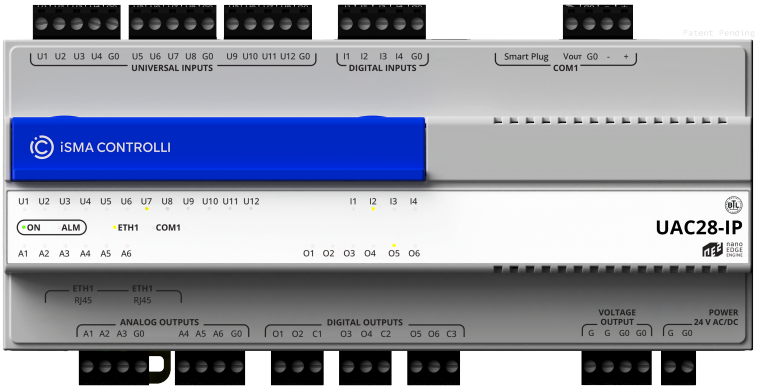
UAC28-IP

User Manual

Software



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

Table of Contents

1 Introduction

UAC28-IP is a freely programmable controller with the nano EDGE ENGINE embedded. It is Niagara-enabled and IP-based controller with built-in 28 I/Os, fail-safe Ethernet switch, and RS485 port with spring terminal and the Smart Plug™ for easy connection of iC wall panels for occupancy, fan, comfort, light and blinds control.
Ideal solution for HVAC and lighting control in two different zones, with a native support for IP communication with BACnet and Modbus protocols.

1.1 Revision History

Date	Manual rev.	nano EDGE ENGINE OS ver.	nE2 Link module ver.	Description
16 Jun 2026	1.0.0	1.10.0	1.10	First edition

Programming Tools

iC Tool

The iC Tool is a free programming tool dedicated to all devices based on the nano EDGE ENGINE software engine. The iC Tool is a portable, Windows-based software for commissioning, real-time programming, updating, and managing the supported nano EDGE ENGINE devices. The iC Tool covers all requirements to create and manage applications: it has a wire sheet for convenient visual programming, property sheets for details; it offers libraries management, real-time monitoring of system states and slots values, logs, and deployment.

nE2 Link

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.

2 Quick Start-up

Start working with the UAC28-IP controller in just 5 easy steps:

Step 1: Connection

Unbox the device. Connect power supply. The controller requires 24 V AC/DC power supply.

Connect the device to the network. The controller is equipped with the Ethernet IP fail-safe protected port, RS485 to connect with other devices in the network, and RJ45 to connect with external panels.

Step 2: Software tool

Make sure to download the software to program the device. The native tool for the controller's programming is the iC Tool, however, the it is also Niagara-enabled through the nE2 Link module. Go to iC Connect to download the best suited software and to Online Docs platform to explore how to use the iC Tool or nE2 Link module.

iC Tool: Open the iC Tool and follow one of 2 options:

(a) add the device to the project: open the context menu of the default project in the Workspace Tree (Site; if required, rename the project), and select the Add Device option. In the dialog window that pops up, set the device's port number (the default port number is 88). The default IP address of the device is 192.168.1.123, and it can be changed in the Ethernet component in the System container;

(b) use the USB connection: connect the device to the PC with the USB cable and navigate to the USB Device option in the Workspace Tree; open the context menu there and select connect;

nE2 Link: Follow the instructions available in the nE2 Link user manual.

Step 3: IP addressing

iC Tool: The IP Manager view in the iC Tool is a convenient tool for discovering devices connected to the IP network and configuring their network settings. It facilitates a first connection to a device, providing a simple way to set IP addresses on multiple discovered devices at a time. For details, please see: IP Manager.

nE2 Link: First, connect to the device (step 4). Then, to edit default IP address and network configuration, follow the instruction available in the IP Network Configuration.

Step 4: Logging in

Connect with the device. Once the device is added, the Authentication dialog window pops up. The default credentials are:

- username: admin
- password: admin

> **Note**
> When logging in to a factory-set or reset device, it is required to change password. Please refer to First logging-in.

Go to the System container in the tree and configure essential properties in the Platform component. The Platform component provides essential data about the device, and it also includes the Ethernet component, where the IP address, mask, and default gateway can be changed. When setting the network parameters of the device, make sure that they are consistent with the computer network parameters that the device is connected to (more on this: Address Pool Consistency).

Step 5: Application

The device is ready to work with the cycle-driven, multithreaded application.

3 Data Points

Data Points are universal components that represent a value in the application logic. They may serve as setpoints, sensors values, non-volatile variables, or any other data values. Data Points represent a layer of the application logic that is presented to an end user—this is where the end user is able to adjust desired setpoints (e.g., for air conditioning) or invoke other actions outlined in the application logic. Data Points also read values calculated in applications and control local or remote outputs.

Data Points in the application logic may work as regular writable variables with priorities, or—with Reference linking—they may be connected with network points, such as local I/O components.

Data Points are universal components that facilitate building application logics. They are adaptable by extensions, which expand their functionality by enabling them to be exposed to communication protocols. Among the fundamental extensions for Data Points are Priorities (Analog, Binary, and Multistate), which add 16 priority input slots to Data Points. It allows to build logics based on using value significance priorities—a value given in the highest priority slot dismisses all values of lower priorities, which enables, for example, making sure that no emergency is omitted due to values of lower significance. Slots, which are not linked and have a value entered, save the value on a given priority.

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.

In order to operate properly, Data Points must be placed in one of Application components in the Applications container. Only from this position, they may work properly included in the application logic.

Data Points Limitation

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 Data Points**, 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.
The Data Points limit assigned to the controller cannot be changed. The numbers of used and available Data Points are always indicated in the License component in the System container.

4 Linking

The Reference linking is a unique characteristic for the **nano EDGE ENGINE**, and it offers an exceptionally user friendly experience while creating applications.

Linking within applications, built with the **nano EDGE ENGINE**, may be performed twofold:

- using a special Reference link designed specifically to connect Data Point class components (in the Applications container) with network point class components (in the Networks container);
- using a standard linking method, which involves simple creating links between the input and output slots; a standard link transfers a value between the connected slots. Standard linking may be applied between all four containers of the **nano EDGE ENGINE** device structure.

4.1 Reference Linking

The nano EDGE ENGINE offers an innovative linking method called the Reference link.

The Reference linking method is unique for the nano EDGE ENGINE devices. Is it designed to link Data Points with network points, and it offers much more advantage than the standard linking method.

The **Reference link** is a special compound link designed to connect **Data Points with network points**. The Reference link is created between special Reference slots and **transfers values along with the component's status**. Alternatively, it may transfer values between Data Points and network points at the same time returning status from network points to Data Points, or it may return values from network points to Data Points. 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, and they are created between the tabs (or, for example, between the Application tab and the network points expanded in the Workspace Tree window) or within the Workspace Tree window between Data Points in the Applications container and network points in the Networks container. Either way the Reference link between tabs is displayed in the Wire Sheet as a bubble connected to the component's Reference slot.

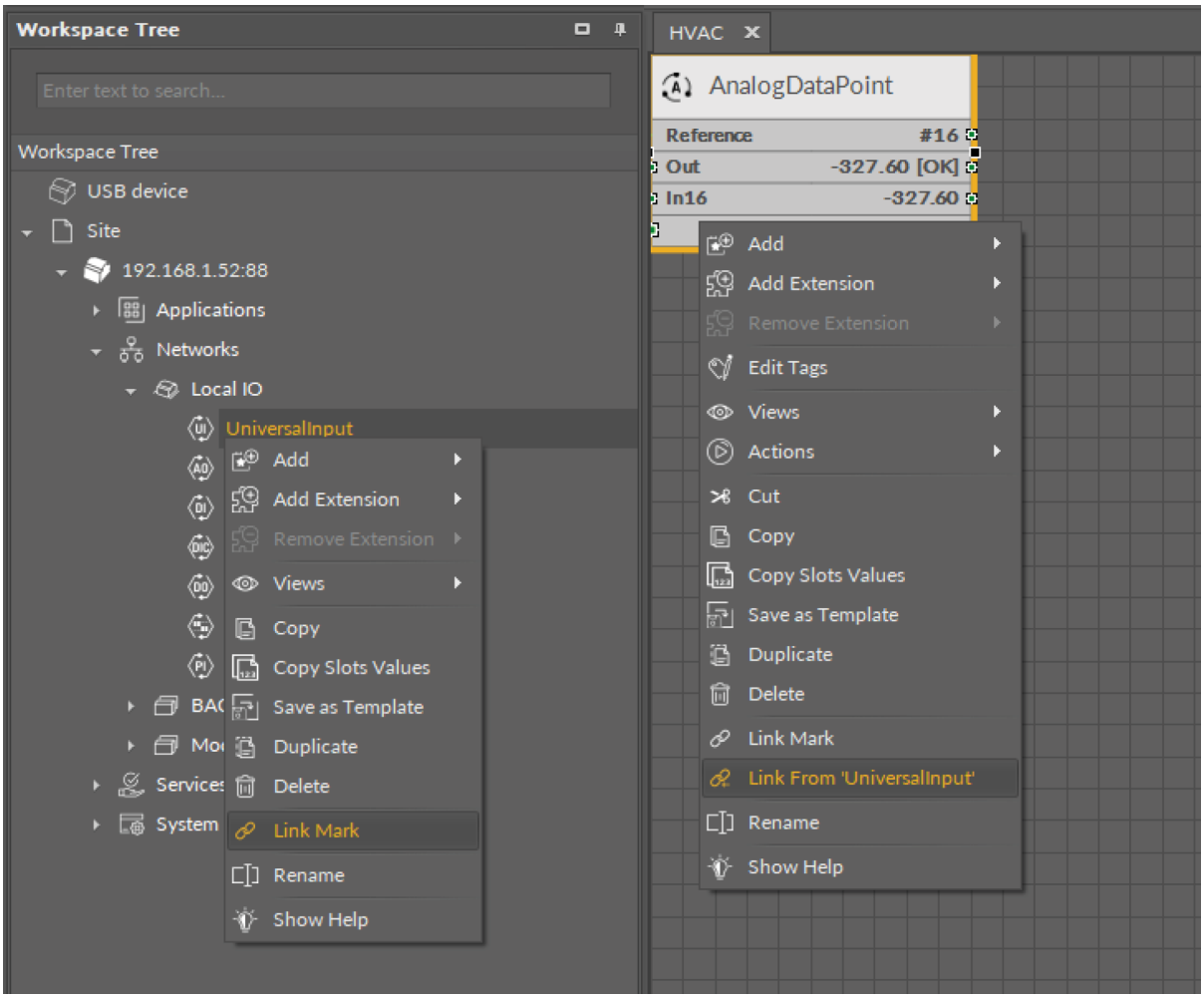


Figure 1. Link Mark-Link From options

The basic and exceptional feature of the Reference links is the fact that they are made to **transfer the value along with the component's status**. This feature gives a major advantage and translates to substantially enhanced functionality of linking. The fact of transferring the Status along with component's value is exceptionally important for the functionality of Data Points. Data Points are central elements in the nano EDGE ENGINE applications, and they represent values in applications on the Wire Sheet. Therefore, displaying network point's status makes the Data Point much more informative, and allows to display this important information directly in the Wire Sheet.

4.1.1 Data Point and the Input-type Network Point Links

If the Data Point is linked with the input-type network point, the Reference link transfers the network point's value and status to the Data Point (if the Data Point has a priorities array extension added, there is also the option to set the Input Priority slot in the network point, which defines the input priority in the Data Point receiving the value from the network point). In this variant, the Reference link is unidirectional, and provides the information about the change of value and the network point's status.

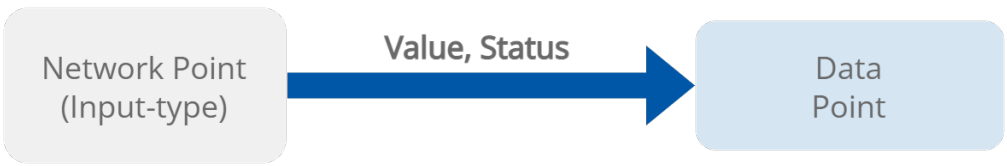


Figure 2. Data Point and the input-type network point links

If the Data Point is linked with the output-type network point (or network points), it offers even more advantages.

4.1.2 Data Point and the Output-type Network Point Links

- First of all, in such case, the Reference link behaves **bidirectionally**. It transfers the value from the Data Point to the network point, and **in turn it informs whether the value has been correctly received by the network point by sending back the network point's status**. This hugely advantageous feature allows to instantly identify that at least one of the linked network points has gone into the fault status.

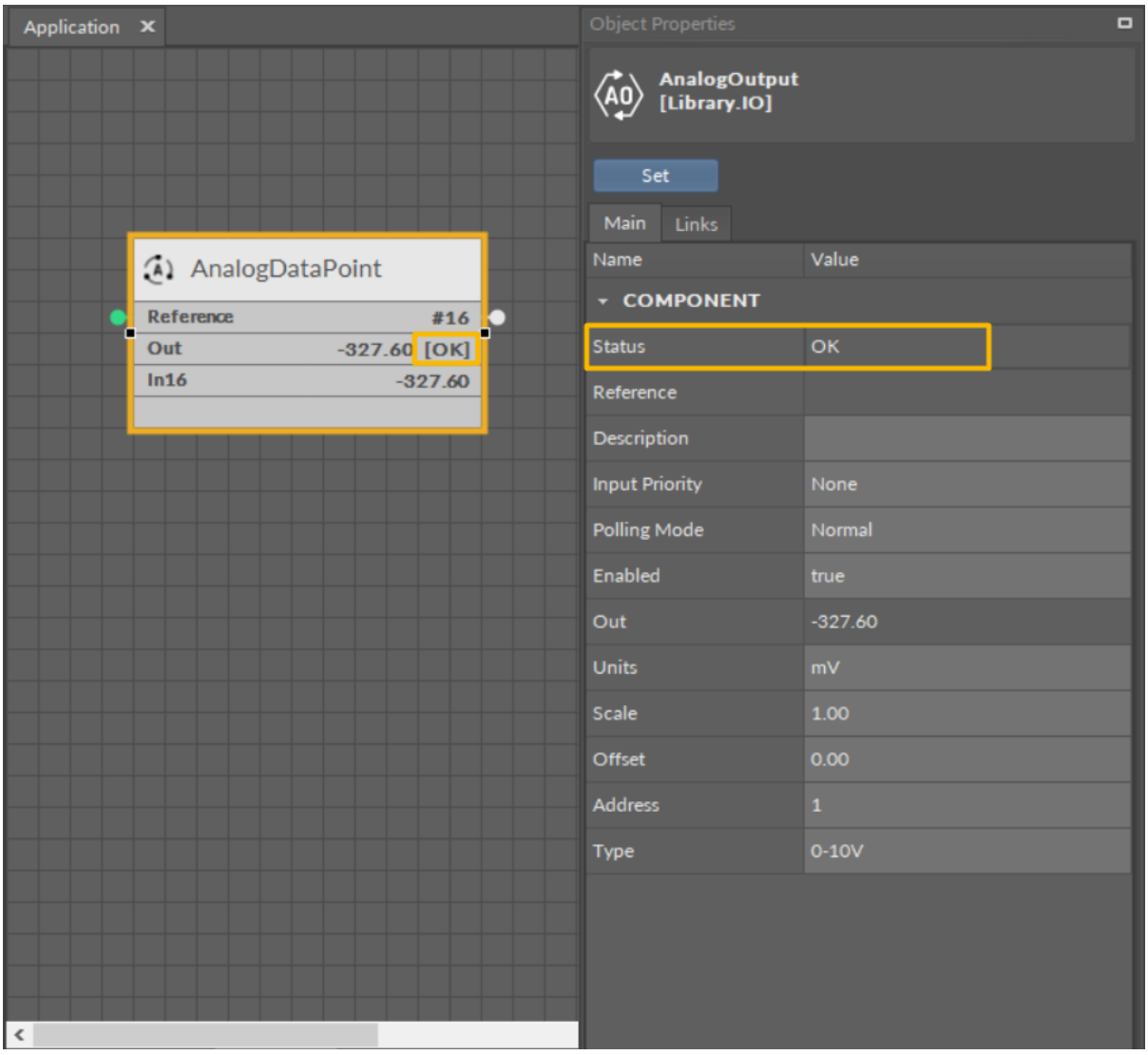


Figure 3. Output-type network point (status OK) referenced to the Data Point

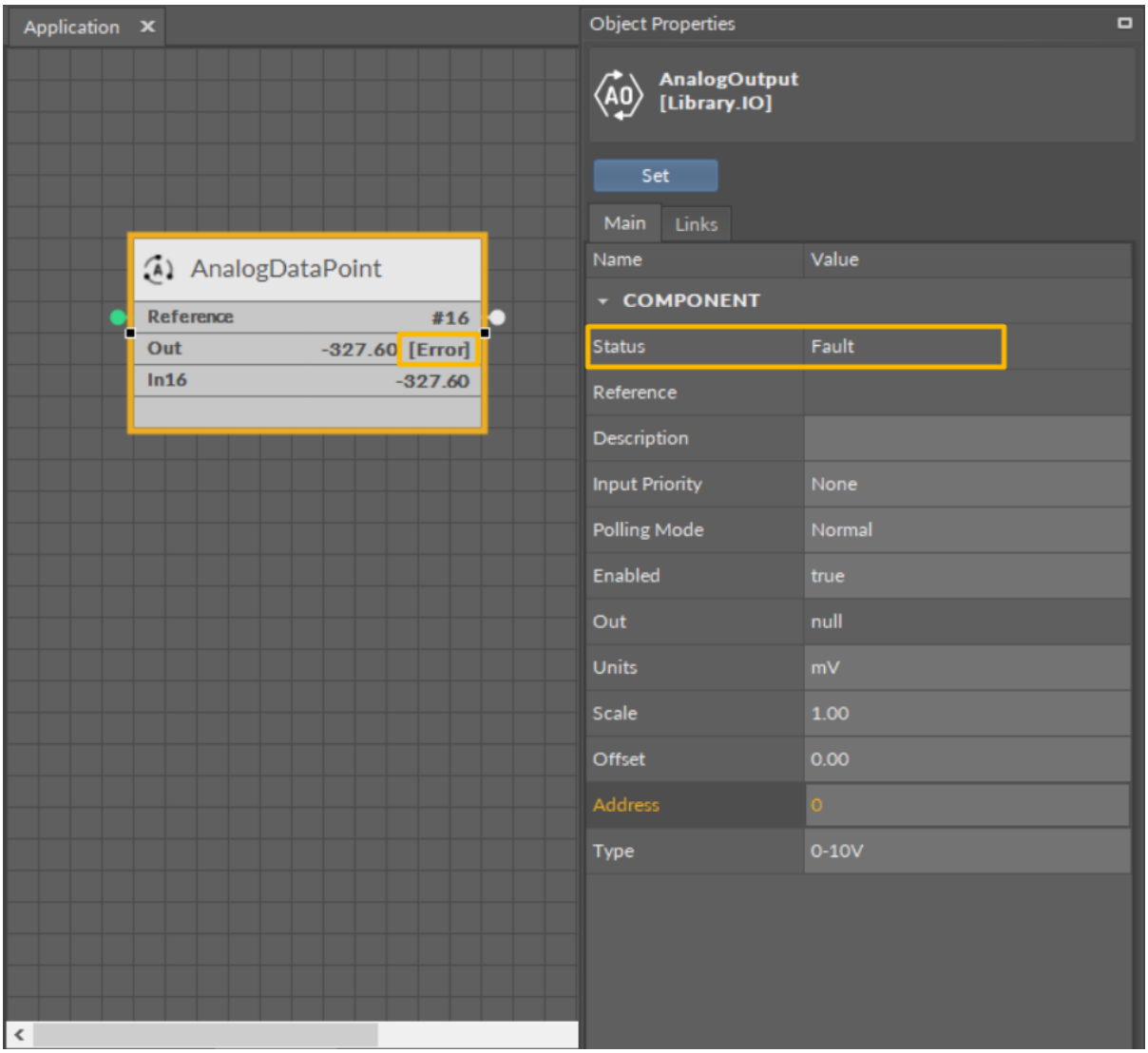


Figure 4. Output-type network point in the Fault status returning an Error status to the referenced Data Point

- Another innovative feature of the Reference linking is that the Reference link is able not only to return status from the network point but also to **return its value**. The network point's slot, Input Priority, is designed to identify the Data Point's priority, which the network point will transfer its value back to. For example, if the network point's Input Priority slot is set to In16, it will transfer its value back to the Data Point's 16th priority slot. In turn, if this value is the highest priority for the Data Point, it can distribute it to all network points linked with the Reference link. **This way, if there are more output-type network points linked with the Data Point, and one of them changes its value and sends it back to the Data Point, thanks to the bidirectional Reference link, the Data Point can synchronize values in all linked network points.**

Note: The difference between the two possible options for linking Data Points with output-type network points is derived from the network point's **Input Priority slot**. In the first scenario, the network point does not have the Input Priority set to any value, therefore, it cannot return a value back to the Data Point. In the second option, the network point has the Input Priority slot set and it reacts to the change of value– if the network point's value is changed, it is automatically sent back to the Data Point by the Reference link, and is updated on the defined input priority.

Priorities Array Extension

Setting the network point's Input Priority slot is effective providing that the Data Point has the priorities array extension added. The Data Point is available in its basic version with one input slot (In16), however, it can be expanded by another 15 writable input slots with the priorities array extension (available at the right-click on the Data Point). If the Data Point has 16 writable slots, setting the Input

Priority slot in the network point defines the Data Point's input receiving the value from the output-type network point over the Reference link. If the Data Point remains in its basic version, setting the Input Priority slot in the network point has no actual effect, and the value is sent to the 16th input priority in the Data Point.

