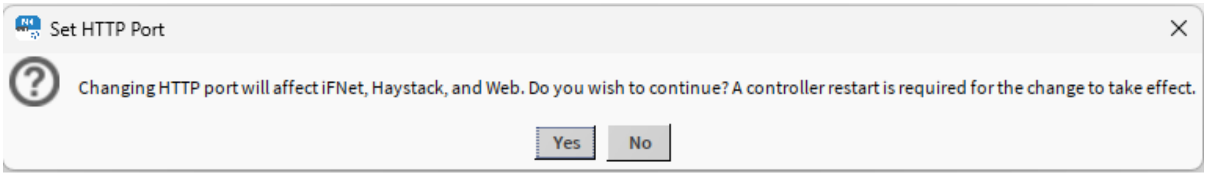
The Haystack service has the following action:

- Set HTTP Port: allows to change the HTTP communication port.

> The Set HTTP Port action is available also in the iFnet component and Web service.

Warning!
Changing the HTTP port affects the iFnet communication port number and will disable connecting with the controller using the current iFnet port number.

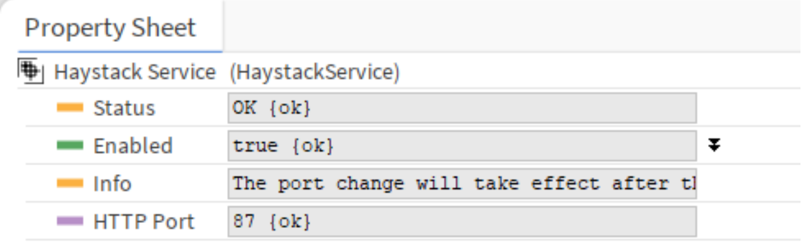
The HTTP port number can be changed by the user using the Set HTTP Port action. Changing the HTTP port number affects the communication port used to connect to the device, Haystack service, and connection to the nanoWebUI™ interface. The following notification is displayed before changing the port number:


Set HTTP Port

Changing HTTP port will affect iFNet, Haystack, and Web. Do you wish to continue? A controller restart is required for the change to take effect.

Figure 175. HTTP port change notification

Once the change is introduced, it requires saving and restarting the device. Before saving, the component displays a notice in the Info slot:


Property Sheet

Haystack Service	(HaystackService)
Status	OK {ok}
Enabled	true {ok}
Info	The port change will take effect after tl
HTTP Port	87 {ok}

Figure 176. Info slot notification after HTTP port change

HTTP API

The Haystack HTTP API has been implemented to facilitate exchanging data between servers and devices. The Haystack HTTP API server executes operations: receives requests and returns responses.

Haystack HTTP API
To learn more about the Haystack HTTP API, please visit: Project Haystack - HTTP API.

The nano EDGE ENGINE implementation supports the following version:

- API Protocol: Haystack Protocol (haystack-core 3.0.3).

Supported Requests

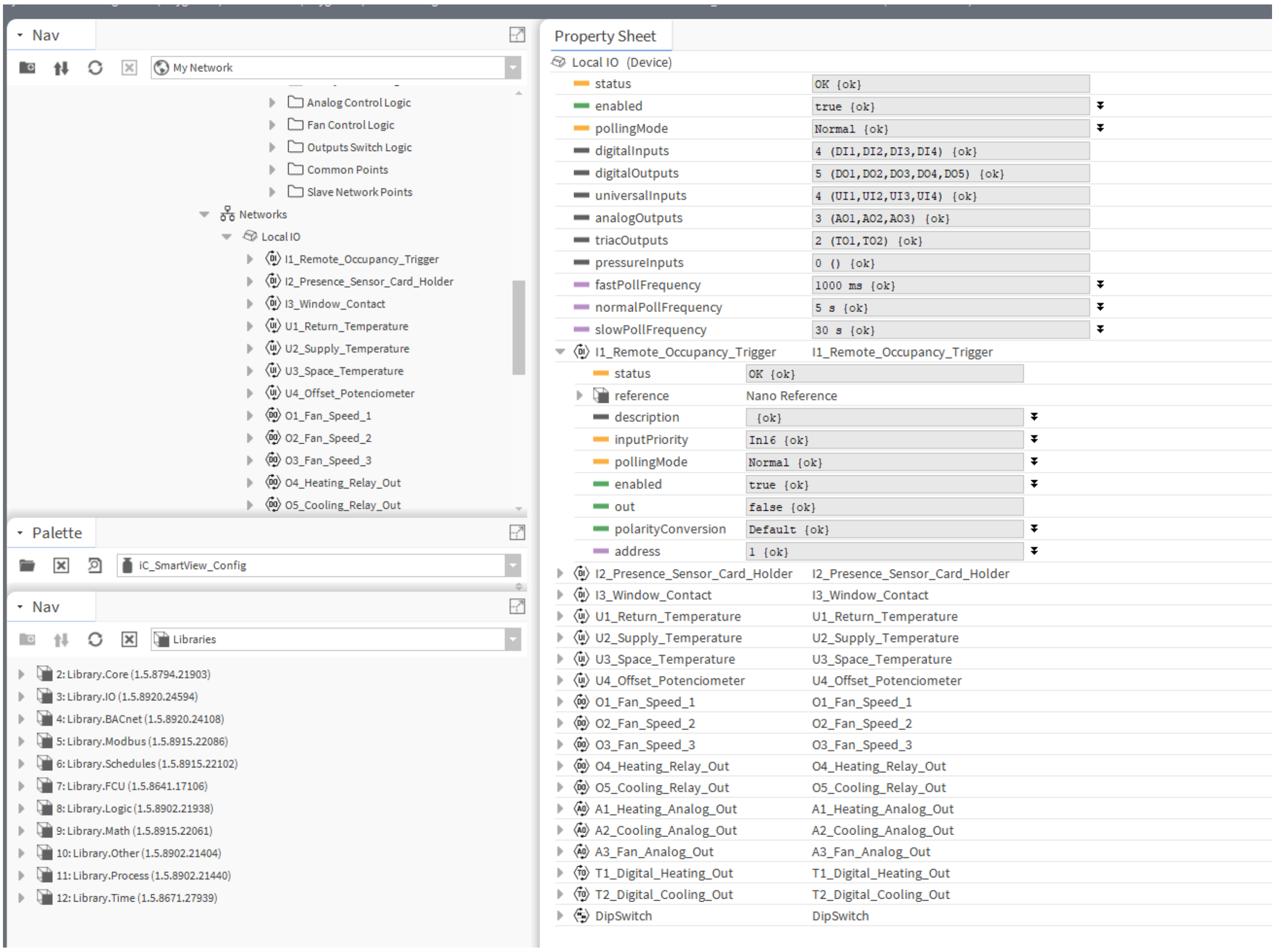
- /haystack/about/
- /haystack/read/
- /haystack/watchSub/
- /haystack/watchPoll/

The auto-refresh mechanism is set to 5 minutes.

5.8 Local IO

5.8.1 Configuration

To setup local inputs and outputs on the nano EDGE ENGINE device, expand the Library.IO. The IO components must be placed in the Networks container, under the LocalIO component.



```
Analog Control Logic
Fan Control Logic
Outputs Switch Logic
Common Points
Slave Network Points
Networks
  Local IO
    I1_Remote_Occupancy_Trigger
    I2_Presence_Sensor_Card_Holder
    I3_Window_Contact
    U1_Return_Temperature
    U2_Supply_Temperature
    U3_Space_Temperature
    U4_Offset_Potenciometer
    O1_Fan_Speed_1
    O2_Fan_Speed_2
    O3_Fan_Speed_3
    O4_Heating_Relay_Out
    O5_Cooling_Relay_Out
```

```
2: Library.Core (1.5.8794.21903)
3: Library.IO (1.5.8920.24594)
4: Library.BACnet (1.5.8920.24108)
5: Library.Modbus (1.5.8915.22086)
6: Library.Schedules (1.5.8915.22102)
7: Library.FCU (1.5.8641.17106)
8: Library.Logic (1.5.8902.21938)
9: Library.Math (1.5.8915.22061)
10: Library.Other (1.5.8902.21404)
11: Library.Process (1.5.8902.21440)
12: Library.Time (1.5.8671.27939)
```

Local IO (Device)

status	OK {ok}
enabled	true {ok}
pollingMode	Normal {ok}
digitalInputs	4 (DI1,DI2,DI3,DI4) {ok}
digitalOutputs	5 (DO1,DO2,DO3,DO4,DO5) {ok}
universalInputs	4 (UI1,UI2,UI3,UI4) {ok}
analogOutputs	3 (AO1,AO2,AO3) {ok}
triacOutputs	2 (TO1,TO2) {ok}
pressureInputs	0 () {ok}
fastPollFrequency	1000 ms {ok}
normalPollFrequency	5 s {ok}
slowPollFrequency	30 s {ok}
I1_Remote_Occupancy_Trigger	I1_Remote_Occupancy_Trigger
status	OK {ok}
reference	Nano Reference
description	{ok}
inputPriority	In16 {ok}
pollingMode	Normal {ok}
enabled	true {ok}
out	false {ok}
polarityConversion	Default {ok}
address	1 {ok}
I2_Presence_Sensor_Card_Holder	I2_Presence_Sensor_Card_Holder
I3_Window_Contact	I3_Window_Contact
U1_Return_Temperature	U1_Return_Temperature
U2_Supply_Temperature	U2_Supply_Temperature
U3_Space_Temperature	U3_Space_Temperature
U4_Offset_Potenciometer	U4_Offset_Potenciometer
O1_Fan_Speed_1	O1_Fan_Speed_1
O2_Fan_Speed_2	O2_Fan_Speed_2
O3_Fan_Speed_3	O3_Fan_Speed_3
O4_Heating_Relay_Out	O4_Heating_Relay_Out
O5_Cooling_Relay_Out	O5_Cooling_Relay_Out
A1_Heating_Analog_Out	A1_Heating_Analog_Out
A2_Cooling_Analog_Out	A2_Cooling_Analog_Out
A3_Fan_Analog_Out	A3_Fan_Analog_Out
T1_Digital_Heating_Out	T1_Digital_Heating_Out
T2_Digital_Cooling_Out	T2_Digital_Cooling_Out
DipSwitch	DipSwitch

Figure 177. LocalIO components added to the Networks container

To learn more about the local IO components, please refer to the nano EDGE ENGINE Programming user manual.

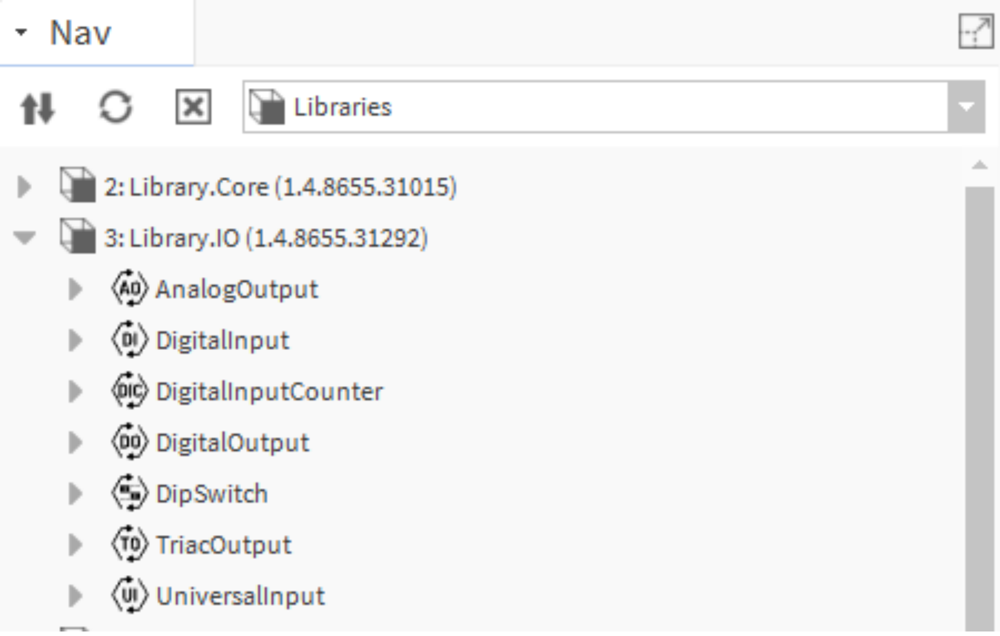


Figure 178. Library.IO

Drag and drop the desired IO component under the Local IO network.

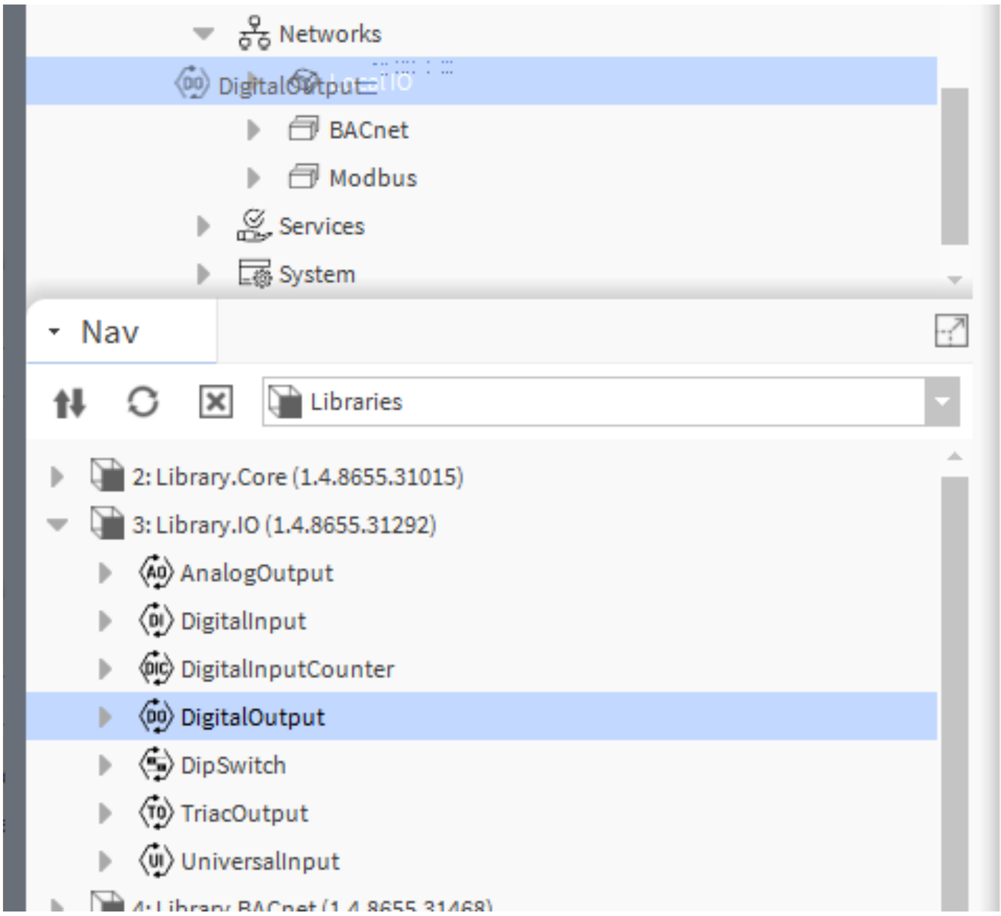
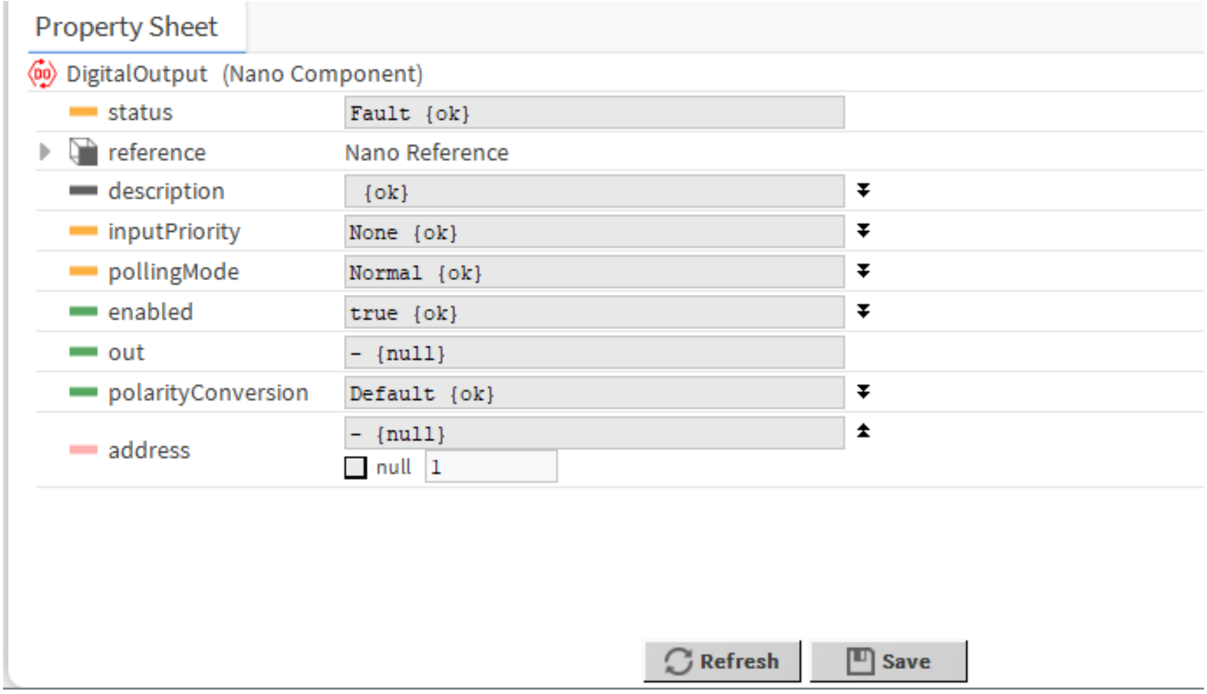


Figure 179. Adding DigitalOutput to the Local IO network

Double-click the IO point to open its property sheet. By default, the status of the point is Fault because the address must be set. Configure the point's address in the Address slot and click Save.



Property Sheet

DigitalOutput (Nano Component)

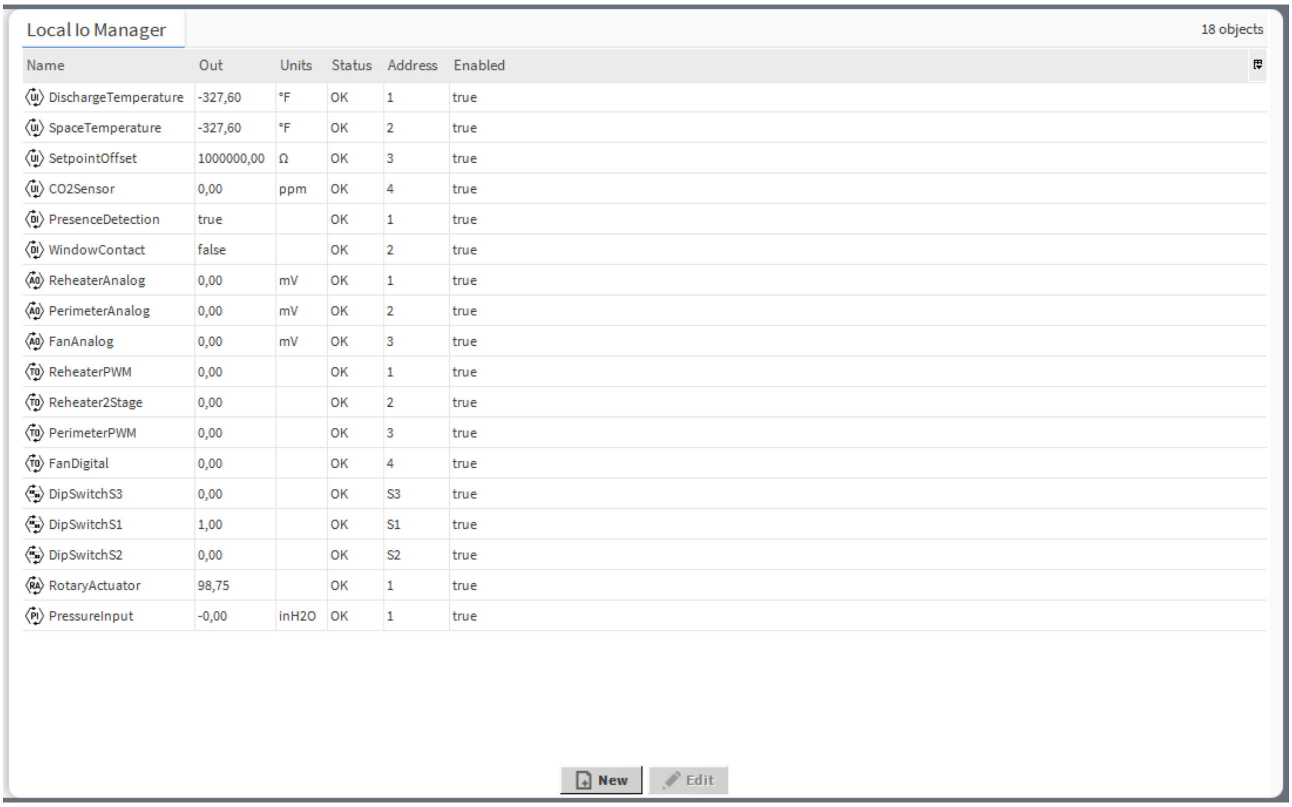
status	Fault {ok}
reference	Nano Reference
description	{ok}
inputPriority	None {ok}
pollingMode	Normal {ok}
enabled	true {ok}
out	- {null}
polarityConversion	Default {ok}
address	- {null} □ null 1

Figure 180. Setting the IO point address

5.8.2 LocalIO Manager

The LocalIO Manager view is available for the LocalIO component. It lists all I/O points added to the LocalIO component, and shows their:

- name;
- Out slot value;
- unit (for analog values);
- status;
- address;
- enabled or disabled state.



