iSMACONTROLLI

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

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
			• 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

Date	Manual Rev.	nE2 Link Module Ver.	Description
			◦ 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.

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`, and `nE2link-wb.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.

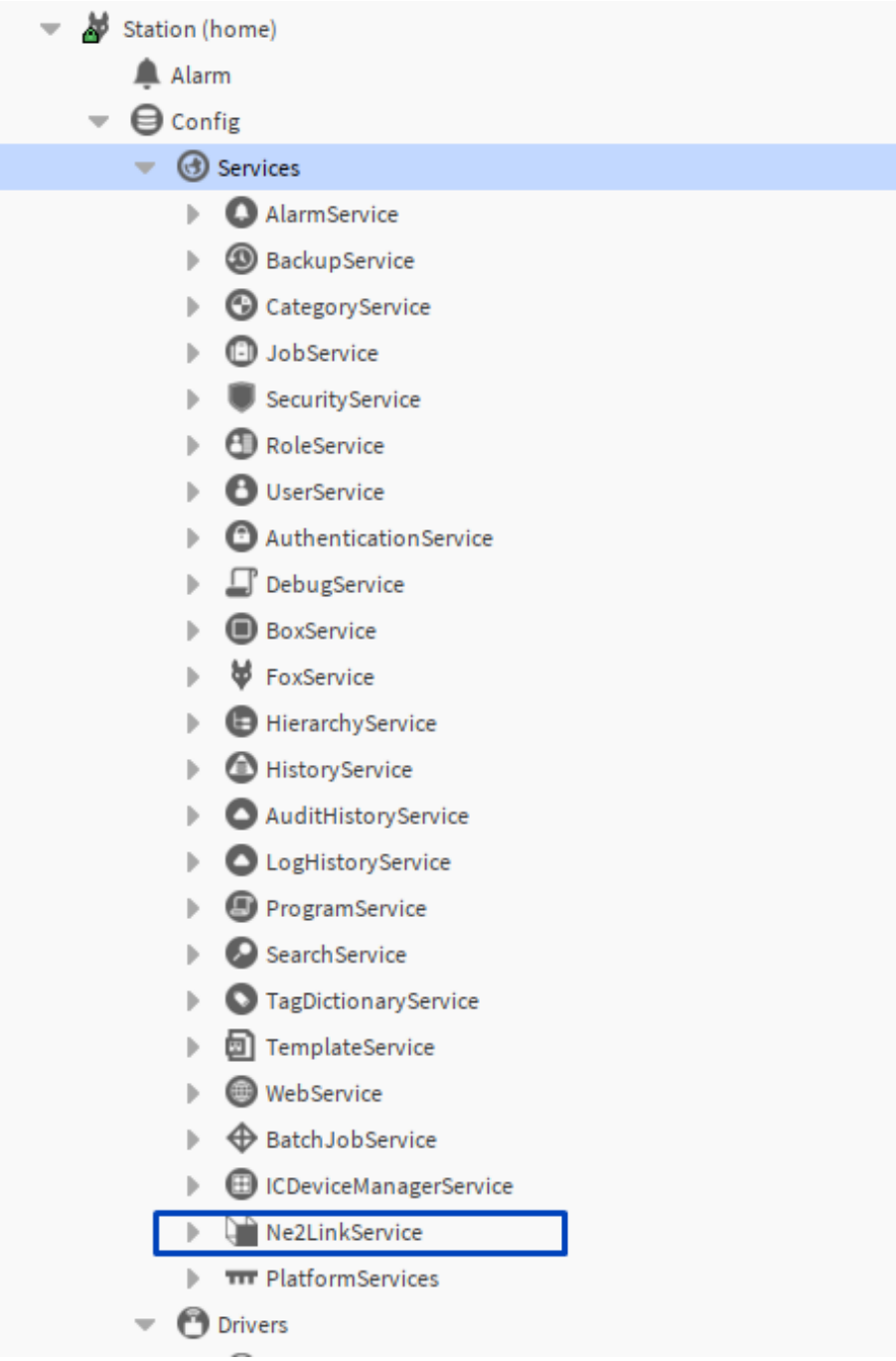


Figure 1. Ne2LinkService

The Ne2LinkService is added automatically to the station Services once the first nE2deviceExt is added to the BACnet device.

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).

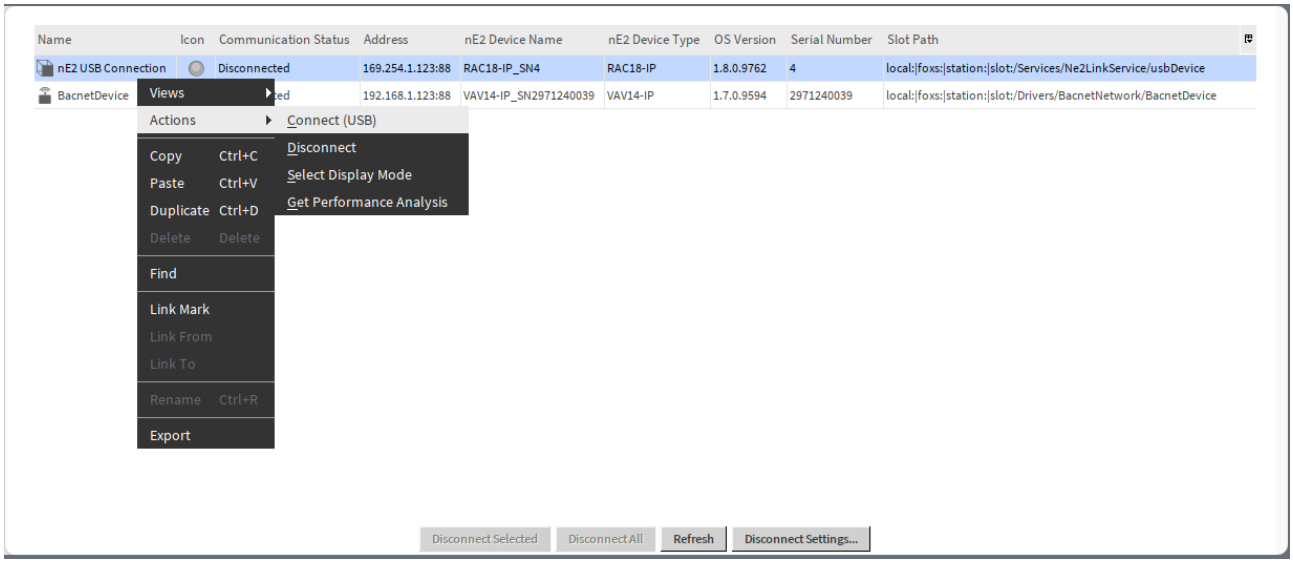


Figure 2. Ne2LinkService - USB connection

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

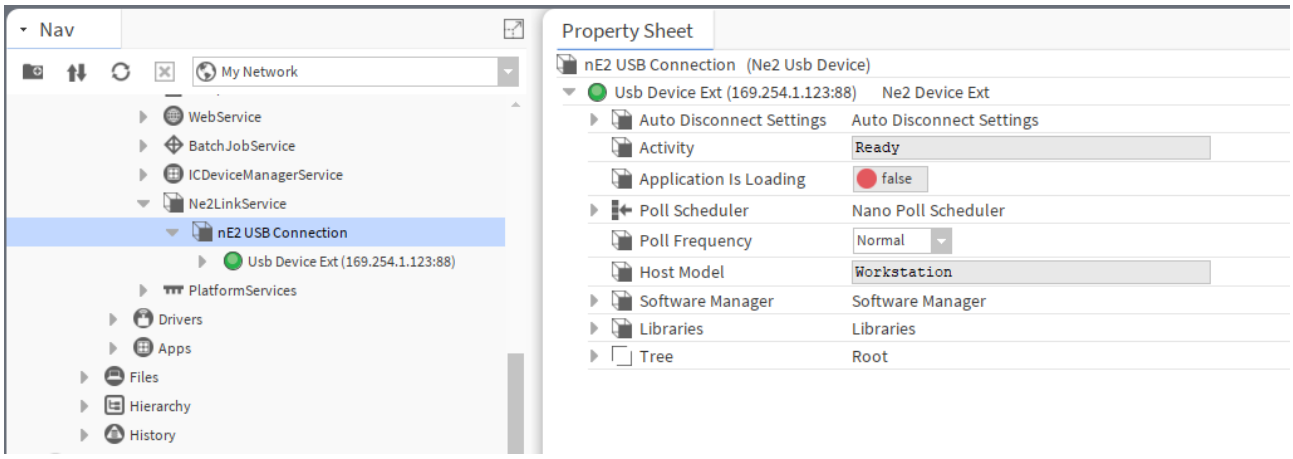


Figure 3. nE2 USB Connection component

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.

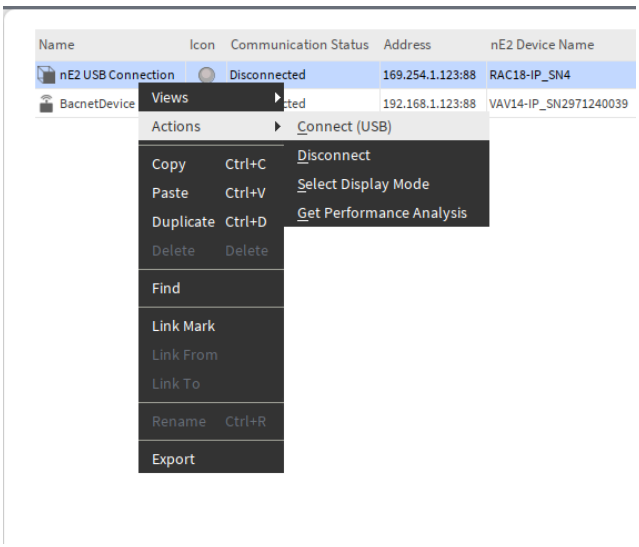


Figure 4. Connect (USB) option in the Ne2LinkService main view

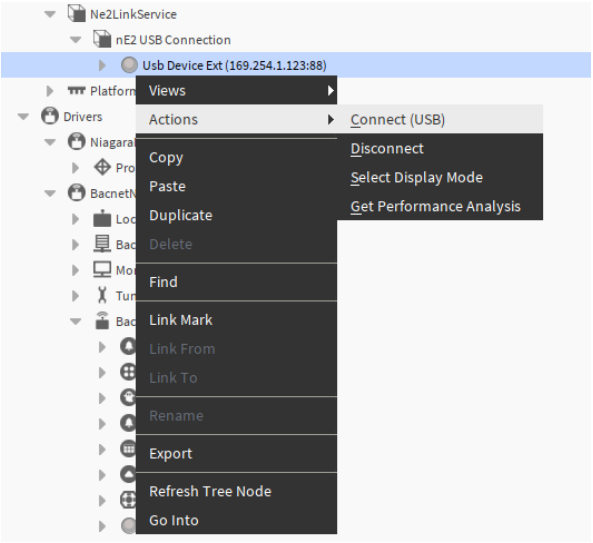


Figure 5. Connect (USB) option in the nE2 USB Connection component

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

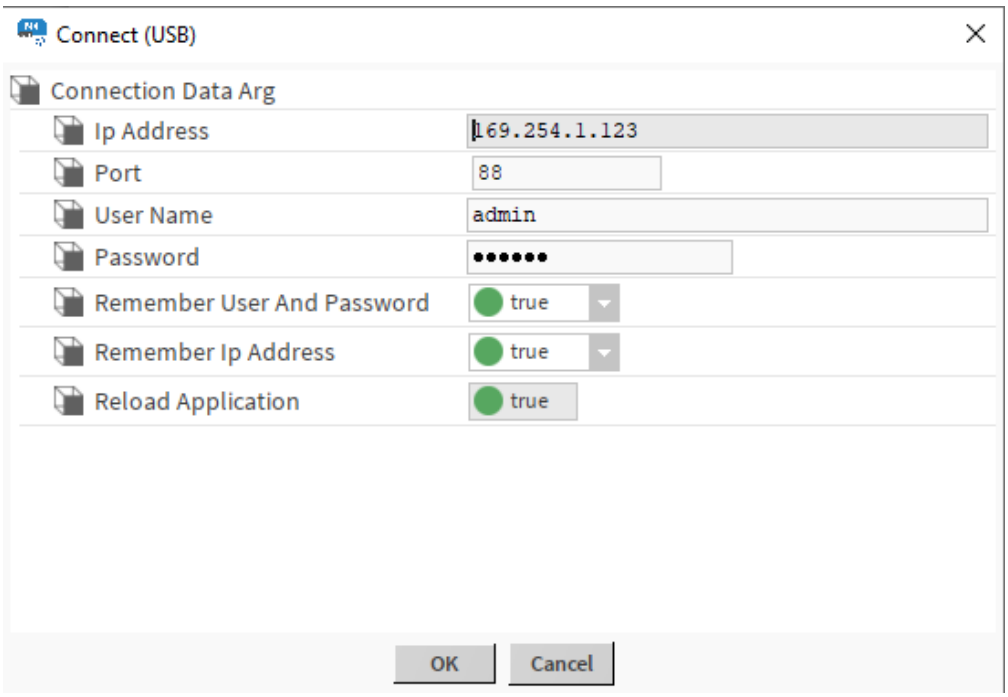


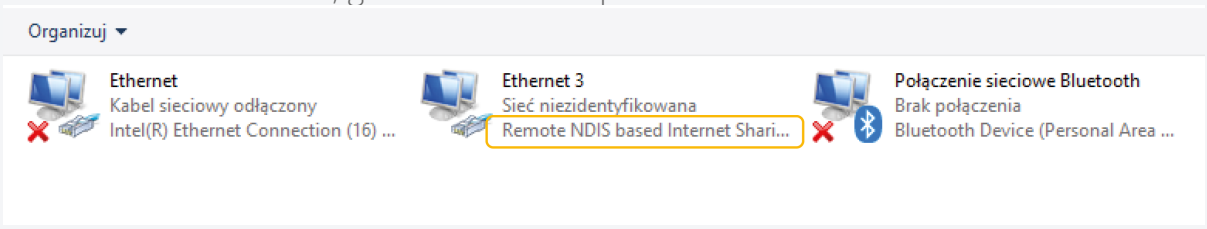
Figure 6. USB connection dialog window

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:



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.

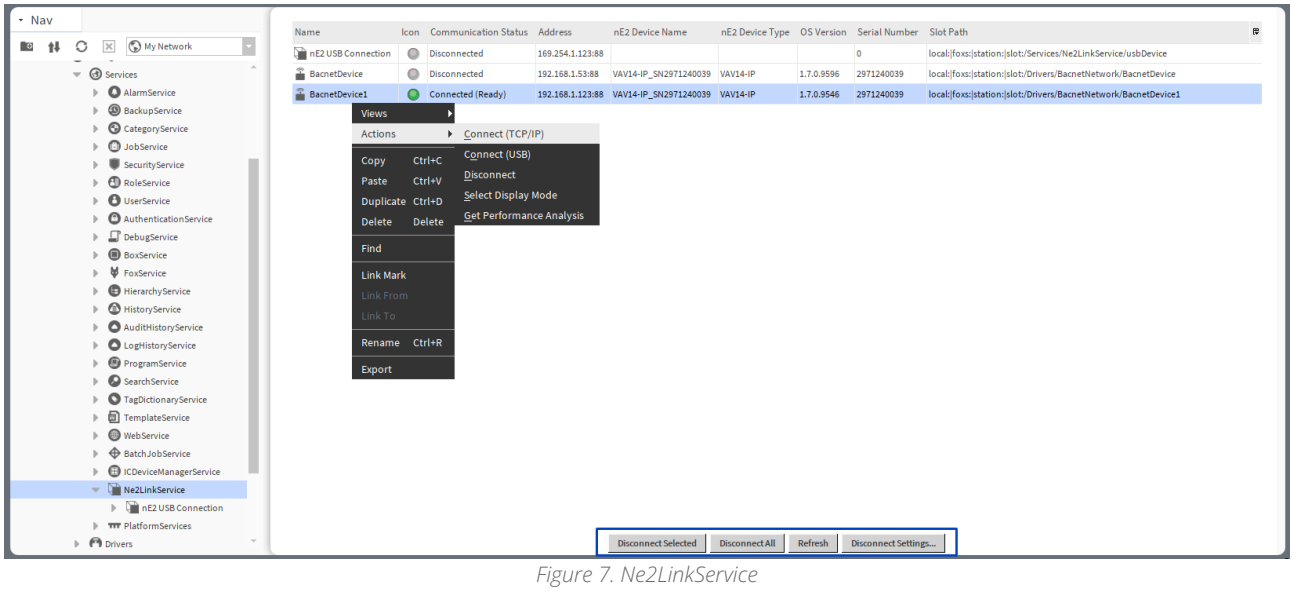


Figure 7. Ne2LinkService

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.

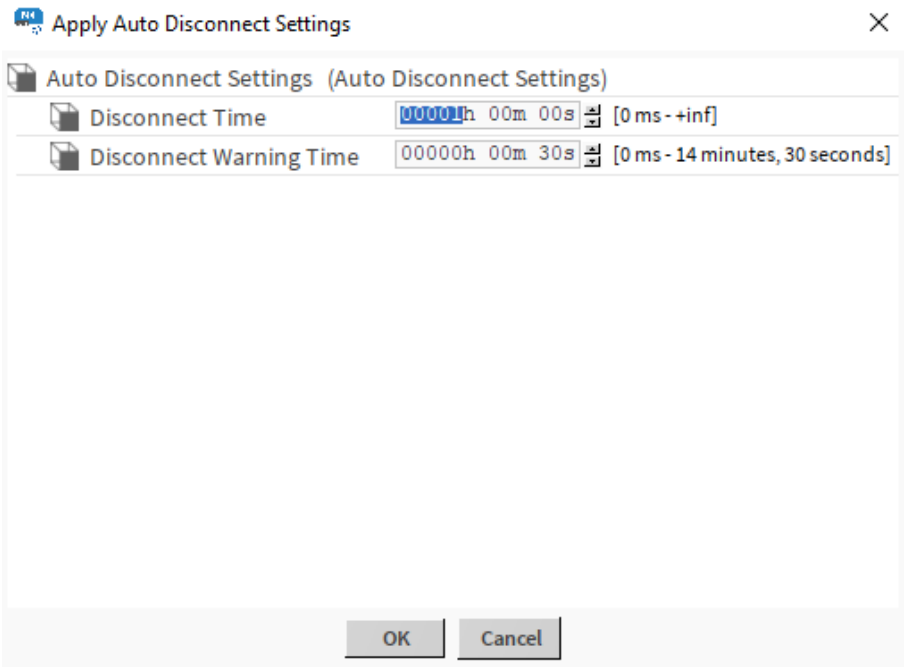
The Ne2LinkService has four action buttons which allow to perform the following:

- **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).

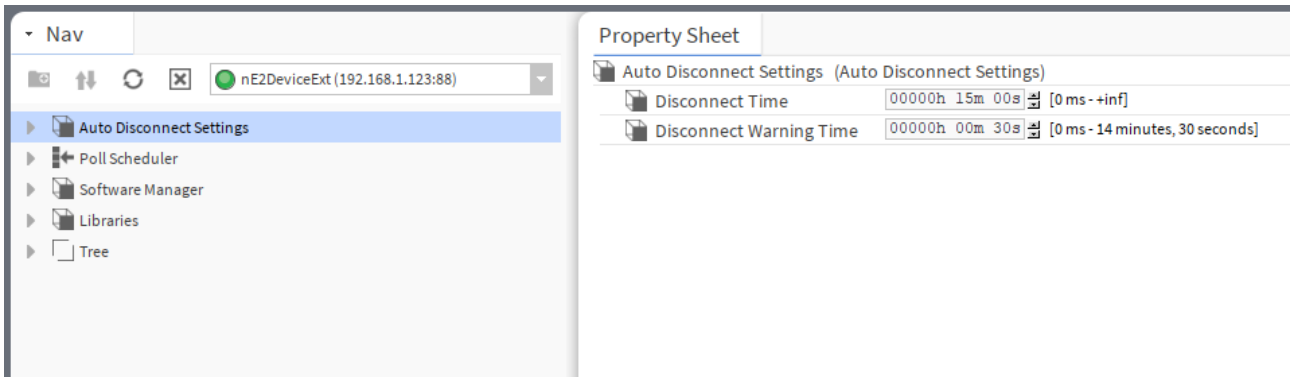
For user convenience, it is recommended to disconnect devices after commissioning or programming. The devices can be disconnected manually or by the auto-disconnect function, which helps unnecessary consumption of the resources, while the service also manages inactive connections (they are not lost after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



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



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

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.

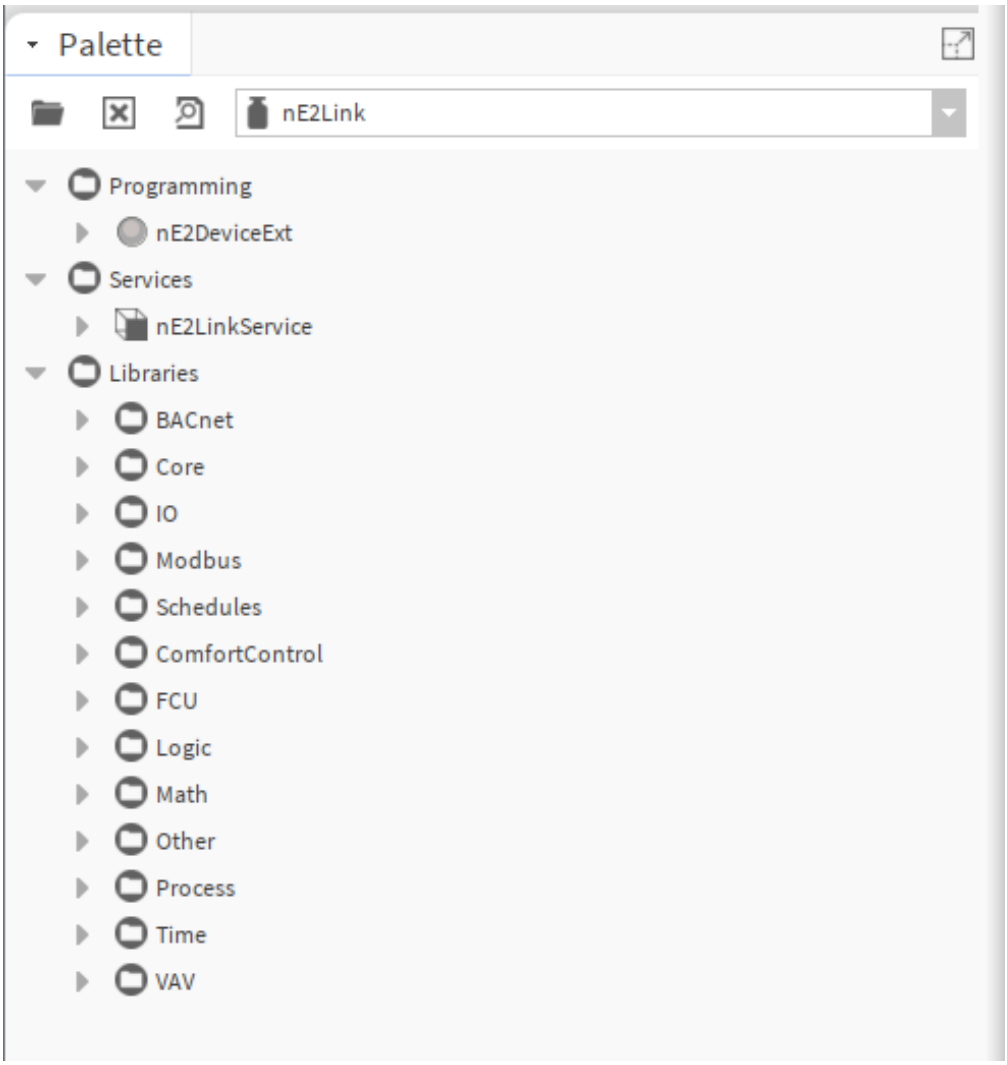


Figure 8. nE2Device extension in the nE2Link module

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.

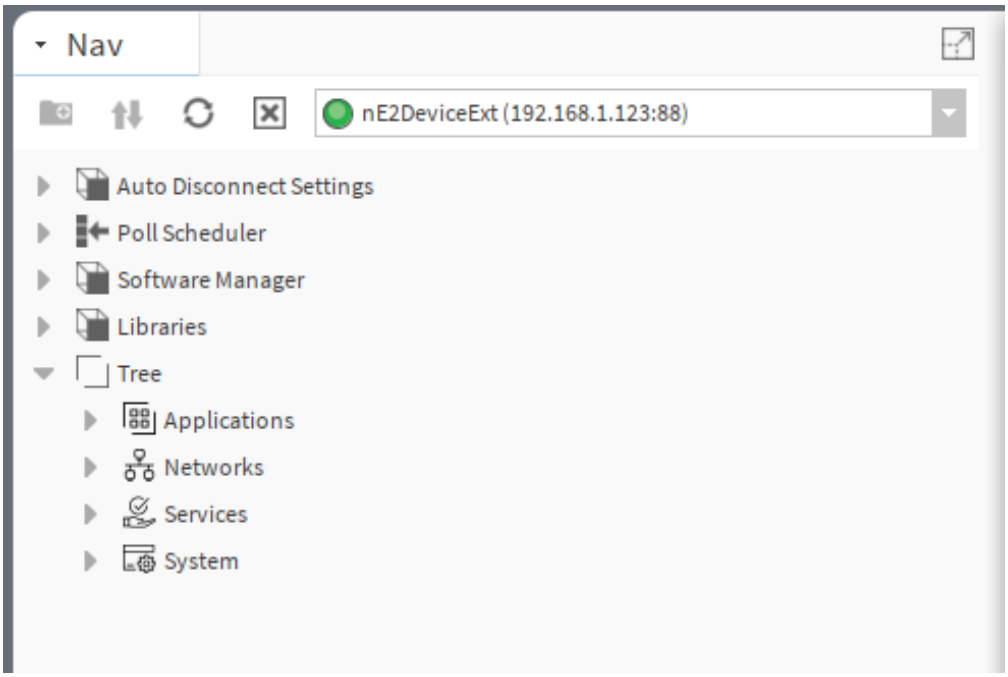


Figure 9. nE2DeviceExt contents

(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.

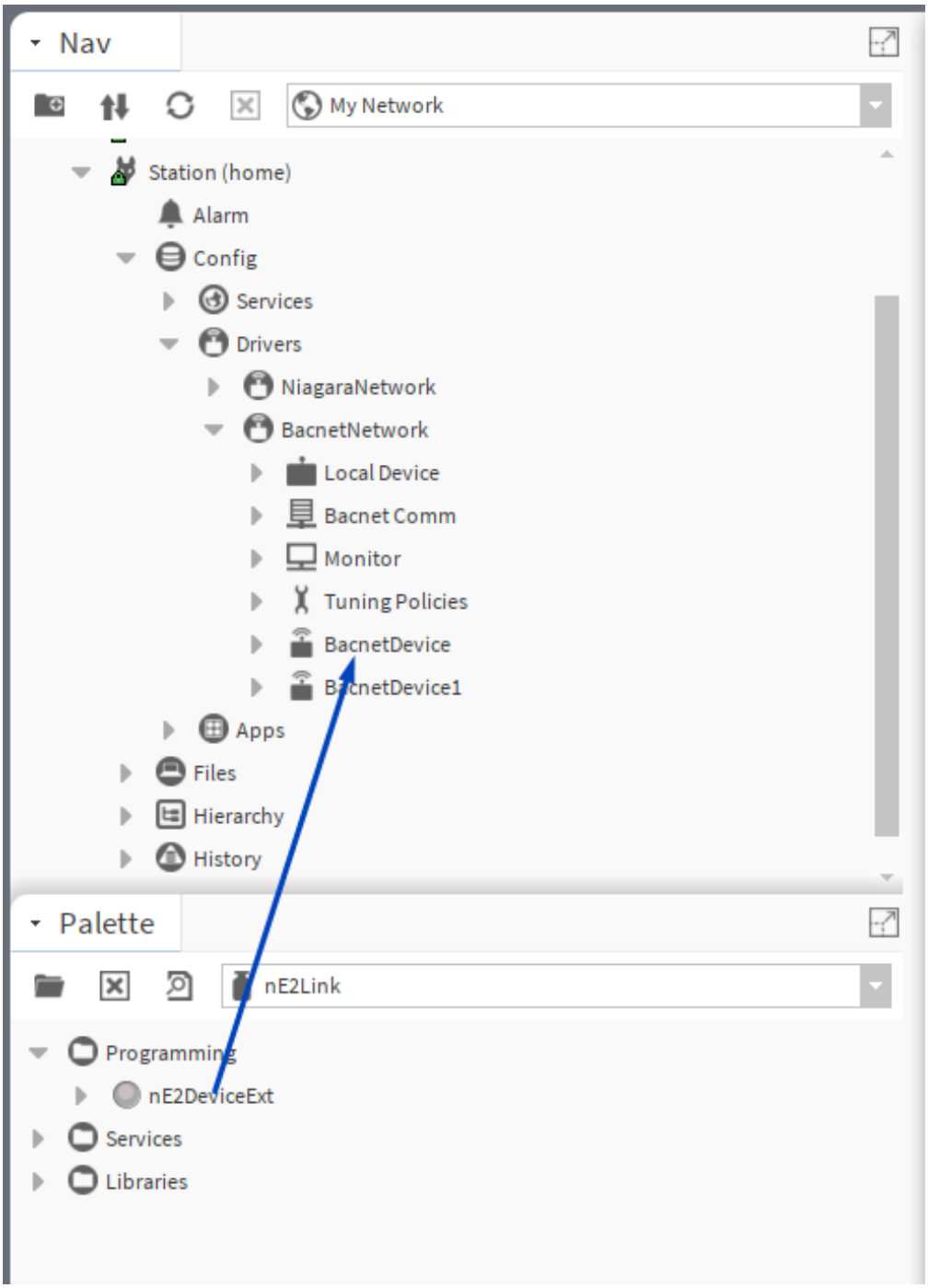


Figure 10. Adding nE2Device Ext to the BACnet device

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.

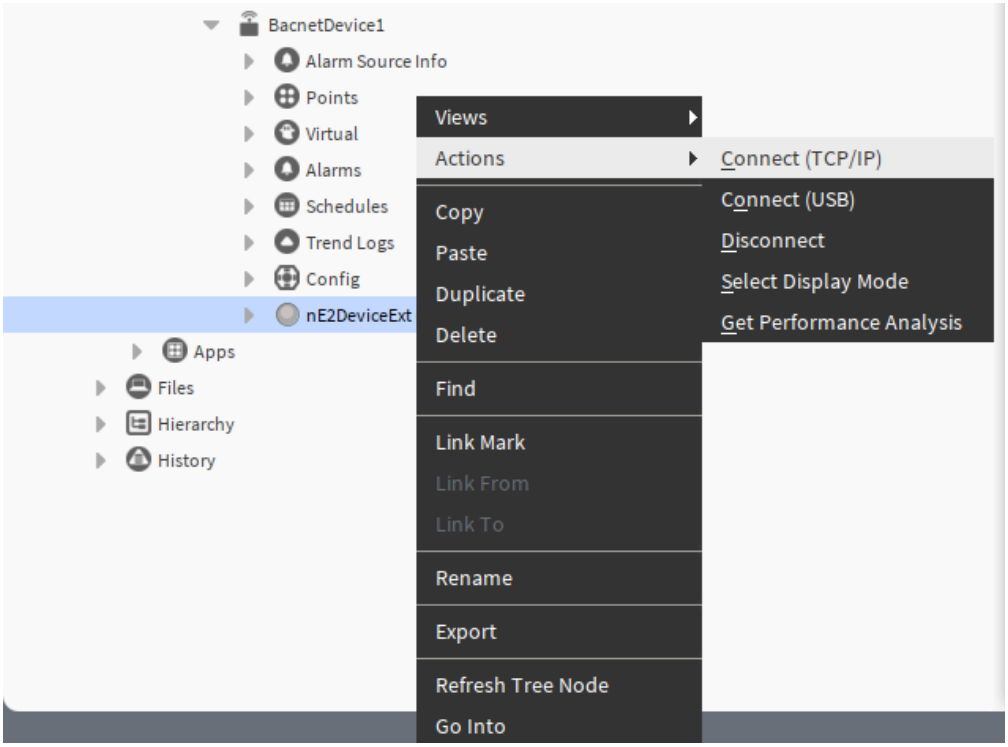


Figure 11. Connect option

(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.

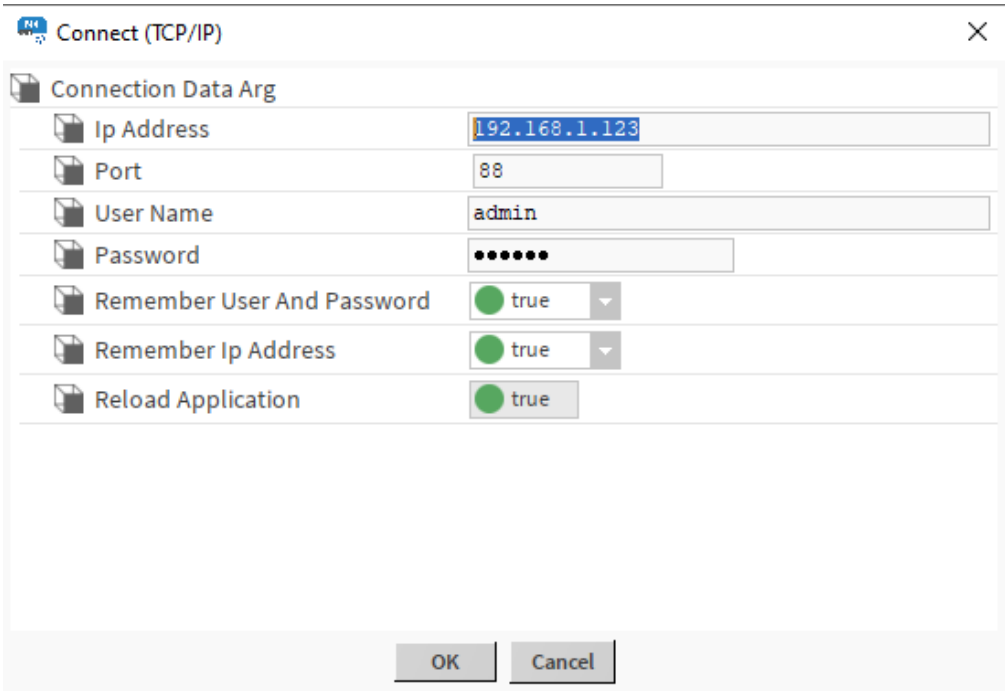


Figure 12. TCP/IP connection pop-up

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.

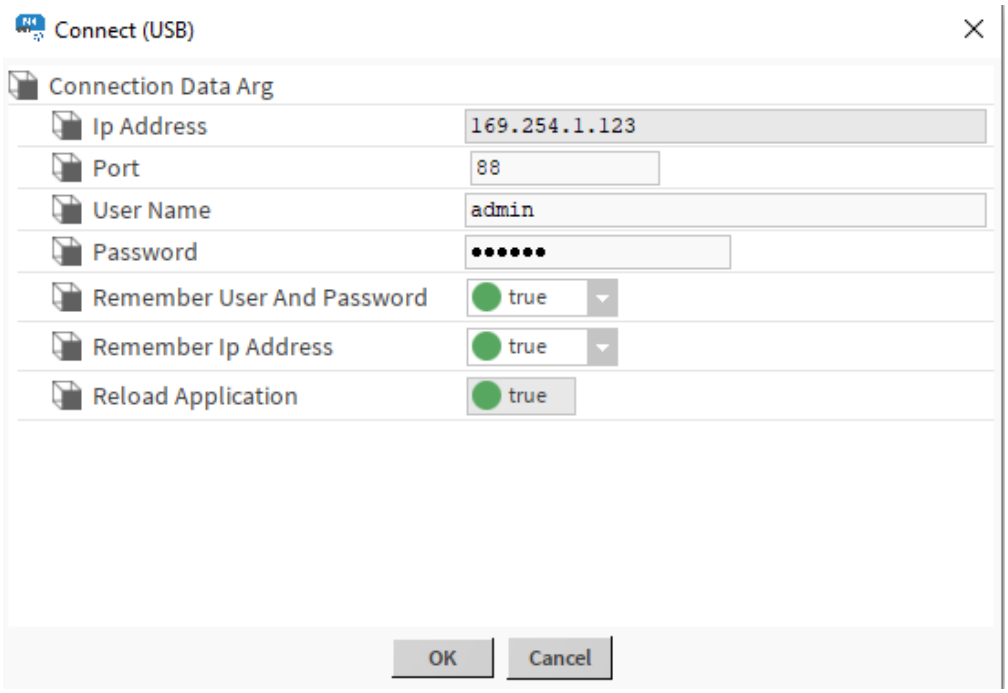


Figure 13. USB connection pop-up

(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 14. 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.

Worth to notice:

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

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.

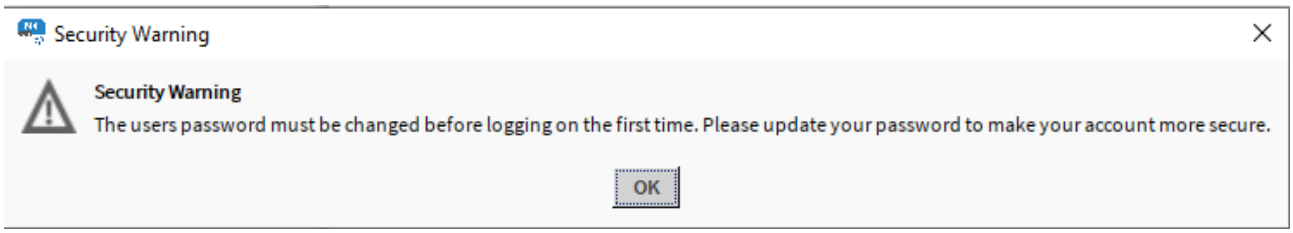


Figure 15. Change password prompt

Changing password is obligatory when first connecting to the device!

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

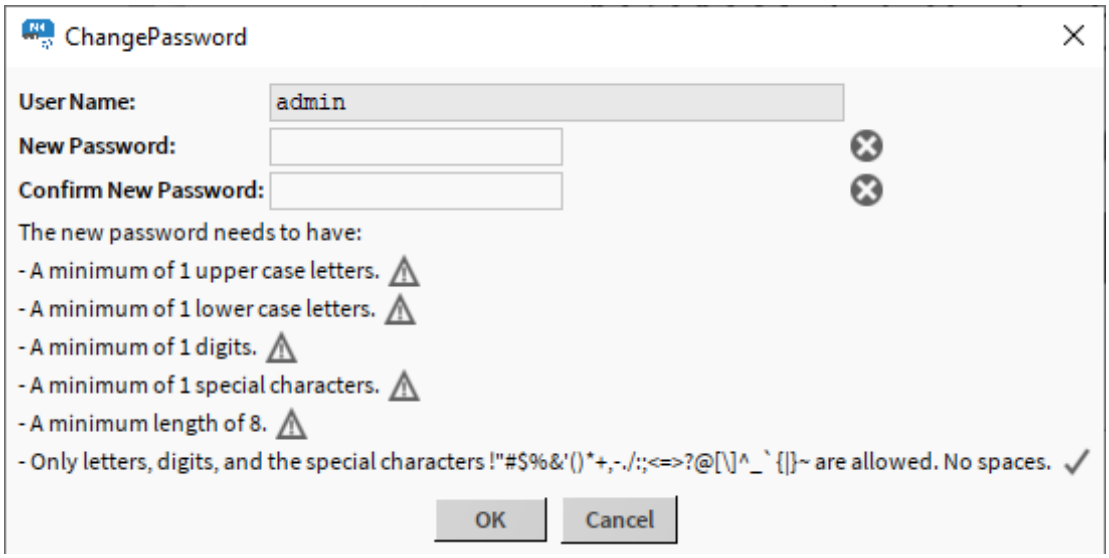


Figure 16. ChangePassword dialog window

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.

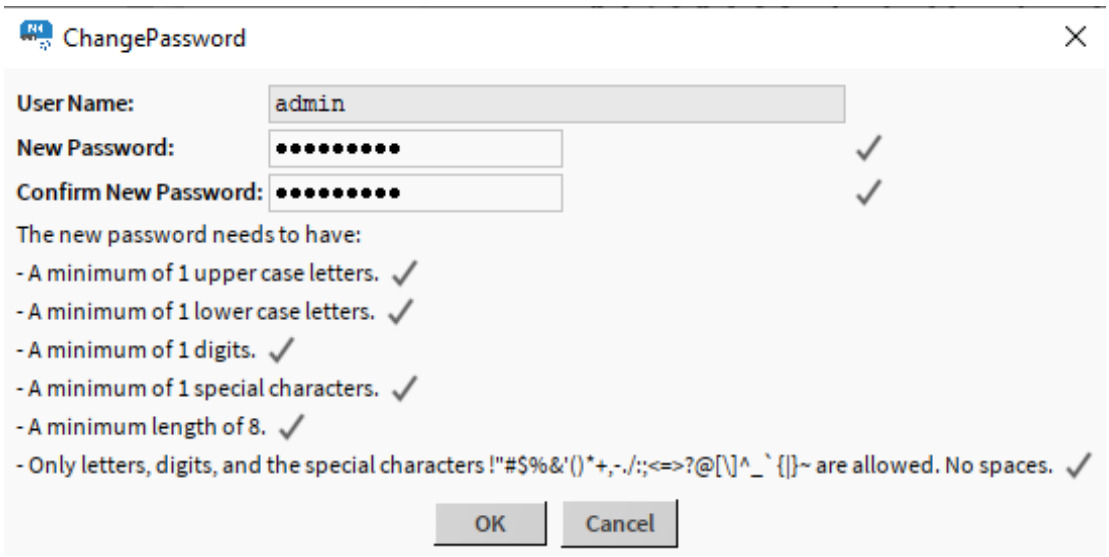


Figure 17. New password fulfilling all requirements

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:

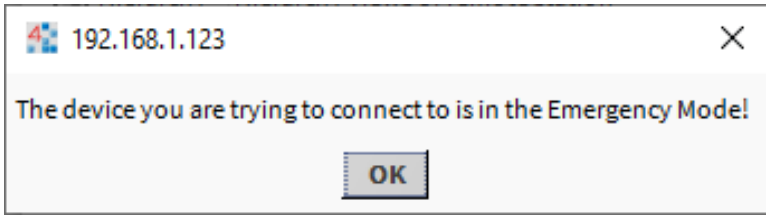


Figure 18. Emergency mode notice

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;

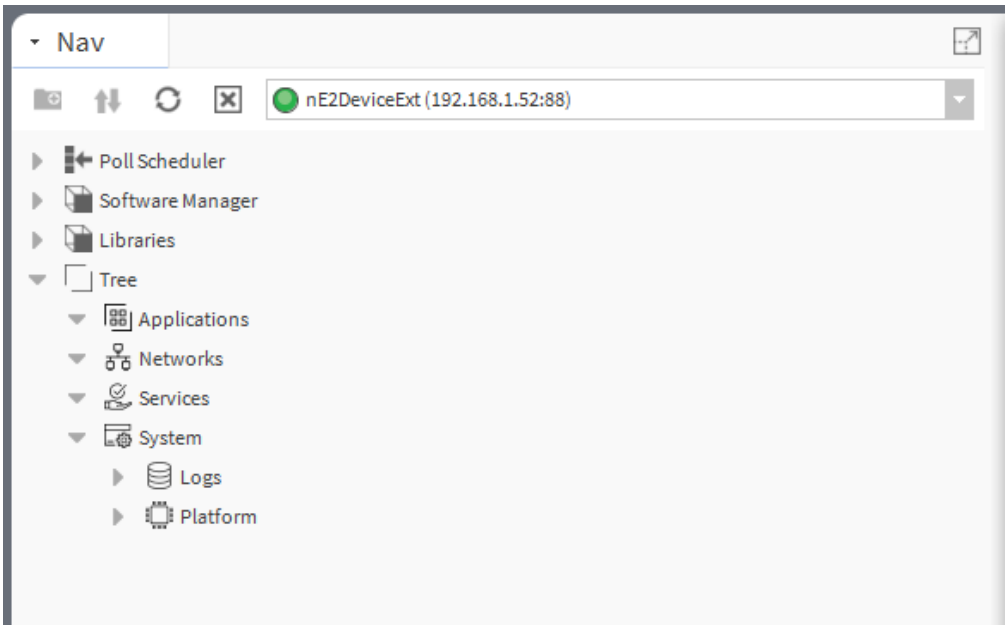


Figure 19. Device tree in the emergency mode

- 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).

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 20. 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);

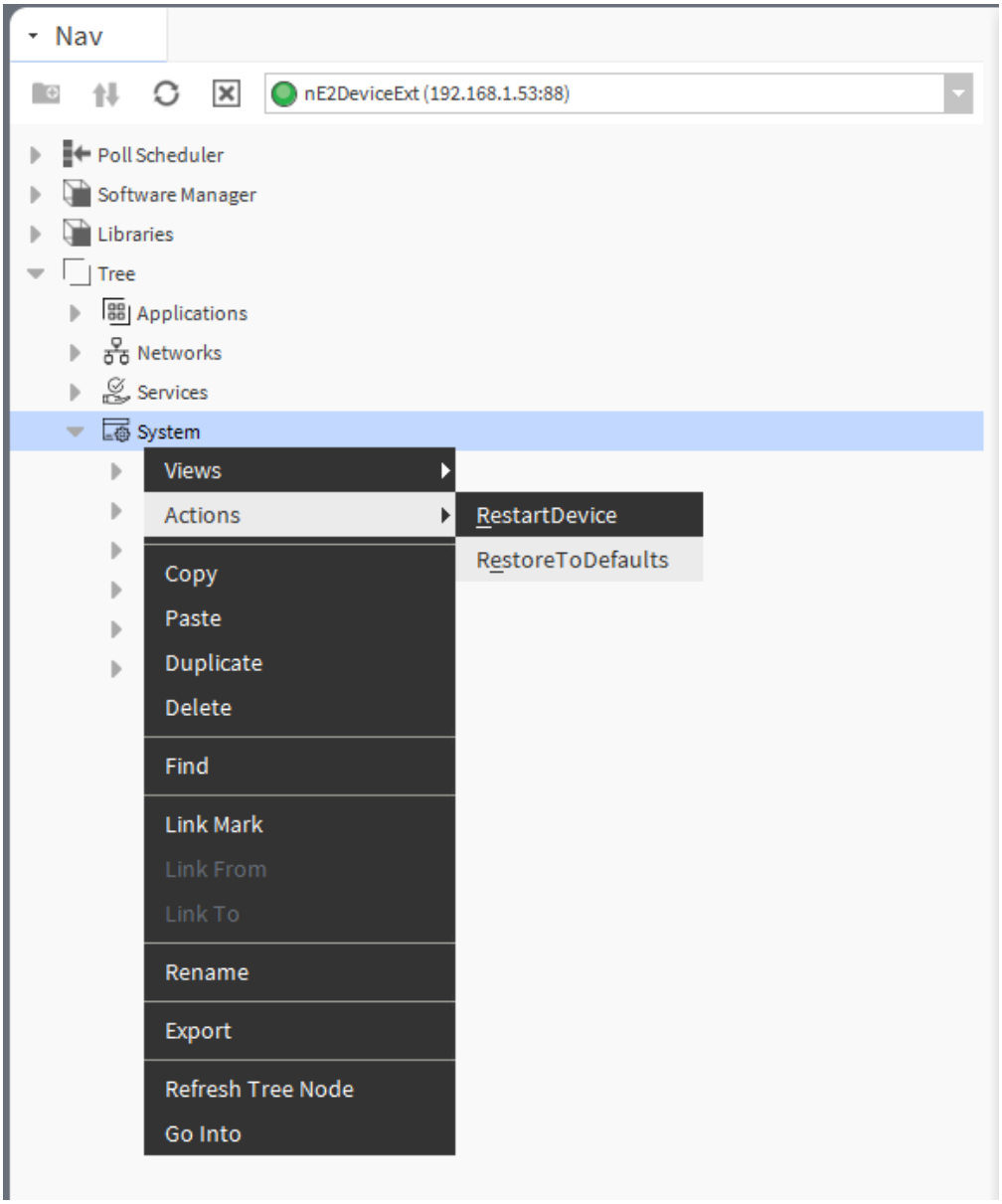


Figure 21. RestoreToDefaults action

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.

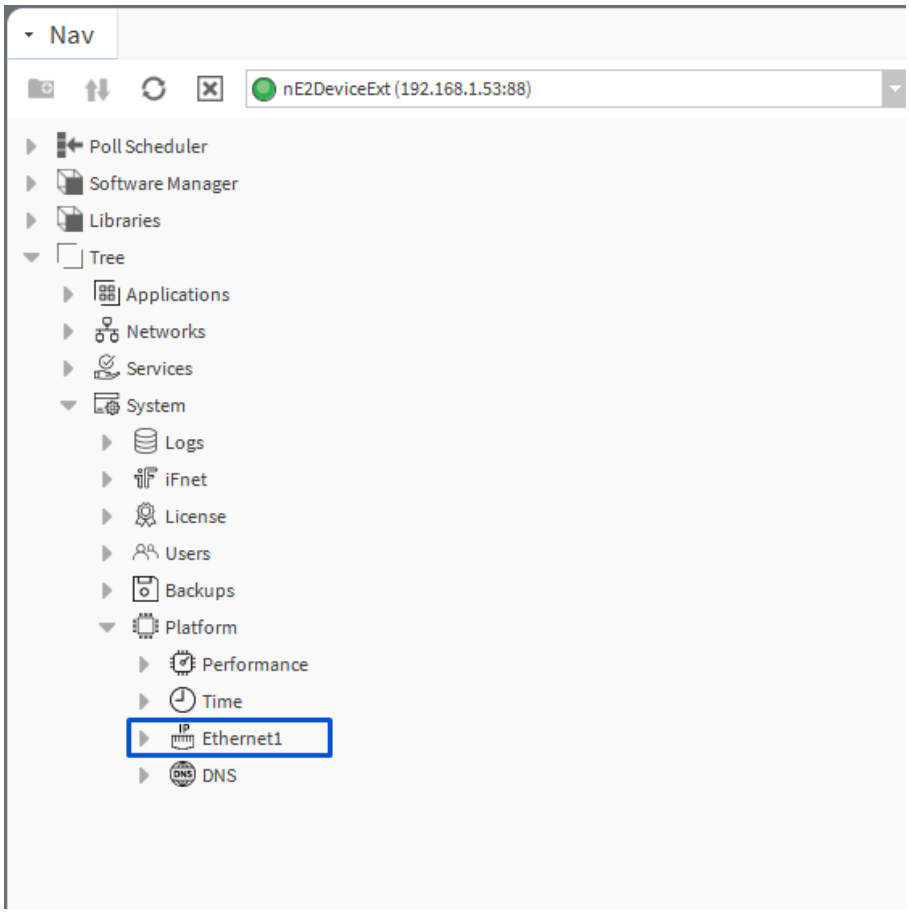


Figure 22. The Ethernet1 component in the tree

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.