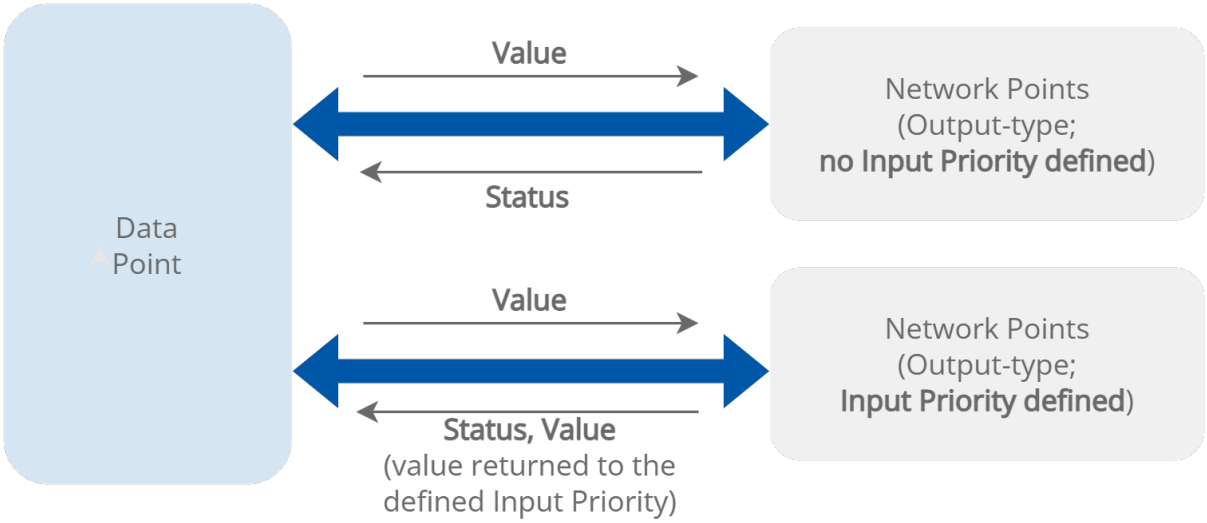


Figure 5. The Reference links between the Data Point and output-type network points

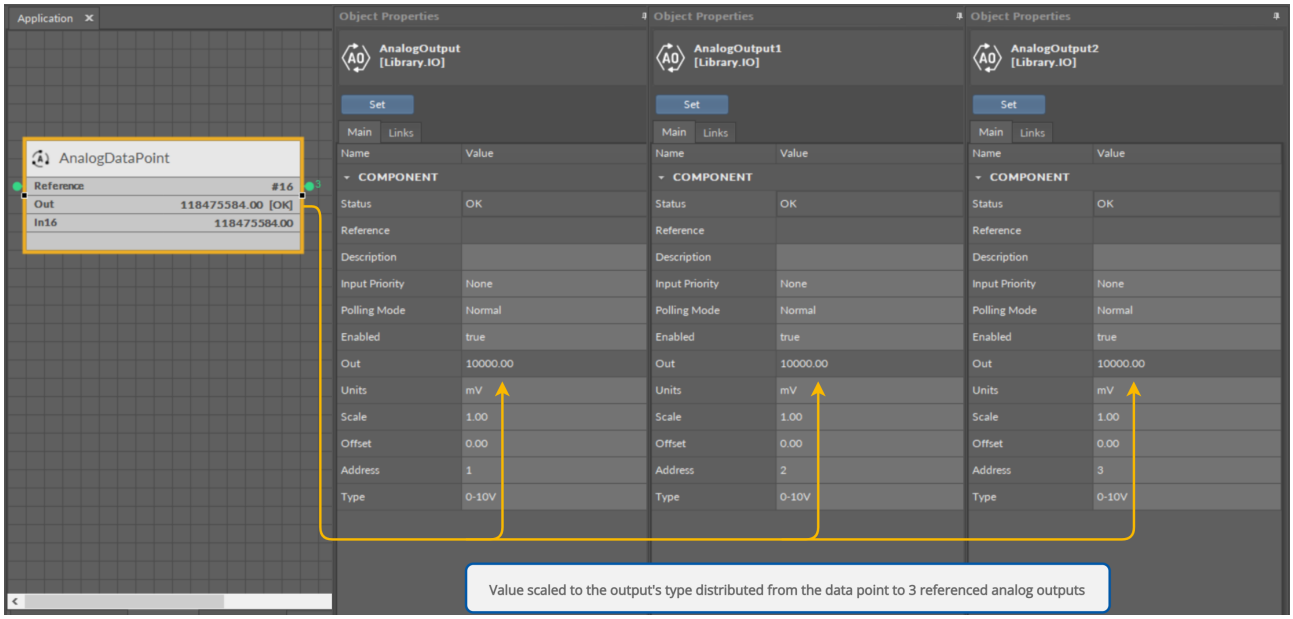


Figure 6. Values distributed from the Data Point to the referenced output-type network points

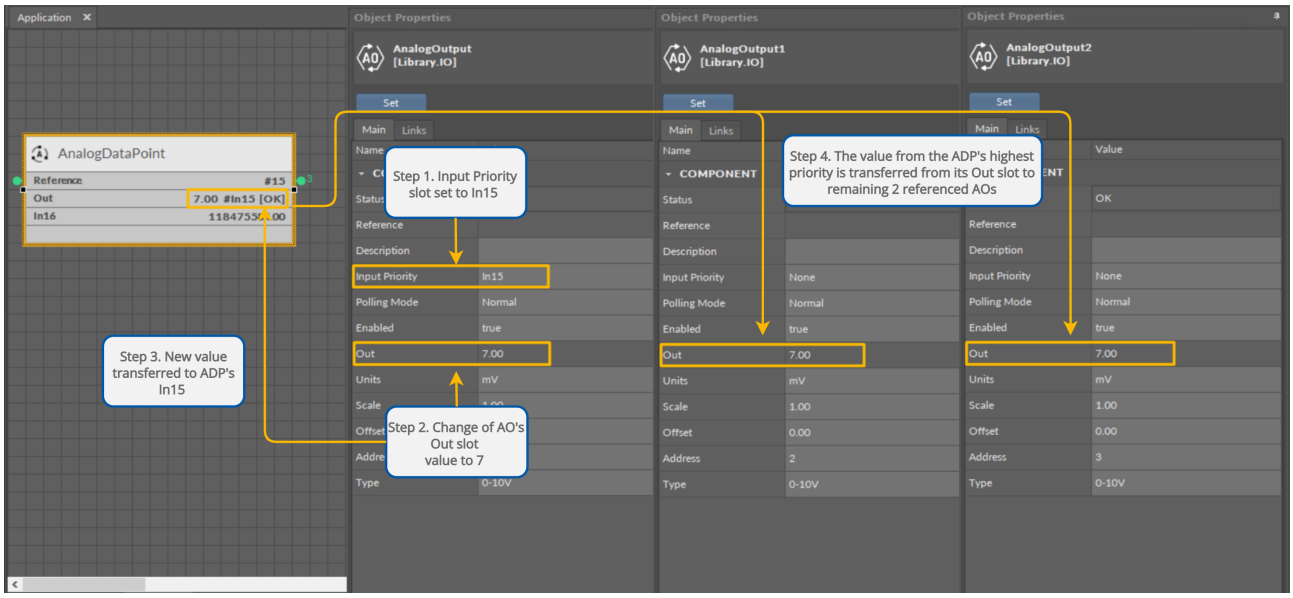


Figure 7. Value sent back to the specified input priority in the Data Point and distributed to remaining referenced output-type network points

The Reference linking is recommended for linking components in the Applications with components in the Networks container.

- Using the Link Mark and Link From options from the context menu of network points and Data Points that are to be linked, opens the dialog window, which allows to select Reference slots on both sides of the link:

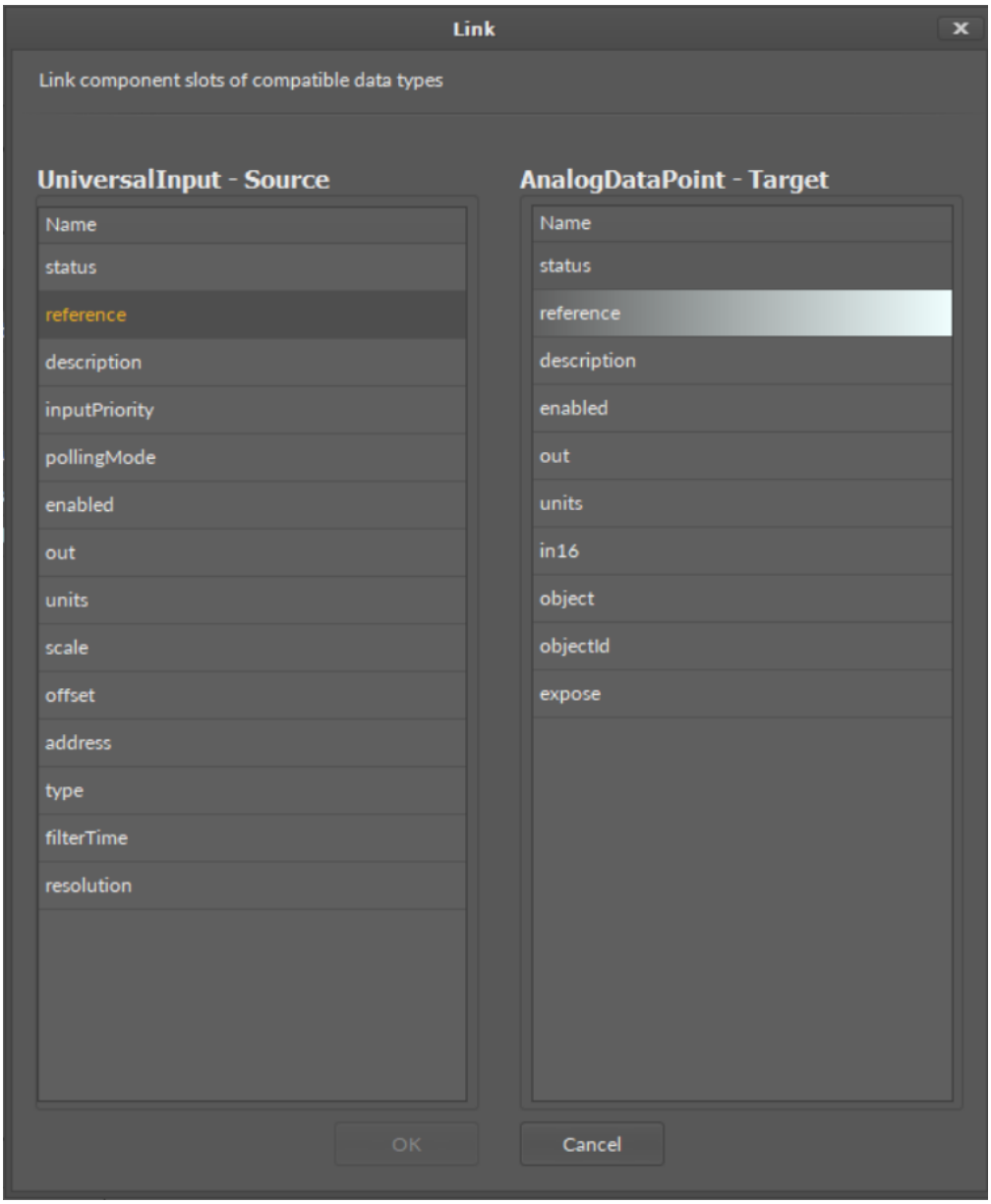


Figure 8. The linking dialog window

- As the Reference links connect elements from the Applications container (Data Points) and the Networks container (network points), in the Wire sheet they are displayed as bubbles: on the left side of the Reference slot for the input-type linking, and on the right side of the Reference slot for the output-type linking:

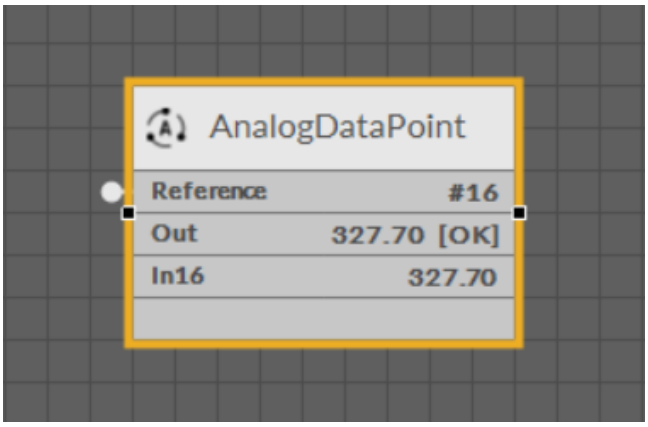


Figure 9. Analog Data Point linked from the Universal Input

- The details of the Reference links are always visible in the Links tab in the Object Properties window of the linked element:

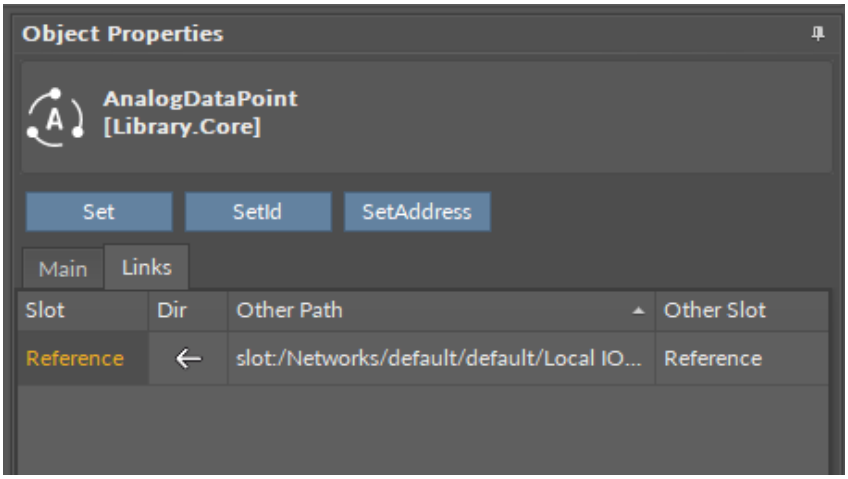


Figure 10. Link's path

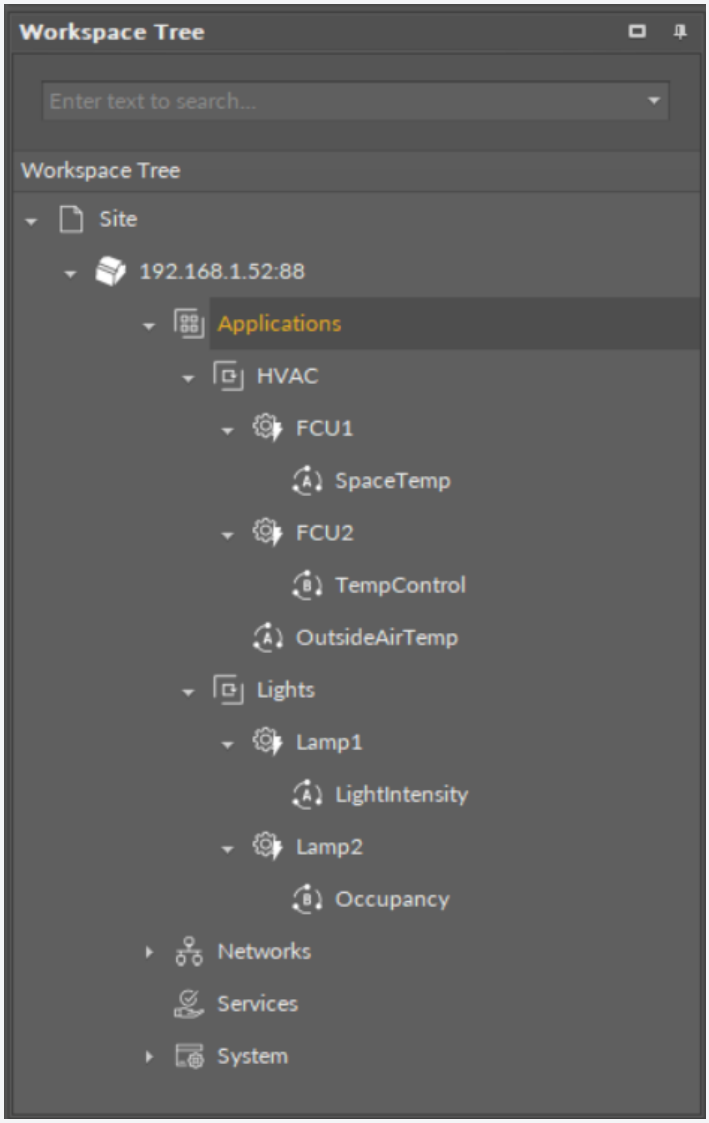
5 Applications

The first element in the Device structure is the Applications container. It is a non-removable element, which offers a space to create user applications.

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).

Tip
Multithreaded applications allow to differentiate cycles of applications (scan period) in order to adjust them according to a purpose of an application.
For example, a HVAC application may be set to 1000 ms scan period, and a Lights application to 200 ms scan period, as changes to lights operation have to be implemented immediately. Setting such scan periods would mean that the Lights application would be executed 5 times for each HVAC application cycle.



The user may create as many applications as needed—as long as the overall number of Data Points used within user applications does not exceed the device license.

Note: For top performance, it is recommended to use up to three different applications.

Application Structure for Tags

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

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

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

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

💡 Remember - Great Licensing Scheme

The Applications container includes the only elements subject to license in the nano EDGE ENGINE devices.

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.

5.1 Quick Start-up of Applications

The UAC28-IP controller offers a smart and easy way of creating applications.

The nano EDGE ENGINE application is cycle-driven, multithreaded, and can address multiple purposes. Structure of the nano EDGE ENGINE application is based on Application components, each of which defines a separate application thread and, possibly, a cycle of execution. In the device structure, the application refers to the following containers:

- Applications;
- Networks;
- Services.

The Applications container is where the main logic is contained; the Networks container is where the external communication is configured, and the Services container is the place to connect with additional functionalities.

In order to create a fully functioning application follow these steps:

Step 1: Connection and networks configuration

Make sure that the device is correctly connected and set up. Detailed instructions are available in the device's UAC28-IP Quick Start-up (Step 1-7).

Go to the Networks container to configure the external communication settings (BACnet, Modbus). Configure all the network-specific parameters there in order to enable proper communication as a client or server device in the Modbus TCP/IP or RTU, or BACnet IP or MS/TP protocol.

Remain in the Network container. Go to the LocalIO component, and configure all input and output components that are going to be utilized in the main application logic. The input and output components are grouped in the IO library, available in the Device Libraries window. Most importantly, set their addresses (the Address slot in each input or output component)—unless the addresses are set properly, the components are in fault statuses.

Step 2: Creating application

Go to the Applications container. This is the place to add Application components from the Core library—the nano EDGE ENGINE allows to add as many Application components to the container as necessary. Each application created this way is independent and cycle-driven. The Applications Manager is a view designed to manage Application components.

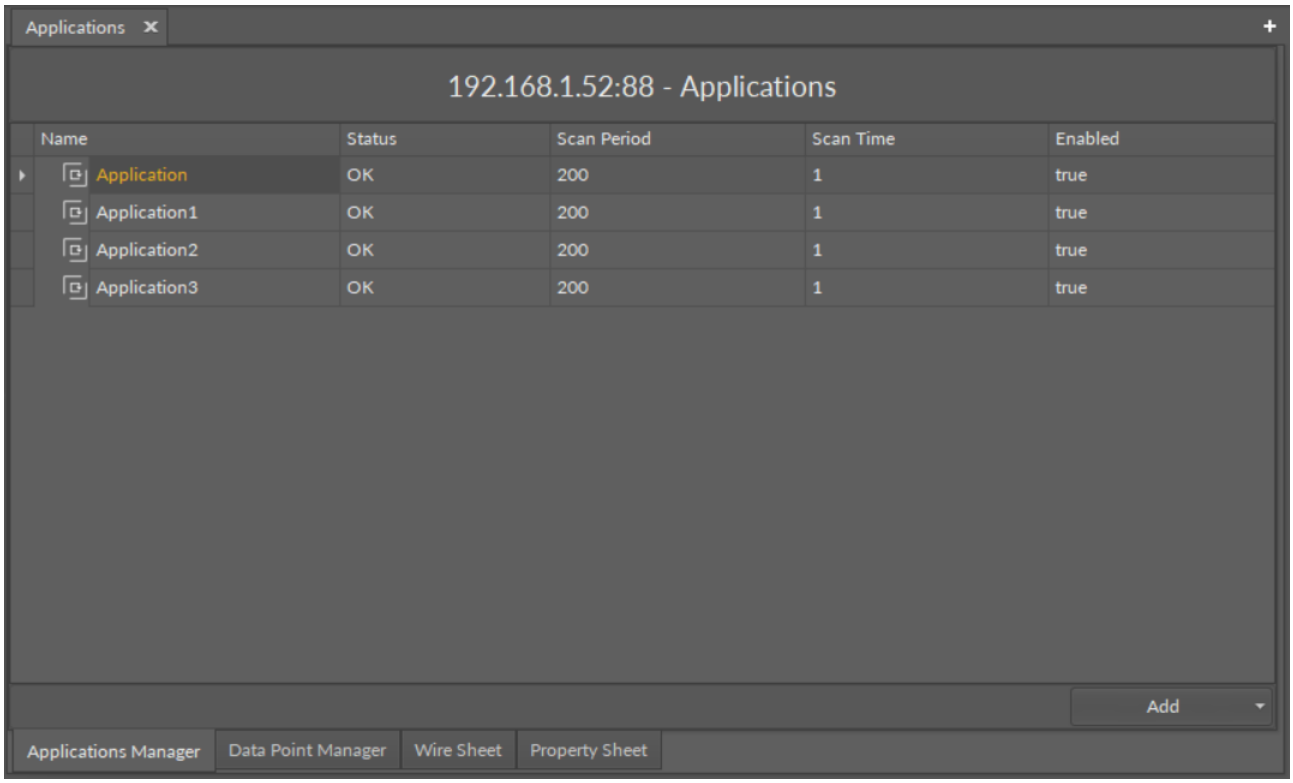


Figure 11. Applications Manager

(Recommended) Once all Application components are added to the Applications container, go to the Core library, and add Equipment components for each item controlled by the application. Then add Data Points to the Equipment components. It is advised to rename components to fit the application characteristics.

Application Structure for Tags

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

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

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

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

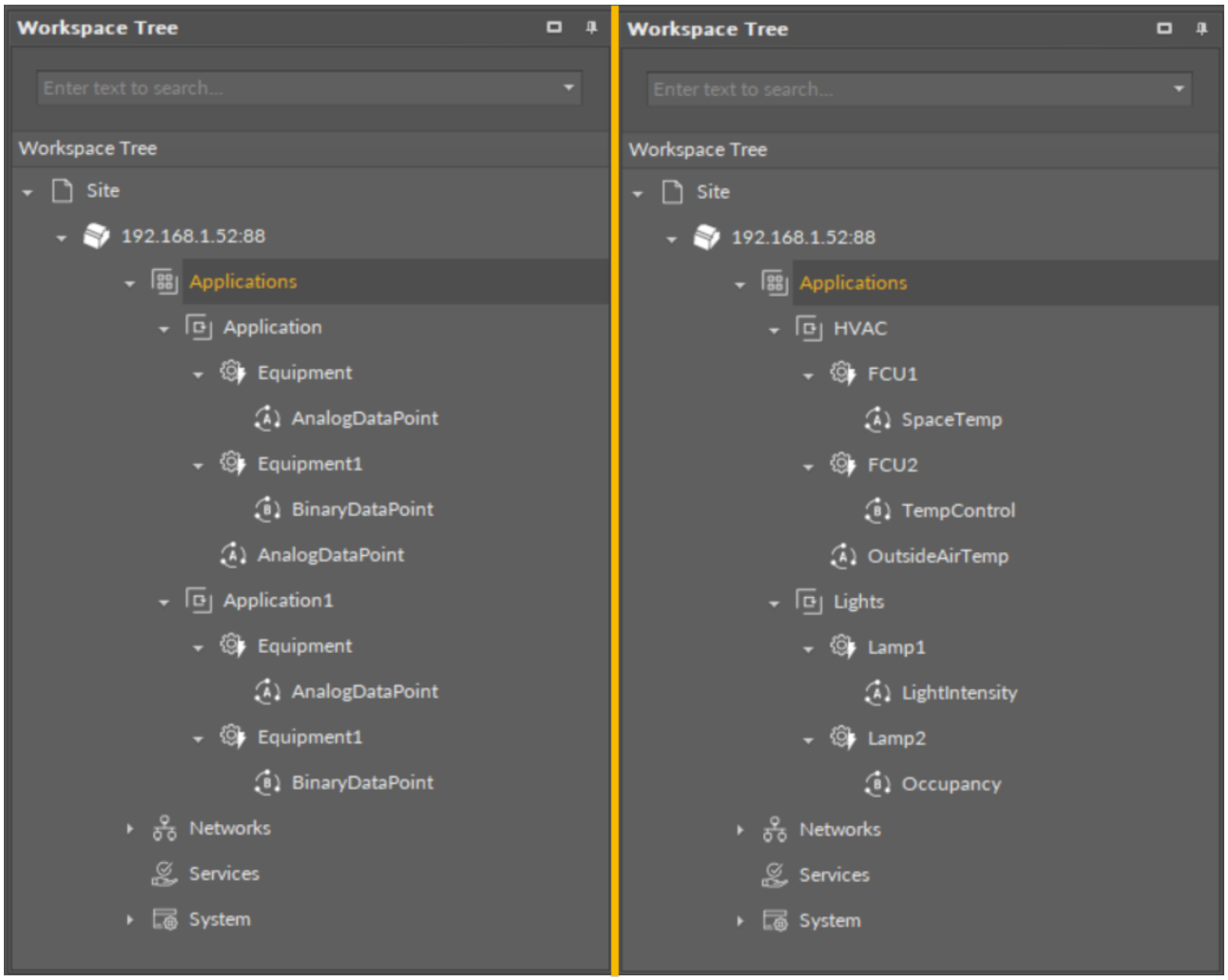


Figure 12. Recommended structure for applications

(Recommended) Start adding other components to the Equipment component (or components) to create logics with added Data Points. All components available for creating applications are grouped in libraries in the Device Libraries window.

Tip
Data Points are universal components that represent a value in the application logic; they may serve as setpoints, sensors values, non-volatile variables, or any other data values. Data Points represent a layer of the application logic that is presented to an end user—this is where the end user is able to adjust desired setpoints (e.g., for air conditioning) or invoke other actions included in the application logic. Data Points also read values calculated in applications and control local or remote outputs.
Data Points in the application logic may work as regular writable variables with priorities, or—with Reference linking—they may be connected with network points, such as local IO components.

- AnalogDataPoint,
- BinaryDataPoint,
- MultistateDataPoint.

Step 3: Linking

Added component is ready to be linked. The nano EDGE ENGINE allows two methods of linking, standard and Reference linking. The Reference linking method is especially recommended to use to connect Data Points and network point class components. Detailed information is available in the Linking section.

Tip

Here is the list of other quick start-ups that offer detailed instructions on how to configure external communication, network points, explain linking methods, etc.:

- Device quick start-up;
- BACnet quick start-up;
- Modbus quick start-up;
- LocalIO quick start-up;
- nano EDGE ENGINE linking methods.

5.1.1 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.