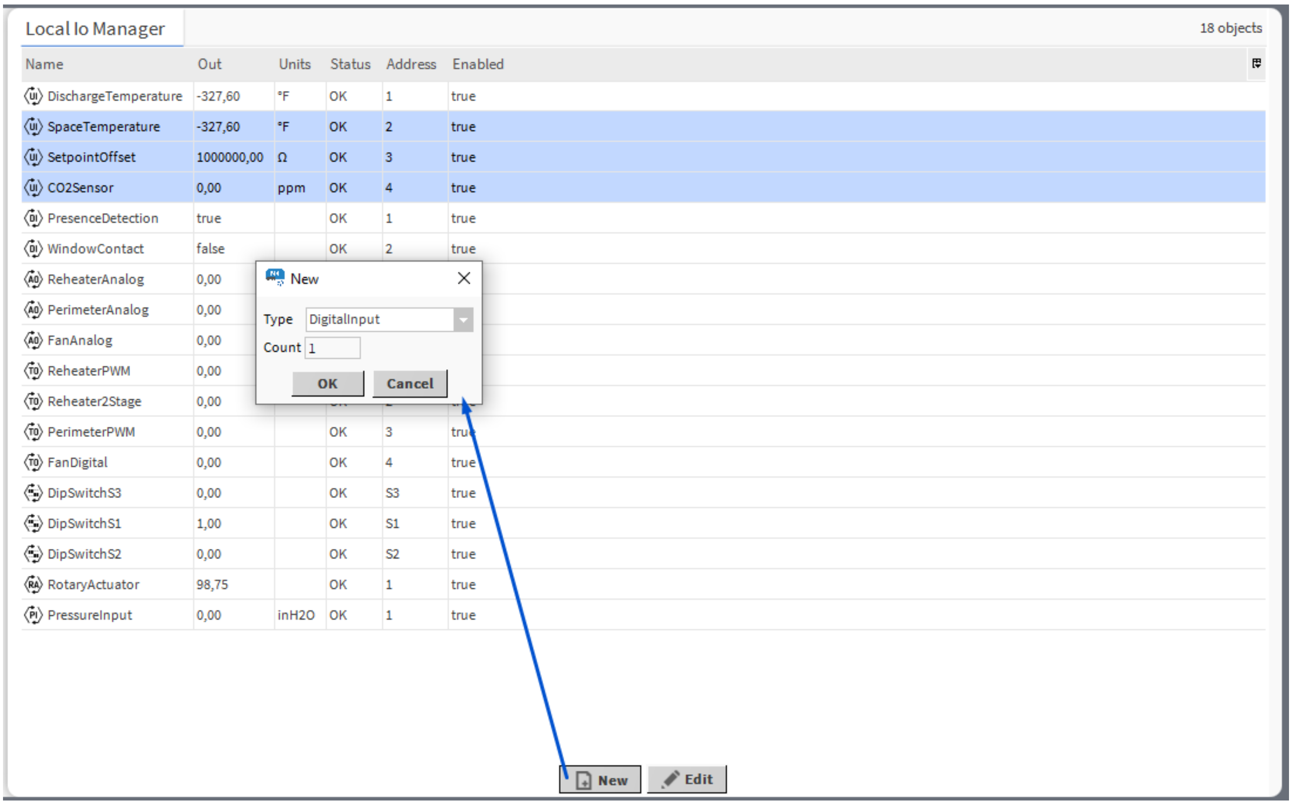
Local Io Manager — 18 objects

Name	Out	Units	Status	Address	Enabled
DischargeTemperature	-327,60	°F	OK	1	true
SpaceTemperature	-327,60	°F	OK	2	true
SetpointOffset	1000000,00	Ω	OK	3	true
CO2Sensor	0,00	ppm	OK	4	true
PresenceDetection	true		OK	1	true
WindowContact	false		OK	2	true
ReheaterAnalog	0,00	mV	OK	1	true
PerimeterAnalog	0,00	mV	OK	2	true
FanAnalog	0,00	mV	OK	3	true
ReheaterPWM	0,00		OK	1	true
Reheater2Stage	0,00		OK	2	true
PerimeterPWM	0,00		OK	3	true
FanDigital	0,00		OK	4	true
DipSwitchS3	0,00		OK	S3	true
DipSwitchS1	1,00		OK	S1	true
DipSwitchS2	0,00		OK	S2	true
RotaryActuator	98,75		OK	1	true
PressureInput	-0,00	inH2O	OK	1	true

Figure 181. LocalIO Manager

In the LocalIO Manager, it is possible to:

- add local I/O components:



Local Io Manager — 18 objects

Name	Out	Units	Status	Address	Enabled
DischargeTemperature	-327,60	°F	OK	1	true
SpaceTemperature	-327,60	°F	OK	2	true
SetpointOffset	1000000,00	Ω	OK	3	true
CO2Sensor	0,00	ppm	OK	4	true
PresenceDetection	true		OK	1	true
WindowContact	false		OK	2	true
ReheaterAnalog	0,00				
PerimeterAnalog	0,00				
FanAnalog	0,00				
ReheaterPWM	0,00				
Reheater2Stage	0,00				
PerimeterPWM	0,00		OK	3	true
FanDigital	0,00		OK	4	true
DipSwitchS3	0,00		OK	S3	true
DipSwitchS1	1,00		OK	S1	true
DipSwitchS2	0,00		OK	S2	true
RotaryActuator	98,75		OK	1	true
PressureInput	0,00	inH2O	OK	1	true

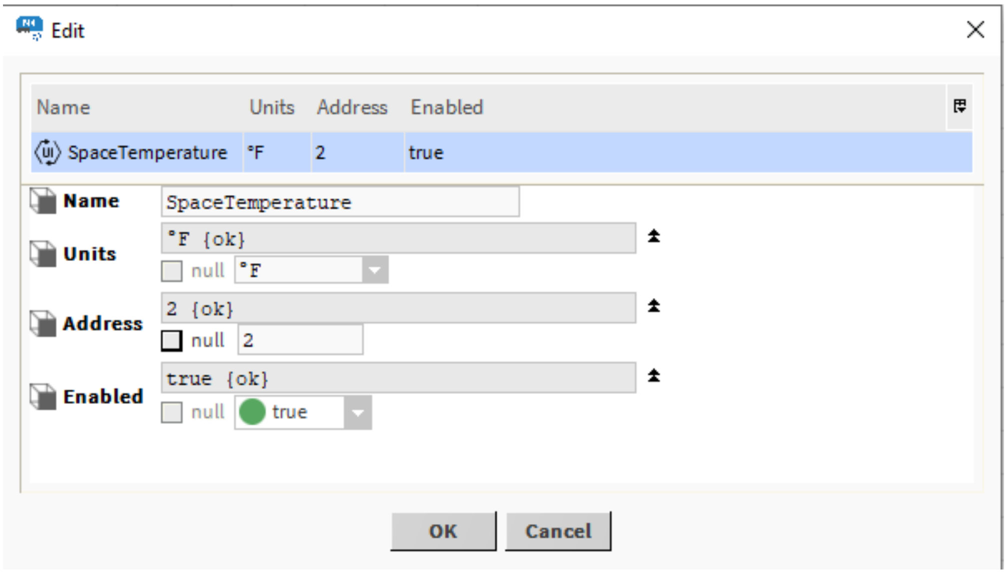
New

Type: DigitalInput

Count: 1

Figure 182. Adding I/O points in the LocalIO Manager

- edit the I/O's name, units, address and enable/disable the component:



Edit

Name	Units	Address	Enabled
SpaceTemperature	°F	2	true

Name: SpaceTemperature

Units: °F {ok}
□ null °F

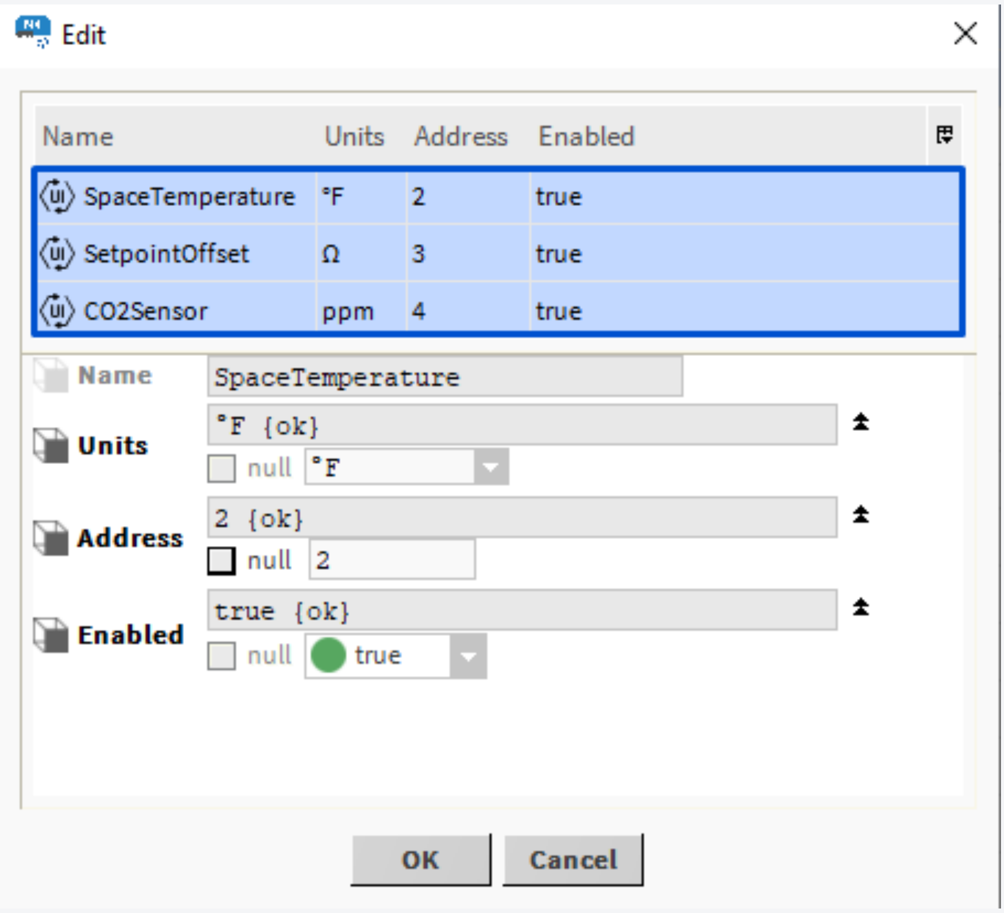
Address: 2 {ok}
□ null 2

Enabled: true {ok}
□ null ● true

Figure 183. Editing pop-up

Note

Editing is possible for more than one I/O point at a time. If multiple points are edited, the same new value is written to common slots, so individual slots, such as Name, cannot be edited in this manner.



Edit

Name	Units	Address	Enabled
SpaceTemperature	°F	2	true
SetpointOffset	Ω	3	true
CO2Sensor	ppm	4	true

Name: SpaceTemperature

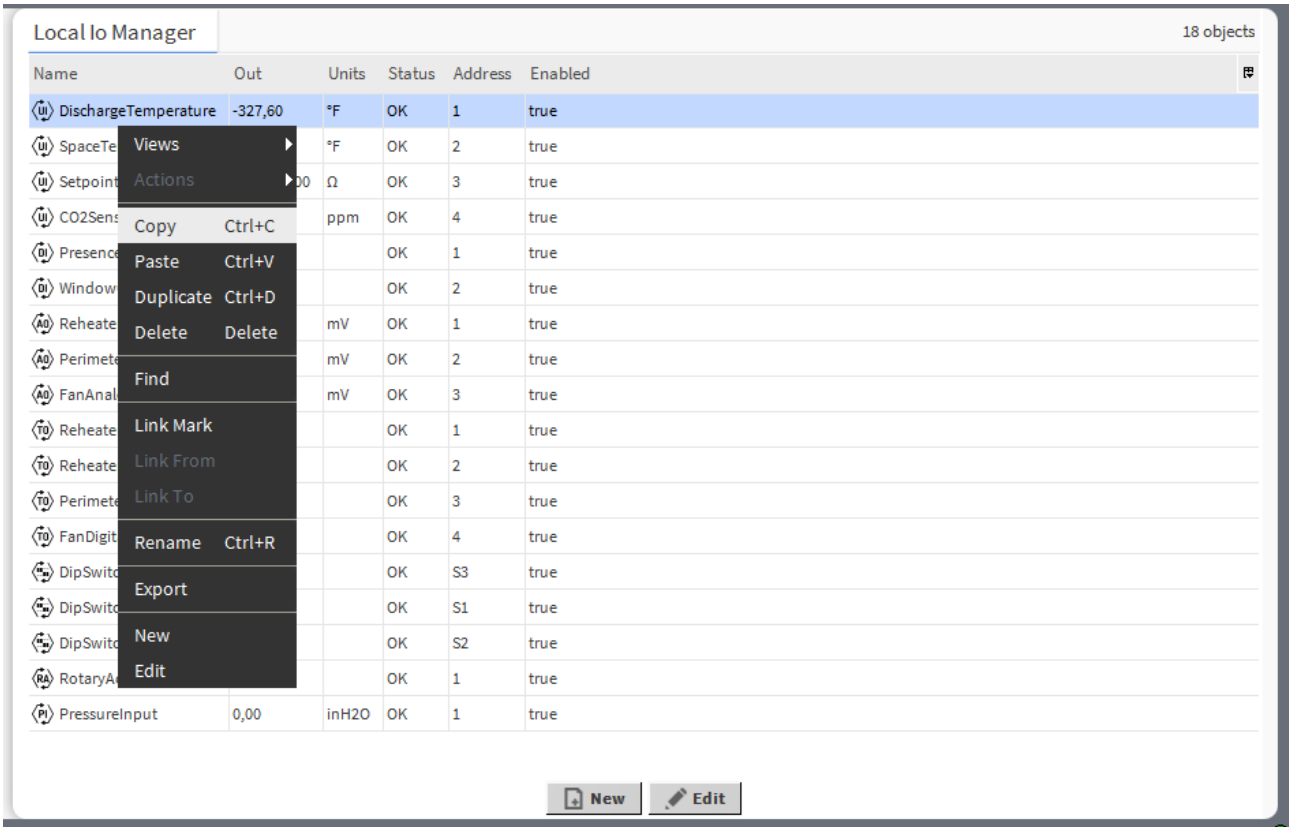
Units: °F {ok}
□ null °F

Address: 2 {ok}
□ null 2

Enabled: true {ok}
□ null ● true

Figure 184. Editing of multiple I/O points

- copy/duplicate/remove I/O components:



Local Io Manager

18 objects

Name	Out	Units	Status	Address	Enabled
DischargeTemperature	-327,60	°F	OK	1	true
SpaceTe		°F	OK	2	true
Setpoint	00	Ω	OK	3	true
CO2Sens		ppm	OK	4	true
Presence			OK	1	true
Window			OK	2	true
Reheate		mV	OK	1	true
Perimete		mV	OK	2	true
FanAnal		mV	OK	3	true
Reheate			OK	1	true
Reheate			OK	2	true
Perimete			OK	3	true
FanDigit			OK	4	true
DipSwitc			OK	S3	true
DipSwitc			OK	S1	true
DipSwitc			OK	S2	true
RotaryA			OK	1	true
PressureInput	0,00	inH2O	OK	1	true

Figure 185. Context menu options for local I/Os

Opening LocalIO Manager

The LocalIO Manager view is accessible from the context menu of the LocalIO component. It is also automatically opened if the LocalIO component is double-clicked in the nav tree window.

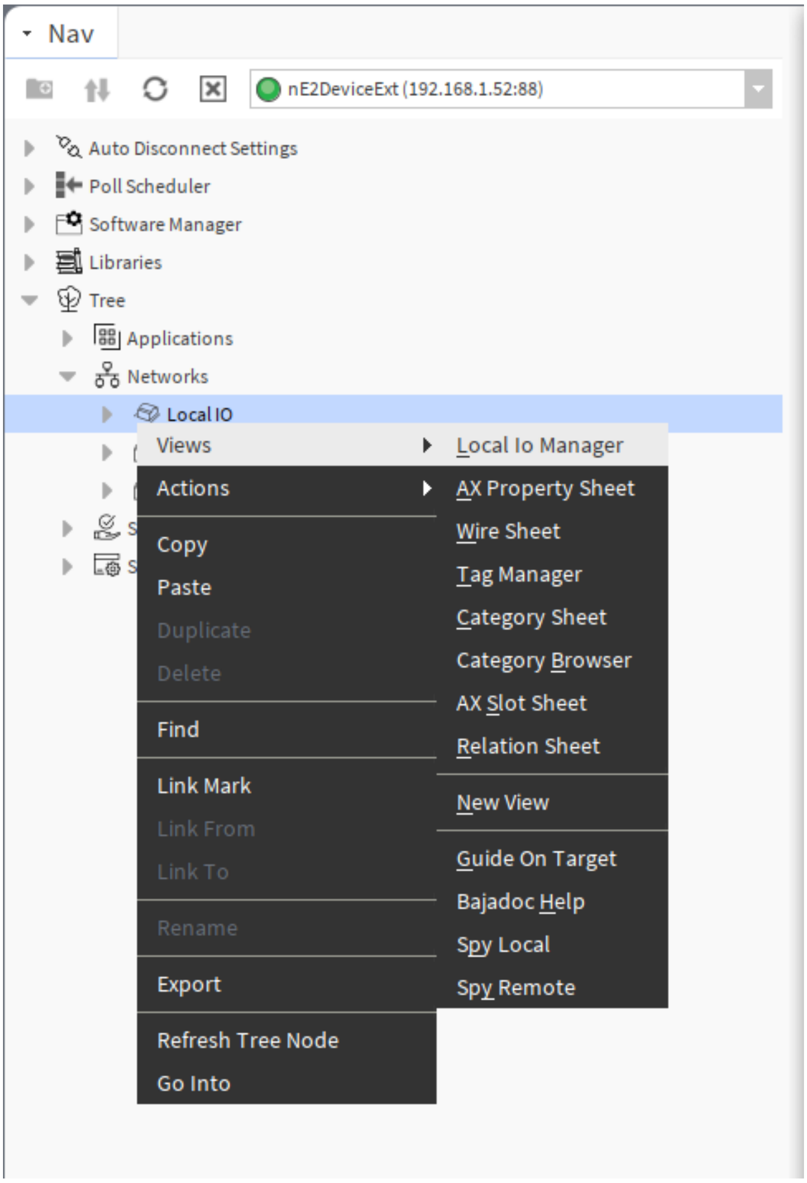
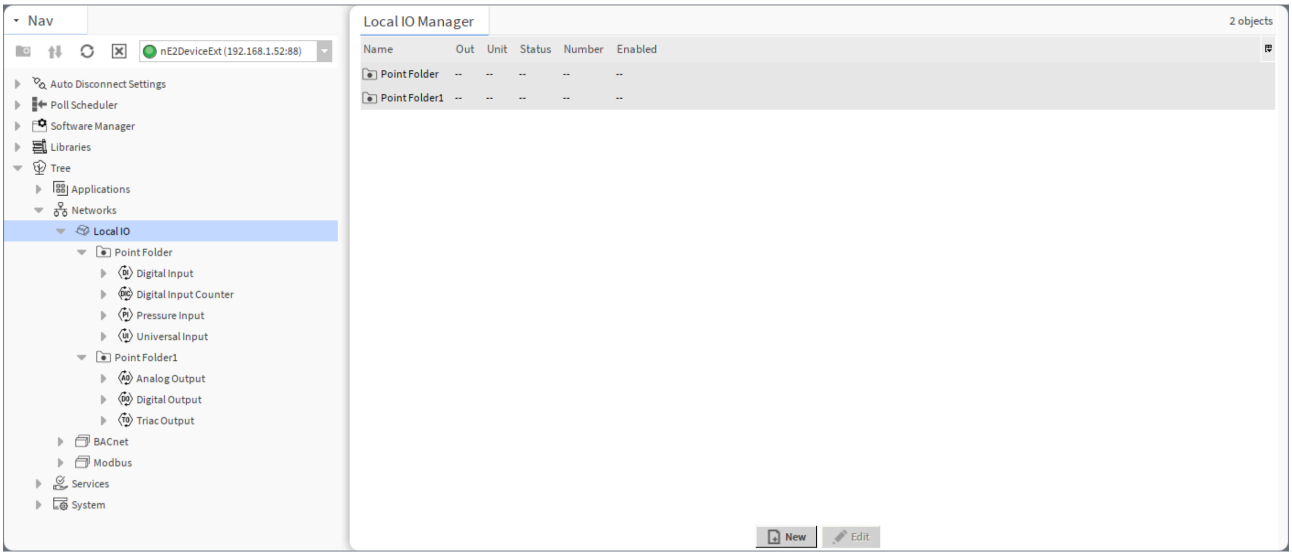


Figure 186. Accessing the LocalIO Manager from the context menu

PointFolder

The PointFolder component is a point grouping component, which can be added under the LocalIO device. It allows to organize points of the device on the nav tree and maintains the grouping in the Point Manager view. It is clickable in the Point Manager allowing to display only points grouped in one PointFolder component.



- Auto Disconnect Settings
- Poll Scheduler
- Software Manager
- Libraries
- Tree
  - Applications
  - Networks
    - Local IO
      - Point Folder
        - Digital Input
        - Digital Input Counter
        - Pressure Input
        - Universal Input
      - Point Folder1
        - Analog Output
        - Digital Output
        - Triac Output
    - BACnet
    - Modbus
  - Services
  - System

Name	Out	Unit	Status	Number	Enabled
Point Folder	--	--	--	--	--
Point Folder1	--	--	--	--	--

Figure 187. PointFolder

The PointFolder component has no slots nor actions. The PointFolder is available in the Core library.

5.9 Linking

In nE2DeviceExt, it is possible to link nano EDGE ENGINE components by dragging a wire from one component to another.

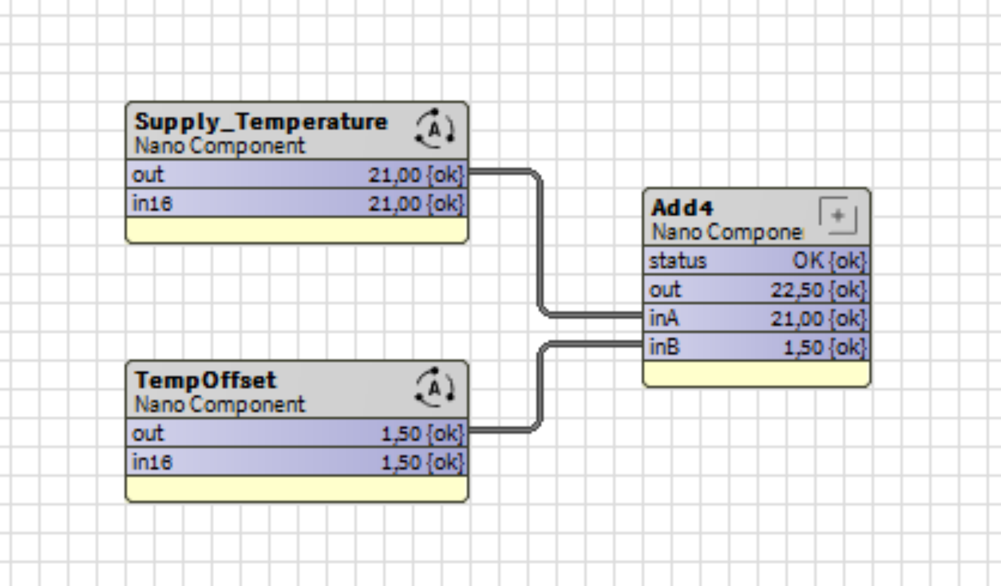


Figure 188. Linked components

Warning!
Linking in nE2DeviceExt works only between nano EDGE ENGINE components. Do not link Niagara components from other modules/drivers with nano EDGE ENGINE components.