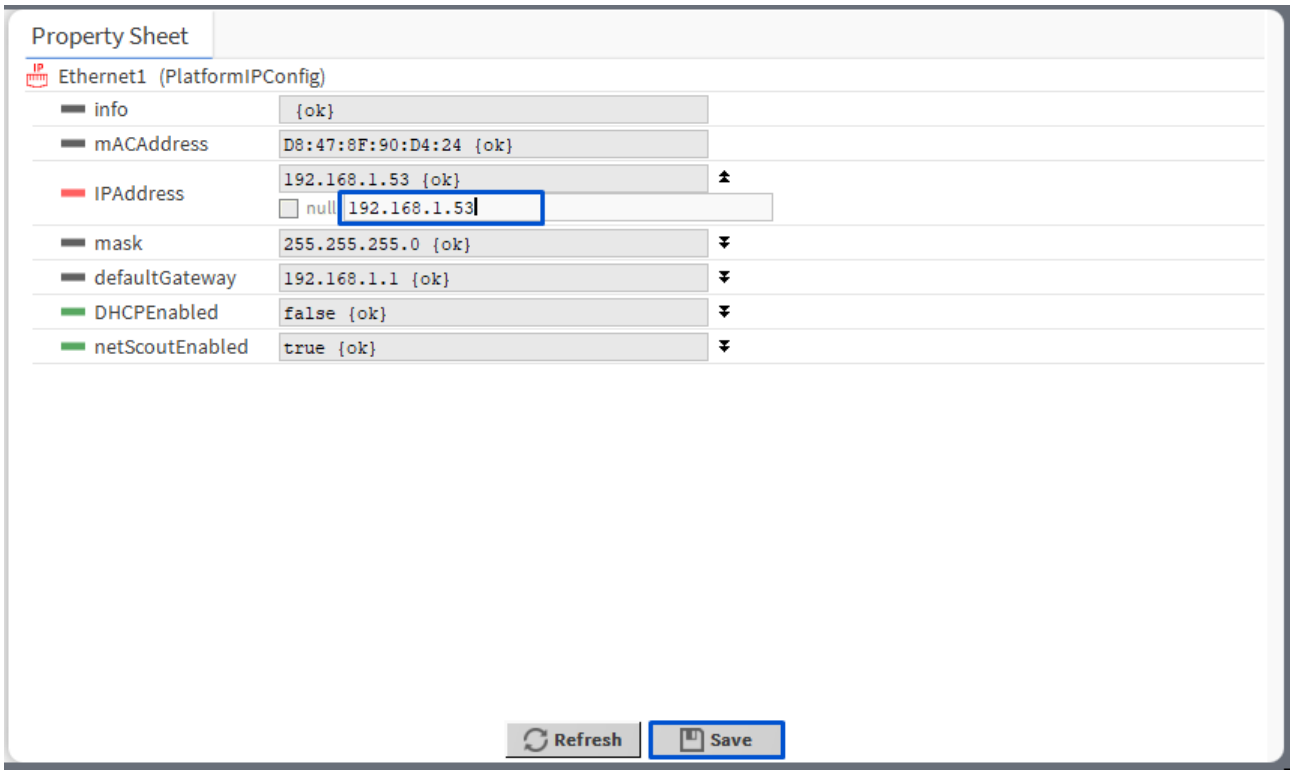


Figure 23. Ethernet1 properties

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:

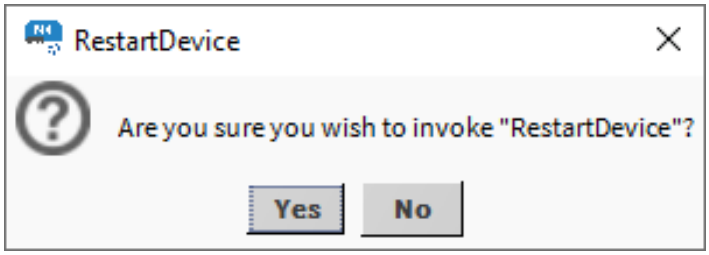


Figure 24. RestartDevcie prompt

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

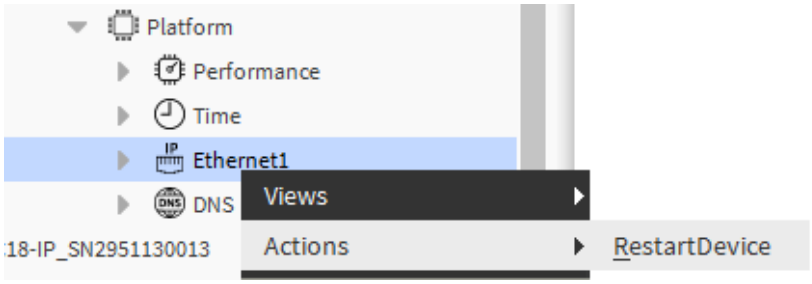


Figure 25. RestartDevice action in the Ethernet1 component

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.

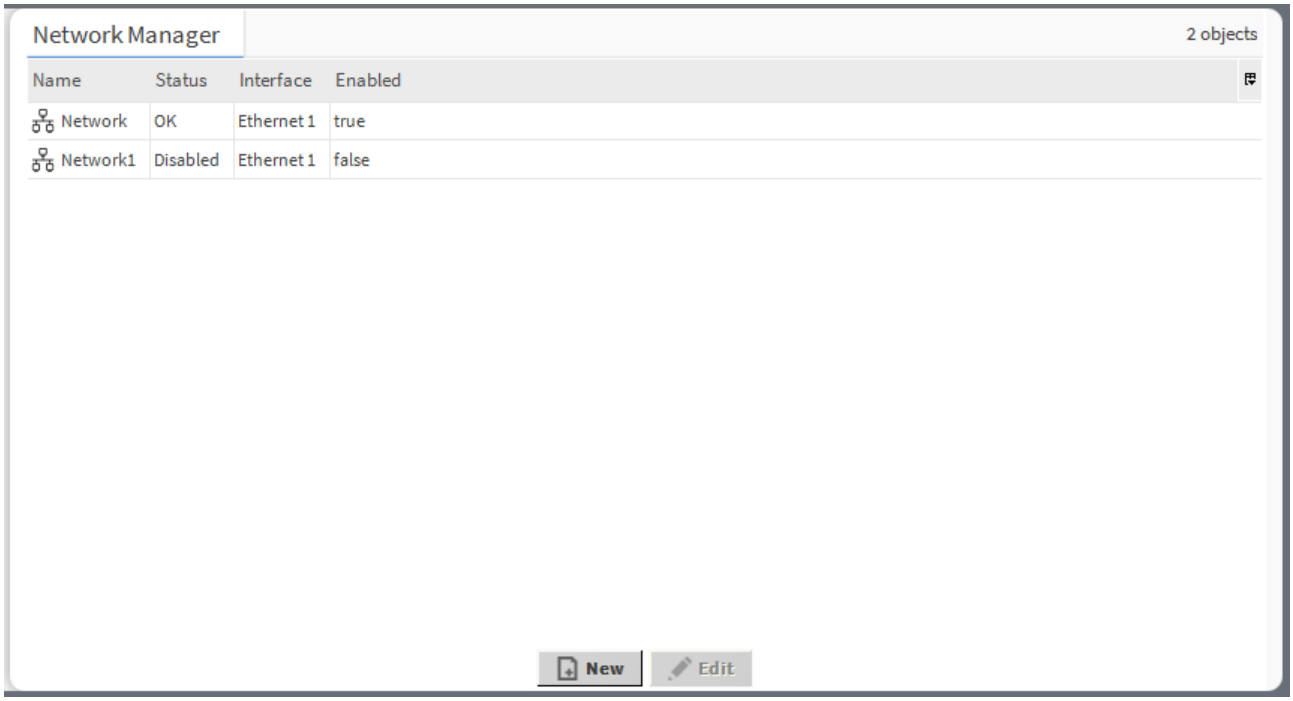


Figure 26. BACnet Network Manager

In the BACnet Network Manager, it is possible to:

- add BACnet network component:

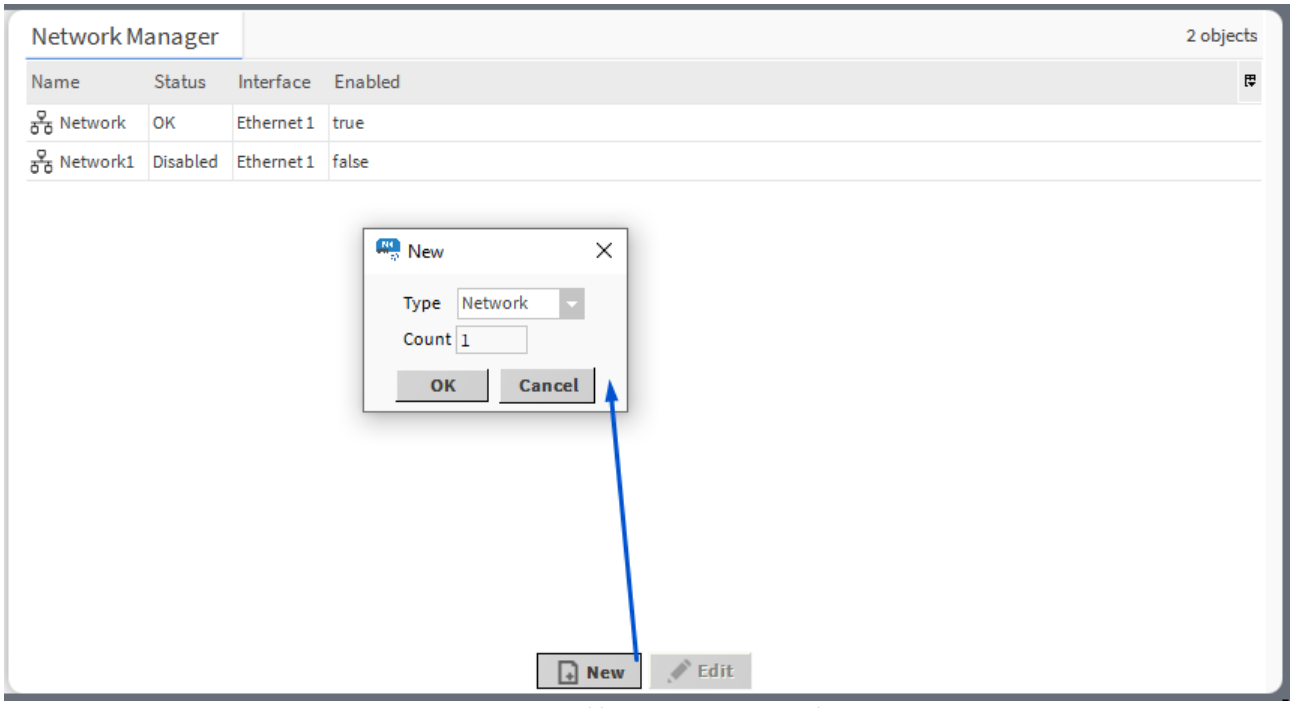


Figure 27. Adding new BACnet network

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

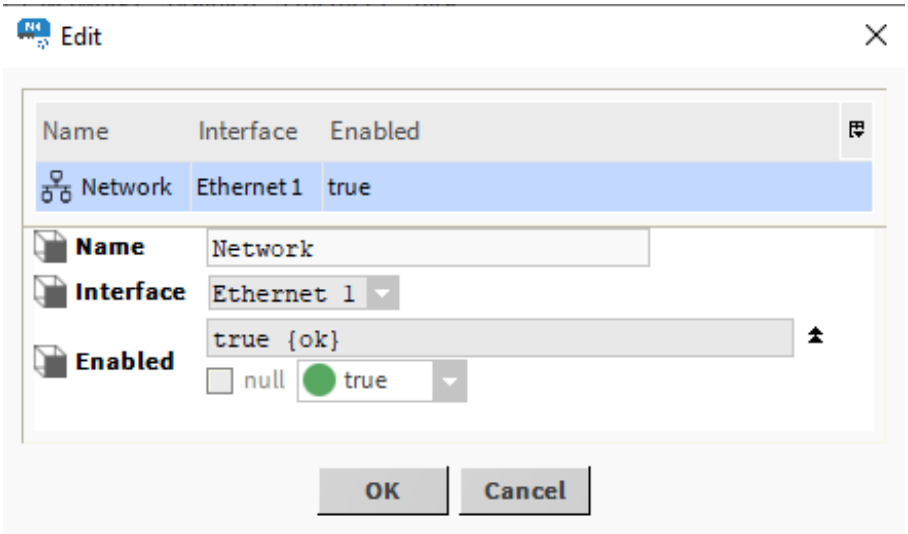


Figure 28. Editing pop-up

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.

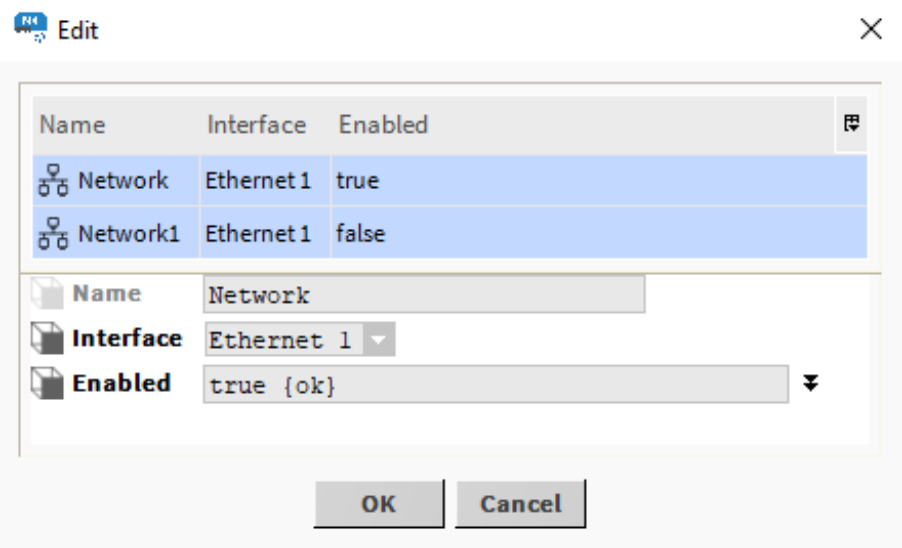


Figure 29. Editing multiple BACnet networks

- copy/duplicate/remove BACnet network components:

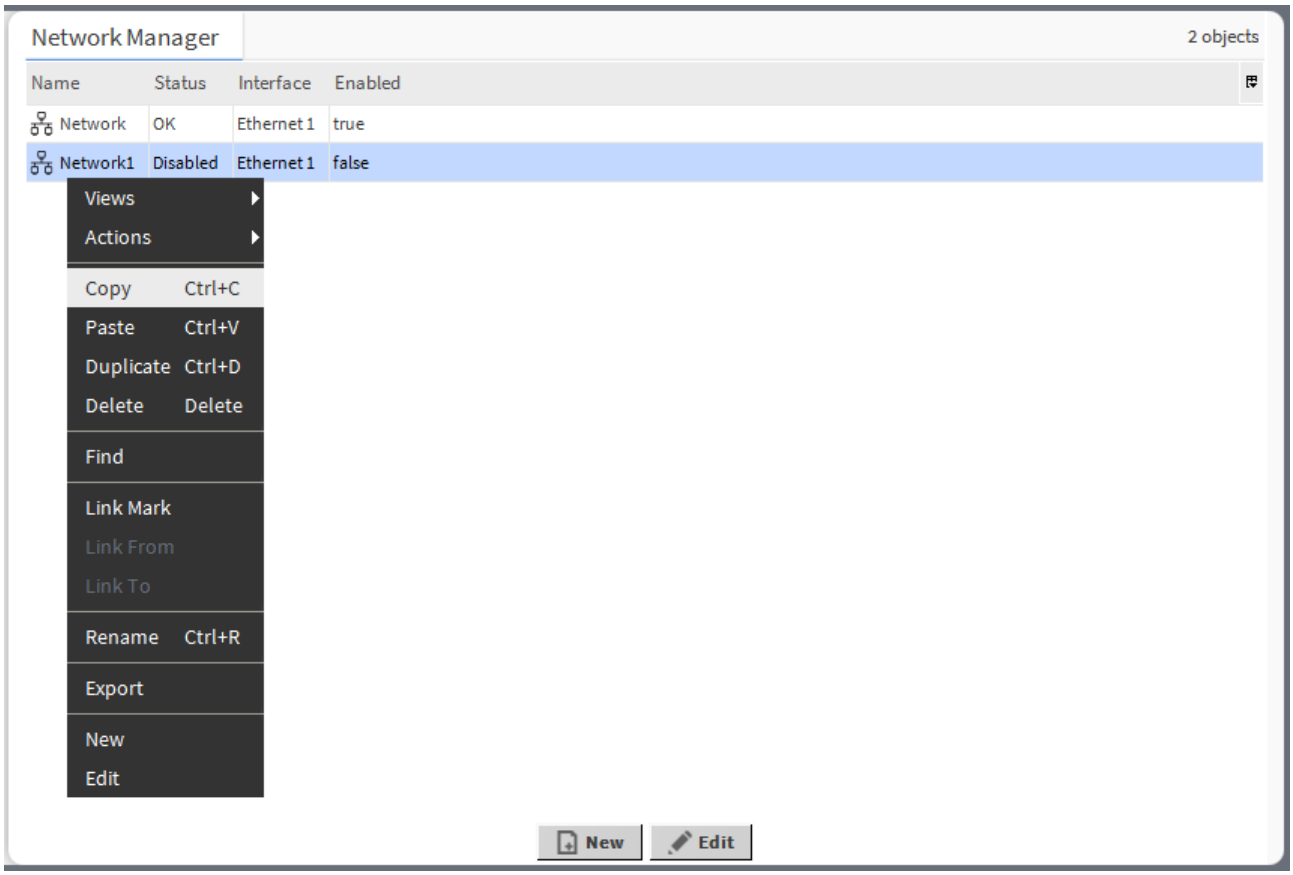


Figure 30. Context menu options for BACnet network

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.

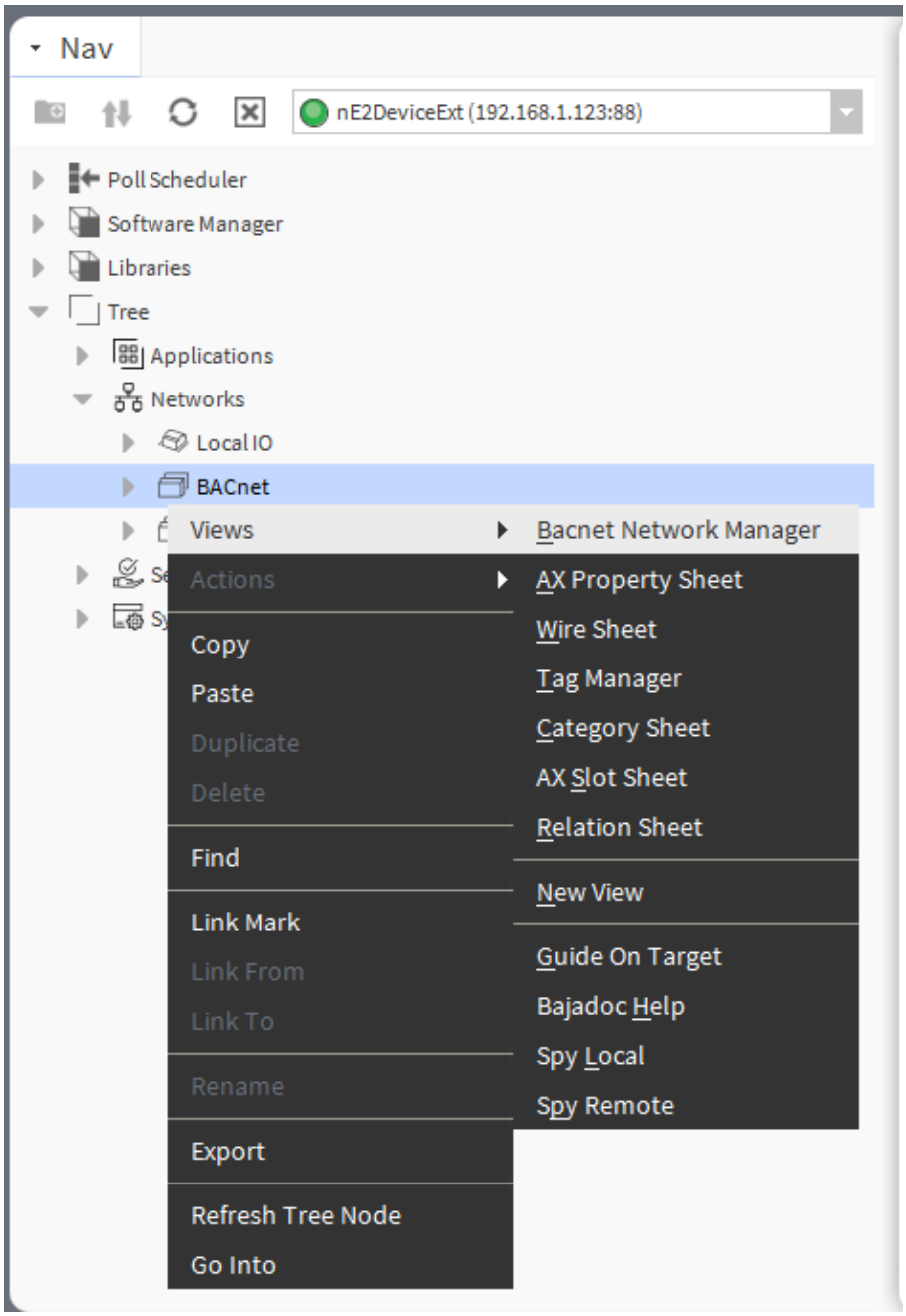


Figure 31. Accessing the BACnet Network Manager

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).

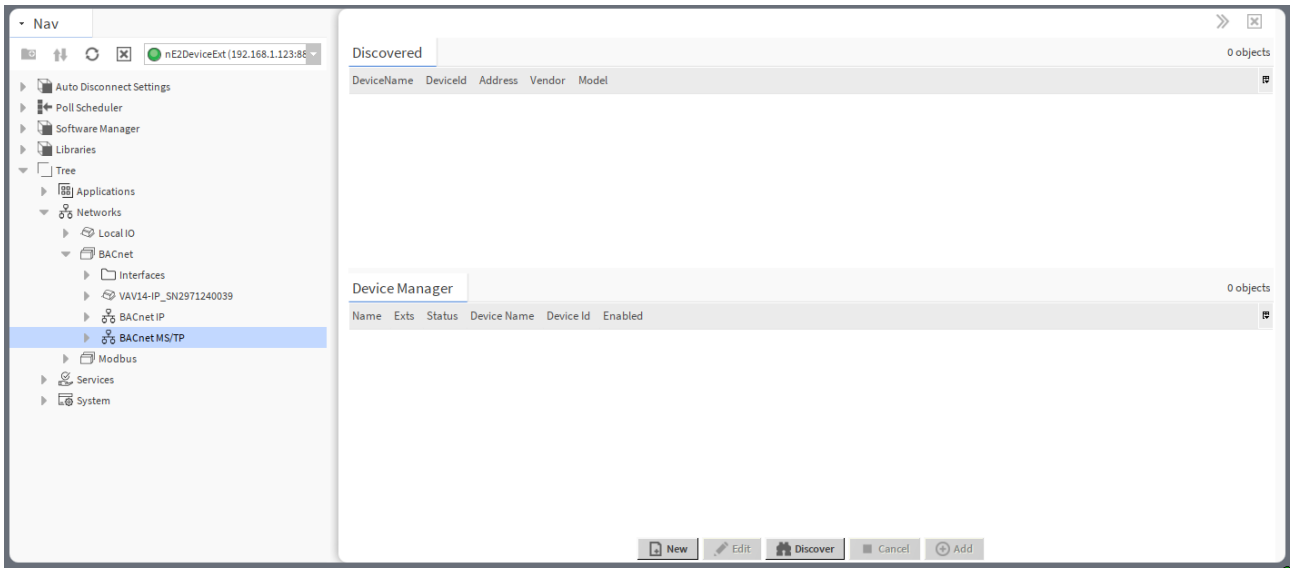


Figure 32. BACnet Device Discover Manager

In the BACnet Device Manager, it is possible to:

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

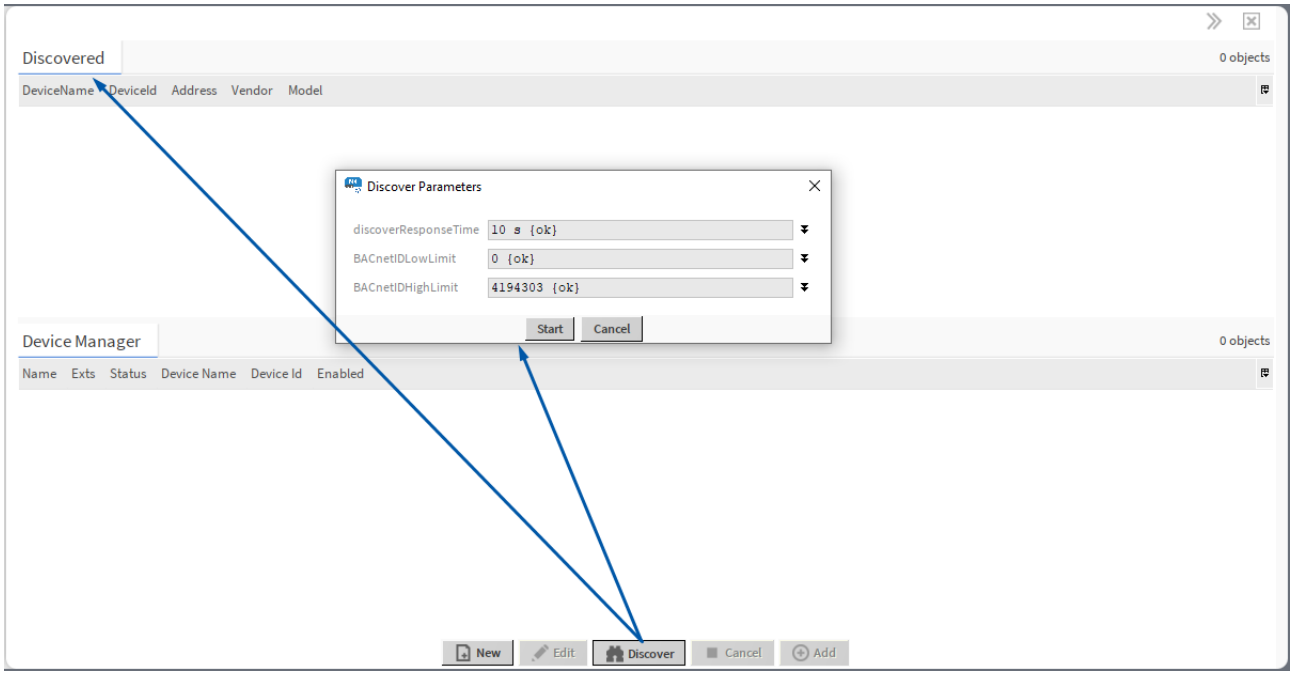


Figure 33. Discovering devices on the BACnet network

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.

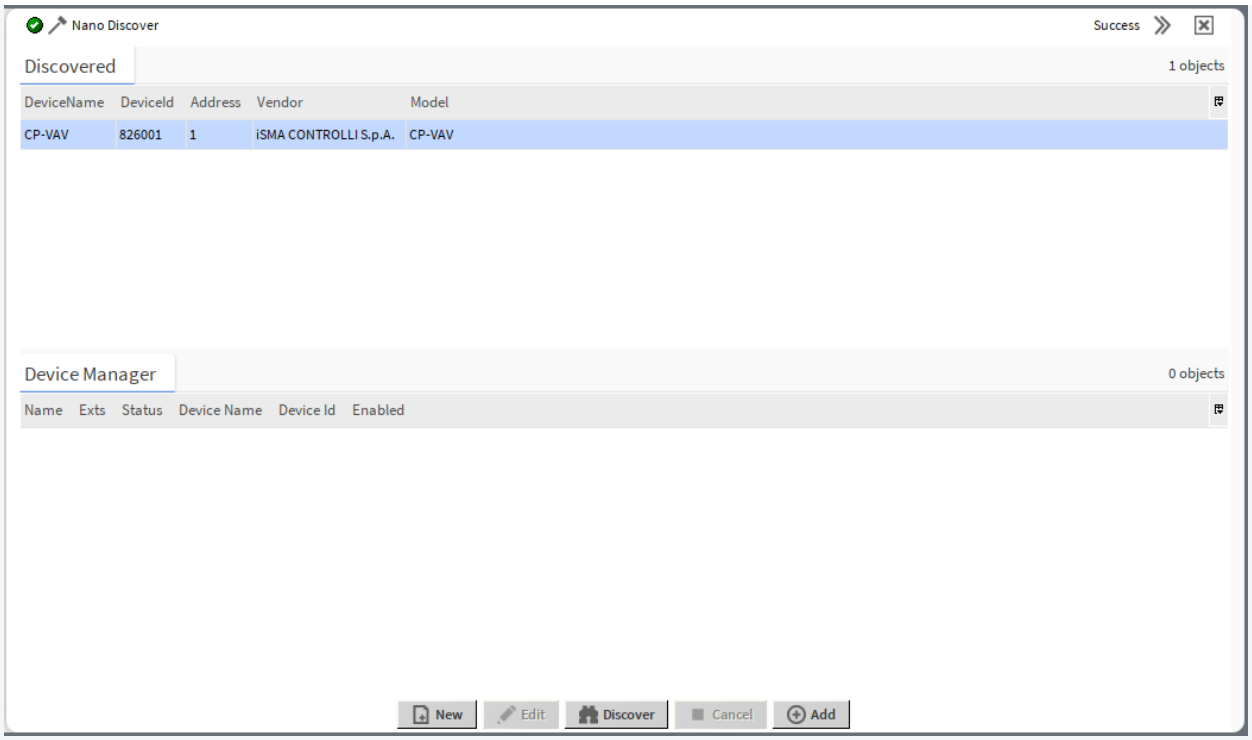


Figure 34. Device discovered

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:

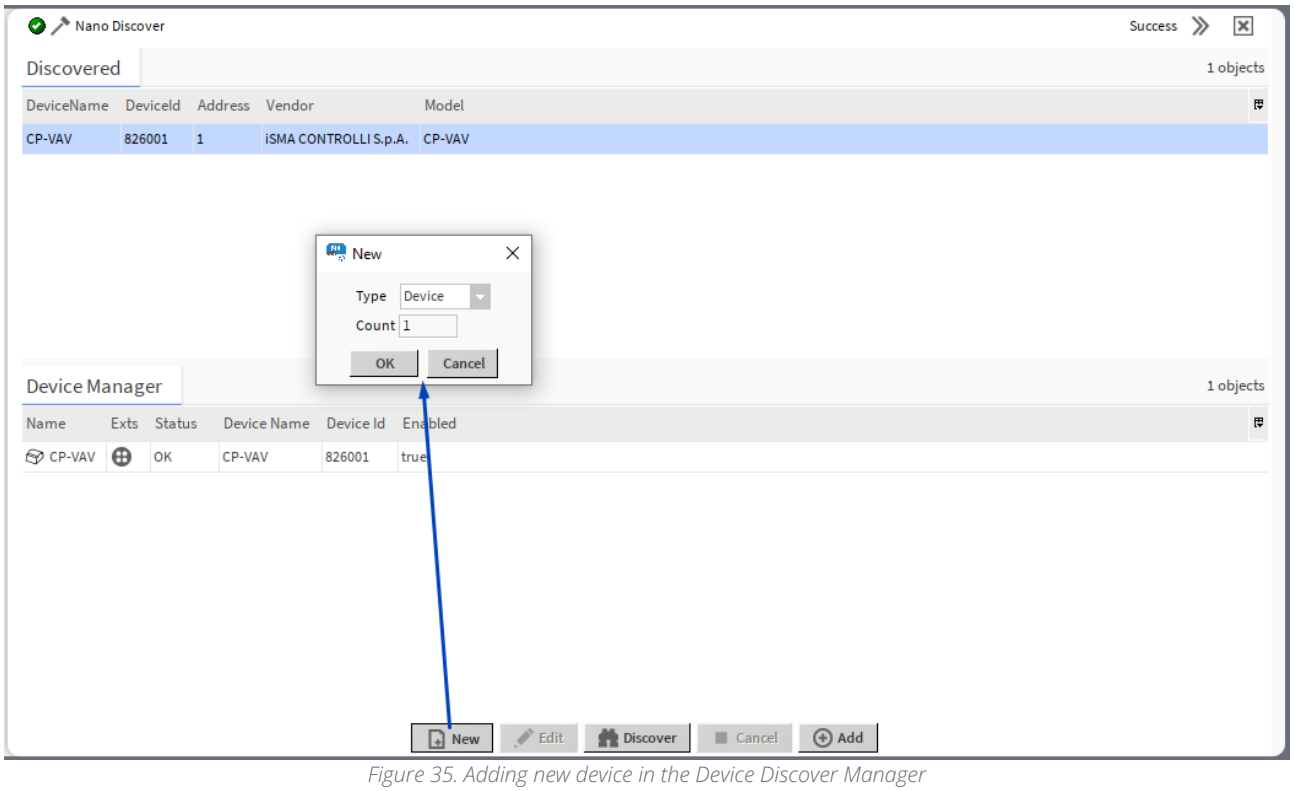


Figure 35. Adding new device in the Device Discover Manager

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

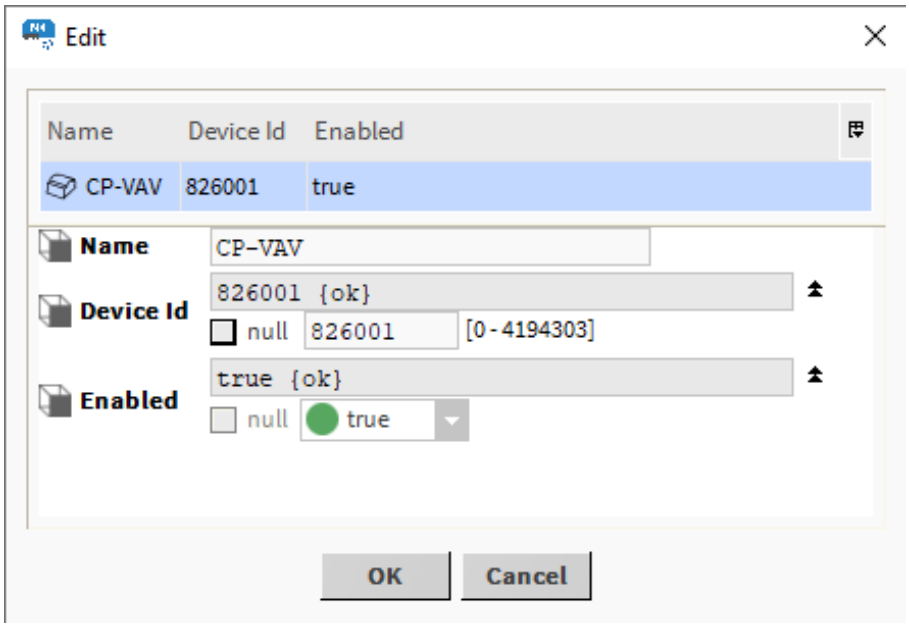


Figure 36. Editing pop-up

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.

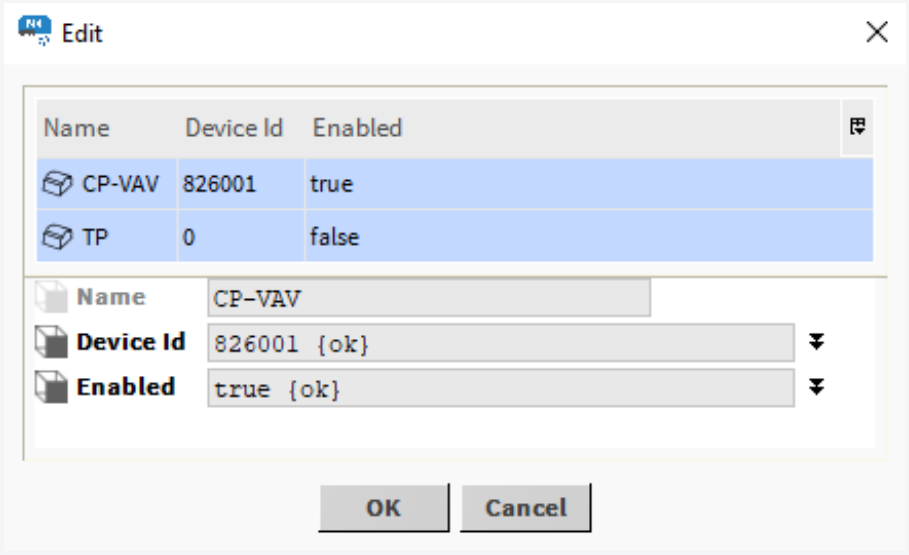


Figure 37. Editing multiple devices

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

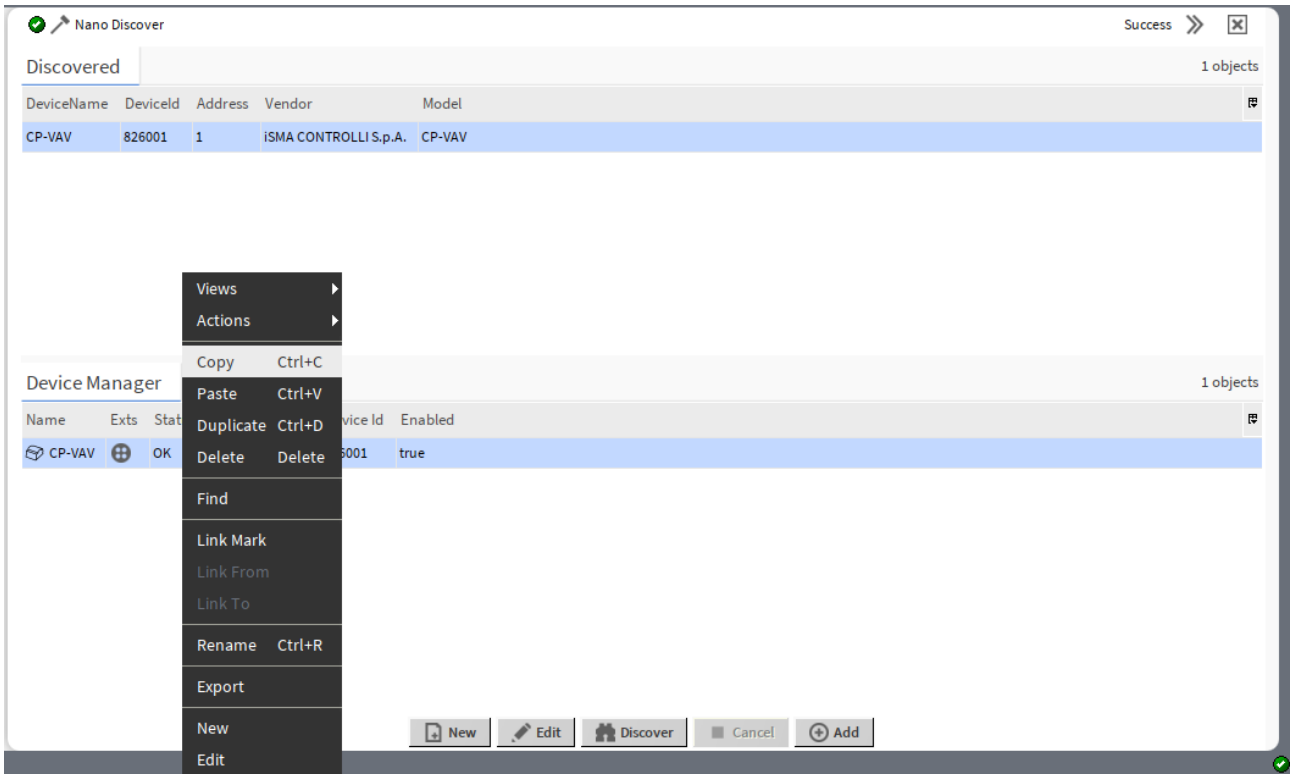


Figure 38. Context menu options for the BACnet device

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.

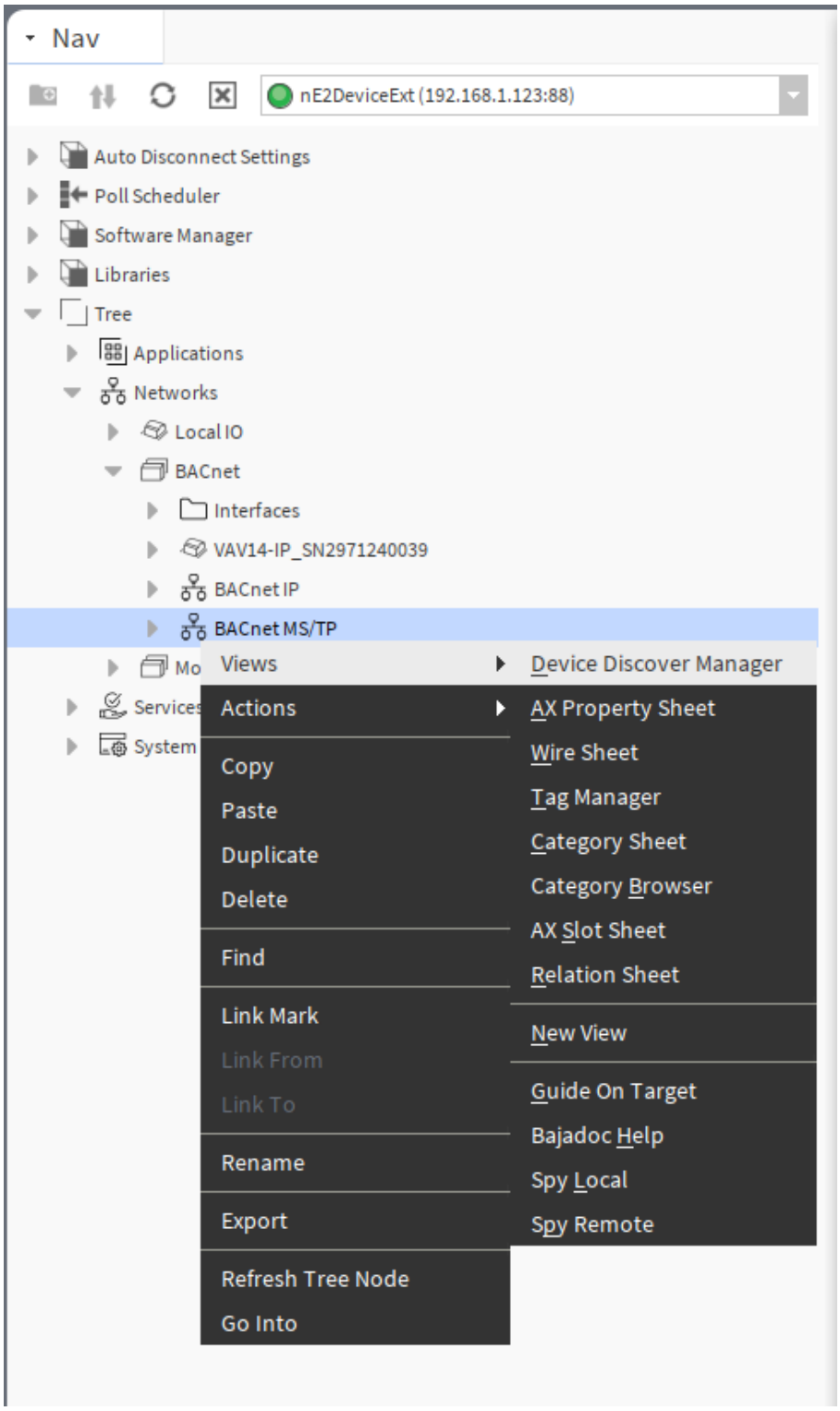


Figure 39. Accessing the BACnet Device Discover Manager

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.

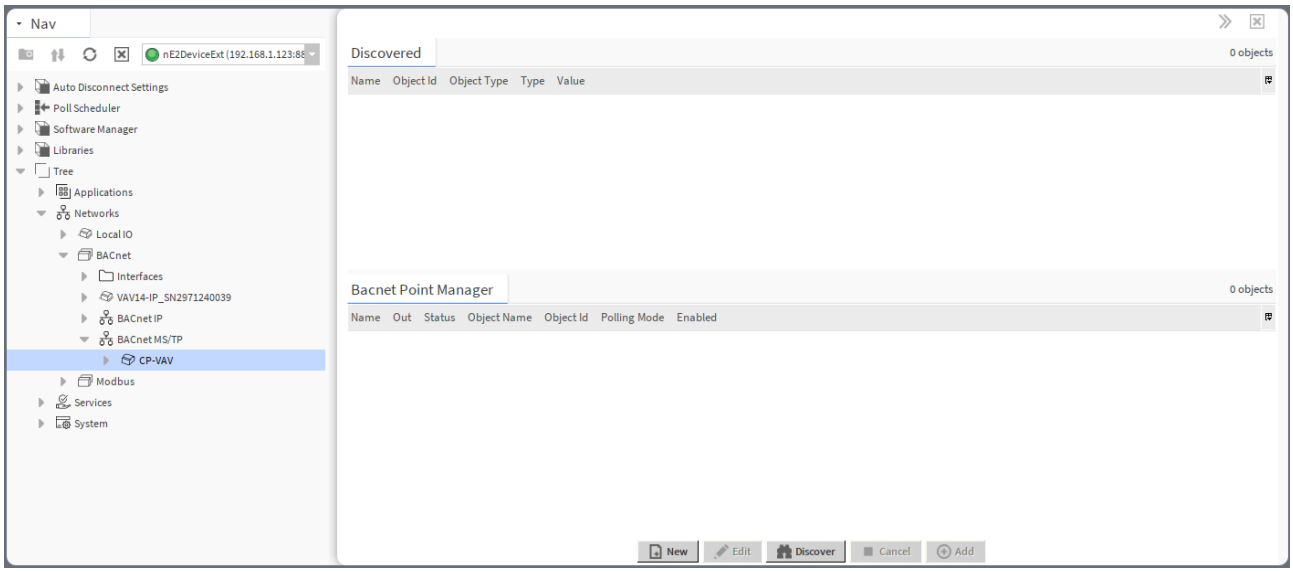


Figure 40. BACnet Point Manager

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.

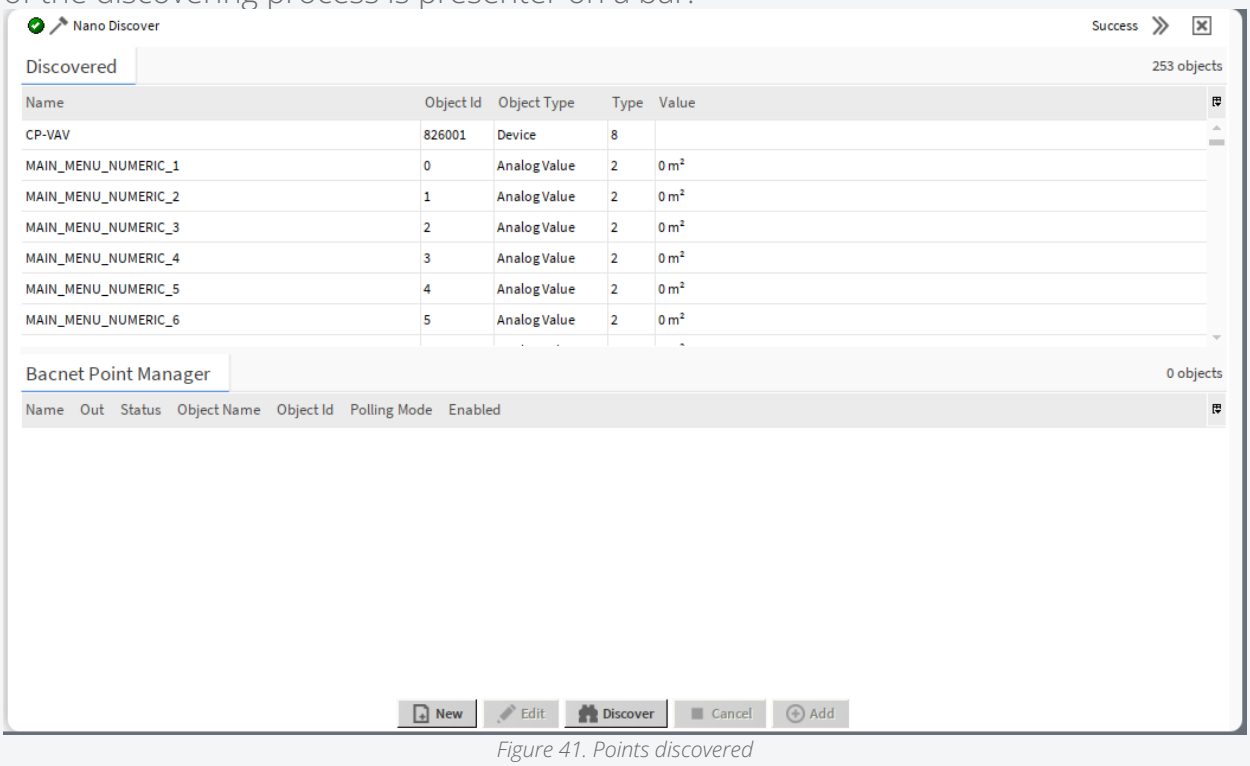


Figure 41. Points discovered

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:

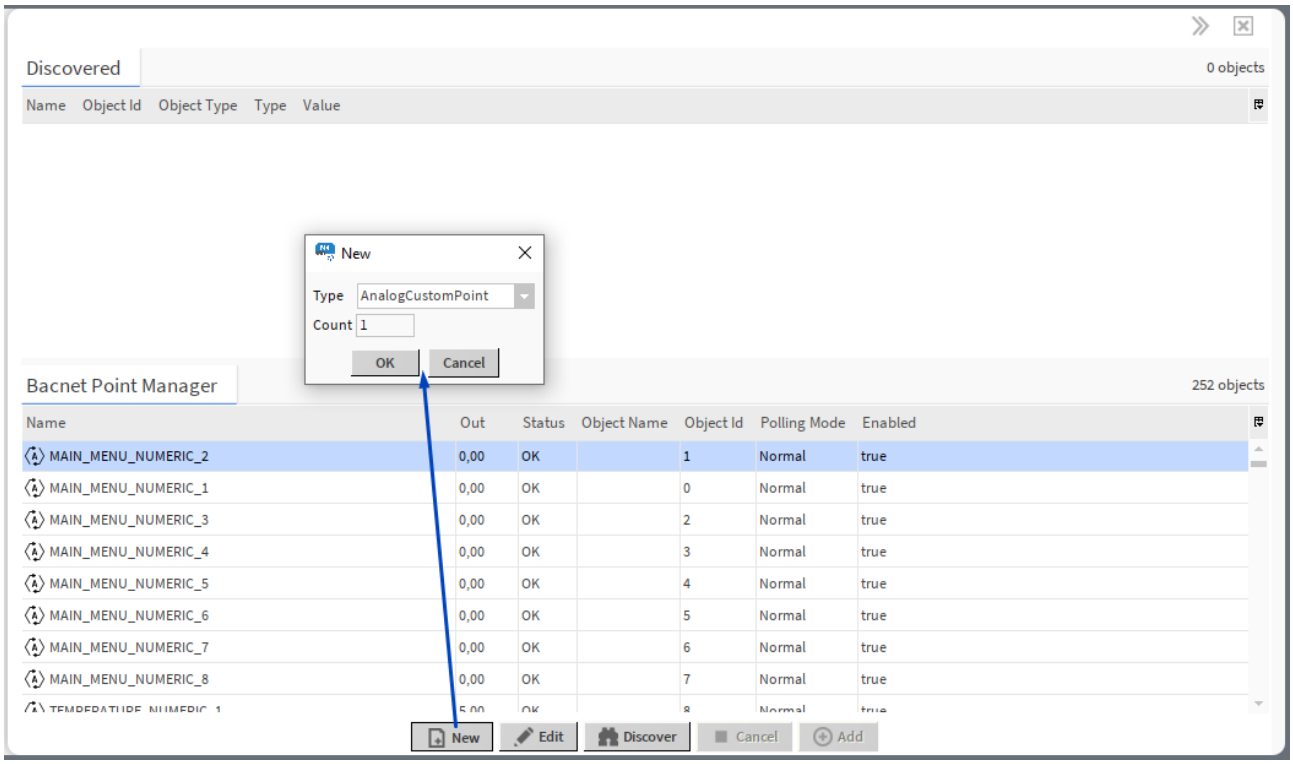


Figure 42. Adding BACnet point

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

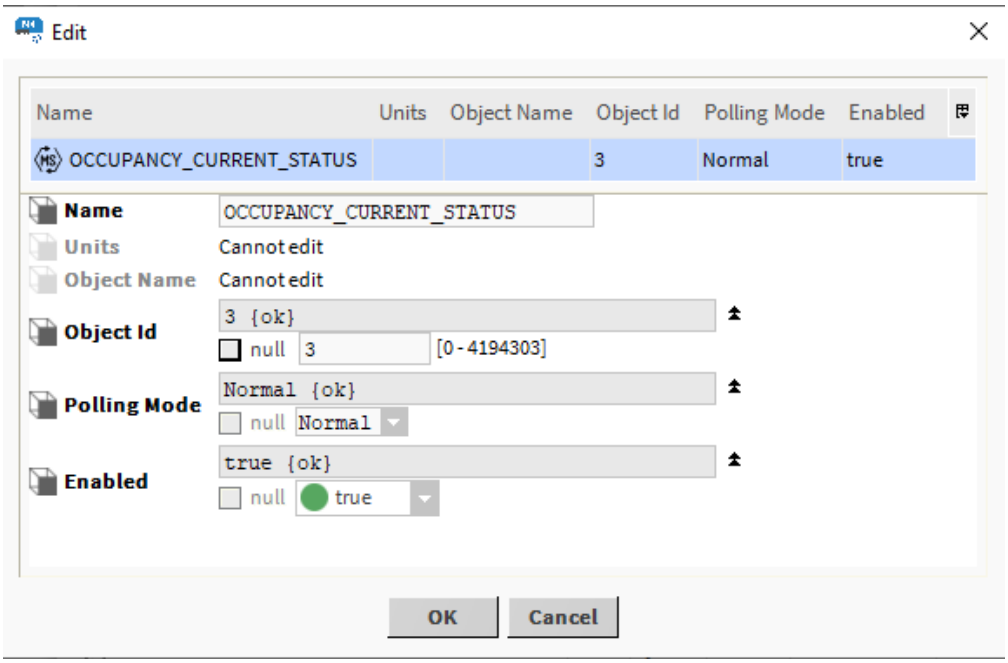


Figure 43. Editing pop-up

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.