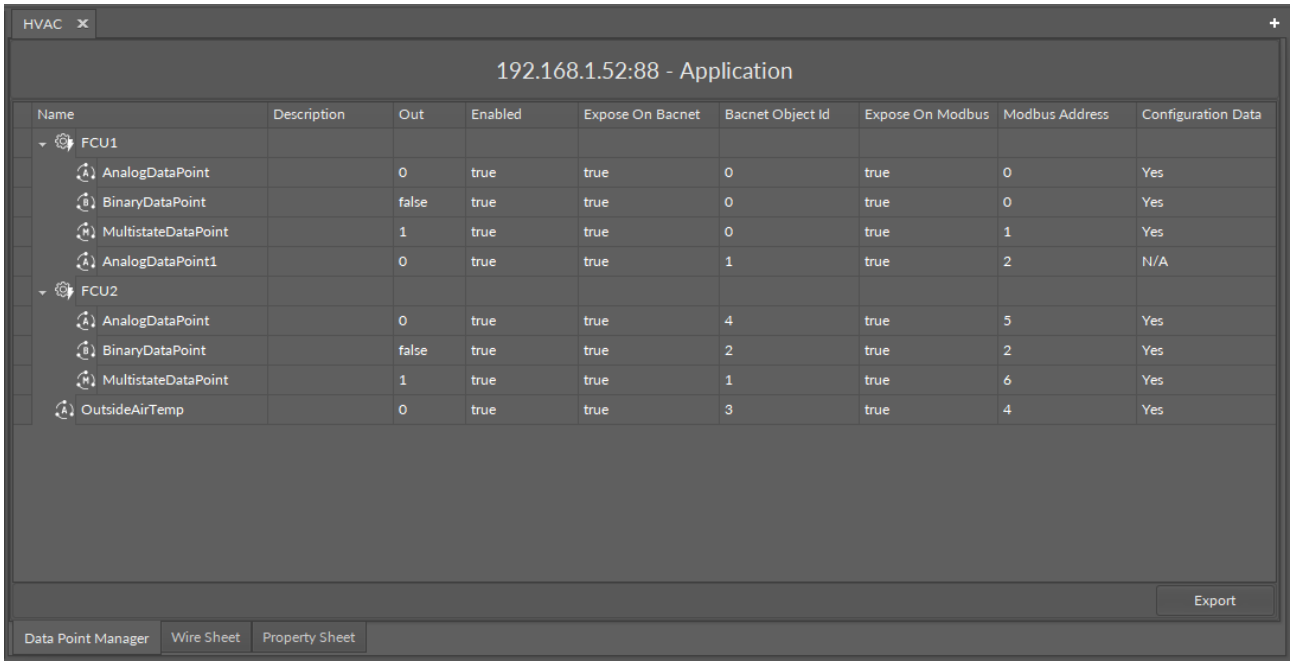


Figure 13. Data Point Manager

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

- Equipment, which the Data Point belongs to;
- 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.

The Data Point Manager view is not editable; however, once a specific Data Point is clicked in the Manager view, it is displayed in the Object Properties window, where it can be freely edited. Also, the view allows for multiediting of compatible Data Points–if compatible Data Points are selected in the Data Point Manager, their common slots are available for multiedition in the Object Properties window.

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 LocalDevice for BACnet component;
- in the context menu of the LocalDevice for Modbus component.

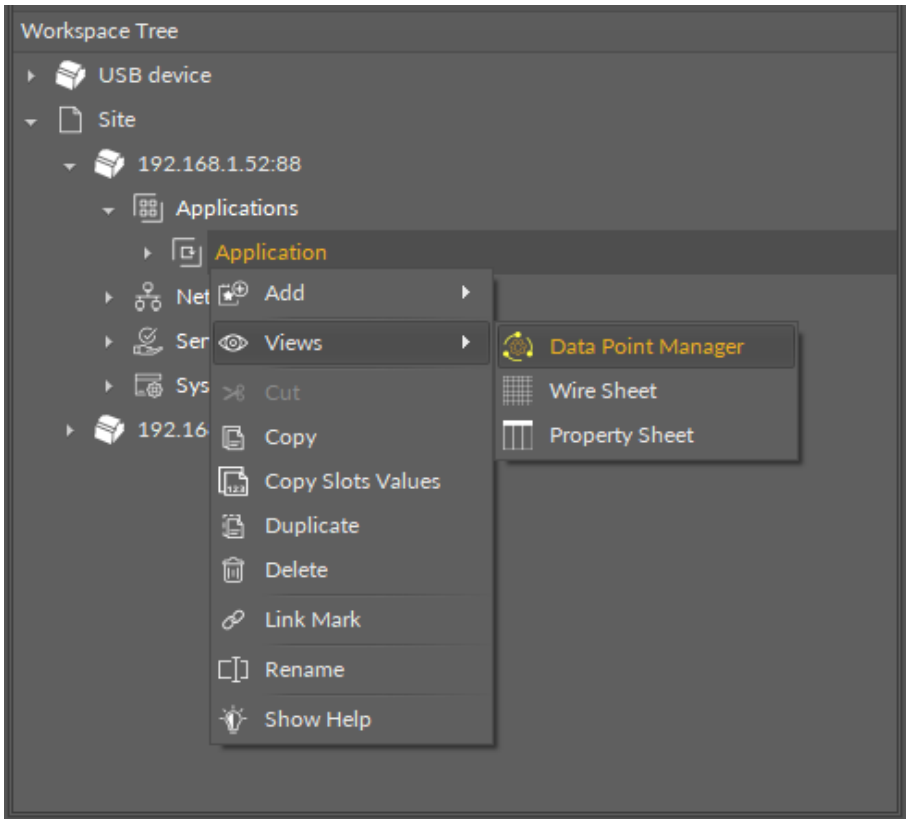


Figure 14. Accessing the Data Point Manager

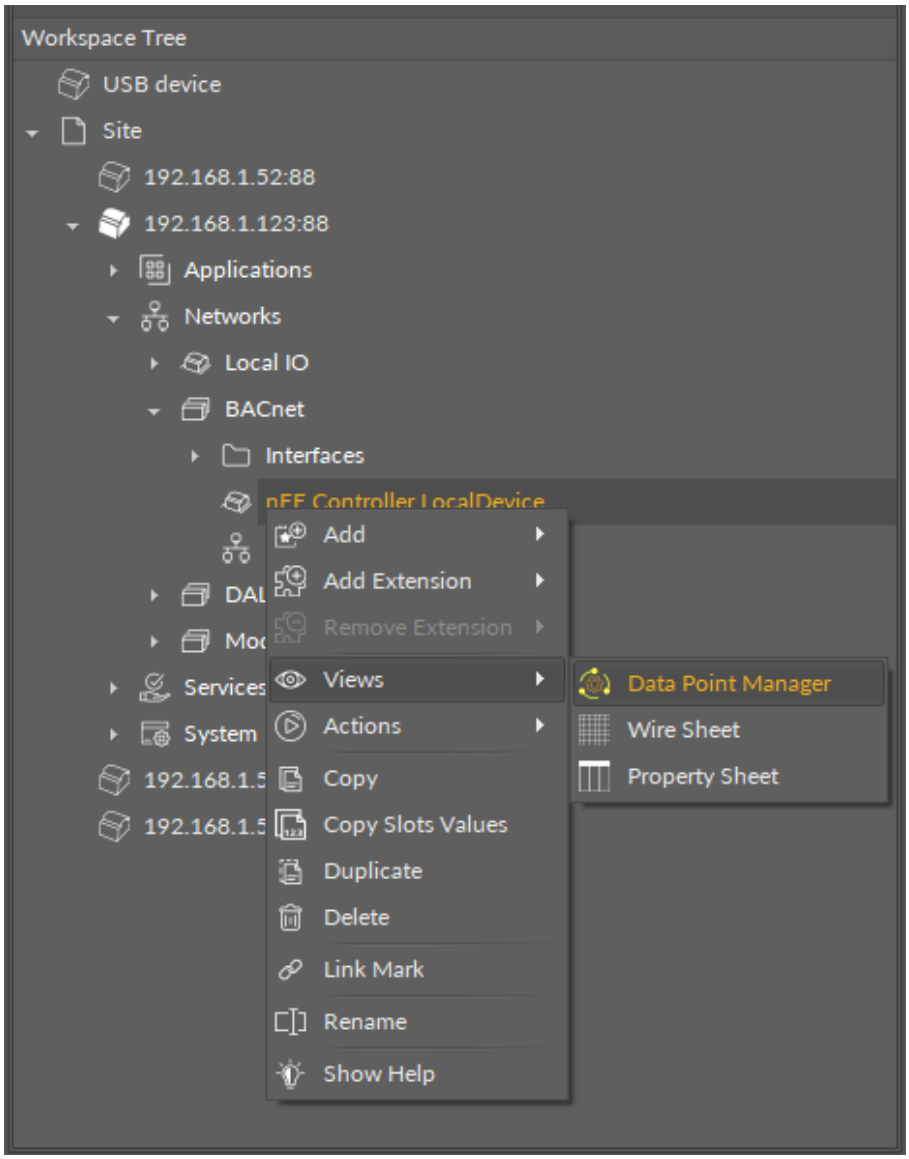
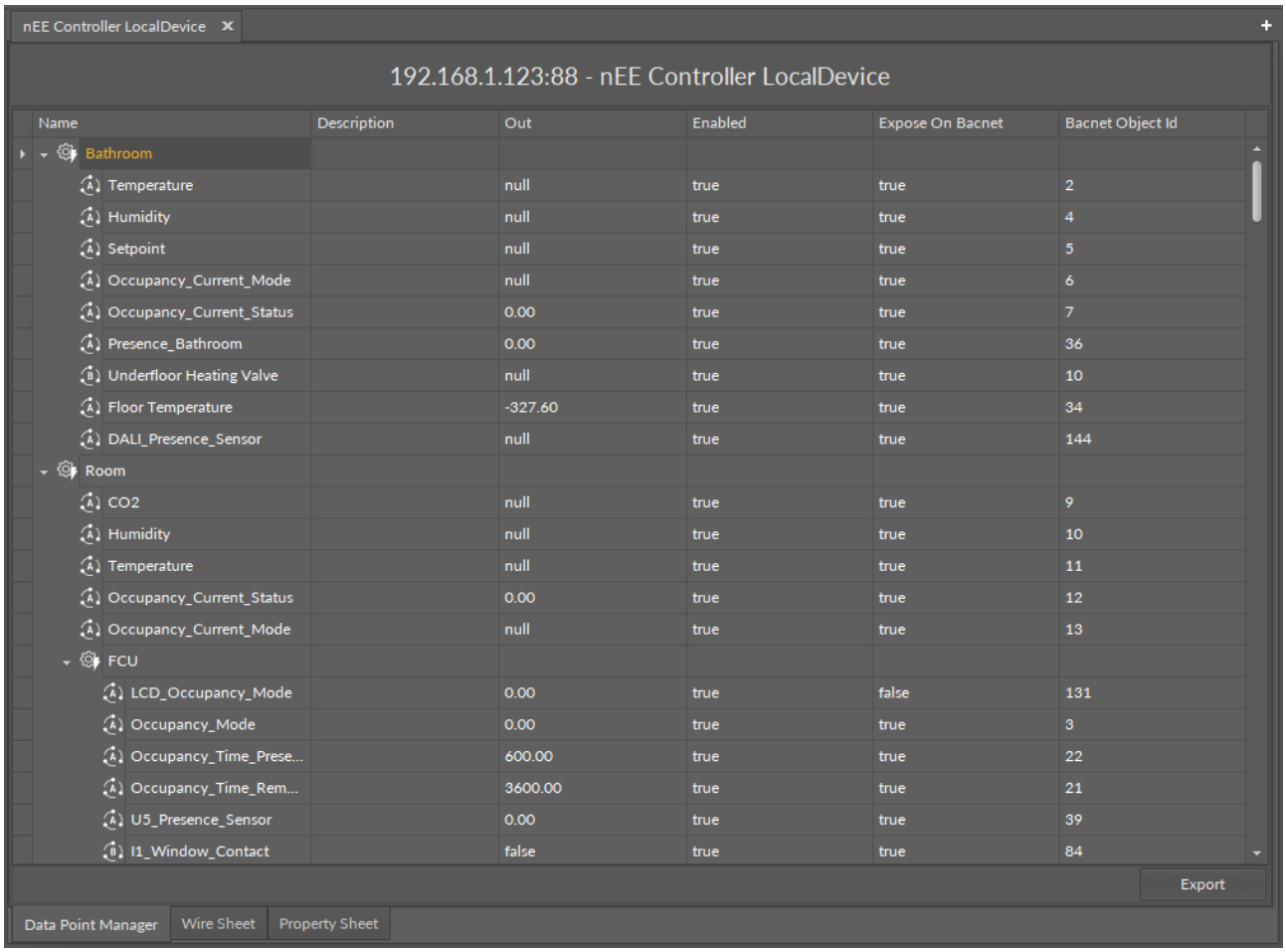


Figure 15. Accessing the Data Point Manager

The Data Point Manager view is also automatically opened if either the Applications container or the LocalDevice for BACnet or Modbus component is double-clicked in the Workspace Tree window.

The view adjusts its contents if opened for the BACnet or Modbus LocalDevice showing information specific to the network:



Name	Description	Out	Enabled	Expose On Bacnet	Bacnet Object Id
Bathroom					
Temperature		null	true	true	2
Humidity		null	true	true	4
Setpoint		null	true	true	5
Occupancy_Current_Mode		null	true	true	6
Occupancy_Current_Status		0.00	true	true	7
Presence_Bathroom		0.00	true	true	36
Underfloor Heating Valve		null	true	true	10
Floor Temperature		-327.60	true	true	34
DALI_Presence_Sensor		null	true	true	144
Room					
CO2		null	true	true	9
Humidity		null	true	true	10
Temperature		null	true	true	11
Occupancy_Current_Status		0.00	true	true	12
Occupancy_Current_Mode		null	true	true	13
FCU					
LCD_Occupancy_Mode		0.00	true	false	131
Occupancy_Mode		0.00	true	true	3
Occupancy_Time_Prese...		600.00	true	true	22
Occupancy_Time_Rem...		3600.00	true	true	21
U5_Presence_Sensor		0.00	true	true	39
I1_Window_Contact		false	true	true	84

Figure 16. The Data Point Manager for the BACnet LocalDevice

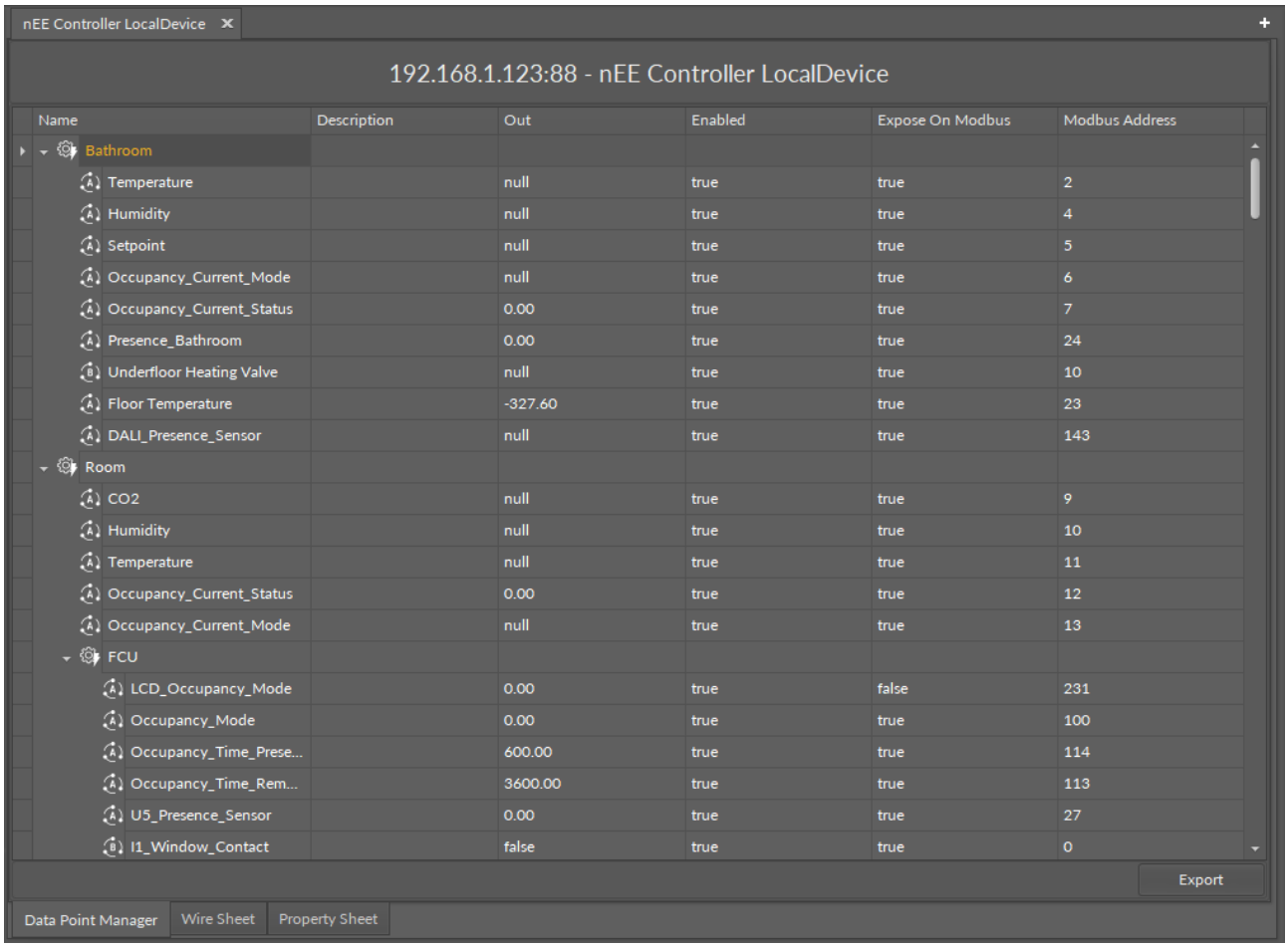


Figure 17. The Data Point Manager for the Modbus LocalDevice

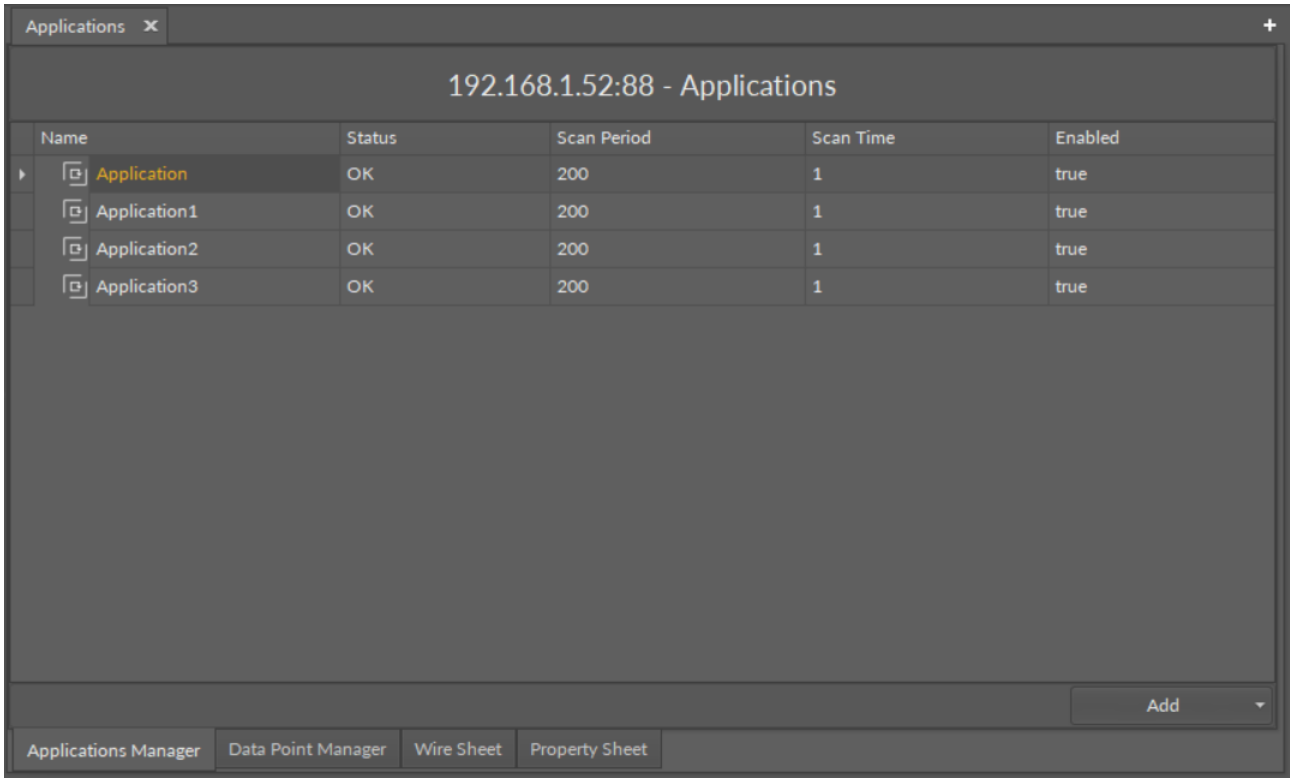
Licensing

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.

5.1.2 Applications Manager

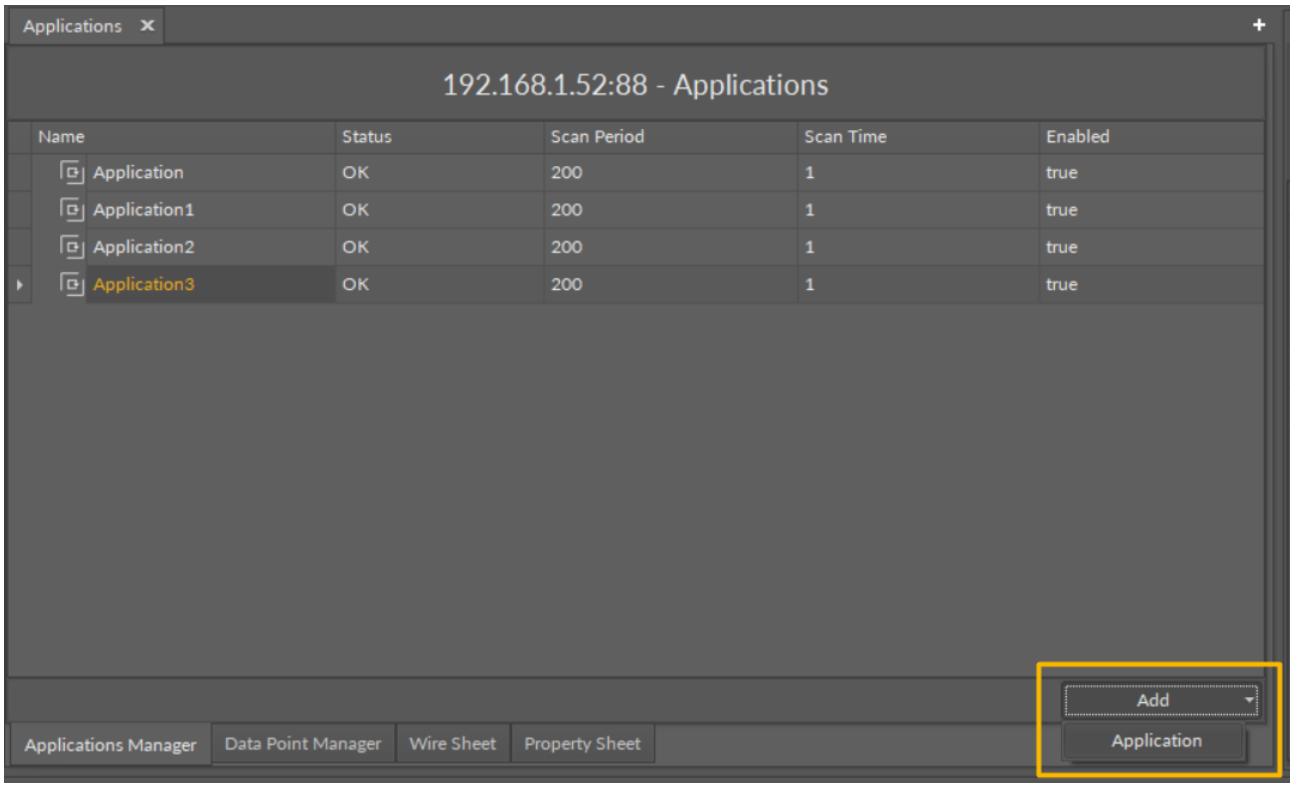
The Applications Manager is a special view that allows to manage the Application components added to the Applications container.



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

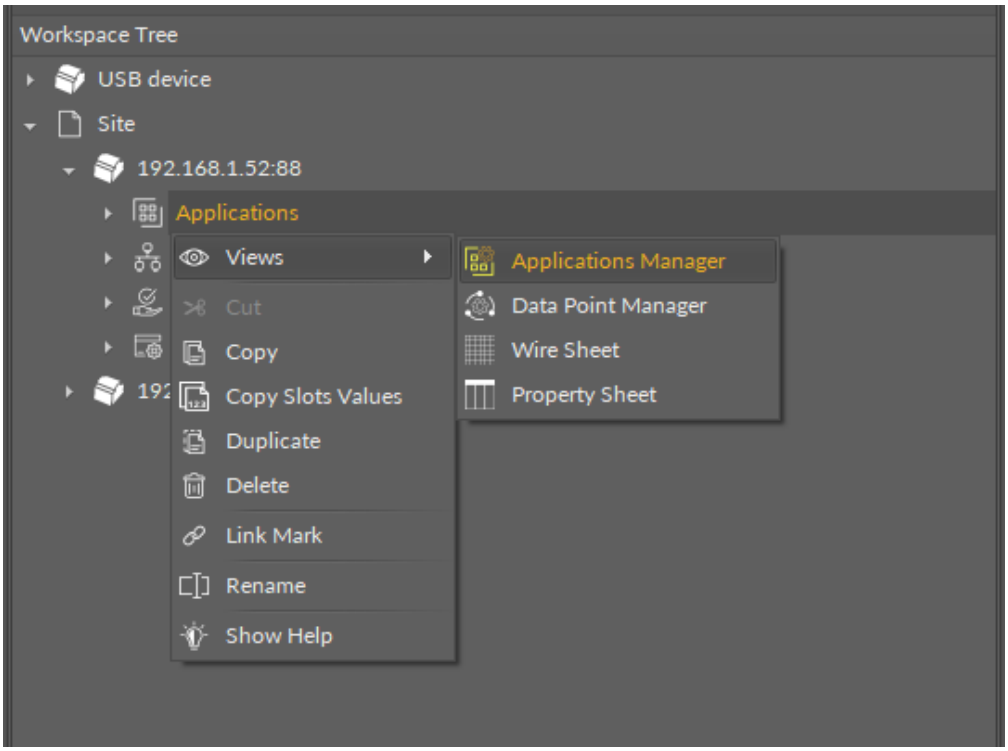
- name of the application;
- status;
- scan period;
- scan time;
- enabled or disabled status.