5.9.1 Reference Linking

Reference link is designed specifically to connect Data Point class components (in the Applications container) with network point class components (in the Networks container). A reference link transfers values along with the component's status.

To learn more about the reference linking, please refer to the nano EDGE ENGINE Programming user manual.

As network points are situated in the Networks container and Data Points are situated in the Applications container, Reference links are created using the Link Mark and Link From options from the context menu.

To create a reference link, right-click a network point and select the Link Mark option.

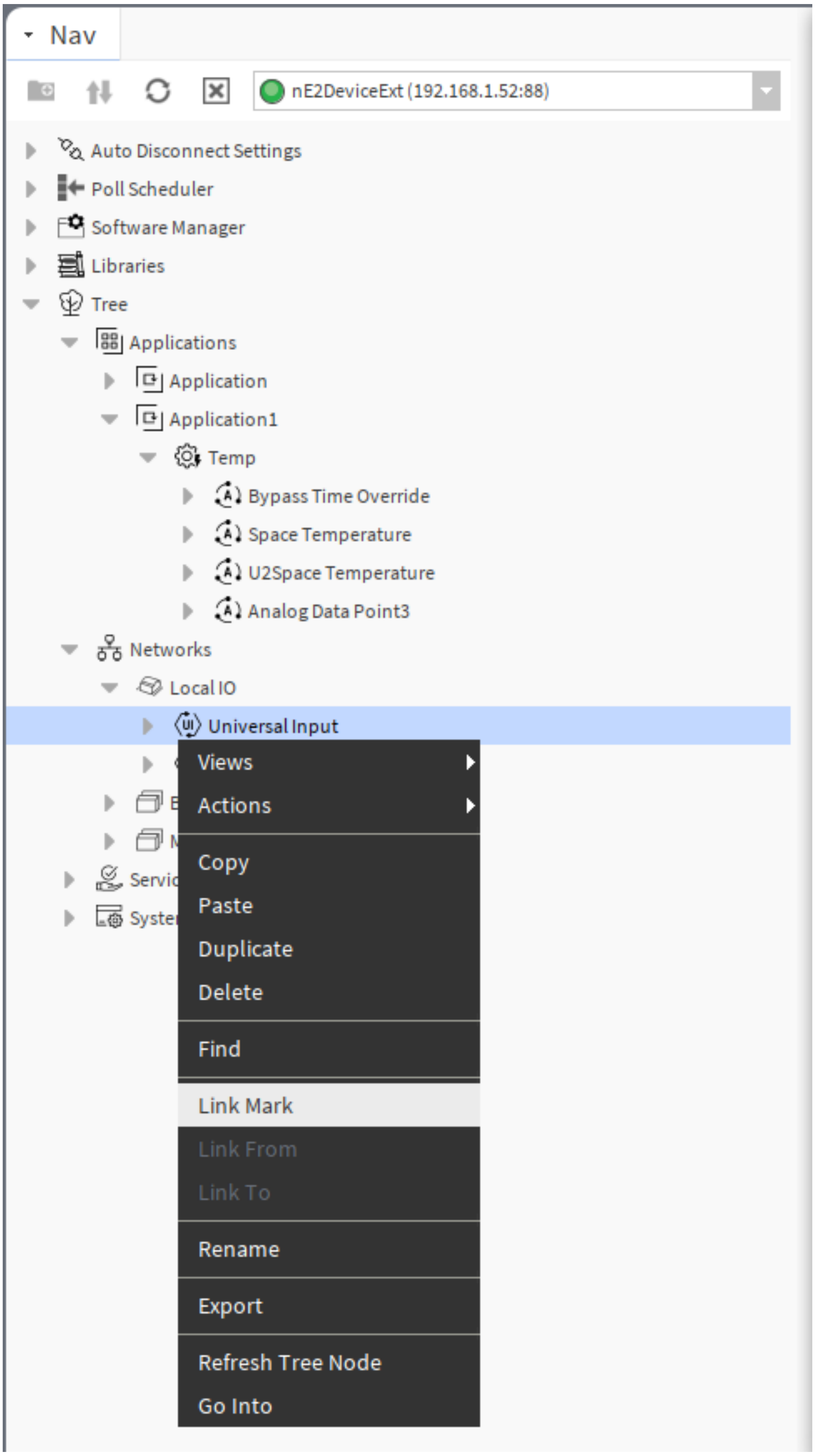


Figure 189. Link Mark option in the network point - UniversalInput

Once the link is marked, right-click on the desired Data Point and click the Link From option.

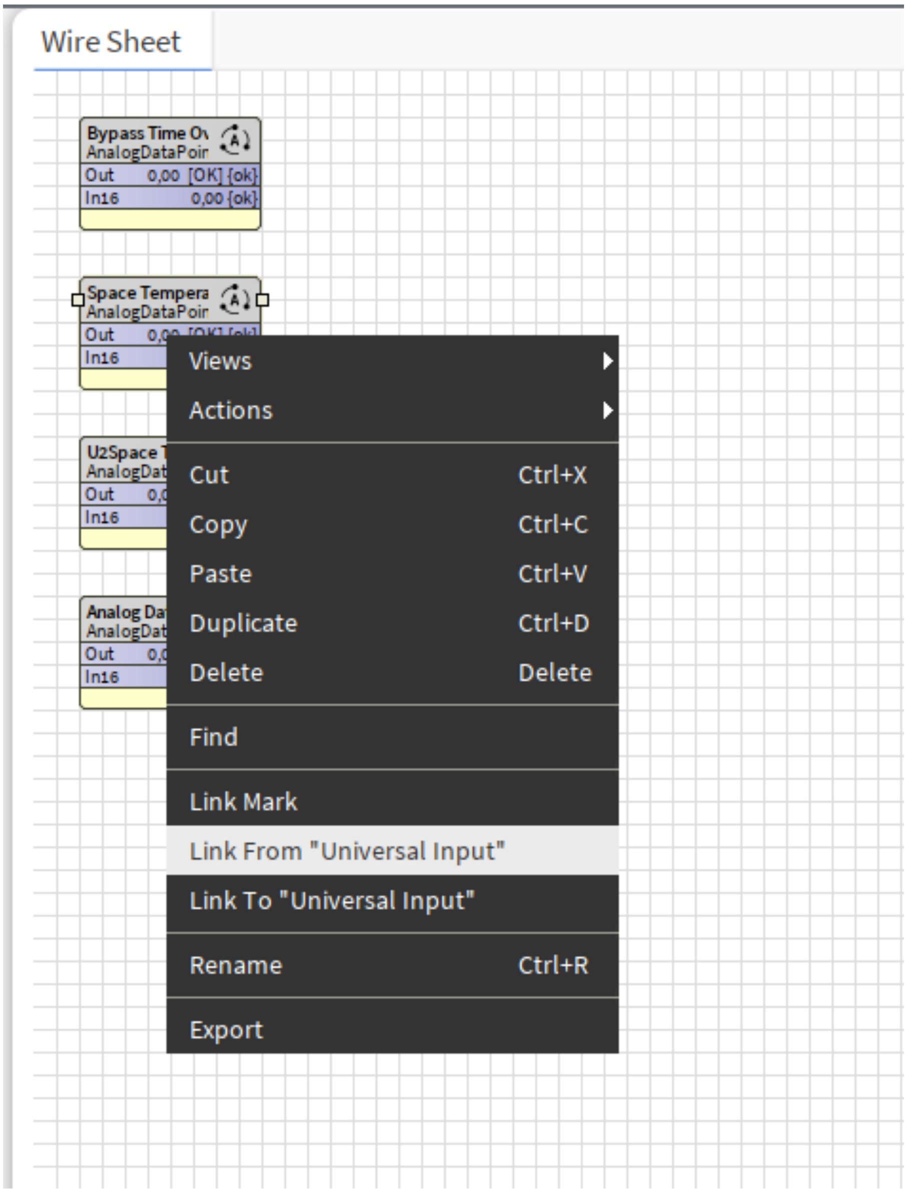
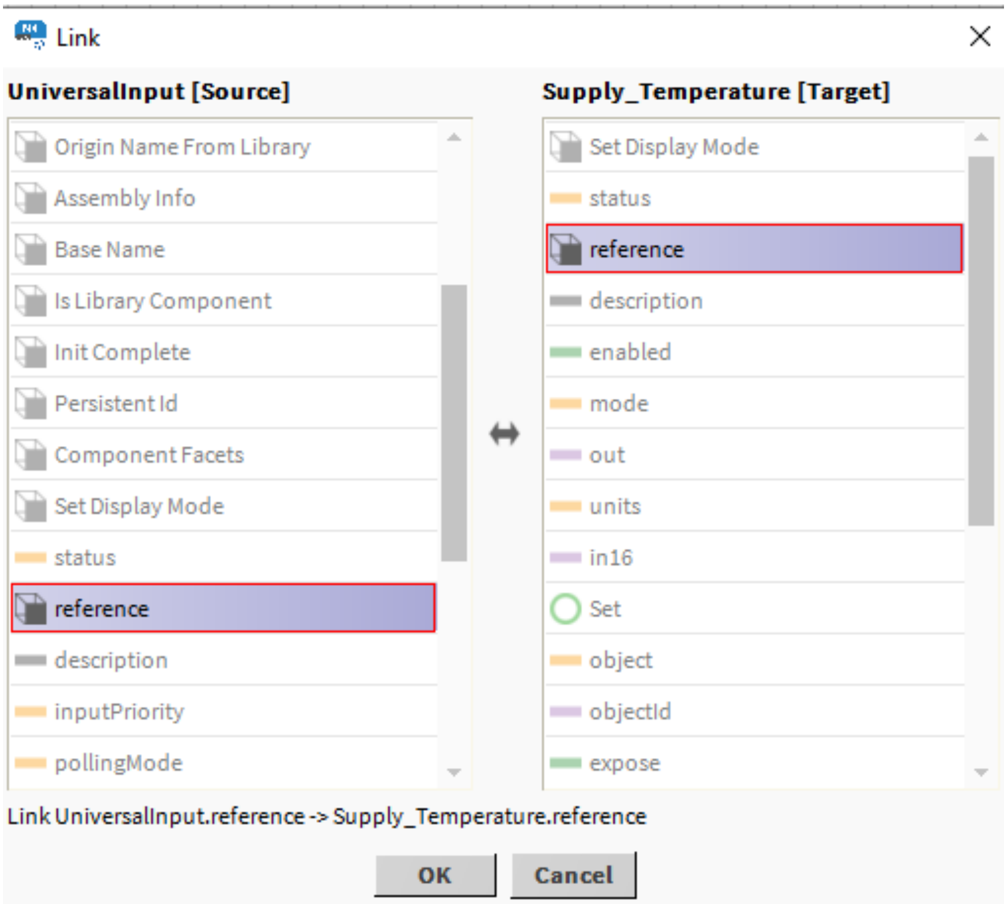


Figure 190. Link From option

In the dialog window that pops up, select “reference” slots on both Source and Target points.



Link

UniversalInput [Source]

- Origin Name From Library
- Assembly Info
- Base Name
- Is Library Component
- Init Complete
- Persistent Id
- Component Facets
- Set Display Mode
- status
- reference
- description
- inputPriority
- pollingMode

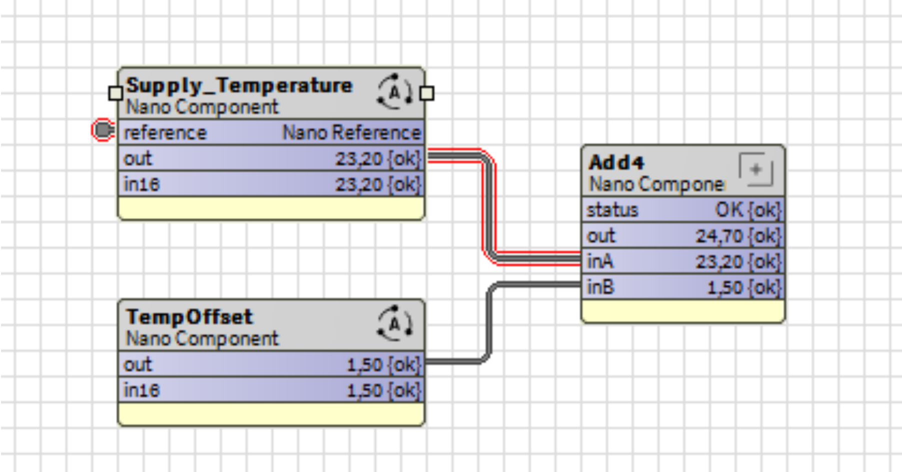
Supply_Temperature [Target]

- Set Display Mode
- status
- reference
- description
- enabled
- mode
- out
- units
- in16
- Set
- object
- objectId
- expose

Link UniversalInput.reference -> Supply_Temperature.reference

Figure 191. Creating reference between points

Once the reference is created, a new 'Nano Reference' slot appears in the component.



Supply_Temperature
Nano Component

reference	Nano Reference
out	23,20 {ok}
in16	23,20 {ok}

Add4
Nano Compone

status	OK {ok}
out	24,70 {ok}
inA	23,20 {ok}
inB	1,50 {ok}

TempOffset
Nano Component

out	1,50 {ok}
in16	1,50 {ok}

Figure 192. Data Point with a reference

5.10 Saving Applications to Niagara Palette

To speed up engineering and working on multiple devices and projects, nE2 Link supports storing applications, equipment, and logic as part of standard Niagara palettes, which can then be copied and pasted on other nano EDGE ENGINE devices.

Note: The controller must be updated to the OS V1.7 and up to properly support this functionality.

The mechanism is part of the standard Workbench functionality. It is supported both on USB and TCP/IP connections.

There are 3 major steps to saving the application or part of the application for future reuse:

- Creating a Custom Palette
- Saving Components
- Using Components from the Palette

5.10.1 Creating a Custom Palette

First, it is required to create a dedicated user palette (or palettes):

- navigate to the desired location in the User File System;
- right-click on the chosen location to open the context menu;
- select New → PaletteFile.palette to create a new palette;
- name the palette (e.g., "CustomComponents") and confirm.

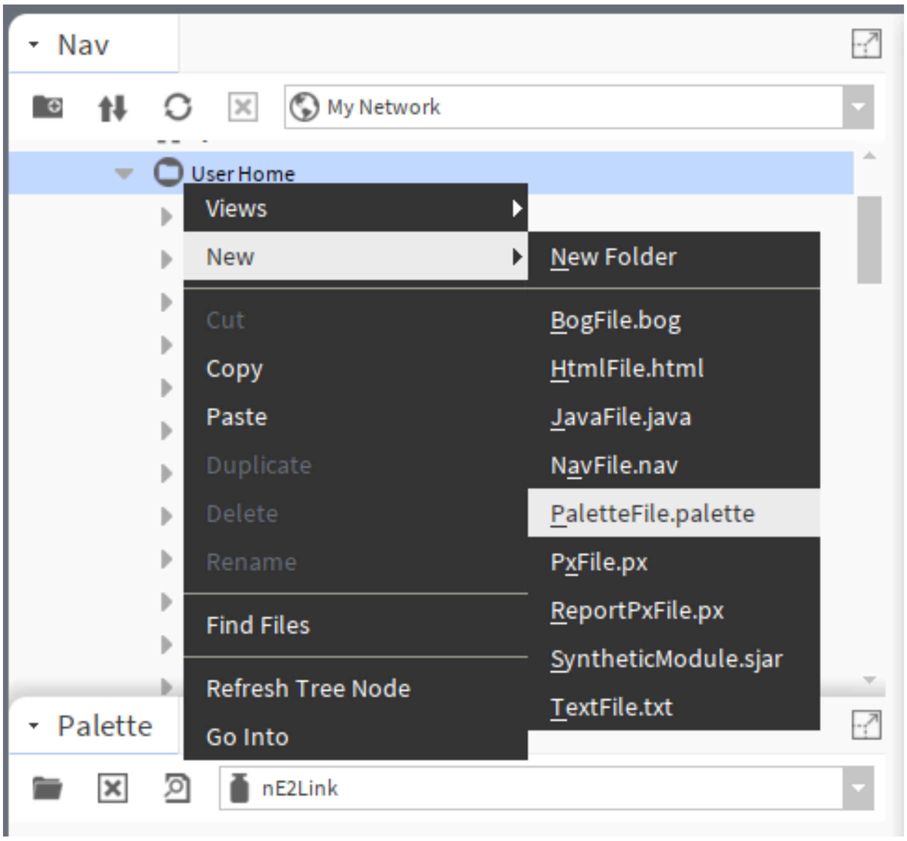


Figure 193. Creating a new palette

5.10.2 Saving Components

To save the logic or parts of the logic, follow these steps:

- locate the components to be saved;

> Note
> It is recommended to save components under the level of the Application component in the tree. Depending on the application's structure, it will be components located in the Equipment or Folder components, or other components located directly under the Application component.

- save the selected components to the created palette:

    - mark the components in the application tree and drag and drop them into the custom palette,
    - mark the components in the application tree, copy them and then paste into the custom palette;
- save the palette with new components using the Save option in the palette workspace.

5.10.3 Using Components from the Palette

To use components saved to the palette:

- open the target station;
- open the palette workspace and click Open Palette;
- reuse components from palette:
    - drag the component(s) from the palette into the application tree in the target station and drop them in the Applications container under the Application component.
    - copy the component(s) from the palette and paste them into the application tree in the target station (in the Applications container under the Application component).

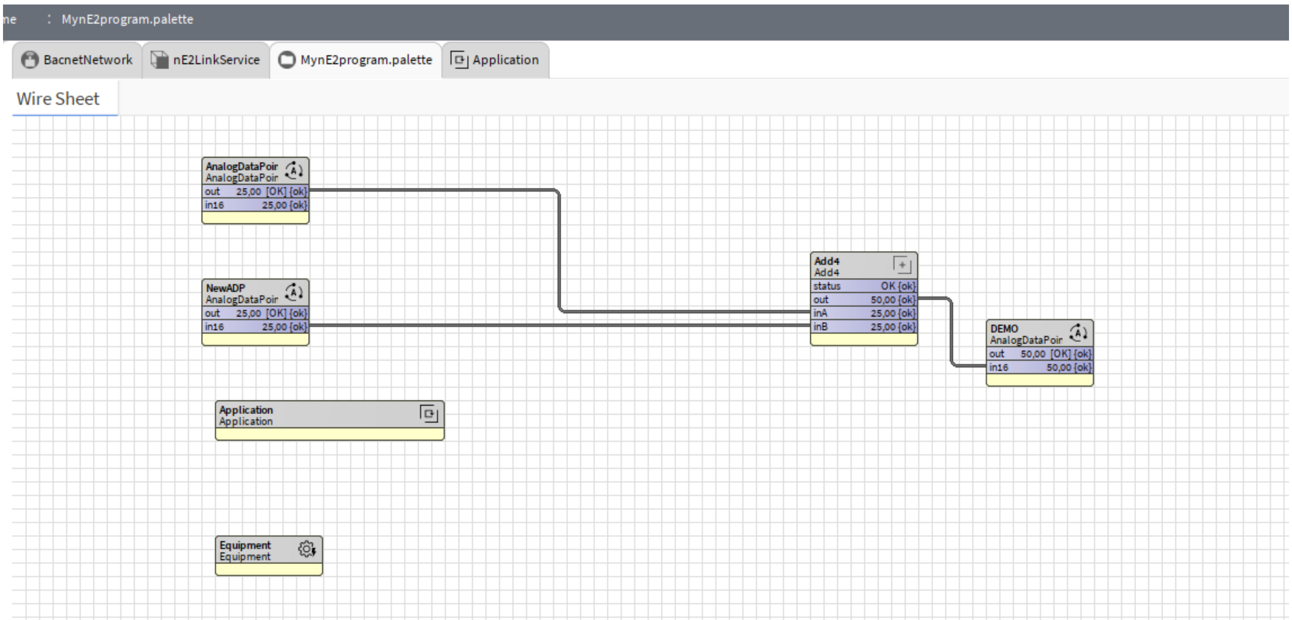


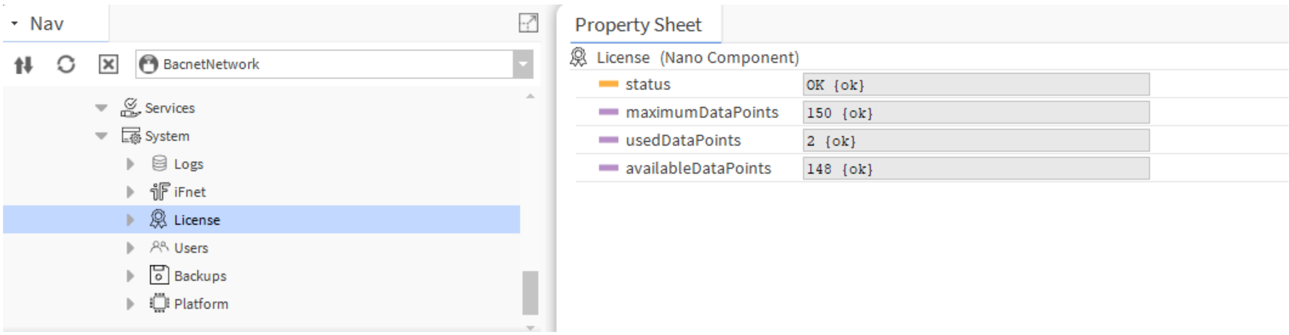
Figure 194. Copied Application and Equipment on the wire sheet

Note: The palette cannot be used for offline programming. The control logic or application must first be copied from the controller in order to work properly when copied back to the controller application.

6 Integration to Niagara

Warning!
To integrate data in the Niagara Framework, user must use standard Niagara networks. nano EDGE ENGINE components must not be linked to Niagara components.

To integrate with Niagara, it is important to note that only Data Points can be exposed over networks. Each nano EDGE ENGINE device has a limit on the number of Data Points that can be exposed. The available number of Data Points can be found in the License component in the System container.



status	OK {ok}
maximumDataPoints	150 {ok}
usedDataPoints	2 {ok}
availableDataPoints	148 {ok}

Figure 195. Number of available Data Points visible in the License component

6.1 Exposing Data Points

To integrate Data Points with Niagara, the points must be exposed over networks. By default, nano EDGE ENGINE exposes points over BACnet and Modbus.

In BACnet, all Data Points are exposed as BACnet objects by default. Individual Data Points can be hidden by manually changing the Expose slot value in their BACnet extension (e.g., BACnetAnalogPoint, BACnetBinaryPoint, BACnetMultistatePoint). The BACnet object type and object Id is visible in the Data Point BACnet extension. To change the Data Point's BACnetID, right-click on the Data Point and select the SetId action.