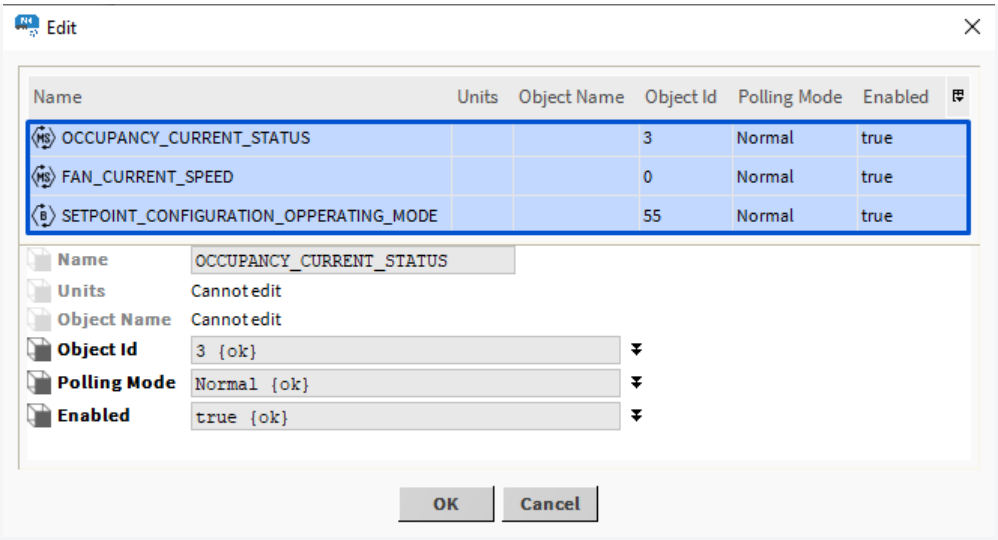


Figure 44. Editing of multiple points

- copy/paste/duplicate/delete BACnet points:

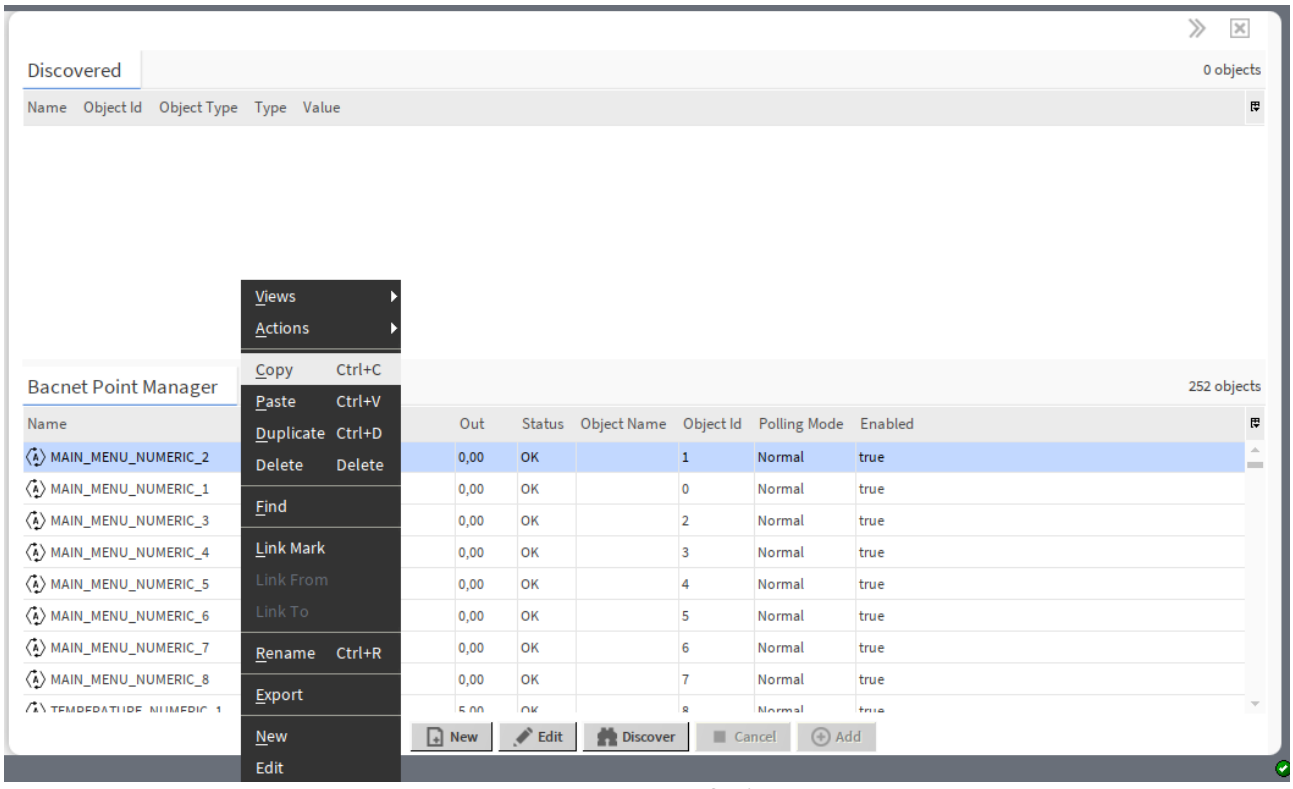


Figure 45. Context menu options for the BACnet point

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.

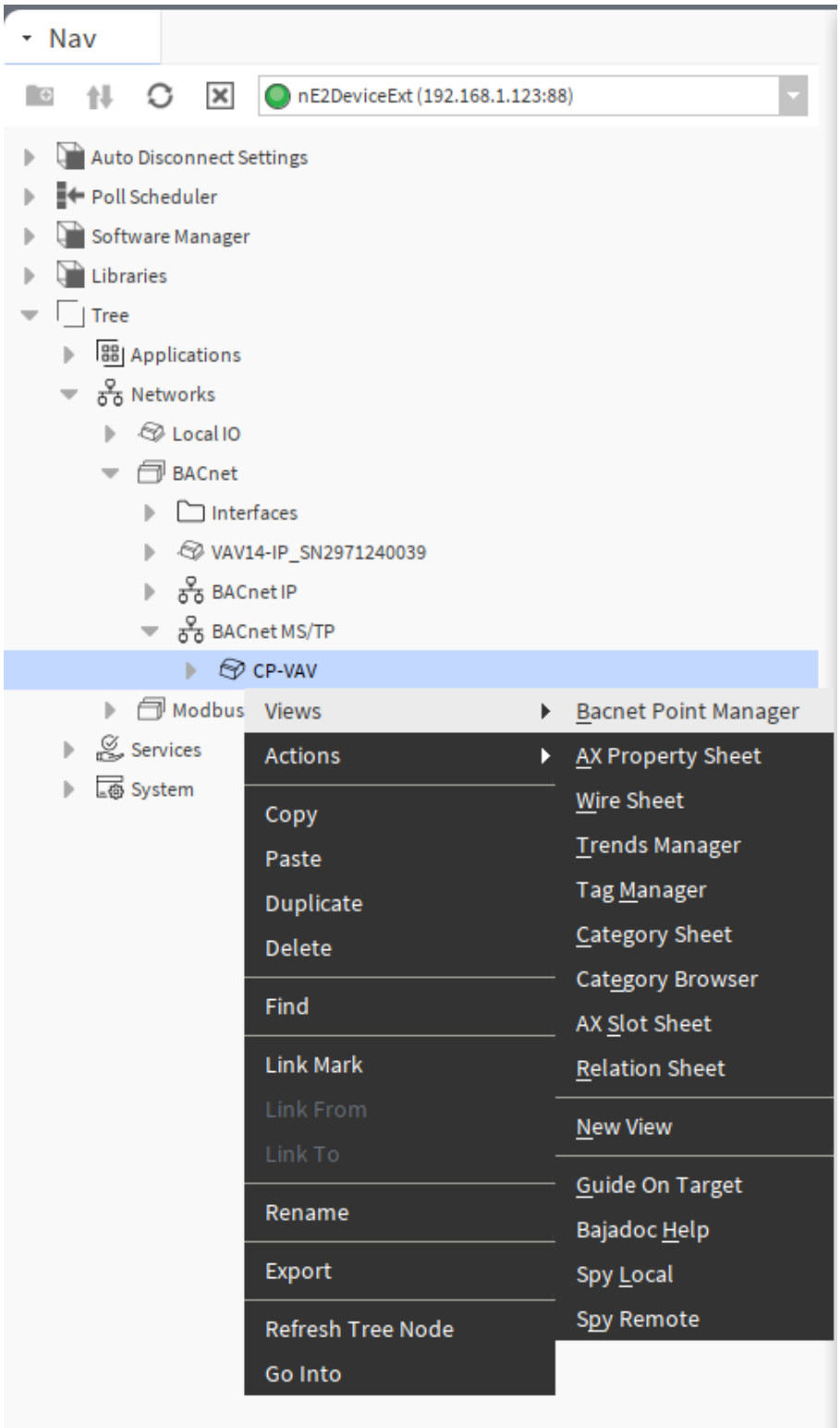


Figure 46. Accessing the BACnet Point Manager from the context menu

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.

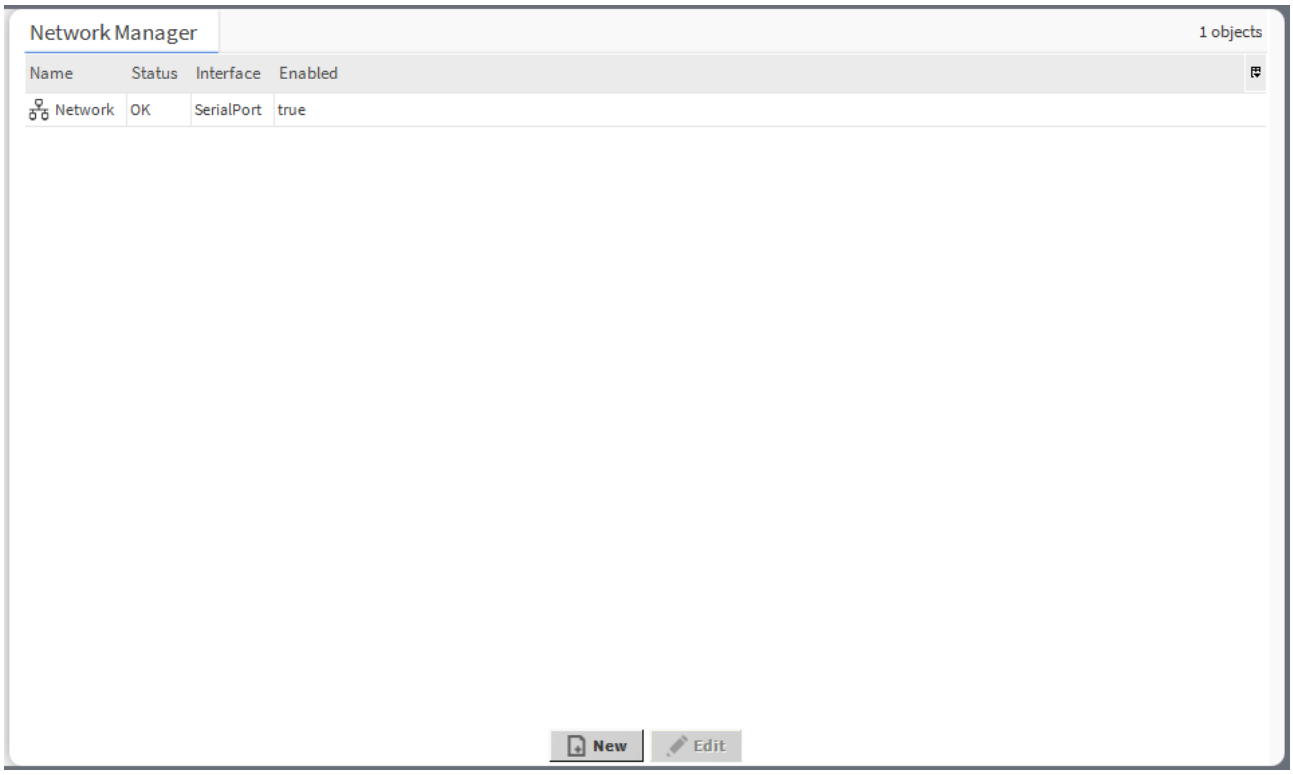


Figure 47. Modbus Network Manager

In the Modbus Network Manager, it is possible to:

- add Modbus network component:

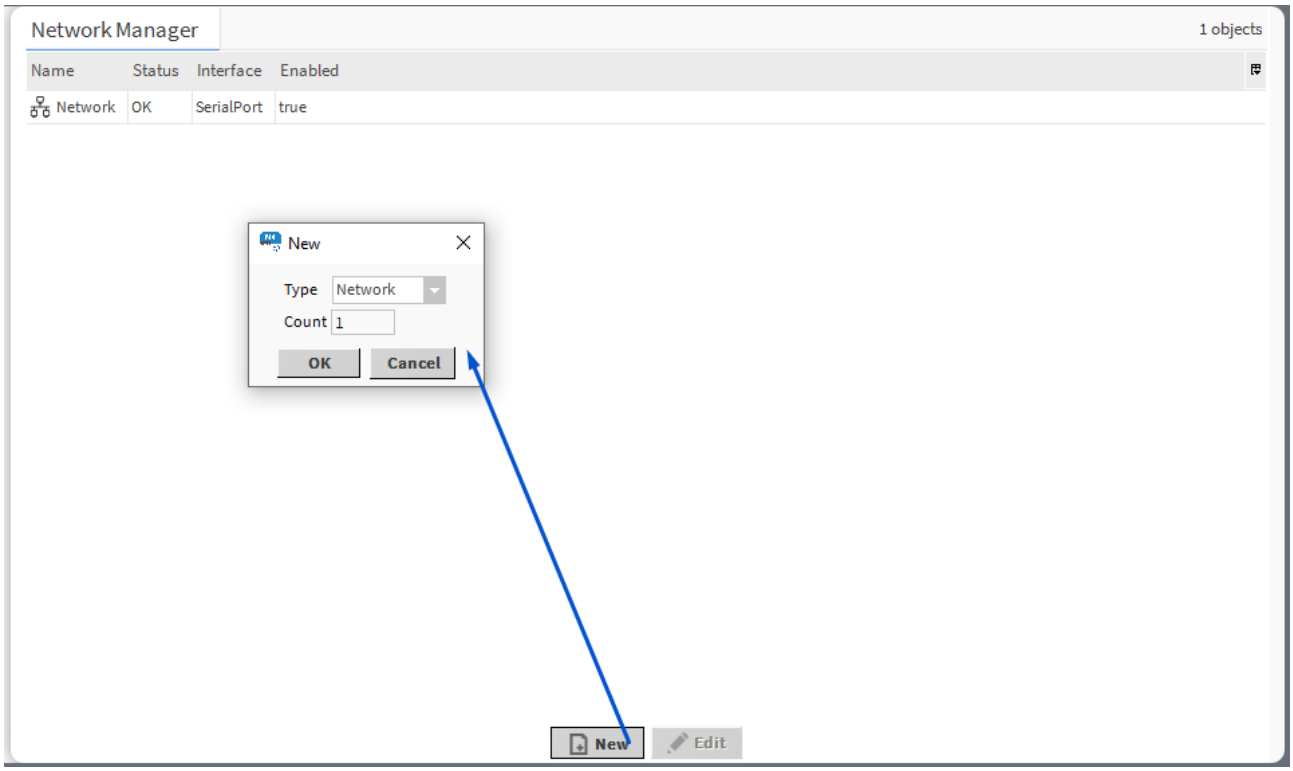


Figure 48. Adding new Modbus network

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

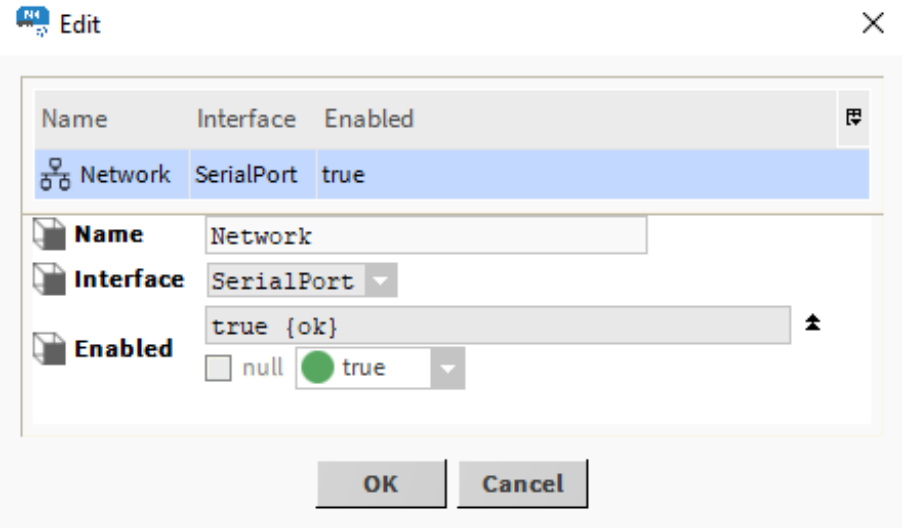


Figure 49. Editing pop-up

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.

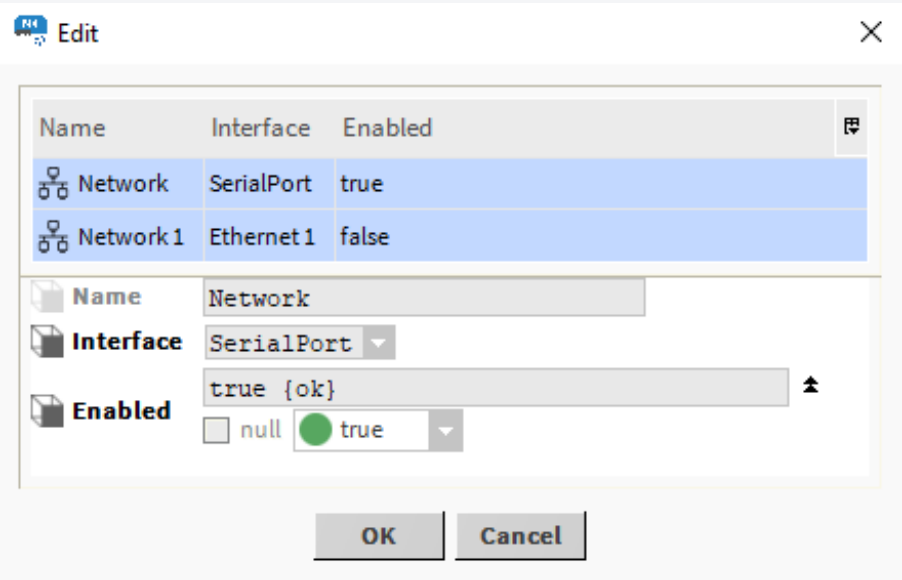


Figure 50. Editing multiple Modbus networks

- copy/duplicate/remove Modbus network components:

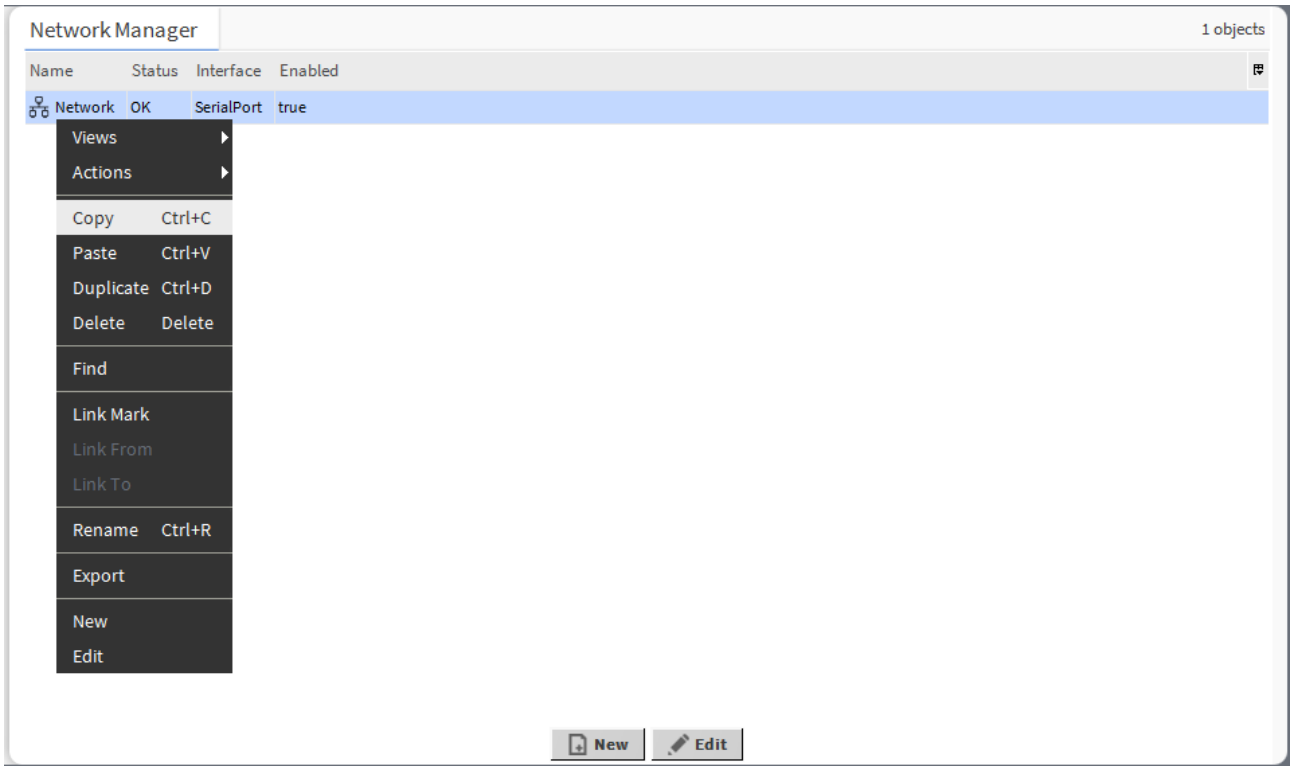


Figure 51. Context menu options for Modbus network

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.

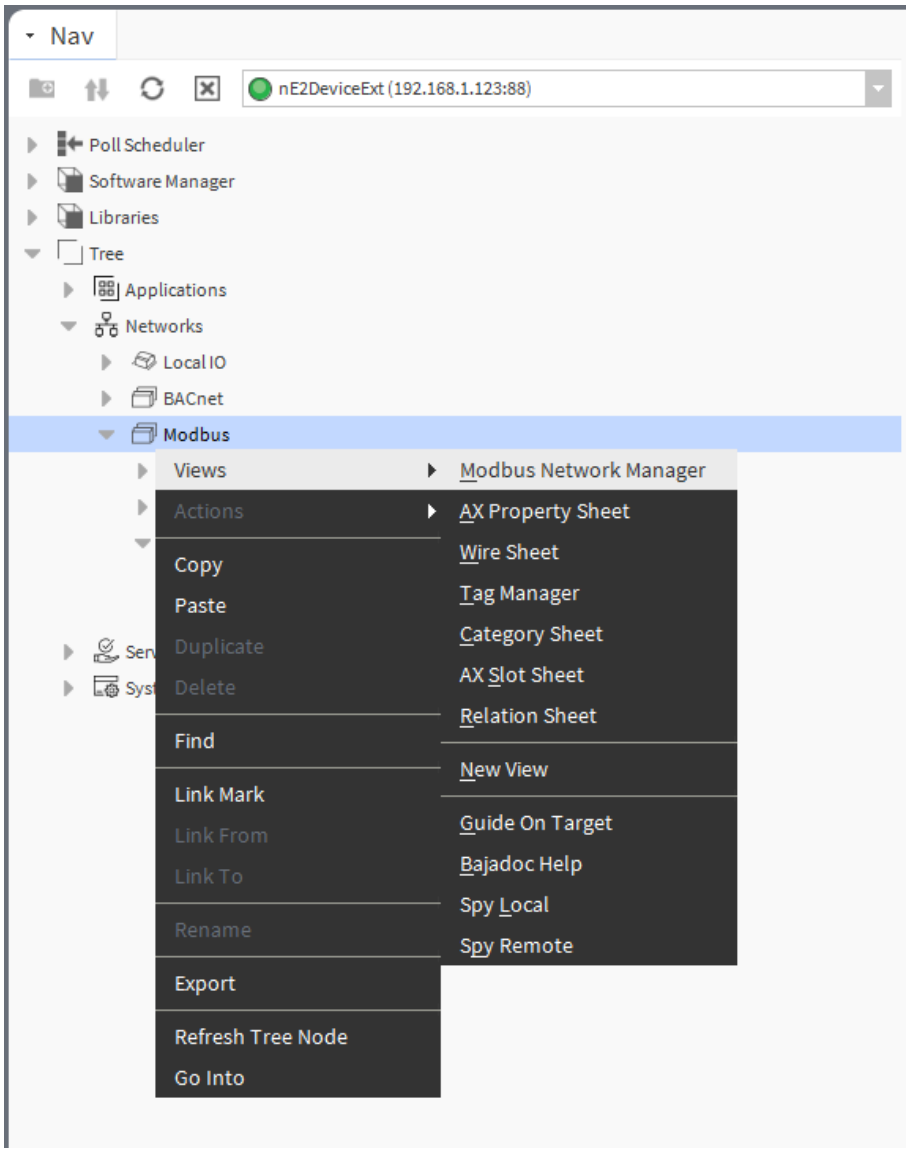


Figure 52. Accessing the Modbus Network Manager

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).

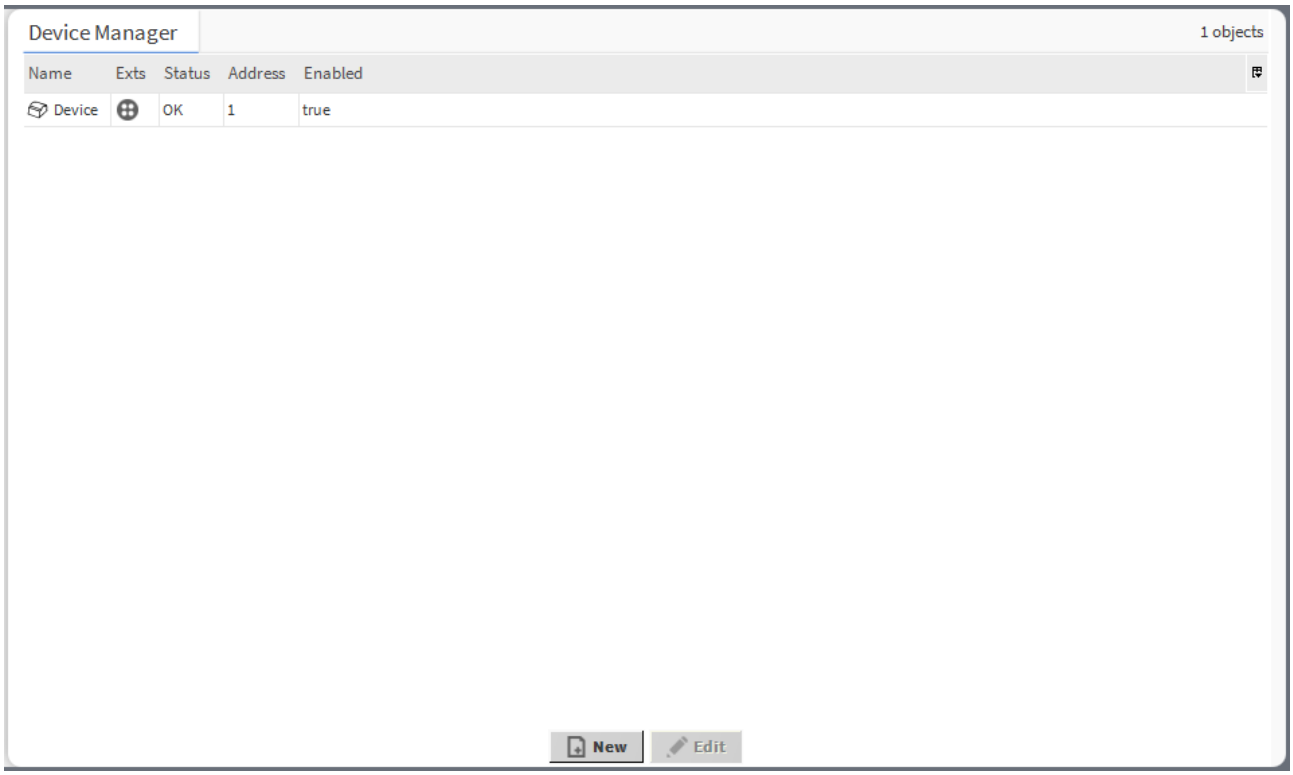


Figure 53. Modbus Device Manager

In the Modbus Device Manager, it is possible to:

- add Modbus Device component:

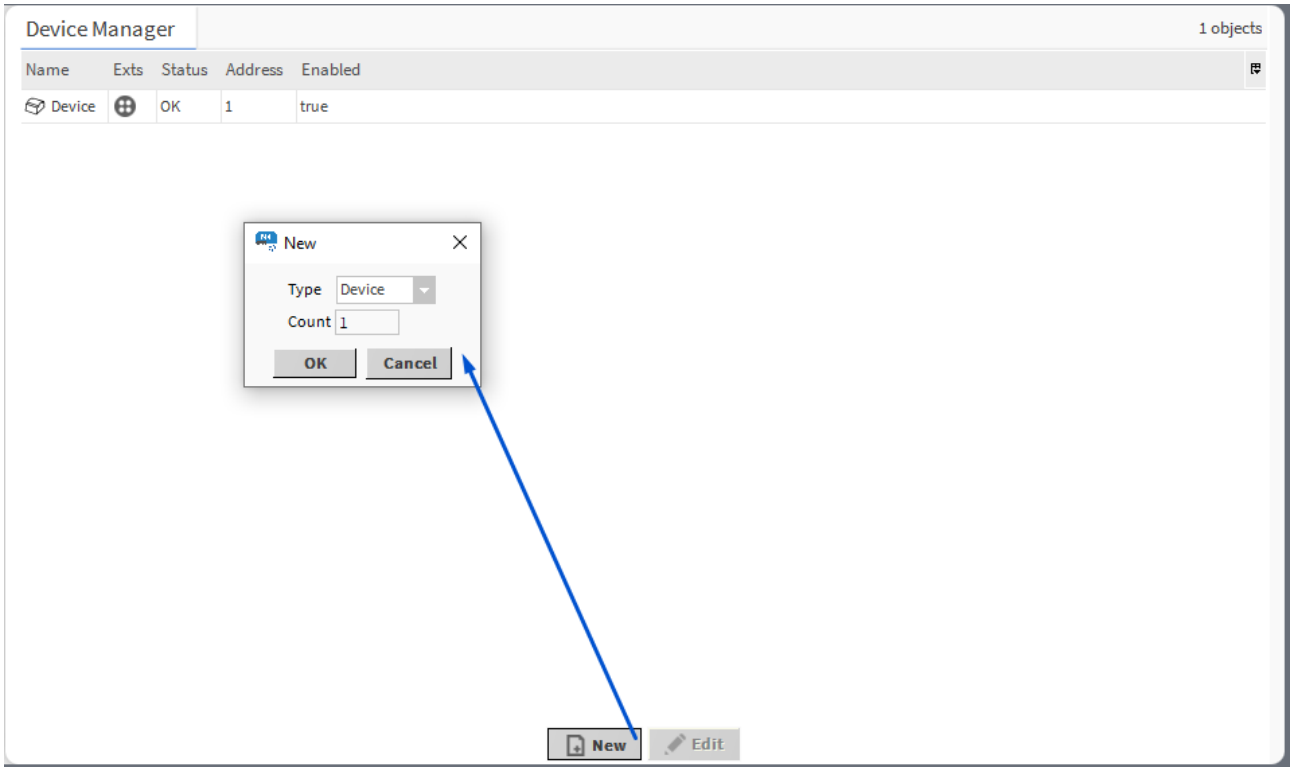


Figure 54. Adding new Modbus device

- edit the Modbus device's name:

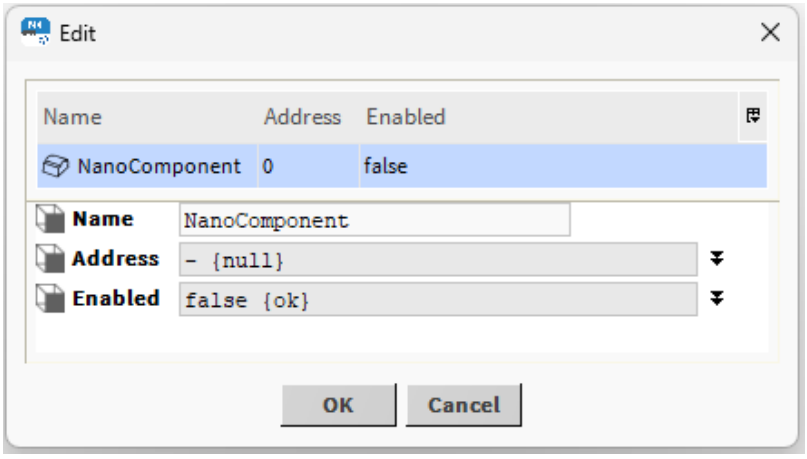


Figure 55. Editing pop-up

- copy/duplicate/remove Modbus device components:

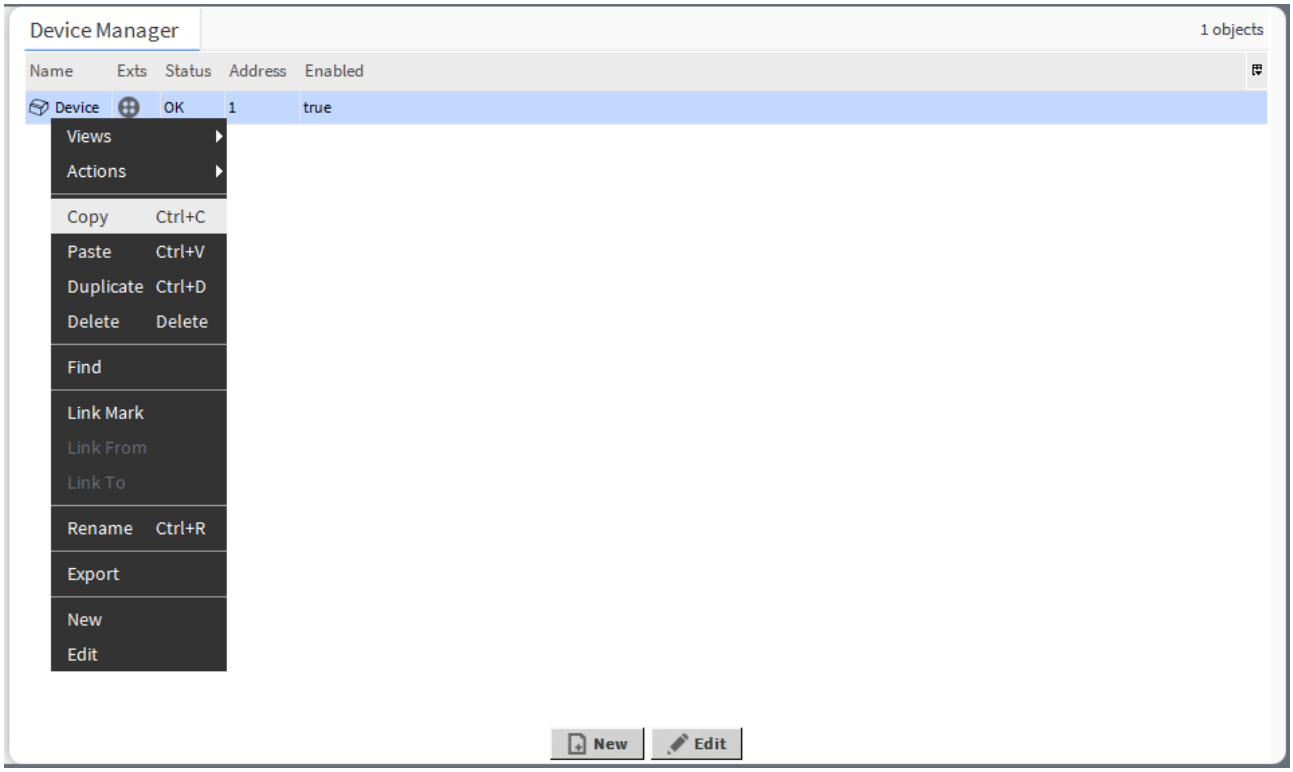


Figure 56. Context menu options for Modbus device

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.

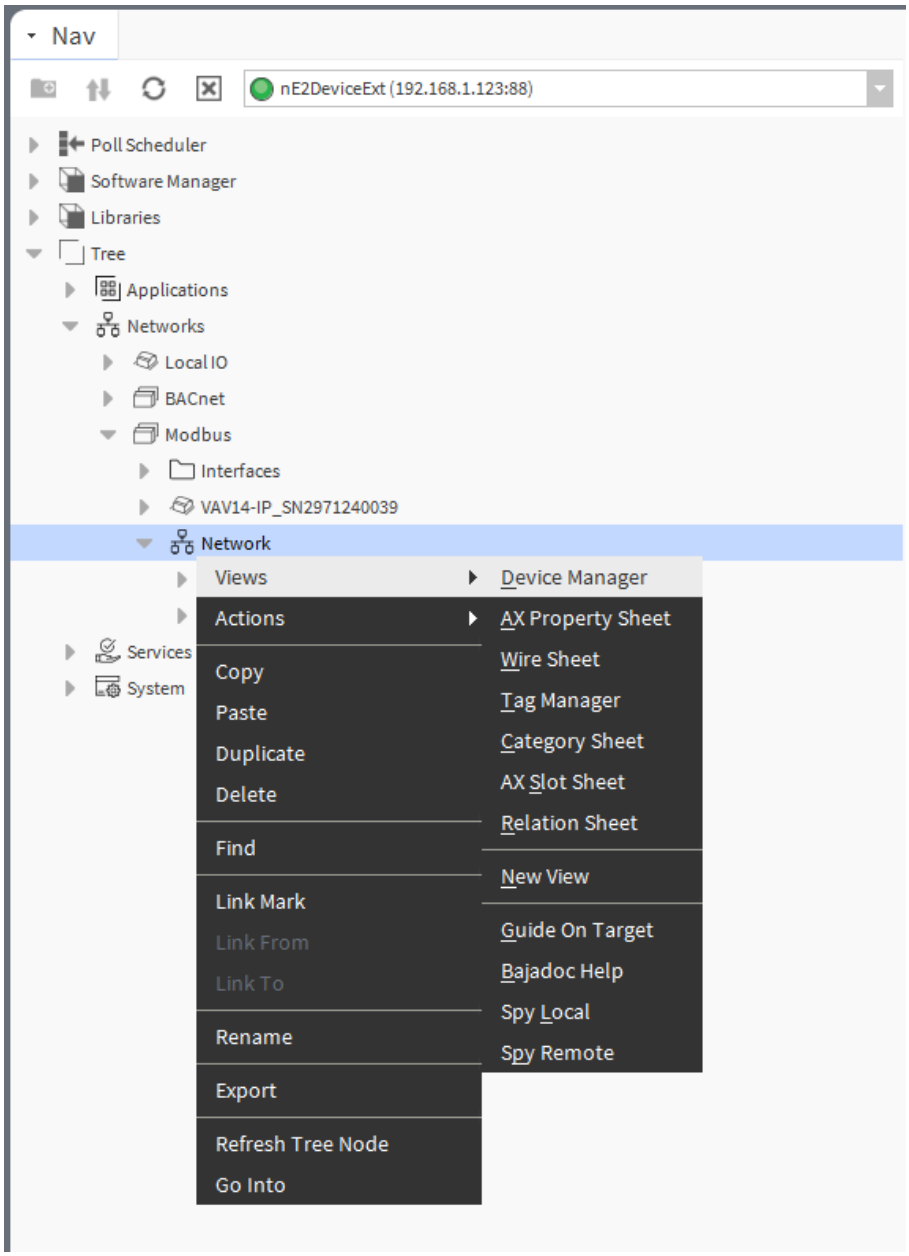


Figure 57. Accessing the Modbus Device Manager

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.

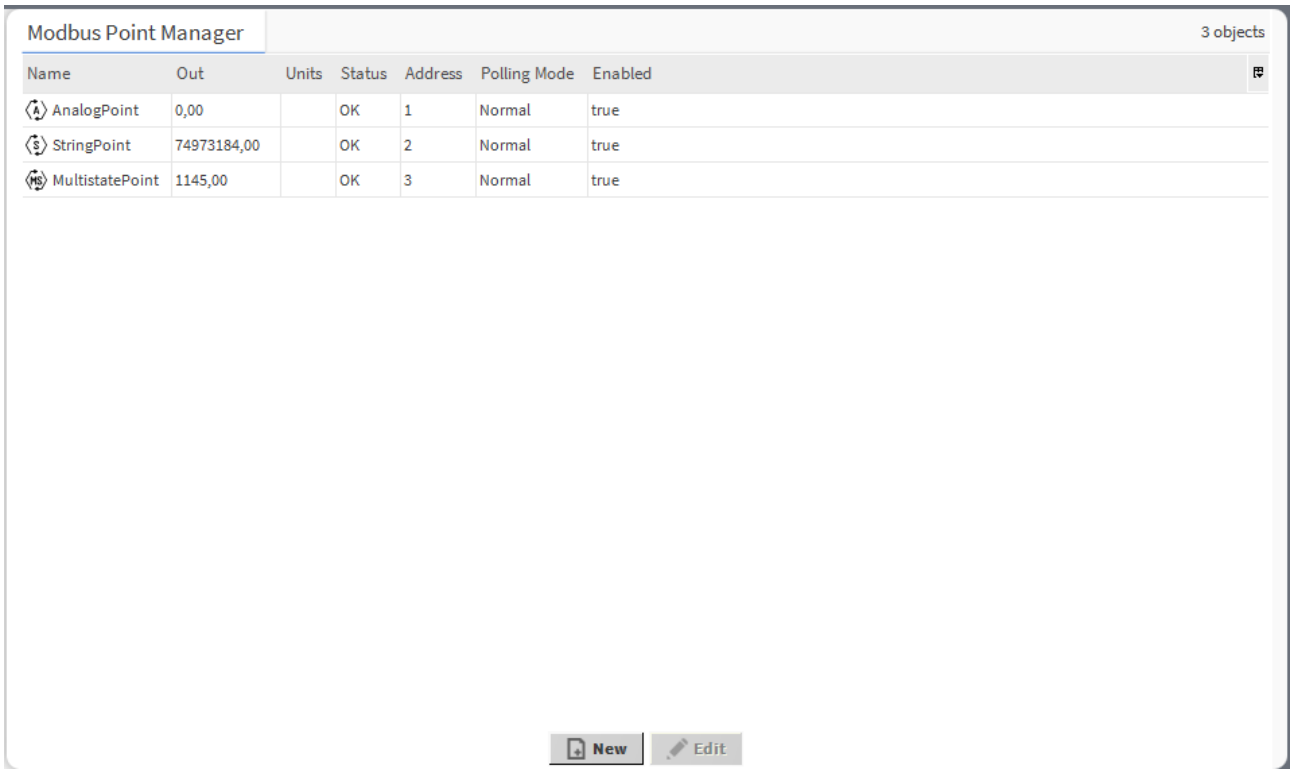


Figure 58. Modbus Point Manager

In the Modbus Point Manager, it is possible to:

- add Modbus points:

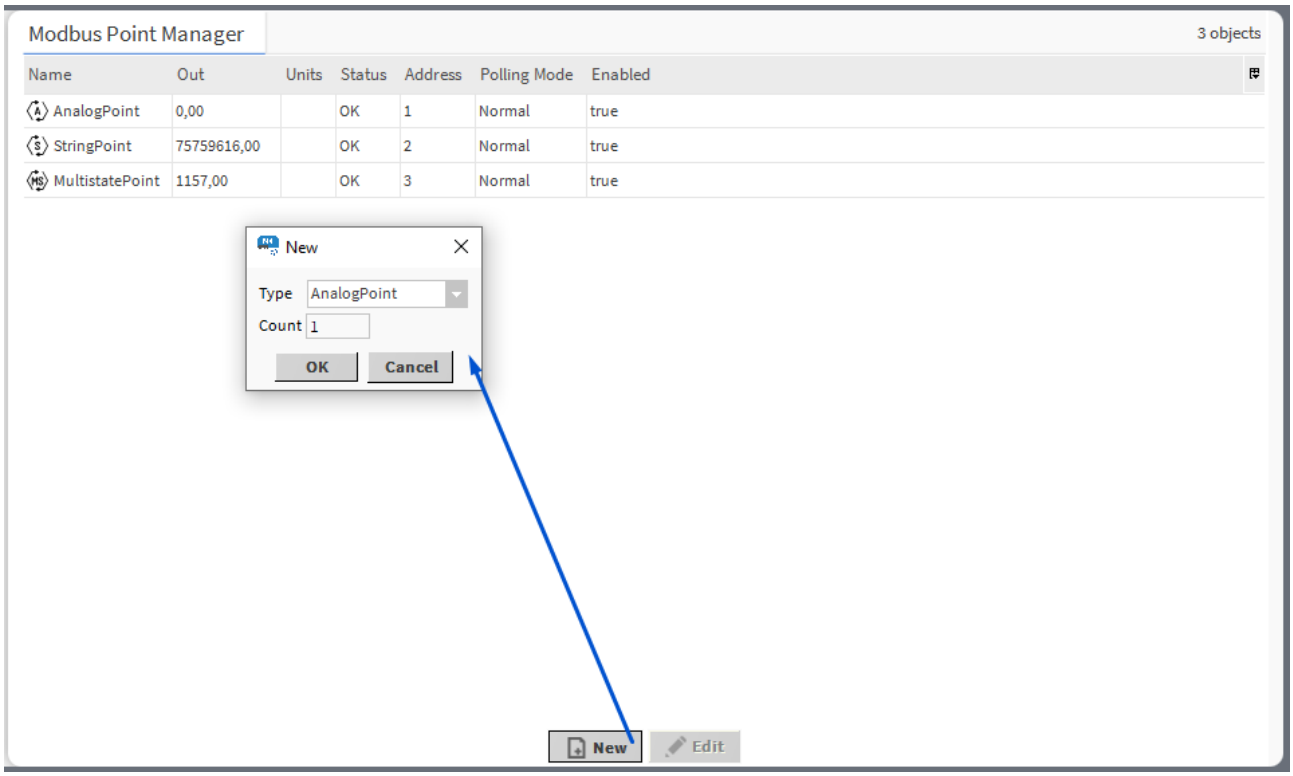


Figure 59. Adding new Modbus point

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

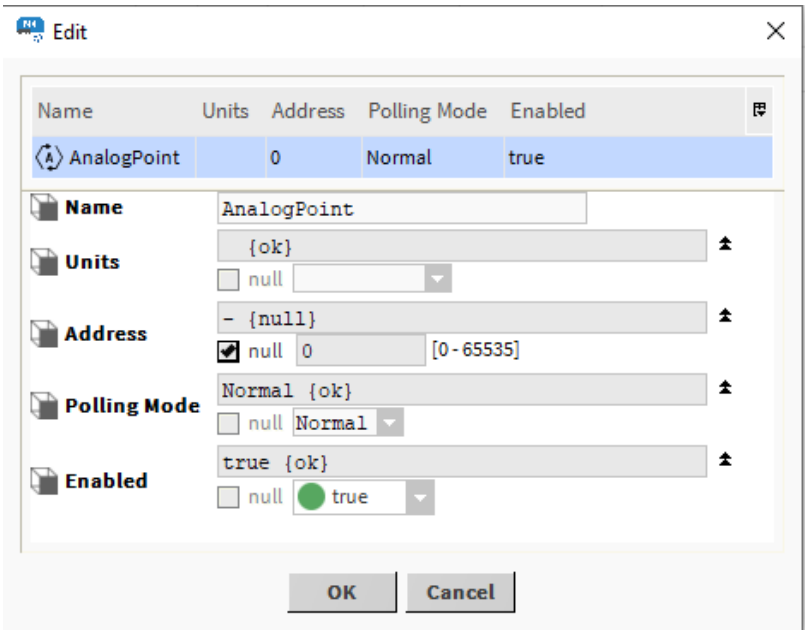


Figure 60. Editing pop-up

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.