In the Applications Manager, it is possible to add, remove, copy, or duplicate Application components.



Opening the Applications Manager

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



The Applications Manager view is also automatically opened if the Applications container is double-clicked in the Workspace Tree window.

5.2 Applications Libraries

The nano EDGE ENGINE Applications libraries provide sets of components (service and basic) that allow to create HVAC and lighting applications. Libraries are grouped by specific functionalities, for example, by families of algorithms, etc. This section provides information about the **nano EDGE ENGINE** libraries designed for the Applications container, their functionalities and elements.

- Core library: including components fundamental to creating applications, such as Data Point, Equipment, etc.,
- FCU library: including components for FCU applications,
- Logic library: including components representing logical operations,
- Math library: including components representing mathematical operations,
- Other library: including components that can facilitate operating of an application or that fulfill the roles, which are not met by any other thematic libraries,
- Process library: including components essential to represent processes, which can be applied to HVAC management or other uses,
- Time library: including components representing time management operations essential for creating logics,
- Schedules library: including components that allow to set values in a logic according to schedules depending on calendar time,
- ComfortControl library: including components designed for comfort management,
- VAV library: including components designed for applications for various VAV systems,
- LightControl library: including components designed for light management,
- DALI library: including components converting values for DALI-2 communication (applicable in controllers with DALI port),
- Trends library: including extensions to Data Points allowing the trends functionality.

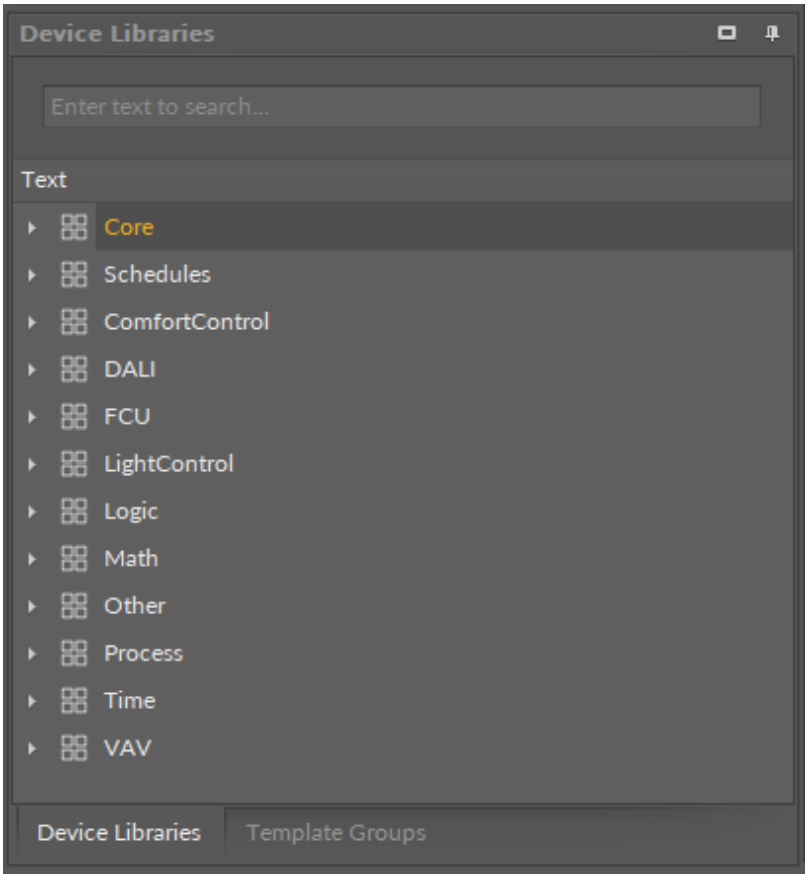


Figure 18. nano EDGE ENGINE libraries

6 Networks

The Networks is a container that provides a space for components supporting external communications of the controller, for example, components managing physical inputs and outputs or the ones enabling communication protocols. Components within the Networks container provide for the controller's basic requirements to be able to communicate with external systems; these components allow to exchange fundamental automation data, which are used to create algorithms operating in user applications.

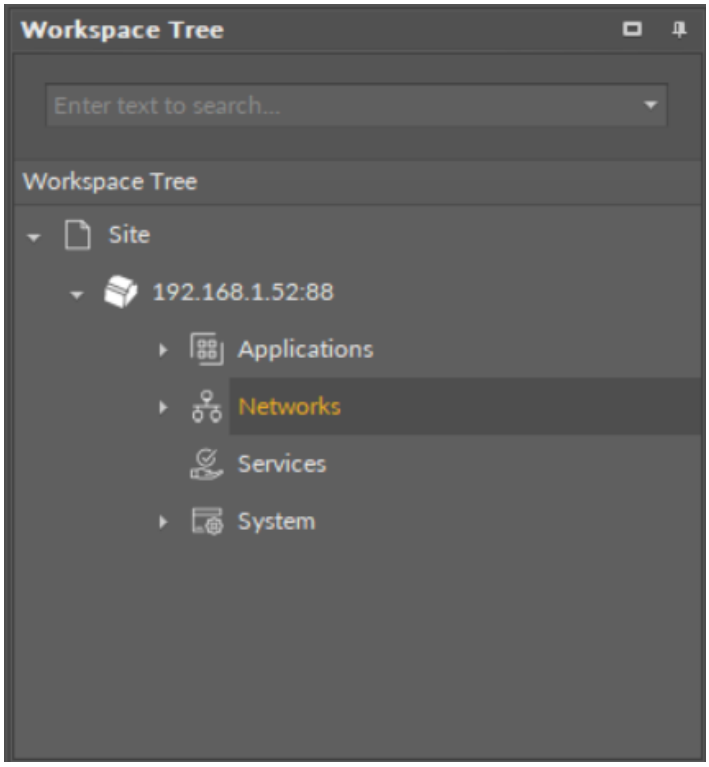


Figure 19. Networks container

6.1 Quick Start-up of LocalIO

In order to properly set up local IO components, it is required to follow these steps:

Step 1: Adding points

Having the device added correctly in the nano EDGE ENGINE software tool (iC Tool or nE2 Link module), expand the Networks container, and go to the LocalIO component.

Go to the Device Libraries and expand the IO library. Choose components to be added—components may be added one by one or grouped in one selection. Drag the selected component(s) and drop it(them) under the LocalIO component in the Networks container.

Step 2: Addressing

Go to each added component in the LocalIO, open its Property Sheet (or go the the LocalIO Property Sheet and expand each component), and set a proper number in the Address slot (a number representing the address of the physical input or output of the device). Configure all the other slots (Units, Type, etc.) according to the purpose of the component in the application. Make sure to save the changes.

> **Note**
> In order to facilitate working with LocalIO component, a special Point Manager view has been developed.

Ready to Use: Configured components are ready to be included in the application.

6.1.1 Point Manager for LocalIO

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

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

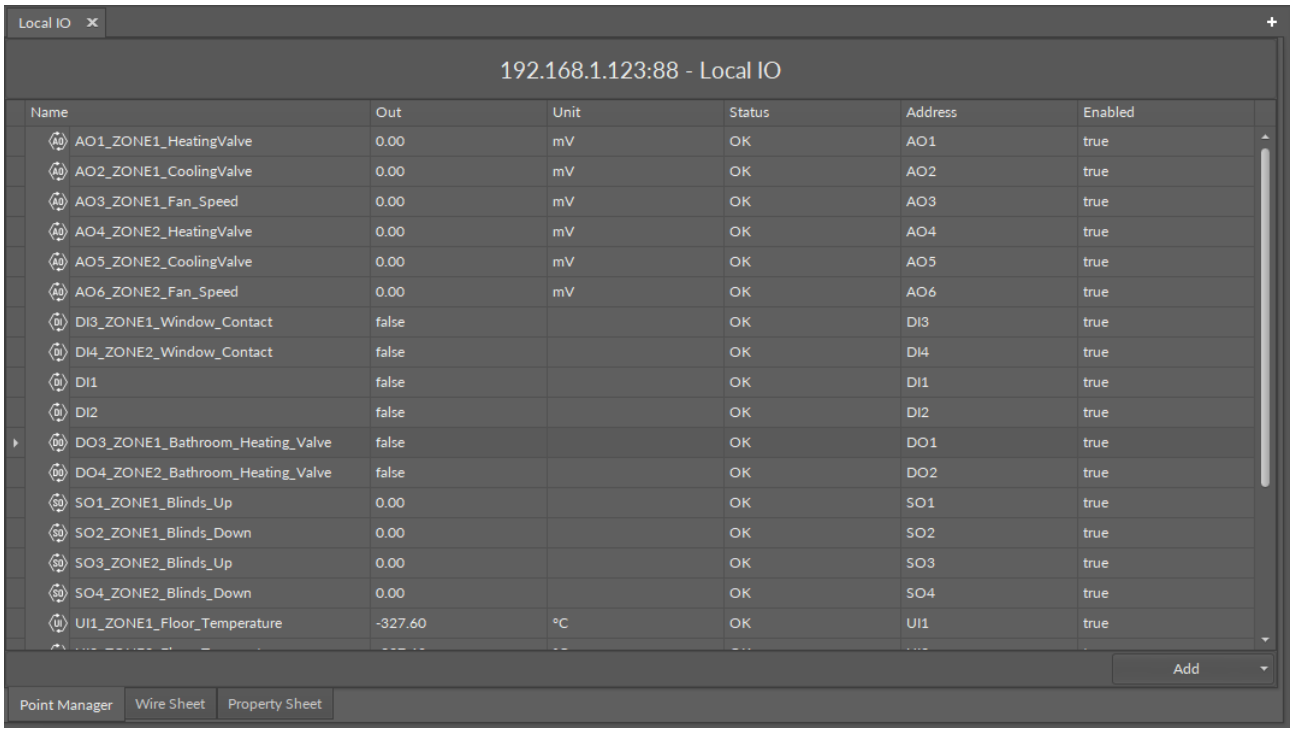


Figure 20. Point Manager

Opening Point Manager

The Point 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 Workspace Tree window.

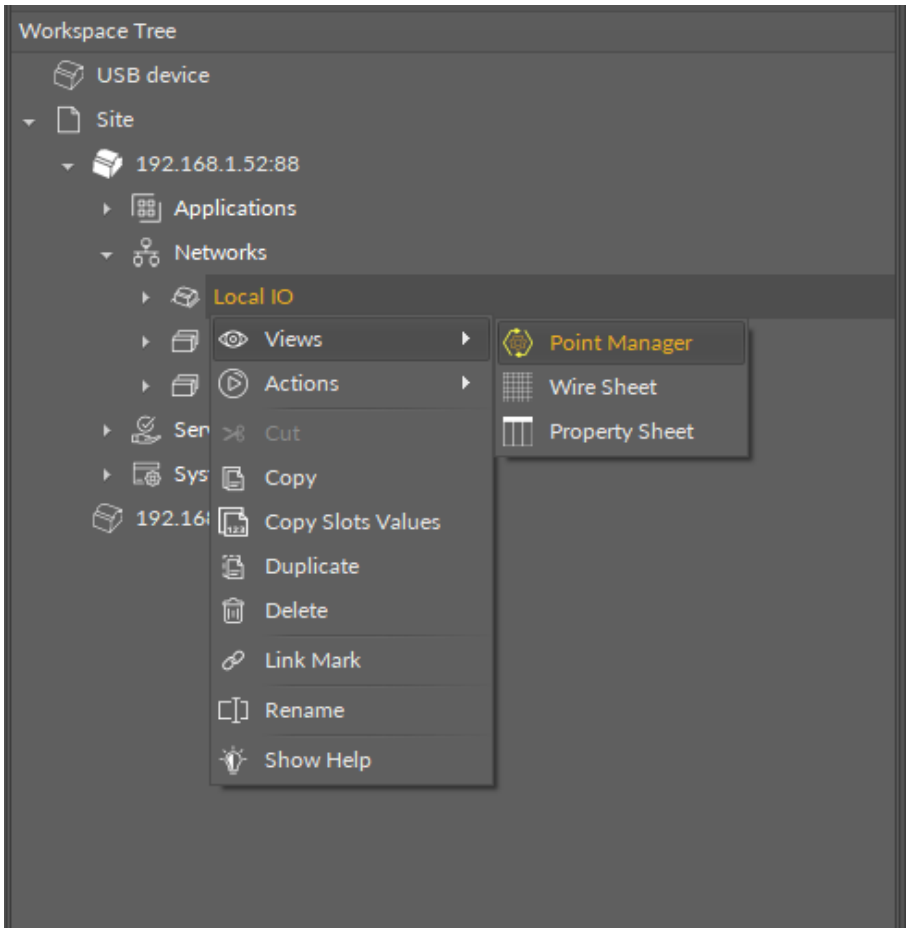


Figure 21. Opening the Point Manager

Adding I/O Points

The I/O points may be added twofold: dragging and dropping the I/O points to the LocalIO component from the IO library (in the Device Libraries window), or using a special Add function in the LocalIO Point Manager view available in the bottom right corner. The Add function allows to add any of the I/O points available in the IO library.

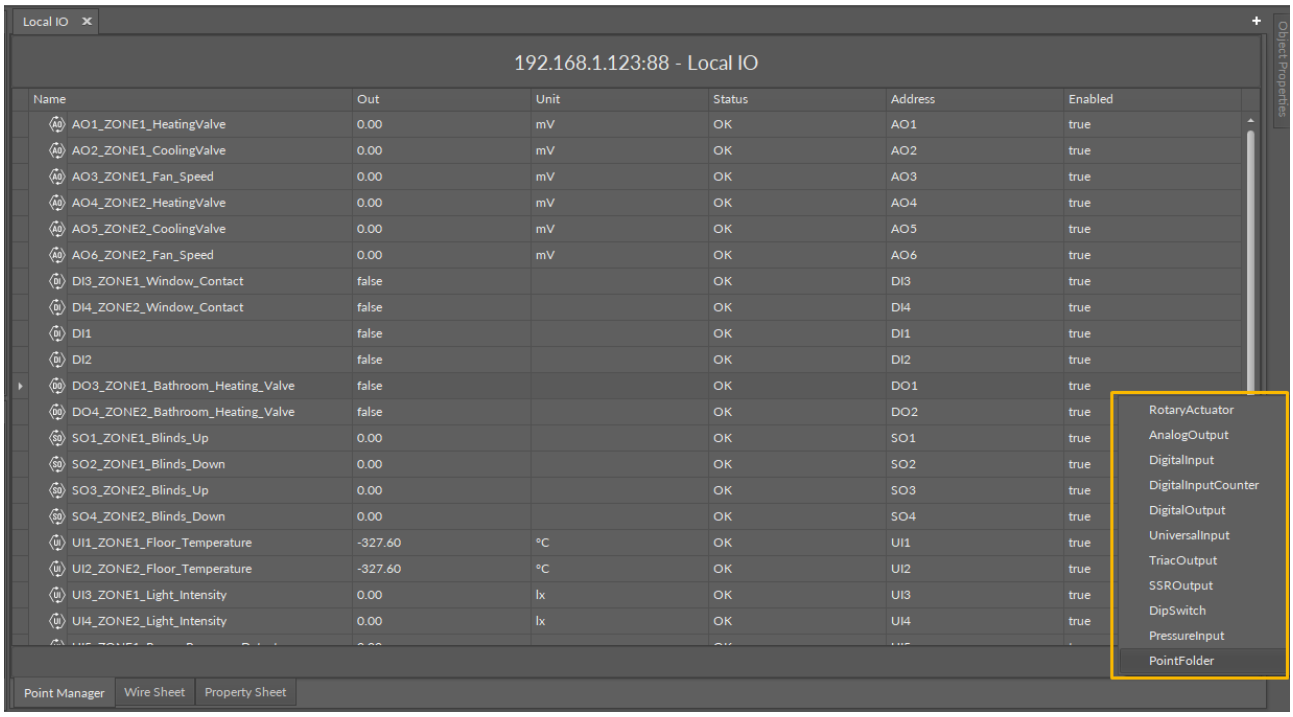


Figure 22. Adding I/O points

Using this Add button opens the dialog window, which allows to adjust the quantity of I/O points to be added.

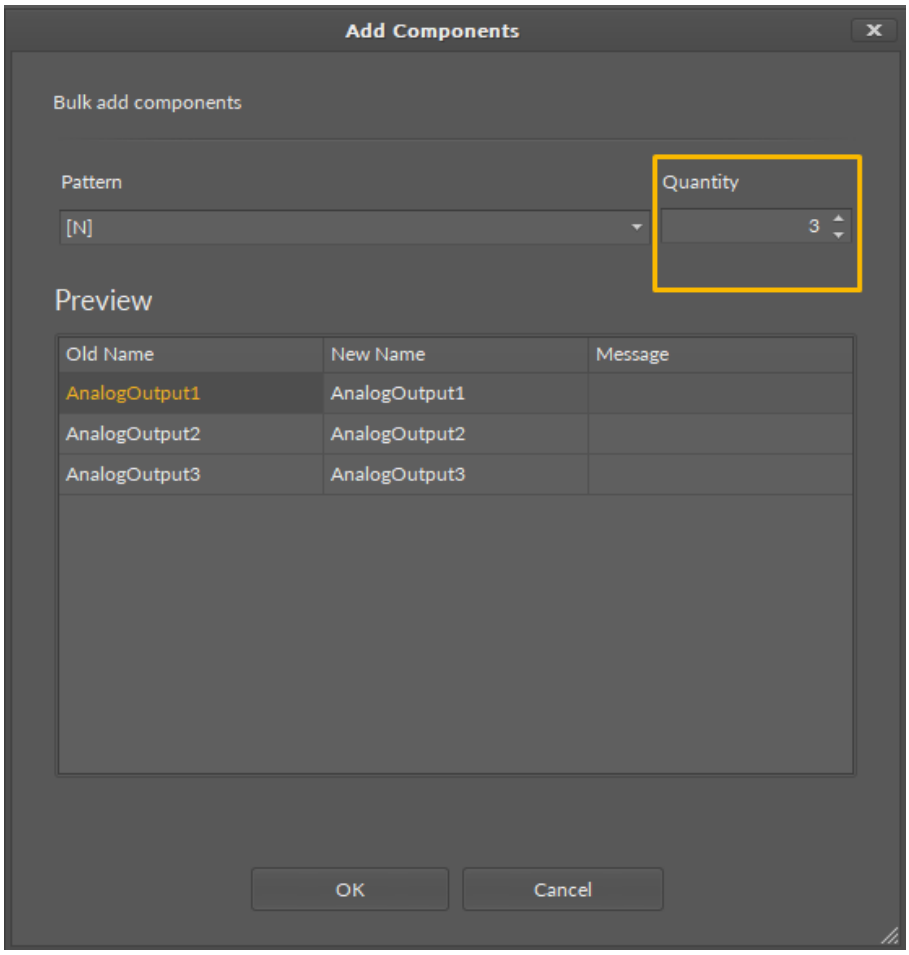


Figure 23. Dialog window for adding components

Multiediting of Common Slots

The Point Manager view allows multiediting of common slots in components of the same type, for example, to enable all one-type components at once. Multiediting is available in the Object Properties window, upon selecting one-type components in the Point Manager with Ctrl or Shift keys.

6.2 Quick Start-up of BACnet

In order to launch the BACnet communication properly, it is required to follow these steps:

Step 1: Port configuration

Having the device added correctly in the in the nano EDGE ENGINE software tool (iC Tool or nE2 Link module), expand the Networks container, and go to the BACnet component.

Go to the Interfaces component. The Ethernet components is added by default, the SerialPort component has to be added manually, if required.

(a) Go to the Ethernet component to configure the IP port for BACnet IP communication if necessary.

(b) Go to the SerialPort components to configure the RS485 port for BACnet MS/TP communication if necessary.

> Note
> Please remember that the device can communicate only one type of serial protocol on one port. If the BACnet MS/TP protocol is enabled on the serial port, the Modbus RTU protocol is disabled.

> Note
> Please remember that although the nano EDGE ENGINE system is designed to handle communication in BACnet IP, BACnet MS/TP, Modbus TCP/IP, and Modbus RTU protocol, the variants may differ between devices. Always make sure to check the device's hardware manual to check the available communication protocols.

Step 2: Setting up client/server network

The nano EDGE ENGINE is designed to handle controller's communication as the BACnet IP or MS/TP client devices and BACnet IP or MS/TP server devices.

(a) The LocalDevice component (automatically named after the device's model and serial number) for BACnet IP server device is also added by default. In order to configure the BACnet server device, go to that component.

> Note
> The UTCTimeSynchronization (BIBB - Device Management-UTCTimeSynchronization-B DM-UTC-B) service can be added as a BACnet server (Device-B) for all nano EDGE ENGINE devices.

(b) In order to configure the BACnet IP or MS/TP client network, the Network component has to be added. Go to the Device Libraries and expand the Core library. Drag the selected component and drop it under the BACnet component in the Networks container. Make sure to enable the component and select the relevant interface (Ethernet or serial port).

Step 3: Adding devices and their points

There are two ways to add devices available in the network:

(a) automatically discover BACnet devices using the BACnet Device Discover view;

(b) add devices manually: go to the Device Libraries and expand the BACnet library. Drag the Device component and drop it under the Network component.

There are two ways to add points available for the added device:

(a) automatically discover BACnet objects using the BACnet Object Discover view;

(b) add the relevant network point components:

- AnalogPoint
- BinaryPoint
- MultistatePoint
- AnalogCustomPoint
- BinaryCustomPoint
- StringCustomPoint

to the added Devices. Go to the BACnet library and drag and drop the selected component(s) under the Device component.

> **Note**
> Please remember that the components' hierarchy needs to be maintained here: the Network component has to be located under the BACnet component, the Device component under the Network component, and the network point class components have to be placed under the Device component.
> If the superior component is selected in the Workspace Tree window and its special view (Network Manager/Device Manager/Point Manager) is opened in the main screen, the Device Libraries shows only the components that can be added directly under it. For example, if the BACnet component is selected in the Workspace Tree window, the Device Libraries shows only the Network component in the Core library.

Go to each added component, open its Property Sheet (or go the the BACnet Property Sheet and expand each component), and set their parameters (enable, device Id, etc.).

> **More**
> In order to facilitate working with BACnet, special views have been developed:
> - Network Manager, available in the BACnet component;
> - Data Point Manager, available in the LocalDevice component;
> - Device Manager, available in the Network component;
> - Point Manager, available in the Device component.

Ready to Use: Configured components are ready for proper BACnet communication.

6.2.1 Network Manager for BACnet

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.

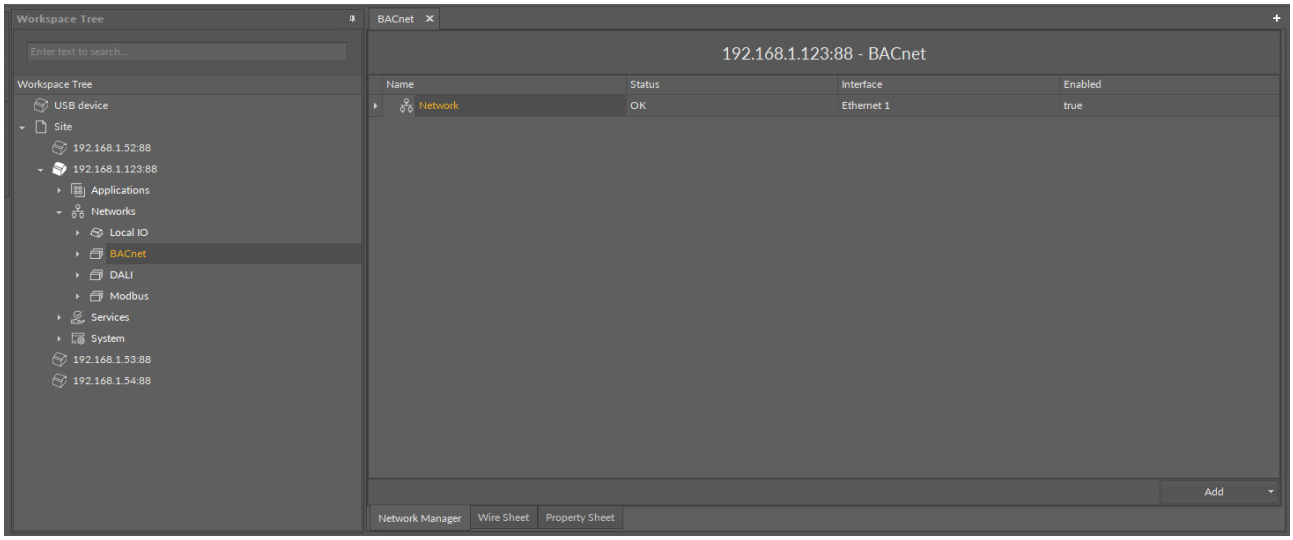


Figure 24. Network Manager for BACnet

Opening Network Manager

The 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 Workspace Tree window.