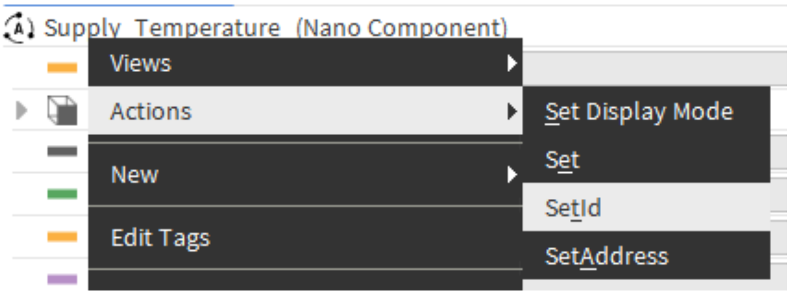
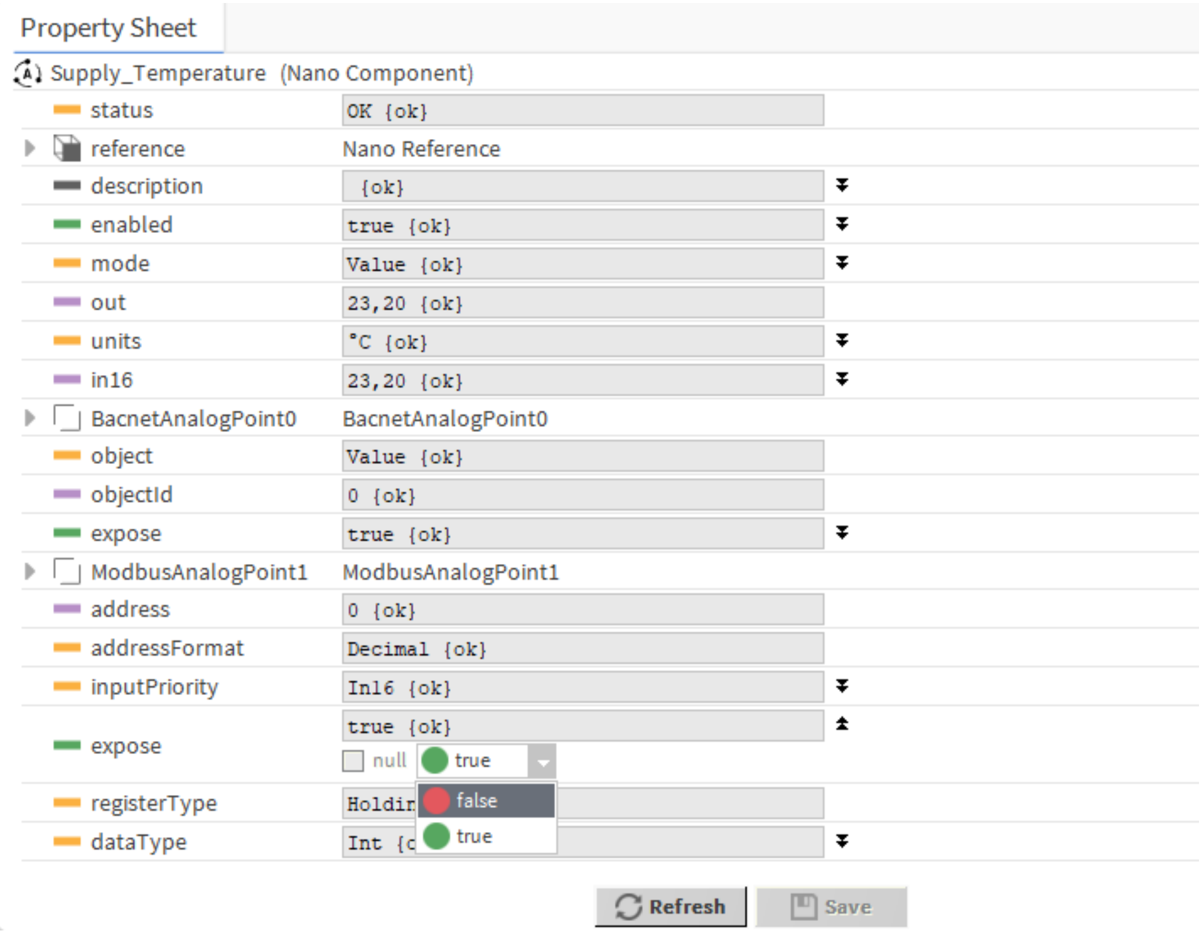


Figure 196. Setting ObjectID to the Data Point

In Modbus, all Data Points in the device are automatically exposed as the Modbus server device. In order to disable the Data Point in the Modbus server network, either set the Autoexposition slot in the Modbus component to false (all Data Points hidden) or go to each Data Point individually and set the Expose slot to false.



Property Sheet

Supply_Temperature (Nano Component)

status	OK {ok}
reference	Nano Reference
description	{ok}
enabled	true {ok}
mode	Value {ok}
out	23,20 {ok}
units	°C {ok}
in16	23,20 {ok}
BacnetAnalogPoint0	BacnetAnalogPoint0
object	Value {ok}
objectId	0 {ok}
expose	true {ok}
ModbusAnalogPoint1	ModbusAnalogPoint1
address	0 {ok}
addressFormat	Decimal {ok}
inputPriority	In16 {ok}
expose	true {ok} □ null ● true
registerType	Holdir
dataType	Int {c

Figure 197. Possibility to disable the exposition on Modbus or BACnet in the Data Point's extension

Modbus address is set automatically. In order to set Modbus address manually, right-click the Data Point and select the SetAddress action.

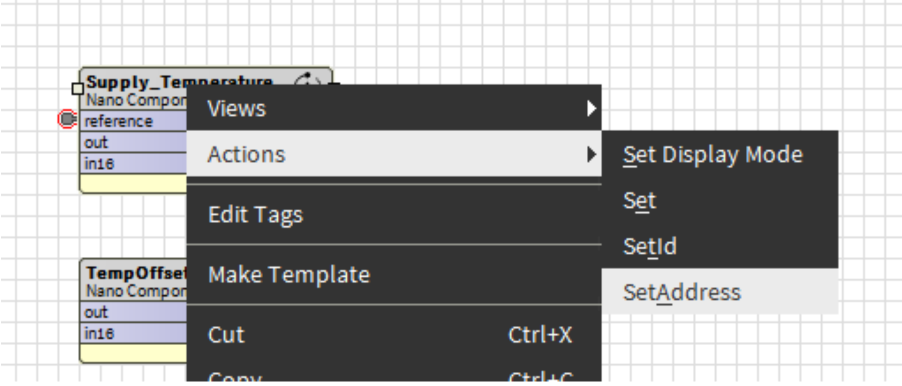
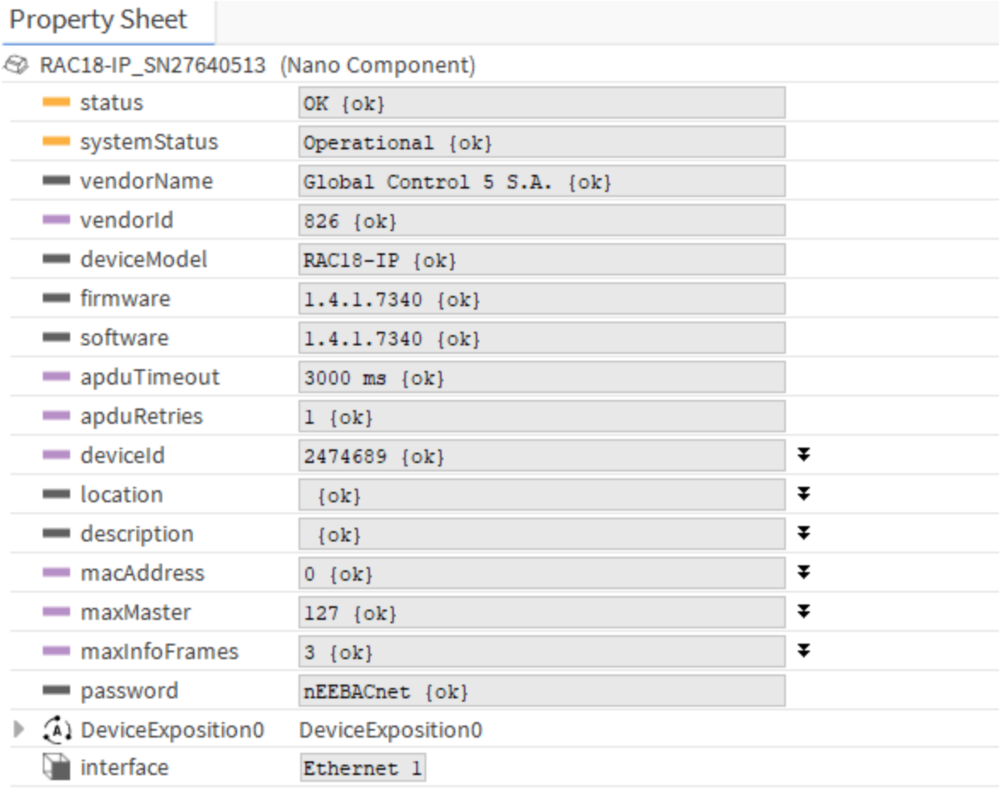


Figure 198. The SetAddress action

6.2 BACnetNetwork Niagara Integration

To integrate the nano EDGE ENGINE device and Data Points exposed over BACnet, make sure the LocalDevice component is properly configured. To change the BACnet Device settings, go to Networks → BACnet → LocalDevice, e.g., RAC18-IP.

> To learn more about the LocalDevice, please refer to the nano EDGE ENGINE Programming user manual.



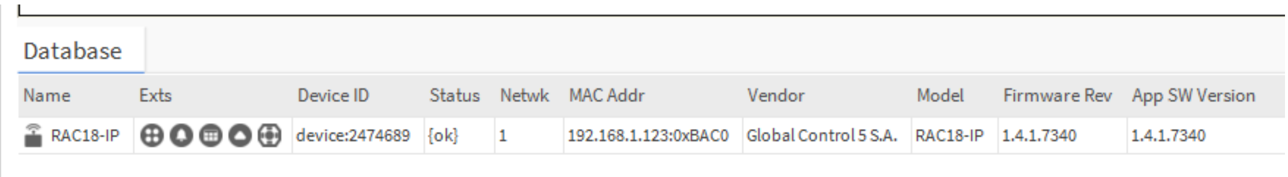
Property Sheet

RAC18-IP_SN27640513 (Nano Component)

status	OK {ok}
systemStatus	Operational {ok}
vendorName	Global Control 5 S.A. {ok}
vendorId	826 {ok}
deviceModel	RAC18-IP {ok}
firmware	1.4.1.7340 {ok}
software	1.4.1.7340 {ok}
apduTimeout	3000 ms {ok}
apduRetries	1 {ok}
deviceId	2474689 {ok}
location	{ok}
description	{ok}
macAddress	0 {ok}
maxMaster	127 {ok}
maxInfoFrames	3 {ok}
password	nEEBACnet {ok}
DeviceExposition0	DeviceExposition0
interface	Ethernet 1

Figure 199. LocalDevice property sheet

Once the deviceID and other parameters are properly configured, go to the BACnetNetwork device in the Niagara station and make sure the configuration of the device is correct.

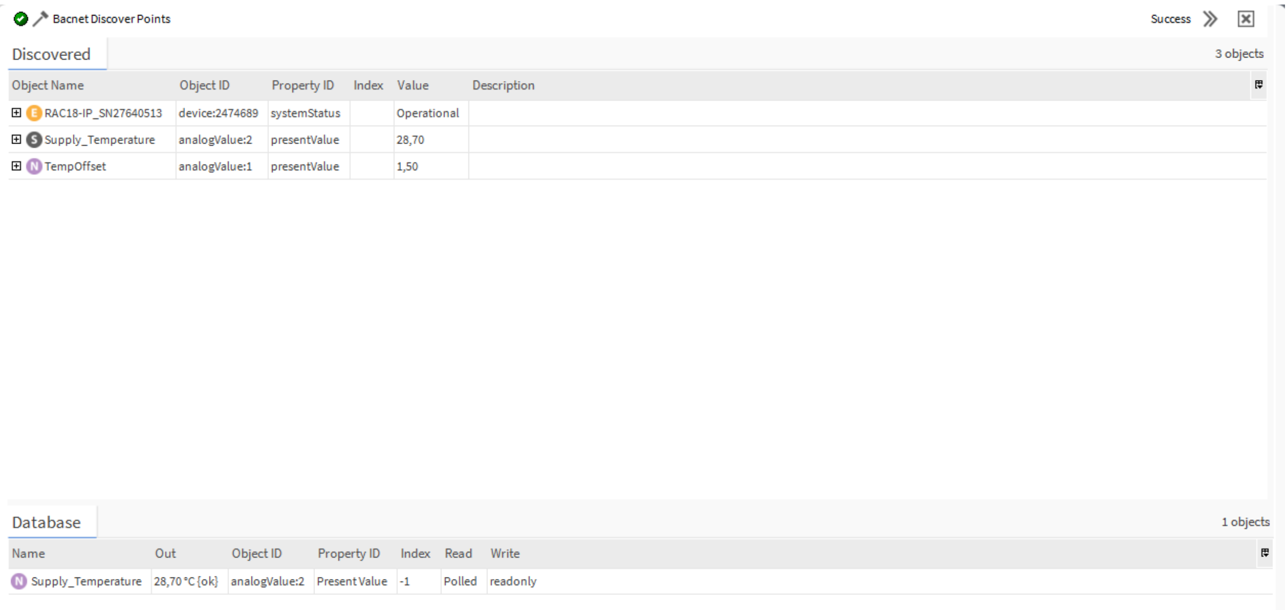


Database

Name	Exts	Device ID	Status	Netwk	MAC Addr	Vendor	Model	Firmware Rev	App SW Version
RAC18-IP		device:2474689	{ok}	1	192.168.1.123:0xBAC0	Global Control 5 S.A.	RAC18-IP	1.4.1.7340	1.4.1.7340

Figure 200. RAC18-IP integrated to Niagara over the BACnet network

To integrate points, go to the device → Points and click Discover. Add required points to Niagara database.



Bacnet Discover Points

Success

Discovered

3 objects

Object Name	Object ID	Property ID	Index	Value	Description
⊞ E RAC18-IP_SN27640513	device:2474689	systemStatus		Operational	
⊞ S Supply_Temperature	analogValue:2	presentValue		28,70	
⊞ N TempOffset	analogValue:1	presentValue		1,50	

Database

1 objects

Name	Out	Object ID	Property ID	Index	Read	Write
N Supply_Temperature	28,70 °C {ok}	analogValue:2	Present Value	-1	Polled	readonly

Figure 201. Points added to the Niagara BACnet database

The points have been successfully integrated into the Niagara BACnet network.

6.3 Trend Logs

With the nE2 Link V1.8 release, the Trends service functionality has been added to the nano EDGE ENGINE operating system.

> Using trends for the nEE devices has been described in the Trends Service section.

Niagara Workbench offers a tool to automatically discover trends which are exposed to the BACnet network on the nano EDGE ENGINE device. To do this, go to Trend Logs under the BACnetDevice in the Niagara station:

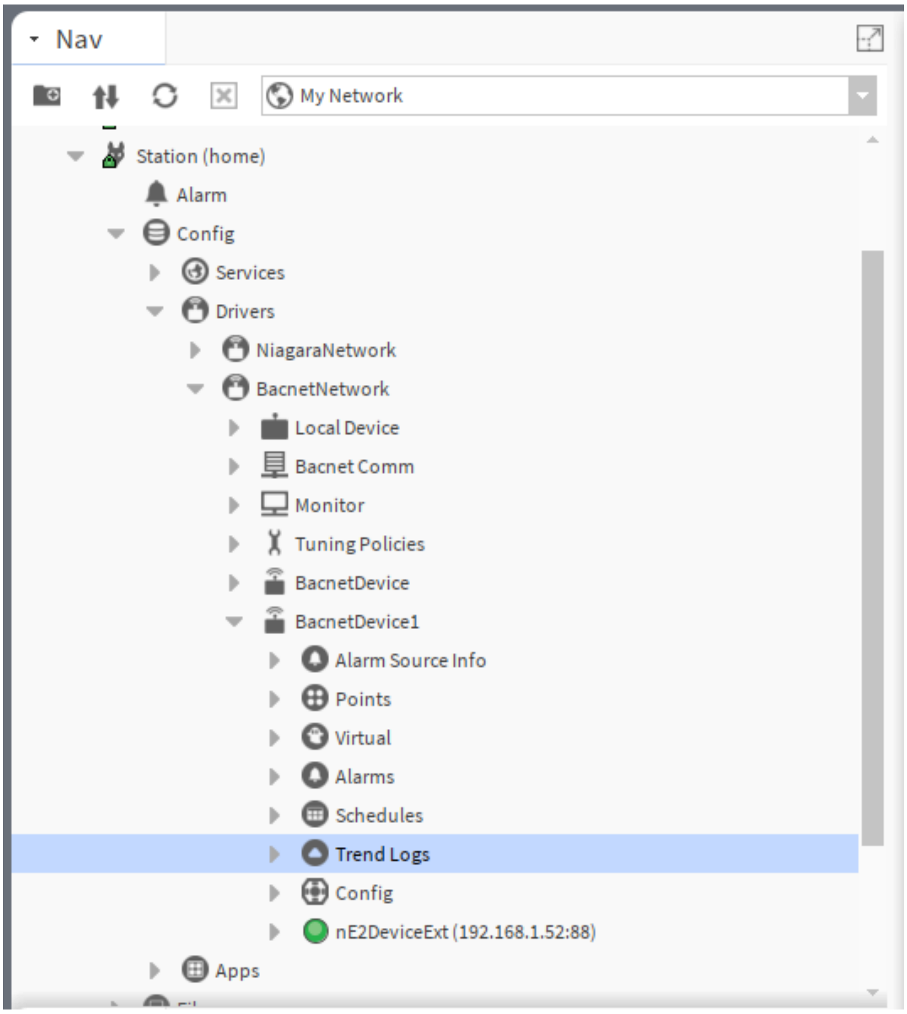


Figure 202. Trend Logs in the BACnetDevice

Double-click Trend Logs or use the context menu and choose AX Bacnet History Import Manager/Bacnet History Import Manager view.

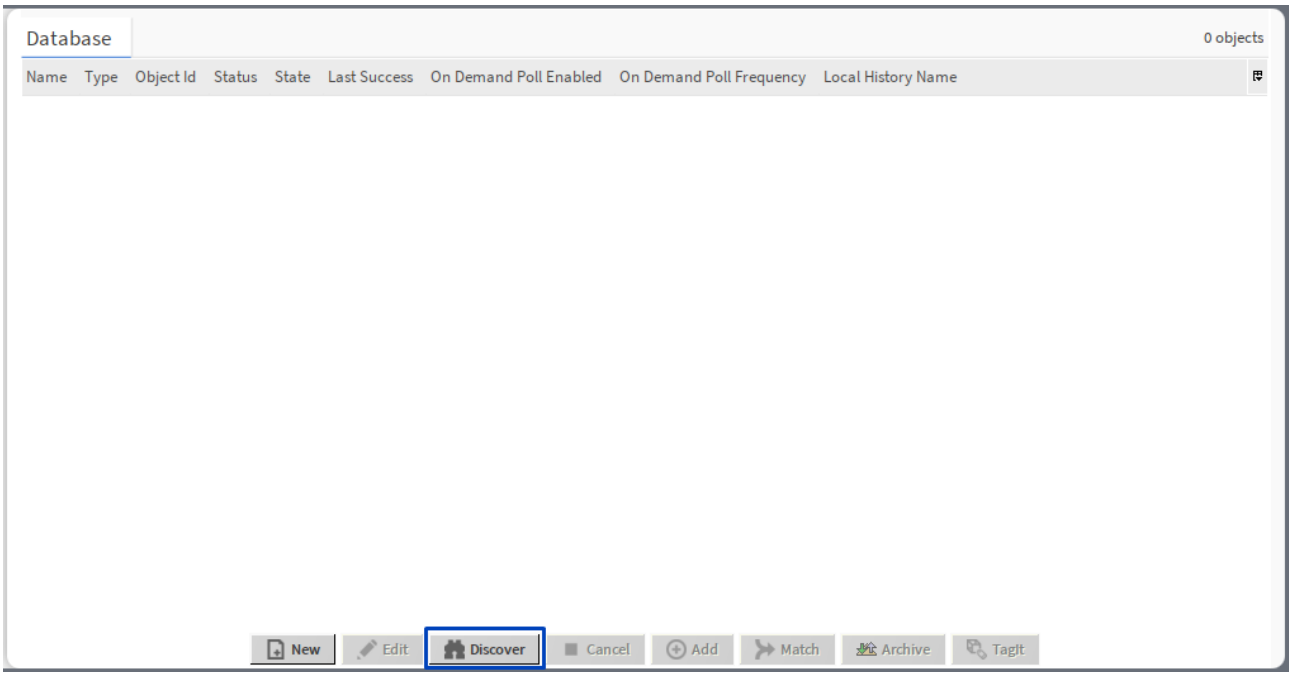
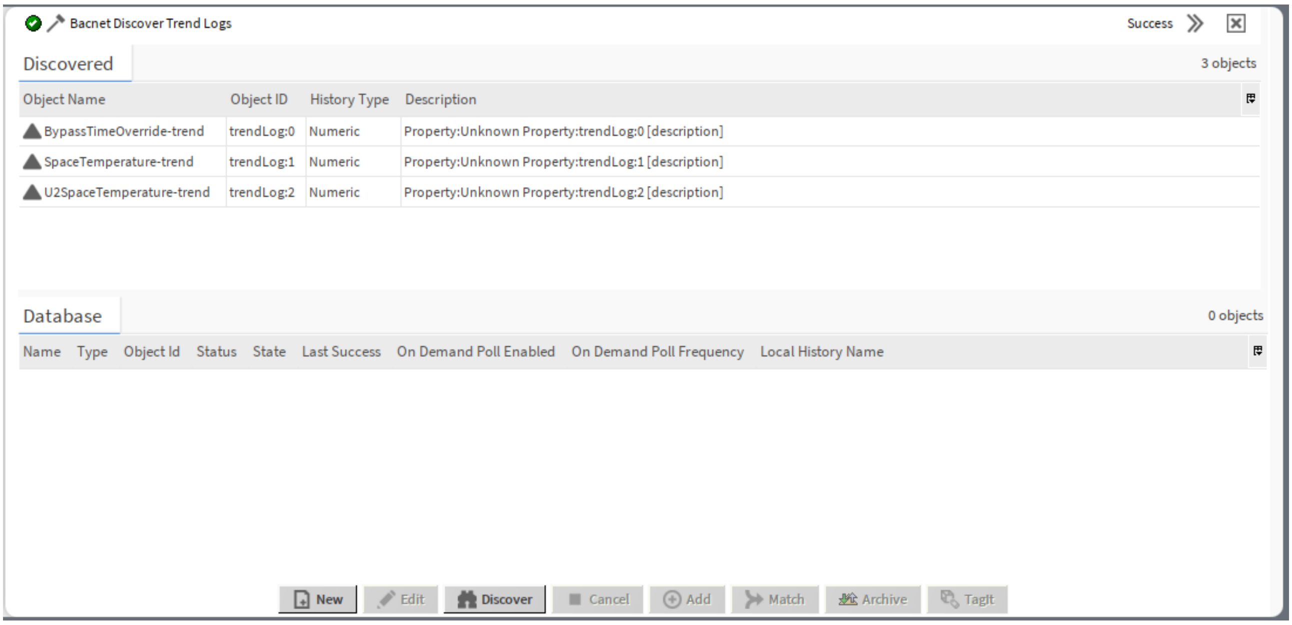


Figure 203. Trend Logs view

To automatically discover trends available on the nano EDGE ENGINE device, make sure that the device is properly connected to the BACnet network and communicating and use the Discover button.