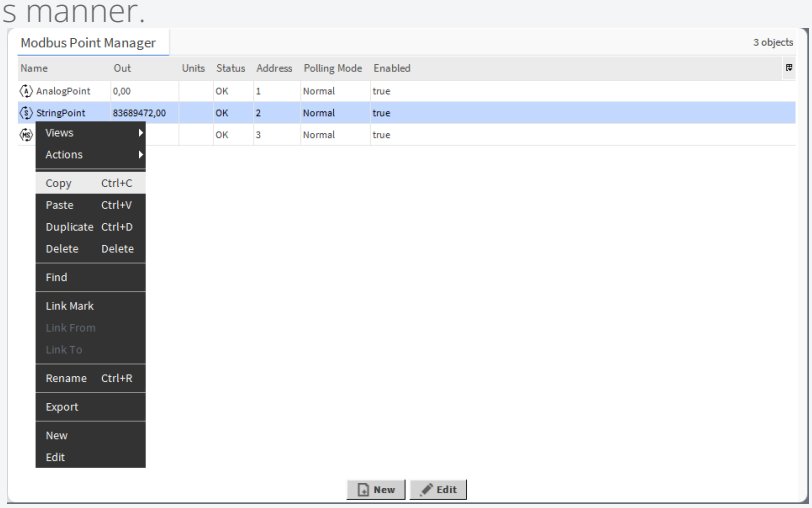


Figure 61. Editing multiple Modbus points

- copy/duplicate/remove Modbus points:

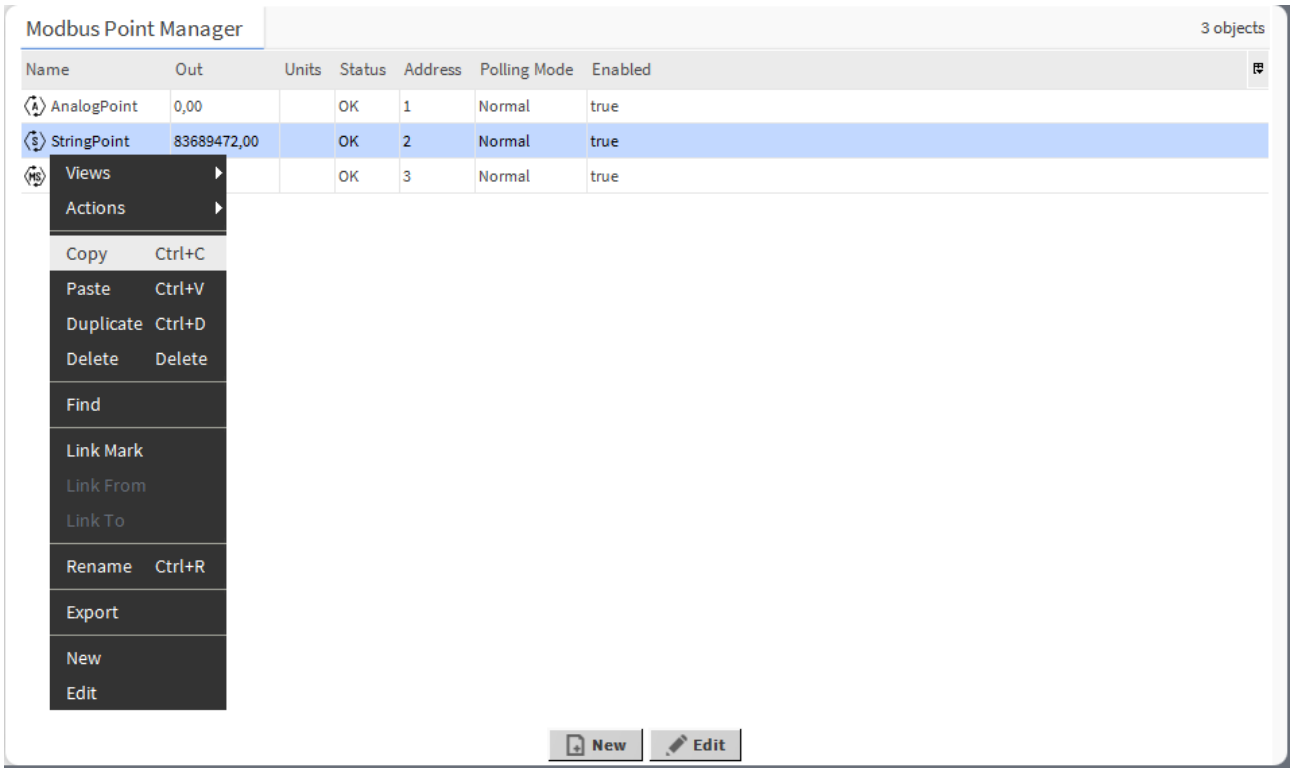


Figure 62. Context menu for Modbus points

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.

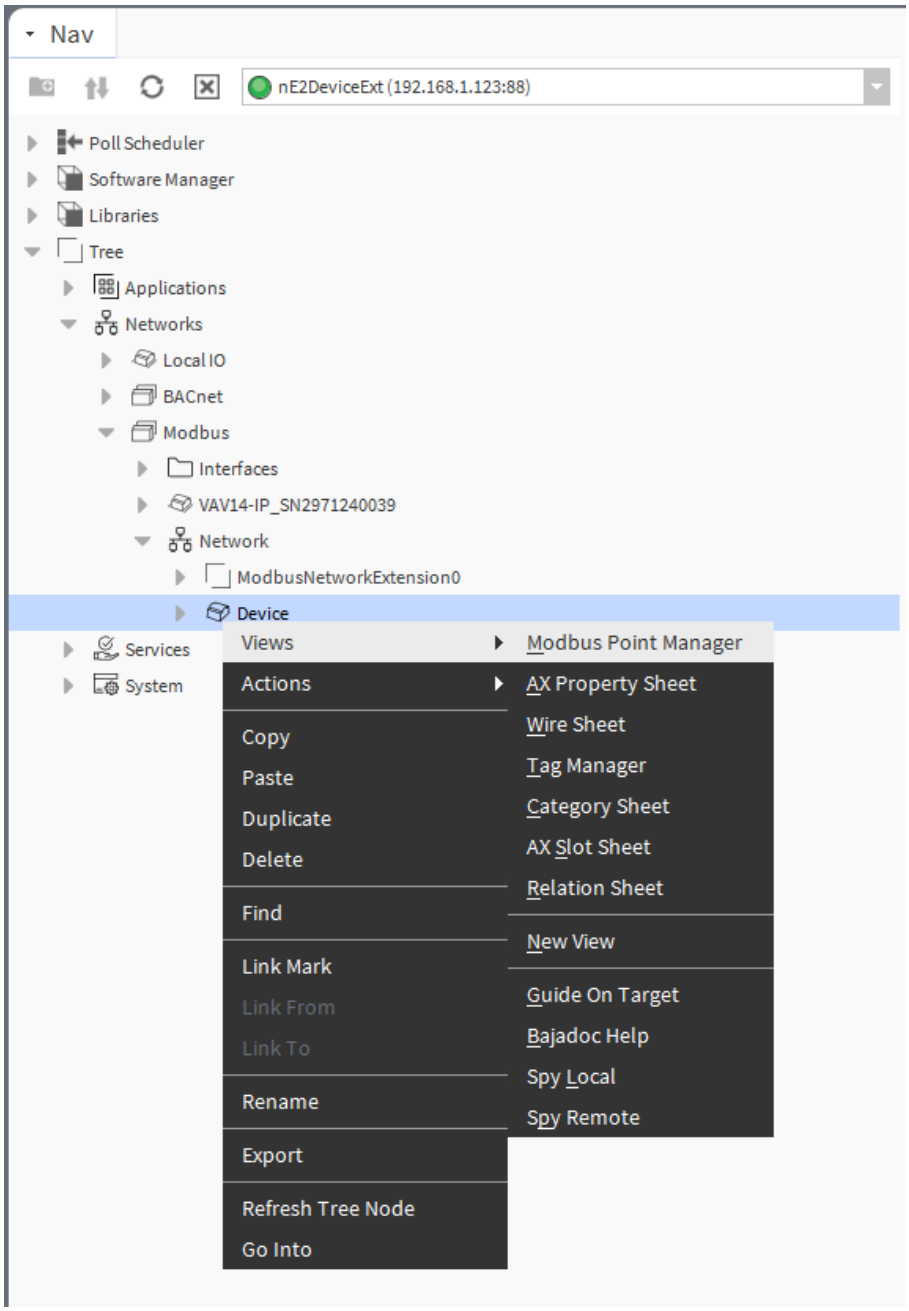


Figure 63. Accessing the Modbus Point Manager from the context menu

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.

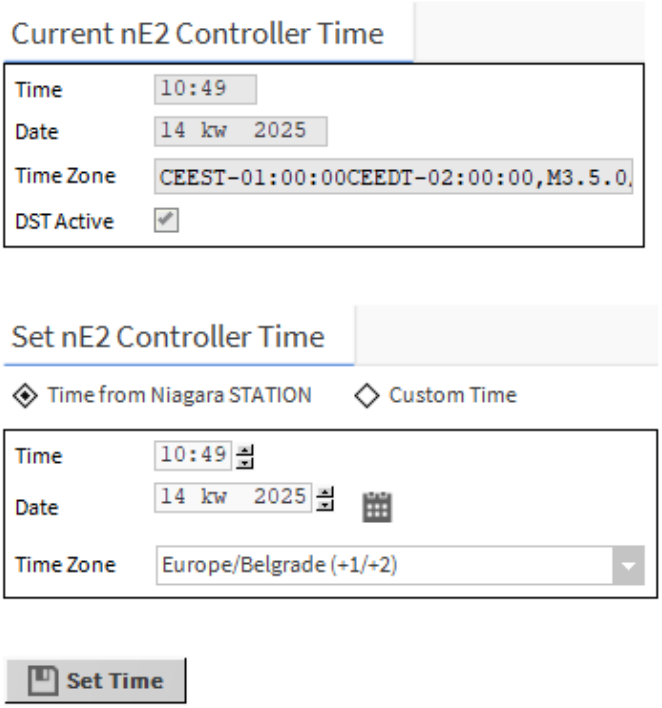


Figure 64. Time settings

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;

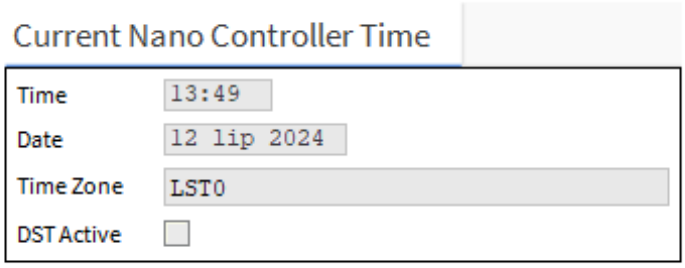
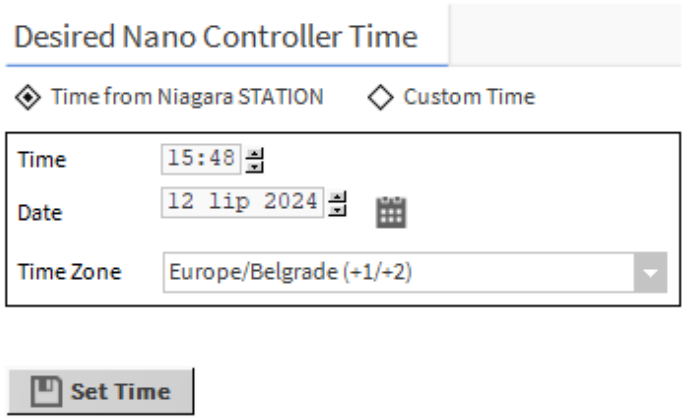


Figure 65. Current nano EDGE ENGINE device time

- 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.



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;

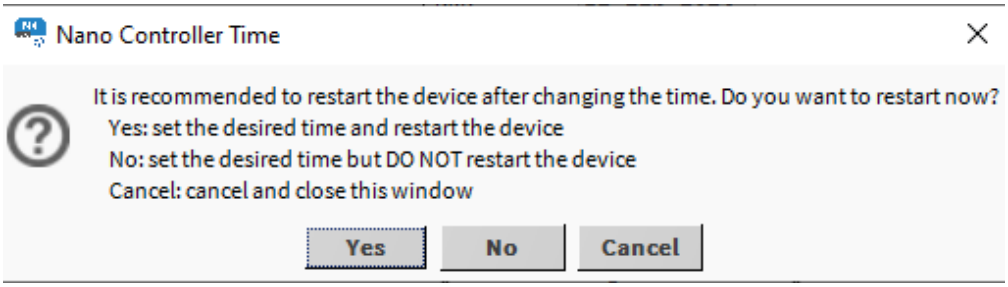


Figure 66. Set the time from the station dialog window

- 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.

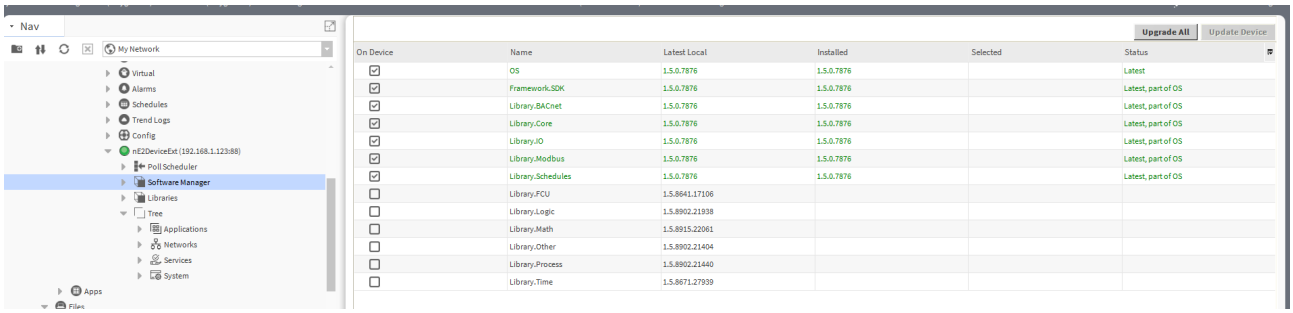


Figure 67. Software Manager view

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.

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 dropdown 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.

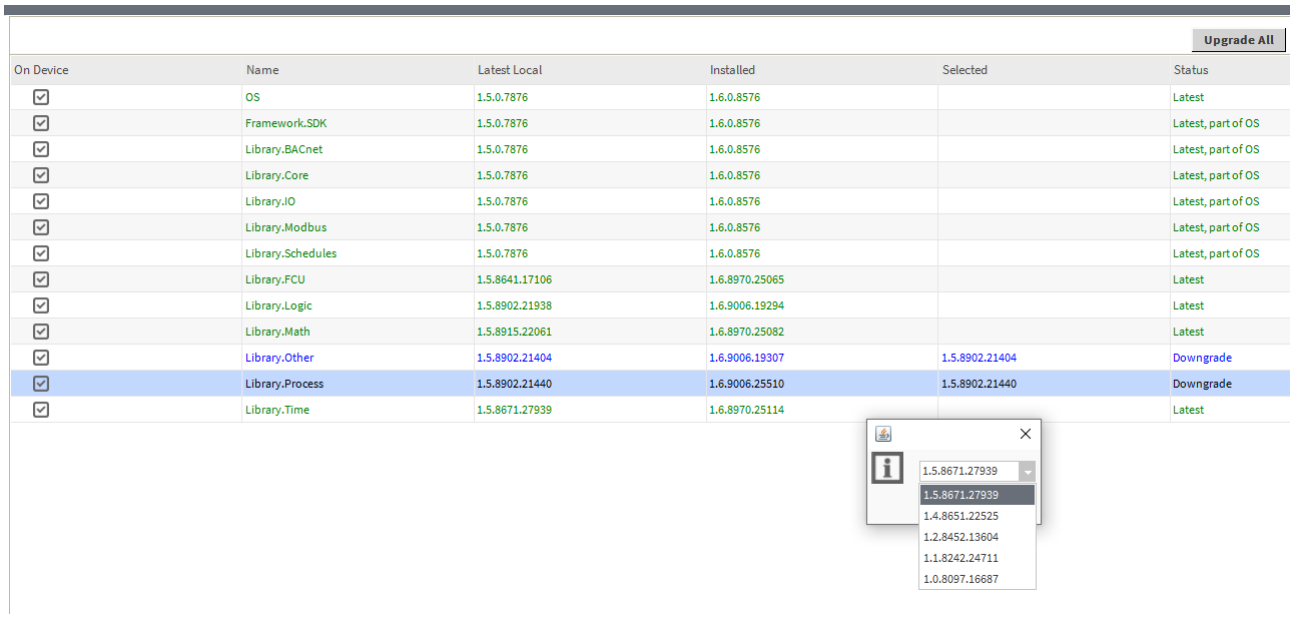


Figure 68. Selecting a library's version

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.

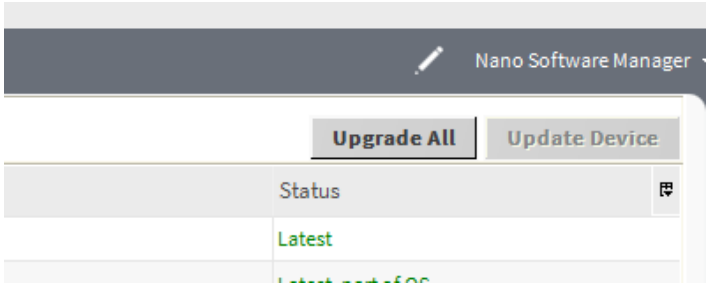


Figure 69. Upgrade All button

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');

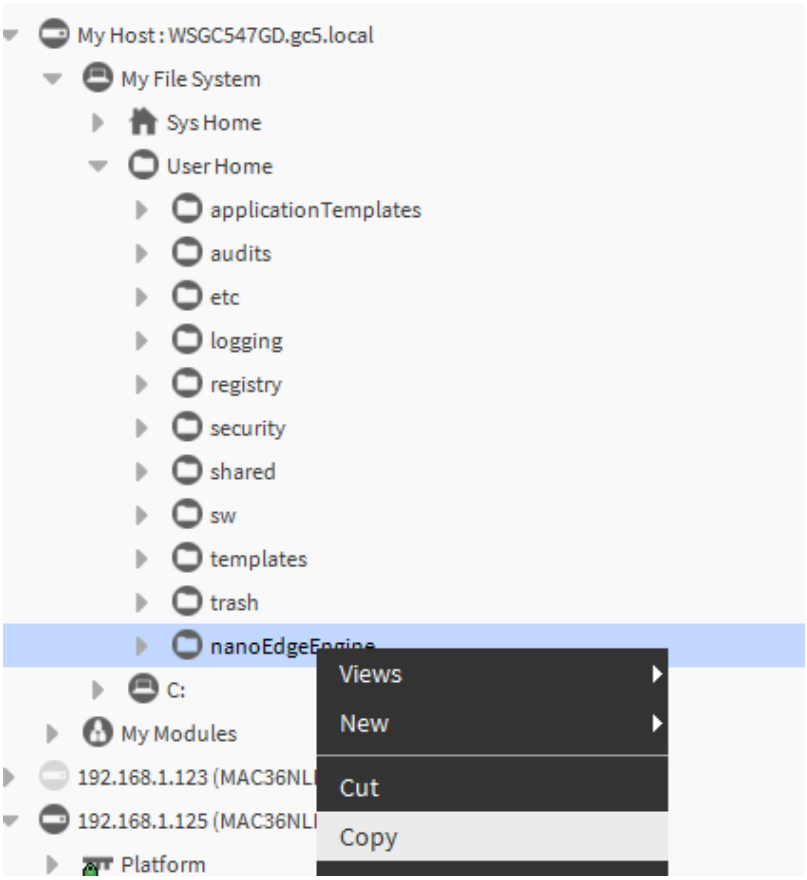


Figure 70. Copying the nanoEdgeEngine folder

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

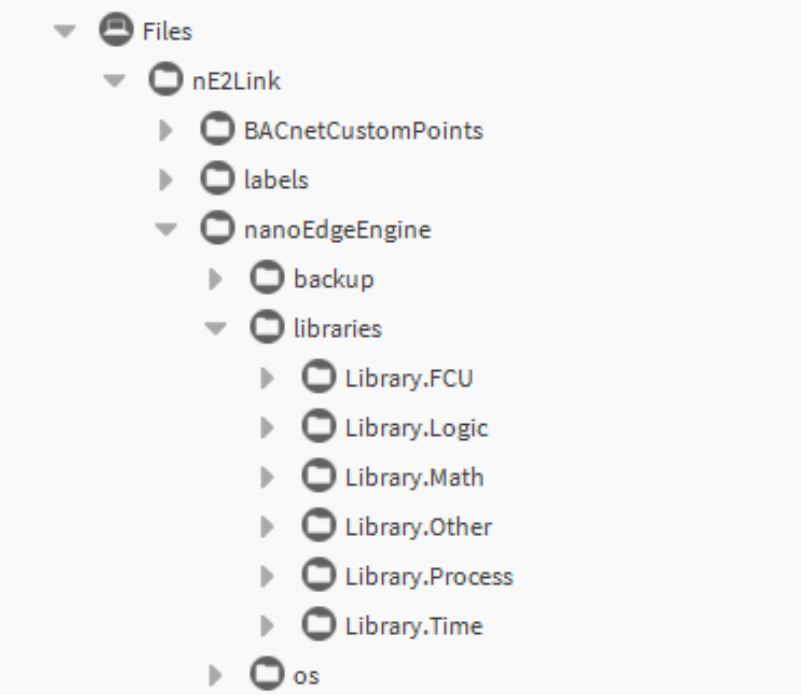


Figure 71. Pasted folder

- 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.

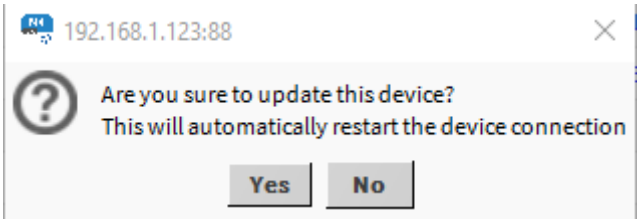


Figure 72. Update device prompt

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.

Upgrade All | Update Device

On Device	Name	Latest Local	Installed	Selected	Status
☑	OS	1.4.1.7340	1.4.1.7340		Latest
☑	Framework.SDK	1.4.8585.25238	1.4.8585.25238		Latest, part of OS
☑	Library.BACnet	1.4.8655.31468	1.4.8655.31468		Latest, part of OS
☑	Library.Core	1.4.8655.31015	1.4.8655.31015		Latest, part of OS
☑	Library.IO	1.4.8655.31292	1.4.8655.31292		Latest, part of OS
☑	Library.Modbus	1.4.8704.26958	1.4.8704.26958		Latest, part of OS
☑	Library.Schedules	1.4.8655.31332	1.4.8655.31332		Latest, part of OS
☑	Library.FCU	1.4.8585.25488	1.4.8585.25488		Latest
☑	Library.Logic	1.4.8585.25337	1.4.8585.25337		Latest
☑	Library.Math	1.4.8621.25370	1.4.8621.25370		Latest
☑	Library.Other	1.4.8641.19349	1.4.8641.19349		Latest
☑	Library.Process	1.4.8585.25565	1.4.8585.25565		Latest
☑	Library.Time	1.4.8651.22525	1.4.8651.22525		Latest

Figure 73. 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.

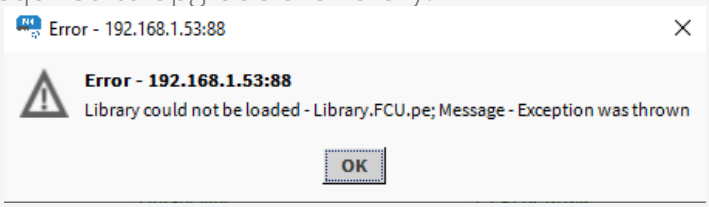
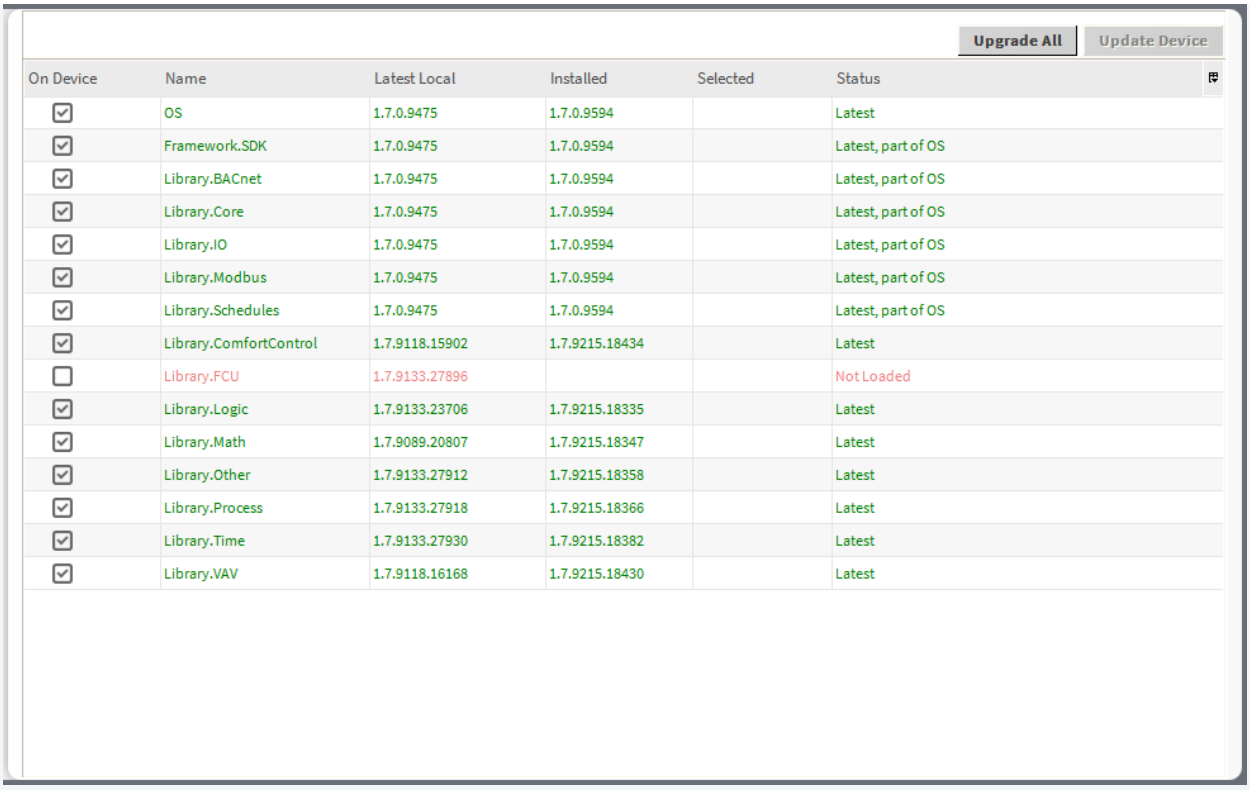


Figure 74. Error prompt



On Device	Name	Latest Local	Installed	Selected	Status
☑	OS	1.7.0.9475	1.7.0.9594		Latest
☑	Framework.SDK	1.7.0.9475	1.7.0.9594		Latest, part of OS
☑	Library.BACnet	1.7.0.9475	1.7.0.9594		Latest, part of OS
☑	Library.Core	1.7.0.9475	1.7.0.9594		Latest, part of OS
☑	Library.IO	1.7.0.9475	1.7.0.9594		Latest, part of OS
☑	Library.Modbus	1.7.0.9475	1.7.0.9594		Latest, part of OS
☑	Library.Schedules	1.7.0.9475	1.7.0.9594		Latest, part of OS
☑	Library.ComfortControl	1.7.9118.15902	1.7.9215.18434		Latest
☐	Library.FCU	1.7.9133.27896			Not Loaded
☑	Library.Logic	1.7.9133.23706	1.7.9215.18335		Latest
☑	Library.Math	1.7.9089.20807	1.7.9215.18347		Latest
☑	Library.Other	1.7.9133.27912	1.7.9215.18358		Latest
☑	Library.Process	1.7.9133.27918	1.7.9215.18366		Latest
☑	Library.Time	1.7.9133.27930	1.7.9215.18382		Latest
☑	Library.VAV	1.7.9118.16168	1.7.9215.18430		Latest

Figure 75. 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.

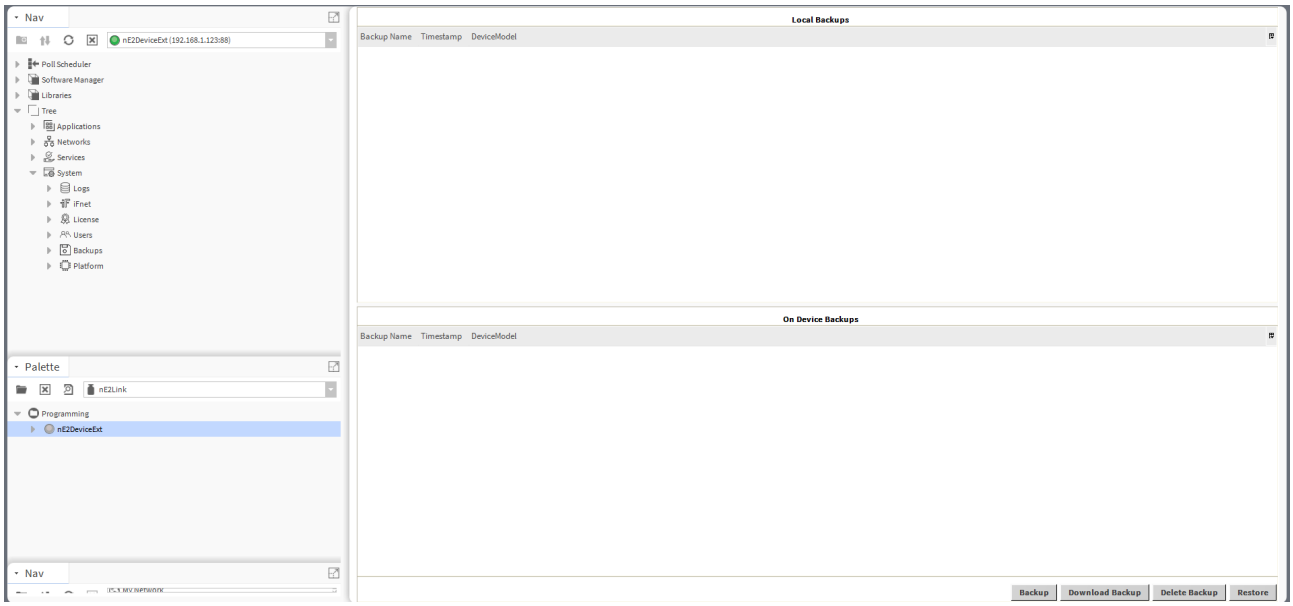


Figure 76. Backup Manager view

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.

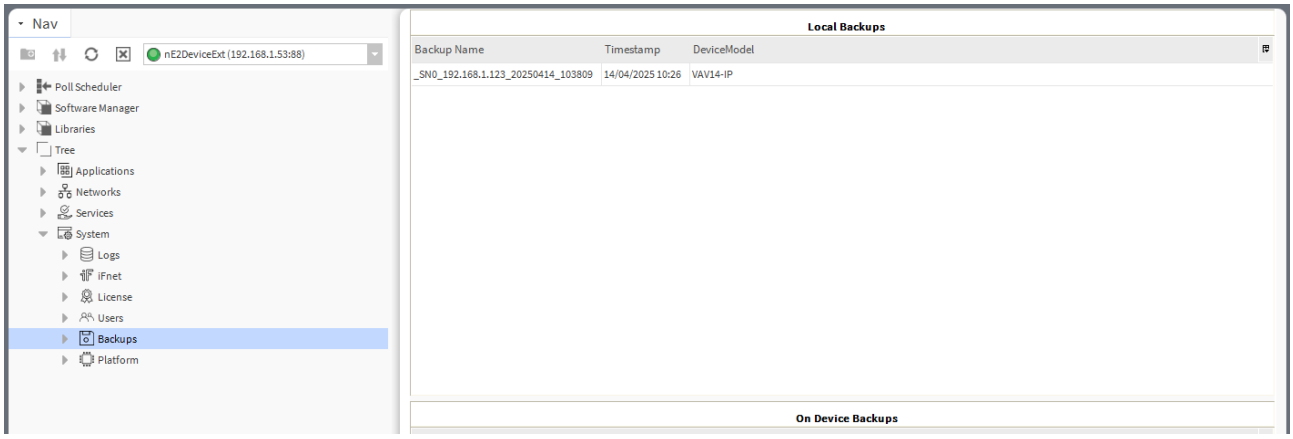


Figure 77. Local station backup stored in the station's Files

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.

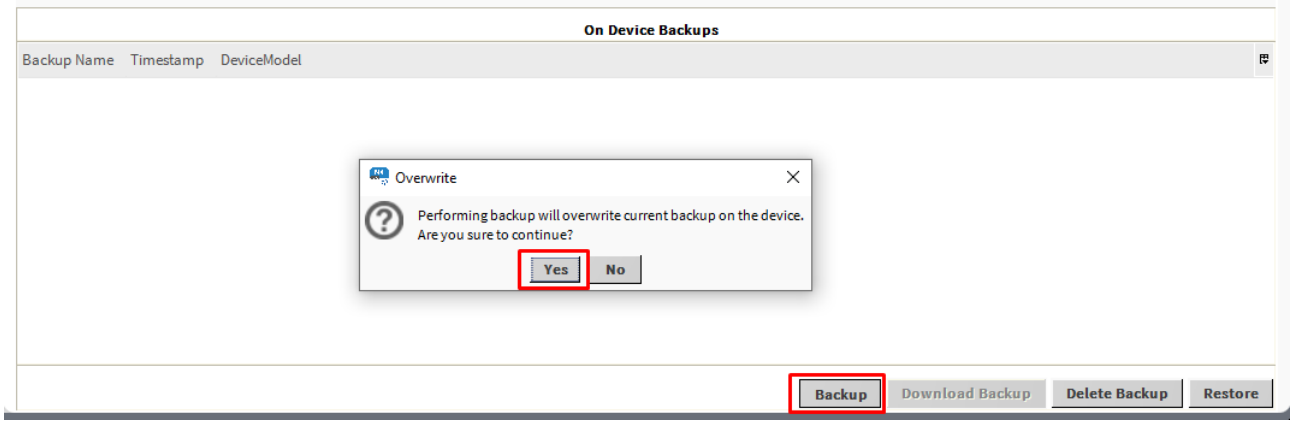


Figure 78. Pop-up informing about the risk of overwriting the existing backup on the device

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.

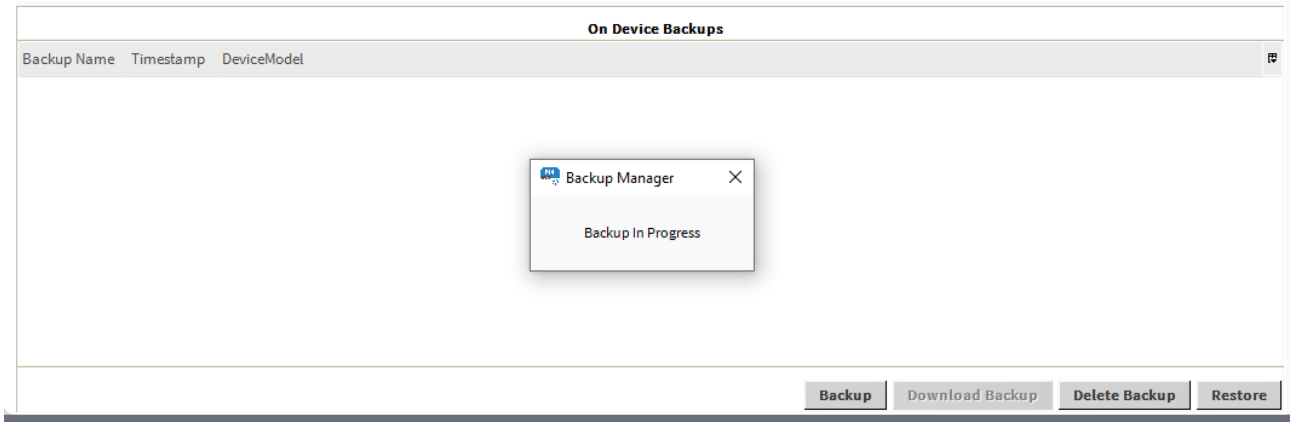


Figure 79. Pop-up informing about the backup progress

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

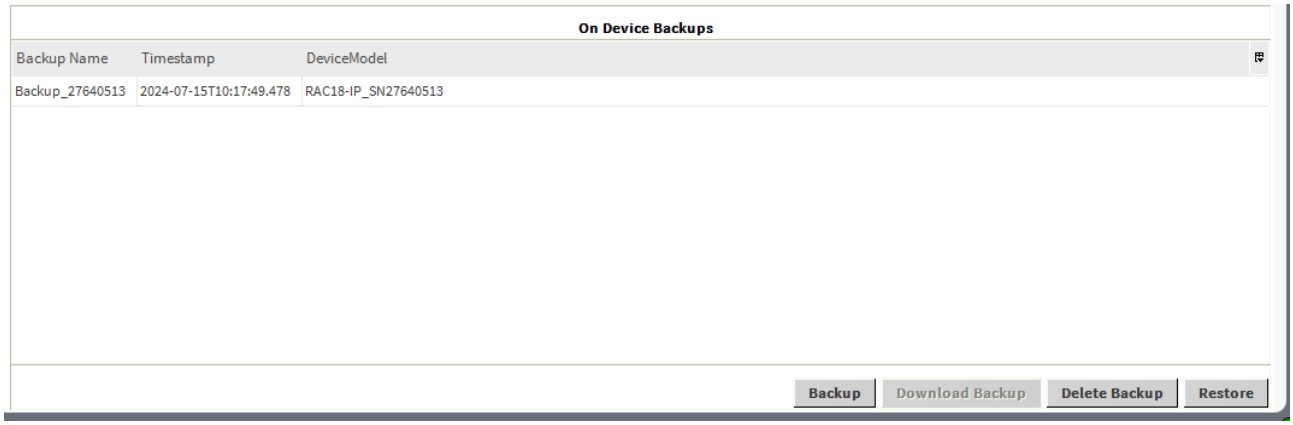


Figure 80. On Device Backups

4.7.2 Downloading Backup

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

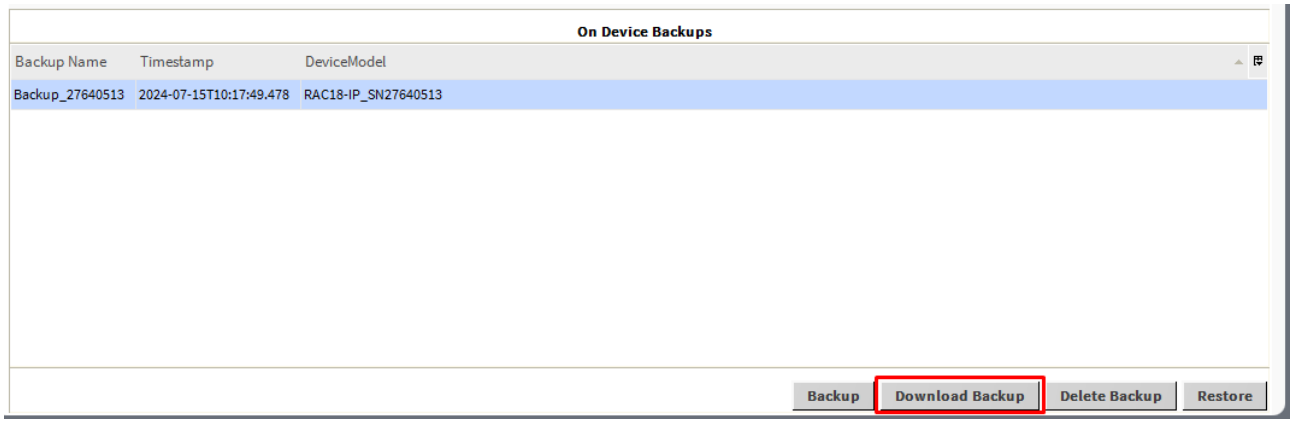


Figure 81. Download Backup button

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.

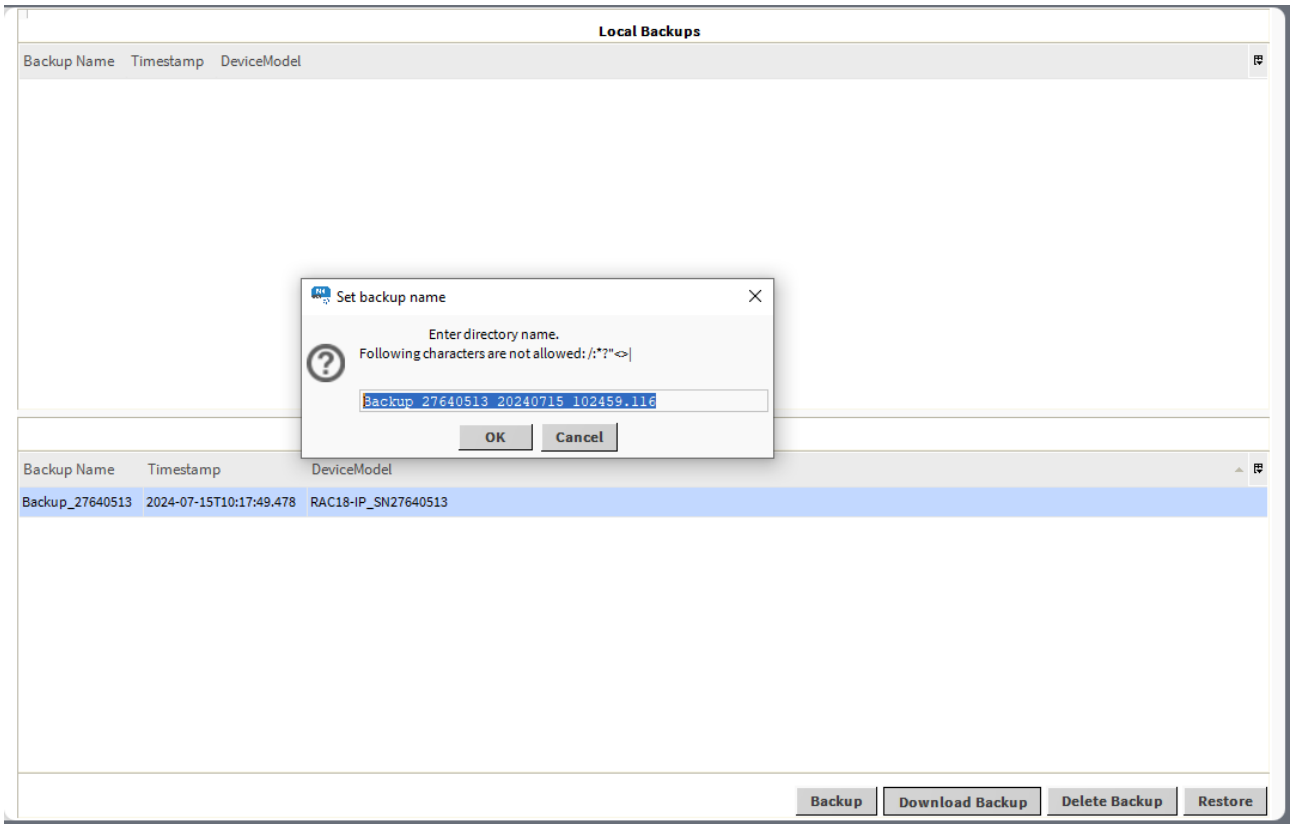


Figure 82. Changing backup name

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:

Figure 83. Invalid characters included in the backup's name

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

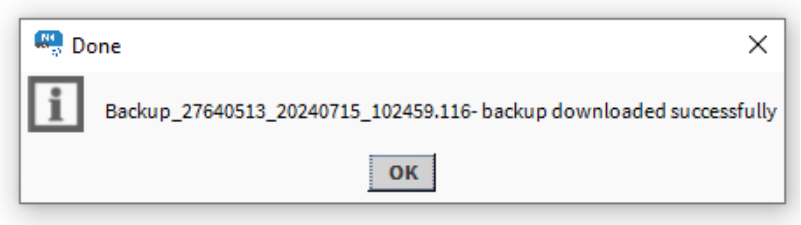


Figure 84. Successful backup confirmation

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.

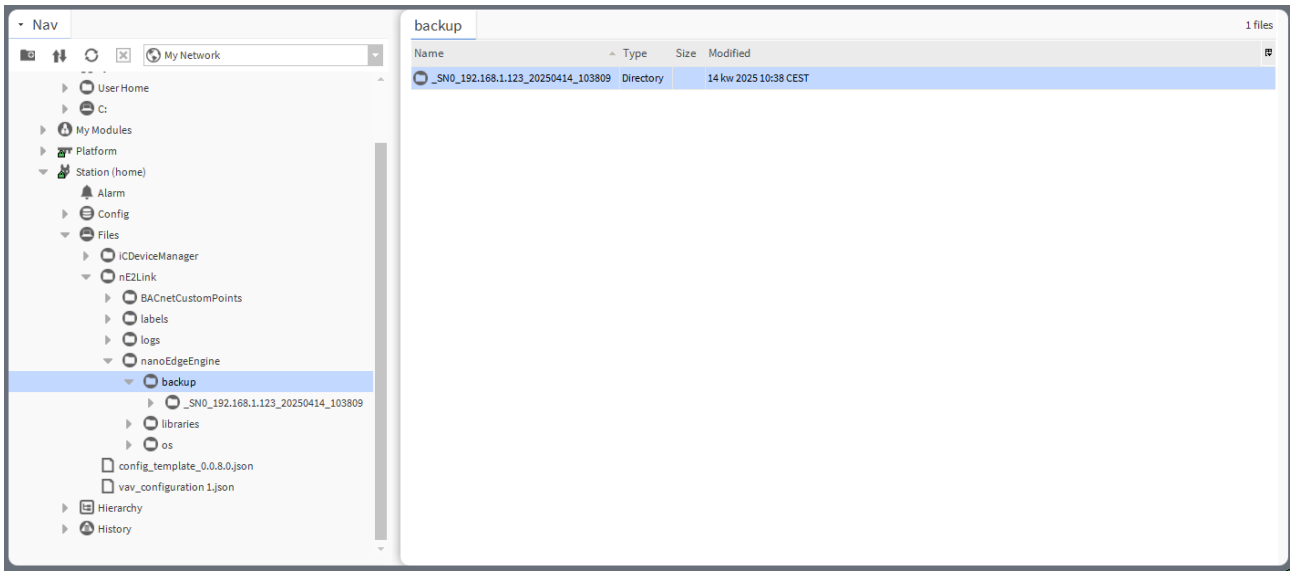


Figure 85. Backup stored in the station files

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.

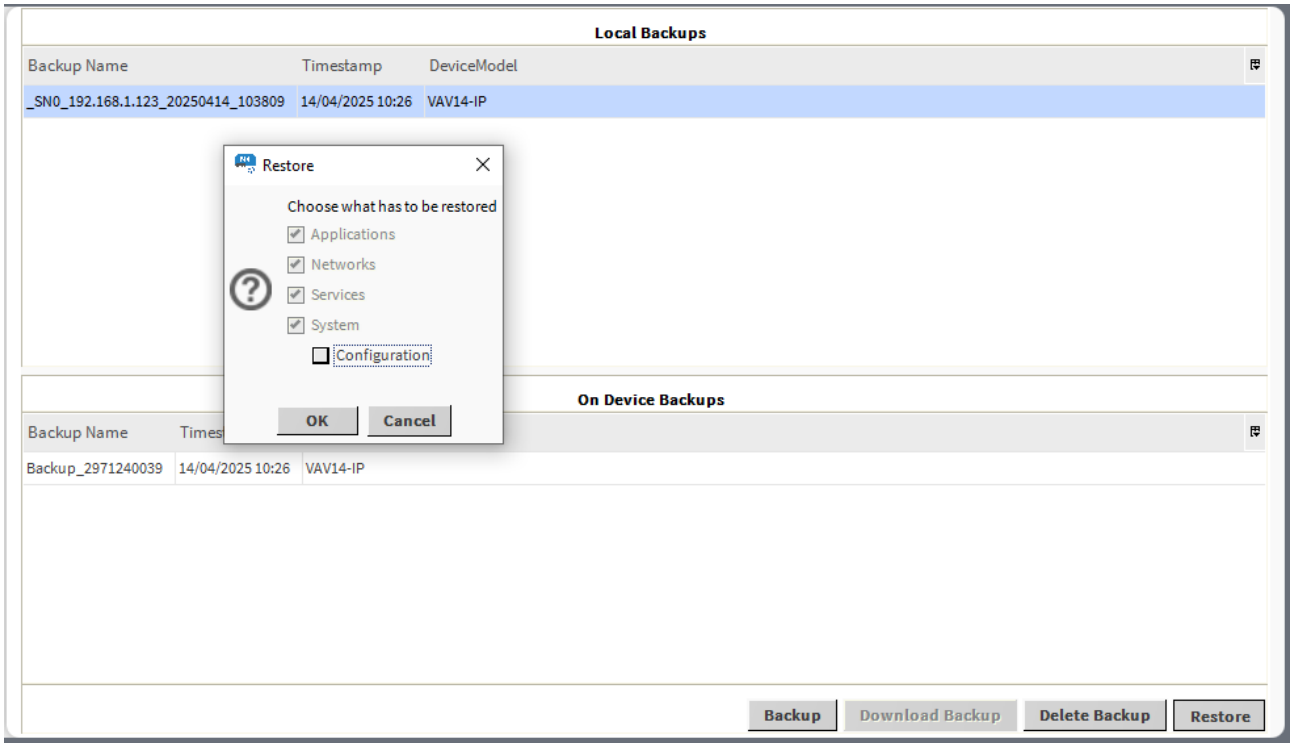


Figure 86. Available restore options will appear after clicking on the Restore button

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.

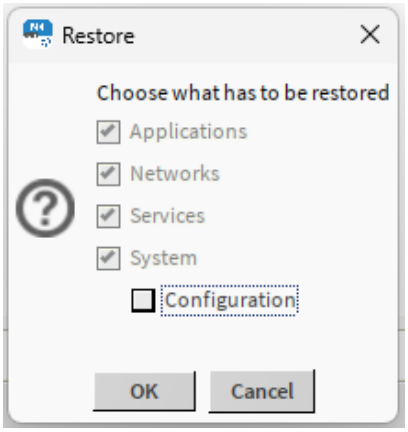


Figure 87. Restore pop-up

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.

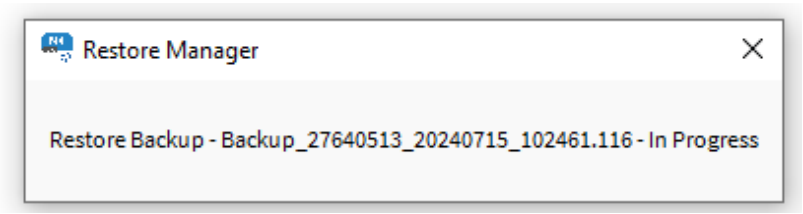


Figure 88. Restoring in progress

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.

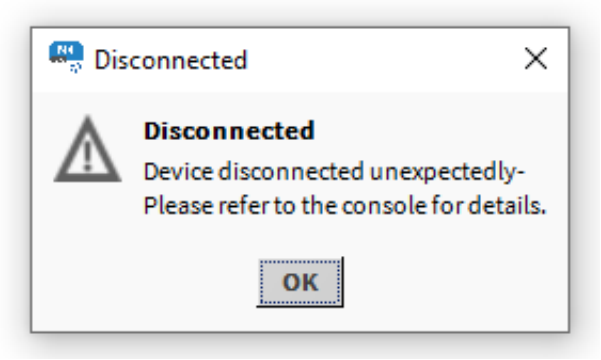


Figure 89. Device disconnected after restoring backup

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).

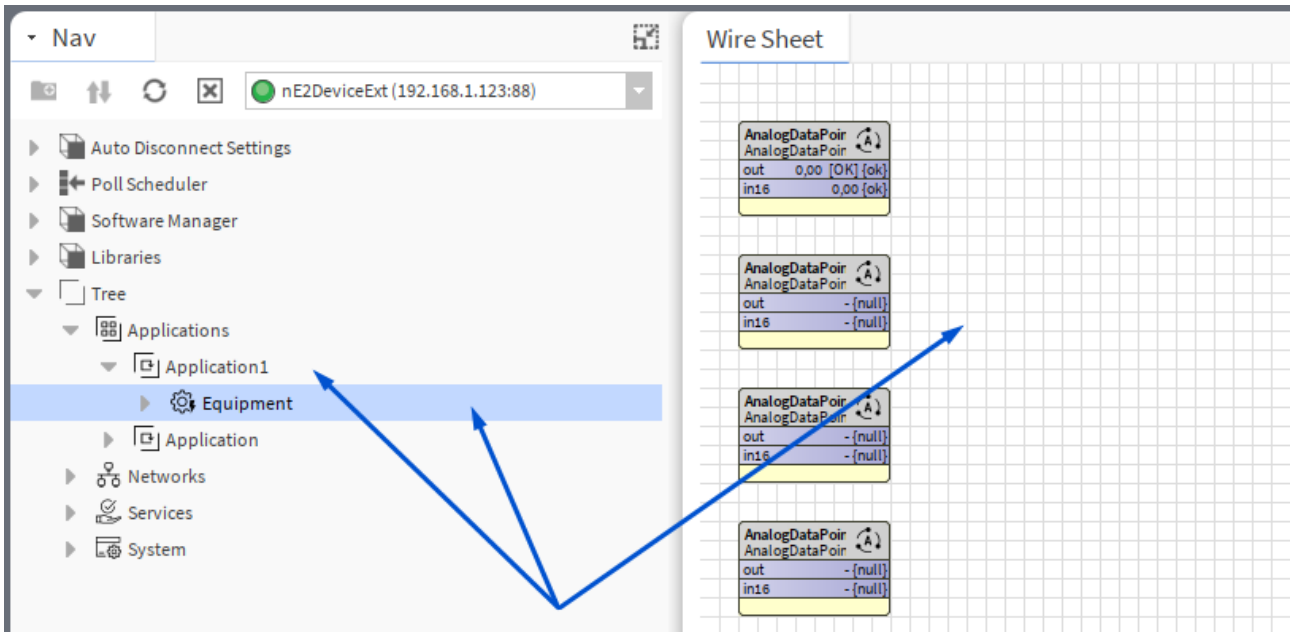


Figure 90. Example locations to drop a component

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.