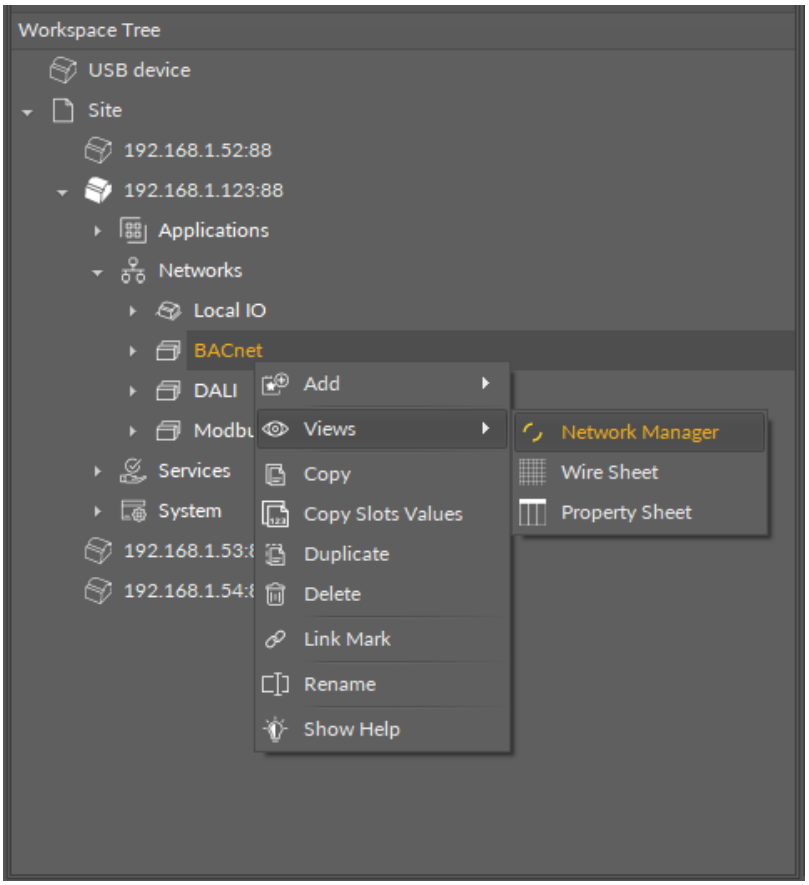
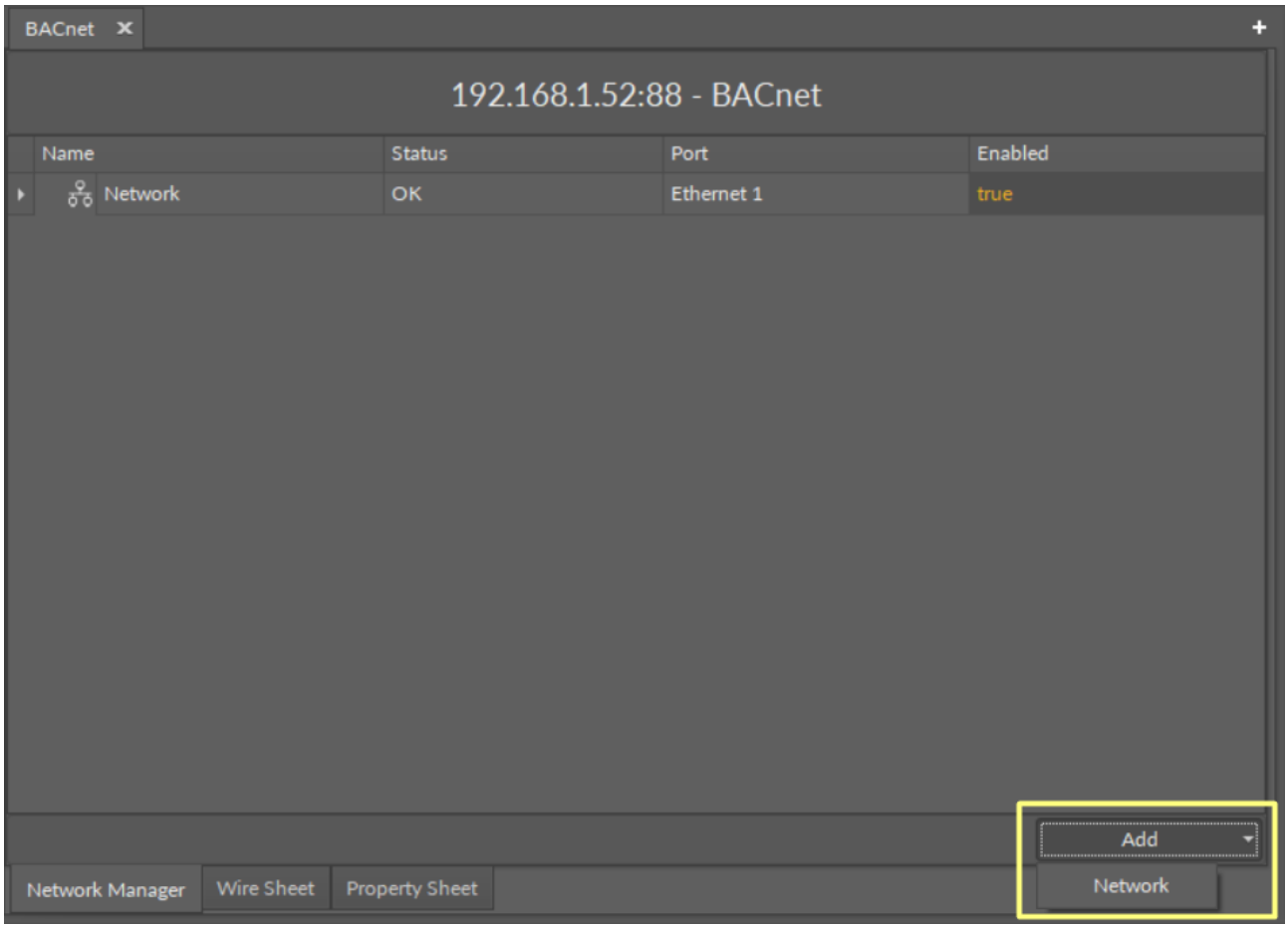


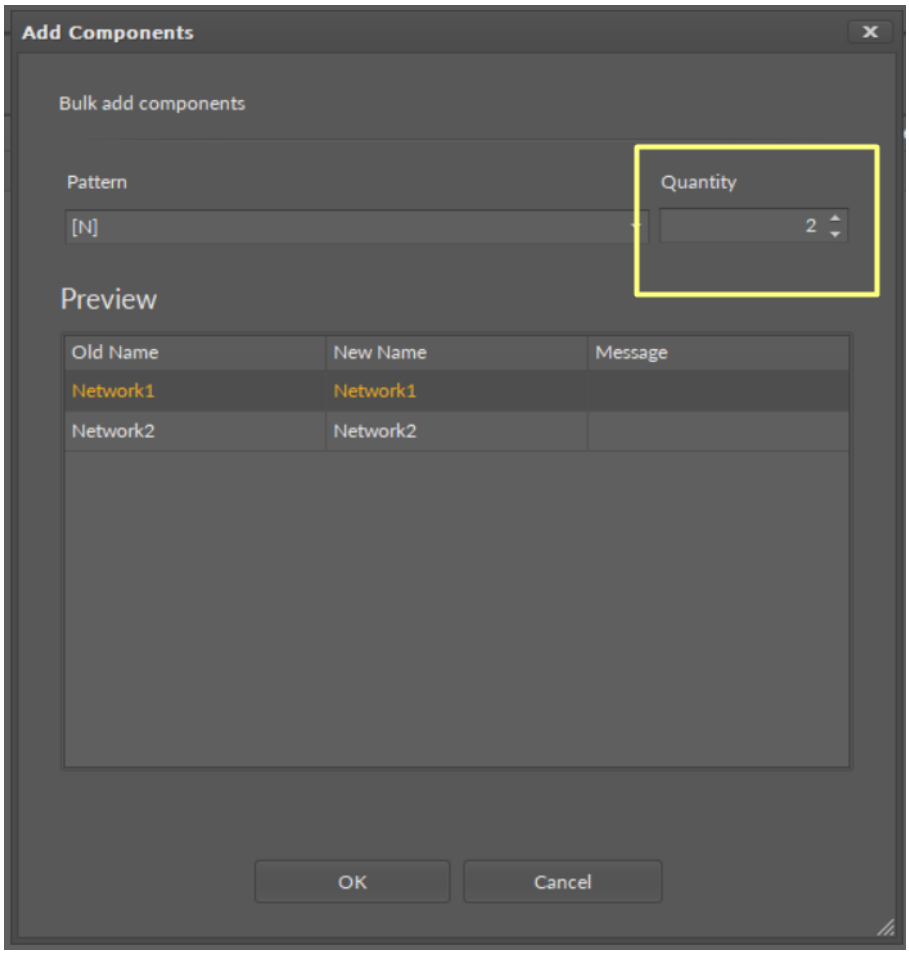
Figure 25. Opening the Network Manager from the context menu

Adding BACnet Networks

The networks may be added to the Network Manager twofold: dragging and dropping the Network component to the BACnet component from the Core library (in the Device Libraries window), or using a special Add function in the Network Manager view available in the bottom right corner.



Using this Add button opens the dialog window, which allows to adjust the quantity of networks to be added.



Multiediting of Common Slots

The Network Manager view allows multiediting of common slots in components of the same type, for example, to enable all one-type components at once. Multiediting is available in the Object Properties window, upon selecting one-type components in the Network Manager with Ctrl or Shift keys.

6.2.2 Device Manager for BACnet

The Device Manager view is available for the BACnet Network component. It lists all devices added to the configured BACnet network. The Device Manager view shows the statuses, BACnet device names and IDs, and enabled or disabled states of the devices in the network. Once the device listed in the Device Manager is double clicked, the respective Device component is opened.

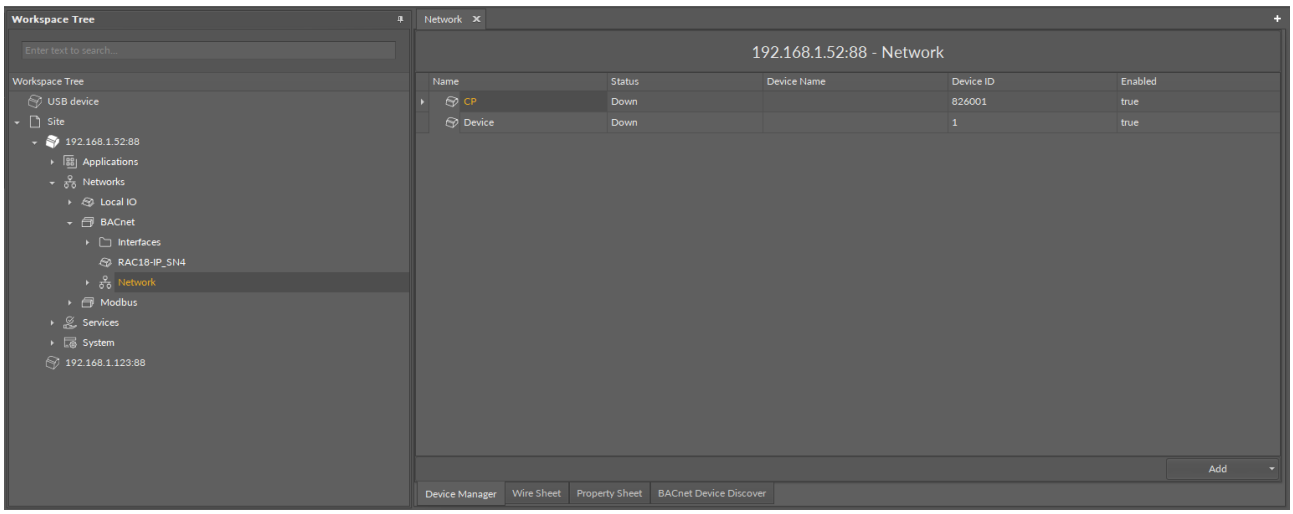


Figure 26. Device Manager for BACnet

Opening Device Manager

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

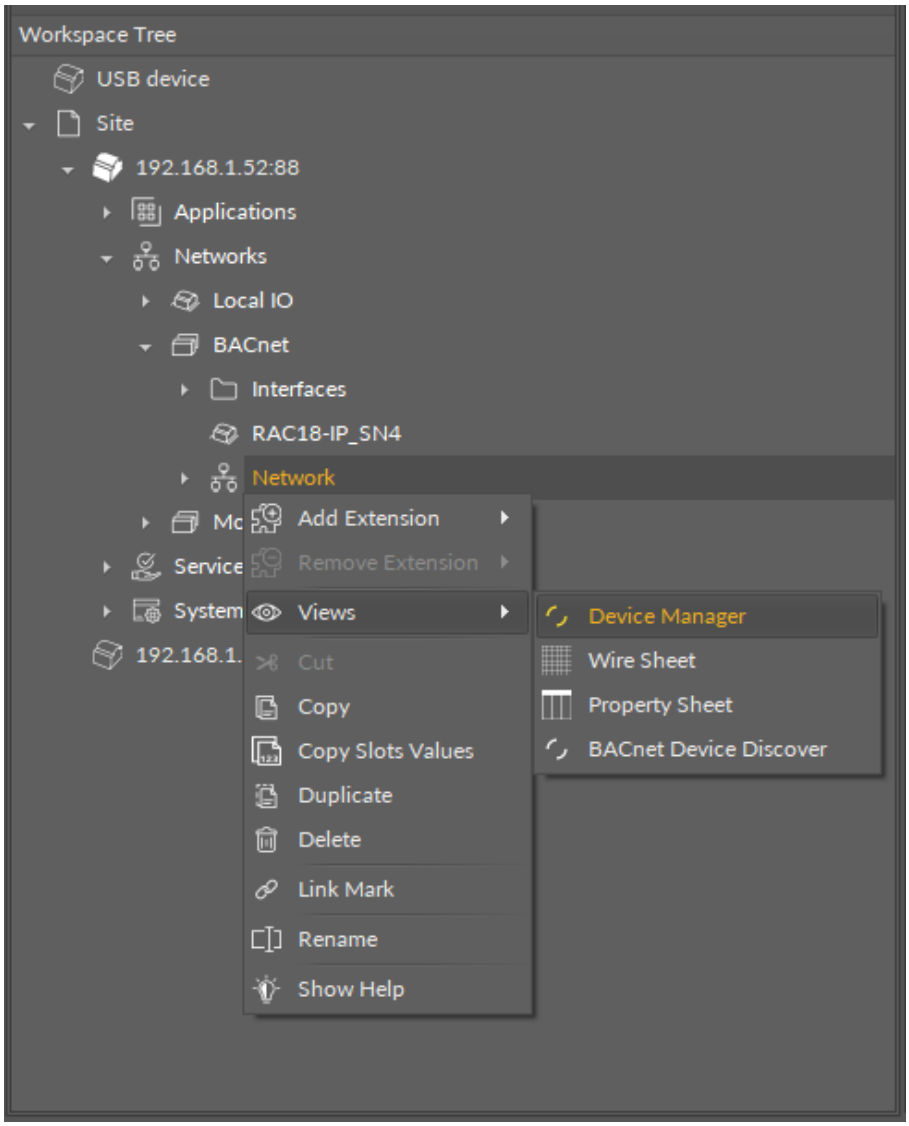


Figure 27. Opening Device Manager from the context menu

Adding BACnet Devices

The devices may be added to the BACnet network twofold: dragging and dropping the Device component to the Network component from the BACnet library (in the Device Libraries window), or using a special Add function in the Device Manager view available in the bottom right corner.

> Note
> A view facilitating adding BACnet devices is the BACnet Device Discover view.

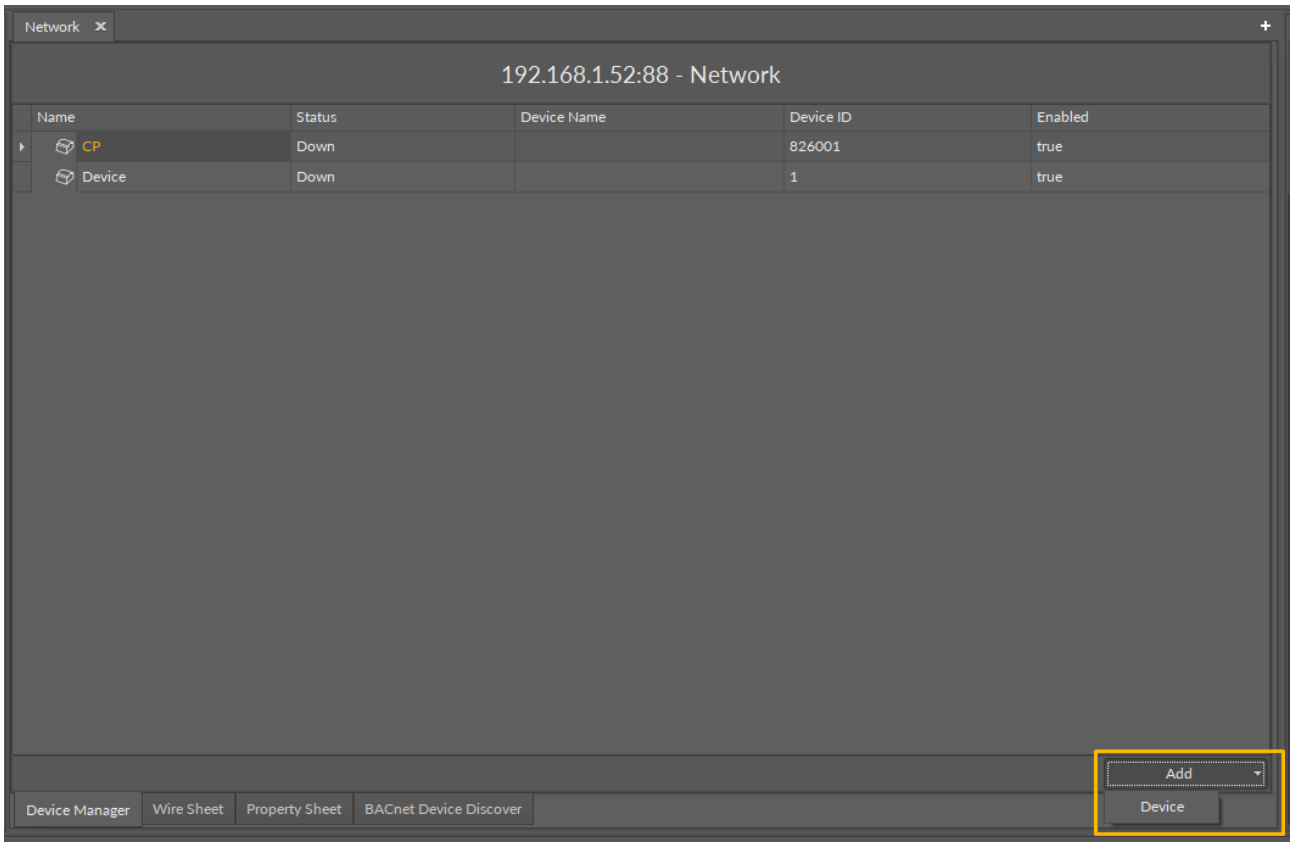


Figure 28. Adding device

Using this Add button opens the dialog window, which allows to adjust the quantity of devices to be added.

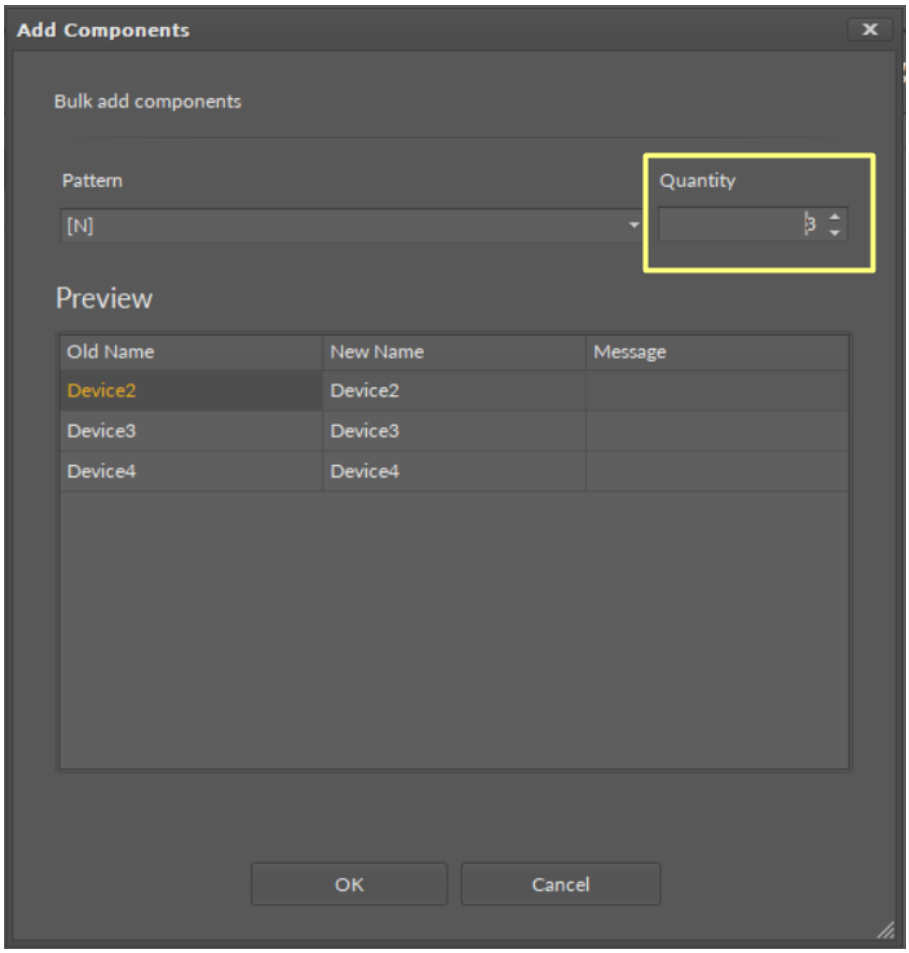


Figure 29. Add Components dialog window

Multiediting of Common Slots

The Device Manager view allows multiediting of common slots in components of the same type, for example, to enable all one-type components at once. Multiediting is available in the Object Properties window, upon selecting one-type components in the Device Manager with Ctrl or Shift keys.

6.2.3 BACnet Device Discover

The BACnet Device Discover view allows to automatically discover devices available in the BACnet IP or BACnet MS/TP network.

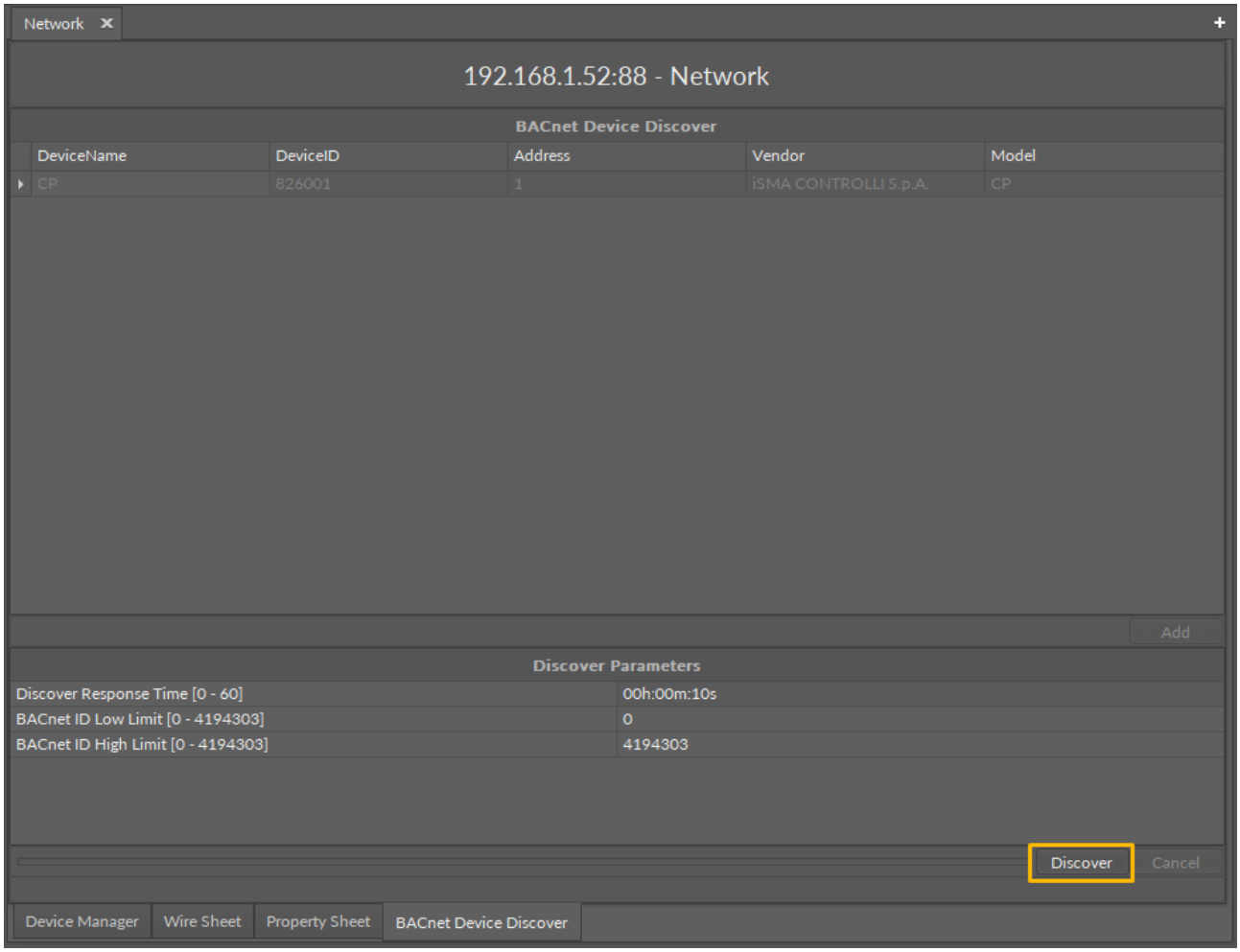


Figure 30. The BACnet Device Discover view

The BACnet Device Discover view is available for the Network component (upon double-click on the component).