The view will be split into to parts, Discovered and Database.



Discovered — 3 objects

Object Name	Object ID	History Type	Description
▲ BypassTimeOverride-trend	trendLog:0	Numeric	Property:Unknown Property:trendLog:0 [description]
▲ SpaceTemperature-trend	trendLog:1	Numeric	Property:Unknown Property:trendLog:1 [description]
▲ U2SpaceTemperature-trend	trendLog:2	Numeric	Property:Unknown Property:trendLog:2 [description]

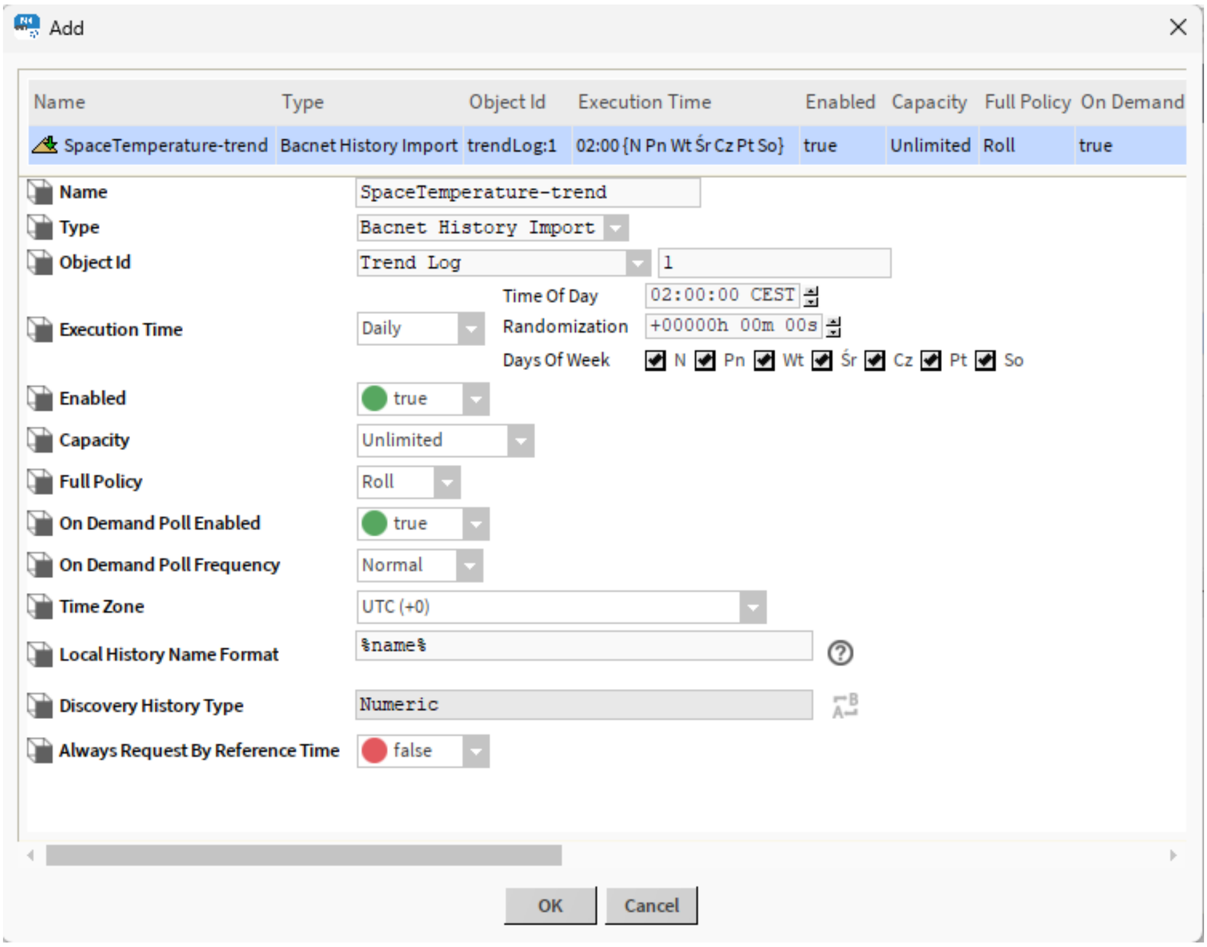
Database — 0 objects

Name	Type	Object Id	Status	State	Last Success	On Demand Poll Enabled	On Demand Poll Frequency	Local History Name

Figure 204. Discovered trends

To add trends to the Database, double-click on a trend, which automatically opens the Add pop-up, or mark trends in the Discovered section (click a single trend or use a Shift or Ctrl keys to mark more than one trend) and use the Add button or drag and drop marked trends in the Database section.

The Add pop-up allows to configure trends added to the Database:



Name	Type	Object Id	Execution Time	Enabled	Capacity	Full Policy	On Demand
SpaceTemperature-trend	Bacnet History Import	trendLog:1	02:00 {N Pn Wt Śr Cz Pt So}	true	Unlimited	Roll	true

Name	SpaceTemperature-trend
Type	Bacnet History Import
Object Id	Trend Log 1
Execution Time	Daily Time Of Day 02:00:00 CEST Randomization +00000h 00m 00s Days Of Week ☑ N ☑ Pn ☑ Wt ☑ Śr ☑ Cz ☑ Pt ☑ So
Enabled	● true
Capacity	Unlimited
Full Policy	Roll
On Demand Poll Enabled	● true
On Demand Poll Frequency	Normal
Time Zone	UTC (+0)
Local History Name Format	%name%
Discovery History Type	Numeric
Always Request By Reference Time	● false

Figure 205. Add trend pop-iup

The Add dialog window has the following columns:

- Name of the trend;
- Type;
- Object Id;
- Execution Time: allows to choose a synchronization method:
  - daily: requires additional specification of time, randomization, and days of week,
  - manual: can be linked to an action in logic,
  - interval: requires additional specification of interval, time, and days of week;
- Enabled status of the trend,
- Capacity: allows to set a number of records to be saved;
- Full Policy: allows to select an action when records limit (if set) is achieved);
- On Demand poll status.

it also allows to define additional parameters:

- On Demand Poll Enabled: allows to enable/disable an on-demand polling function;
- On Demand Poll Frequency: allows to set the on-demand polling frequency;
- Time Zone: allows to set a time zone, which will be automatically applied to records;
- Local History Name Format: allows to set a naming pattern for the history records;
- Discovery History Type: shows the format of history records;
- Always Request By Reference Time: allows to set if the trend records should always request against the reference time.

History Charts
Once the trends are correctly added to the Database, it is possible to further use them in the Niagara history tools, for example, AX History Chart Builder.

7 nanoWebUI™

The nanoWebUI™ is a modern HTML5 web user interface that presents live nano EDGE ENGINE-controller data from applications, Equipment components and Data Points. The view is automatically generated during controller programming.

The nanoWebUI™ is based on the Haystack HTTP API and creates a fully functional, responsive UI the moment a control logic is deployed. It provides a structured real-time visualization, enabling an effective time-saving monitoring and direct control of system data. The web interface is easily accessible from an HMI panel or a standard web browser on PC and mobile devices.

7.1 Initial Requirements

The nanoWebUI™ is easily accessible after entering the controller's IP number and port to a web browser, for example: 198.162.1.123:88. It will be automatically generated and displayed for a correctly connected (ETH cable) and operating controller. However, a few initial requirements have to be met for a correct functioning of the nanoWebUI™ interface.

These initial requirements involve making sure that all required libraries, services, and tags are installed and enabled on the device.

Initial Requirements

- Installed libraries:
    - Haystack,
    - Tagging,
    - Web,
    - Resource.Web;
- Enabled services in the Services container:
    - Tagging service,
    - Web service,
    - Haystack service.

Display Requirements

- Enabled auto-tags:
    - in the Equipment components (enabled by default),
    - in Data Points (required manual enabling);
- `web:expose` tag is set to true in all Equipment components and Data Points, which are supposed to be displayed on the nanoWebUI™ (true by default);
- Data Points are enabled and licensed.

7.2 Structure

The nanoWebUI™ structure is derived from the Resource.Web file, which is a part of the nano EDGE ENGINE OS and is visible in the Software Manager.

The presented structure is based on the Resource.Web V1.9.0.254 and is delivered with the nano EDGE ENGINE OS V1.9. It is the default layout of the nanoWebUI™.

7.2.1 Homepage

The nanoWebUI™ window is divided into two parts: left panel and main window. The left panel displays a search box, homepage reference, and all applications saved on the controller. All elements in the left panel are displayed with a relevant icon and are clickable and expandable, allowing to select an element to display in the main window.

In the main window, the nanoWebUI™'s homepage for a controller, where the application is constructed with a proper application structure (more details: Equipment), initially displays bars with all applications saved on the controller:

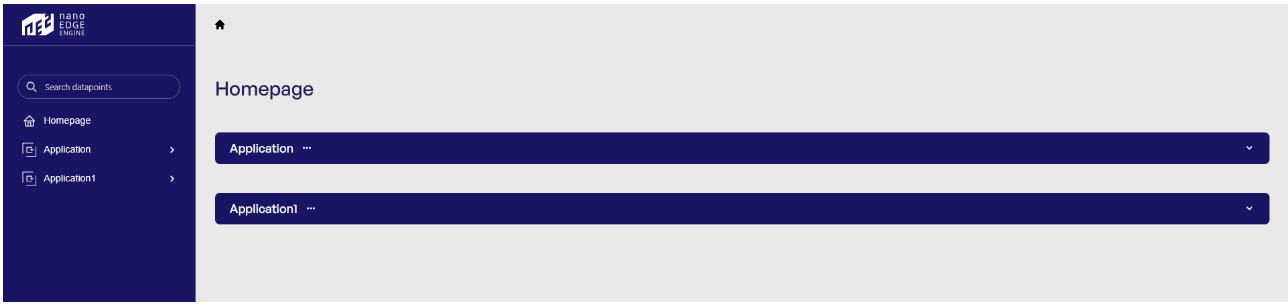


Figure 206. Homepage for the controller with two applications

Tip

To expand an element, click on three dots on a bar or an arrowhead in the left panel.

Note

If there are no tagged Data Points in the application(s) or the `web:expose` tag is configured to false (in either the Data Points or the Equipment component), the nanoWebUI™ will show an empty homepage:

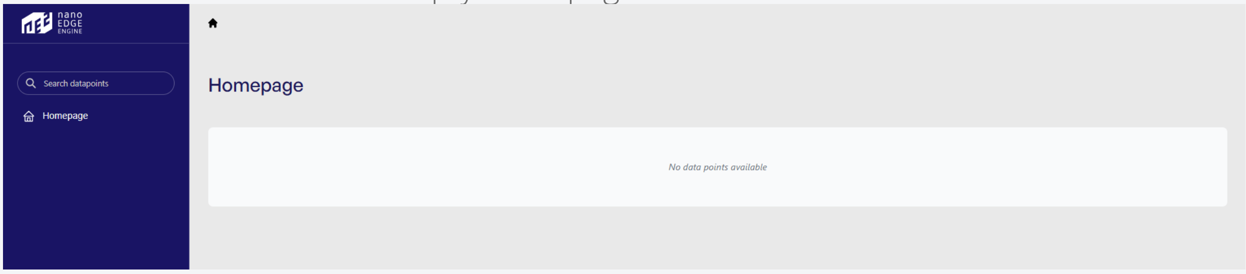


Figure 207. Empty homepage

7.2.2 Application

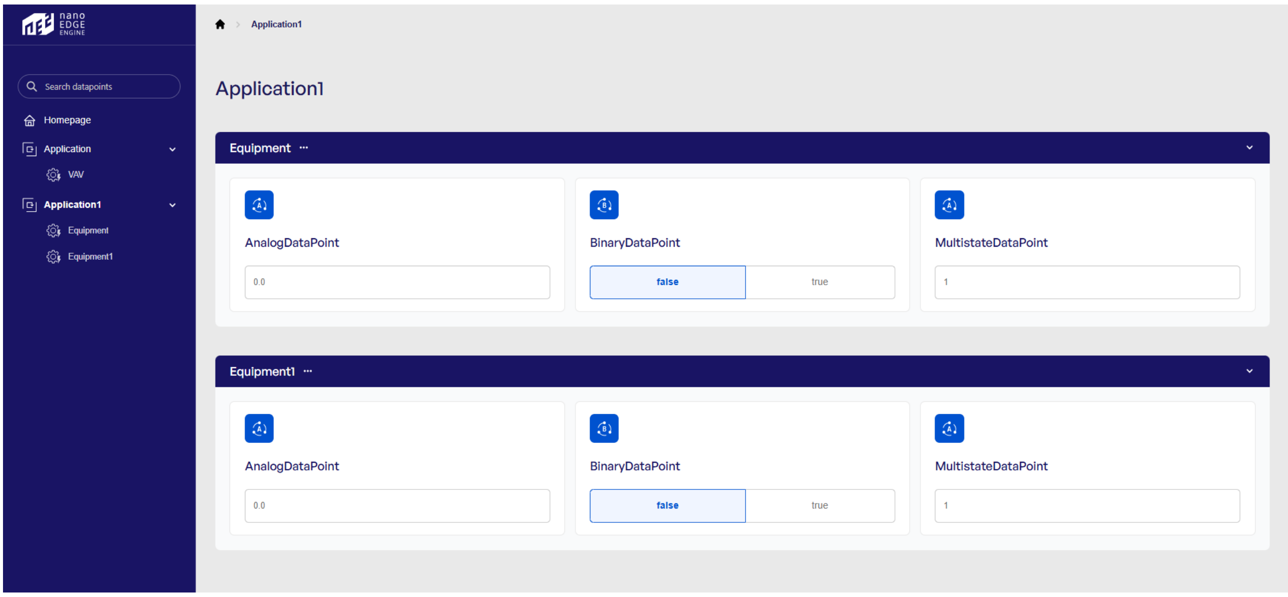


Figure 208. Application with two Equipment components and Data Points

Elements displayed in the nanoWebUI™

Elements displayed in the web server are strictly connected with tagging functionality. Applying tags in the nano EDGE ENGINE is based on a semantic approach that ensures consistent data structure which is easily usable by the nanoWebUI™ and by third-party systems. Tags are applied at the Equipment and Data Point levels, where Equipment serves as the logical container defining what is being controlled, and Data Points represent the measured or commanded values associated with that equipment. This structured model ensures that tagged data is immediately usable by platforms capable of communicating through standardized tag-based HTTP APIs, e.g., Haystack.

Note: Tags can be only applied to the Equipment components and Data Points. Other component types are not supported.

It is therefore recommended (however, not mandatory) to use the following structure when creating applications:

- Applications container
    - Application component
        - Equipment component
            - Data Point(s)
            - other components
        - Equipment component
            - Data Point(s)
            - other components

Note

The Application and Equipment components are displayed provided that at least one licensed Data Point with auto-tags enabled and the `web:expose` tag set to true is placed under either one of them.

The left panel contains the list of applications saved on the controller and Equipment components.

The main view shows Data Points organized by Equipment in the application. Data Points are presented as widgets, which presents the current Data Point's value and can be editable under certain conditions.

7.2.3 Web Order

For the nanoWebUI™ display clarity and accuracy of representation of the user's requirements, it is possible to adjust the web order of Data Points display. The mechanism is dependent on the tool in use:

> Changing Data Points web order:
> - in iC Tool: Web Tags view
> - in iC Workbench with nE2 Link module: Web Tags manager

7.2.4 Widgets

There are four types of widgets applied to display Data Points values.

Selecting Widget

A specific type of widget can be selected in the `web:widgettype` tag.

> Note
> Not every type of widget can be used for all types of Data Points, however, each can be selected. For example, the BinaryDataPoint can only use the basic widget, therefore, setting the step widget for the BinaryDataPoint will simply have no effect on the web server display.
> In the further description, please note which type of widget is applicable to which Data Point.

Types of Widgets

Basic

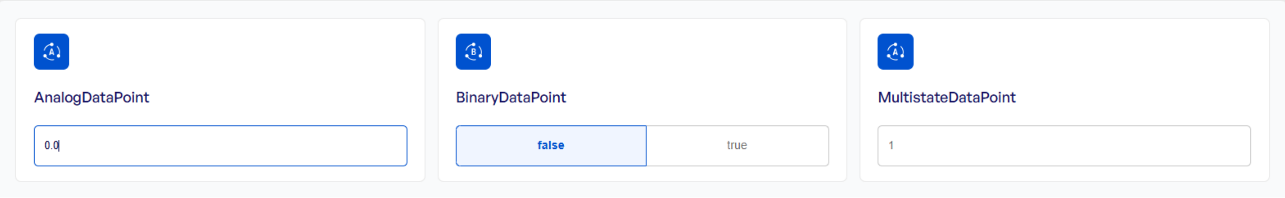


Figure 209. Basic widgets for all types of Data Points

The basic widget can be applied to **all types** of Data Points.

It shows the Data Point's current value and allows to edit it (send new value to the controller) if the Data Point is set to the Output or Value mode and the `hs:writable` tag is active.

Step

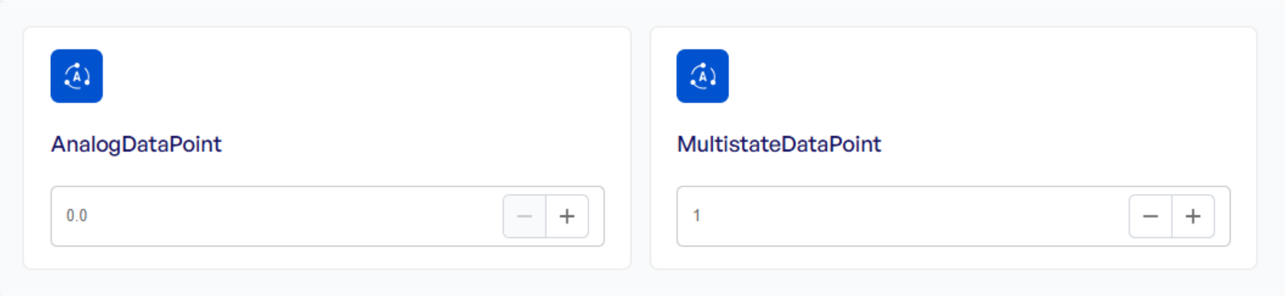


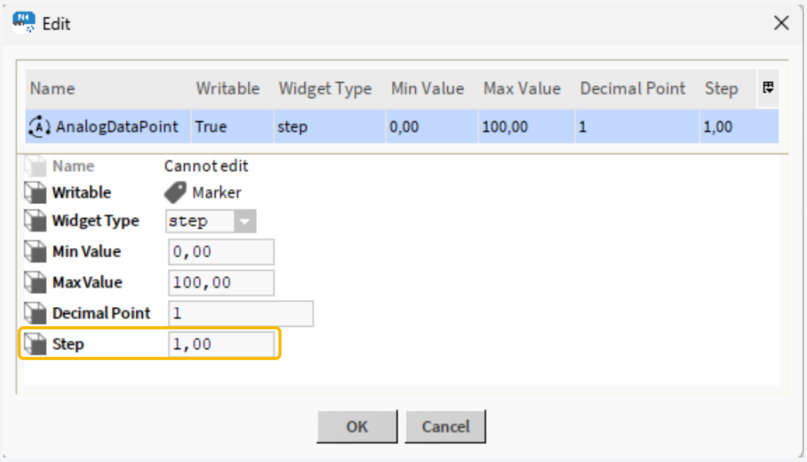
Figure 210. Step widgets

The step widget can be applied to the AnalogDataPoint and MultistateDataPoint.

It shows the Data Point's current value and allows to edit it (send new value to the controller) by the value of a step.

The widget is editable if the Data Point is set to the Output or Value mode and the `hs:writable` tag is active.

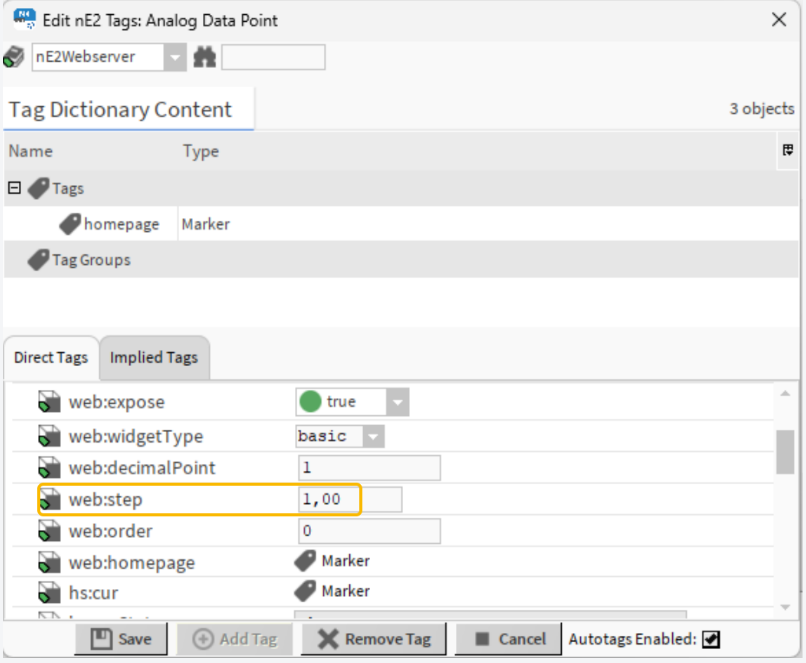
In the case of the AnalogDataPoint (**does not apply** to the MultistateDataPoint), it is possible to set the step value in the Tag Manager or Web Tags Manager using the Edit button:



Name	Writable	Widget Type	Min Value	Max Value	Decimal Point	Step
AnalogDataPoint	True	step	0,00	100,00	1	1,00

Name	Cannot edit
Writable	Marker
Widget Type	step
Min Value	0,00
Max Value	100,00
Decimal Point	1
Step	1,00

Figure 211. Setting a step value for AnalogDataPoint in the Web Tags Manager



Name	Type
Tags	
homepage	Marker
Tag Groups	

web:expose	true
web:widgetType	basic
web:decimalPoint	1
web:step	1,00
web:order	0
web:homepage	Marker
hs:cur	Marker

Figure 212. Setting a step value for AnalogDataPoint in the Tag Manager

Effectively, there are no minimum or maximum limitations as to the step value.