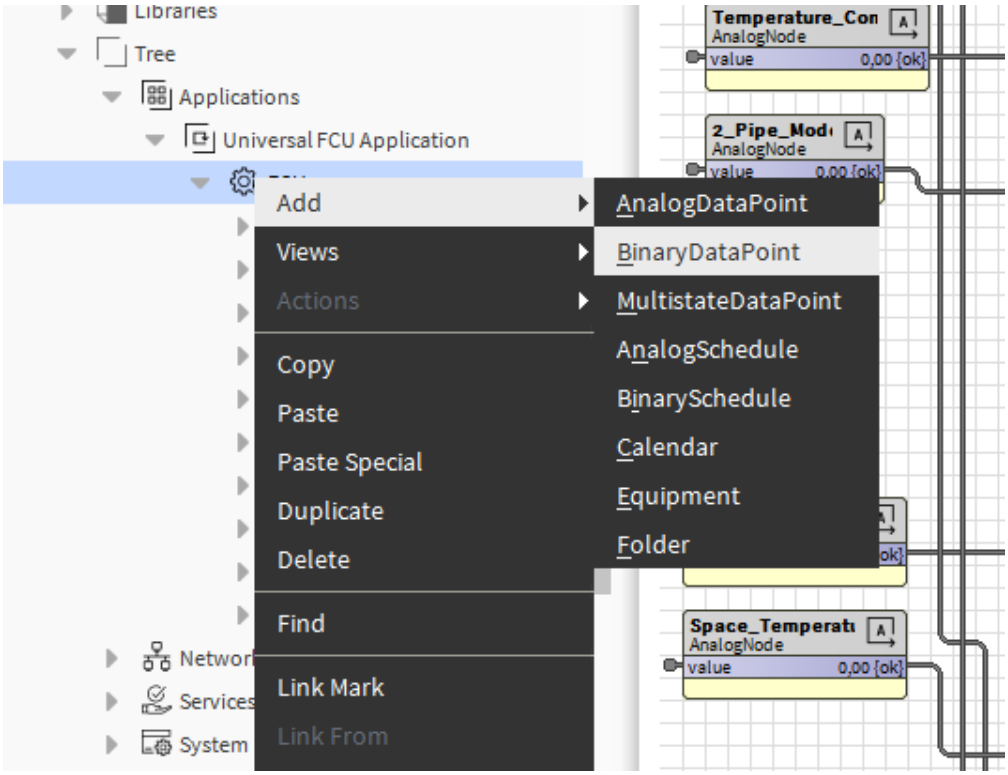


Figure 91. Context menu for adding basic components in the device tree

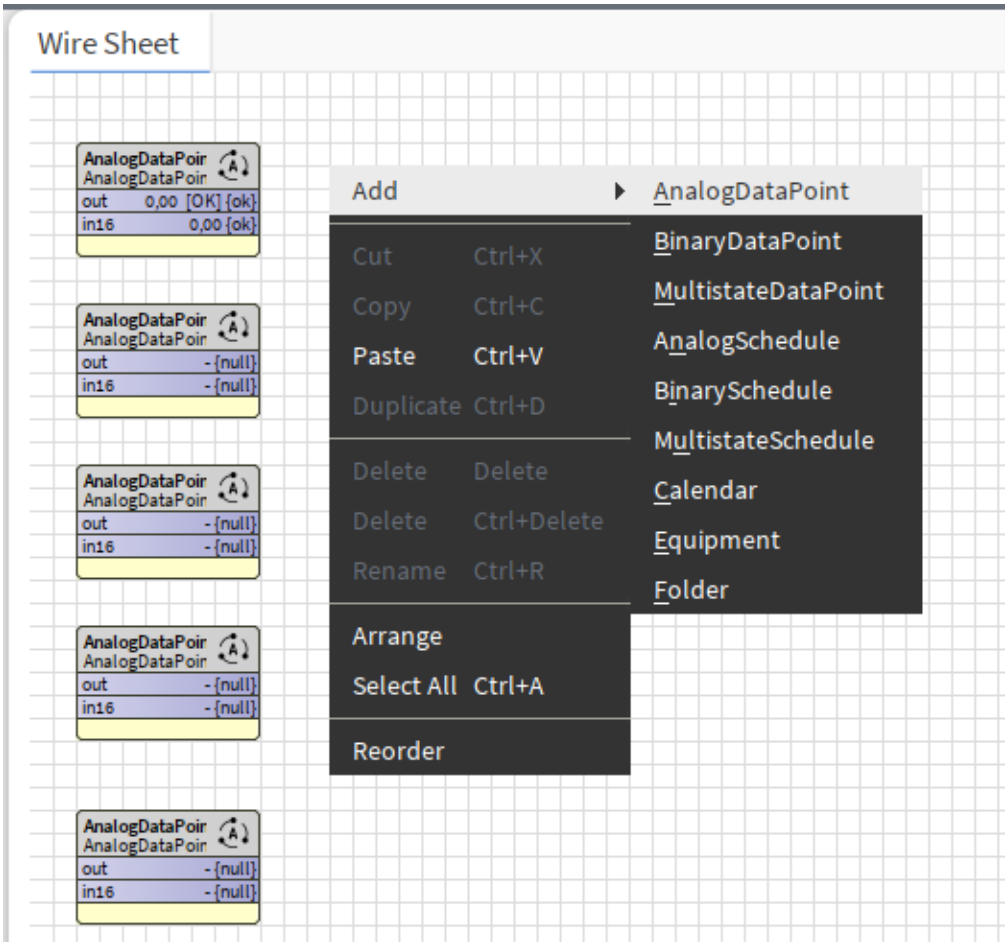


Figure 92. Context menu for adding basic components on the wire sheet

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 nanoEDGE Engine organizes the application structure using a clear, hierarchical model designed for scalable, logical, and intuitive system configuration. The architecture follows the pattern:

- Applications container
  - **Application component** (top-level component required to build any logic in the engine; multiple applications can run independently in different time cycles and contain one or more Equipment objects and Data Points)
    - **Equipment component** (optional but recommended to represent real-world systems such as AHUs, VAVs, meters, etc.; can contain Data Points, logic blocks, and sub-equipment components)
      - **Data Point(s)** (represent all the core system data such as sensors, setpoints, or commands, essential for exposing data to the Niagara Framework and external protocols, and for enabling features like trending, alarming, and tagging)
      - other components
    - Equipment component
      - Data Point(s)
      - other components
    - etc.

To learn more about the Application and Equipment component, Data Points, and other components available for programming, please refer to to the nano EDGE ENGINE Programming user manual.

This structure is not just for the naming convention. It serves a purpose as a data modeling best practice that simplifies engineering and prepares the system for future analytics integration.

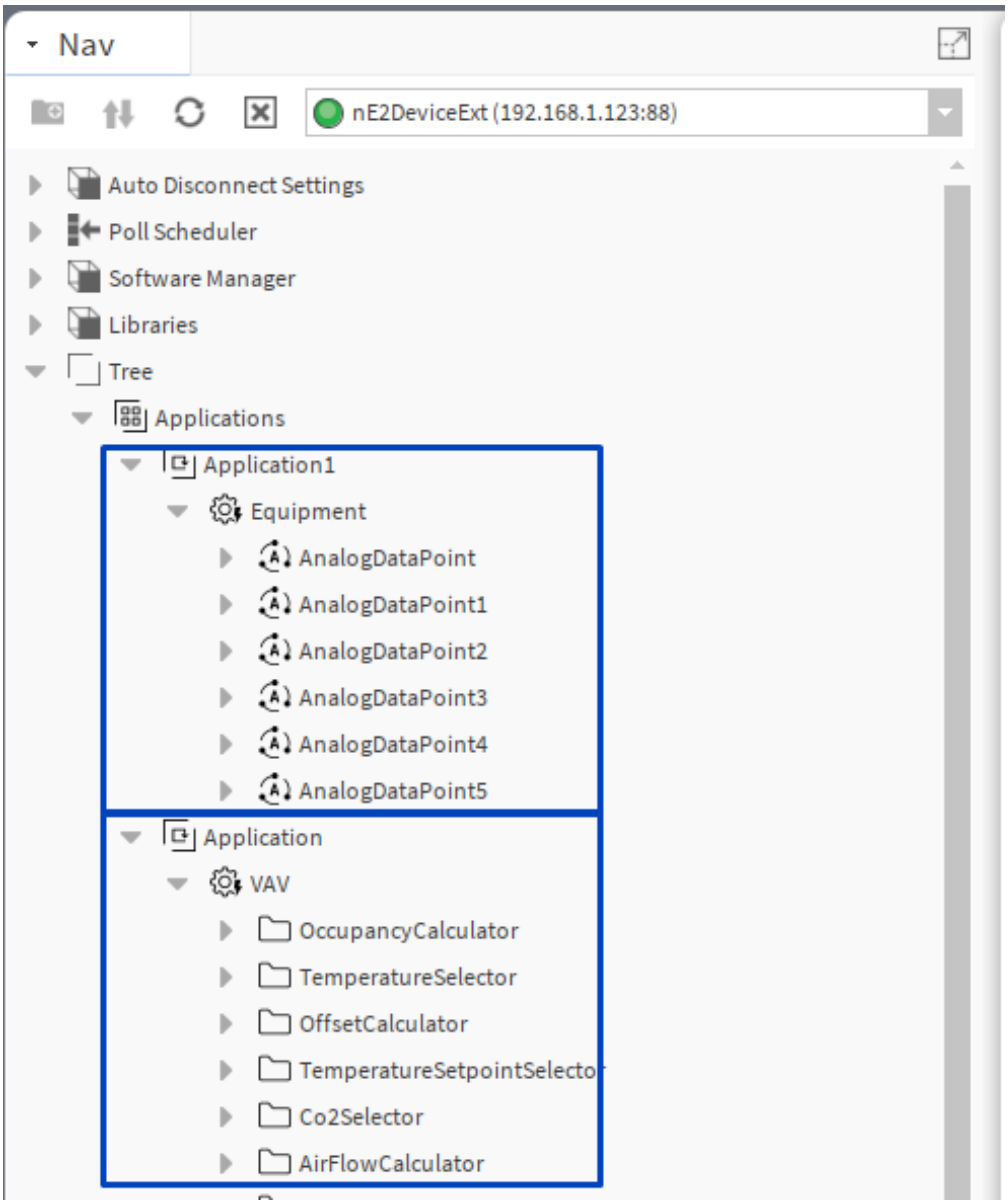


Figure 93. Recommended application structure

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:

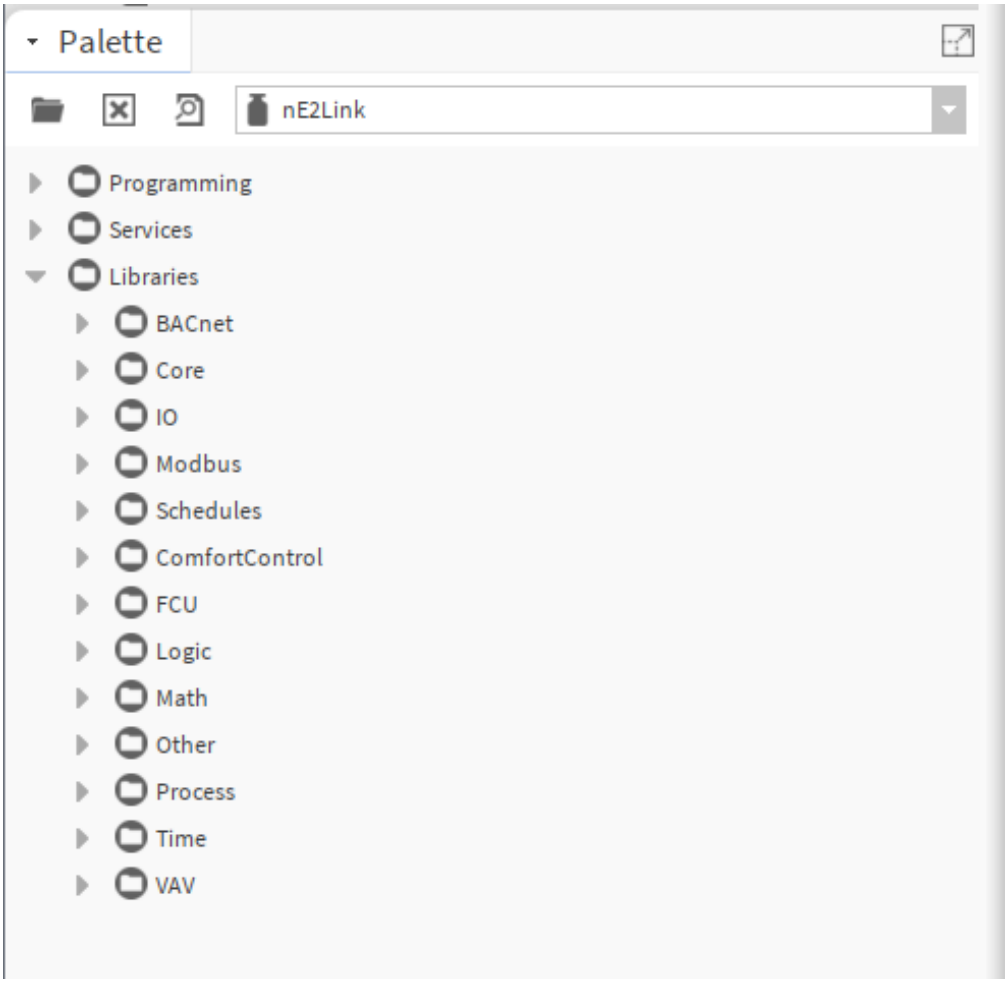


Figure 94. Libraries available in the palette

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:

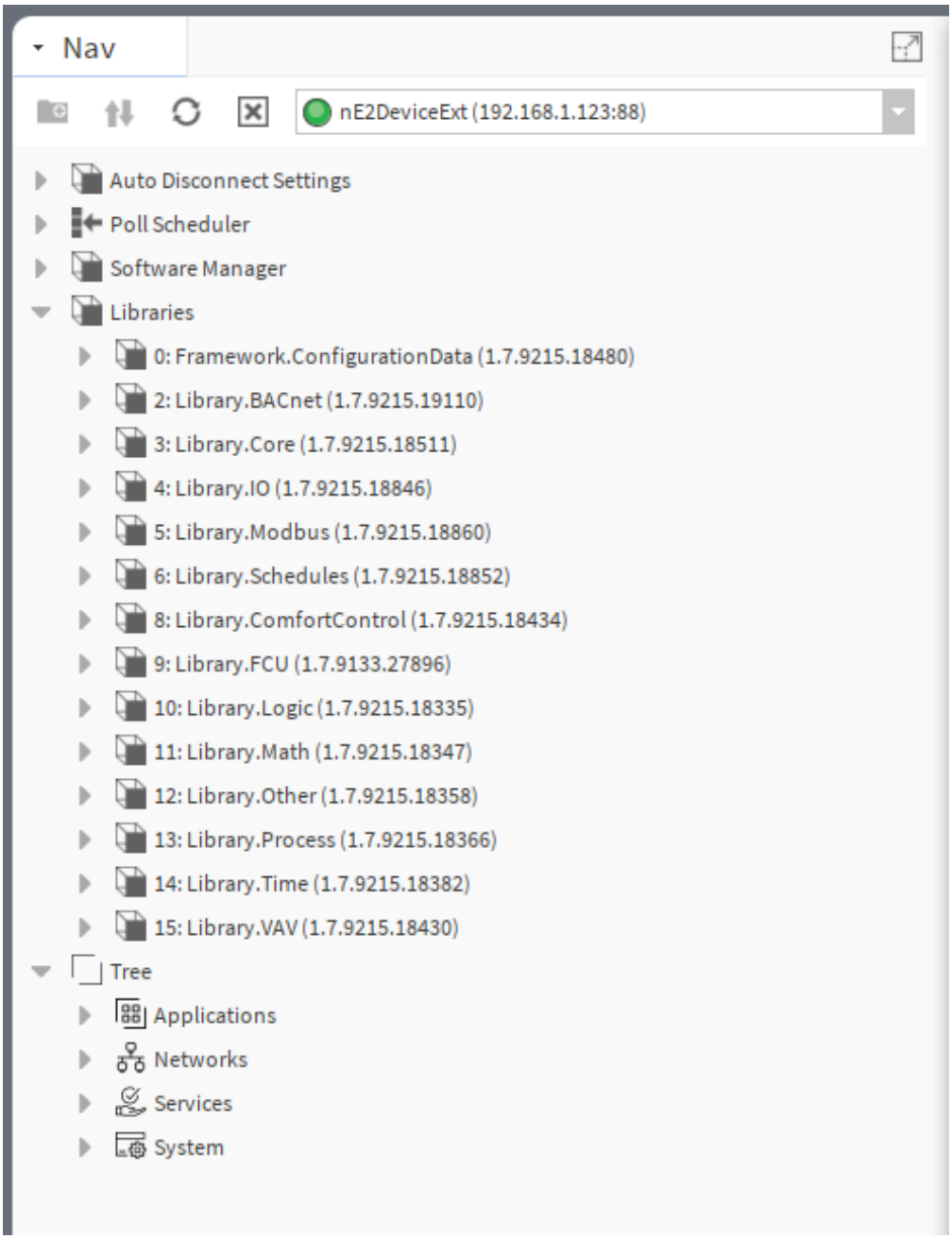
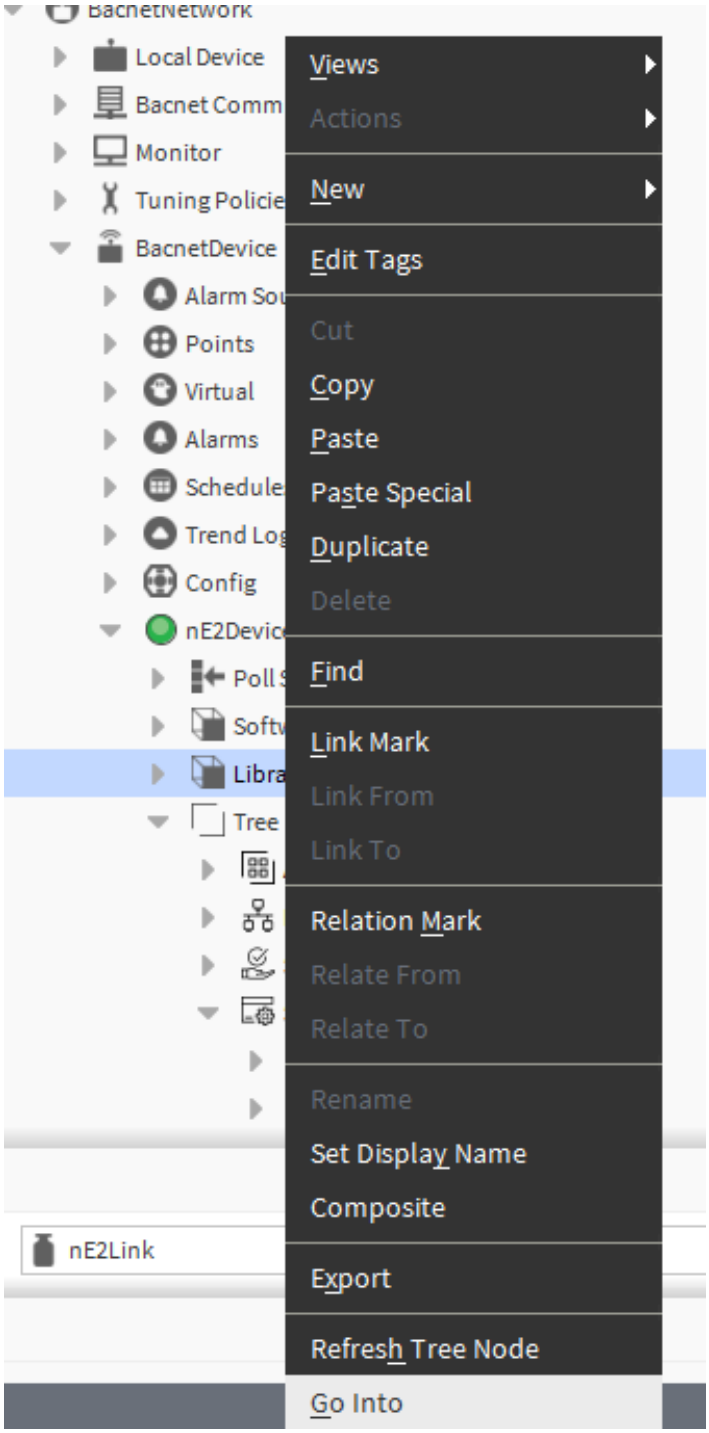


Figure 95. Libraries available in the device tree

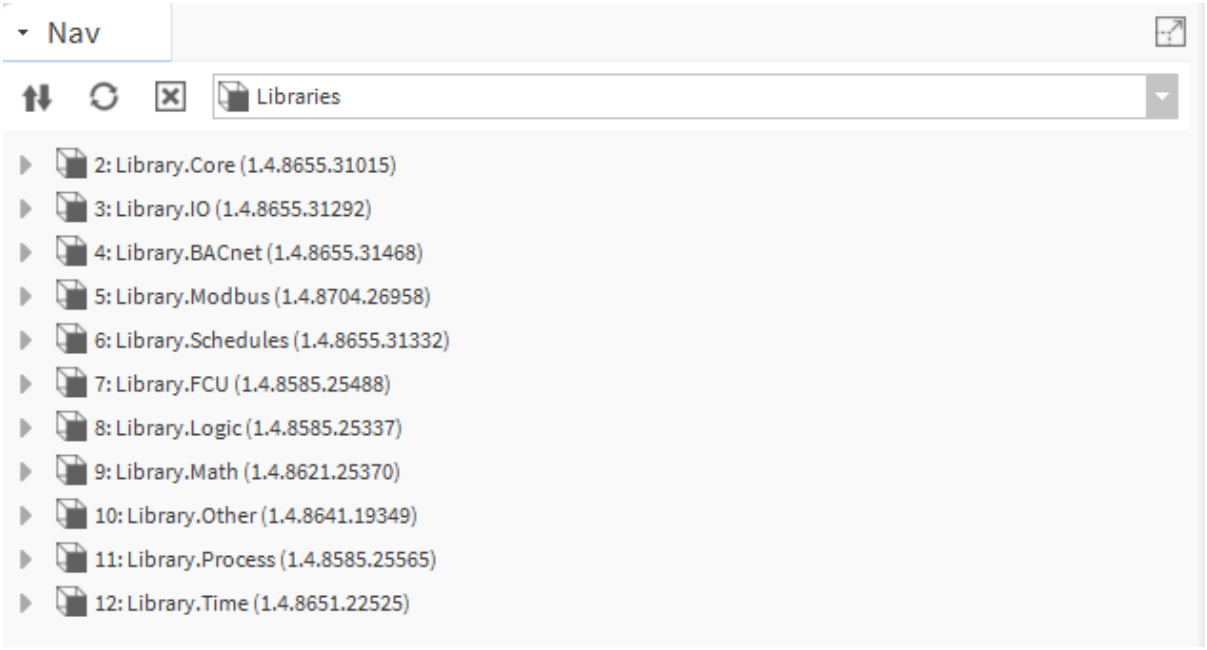
The libraries on the device act as a palette of components that can be dropped into the device logic. To enhance the user experience, it is recommended to open an additional nav view for the Libraries view:

- 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.



An on-device libraries list will appear.



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.

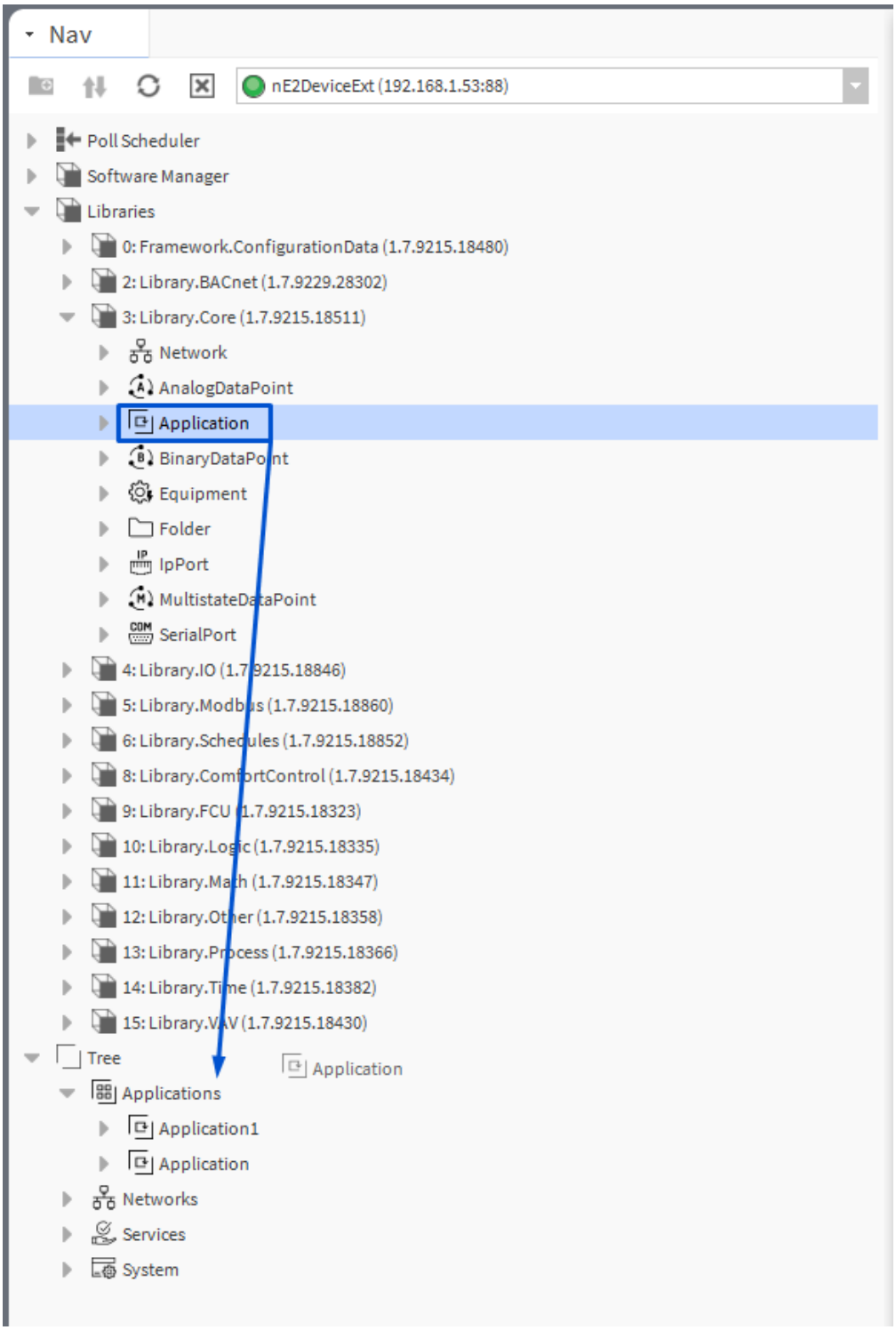


Figure 96. Drag and drop the Application component to Applications

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

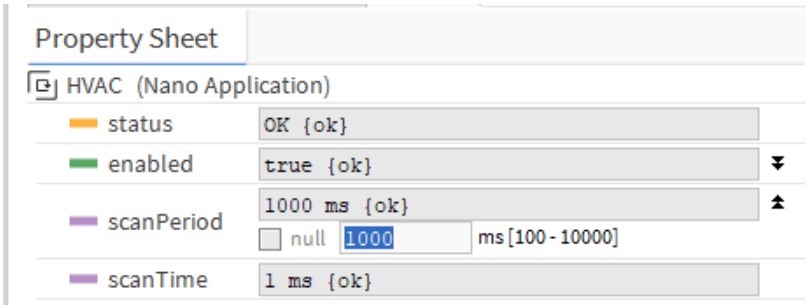


Figure 97. Setting the application's scan period

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.

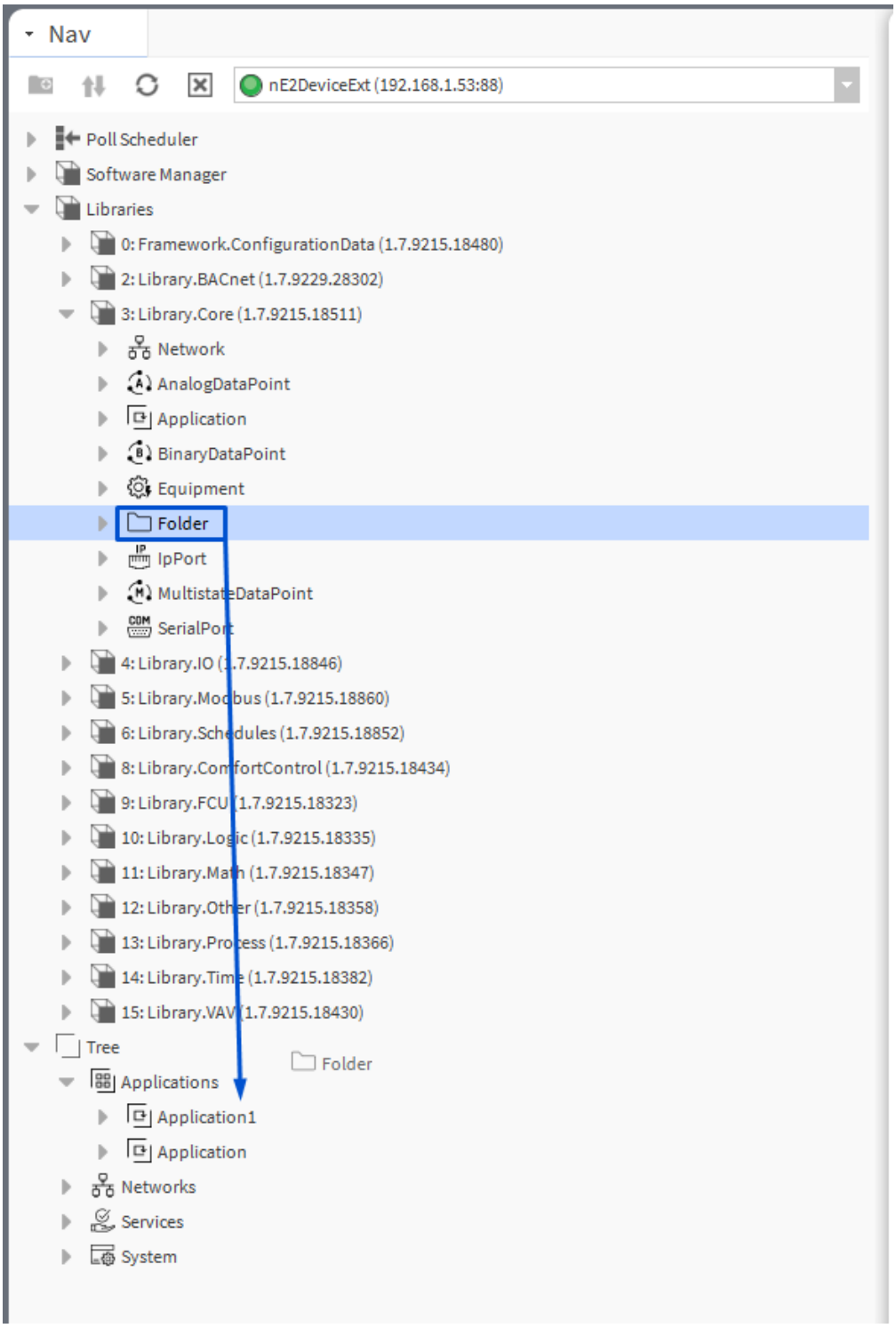


Figure 98. Drag and drop Folder to the Application

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

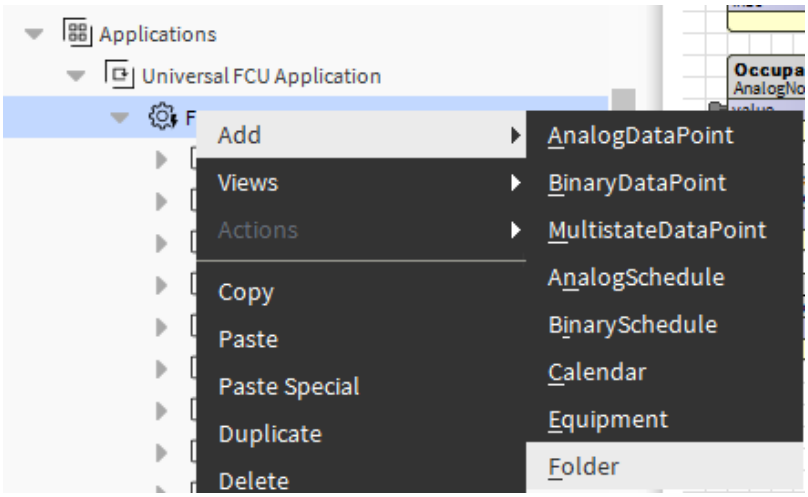


Figure 99. Adding the Folder component from the context menu

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.

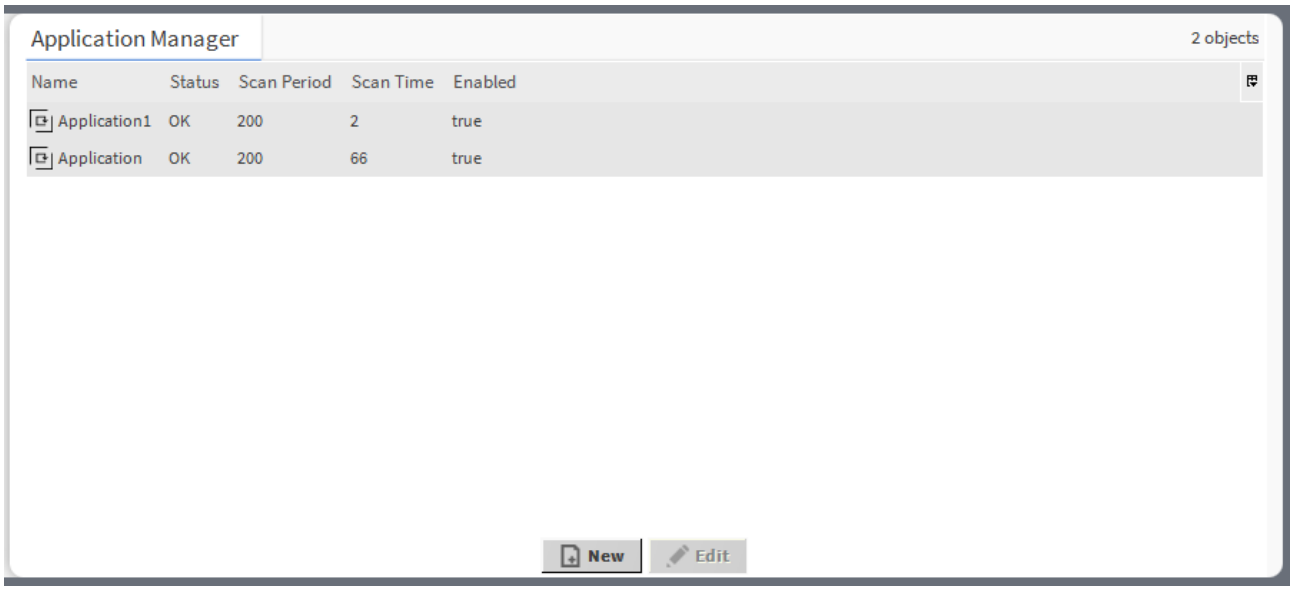


Figure 100. Application Mananger

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:

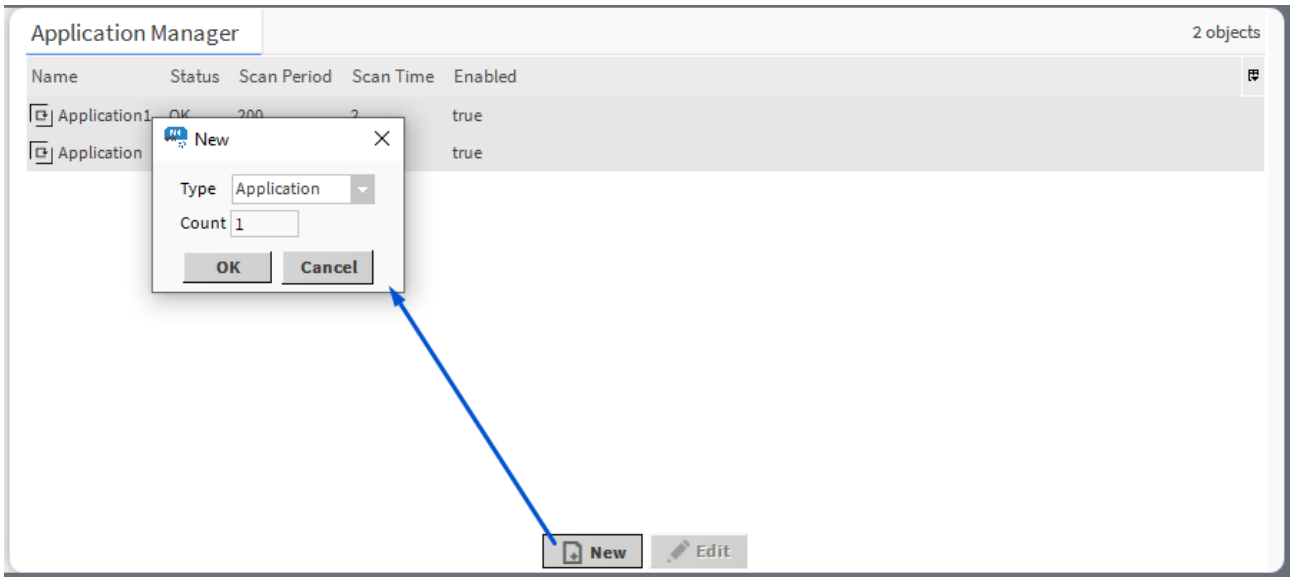


Figure 101. Adding new Application component in the Application Manager

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

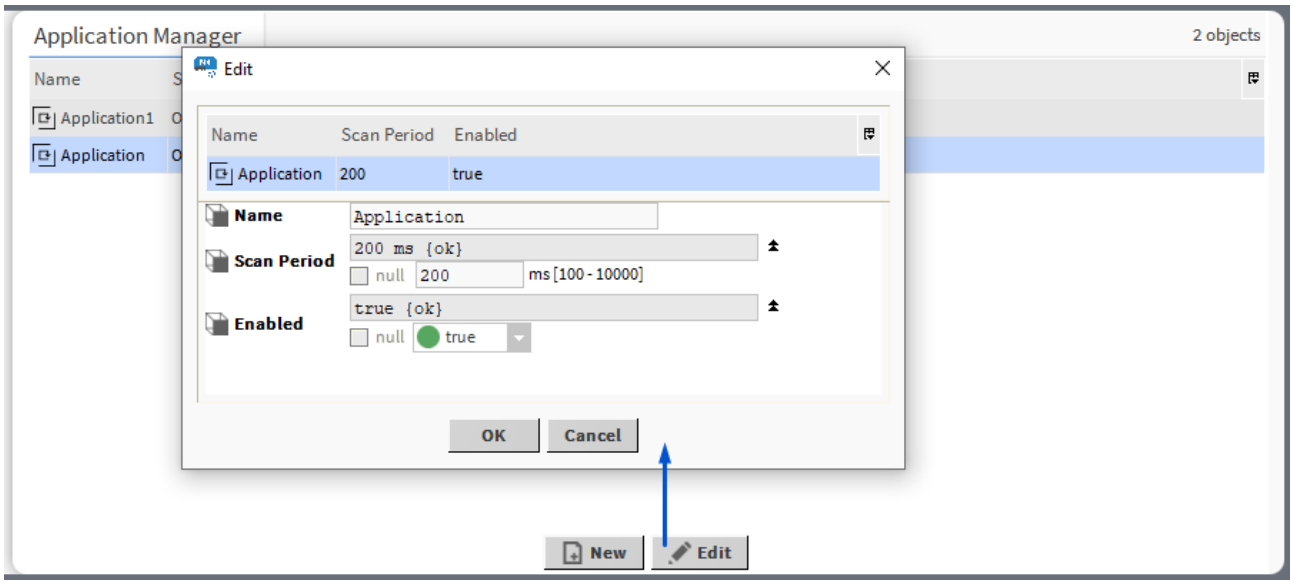


Figure 102. Edit pop-up

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.

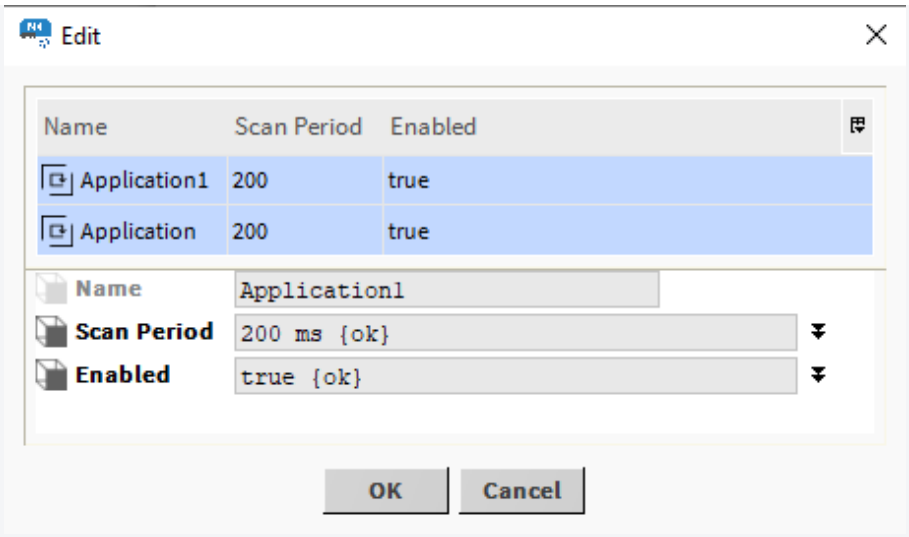


Figure 103. Editing of multiple components

- copy/duplicate/remove Application components:

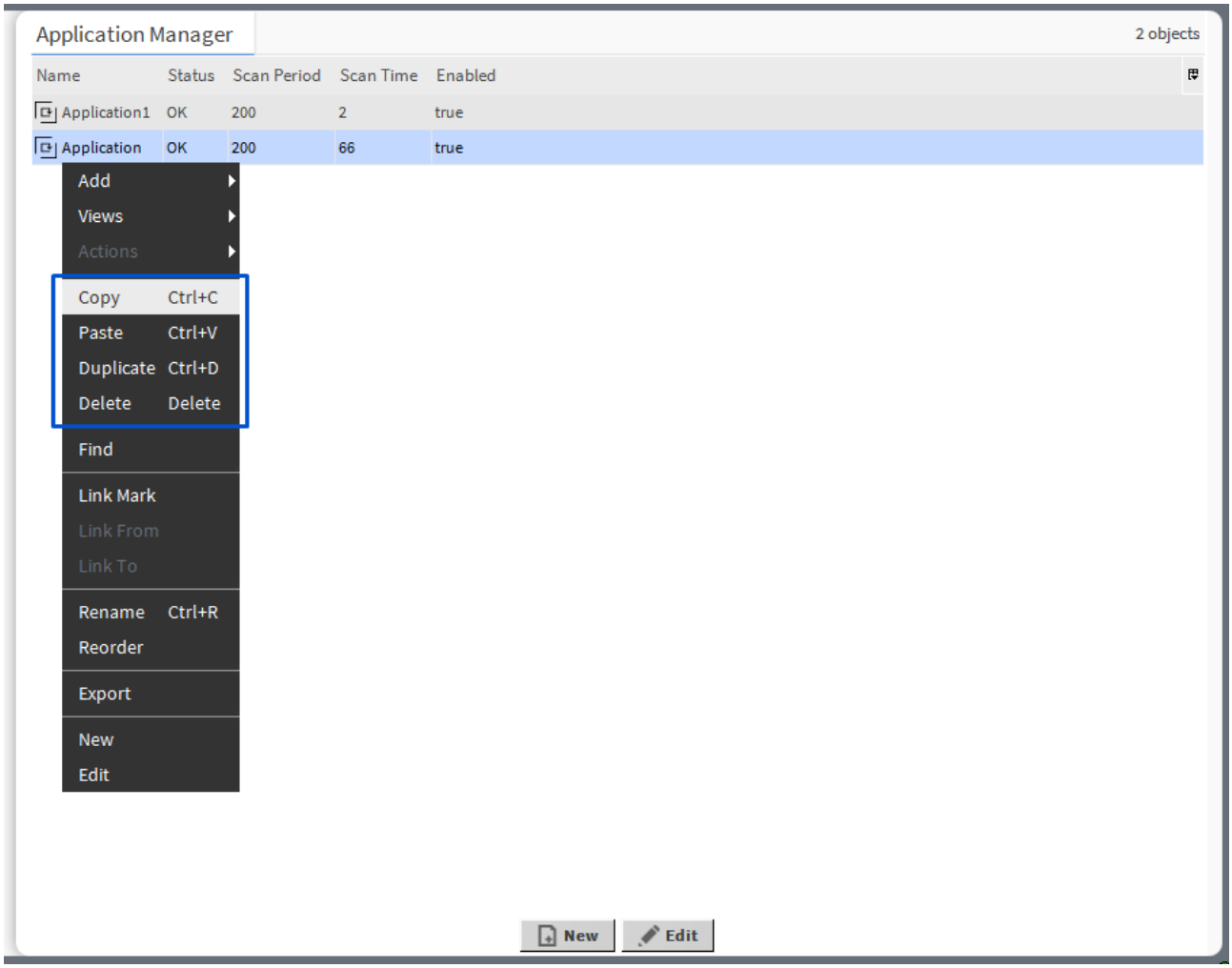


Figure 104. Context menu options for the Application

Opening the Application Manager

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

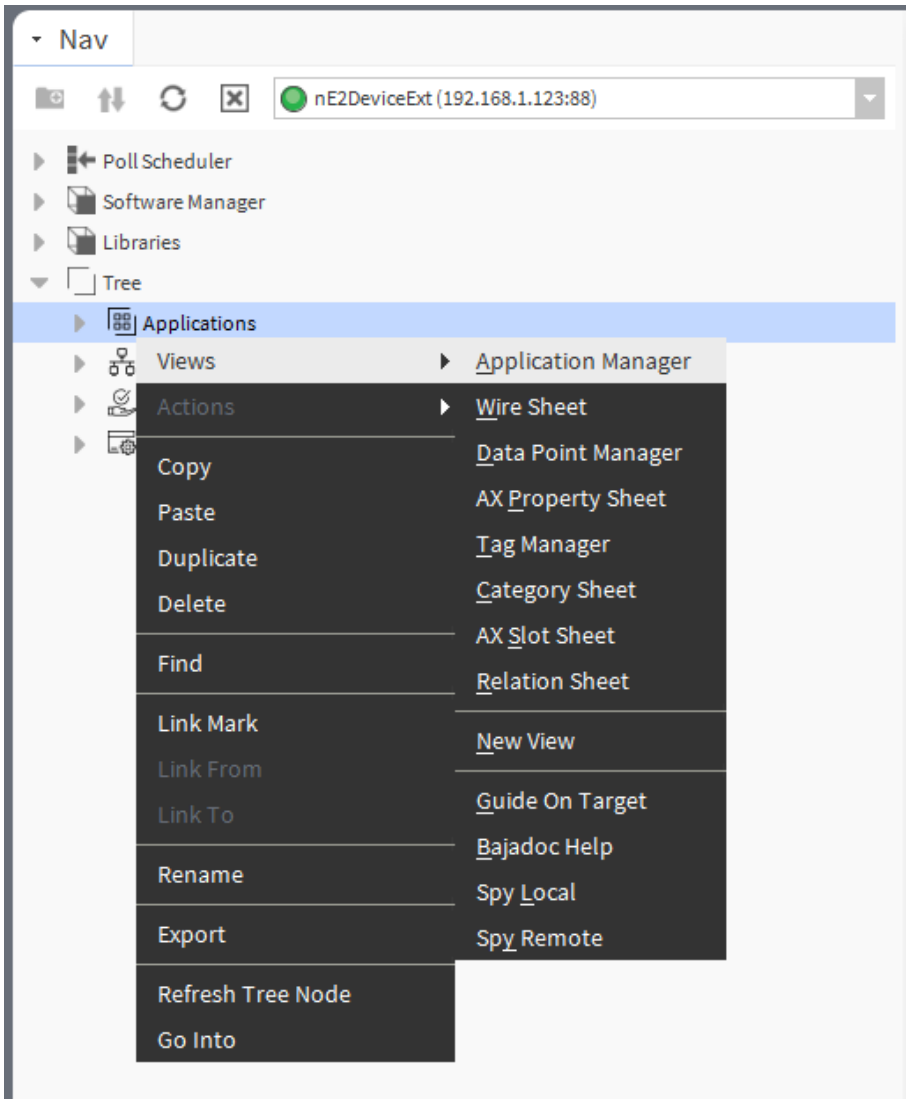


Figure 105. Accessing the Application Manager

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.