It is structured with two windows, lower window which allows to set the discover parameters and initiate the discovering action (the Discover button). Next to the Discover button, there is a progress bar, which indicates how advanced is the discovering action.

The upper part of the view shows a list of discovered devices. It displays the following information in five columns:

- **DeviceName:** shows the discovered device's name;
- **DeviceID:** shows the device ID number provided by the BACnet organization to the vendor;
- **Address:** shows the MAC address attributed to the device;
- **Vendor:** shows the name of the device's vendor;
- **Model:** shows the device's product code.

Discovered devices can be easily added to the Network component. Select the devices on the list and click the Add button.

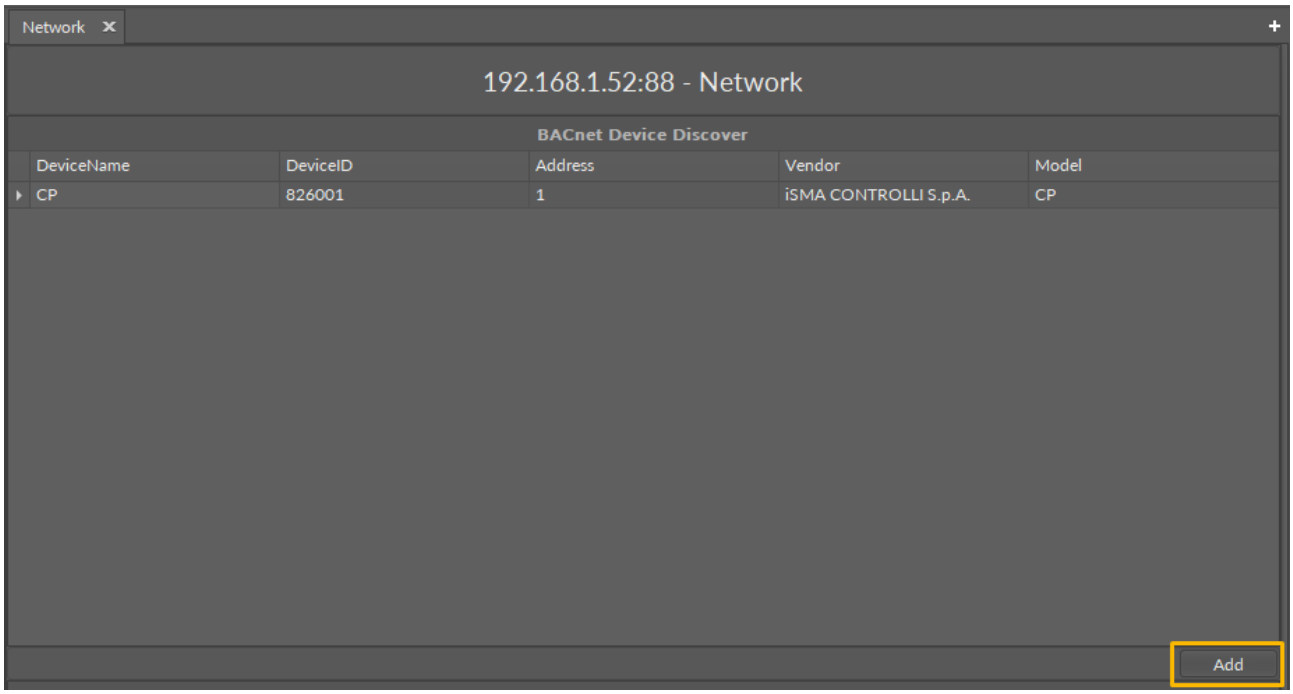


Figure 31. Adding discovered devices

The added devices will be listed under the Network component in the Workspace Tree.

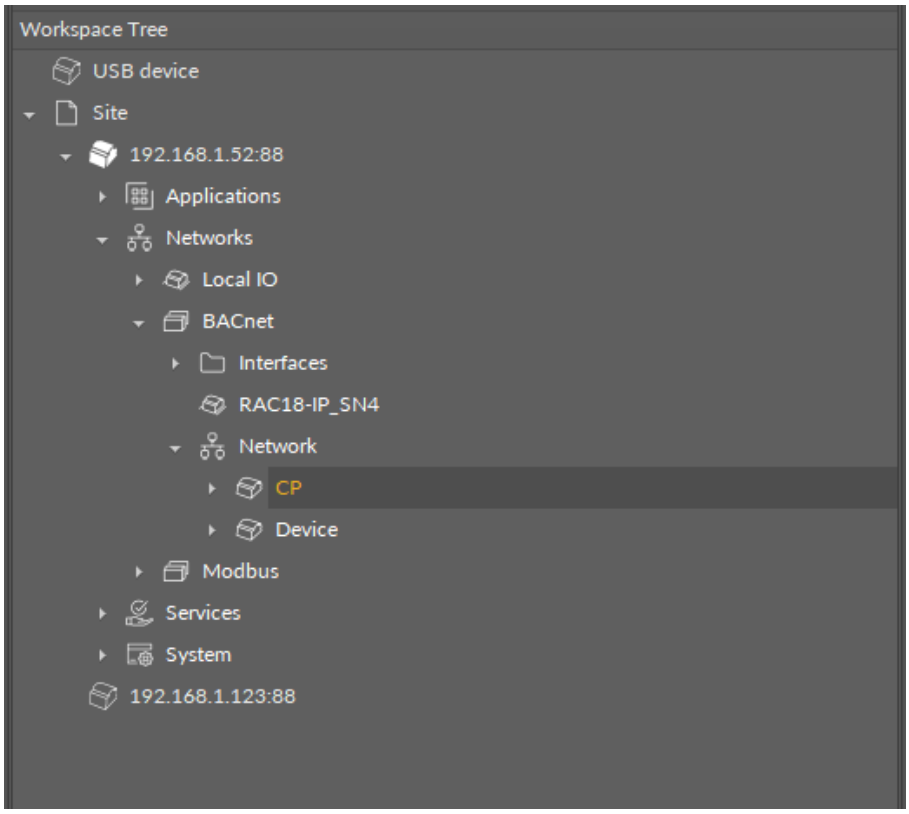


Figure 32. Added device

To add the device's points, proceed to the BACnet Object Discover view.

6.2.4 Point Manager for BACnet

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

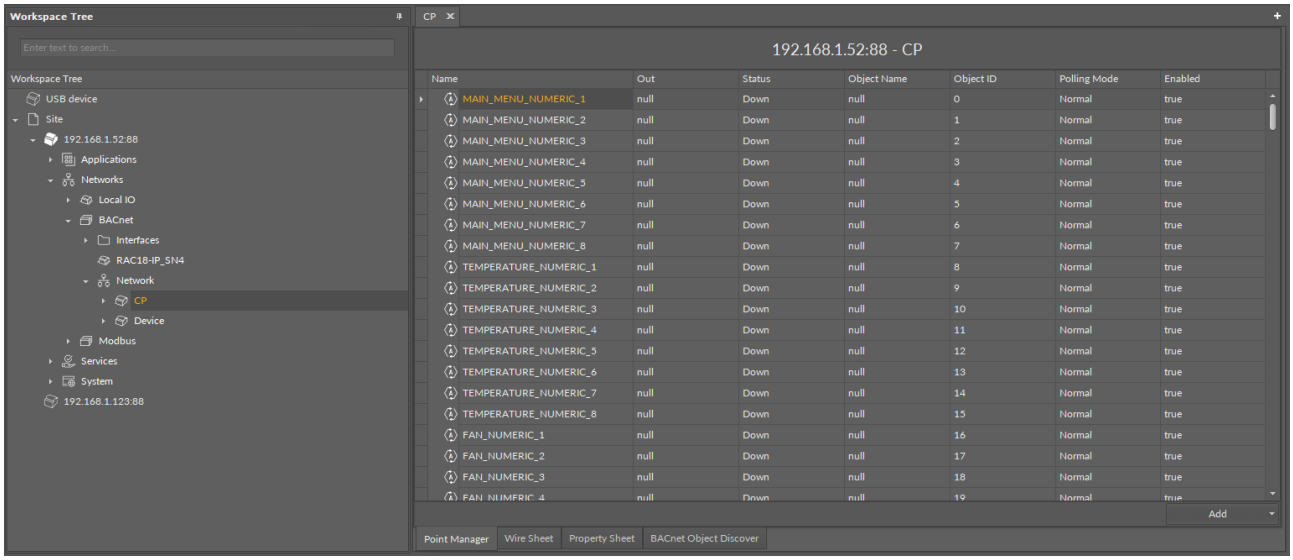


Figure 33. Point Manager for BACnet

Opening the Point Manager

The 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 Workspace Tree window.

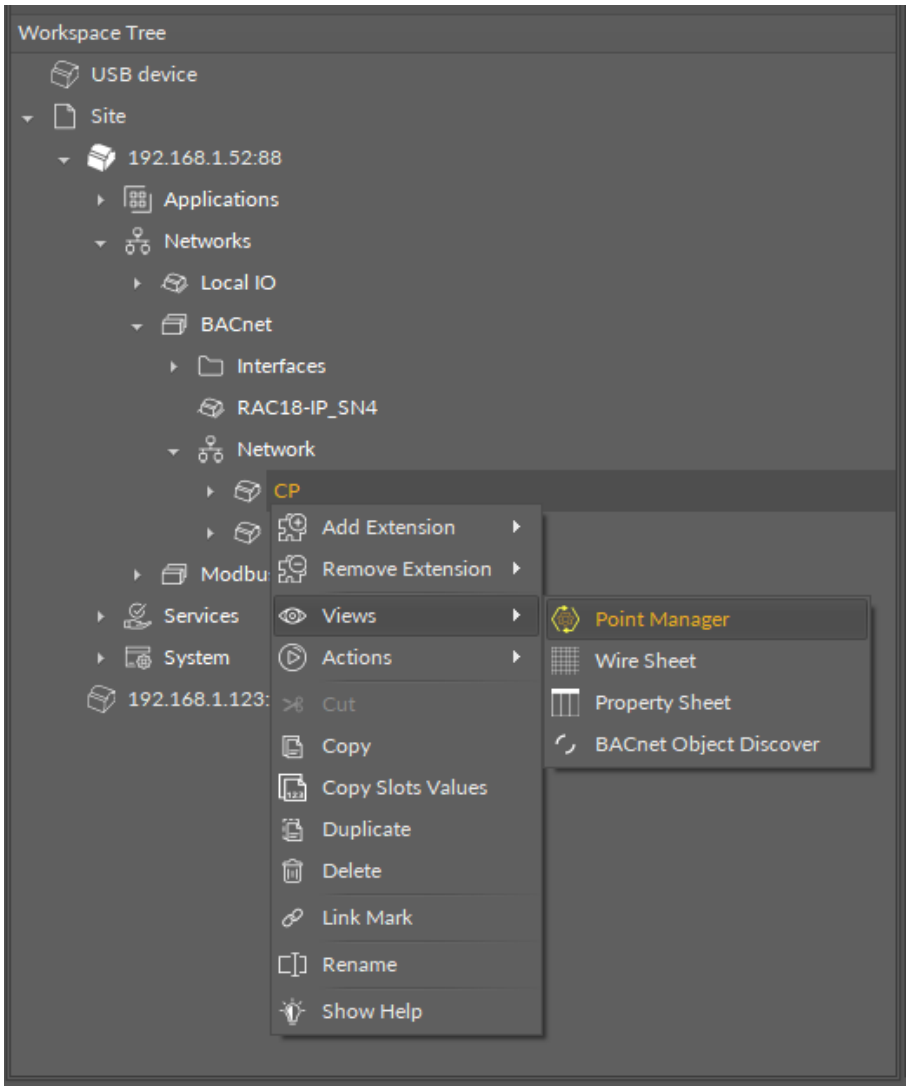


Figure 34. Opening the Point Manager from the context menu

Adding BACnet Points

The BACnet points may be added to the device twofold: dragging and dropping the BACnet points to the Device component from the BACnet library (in the Device Libraries window), or using a special Add function in the Point Manager view available in the bottom right corner. The Add function allows to add any of the BACnet points available in the BACnet library.

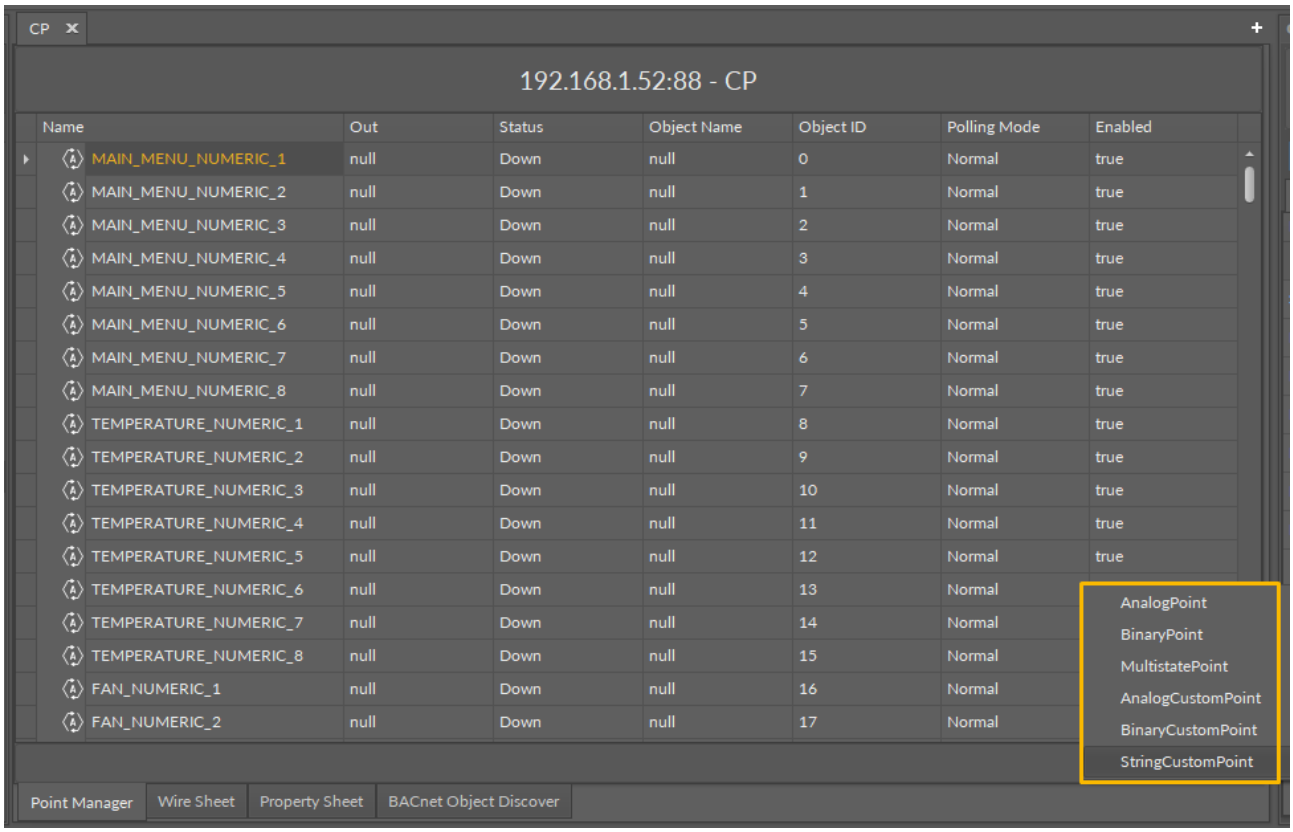


Figure 35. Adding points

Using this Add button opens the dialog window, which allows to adjust the quantity of BACnet Points to be added.

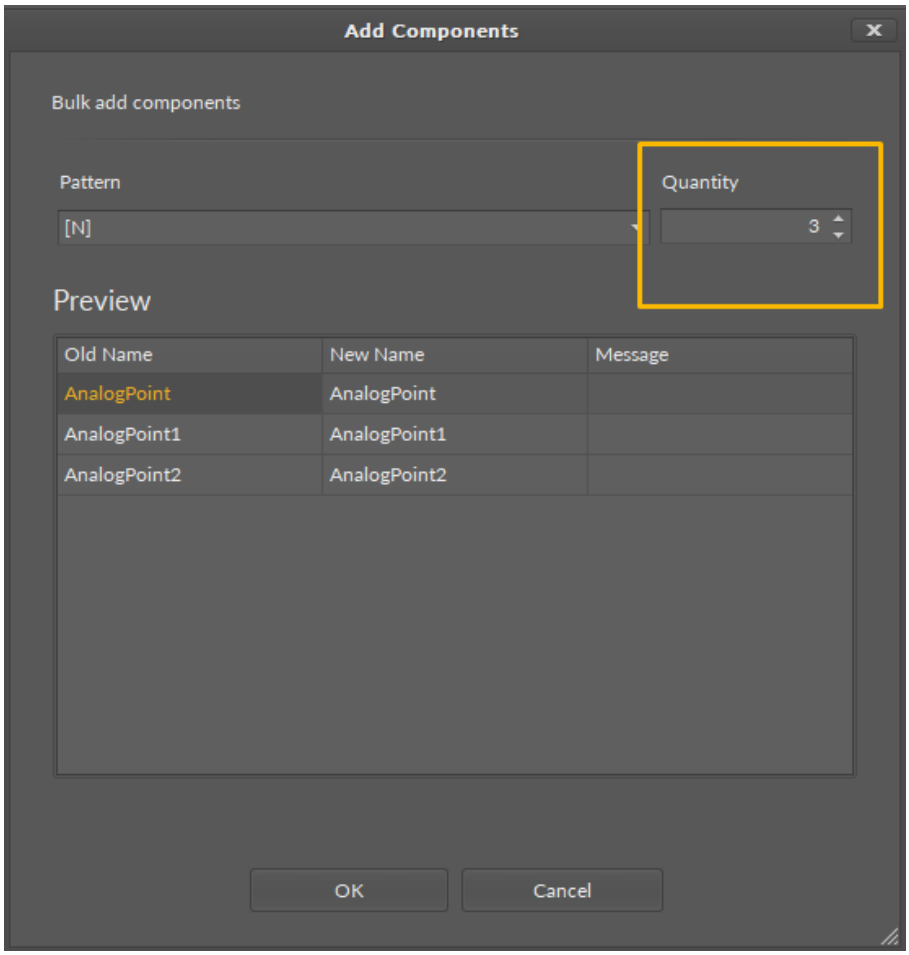


Figure 36. Adding points dialog window

Multiediting of Common Slots

The Point Manager view allows multiediting of common slots in components of the same type, for example, to enable all one-type components at once. Multiediting is available in the Object Properties window, upon selecting one-type components in the Point Manager with Ctrl or Shift keys.

6.2.5 BACnet Object Discover

The BACnet Object Discover view allows to automatically discover points available in the devices added to the BACnet network.

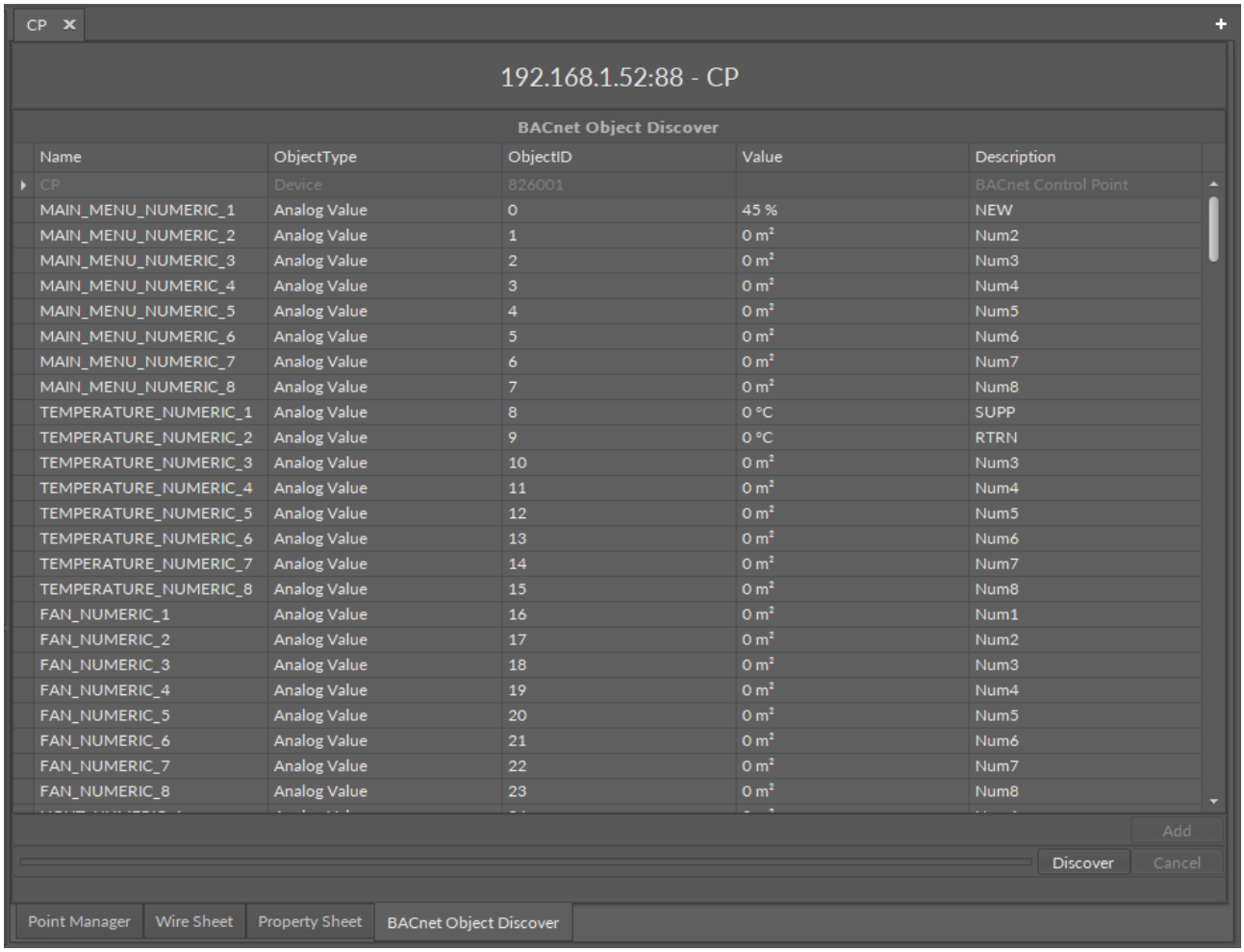


Figure 37. BACnet Object Discover

The BACnet Object Discover view is available for the Device component in the BACnet network (upon double-click on the component).

Once opened for the selected device, the view allows to initiate the discovery action, with the Discover button. All points discovered for the device are listed with the following information in five columns:

- **Name:** shows the discovered point's name;
- **ObjectType:** shows the BACnet object type of the point;
- **ObjectID:** shows the object ID of the point;
- **Value:** shows a current value of the point;
- **Description:** shows an additional description.

First, grayed out row shows information about the device the points are discovered for.

Discovered points can be easily added to the Device component. Select the points on the list and click the Add button.