Slider

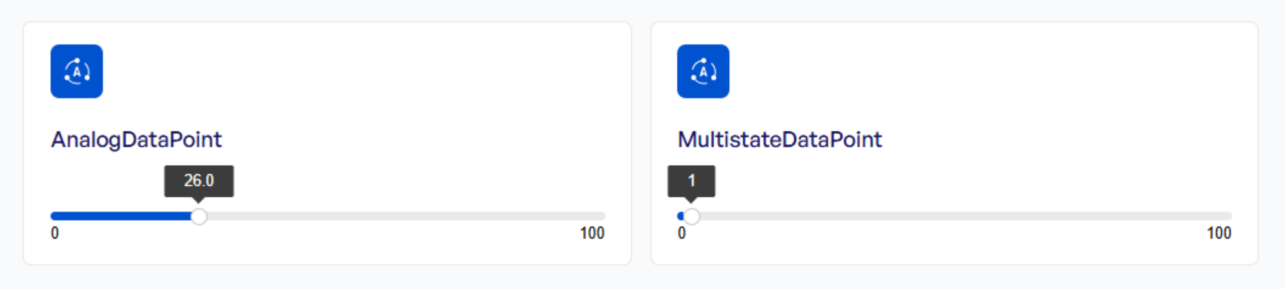


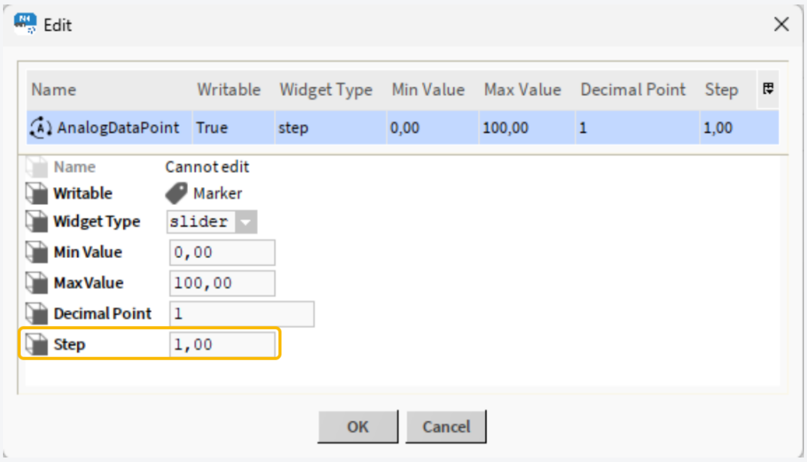
Figure 213. Slider widgets

The slider widget can be applied to the AnalogDataPoint and MultistateDataPoint.

It shows the Data Point's current value and allows to edit it (send new value to the controller) by setting a value on a slider.

The widget is editable if the Data Point is set to the Output or Value mode and the `hs:writable` tag is active.

In the case of the AnalogDataPoint (**does not apply** to the MultistateDataPoint), it is possible to set the step value in the Tag Manager or Web Tags Manager using the Edit button:

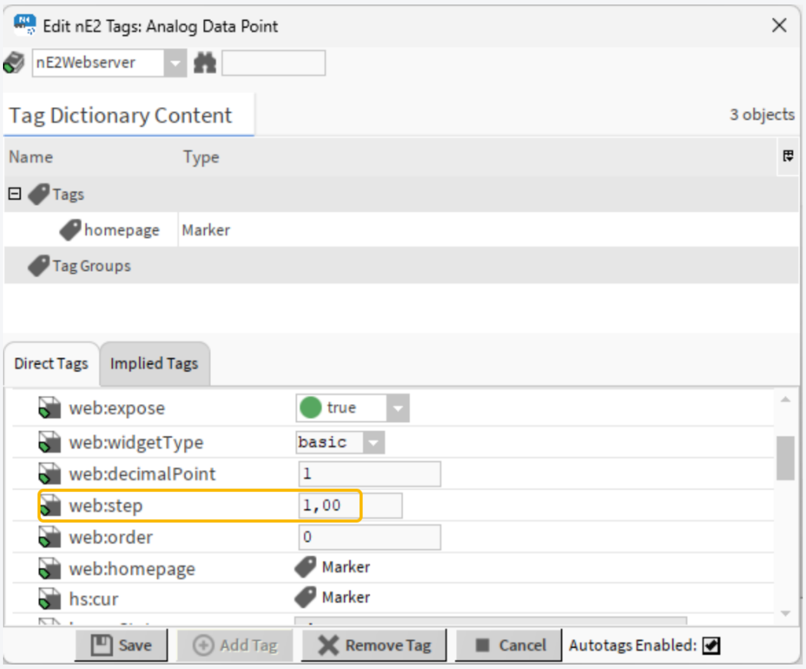


Edit

Name	Writable	Widget Type	Min Value	Max Value	Decimal Point	Step
AnalogDataPoint	True	step	0,00	100,00	1	1,00

Name	Cannot edit
Writable	Marker
Widget Type	slider
Min Value	0,00
Max Value	100,00
Decimal Point	1
Step	1,00

Figure 214. Setting a step value for AnalogDataPoint in the Web Tags Manager



Edit nE2 Tags: Analog Data Point

nE2Webserver

Tag Dictionary Content — 3 objects

Name	Type
Tags	
homepage	Marker
Tag Groups	

Direct Tags | Implied Tags

web:expose	true
web:widgetType	basic
web:decimalPoint	1
web:step	1,00
web:order	0
web:homepage	Marker
hs:cur	Marker

Figure 215. Setting a step value for AnalogDataPoint in the Tag Manager

Effectively, there are no minimum or maximum limitations as to the step value.

Fill

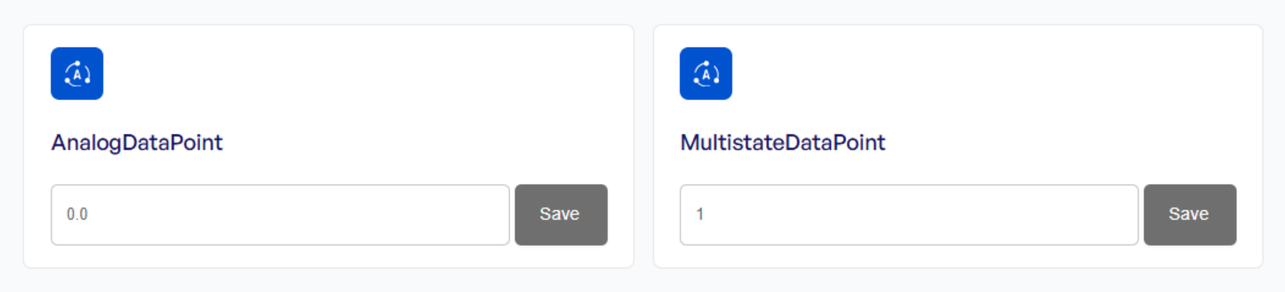
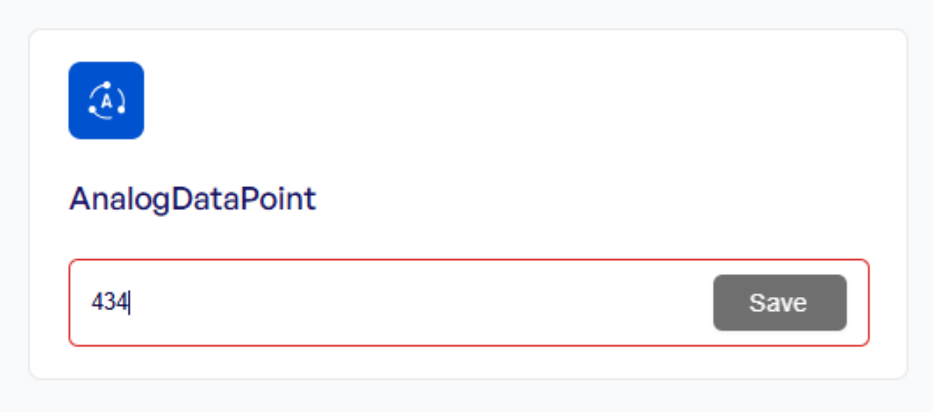


Figure 216. Fill widgets

The fill widget can be applied to the AnalogDataPoint and MultistateDataPoint.

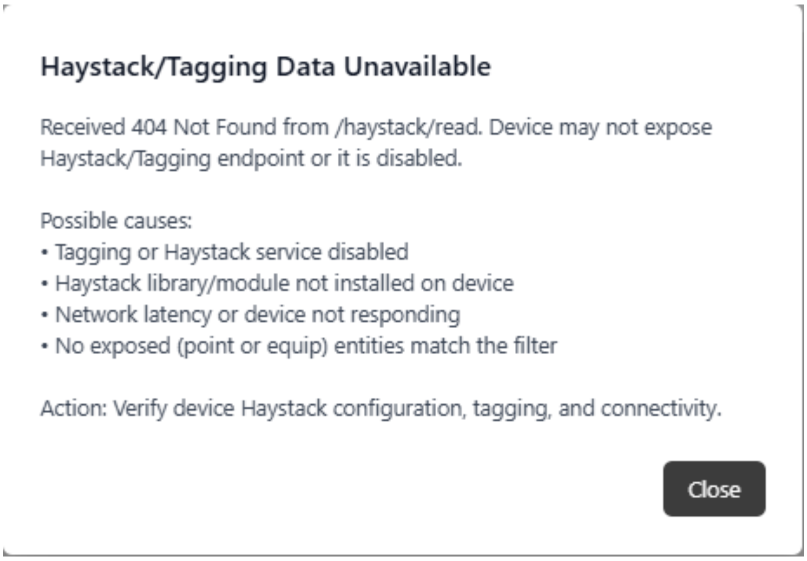
It shows the Data Point's current value and allows to edit it (send new value to the controller) by typing a specific value and pressing the Save button.

The widget is editable if the Data Point is set to the Output or Value mode and the `hs:writable` tag is active.

> Note
> If a widget is editable, in the editing mode the widget's frame is blue. If an entered value cannot be accepted by the Data Point (for example, it exceeds the `hs:maxVal` value), the frame will be marked red and the value will not be sent to the controller:
>
> 
>
>
> *Figure 217. Error value*

7.2.5 Unavailability Notification

If any of the initial requirements for the nanoWebUI™ functioning is not fulfilled (see the top of this section: any of the required libraries is not installed or any of the required services is disabled), the nanoWebUI™ will display the following notification:



Haystack/Tagging Data Unavailable

Received 404 Not Found from /haystack/read. Device may not expose Haystack/Tagging endpoint or it is disabled.

Possible causes:
- Tagging or Haystack service disabled
- Haystack library/module not installed on device
- Network latency or device not responding
- No exposed (point or equip) entities match the filter

Action: Verify device Haystack configuration, tagging, and connectivity.

Figure 218. Unavailability note

For a correct display of the web server, please make sure that all initial and display requirements are fulfilled.

7.2.6 Other Notifications

In case of any other malfunctioning of the nanoWebUI™, a proper notification pops up with a description of an issue and steps suggested to undertake.

Notification	Message
Language selector	Select language
Invalid Haystack response	Haystack operation returned invalid status. Steps: 1. Make sure that the device is in the running state. 2. Make sure that Library.Haystack is installed on the device. 3. Make sure that Haystack service is enabled and running on the device.
Could not parse Haystack response	Could not parse response for Haystack operation. Steps: 1. Make sure that you're using compatible version of Library.Haystack.
No elements to display	Haystack read operation returned no elements to display. Steps: 1. Make sure that Tagging service is enabled 2. Make sure that Data Points are configured with tags.
Haystack timeout	Haystack operation timed out after ''. Steps: 1. Make sure that the device is in the running state. 2. Make sure that the device is connected to the network and network is properly configured . 3. Check your network performance for high latency and/or packet loss. 4. Make sure the device is not overloaded with application logic.

Notification	Message
Empty Haystack Response	Haystack operation returned empty response. Steps: 1. Check your network configuration to make sure nothing interferes with http responses.
Invalid pointWrite Response	Haystack pointWrite operation returned invalid response.
Can't fetch device information	Haystack about operation returned empty zinc grid.
Haystack subscription failed	Haystack watchSub operation returned invalid response.
Invalid WatchId	Device likely lost connection for an extended period of time.

8 Supported Workbench Views

Workbench offers a variety of views, which enable efficient management of devices, logic, and networks. With nE2 Link, the following Workbench views are supported for nano EDGE ENGINE devices:

8.1 Wire Sheet

(supporting the nano EDGE ENGINE context menu):

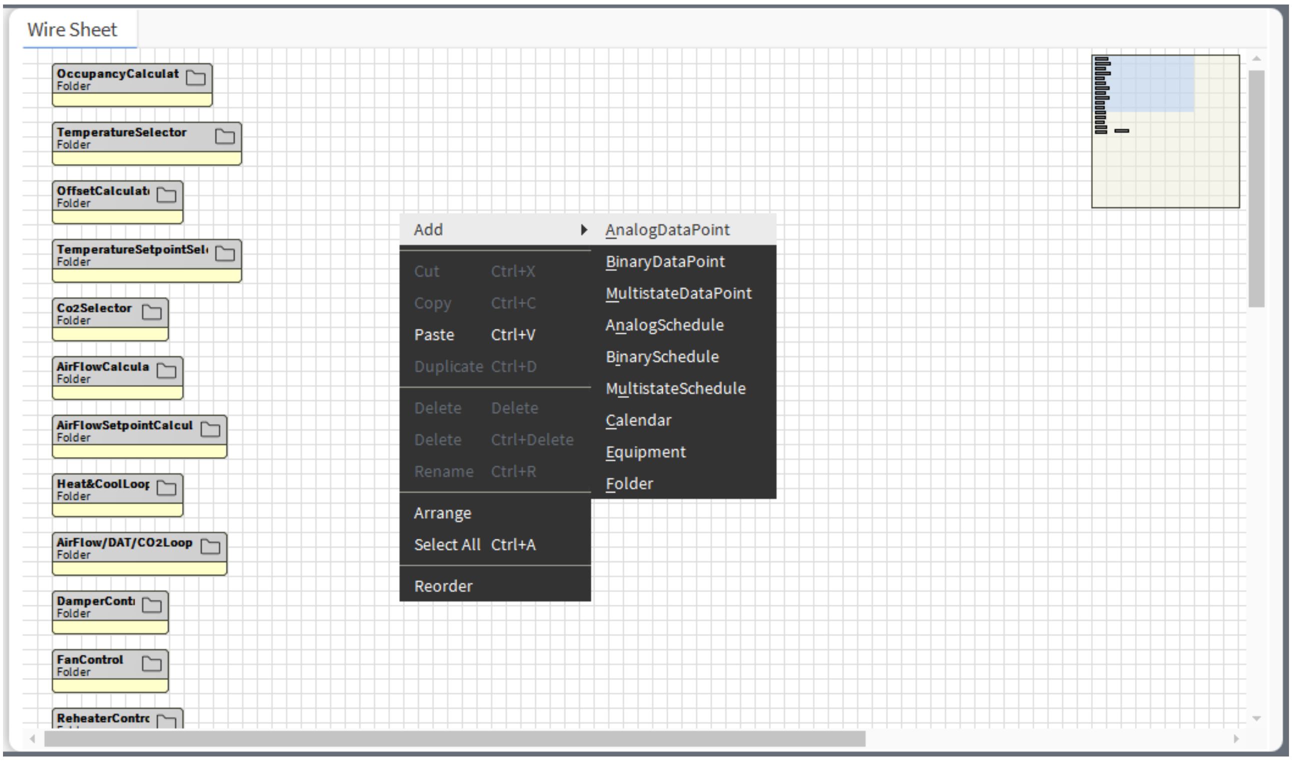
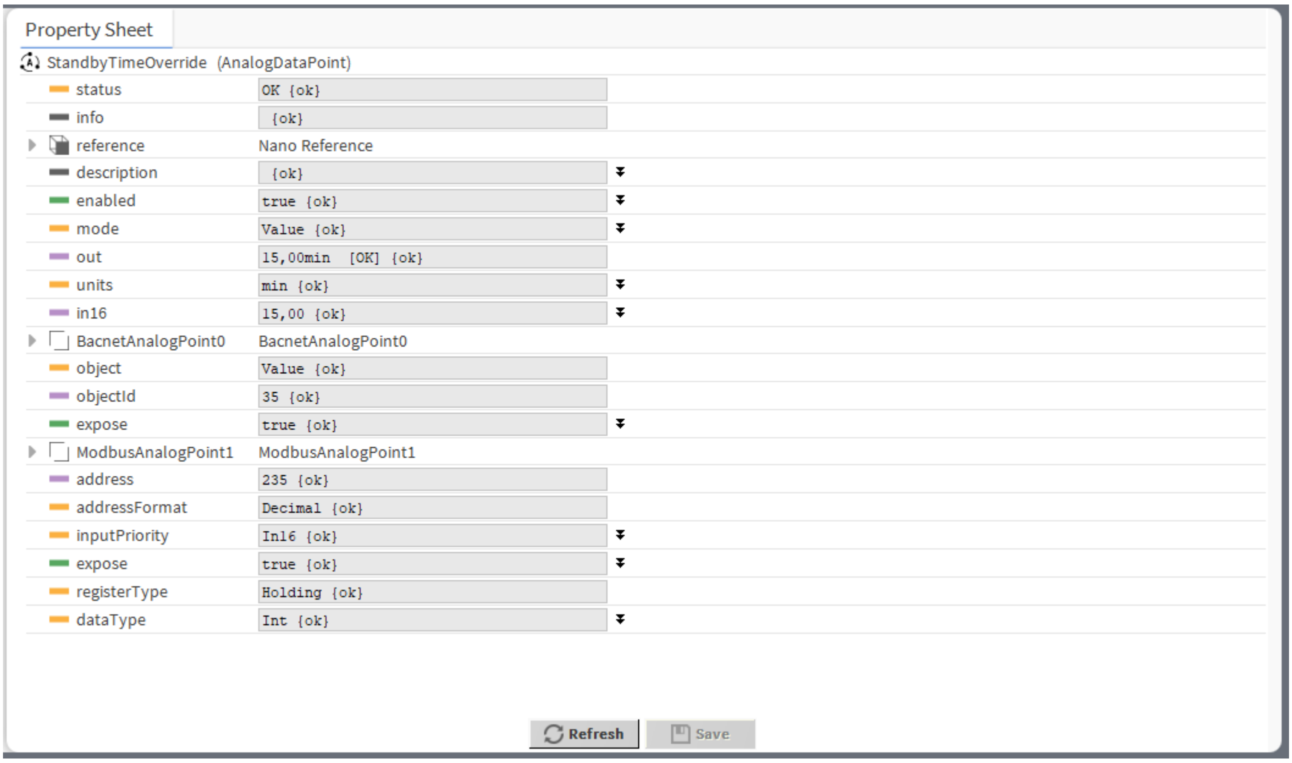


Figure 219. Wire sheet with the nano EDGE ENGINE context menu

8.2 AX Property Sheet

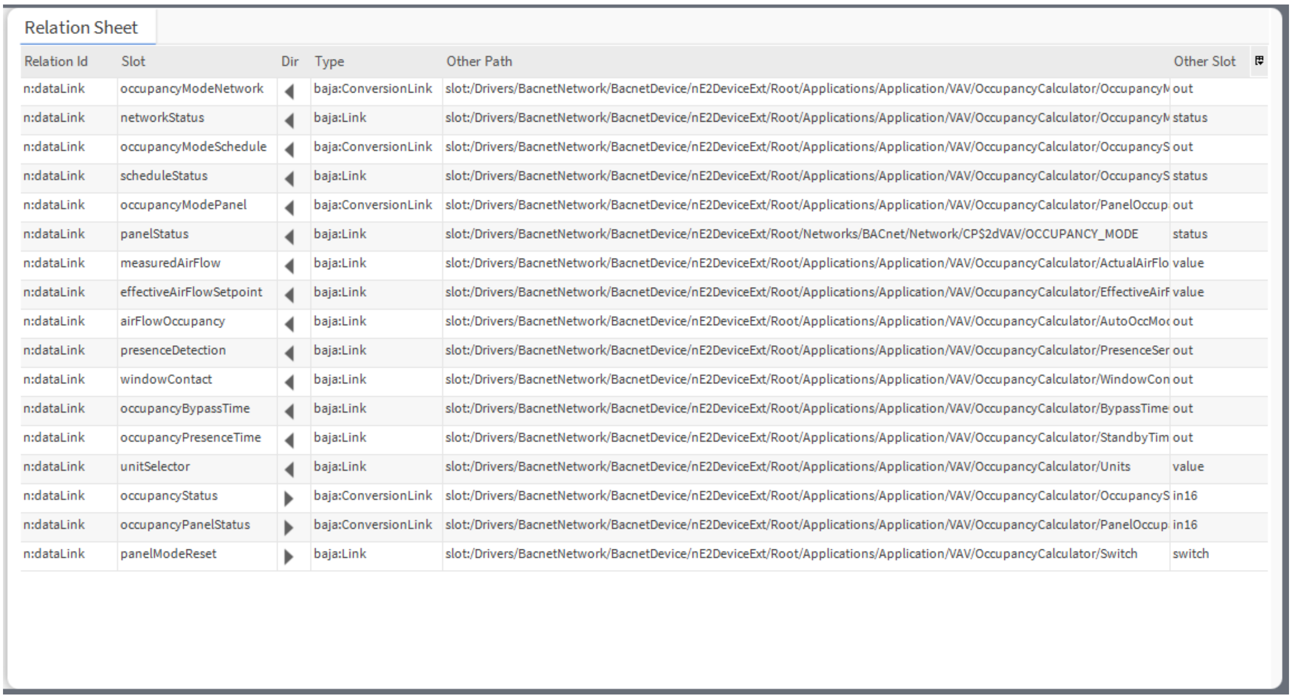


StandbyTimeOverride (AnalogDataPoint)

status	OK {ok}
info	{ok}
reference	Nano Reference
description	{ok}
enabled	true {ok}
mode	Value {ok}
out	15,00min [OK] {ok}
units	min {ok}
in16	15,00 {ok}
BacnetAnalogPoint0	BacnetAnalogPoint0
object	Value {ok}
objectId	35 {ok}
expose	true {ok}
ModbusAnalogPoint1	ModbusAnalogPoint1
address	235 {ok}
addressFormat	Decimal {ok}
inputPriority	In16 {ok}
expose	true {ok}
registerType	Holding {ok}
dataType	Int {ok}