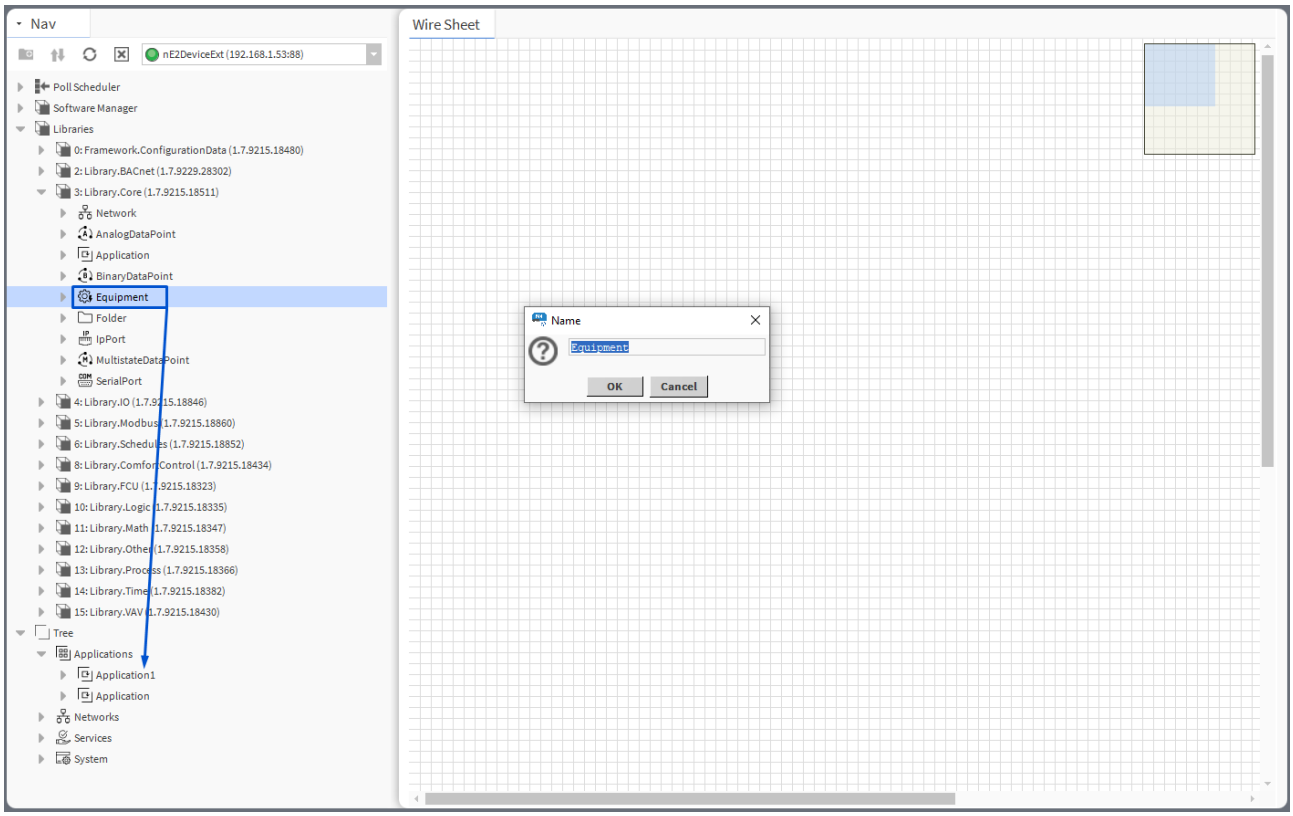


Figure 106. Add (drag and drop) and rename the Equipment component

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

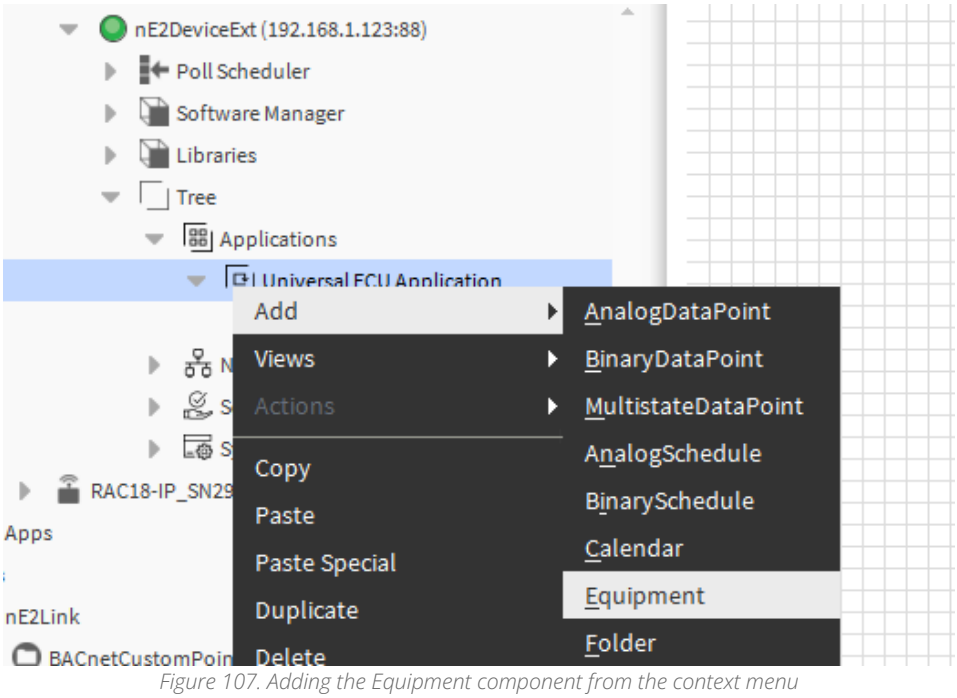


Figure 107. Adding the Equipment component from the context menu

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.

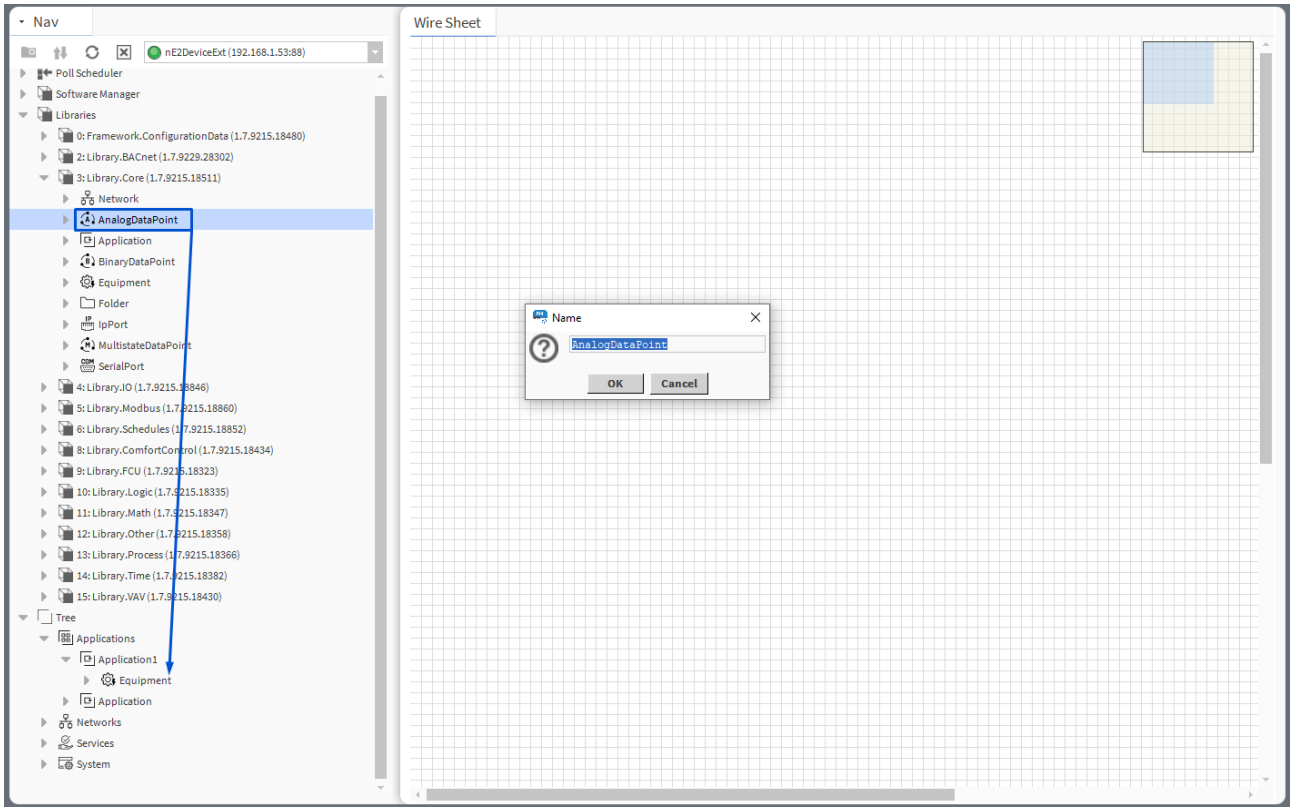


Figure 108. Add (drag and drop) and rename Data Points

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

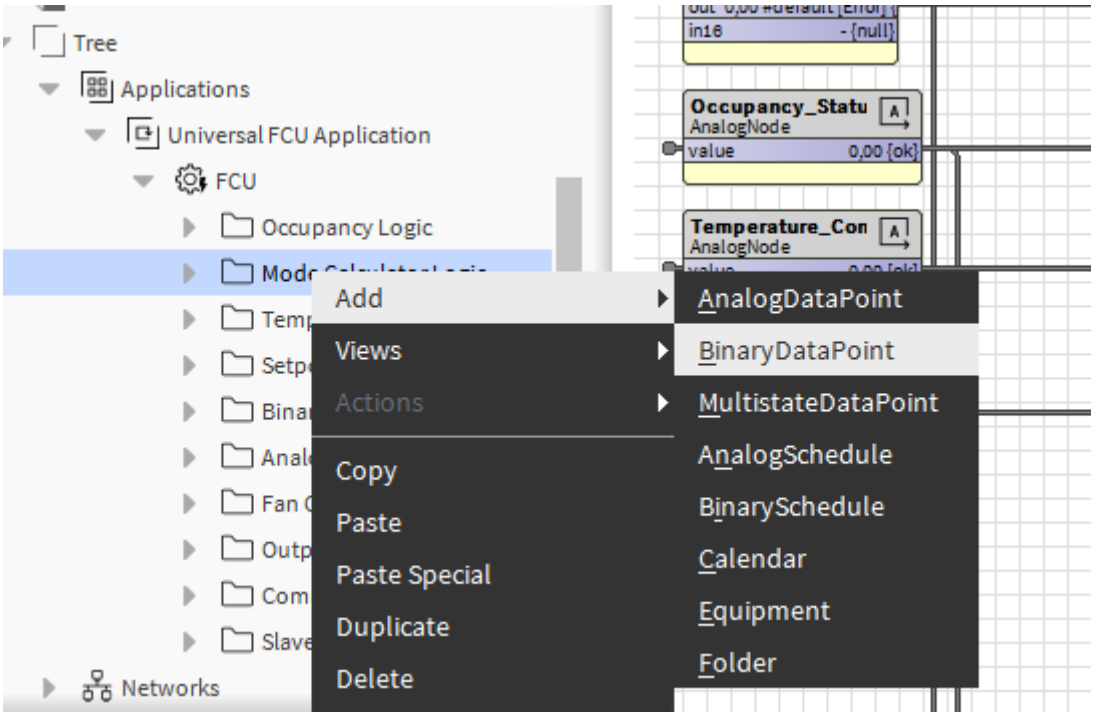


Figure 109. Adding Data Points directly from the context menu

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.

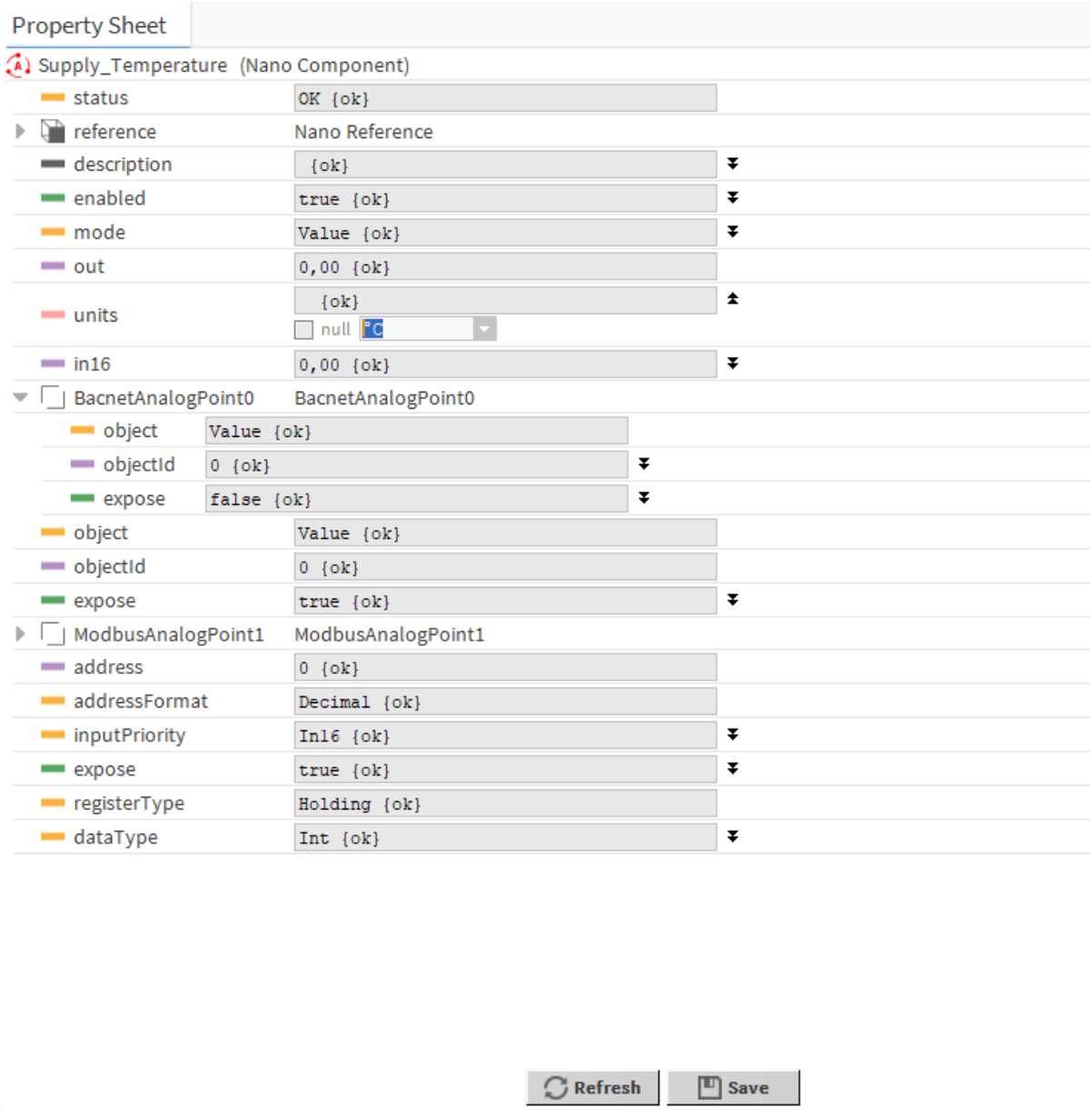


Figure 110. Data Point's Property Sheet view

5.5.3 Adding Extensions

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.

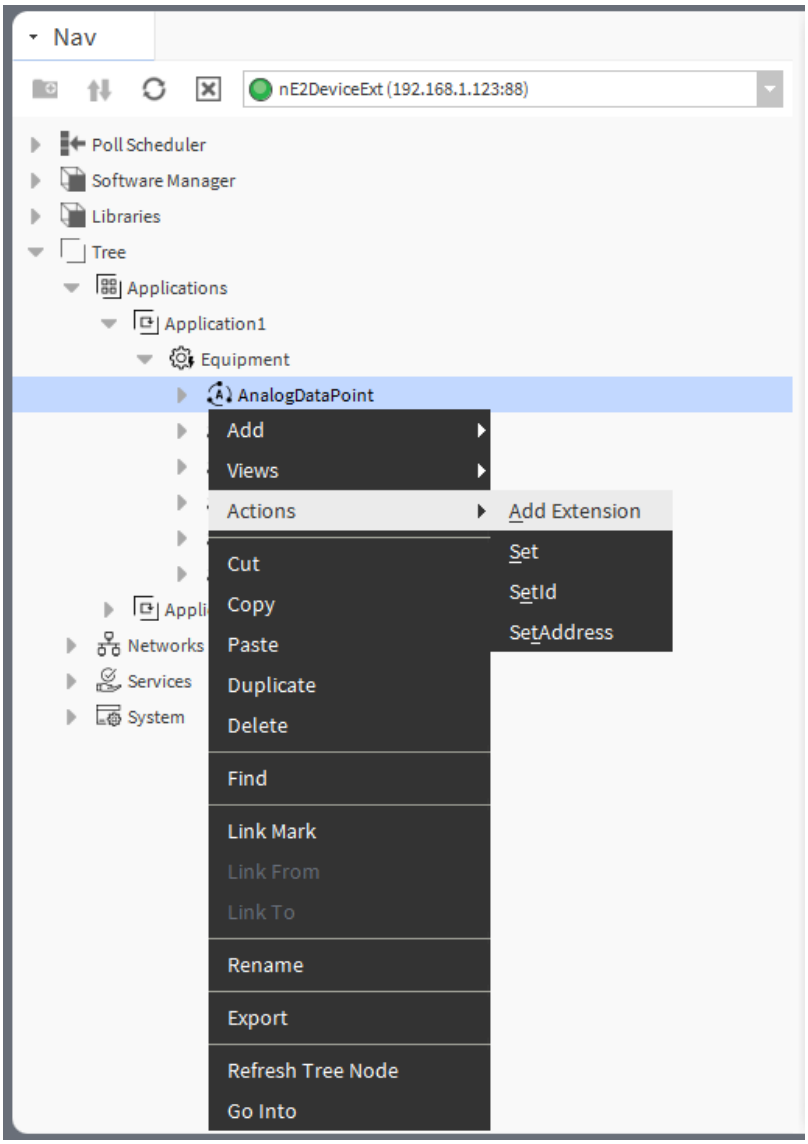


Figure 111. Add Extension action

Available Extensions

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).

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.

Configuration Data

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.

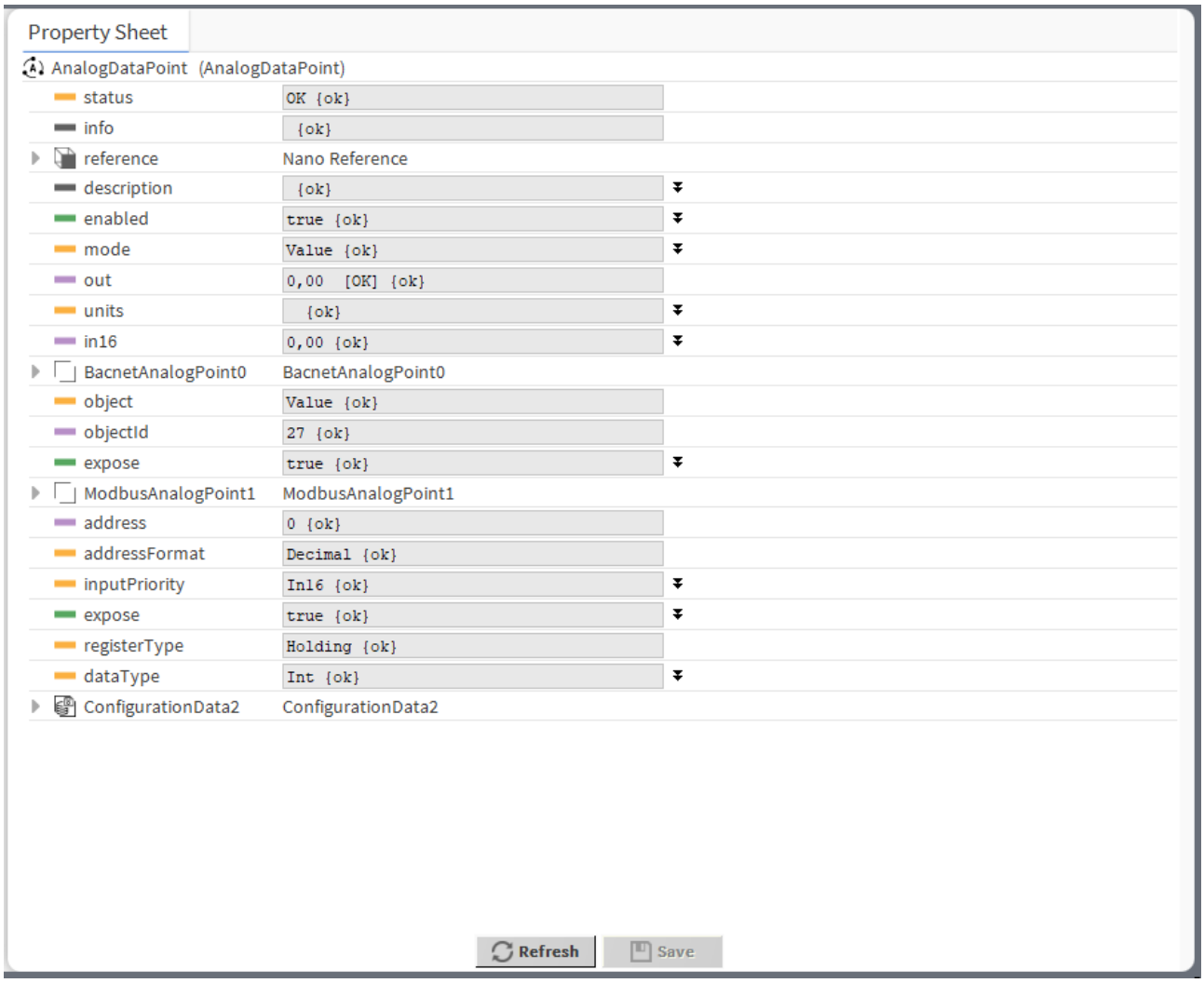


Figure 112. Configuration Data service extension added to the AnalogDataPoint

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.

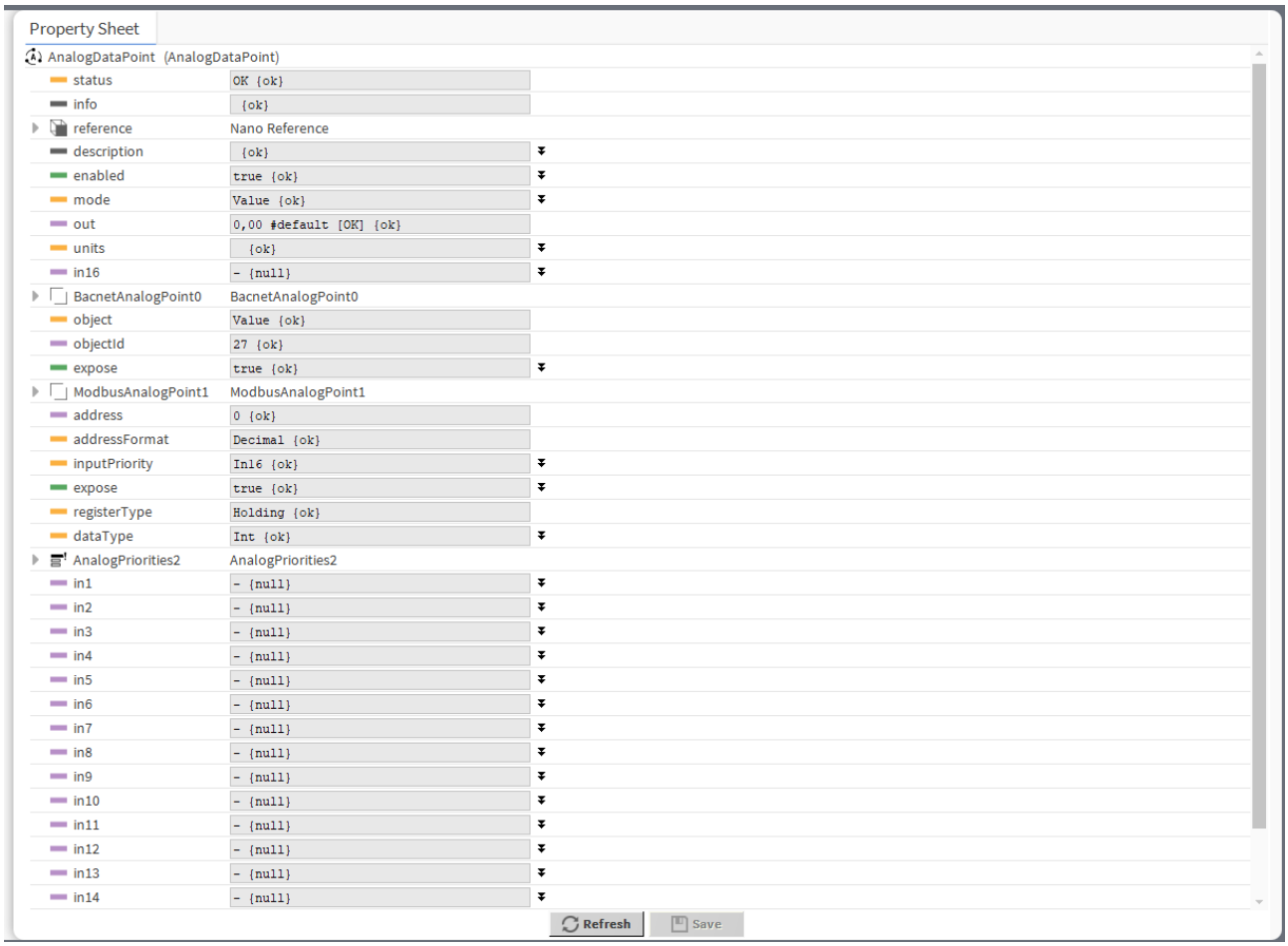


Figure 113. Priorities extension added to the AnalogDataPoint

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.

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).

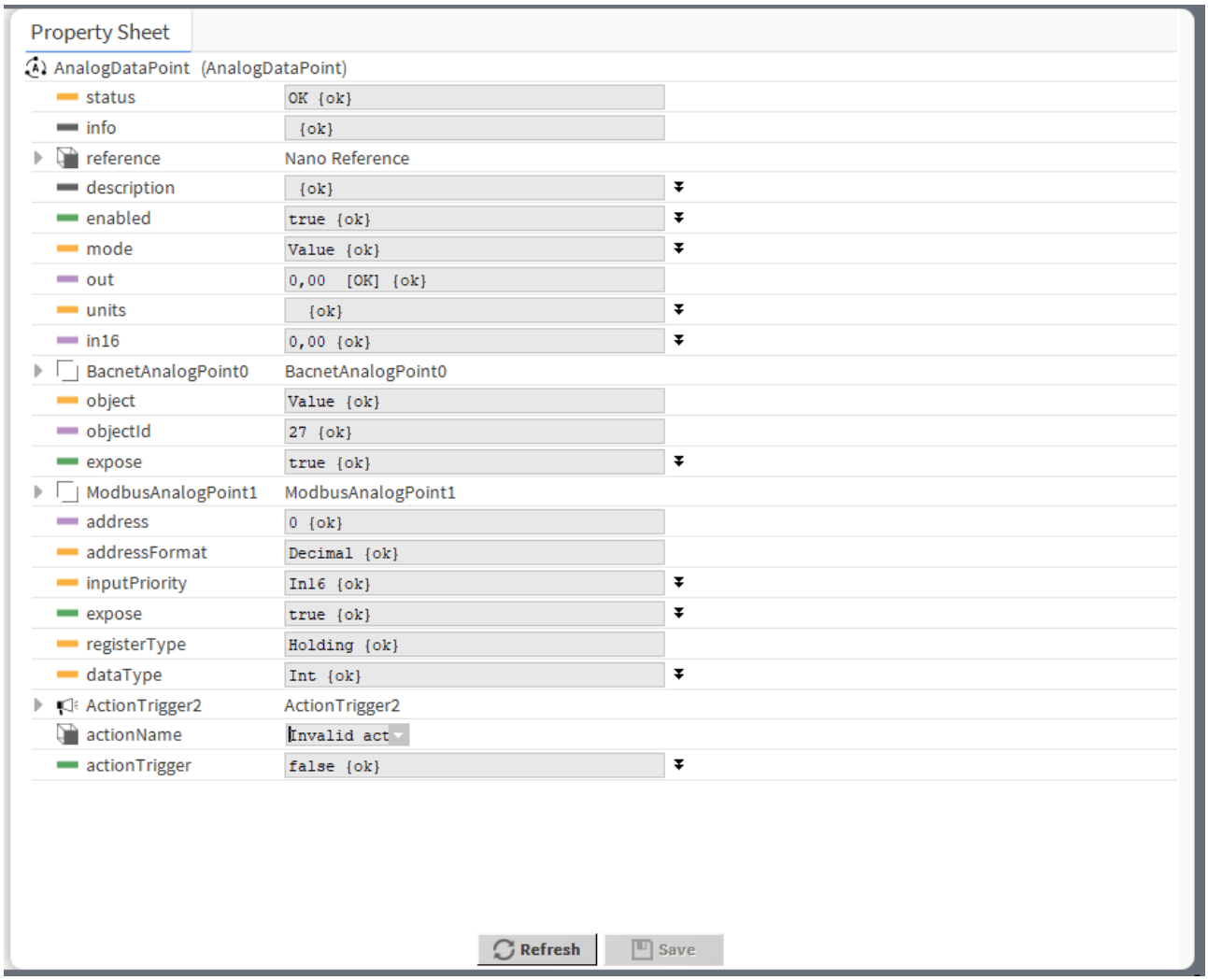
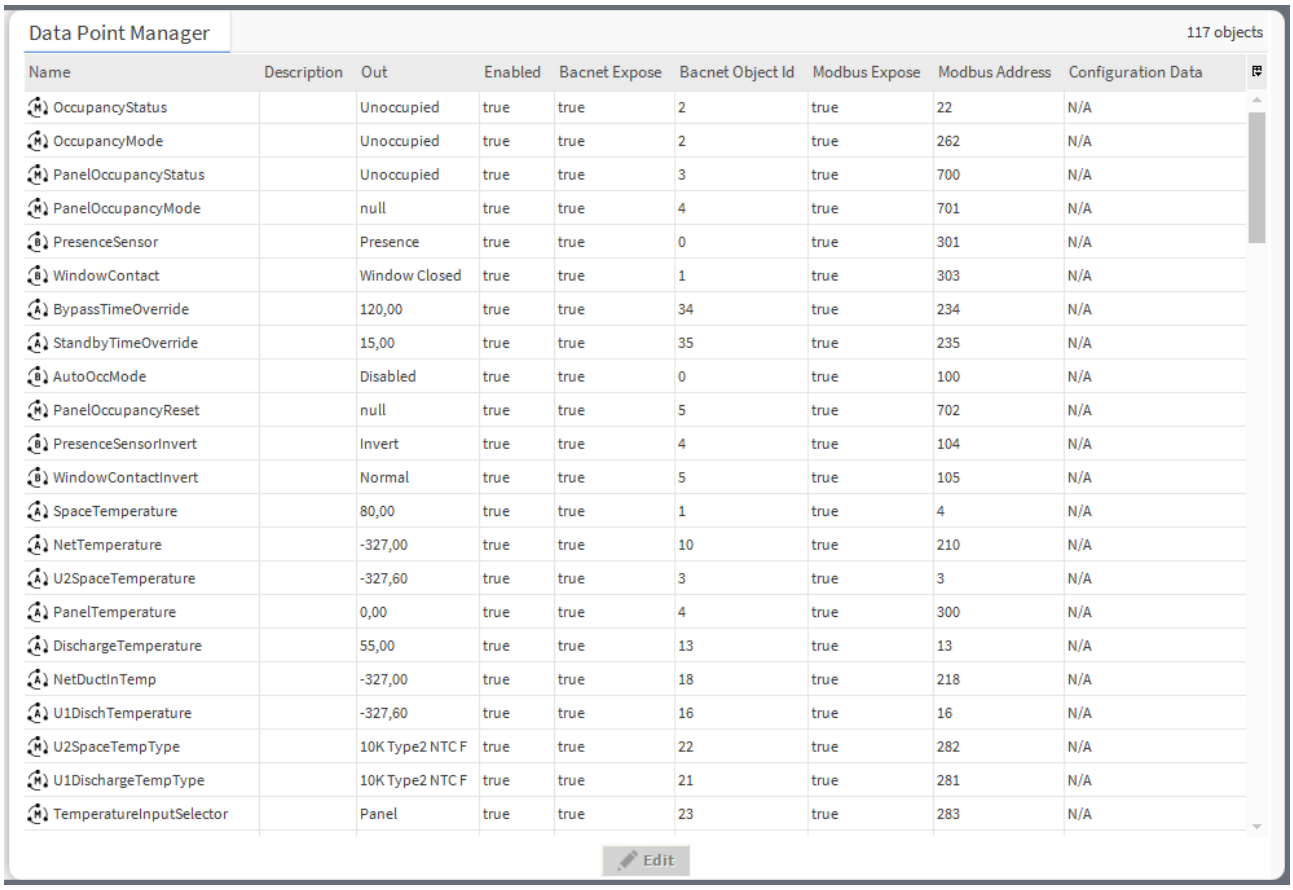


Figure 114. ActionTrigger extension added to the AnalogDataPoint

5.5.4 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

Edit

Figure 115. 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:

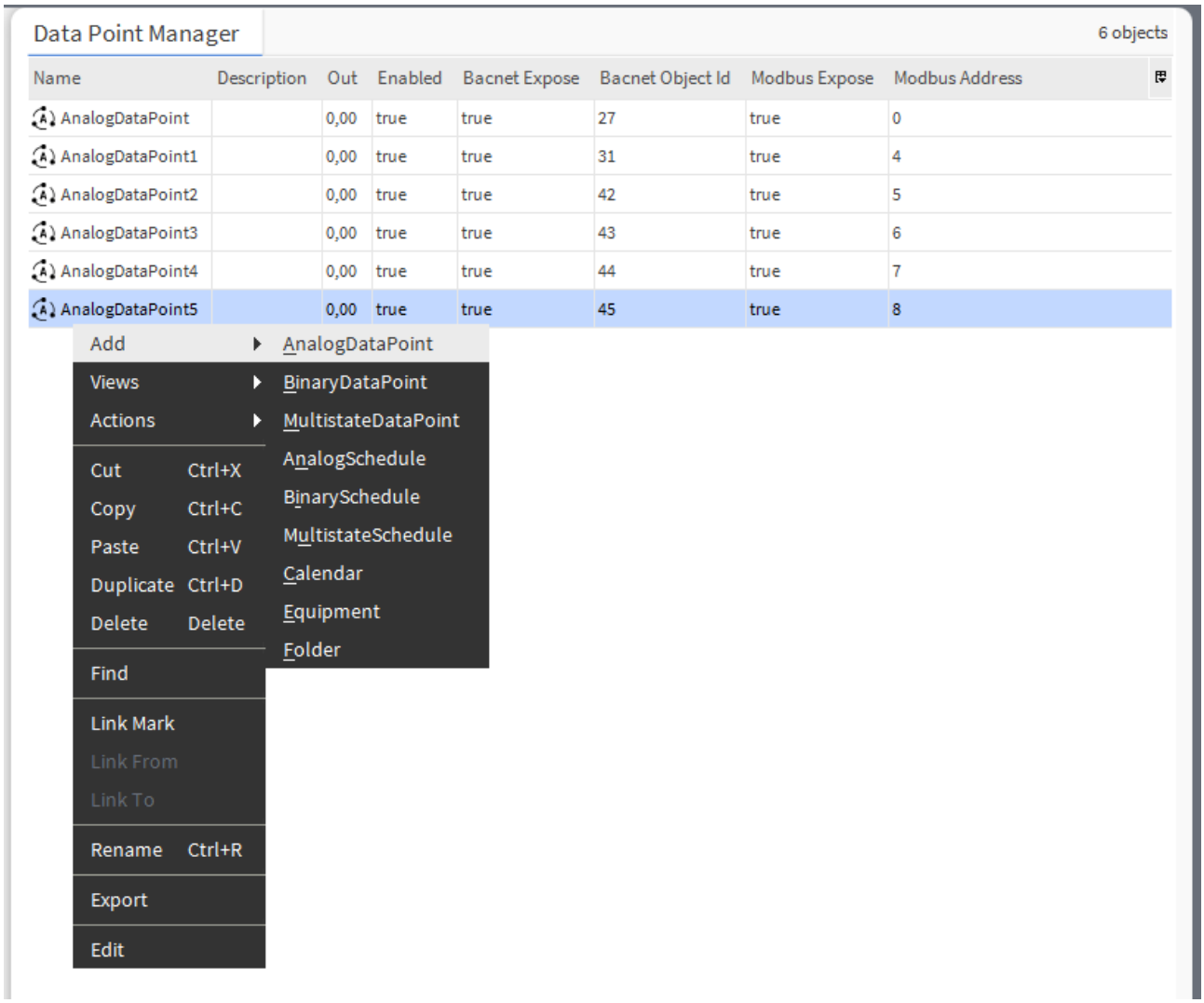


Figure 116. Adding new Data Point

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

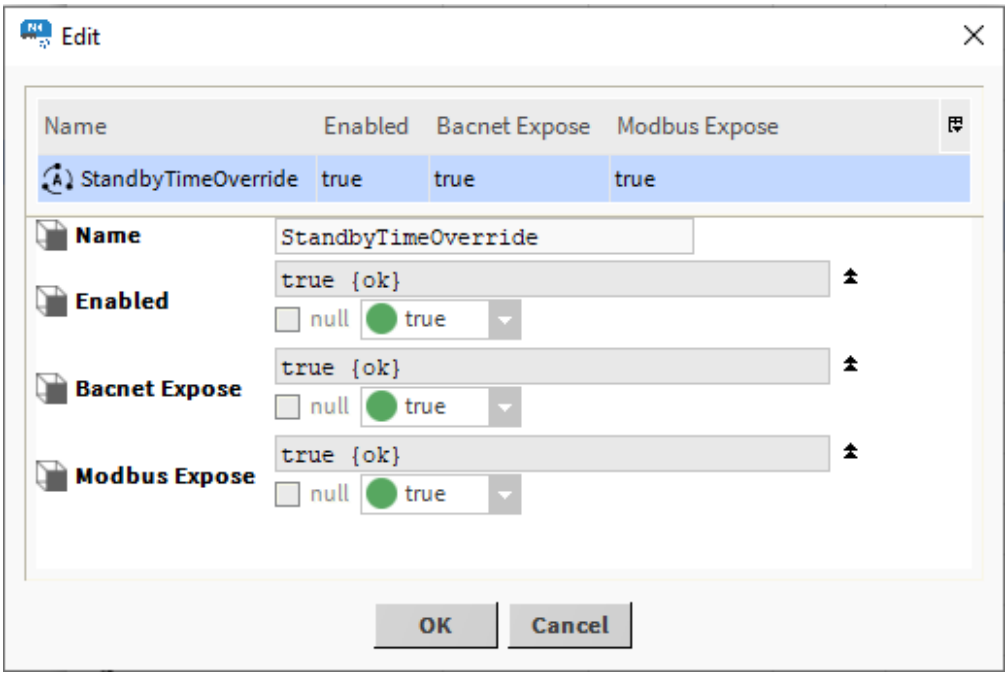


Figure 117. Edit pop-up

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.

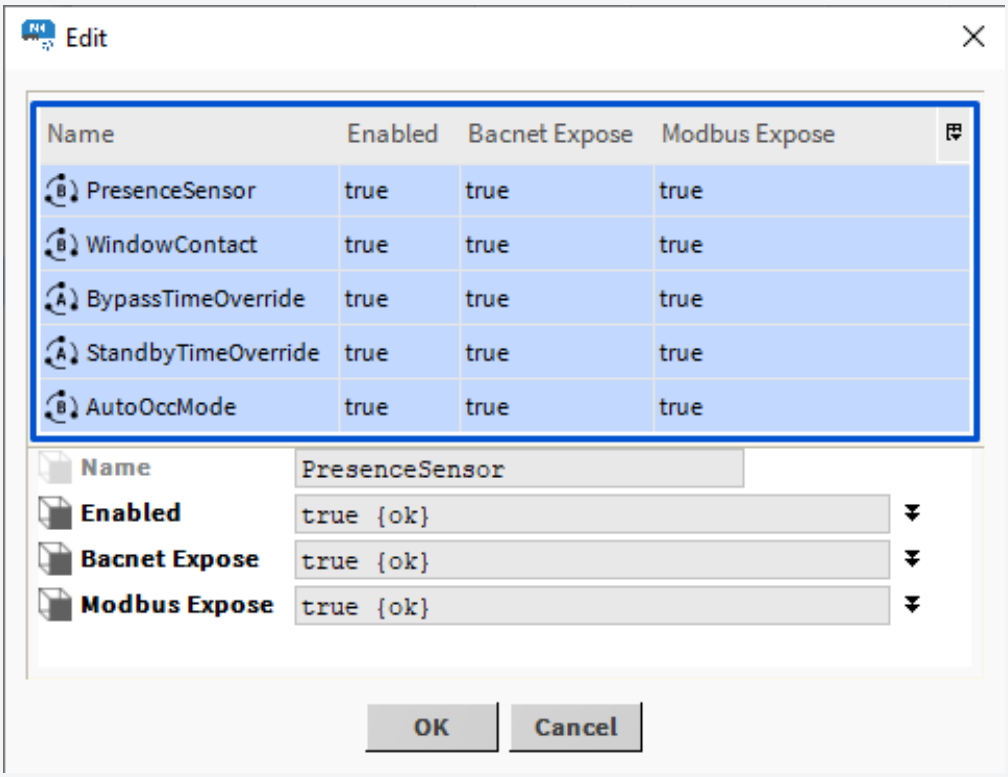


Figure 118. Editing of multiple Data Points

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

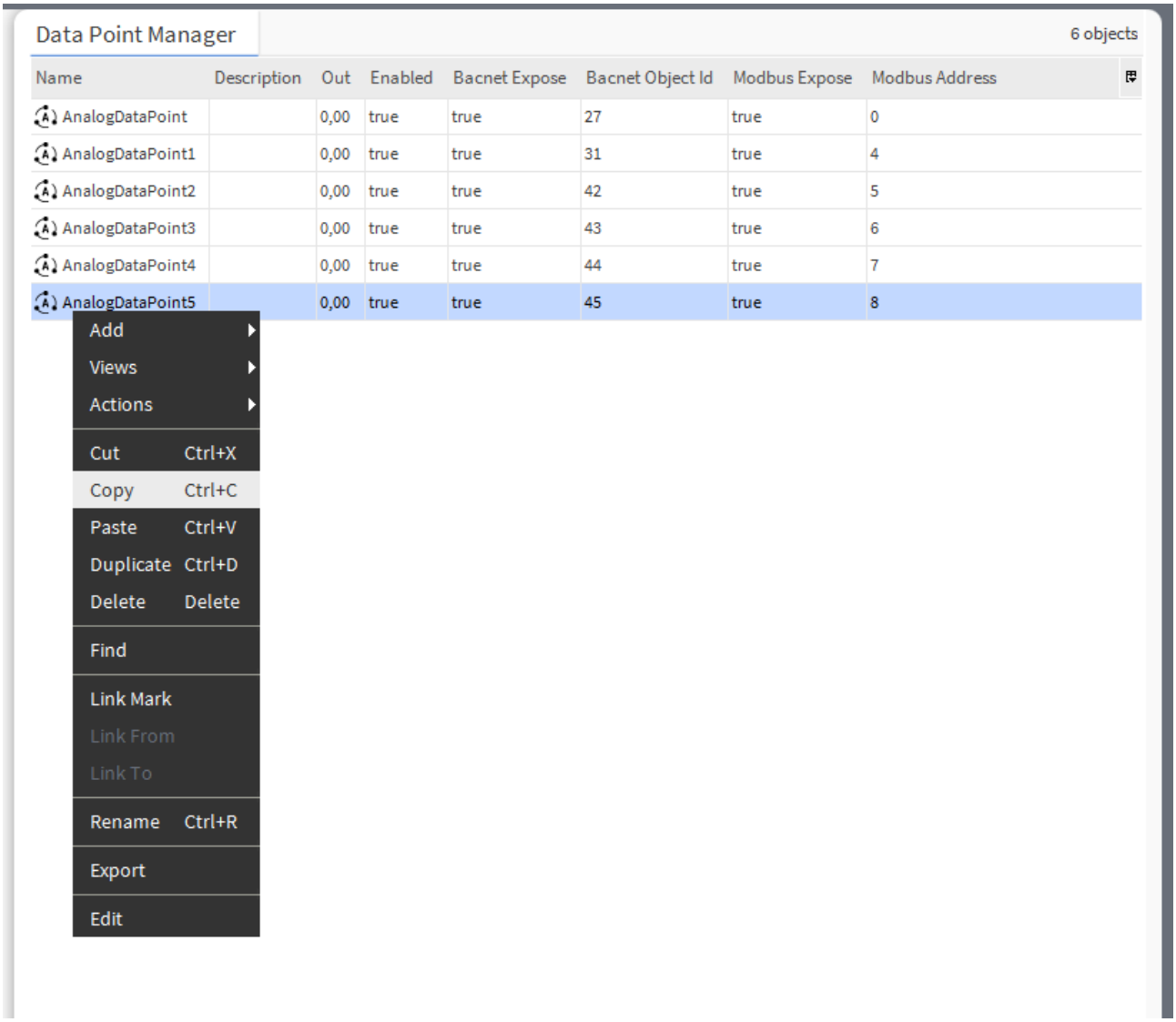


Figure 119. Context menu options for the Data Point

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.

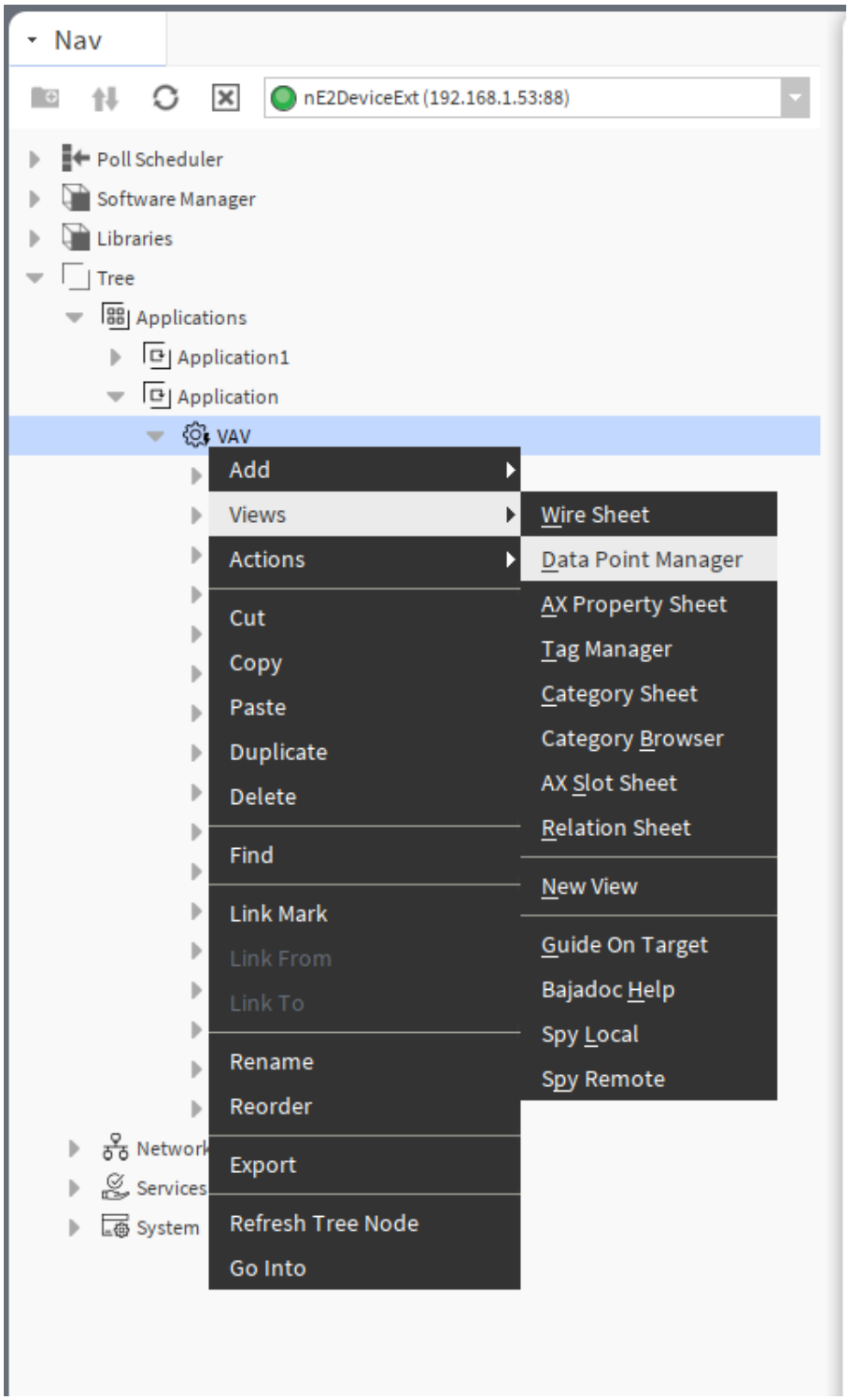


Figure 120. Accessing the Data Point Manager in the context menu

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 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

5.6.1 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.

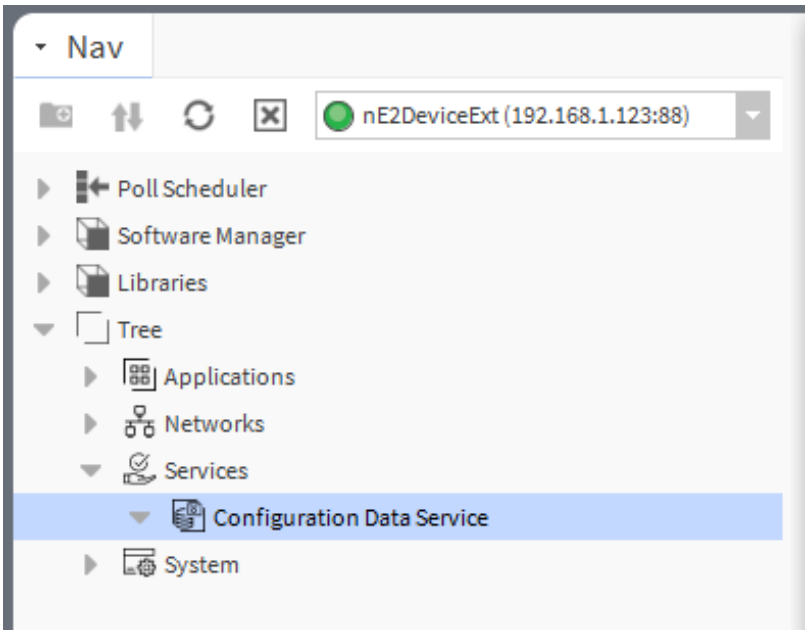


Figure 121. Configuration Data service in the nav tree

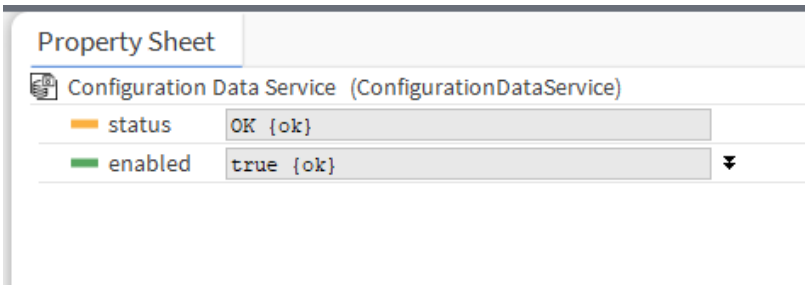
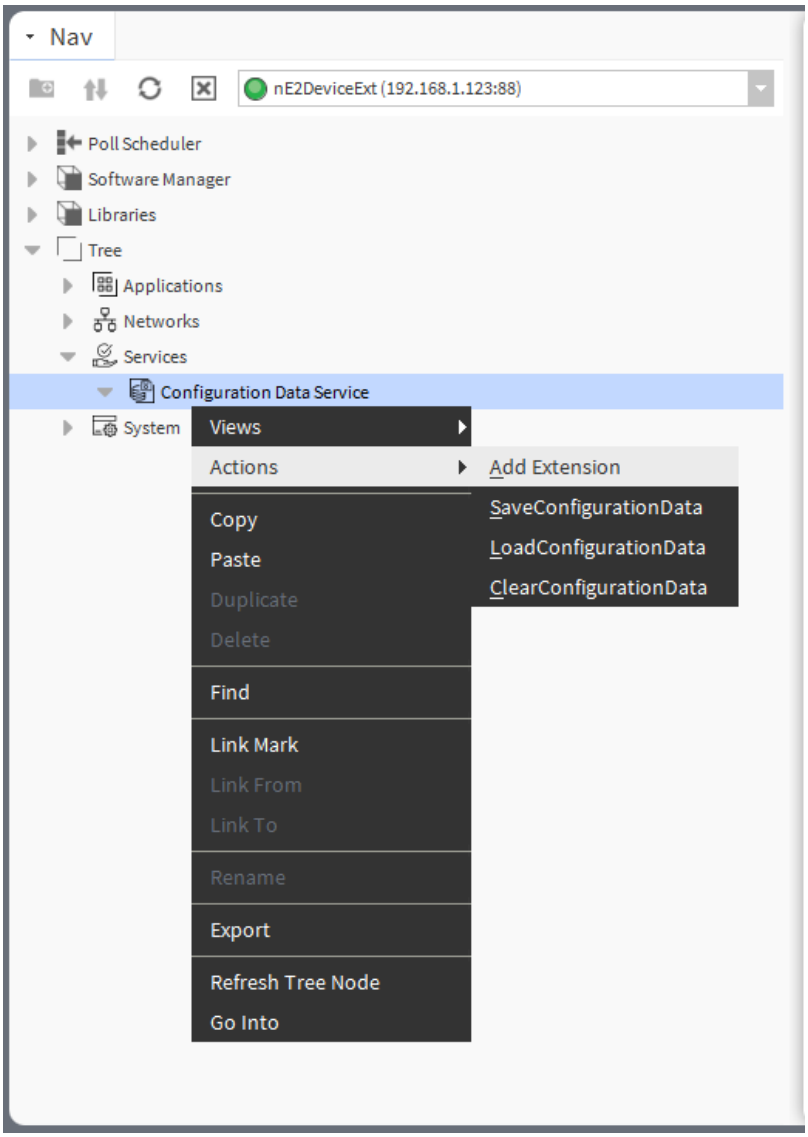


Figure 122. Configuration data service slots

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.