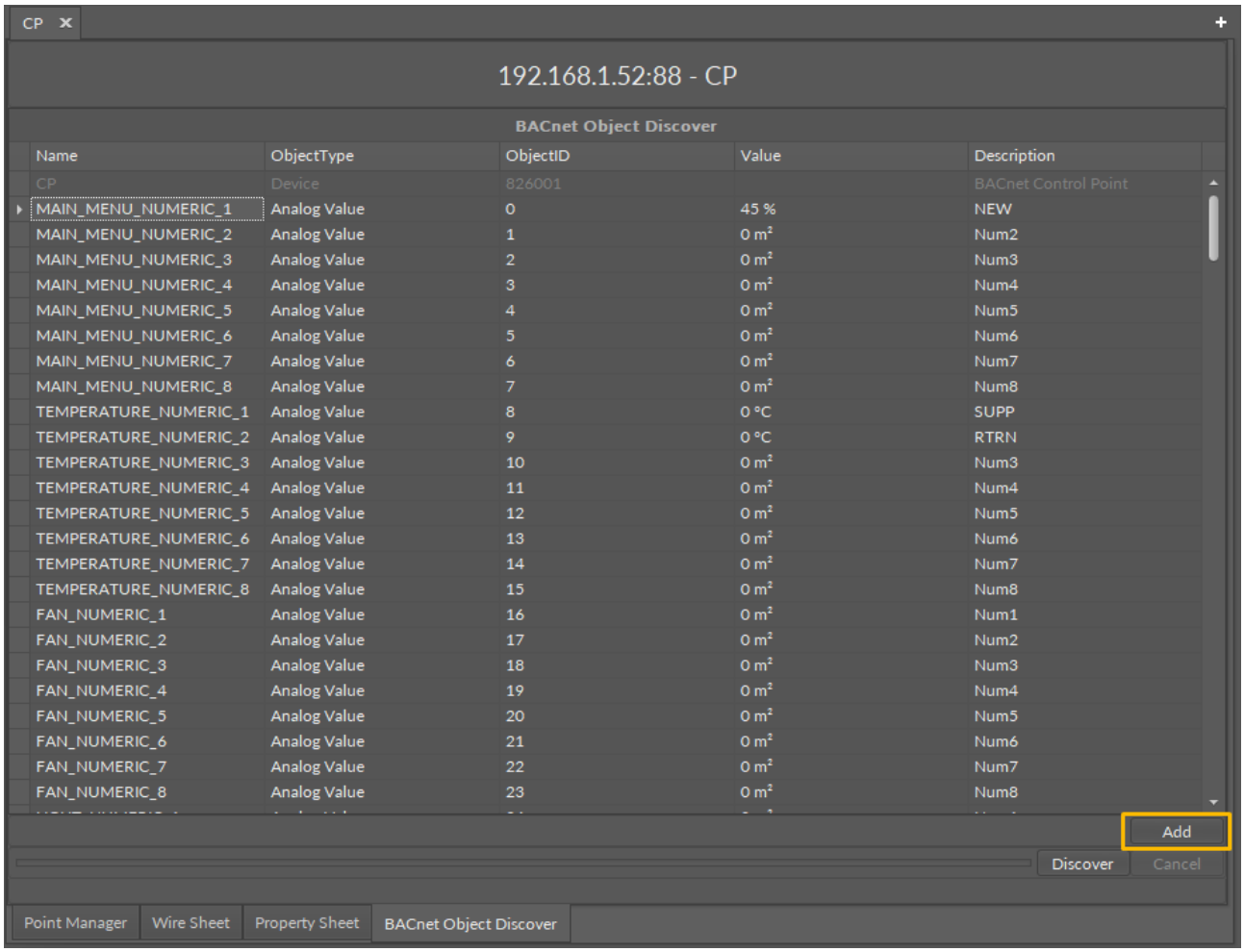


Figure 38. Adding points

The added points will be listed under the Device component in the Workspace Tree.

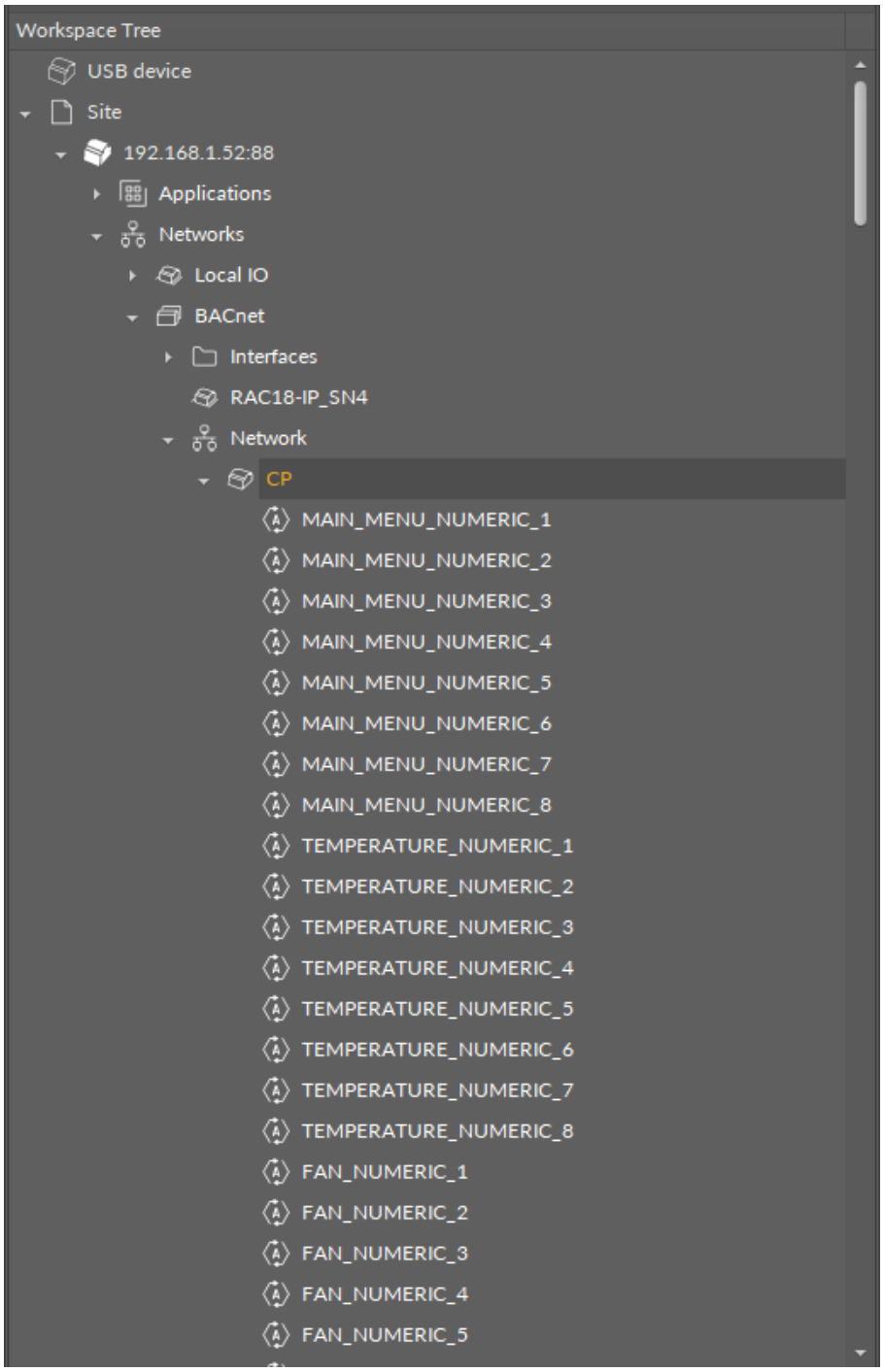


Figure 39. Added points

6.3 Quick Start-up of Modbus

In order to launch the Modbus communication properly, it is necessary to follow these steps:

Step 1: Port configuration

Having the device added correctly in the iC Tool, expand the Networks container, and go to the Modbus component.

Go to the Interfaces component. The Ethernet components is added by default, the SerialPort component has to be added manually, if required.

(a) Go to the Ethernet component to configure the IP port for Modbus TCP/IP communication if necessary.

(b) Go to the SerialPort components to configure the RS485 port for Modbus RTU communication if necessary.

Note
Please remember that the device can communicate only one type of serial protocol on one port. If the Modbus RTU protocol is enabled on the serial port, the BACnet MS/TP protocol is disabled.

Note
Please remember that although the nano EDGE ENGINE system is designed to handle communication in BACnet IP, BACnet MS/TP, Modbus TCP/IP, and Modbus RTU protocol, the variants may differ between devices. Always make sure to check the device's hardware manual to check the available communication protocols.

Step 2: Setting up client/server network

The nano EDGE ENGINE is designed to handle controller's communication as the Modbus RTU client or server device and Modbus TCP/IP client or server device.

(a): The LocalDevice component (automatically named after the device's model and serial number) for Modbus TCP/IP or RTU server device is also added by default. In order to configure the Modbus server device, go to that component.

Note
The LocalDevice component has a Modbus gateway extension, which allows Modbus RTU server devices (for example, RS485 I/O modules) to communicate with the Modbus TCP/IP network through a gateway device.

(b): In order configure the Modbus RTU or TCP/IP network, the Network and Device components need to be added, along with AnalogPoint, BinaryPoint, MultistatePoint, and StringPoint components, as necessary. Go to the Device Libraries and expand the Core library for the Network component and Modbus library for other components. Choose components to be added—components may be added one by one or grouped in one selection. Drag the selected component(s) and drop it(them) under the Modbus component in the Networks container.

Note
Please remember that the components' hierarchy needs to be preserved here: the Network component has to be located under the Modbus component, the Device component under the Network component, and the AnalogPoint/BinaryPoint/MultistatePoint/StringPoint components have to be placed under the Device component.

If the superior component is selected in the Workspace Tree window and its special view (Network Manager/Device Manager/Point Manager) is opened in the main screen, the Device Libraries shows only the components that can be added directly under it. For example, if the Modbus component is selected in the Workspace Tree window, the Device Libraries shows only the Network component in the Core library.

Go to each added component, open its Property Sheet (or go the the Modbus component's Property Sheet and expand each component there), and set their parameters (enable, addresses, etc.). Make sure to save the changes.

More

In order to facilitate working with Modbus, special views have been developed:

- Network Manager, available in the Modbus component;
- Data Point Manager, available in the LocalDevice component;
- Device Manager, available in the Network component;
- Point Manager, available in the Device component.

Ready to Use: Configured components are ready for proper Modbus communication.

6.3.1 Network Manager for Modbus

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 the network. Once the network, listed in the Network Manager, is double-clicked, the respective Network component is opened.

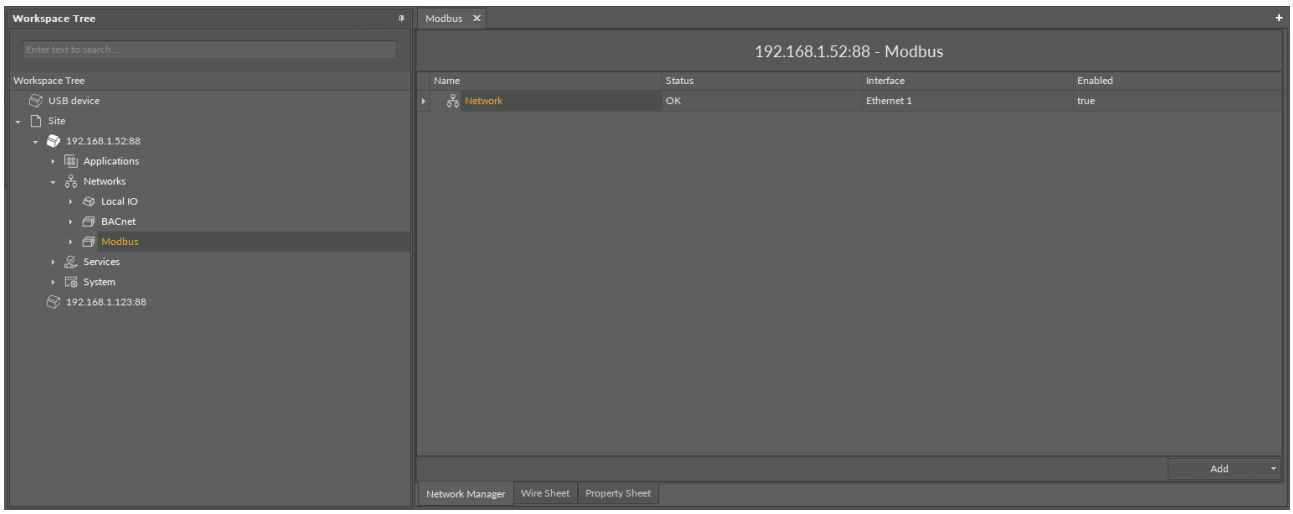


Figure 40. Network Manager for Modbus

Opening Network Manager

The 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 Workspace Tree window.

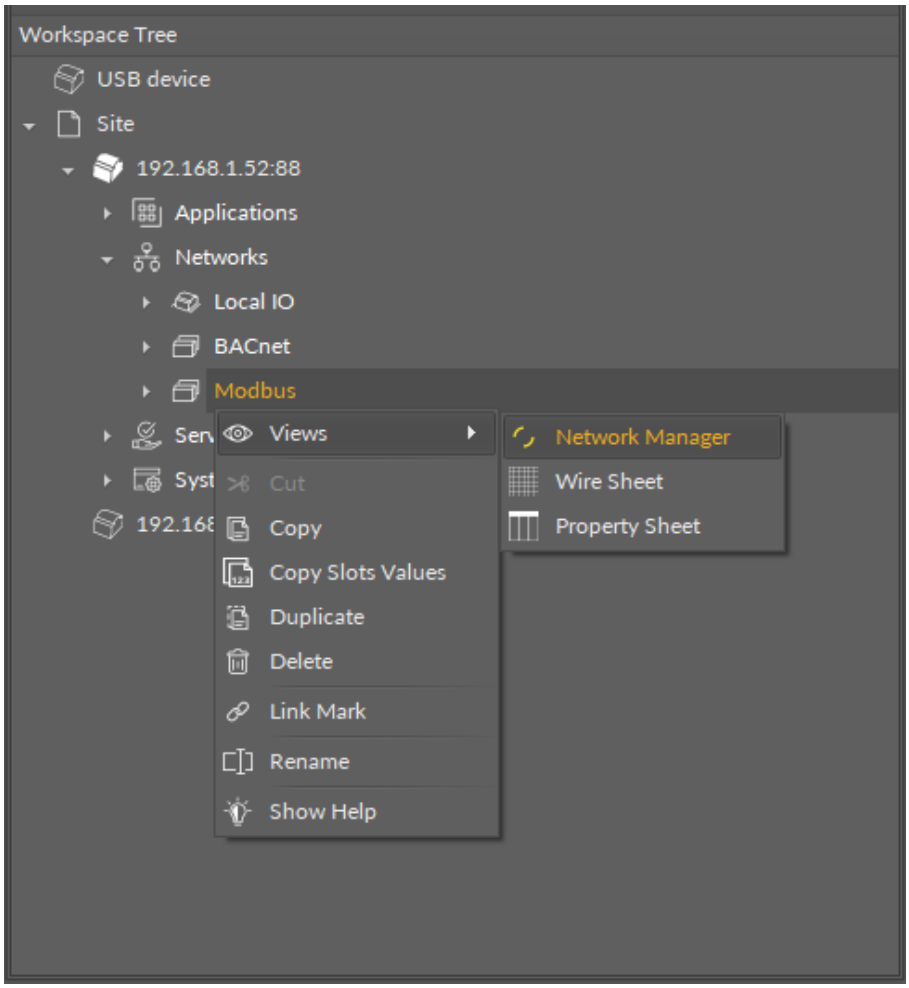


Figure 41. Opening the Network Manager from the context menu

Adding Modbus Networks

The networks may be added to the Network Manager twofold: dragging and dropping the Network component to the Modbus component from the Core library (in the Device Libraries window), or using a special Add function in the Network Manager view available in the bottom right corner.

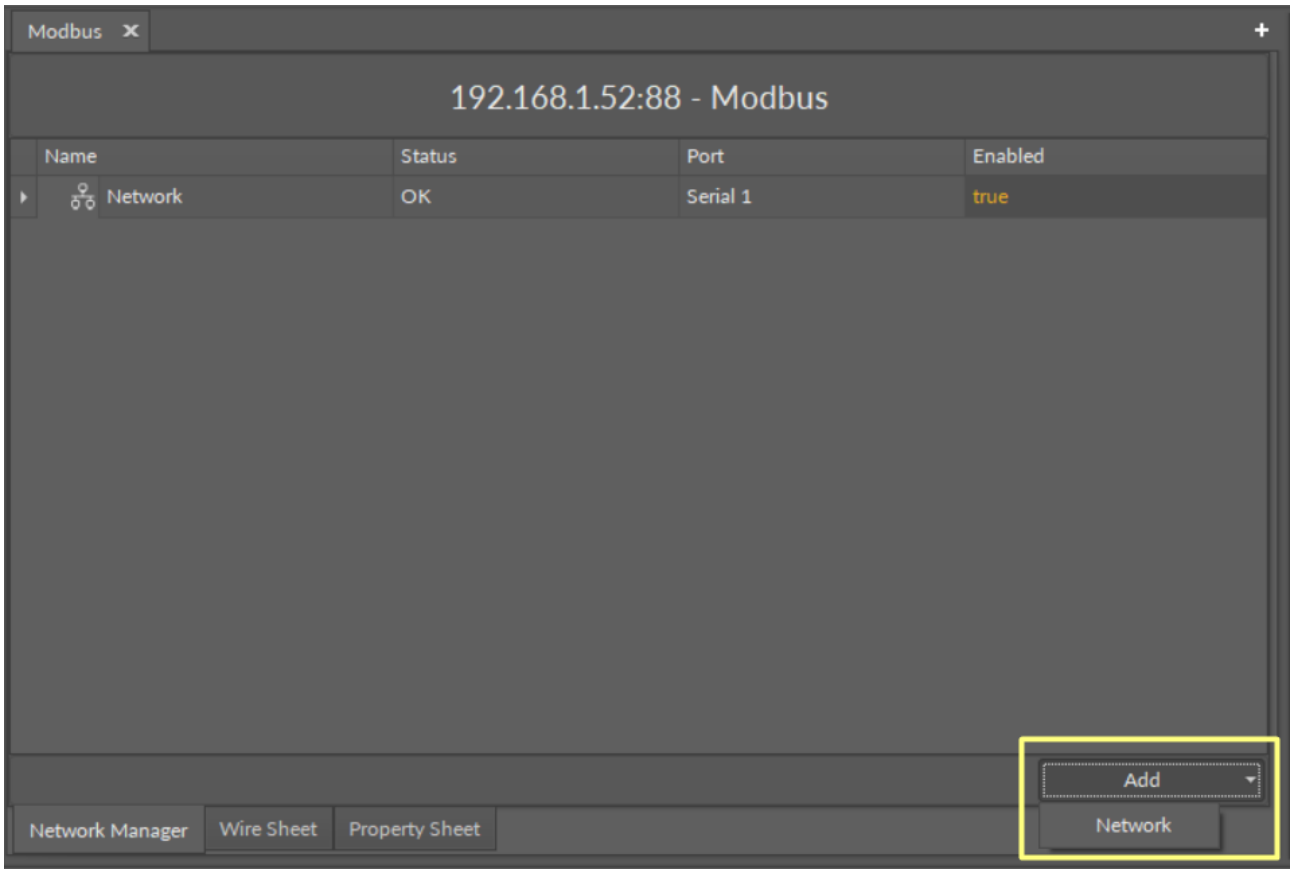


Figure 42. Adding Network component

Using this Add button opens the dialog window, which allows to adjust the quantity of networks to be added.

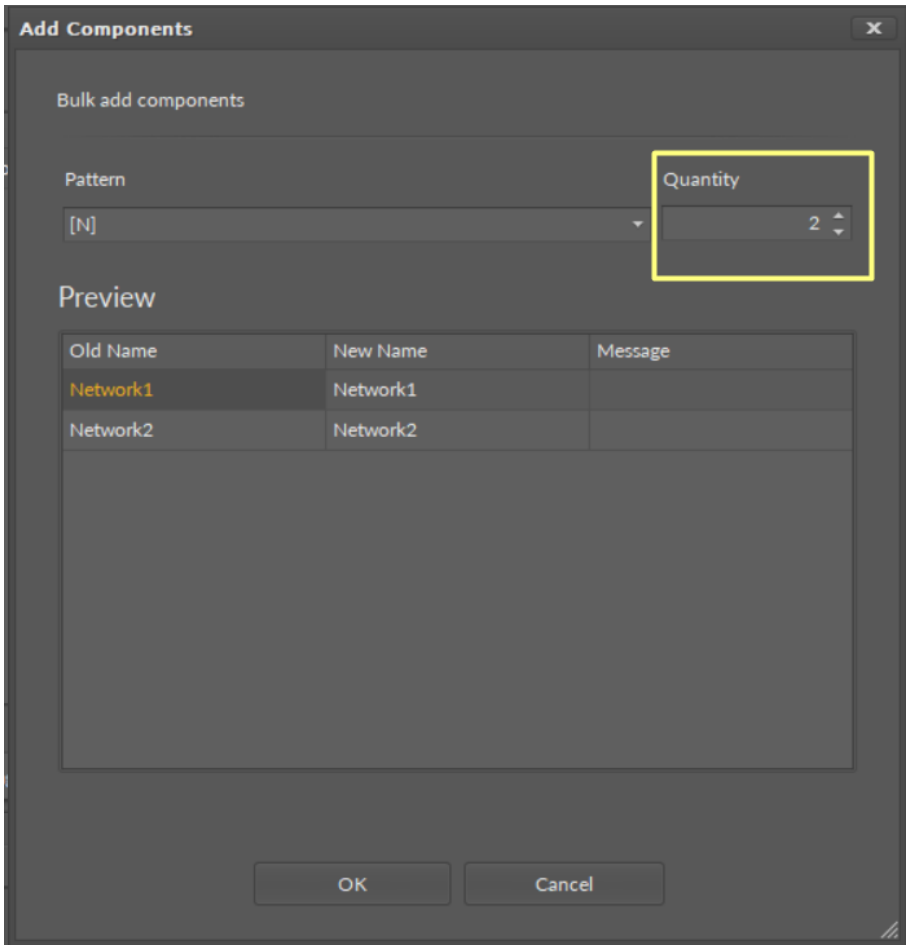


Figure 43. Add Components dialog window

Multiediting of Common Slots

The Network Manager view allows multiediting of common slots in components of the same type, for example, to enable all one-type components at once. Multiediting is available in the Object Properties window, upon selecting one-type components in the Network Manager with Ctrl or Shift keys.

6.3.2 Device Manager for Modbus

The Device Manager view is available for the Modbus Network component. It lists all devices added to the configured Modbus network. The Device Manager view shows the statuses, device addresses, and enabled or disabled states of the devices in the network. Once the device, listed in the Device Manager, is double clicked, the respective Device component is opened.

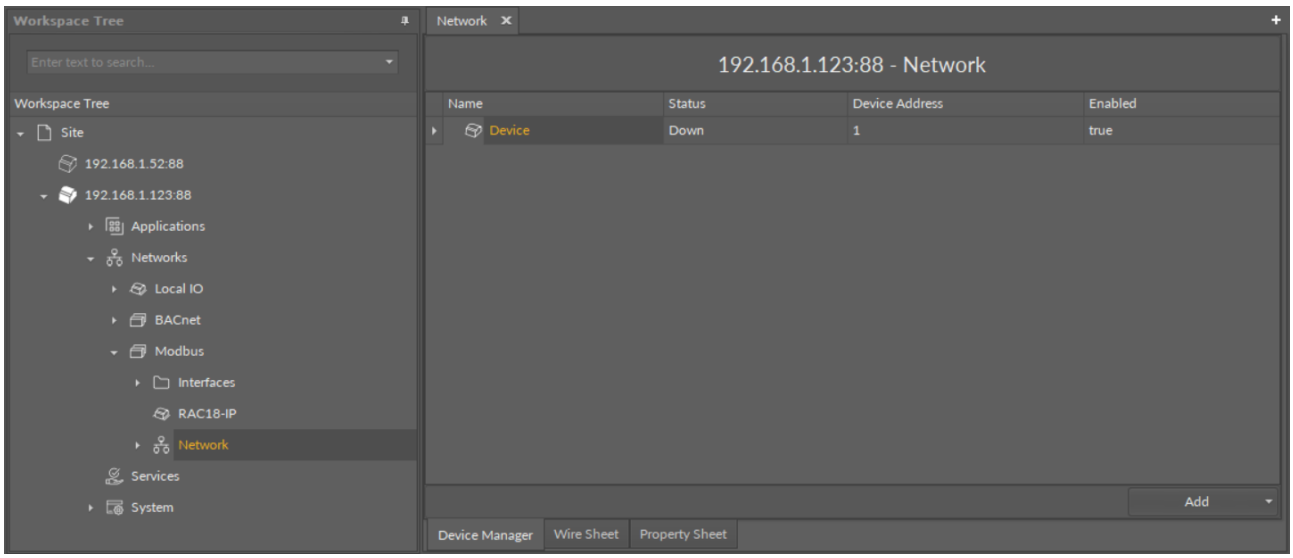


Figure 44. Device Manager for Modbus

Opening Device Manager

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

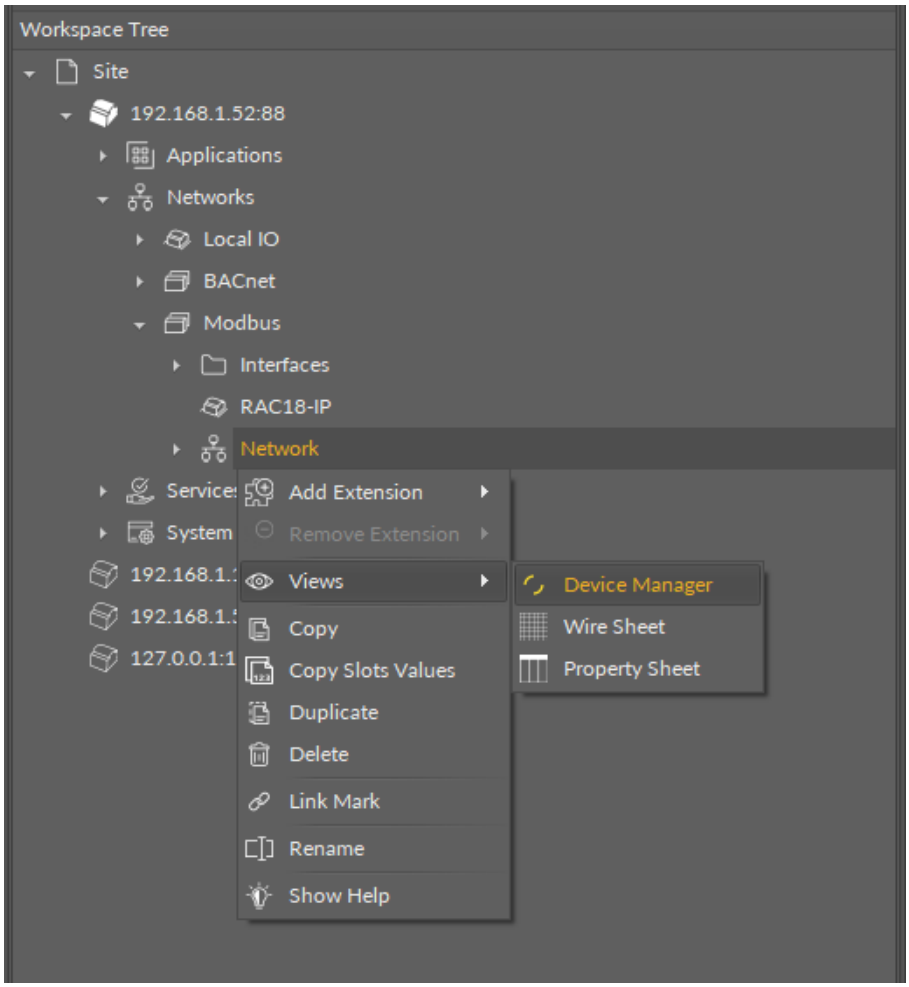


Figure 45. Opening Device Manager from the context menu

Adding Modbus Devices

The devices may be added to the Modbus network twofold: dragging and dropping the Device component to the Network component from the Modbus library (in the Device Libraries window), or using a special Add function in the Device Manager view available in the bottom right corner.