Figure 220. Property sheet view for the AnalogDataPoint

8.3 Relation Sheet



Relation Id	Slot	Dir	Type	Other Path	Other Slot
n:dataLink	occupancyModeNetwork	◀	baja:ConversionLink	slot:/Drivers/BacnetNetwork/BacnetDevice/nE2DeviceExt/Root/Applications/Application/VAV/OccupancyCalculator/OccupancyM	out
n:dataLink	networkStatus	◀	baja:Link	slot:/Drivers/BacnetNetwork/BacnetDevice/nE2DeviceExt/Root/Applications/Application/VAV/OccupancyCalculator/OccupancyM	status
n:dataLink	occupancyModeSchedule	◀	baja:ConversionLink	slot:/Drivers/BacnetNetwork/BacnetDevice/nE2DeviceExt/Root/Applications/Application/VAV/OccupancyCalculator/OccupancyS	out
n:dataLink	scheduleStatus	◀	baja:Link	slot:/Drivers/BacnetNetwork/BacnetDevice/nE2DeviceExt/Root/Applications/Application/VAV/OccupancyCalculator/OccupancyS	status
n:dataLink	occupancyModePanel	◀	baja:ConversionLink	slot:/Drivers/BacnetNetwork/BacnetDevice/nE2DeviceExt/Root/Applications/Application/VAV/OccupancyCalculator/PanelOccup	out
n:dataLink	panelStatus	◀	baja:Link	slot:/Drivers/BacnetNetwork/BacnetDevice/nE2DeviceExt/Root/Networks/BACnet/Network/CP$2dVAV/OCCUPANCY_MODE	status
n:dataLink	measuredAirFlow	◀	baja:Link	slot:/Drivers/BacnetNetwork/BacnetDevice/nE2DeviceExt/Root/Applications/Application/VAV/OccupancyCalculator/ActualAirFlo	value
n:dataLink	effectiveAirFlowSetpoint	◀	baja:Link	slot:/Drivers/BacnetNetwork/BacnetDevice/nE2DeviceExt/Root/Applications/Application/VAV/OccupancyCalculator/EffectiveAirF	value
n:dataLink	airFlowOccupancy	◀	baja:Link	slot:/Drivers/BacnetNetwork/BacnetDevice/nE2DeviceExt/Root/Applications/Application/VAV/OccupancyCalculator/AutoOccMo	out
n:dataLink	presenceDetection	◀	baja:Link	slot:/Drivers/BacnetNetwork/BacnetDevice/nE2DeviceExt/Root/Applications/Application/VAV/OccupancyCalculator/PresenceSer	out
n:dataLink	windowContact	◀	baja:Link	slot:/Drivers/BacnetNetwork/BacnetDevice/nE2DeviceExt/Root/Applications/Application/VAV/OccupancyCalculator/WindowCon	out
n:dataLink	occupancyBypassTime	◀	baja:Link	slot:/Drivers/BacnetNetwork/BacnetDevice/nE2DeviceExt/Root/Applications/Application/VAV/OccupancyCalculator/BypassTime	out
n:dataLink	occupancyPresenceTime	◀	baja:Link	slot:/Drivers/BacnetNetwork/BacnetDevice/nE2DeviceExt/Root/Applications/Application/VAV/OccupancyCalculator/StandbyTim	out
n:dataLink	unitSelector	◀	baja:Link	slot:/Drivers/BacnetNetwork/BacnetDevice/nE2DeviceExt/Root/Applications/Application/VAV/OccupancyCalculator/Units	value
n:dataLink	occupancyStatus	▶	baja:ConversionLink	slot:/Drivers/BacnetNetwork/BacnetDevice/nE2DeviceExt/Root/Applications/Application/VAV/OccupancyCalculator/OccupancyS	in16
n:dataLink	occupancyPanelStatus	▶	baja:ConversionLink	slot:/Drivers/BacnetNetwork/BacnetDevice/nE2DeviceExt/Root/Applications/Application/VAV/OccupancyCalculator/PanelOccup	in16
n:dataLink	panelModeReset	▶	baja:Link	slot:/Drivers/BacnetNetwork/BacnetDevice/nE2DeviceExt/Root/Applications/Application/VAV/OccupancyCalculator/Switch	switch

Figure 221. Relation sheet listing all incoming and outgoing links of the component

Note
As of the nE2 Link V1.1, the web views are not supported.

9 Workbench with nano EDGE ENGINE

Being as advanced a tool as the Workbench is, the nE2 Link primarily supports functions native to the nano EDGE ENGINE, and so the Workbench options may occasionally differ from the ones native to the Niagara environment. This section describes new functions deriving from the nano EDGE ENGINE and existing native Niagara functionalities that were confirmed to work by iSMA CONTROLLI.

9.1 Basic Context Menu Options

Context menu options normally differ between components, for example, in terms of available views or actions. However, there is a set of basic options, which are supported regardless of the type of component the context menu is invoked on.

- Views: allows to display component's data in one of the defined standard views (Wire Sheet, Property Sheet), or in other views if they are available for a particular component (more on Views below);
- Actions: shows a list of actions that may be invoked for the given component;
- Cut: (shortcut Ctrl+X) removes a selected component from an original location and allows it to paste in a new location (applicable only within the Applications container);
- Copy: (shortcut Ctrl+C) remembers and copies a selected component along with all its properties, settings, and links information;
- Paste: (shortcut Ctrl+V) pastes a previously remembered component into a specific place and possibly recreates internal links;
- Duplicate: (shortcut Ctrl+D) duplicates the selected components in the same location;

Copying

Copying of components includes all links created between the copied components, except for external links (links to components placed under a different parent/superior component, for example, reference links between network points in the Networks container and Data Points in the Applications container). If only one component is copied, its links will be neglected in the process.

Cutting

Cutting of components allows to move them to another location while removing them from the original one. The mechanism allows to move one or more components at a time, however, they must be moved within the same container (the option is available only within the Applications container).

Pasting

Copied or cut components may be pasted into a chosen place in the Wire Sheet or Property Sheet views or in the tree, by indicating the pasting place and pressing Ctrl+V, or using the Paste option from the context menu. Pasting copied components creates new components along with their children components and slot settings effective at the moment of copying. The pasted slot values are sourced from the copied component. Pasting cut components moves them from the original location to the new one along with their values and links.

Duplicating

Duplication is a method is used for a quick replication of a selected component along with its values. The slot values for the duplicated component are rewritten from the source component.

- Delete: (shortcut Del) removes the selected components;

- Link Mark: allows to define the component from which a link will be led (used along with the Link From option) or to which a link will be led (used along with the Link To option);
- Link From: allows to create a link from the marked component to the selected component;
- Link To: allows to create a link from the selected component to the marked component;
- Reorder: allows to reorder components in the tree;

Reordering

Reordering helps organize the contents of the application (the option is available only within the Applications container). It is invoked from the context menu on the tree, wire sheet, or property sheet and is executed in a pop-up:

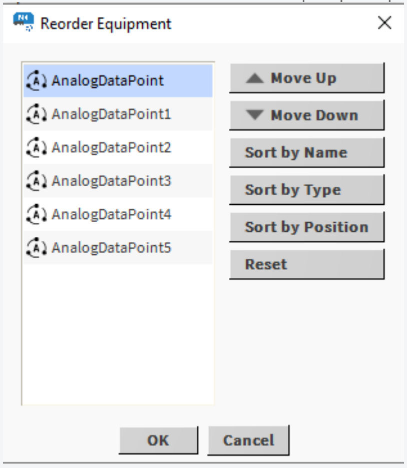


Figure 222. Reordering pop-up always lists children components of the one the option was invoked on

- Rename: allows to change a component's name (can be applied to one or more components);
- Export: allows to initialize the Workbench export to a file;
- Refresh Tree Node: allows to update the tree;
- Go Into: allows to simplify the view in the nav section by making the item, on which the the action is invoked, the top level of visibility.

Go Into

Using the nE2 Link in the station requires adding the device in the BACnet network, which results in a complex structure of the tree when unfolding elements to reach the nE2DeviceExt. Go Into function allows to simplify the view and display only the contents of the extension:

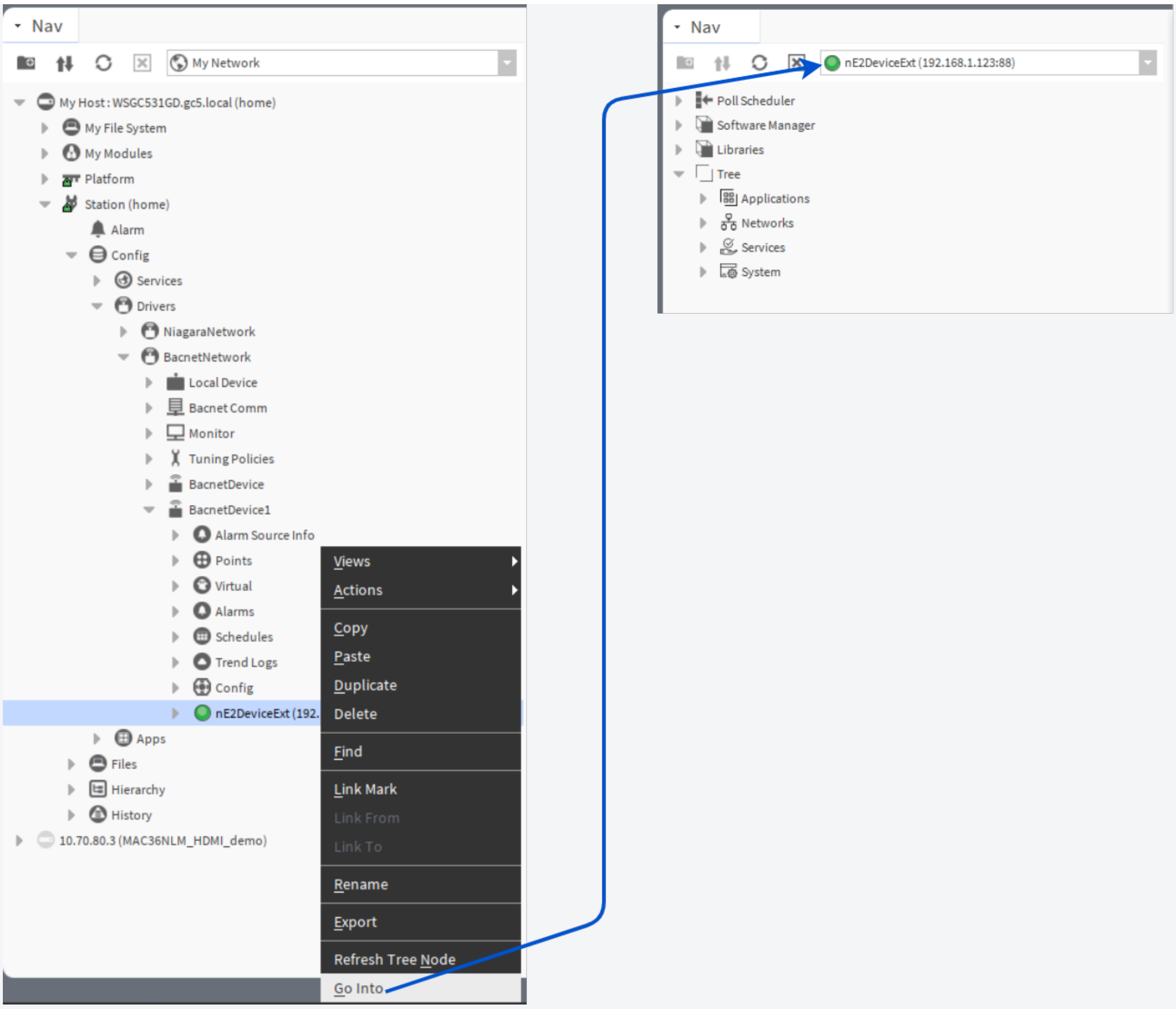


Figure 223. Using the Go Into option to simplify the tree structure

9.2 Specific Context Menu Options

- Add: the function offers an expedited way of adding certain type of components to the application.

Add

Using the Add option from the context menu is available only in the Application container and refers to the following components:

- AnalogDataPoint
- BinaryDataPoint
- MultistateDataPoint
- AnalogSchedule
- BinarySchedule
- MultistateSchedule
- Calendar
- Equipment
- Folder

10 Logs

The Logs component runs an adjustable register of all events happening in the connected device. Such register becomes crucial when troubleshooting; it enables checking records, comparing recent ones with historical feeds, and, if needed, sharing them with the iSMA CONTROLLI Support Team for further diagnostics.

The records in the Logs register are grouped by their genre; they are categorized depending on the area they originated–firmware, core, BACnet, app, etc. Each group may have its individual log level defining priorities of data to be recorded, according to the users needs. These priorities are differentiated from Debug (each event happening is registered) to Critical (only few events that result in the system error are registered). Available log levels are the following:

- Debug: logs used for debugging purposes (e.g., information about sent iFNet requests or elements saved to storage),
- Normal: logs informing about typical actions in the device, which are characteristic for a device’s normal operation (e.g., information about free managed memory, added BACnet objects, steps of creating or restoring backup, etc.),
- Warning: logs informing about a wrong configuration or issued action not working as expected (e.g., issuing the action while the parent service is disabled or initializing the component more than once) – keep in mind that in a properly configured and properly working application these logs may not appear at all,
- Important: logs containing information crucial for device identification (e.g., iFnet IP address and port, firmware version, hardware version, core version), device readiness (e.g., system started, loaded services, BACnet/Modbus ports, loaded libraries, etc.), and important actions (e.g., detected DIP switch reset, factory reset, successful restore, etc.),
- Error: logs informing about errors in applications (i.e., bad link, wrongly configured IO, unable to delete folder, errors in creating or restoring backup, etc.),
- Critical: logs informing about issues that were not handled properly and need to be fixed as soon as possible, (e.g., entering the emergency mode, occurrence of hard fault of the device, stack overflow, or watchdog reset, etc.).

> If the log level is set, for example, to Important, saved logs will include records from the Important level and the less detailed levels, in this case, the Error and Critical levels

The Logs register is written to a file and saved on the SD card in the device. If needed, it may be copied from the SD card and shared with the iSMA CONTROLLI Support Team for troubleshooting.

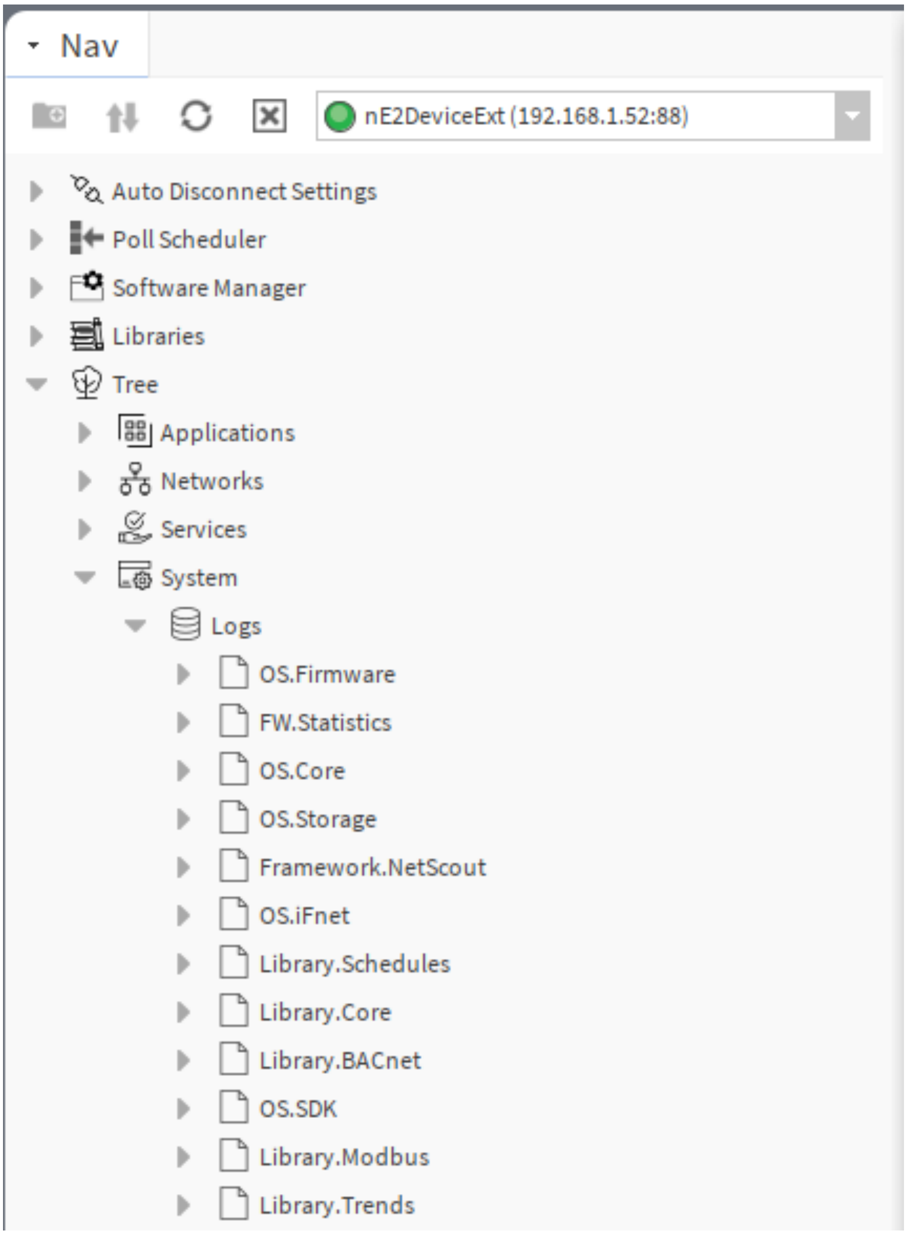
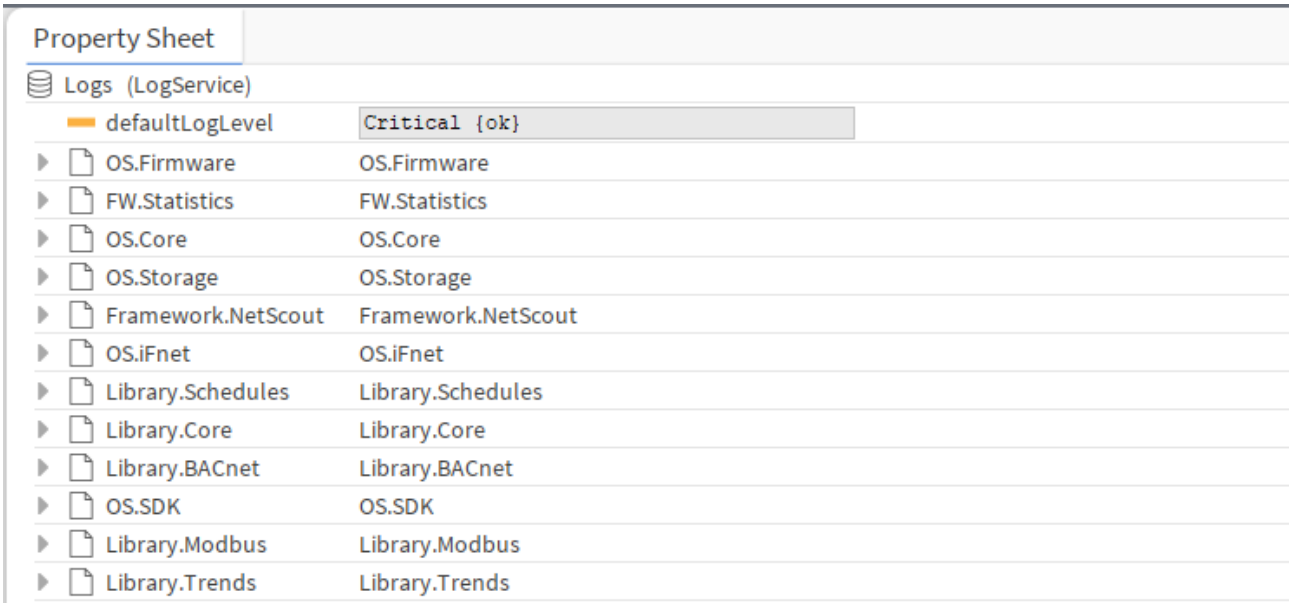


Figure 224. Logs

The Logs component has one slot:

- **Default Log Level:** informs about the importance of logs to be registered from the available log levels. Log levels are listed by level of details (Debug - most detailed, Critical - least detailed).

Note
The default log level is Critical.
Please note that logs have a built-in automatic mechanism, which switches the log level back to the Critical level after 24 hours for any change.