Configuration Data Service Manager

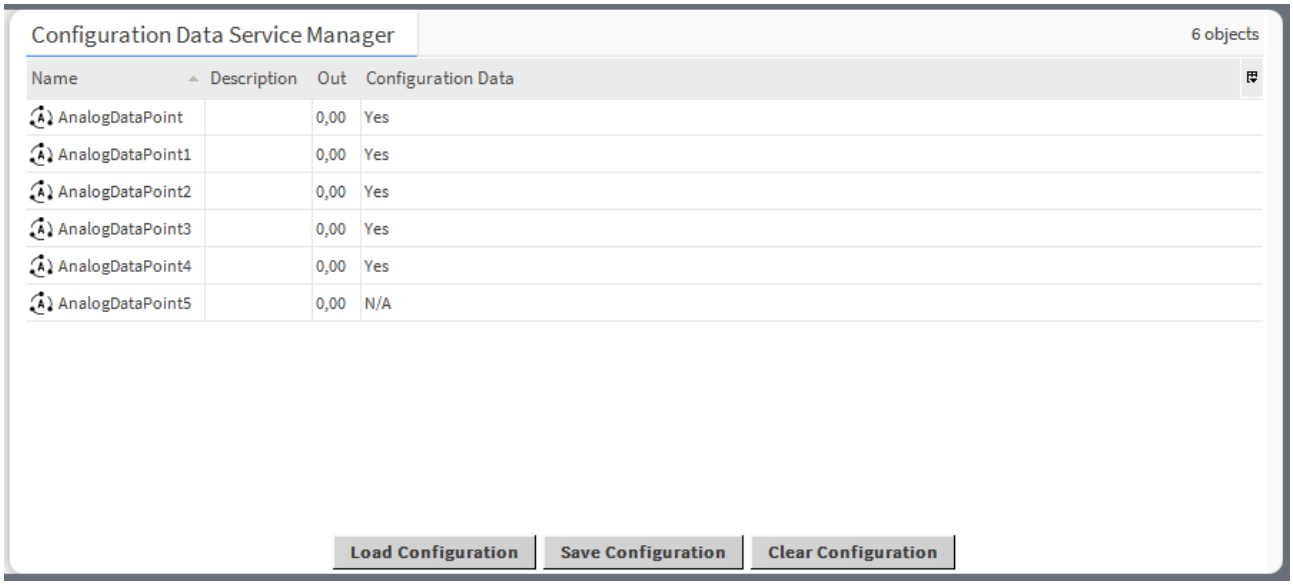


Figure 123. Configuration Data service manager

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 Local IO

5.7.1 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.

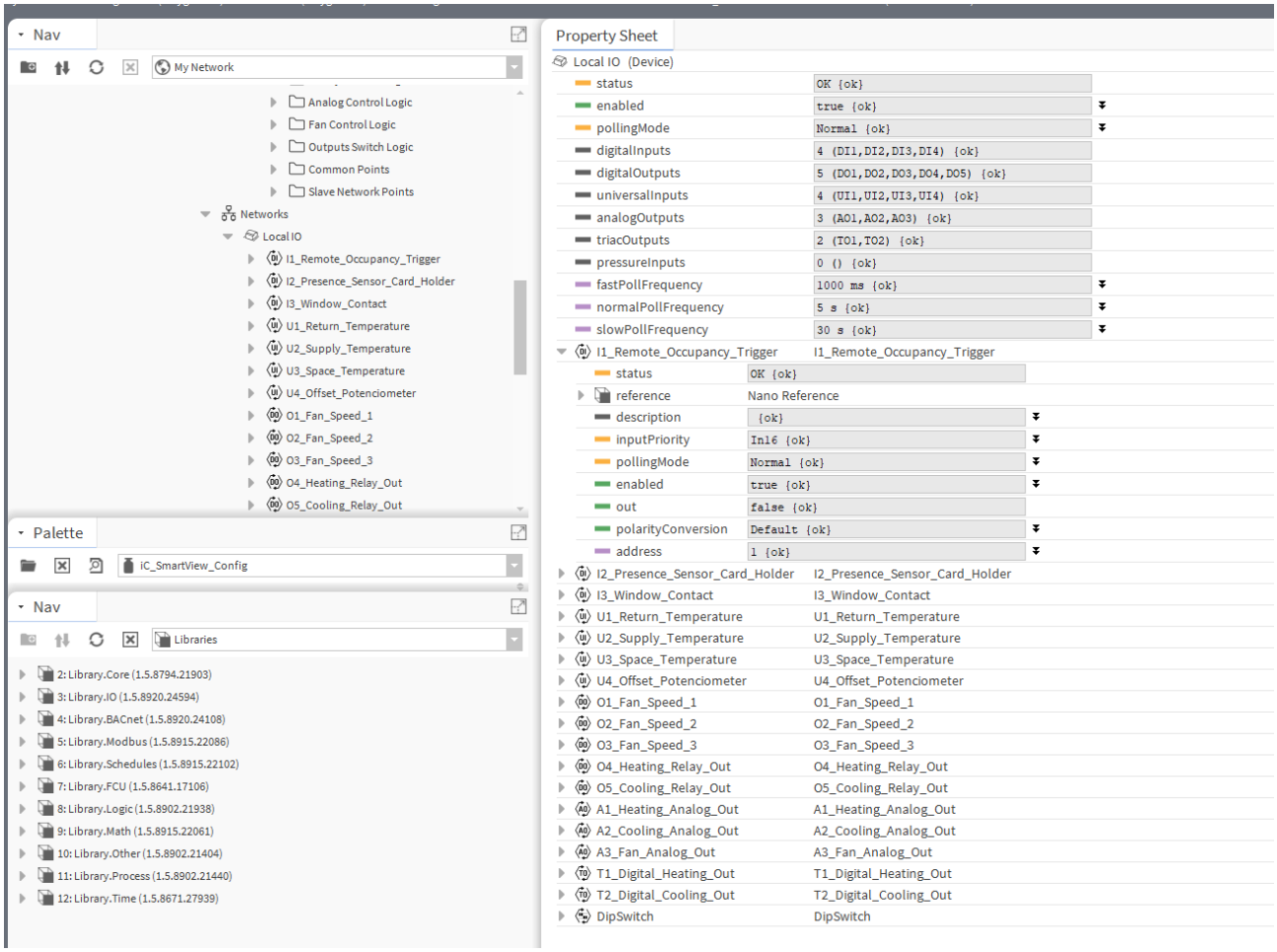


Figure 124. LocalIO components added to the Networks container

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

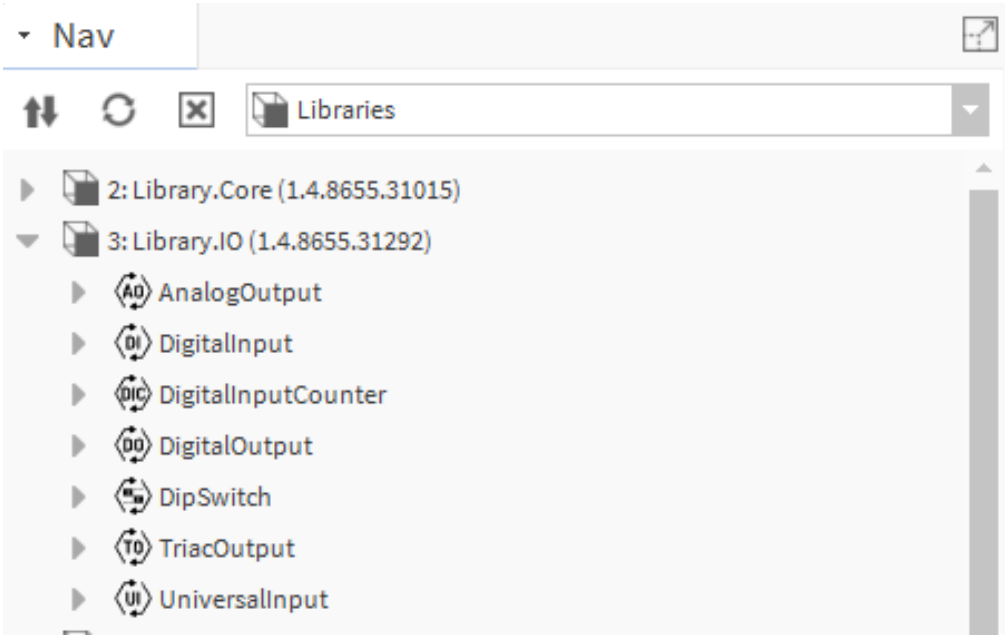


Figure 125. Library.IO

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

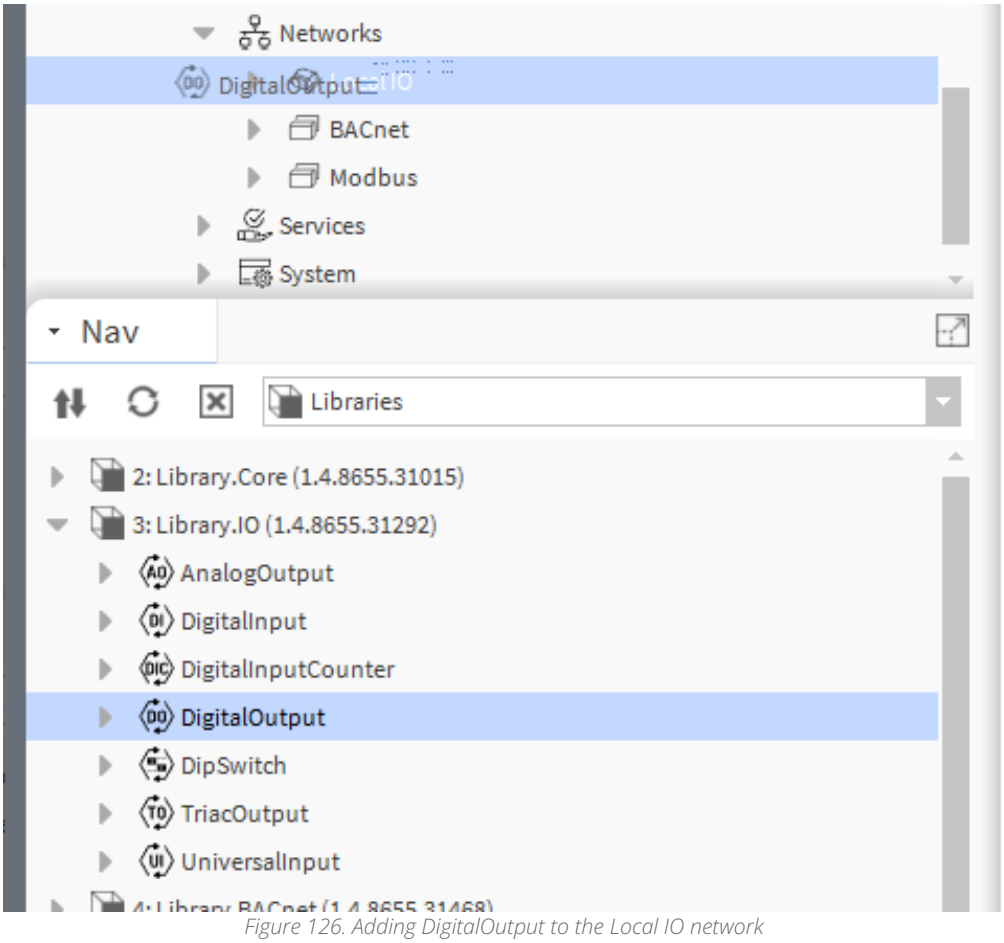


Figure 126. Adding DigitalOutput to the Local IO network

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.

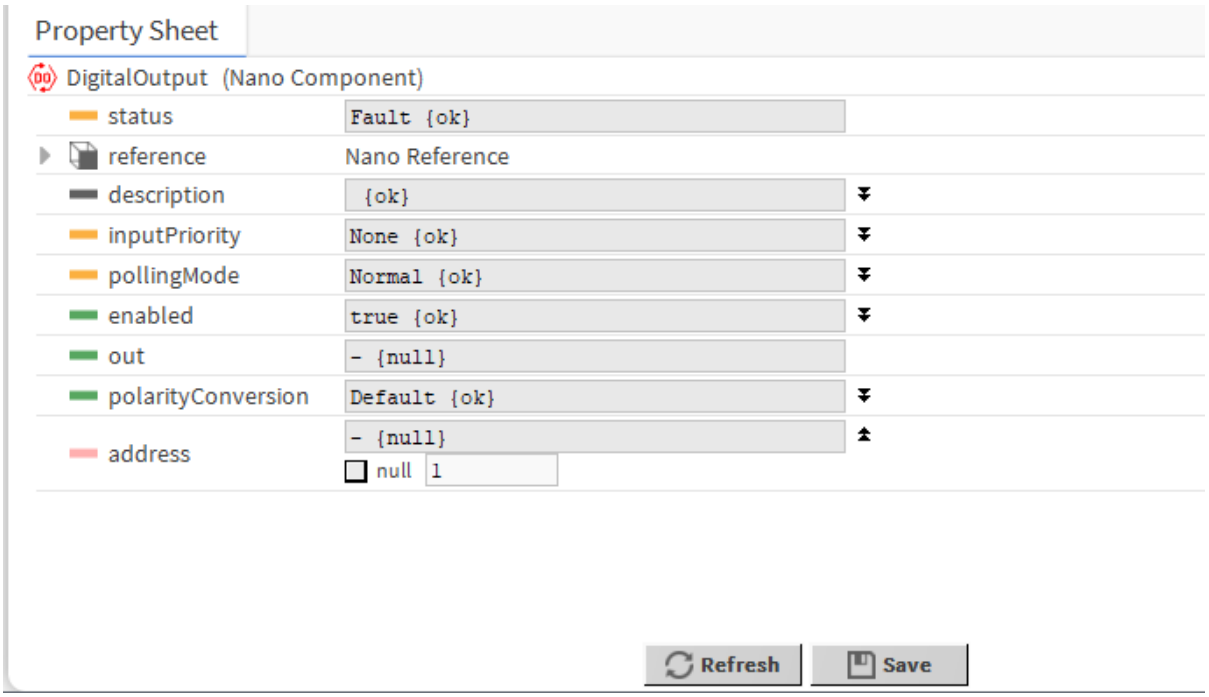


Figure 127. Setting the IO point address

5.7.2 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;
- number;
- enabled or disabled state.

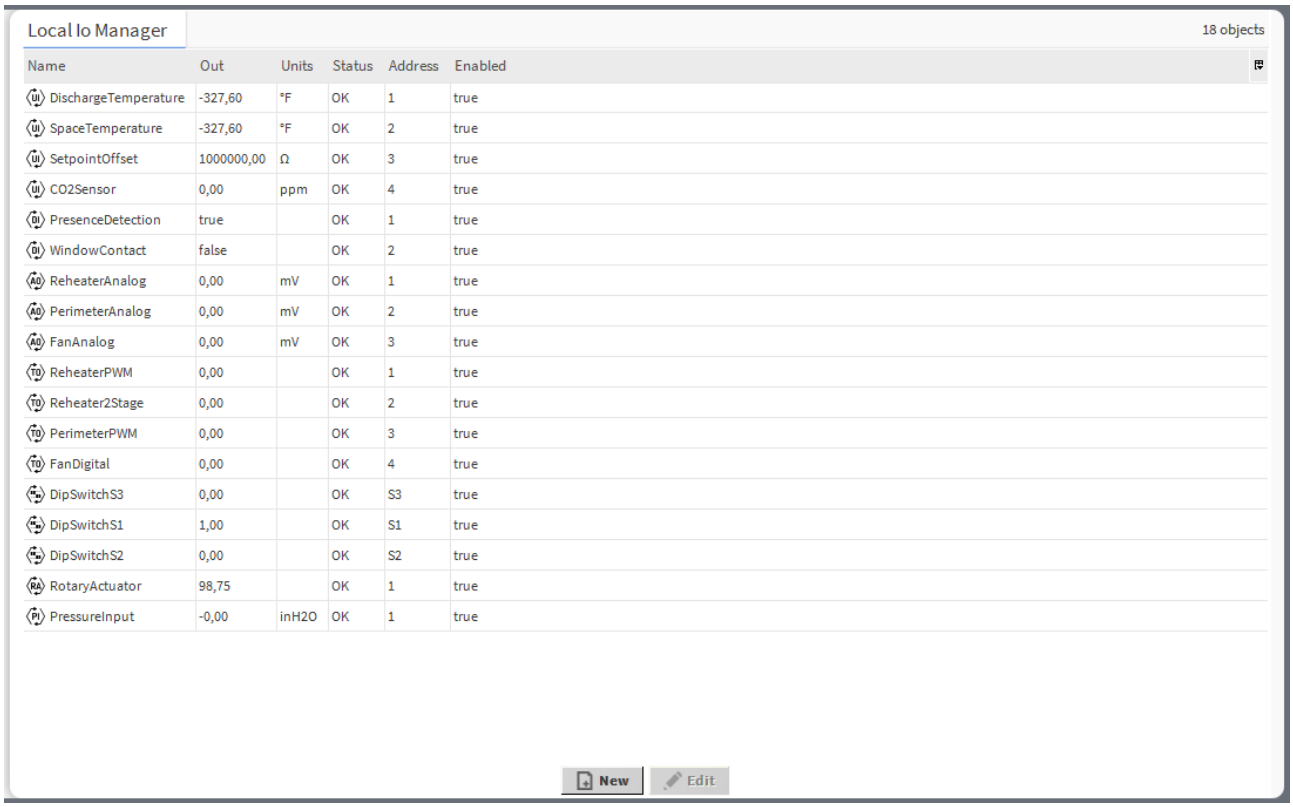


Figure 128. LocalIO Manager

In the LocalIO Manager, it is possible to:

- add local I/O components:

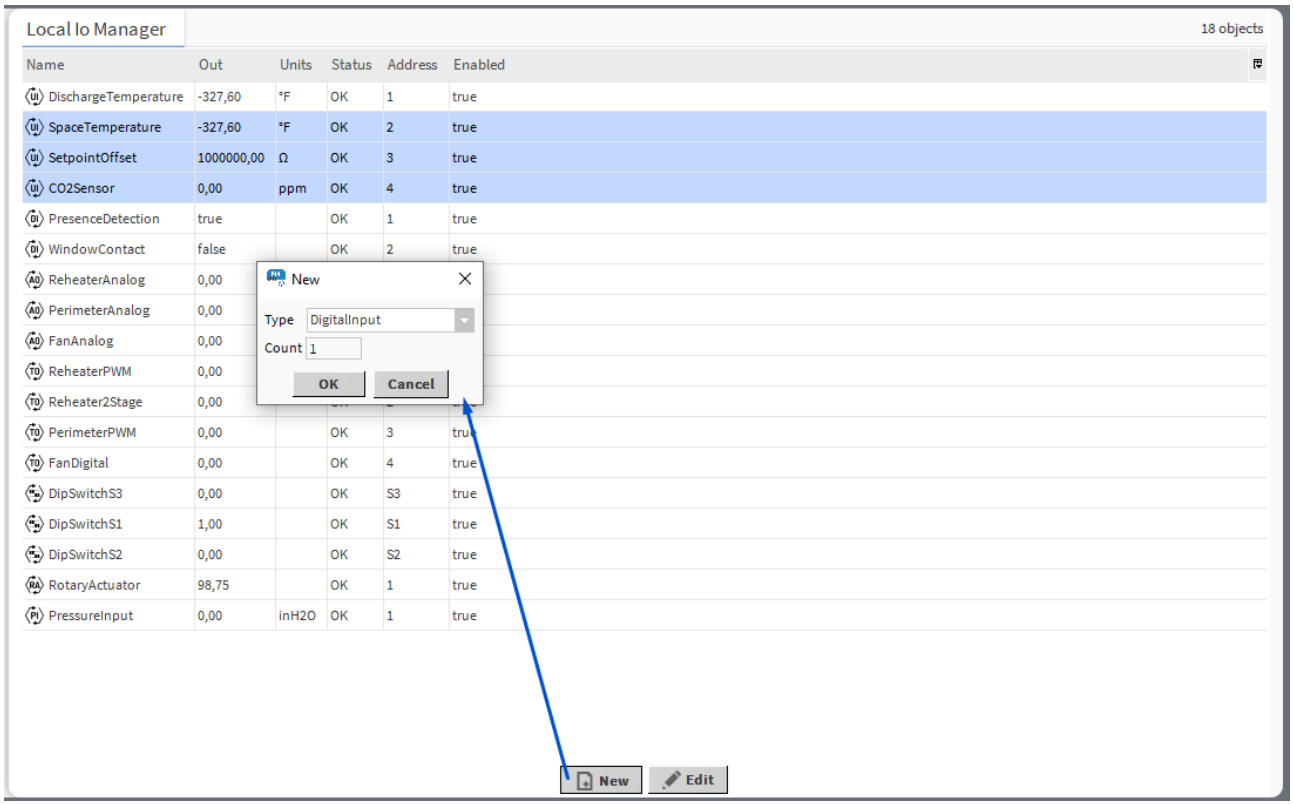


Figure 129. Adding I/O points in the LocalIO Manager

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

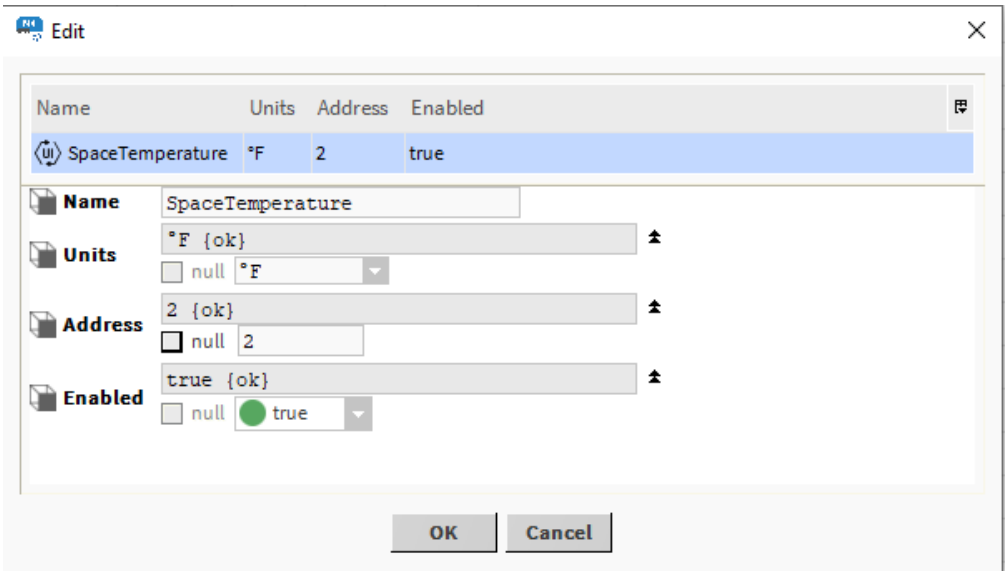


Figure 130. Editing pop-up

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.

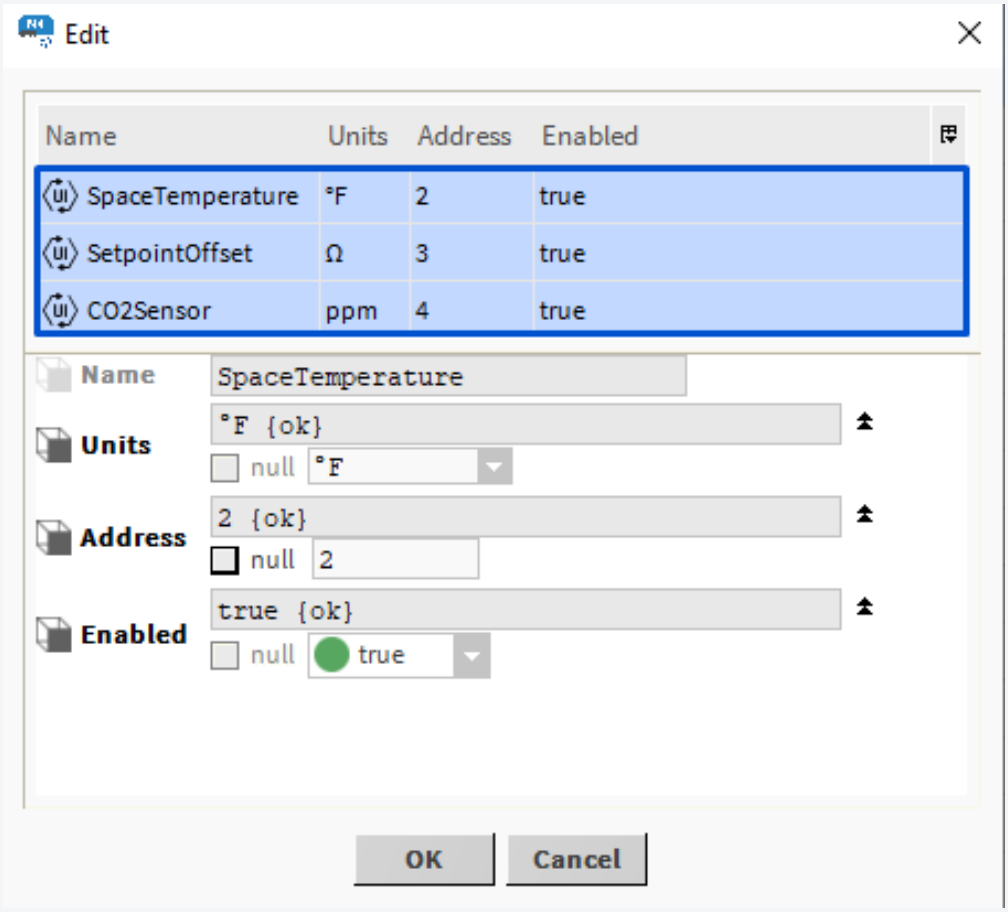


Figure 131. Editing of multiple I/O points

- copy/duplicate/remove I/O components:

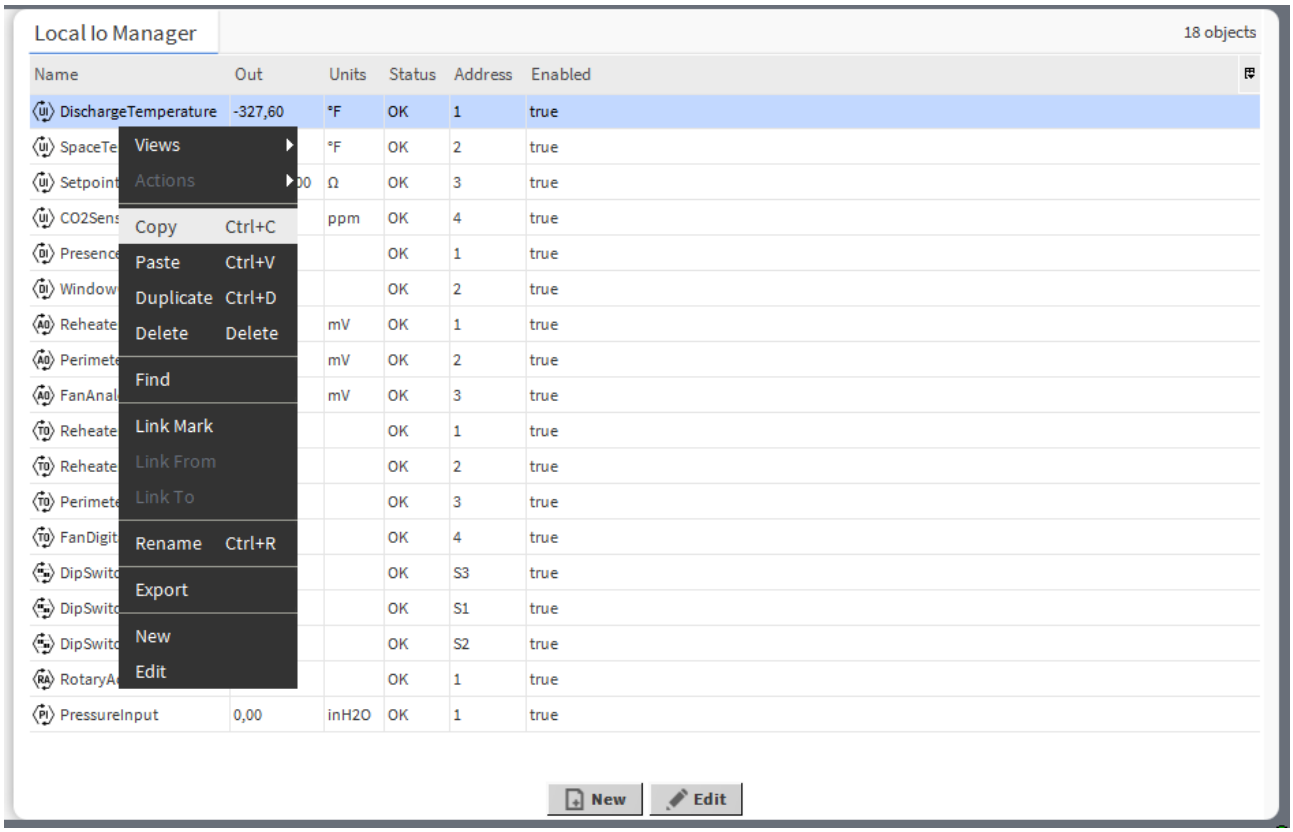


Figure 132. Context menu options for local I/Os

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.

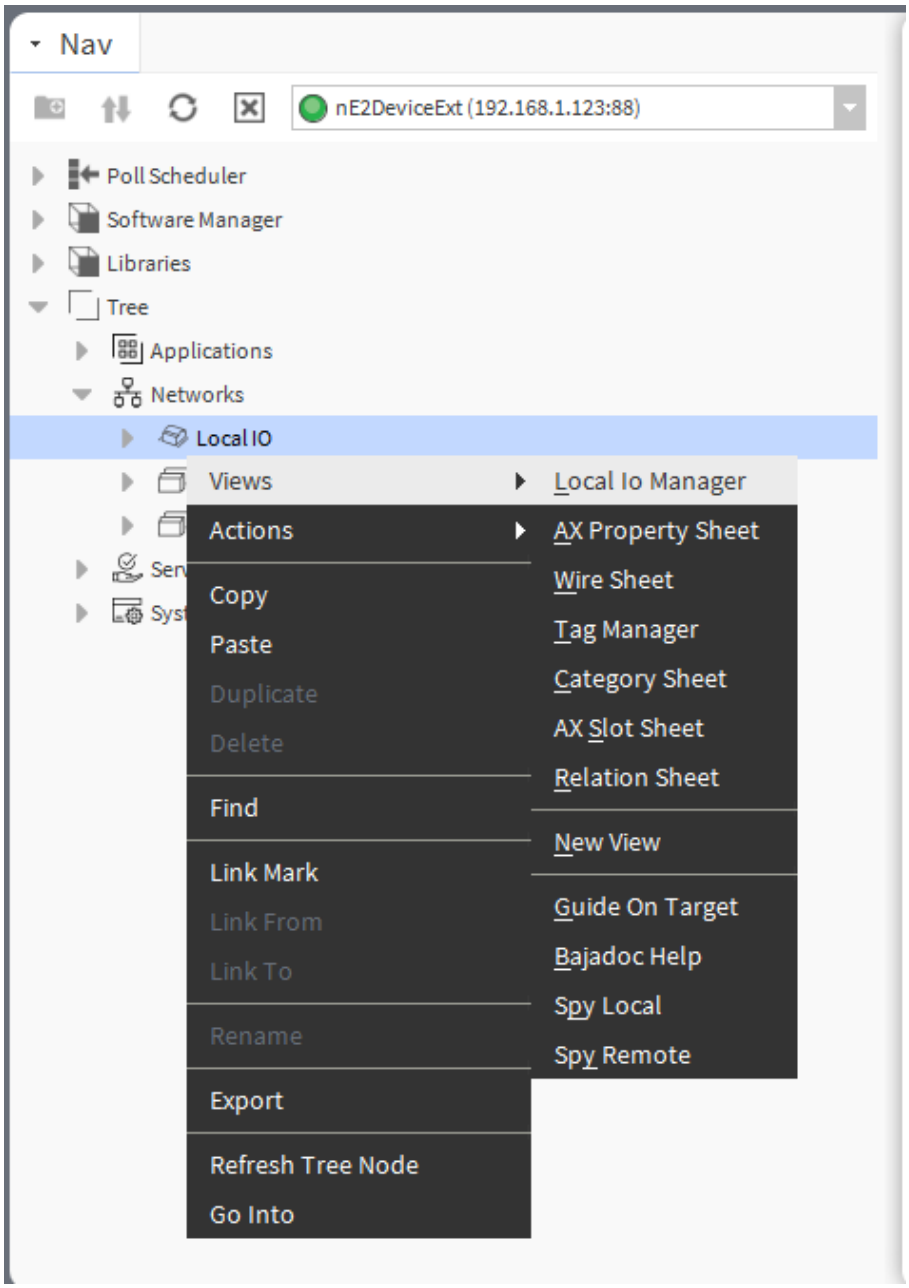


Figure 133. Accessing the LocalIO Manager from the context menu

5.8 Linking

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

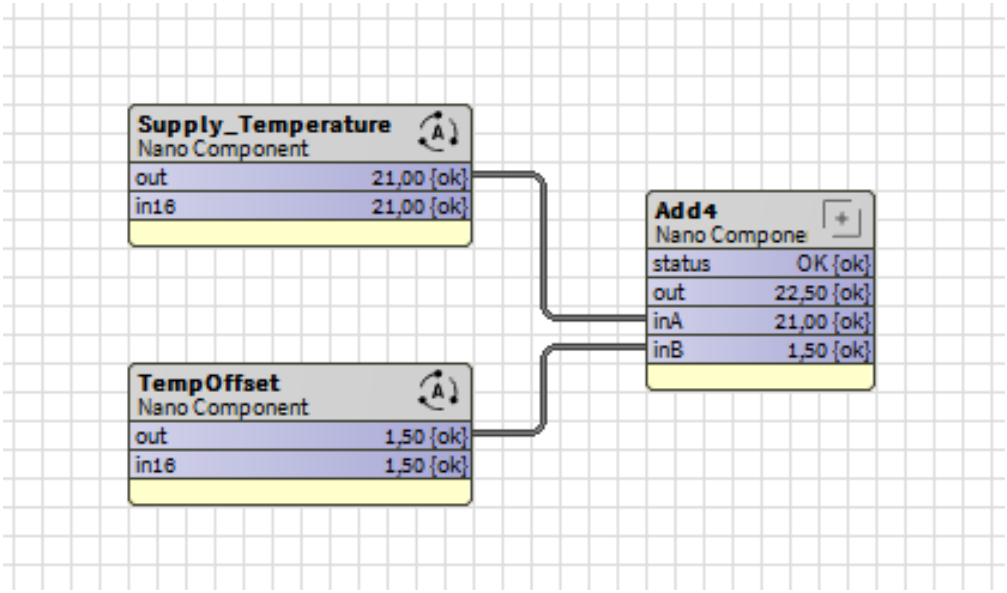


Figure 134. Linked components

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.

5.8.1 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.

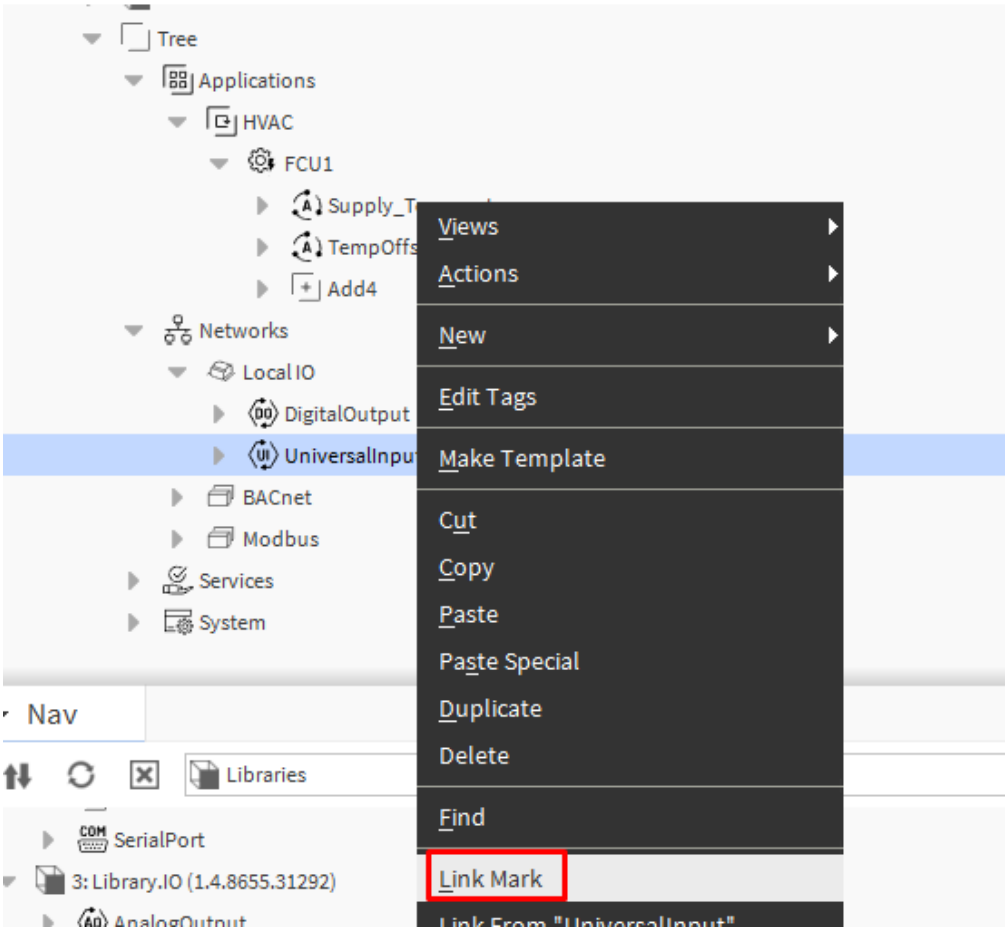


Figure 135. Link Mark option in the network point - UniversalInput

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

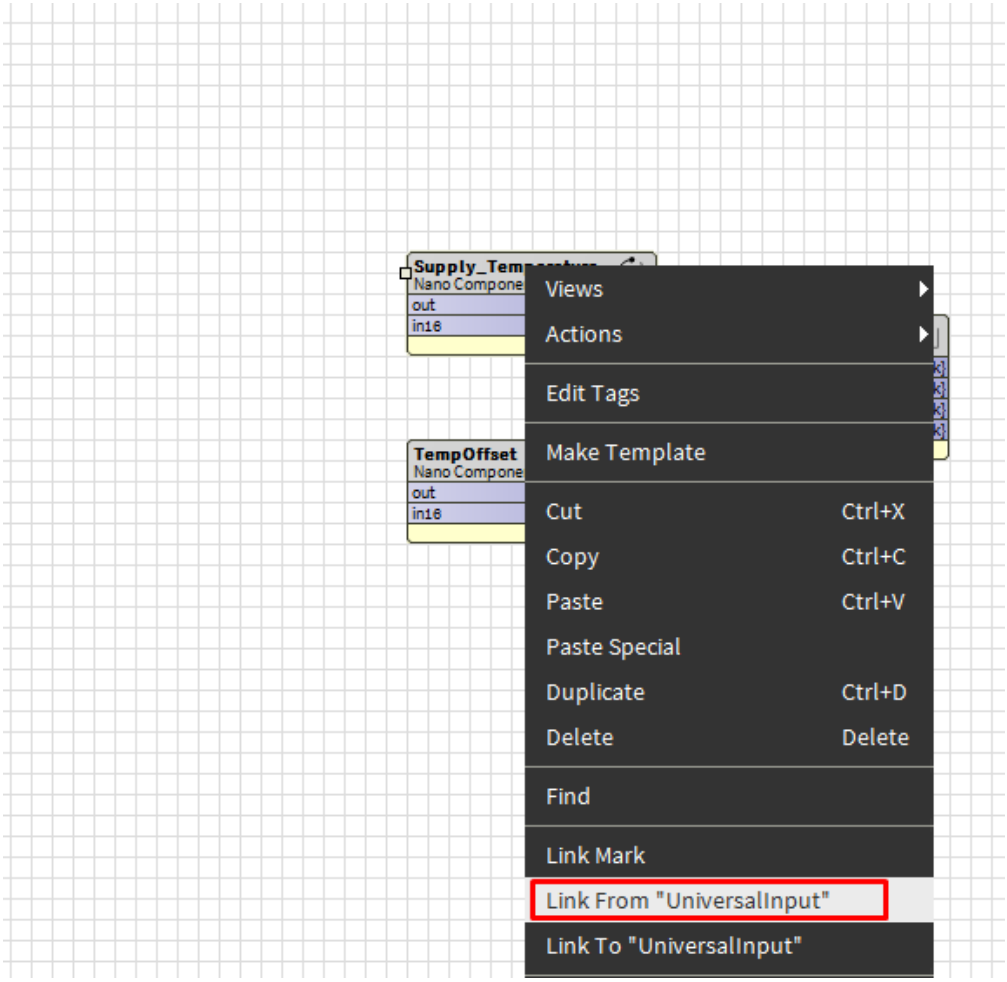


Figure 136. Link From option

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

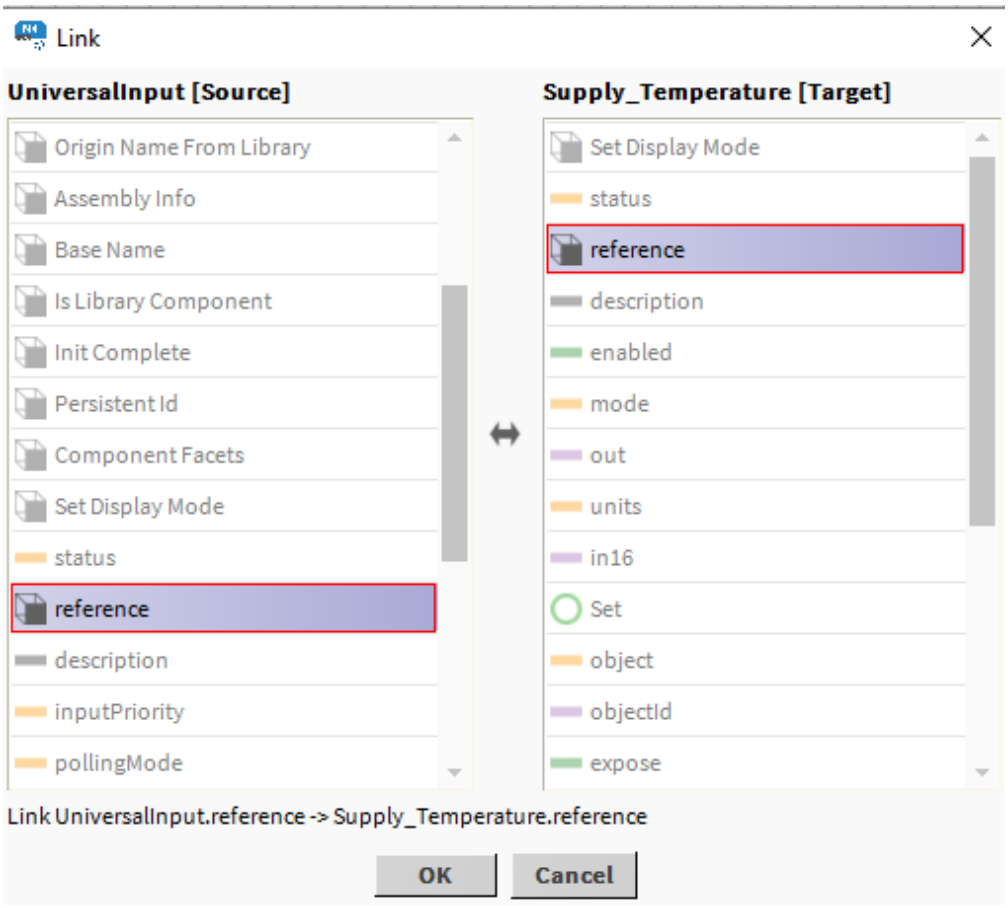


Figure 137. Creating reference between points

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

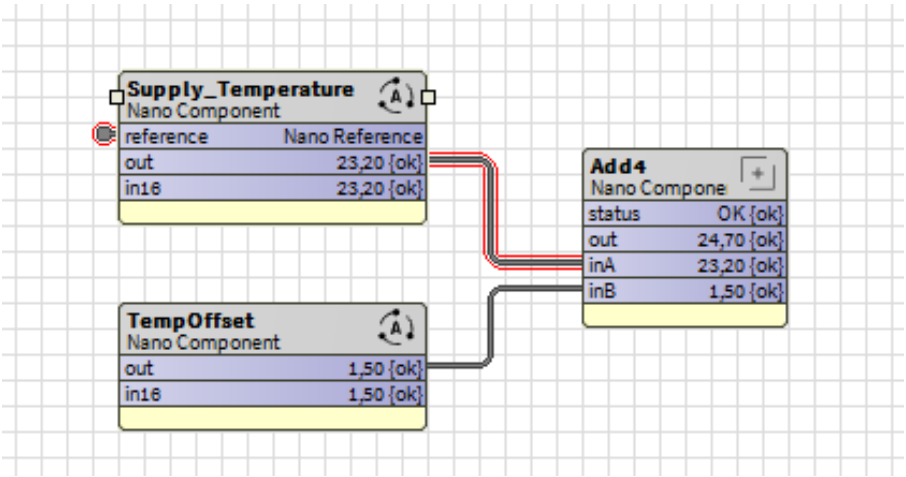


Figure 138. Data Point with a reference

5.9 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

5.9.1 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.

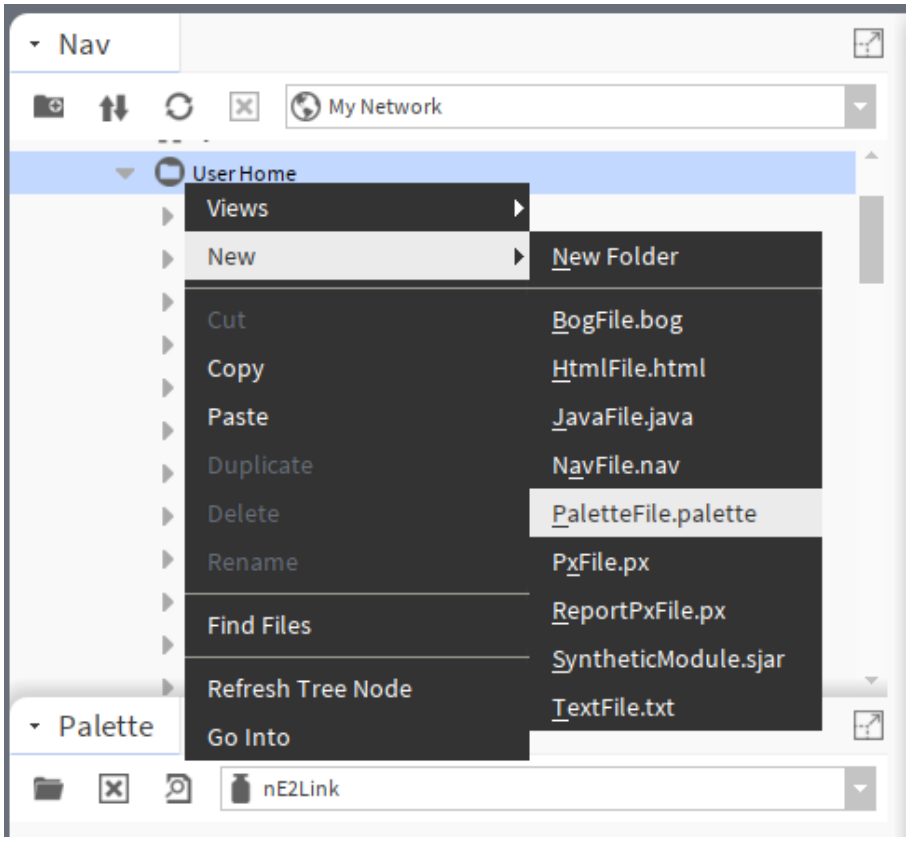


Figure 139. Creating a new palette

5.9.2 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.

5.9.3 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).

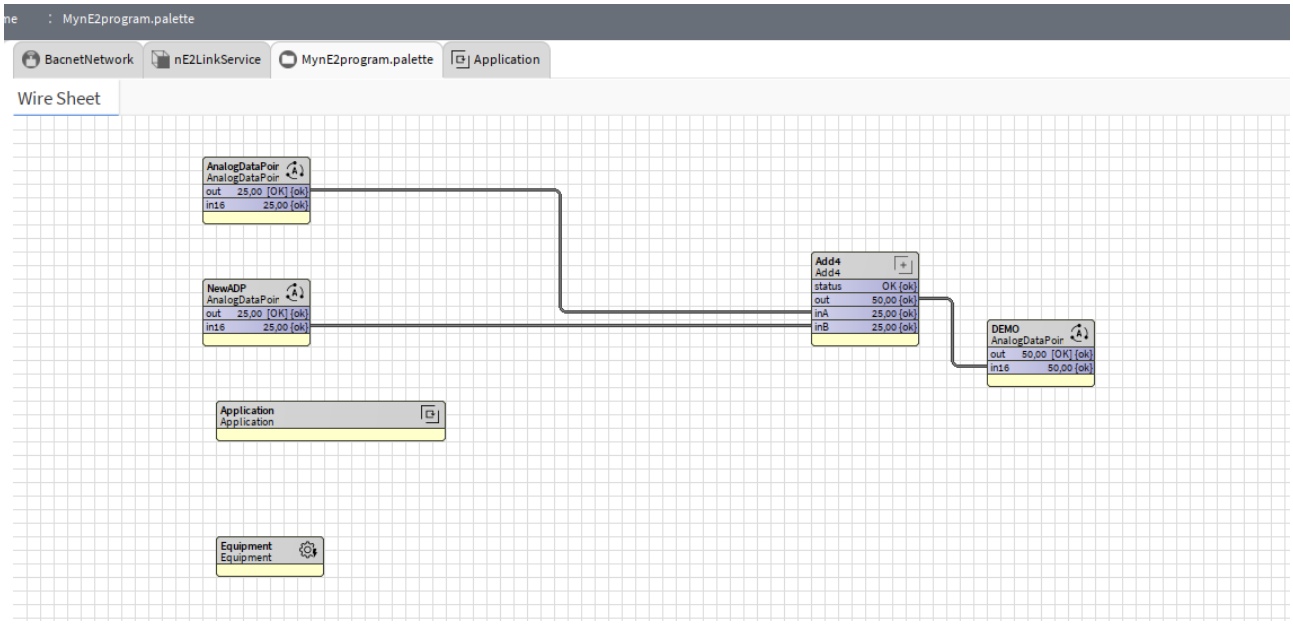


Figure 140. Copied Application and Equipment on the wire sheet

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.