Property Sheet

Logs (LogService)

defaultLogLevel	Critical {ok}
OS.Firmware	OS.Firmware
FW.Statistics	FW.Statistics
OS.Core	OS.Core
OS.Storage	OS.Storage
Framework.NetScout	Framework.NetScout
OS.iFnet	OS.iFnet
Library.Schedules	Library.Schedules
Library.Core	Library.Core
Library.BACnet	Library.BACnet
OS.SDK	OS.SDK
Library.Modbus	Library.Modbus
Library.Trends	Library.Trends

Figure 225. Logs' slot defaultLogLevel

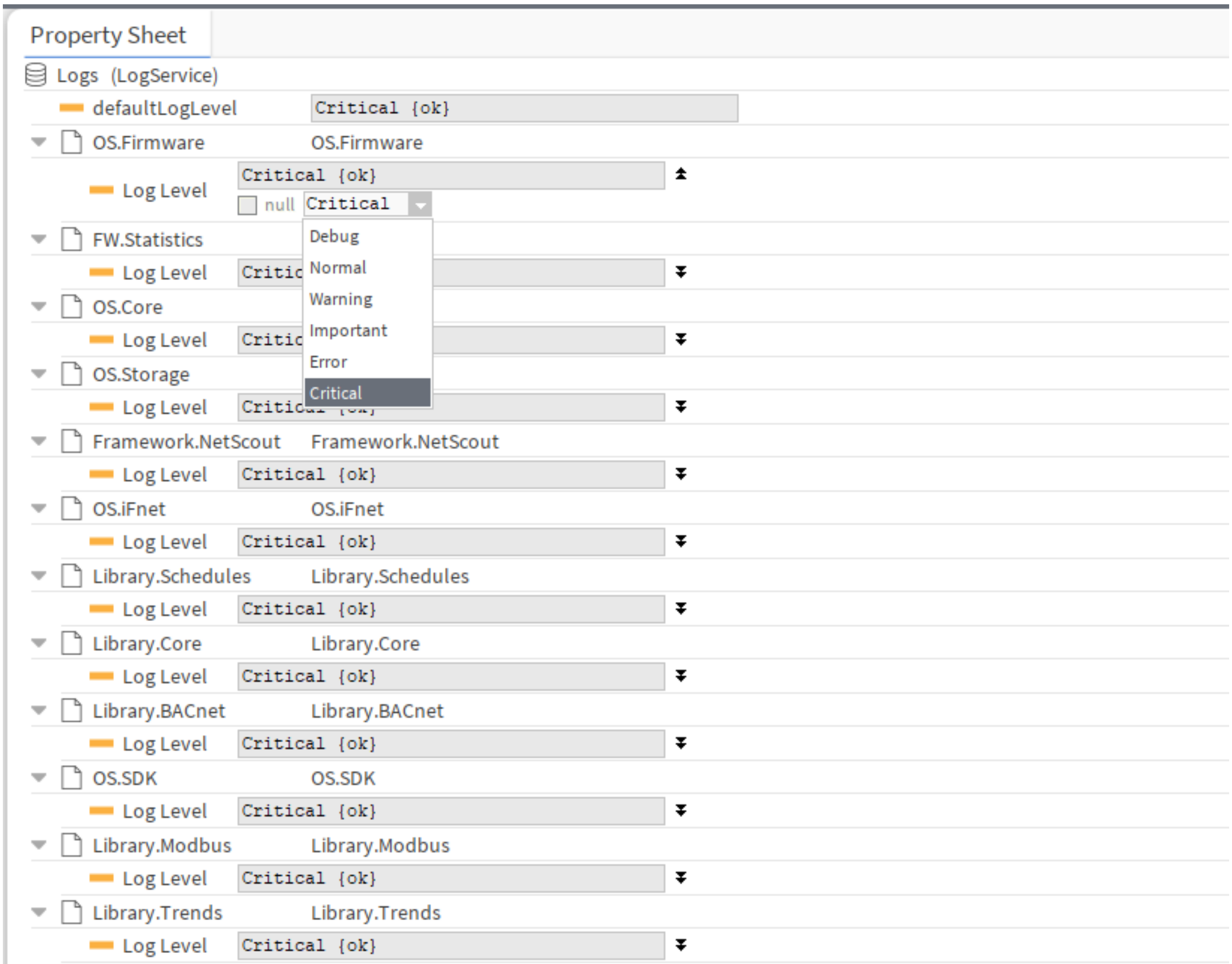
Note
Logs register information about network configuration (IP address, default gateway, mask) providing a convenient way to retrieve such data if lost.

10.1 Extensions

The Logs component has also available the component's extensions for each of groups defined for register:

- OS.Firmware,
- FW.Statistics,
- OS.Storage,
- OS.Core (system elements),
- OS.iFnet,
- Library.Schedules,
- Library.Core (library including Data Points, folders, etc.),
- OS.SDK,
- Library.BACnet,
- Library.Modbus,
- Library.IO,
- Framework.NetScout,
- Framework.ConfigurationData,
- Library.Web,
- Library.Tagging,
- Library.Haystack.

Each of the above has a logLevel slot, which allows to individually set the importance of logs to be registered.



Property Sheet

Logs (LogService)

defaultLogLevel	Critical {ok}
OS.Firmware	OS.Firmware
Log Level	Critical {ok} □ null Critical
FW.Statistics	
Log Level	Critic
OS.Core	
Log Level	Critic
OS.Storage	
Log Level	Critical {ok}
Framework.NetScout	Framework.NetScout
Log Level	Critical {ok}
OS.iFnet	OS.iFnet
Log Level	Critical {ok}
Library.Schedules	Library.Schedules
Log Level	Critical {ok}
Library.Core	Library.Core
Log Level	Critical {ok}
Library.BACnet	Library.BACnet
Log Level	Critical {ok}
OS.SDK	OS.SDK
Log Level	Critical {ok}
Library.Modbus	Library.Modbus
Log Level	Critical {ok}
Library.Trends	Library.Trends
Log Level	Critical {ok}

Figure 226. logLevel slots

10.2 Accessing Logs

Logs are accessible from the Nav Tree under the System → Logs.

To save logs, right-click on Logs and select Actions → Dump Log from the context menu.

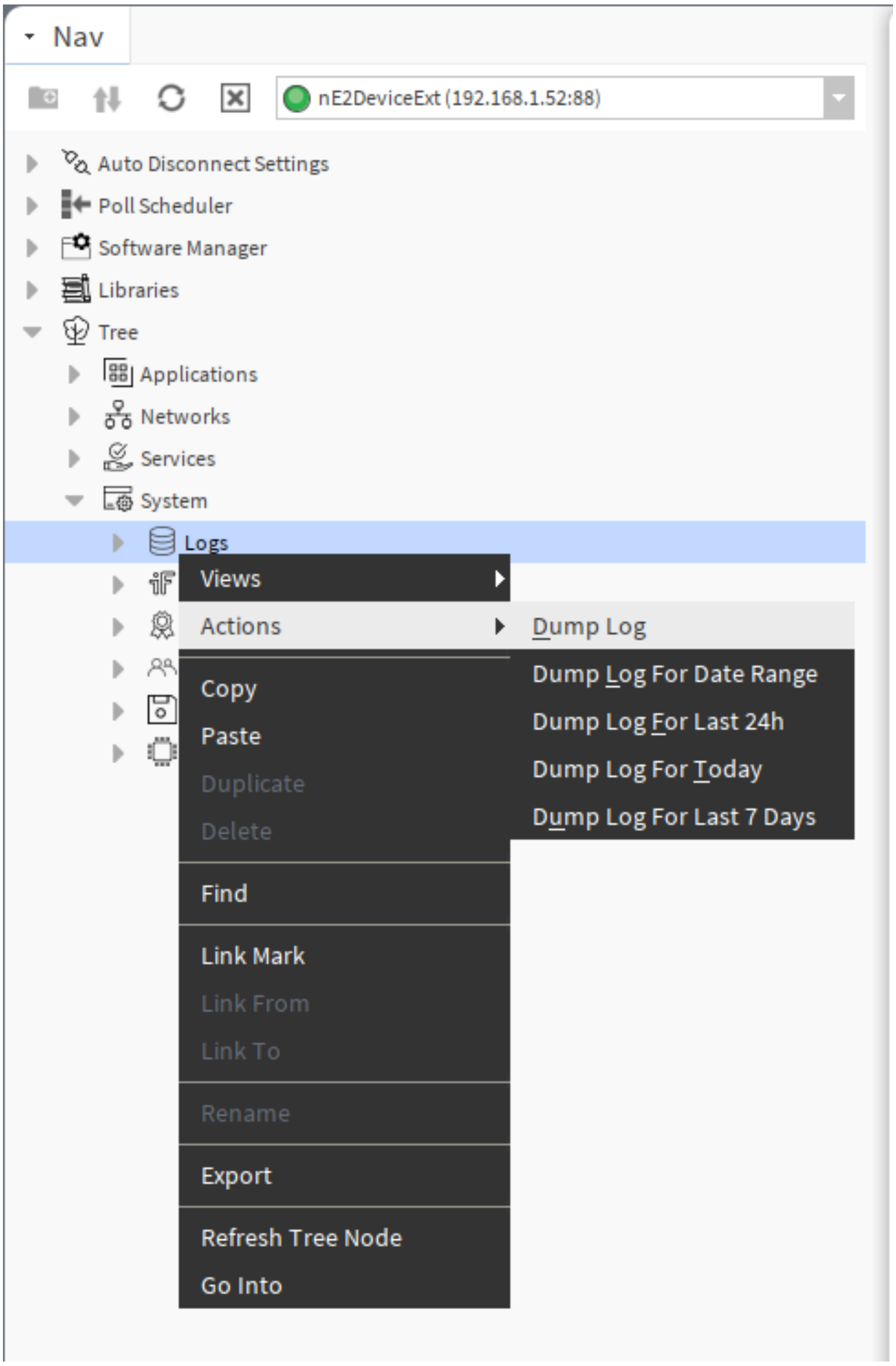
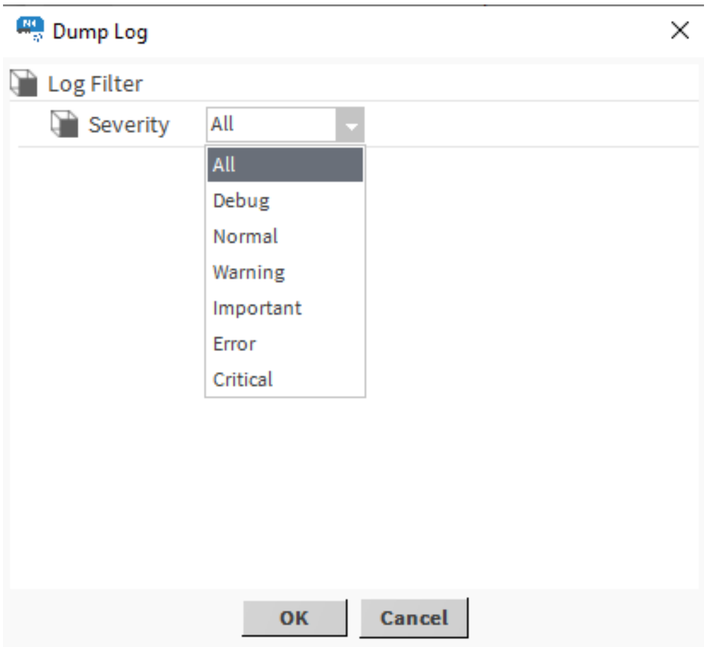


Figure 227. Accessing logs

The following actions are available:

- Dump Log: saves all registered logs to a .txt file,



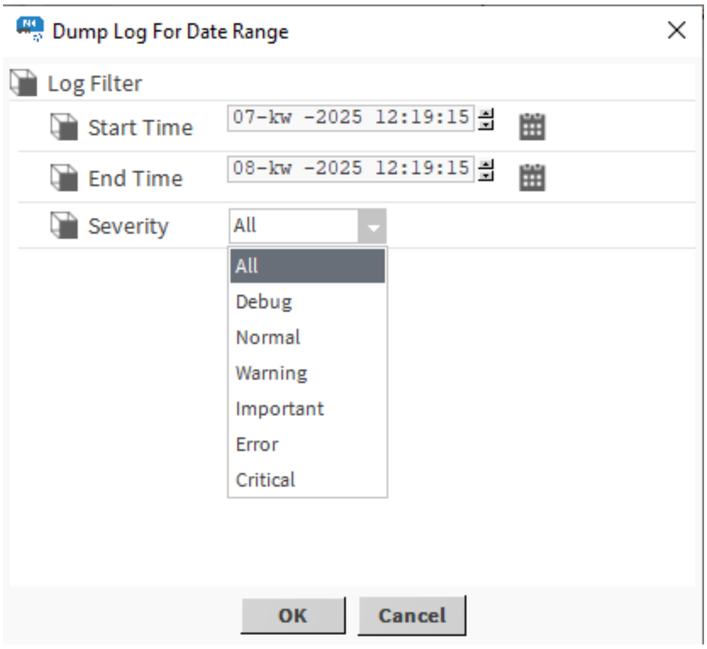
Dump Log

Log Filter

Severity: All

Figure 228. Dump Log action

- **Dump Log for Date Range:** saves logs from a selected date range to a .txt file,



Dump Log For Date Range

Log Filter

Start Time: 07-kw -2025 12:19:15

End Time: 08-kw -2025 12:19:15

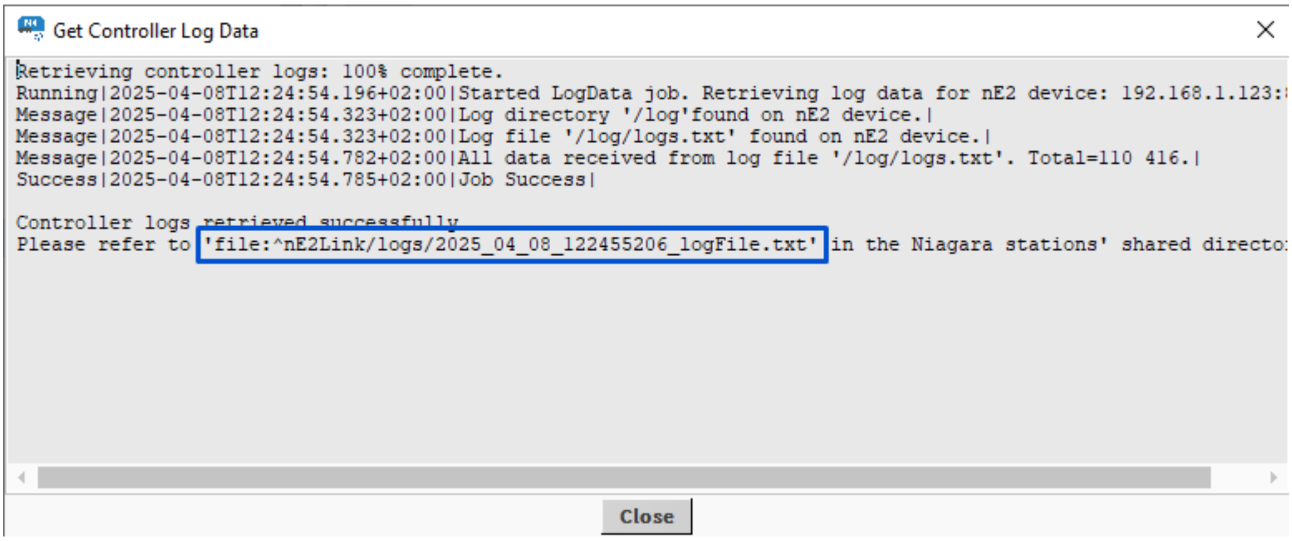
Severity: All

Figure 229. Dump Log for Date Range action

- **Dump Log for Last 24 Hours:** saves logs from lat 24 hours to a .txt file,
- **Dump Log for Today:** saves logs from today to a .txt file,
- **Dump Log for Last 7 Days:** saves logs from last 7 days to a .txt file.

After selecting an action, a Dump Log window pop-up, where it is possible to specify the level of log filtering.

In the final step, a confirmation window is displayed with the status of the task and indication of the location of the saved log file.



```
Retrieving controller logs: 100% complete.
Running|2025-04-08T12:24:54.196+02:00|Started LogData job. Retrieving log data for nE2 device: 192.168.1.123:
Message|2025-04-08T12:24:54.323+02:00|Log directory '/log'found on nE2 device.|
Message|2025-04-08T12:24:54.323+02:00|Log file '/log/logs.txt' found on nE2 device.|
Message|2025-04-08T12:24:54.782+02:00|All data received from log file '/log/logs.txt'. Total=110 416.|
Success|2025-04-08T12:24:54.785+02:00|Job Success|

Controller logs retrieved successfully
Please refer to 'file:^nE2Link/logs/2025_04_08_122455206_logFile.txt' in the Niagara stations' shared directo
```

Figure 230. Saving logs confirmation pop-up

The file can then be accessed in the tree under Files:

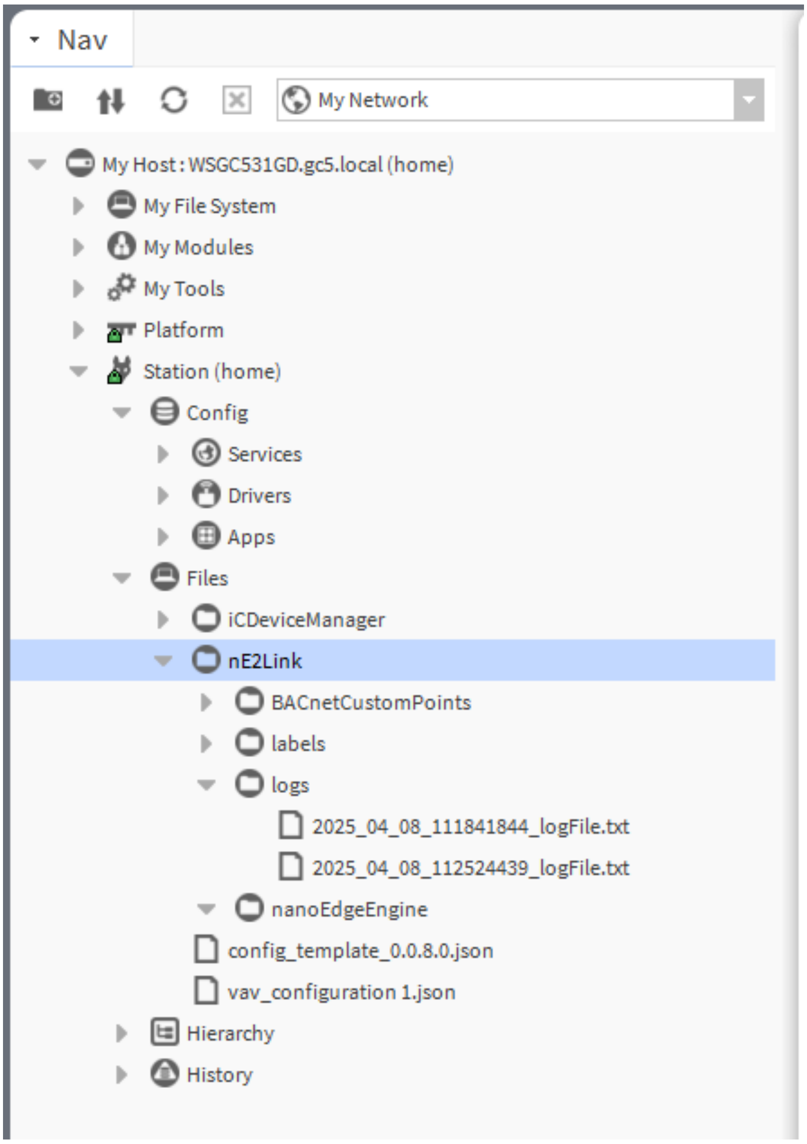
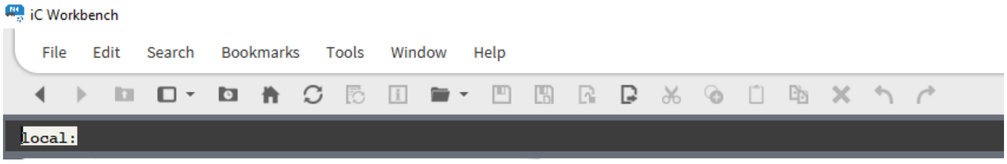


Figure 231. Logs file location

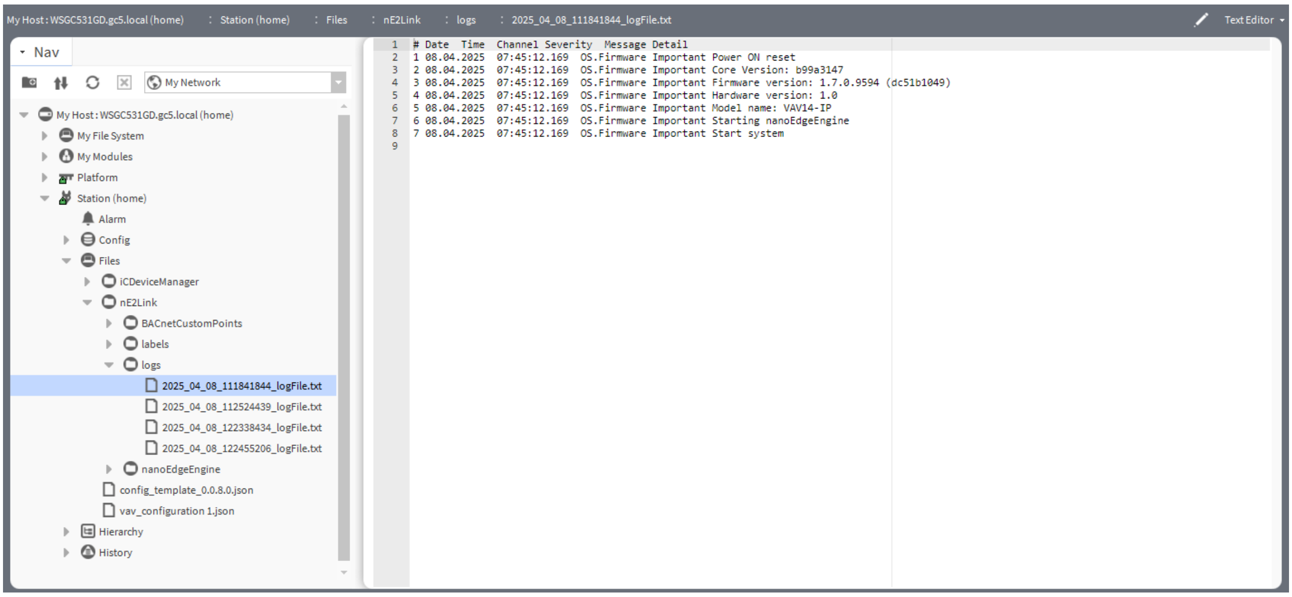
To access the logs, either double-click on the .txt file in the tree or copy the file's path from the confirmation pop-up and paste it to the editable path in Workbench.



```
local:
```

Figure 232. The Workbench bar with files' path becomes editable upon clicking

The file will be opened in the Workbench text editor view.


My Host : WSGC531GD.gc5.local (home) : Station (home) : Files : nE2Link : logs : 2025_04_08_111841844_logFile.txt
Text Editor
Nav
My Network
My Host : WSGC531GD.gc5.local (home)
My File System
My Modules
Platform
Station (home)
Alarm
Config
Files
iCDeviceManager
nE2Link
BACnetCustomPoints
labels
logs
2025_04_08_111841844_logFile.txt
2025_04_08_112524439_logFile.txt
2025_04_08_122338434_logFile.txt
2025_04_08_122455206_logFile.txt
nanoEdgeEngine
config_template_0.0.8.0.json
vav_configuration 1.json
Hierarchy
History

```
# Date  Time  Channel Severity  Message Detail
1 08.04.2025  07:45:12.169  OS.Firmware Important Power ON reset
2 08.04.2025  07:45:12.169  OS.Firmware Important Core Version: b99a3147
3 08.04.2025  07:45:12.169  OS.Firmware Important Firmware version: 1.7.0.9594 (dc51b1049)
4 08.04.2025  07:45:12.169  OS.Firmware Important Hardware version: 1.0
5 08.04.2025  07:45:12.169  OS.Firmware Important Model name: VAV14-IP
6 08.04.2025  07:45:12.169  OS.Firmware Important Starting nanoEdgeEngine
7 08.04.2025  07:45:12.169  OS.Firmware Important Start system
```

Figure 233. Logs presented in text editor