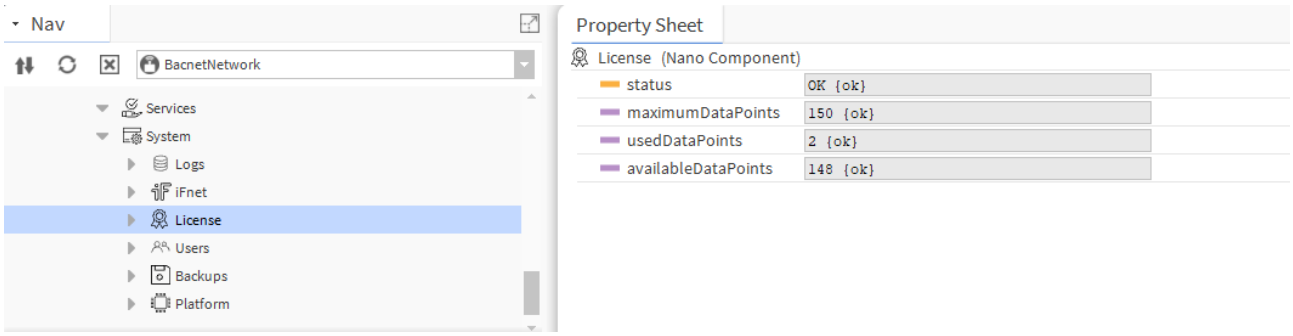


Figure 141. Number of available Data Points visible in the License component

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.

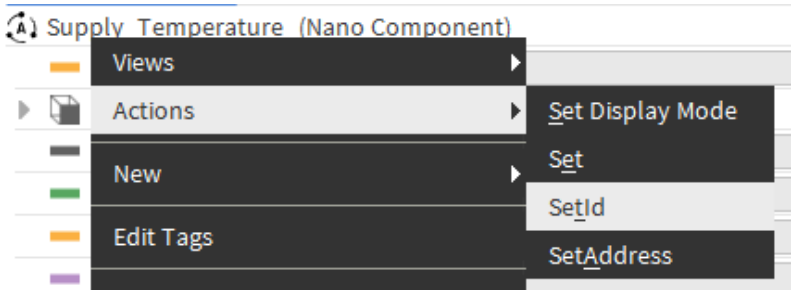


Figure 142. Setting ObjectID to the Data Point

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.

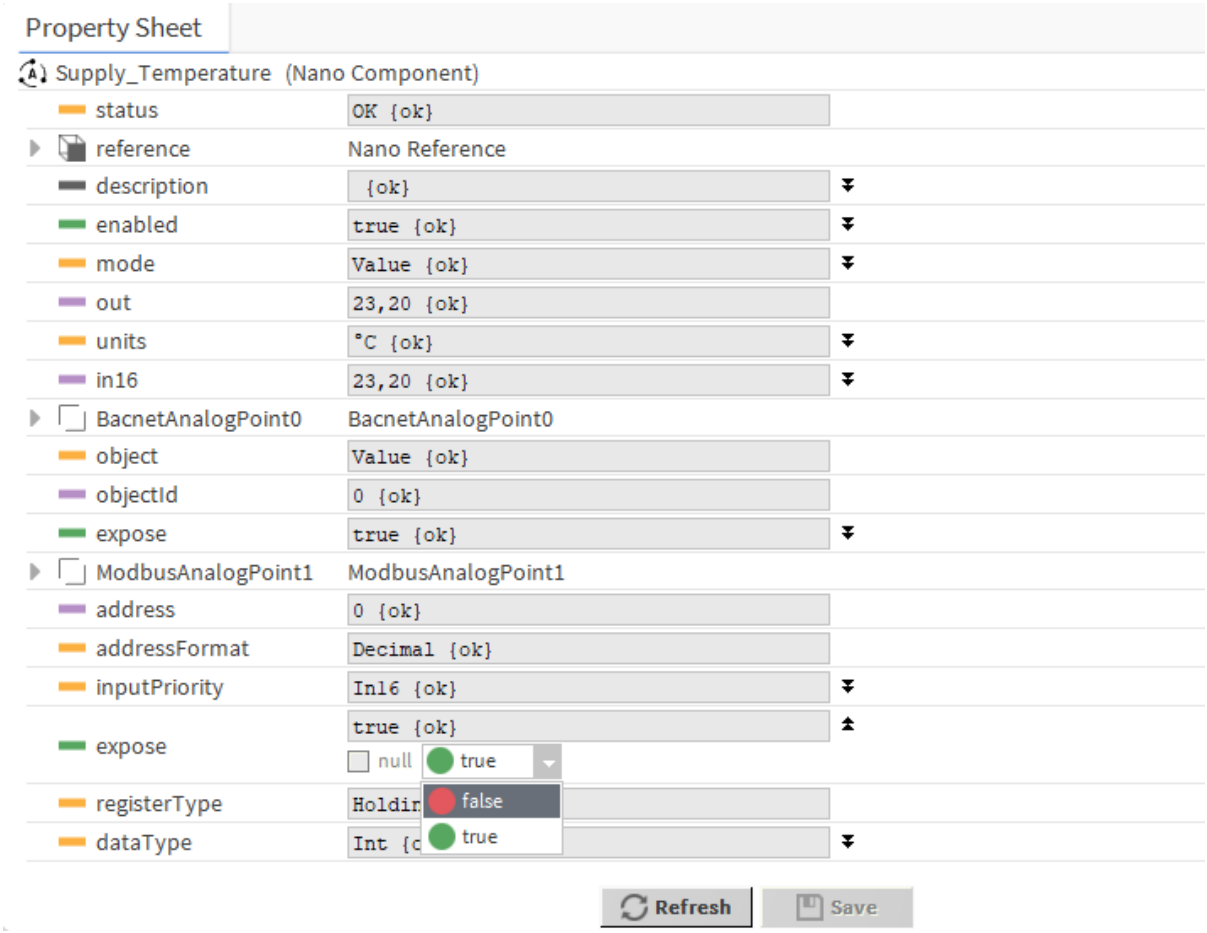


Figure 143. Possibility to disable the exposition on Modbus or BACnet in the Data Point's extension

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

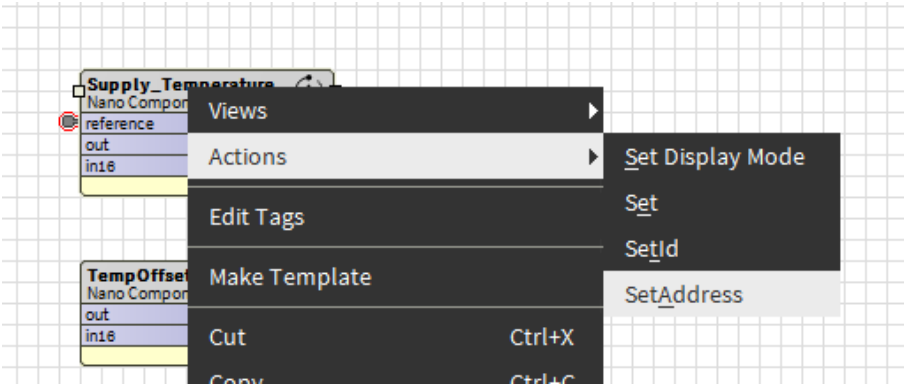


Figure 144. The SetAddress action

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.

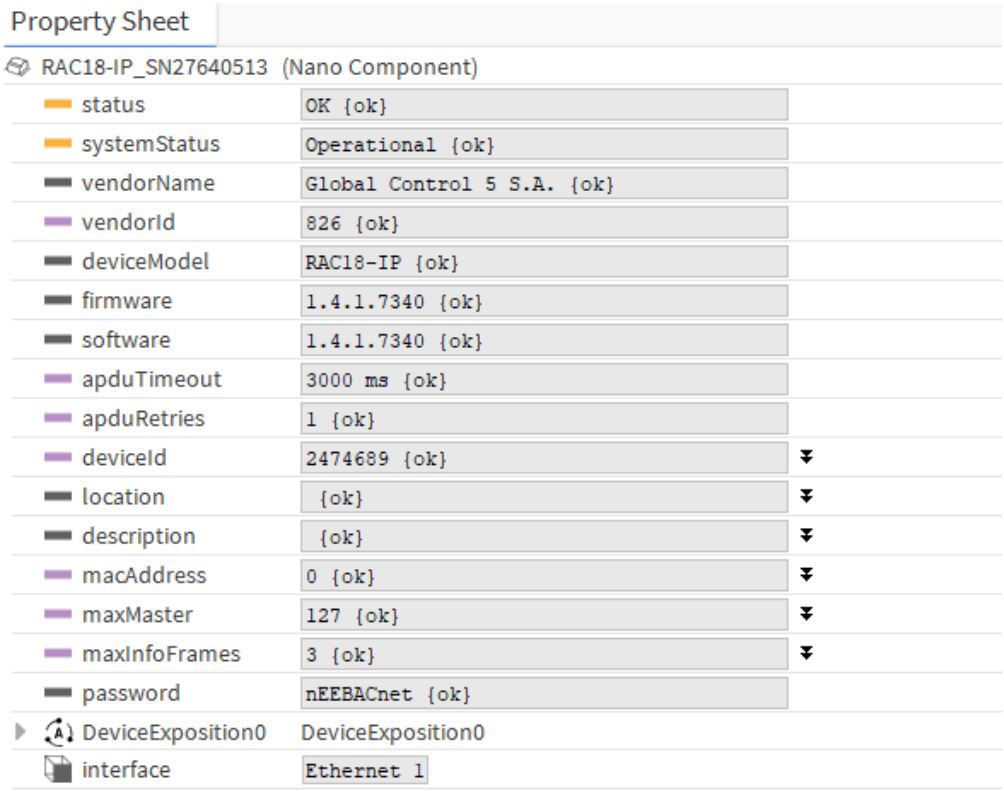


Figure 145. LocalDevice property sheet

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.

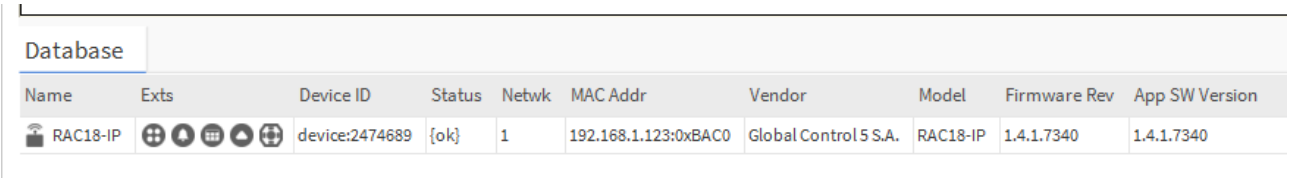


Figure 146. RAC18-IP integrated to Niagara over the BACnet network

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

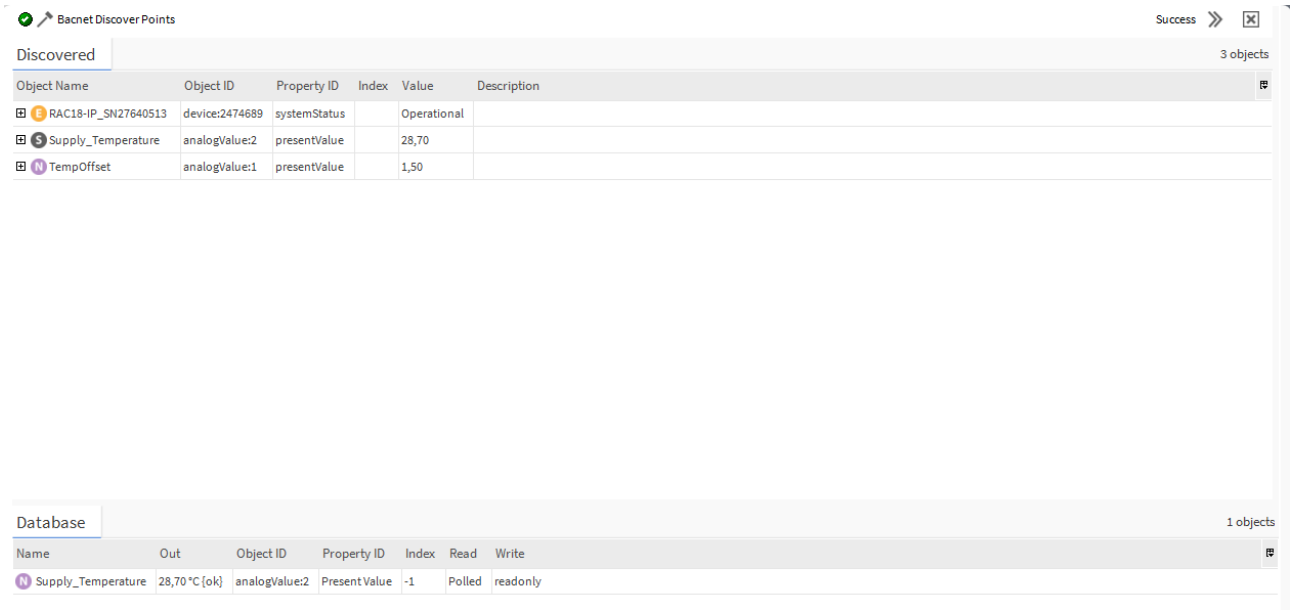


Figure 147. Points added to the Niagara BACnet database

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

7 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:

7.1 Wire Sheet

(supporting the nano EDGE ENGINE context menu):

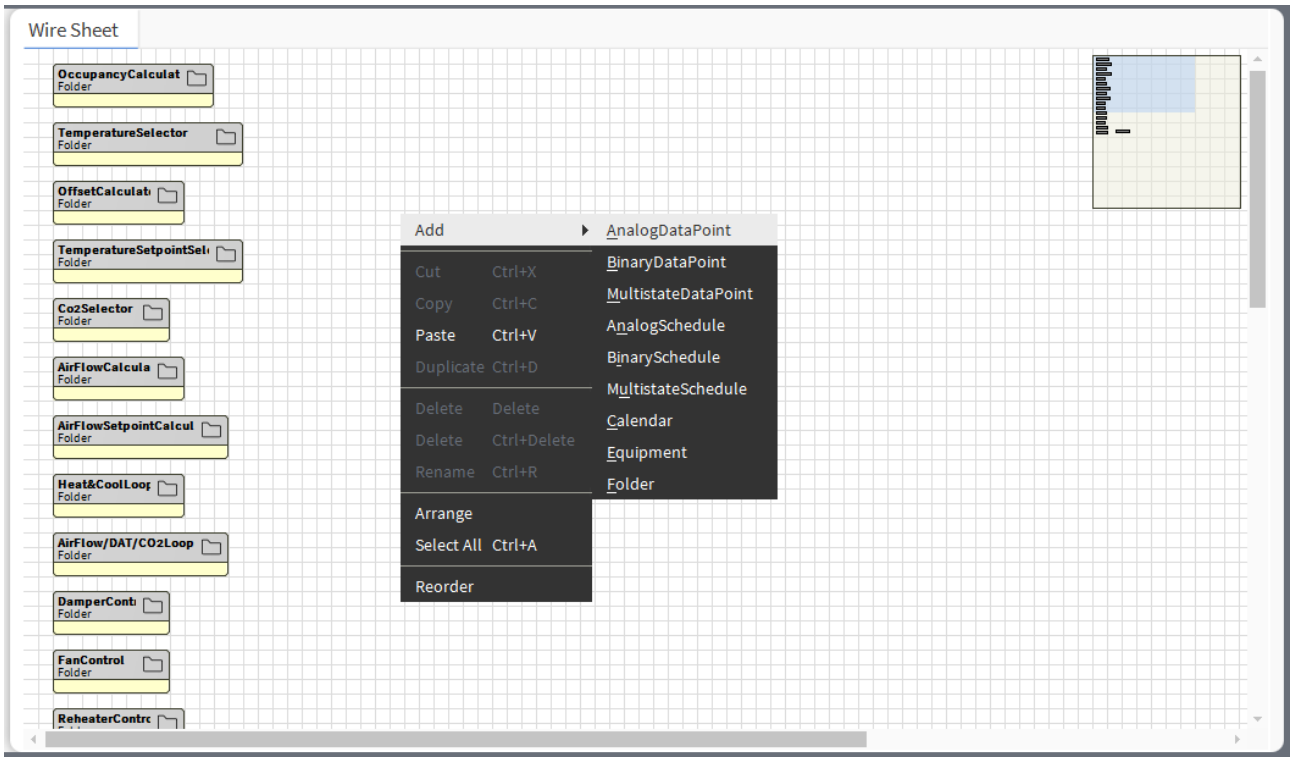


Figure 148. Wire sheet with the nano EDGE ENGINE context menu

7.2 AX Property Sheet

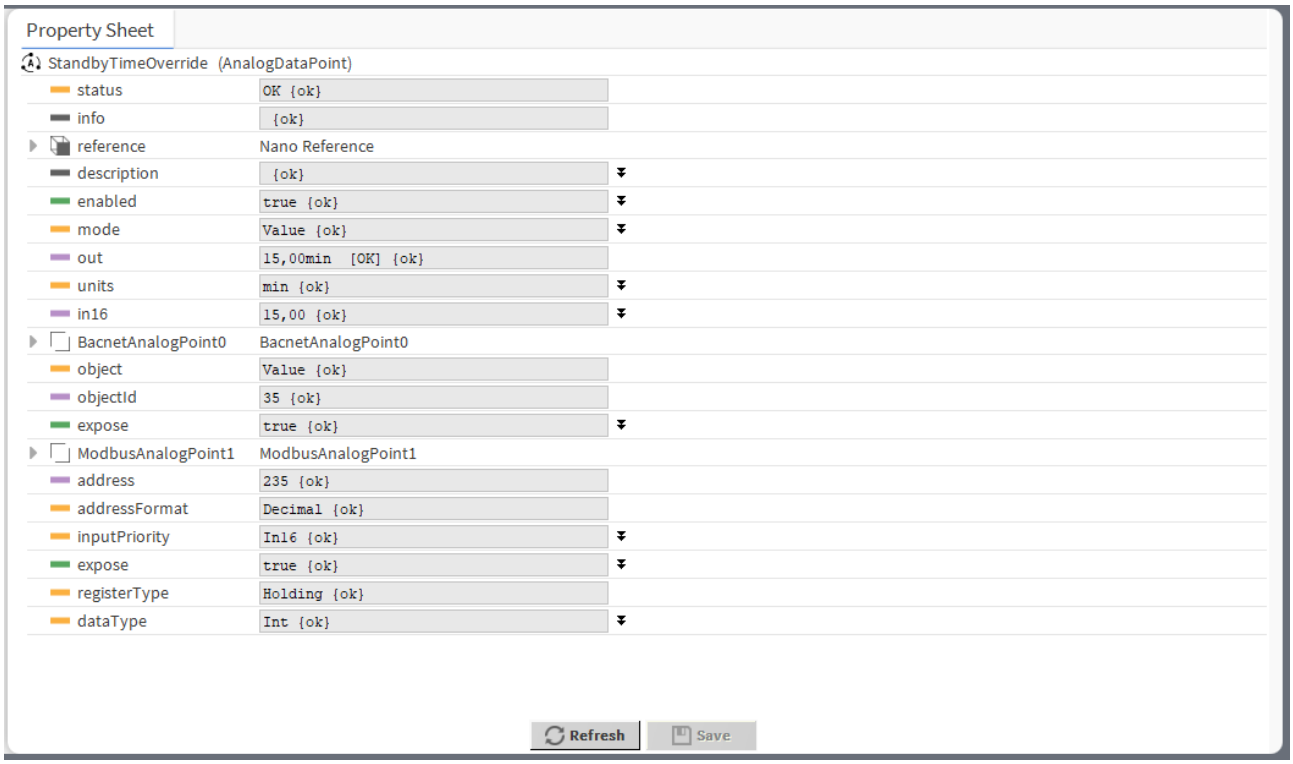


Figure 149. Property sheet view for the AnalogDataPoint

7.3 Relation Sheet

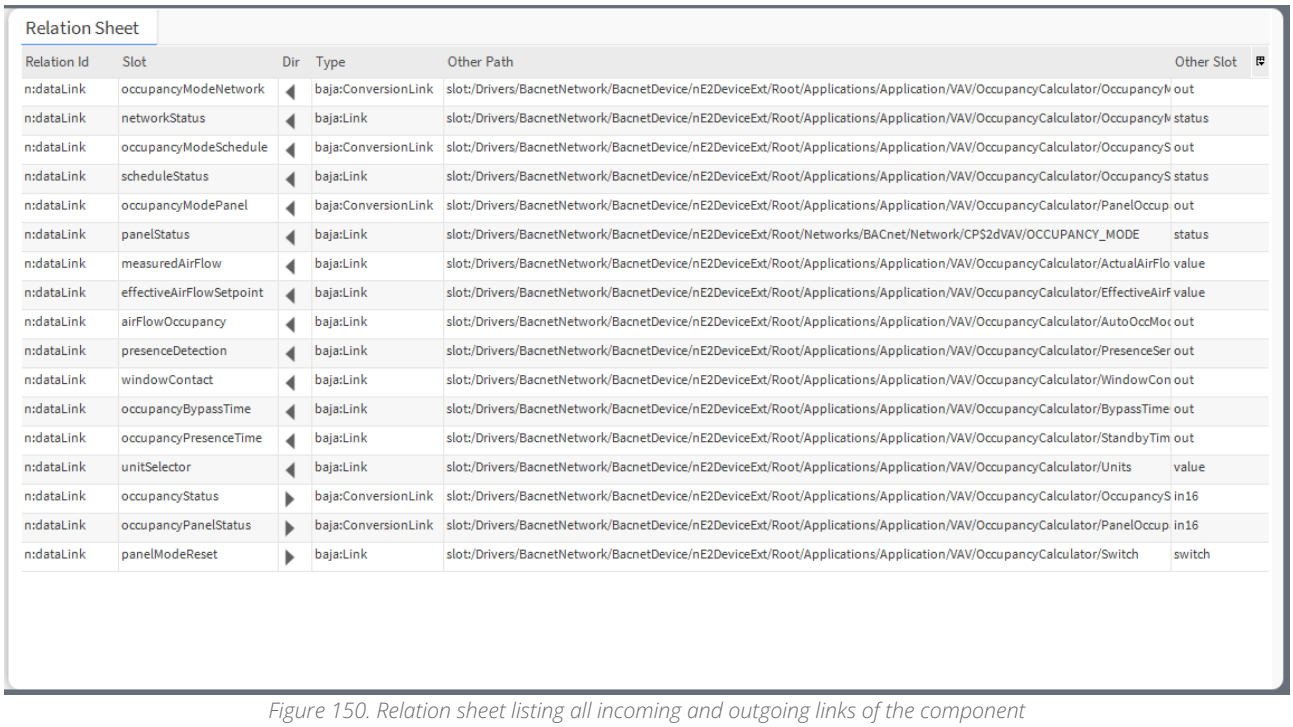


Figure 150. Relation sheet listing all incoming and outgoing links of the component

Note

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

8 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.

8.1 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:

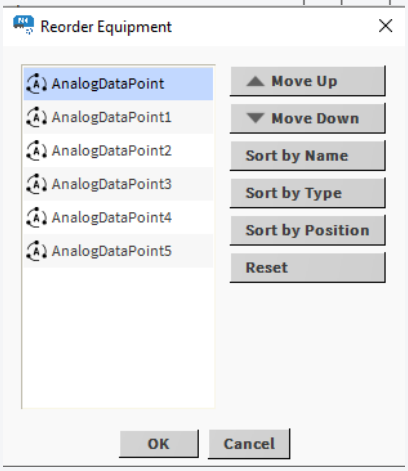


Figure 151. Reordering pop-up always lists children components of the one the option was invoked on

- **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:

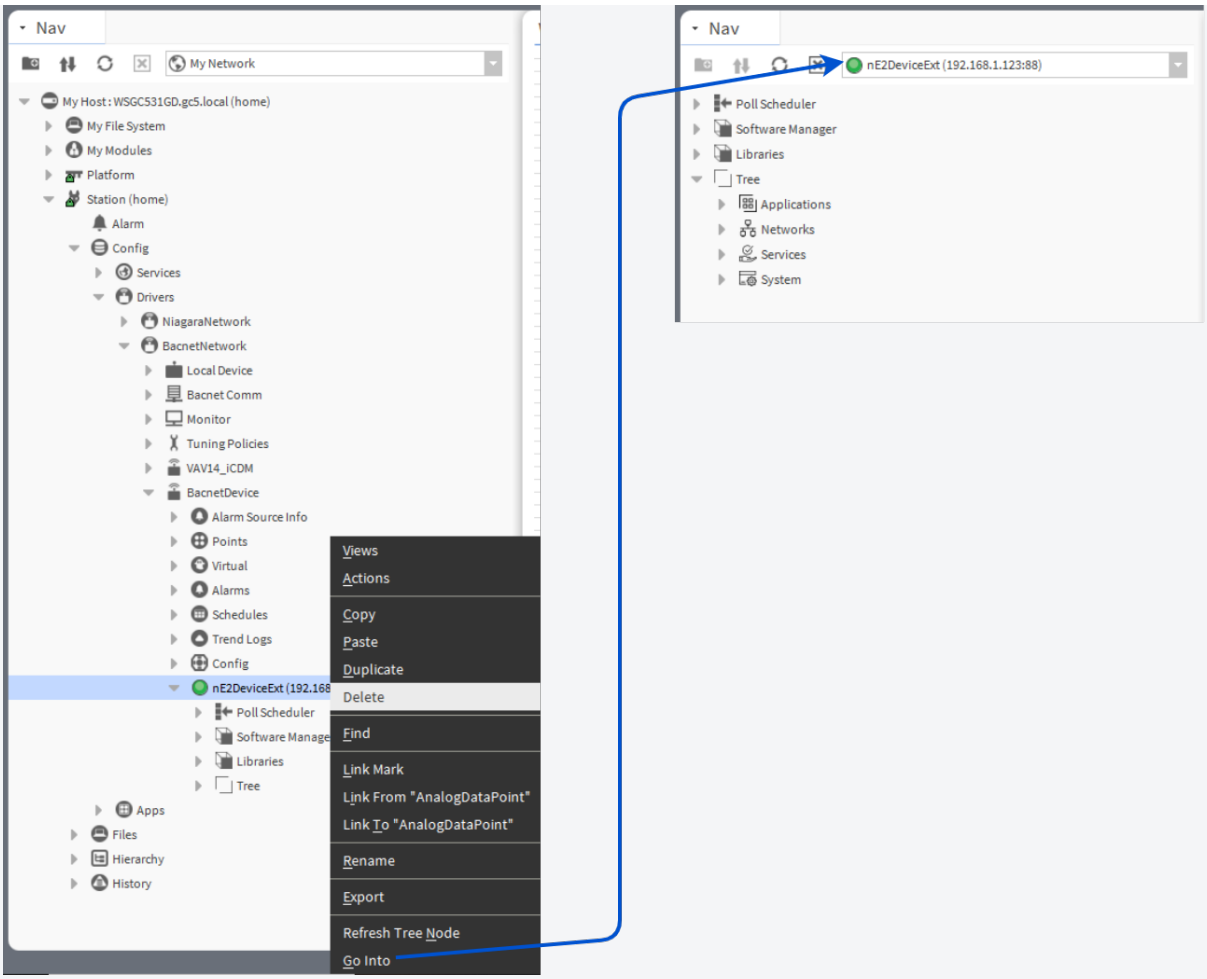


Figure 152. Using the Go Into option to simplify the tree structure

8.2 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

9 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).

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.

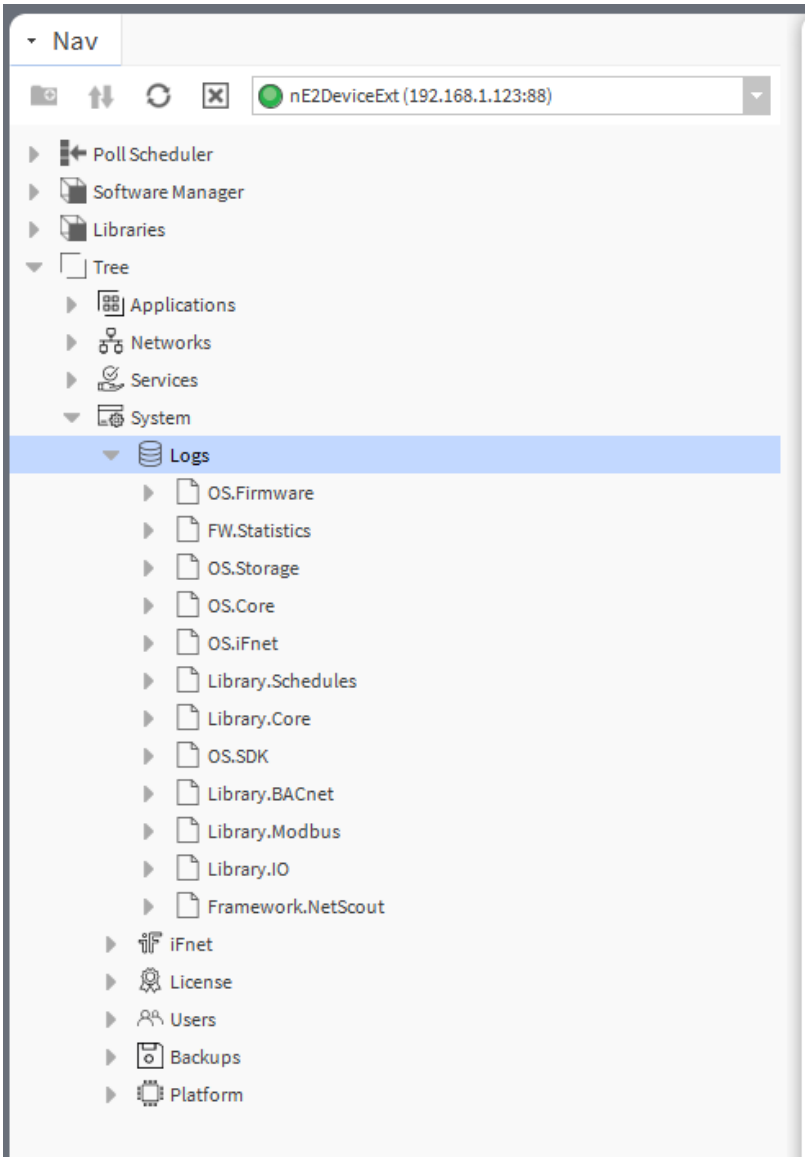


Figure 153. Logs

The Logs component has one slot:

- Default Log Level: assigns the importance of logs to be registered. Log levels are listed by level of details (Debug - most detailed, Critical - least detailed). If the default 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;
    - Available settings:
        - 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.).

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.

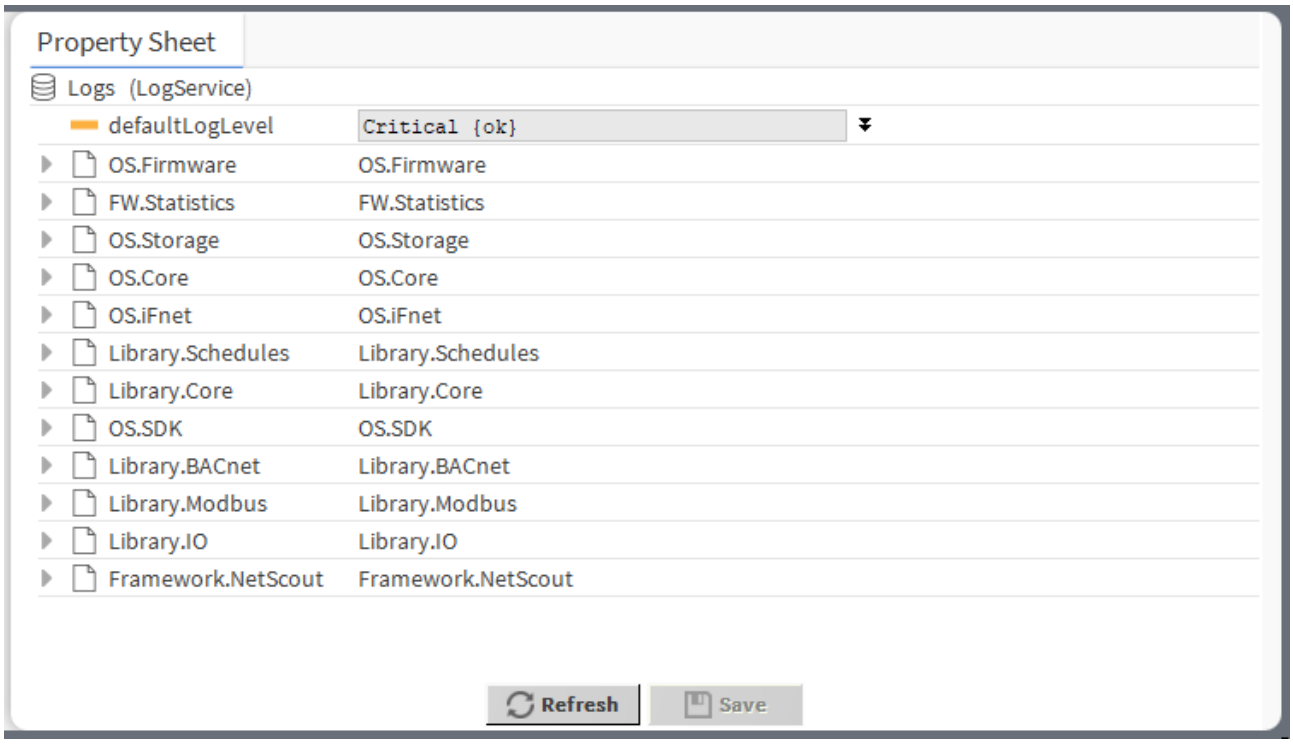


Figure 154. Logs' slot defaultLogLevel

Note

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

9.1 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.

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

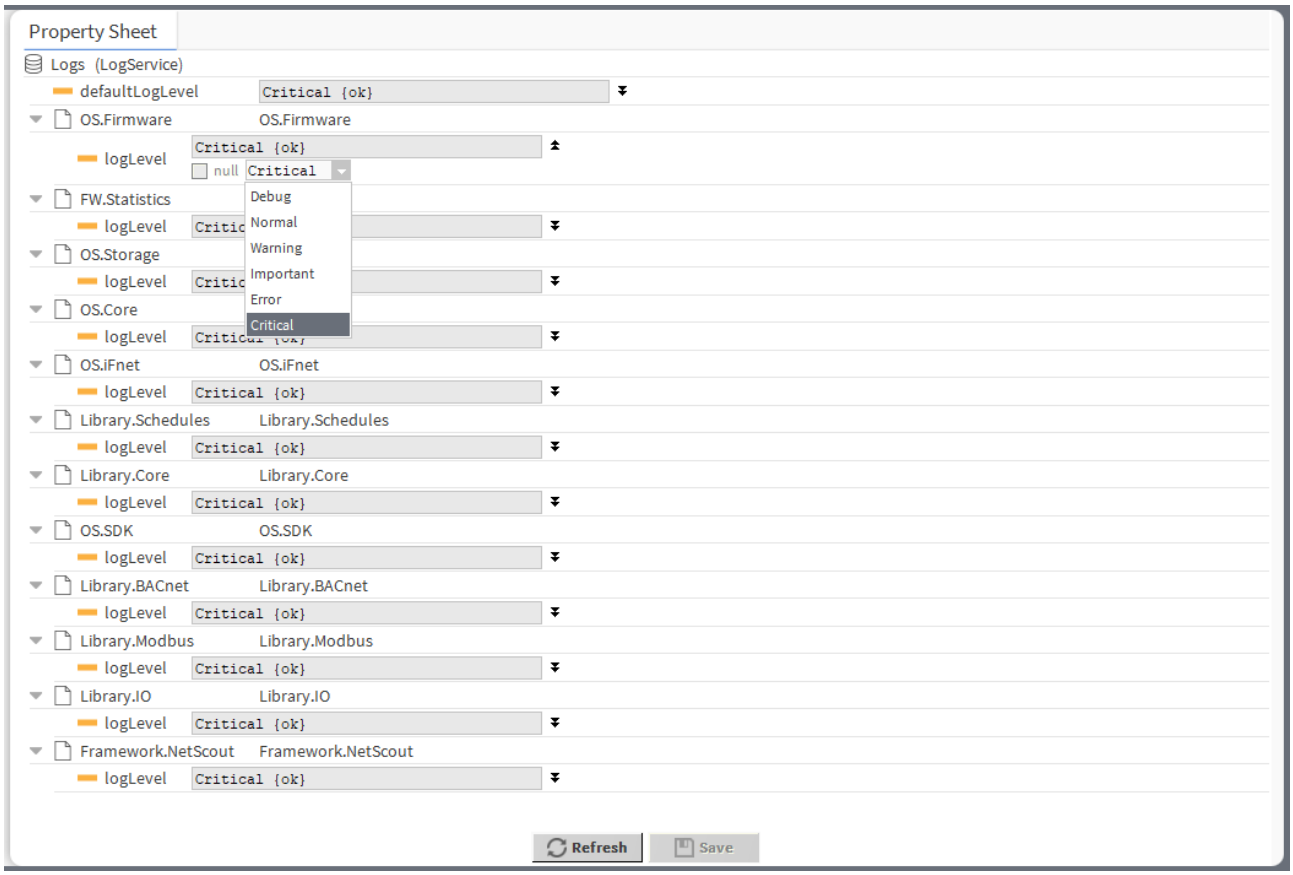
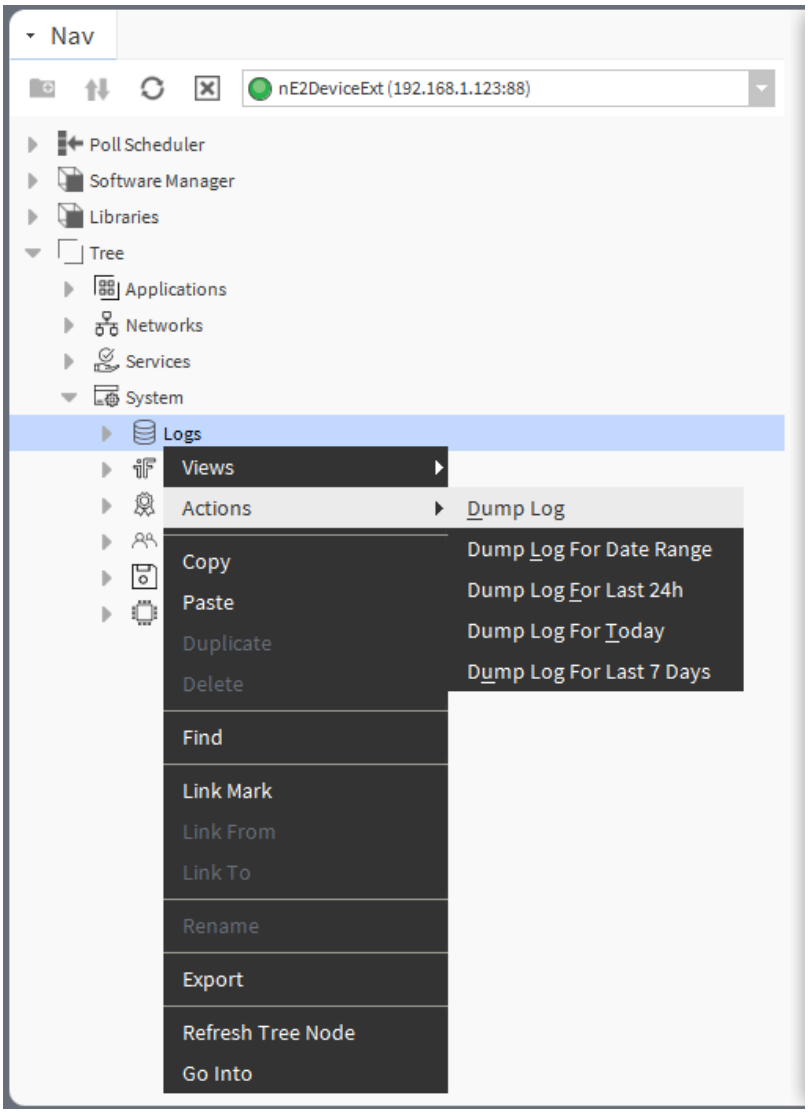


Figure 155. logLevel slots

9.2 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.



The following actions are available:

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

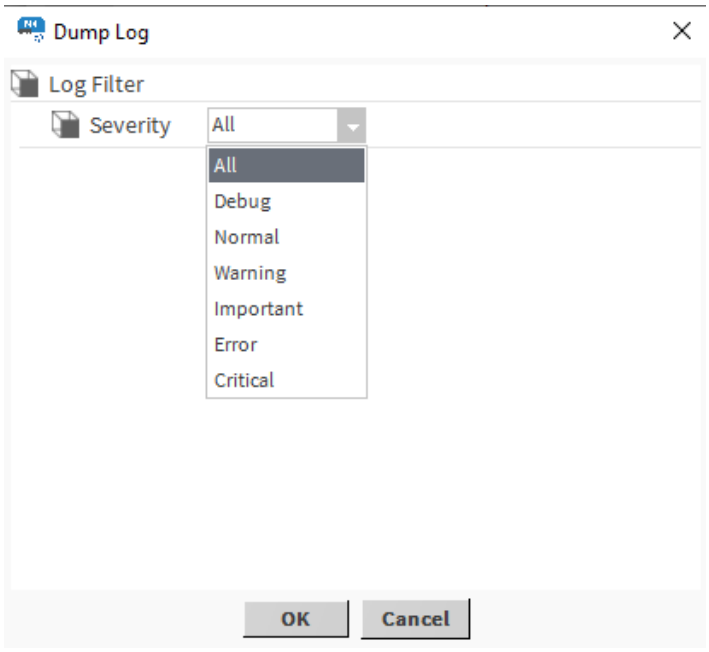


Figure 156. Dump Log action

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

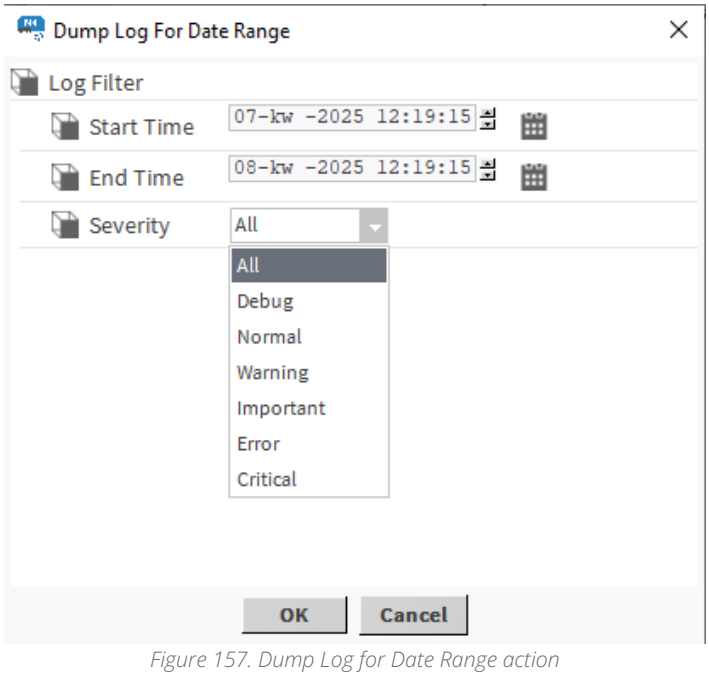


Figure 157. Dump Log for Date Range action

- 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.

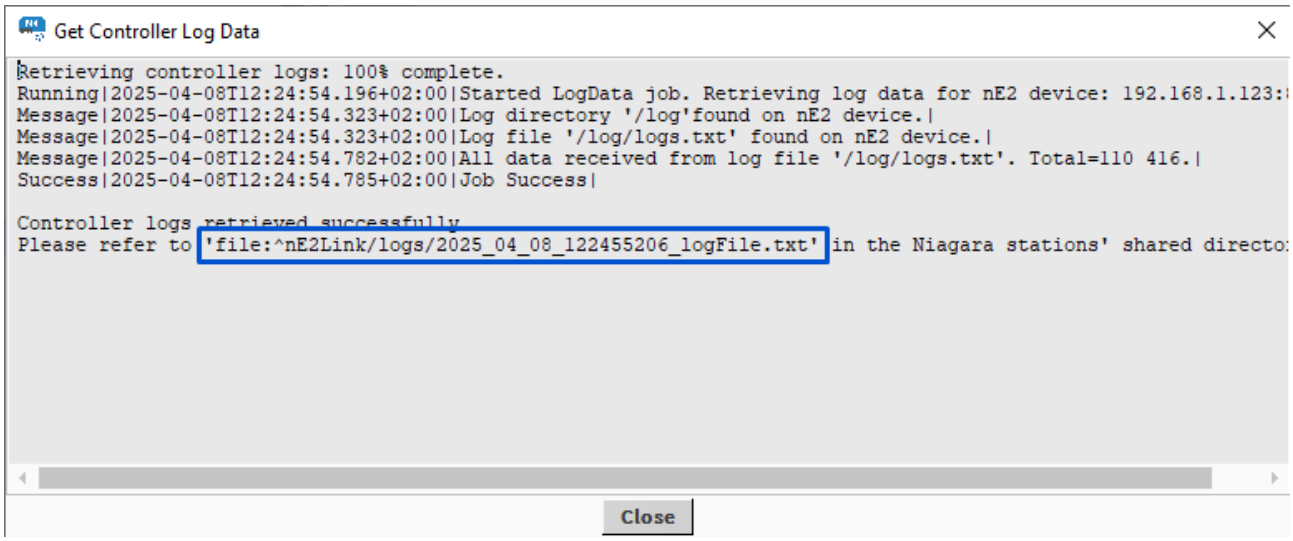


Figure 158. Saving logs confirmation pop-up

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

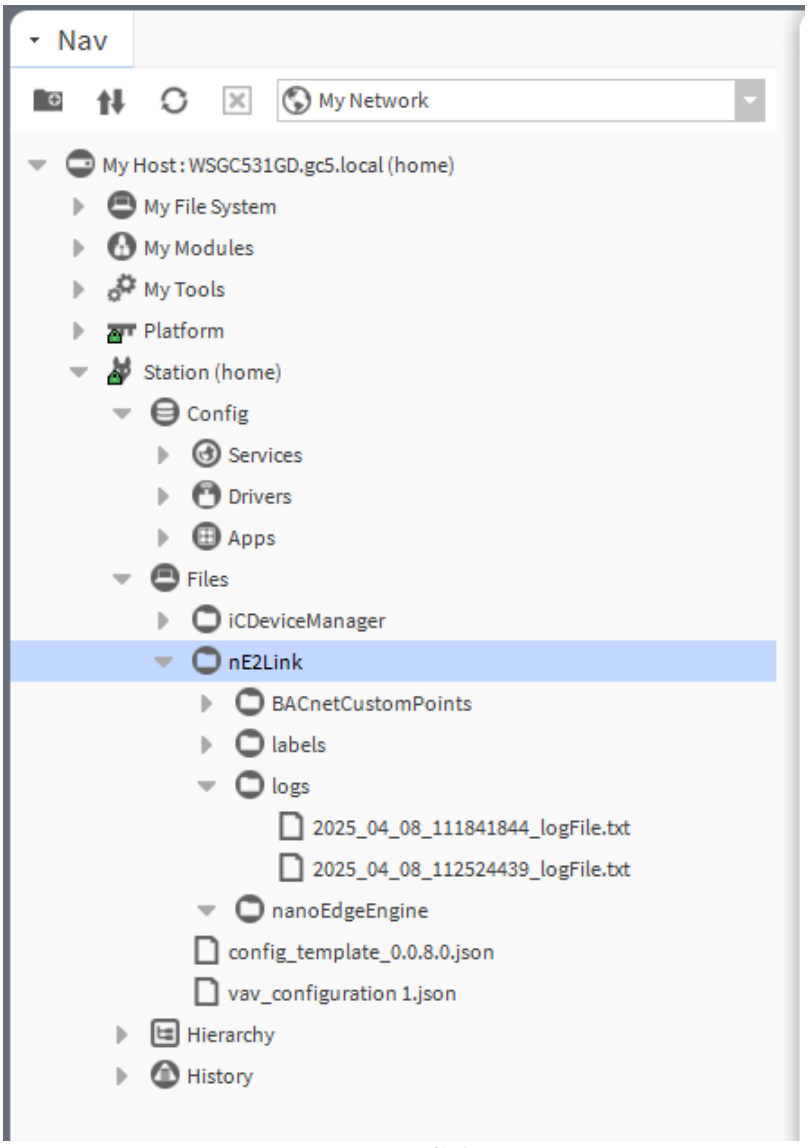


Figure 159. Logs file location

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.

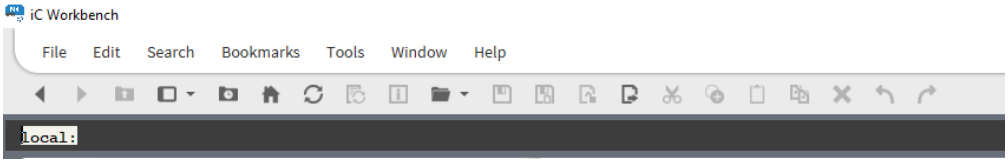


Figure 160. The Workbench bar with files' path becomes editable upon clicking

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

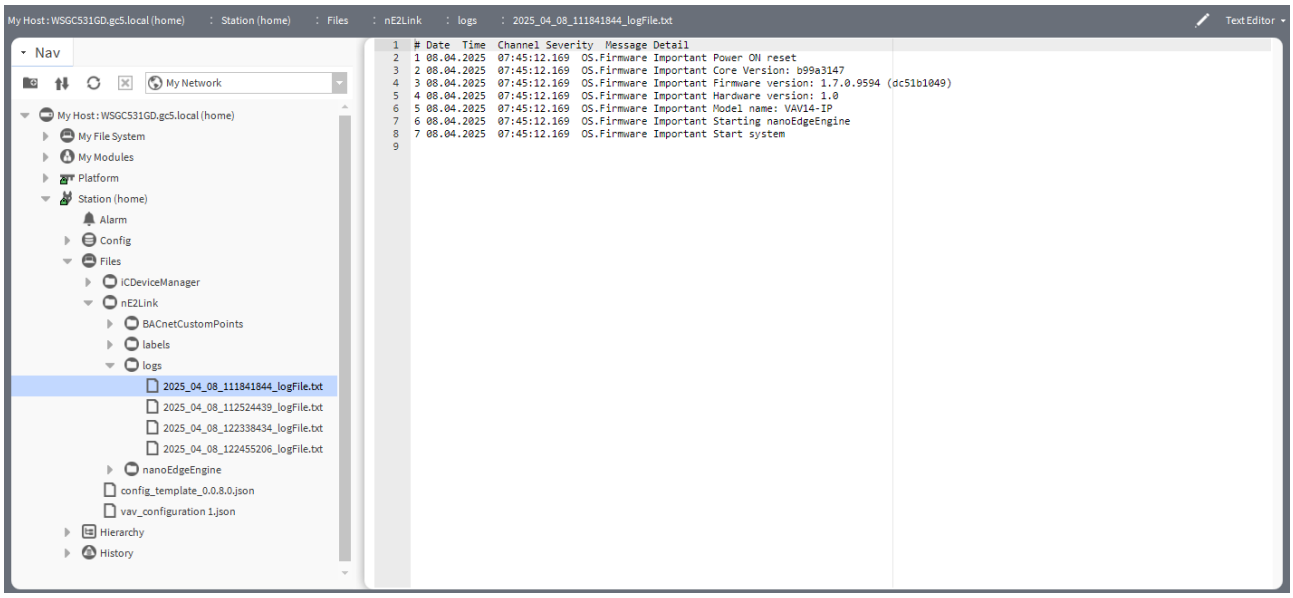


Figure 161. Logs presented in text editor