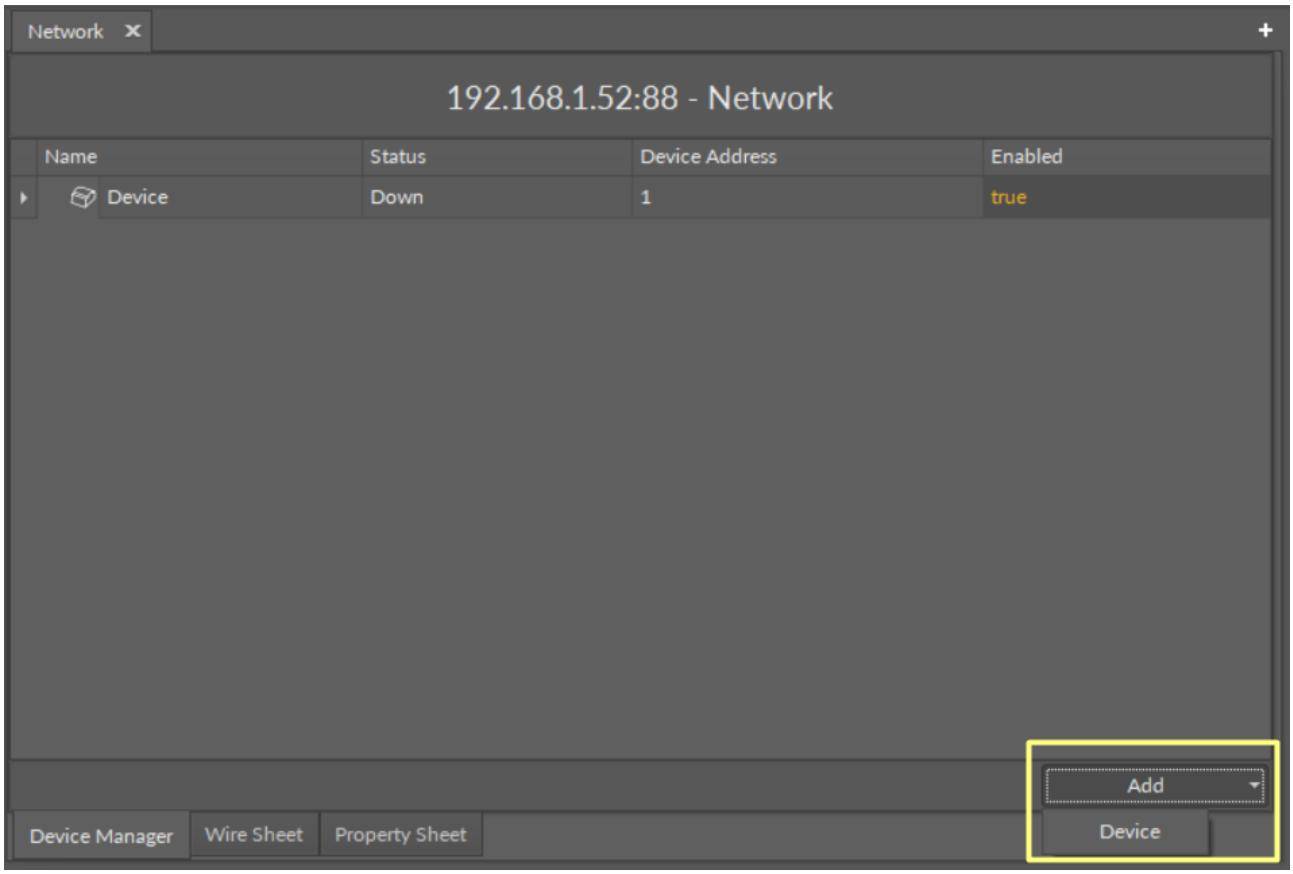


Figure 46. Adding device

Using this Add button opens the dialog window, which allows to adjust the quantity of devices to be added.

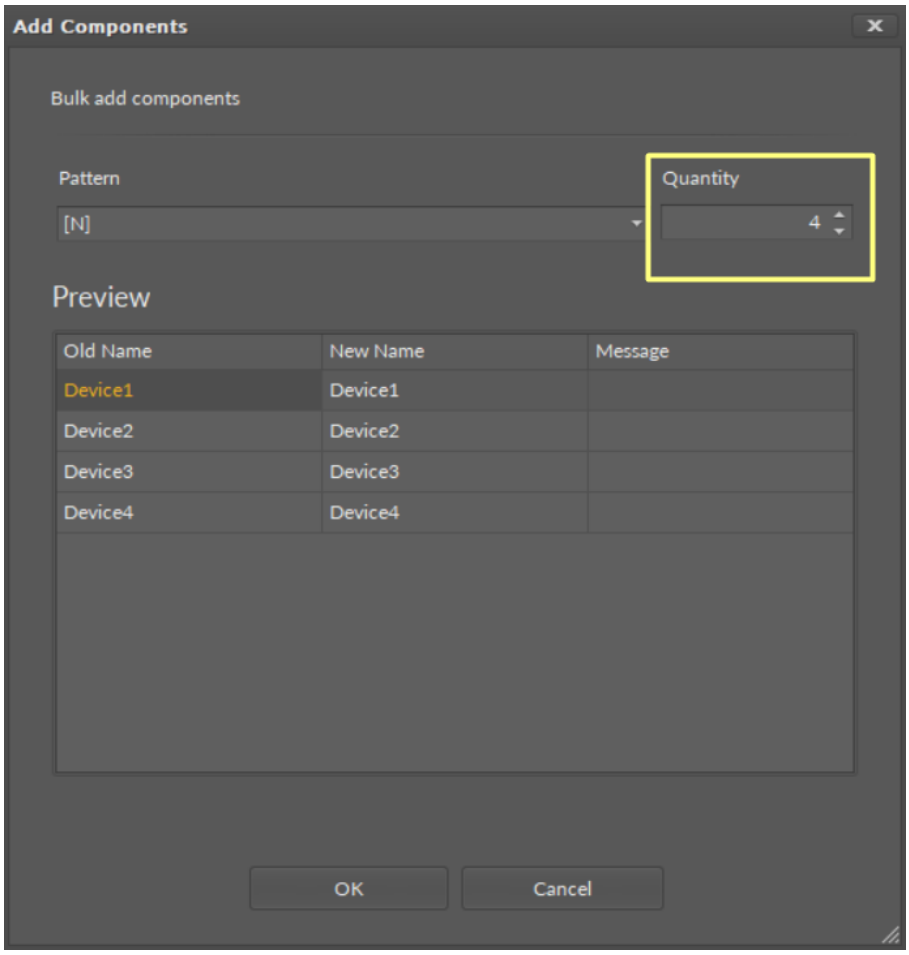


Figure 47. Add Components dialog window

Multiediting of Common Slots

The Device Manager view allows multiediting of common slots in components of the same type, for example, to enable all one-type components at once. Multiediting is available in the Object Properties window, upon selecting one-type components in the Device Manager with Ctrl or Shift keys.

6.3.3 Point Manager for Modbus

The 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, address, polling mode, and enabled or disabled state.

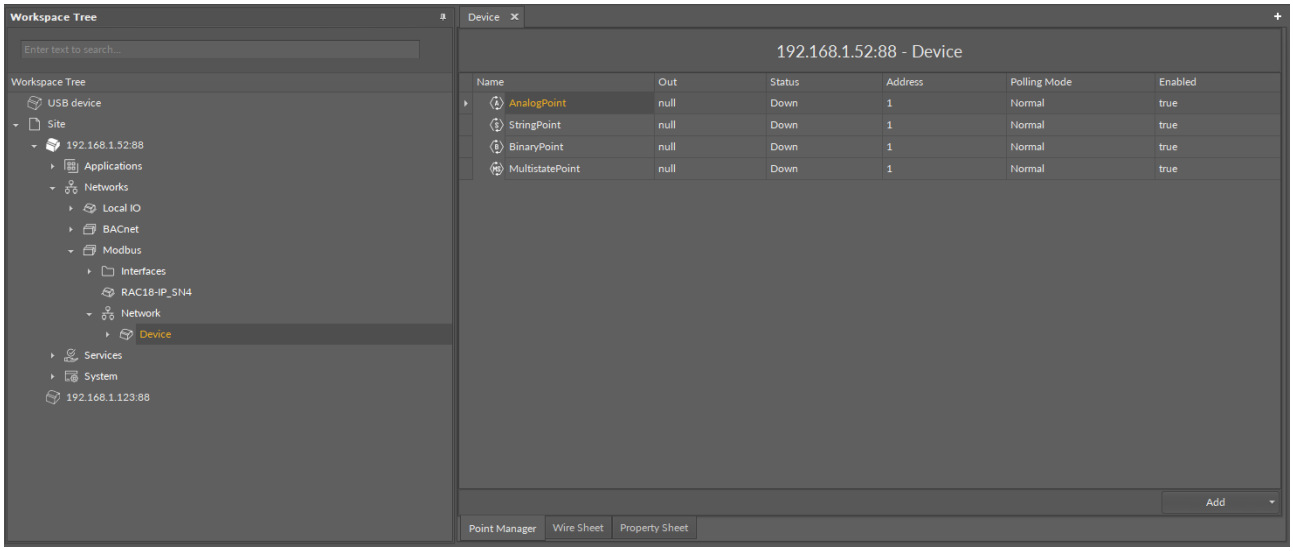


Figure 48. Point Manager for Modbus

Opening Point Manager

The 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 Workspace Tree window.

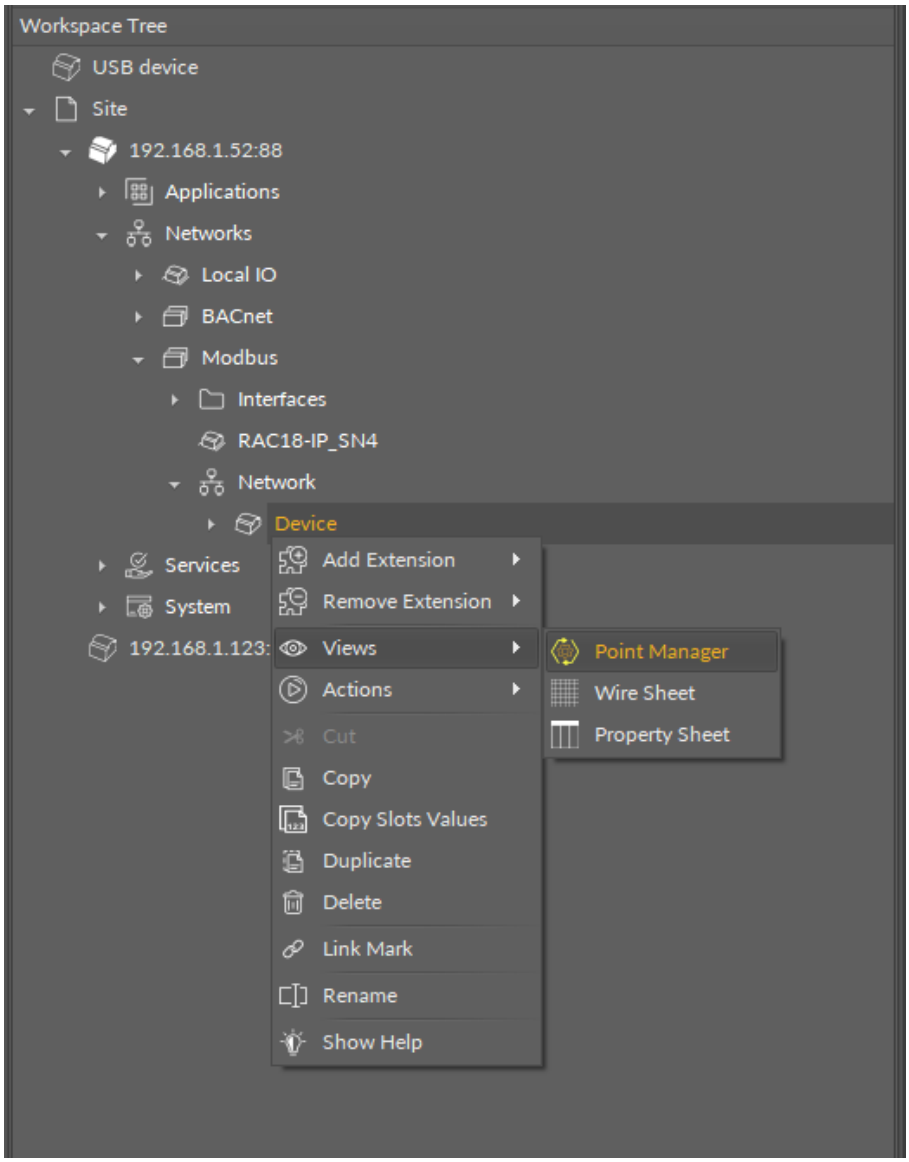
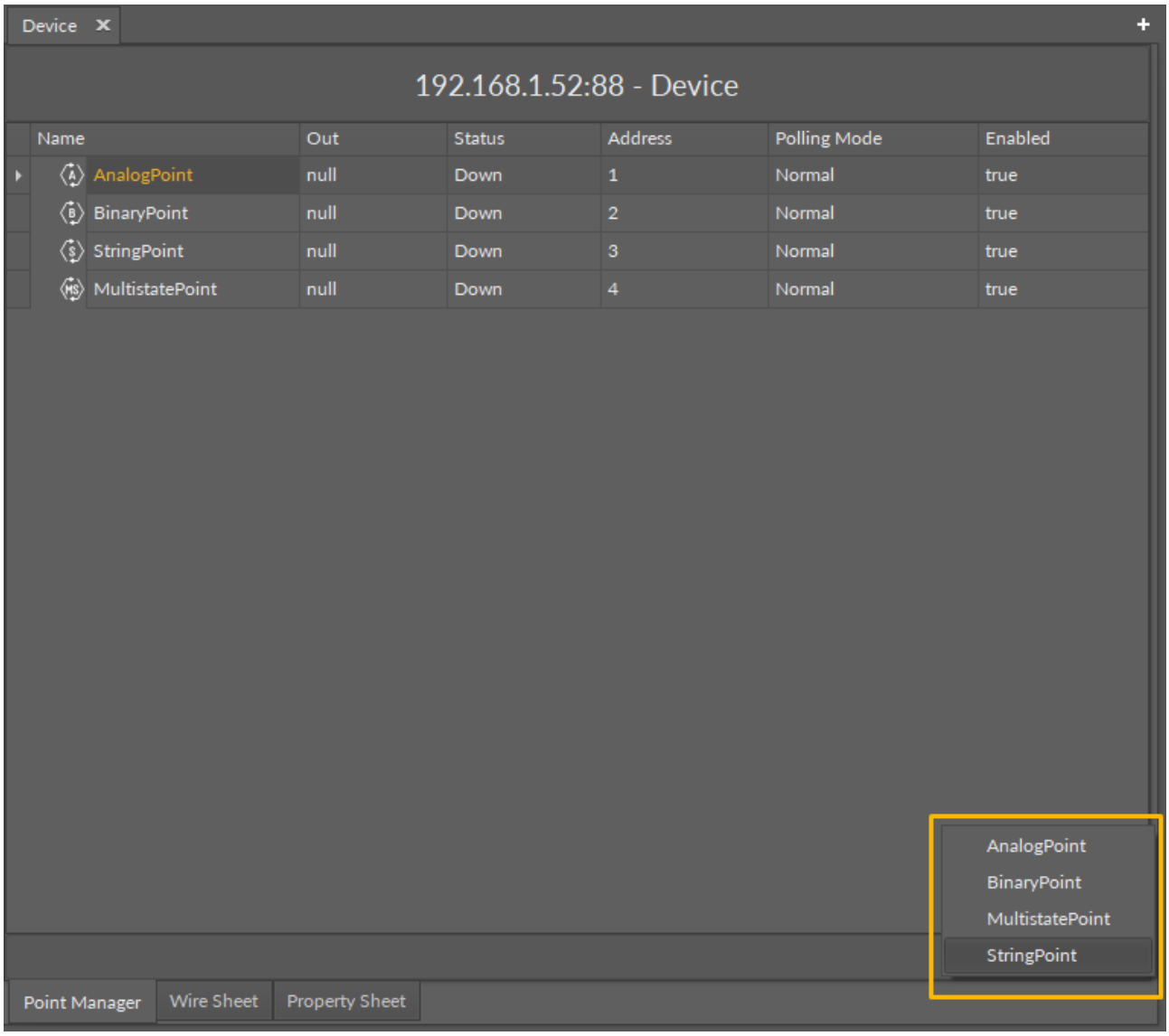


Figure 49. Opening the Point Manager from the context menu

Adding Modbus Points

The Modbus points may be added to the device twofold: dragging and dropping the Modbus points to the Device component from the Modbus library (in the Device Libraries window), or using a special Add function in the Point Manager view available in the bottom right corner. The Add function allows to add any of the Modbus points available in the Modbus library.



Using this Add button opens the dialog window, which allows to adjust the quantity of Modbus Points to be added.

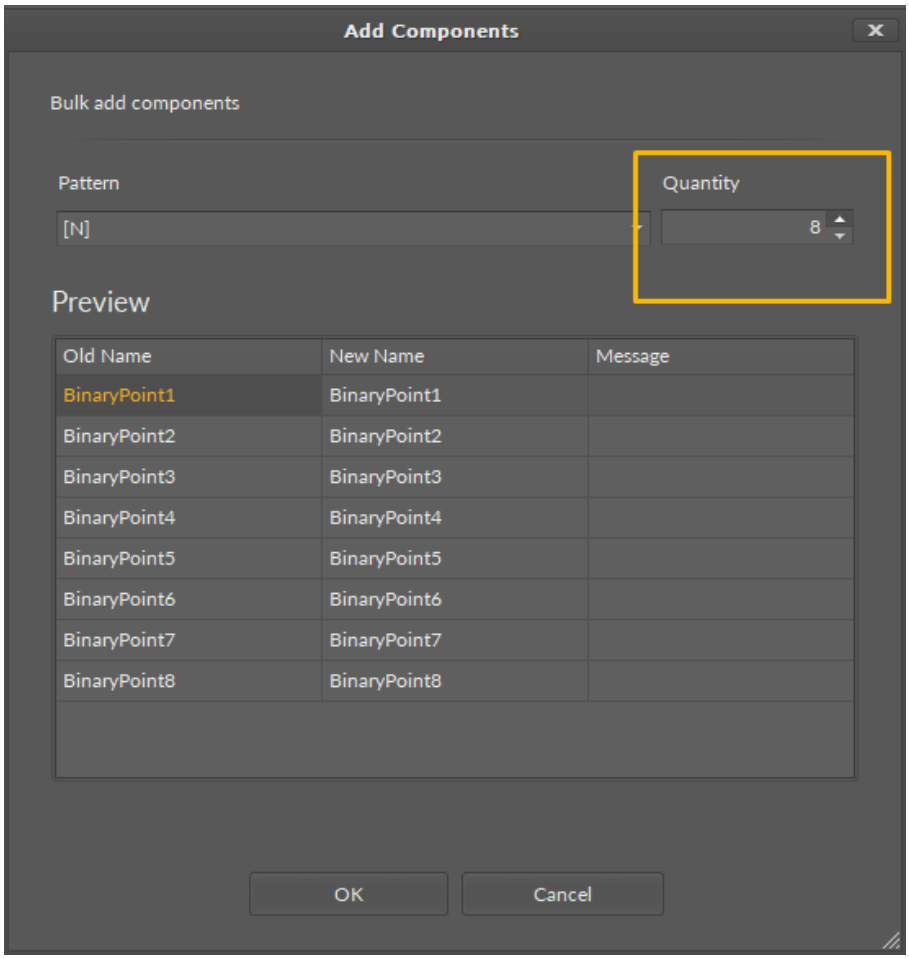


Figure 50. Adding points dialog window

Multiediting of Common Slots

The Point Manager view allows multiediting of common slots in components of the same type, for example, to enable all one-type components at once. Multiediting is available in the Object Properties window, upon selecting one-type components in the Point Manager with Ctrl or Shift keys.

6.4 Networks Libraries

The nano EDGE ENGINE Networks libraries provide sets of components to manage external communications of the device. Libraries are grouped by specific functionalities, for example, managing local inputs and outputs and implementing a BACnet or Modbus protocol. Libraries designed for the Networks container are:

- Core library,
- IO library,
- BACnet library,
- Modbus library,
- DALI library (applicable in controllers with the DALI port).

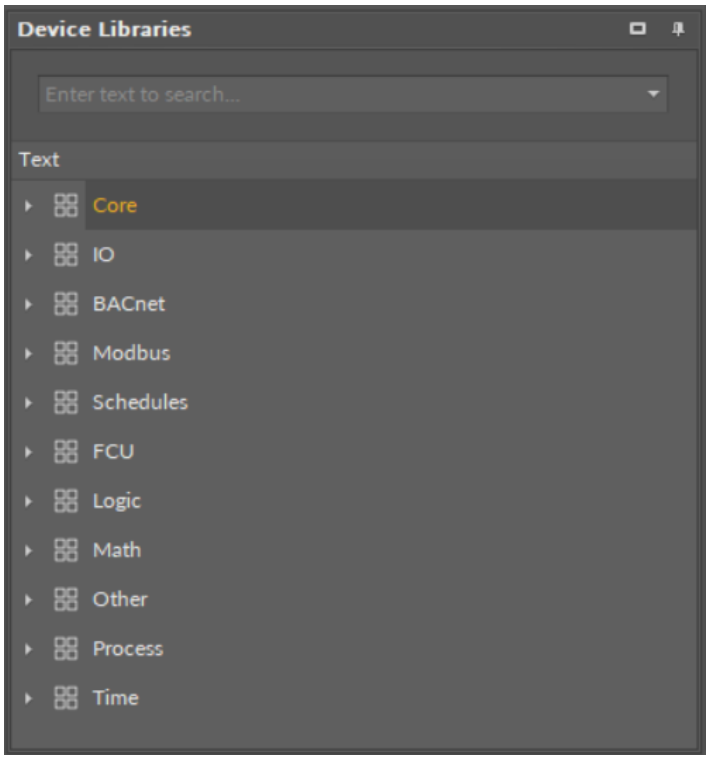


Figure 51. nano EDGE ENGINE libraries

7 Services

The Services container provides 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 (including cloud data exchange) .

Services introduce an additional logical layer to the algorithm in the application–as the algorithm makes calculations in cycles, the service may correlate it with external factors, for example, setting limiting values that invoke specific actions in the application. The basic algorithm, therefore, retrieves some additional information from such service, and enhances its own functionality, for example, taking into account the weather while executing the application for watering the garden.

Unlike Networks, which manage strictly automation data essential for the operation of the device (management of physical inputs and outputs, communication protocols, etc.), Services enhance the algorithm's options, making it more resilient and adjustable for multiple application purposes.

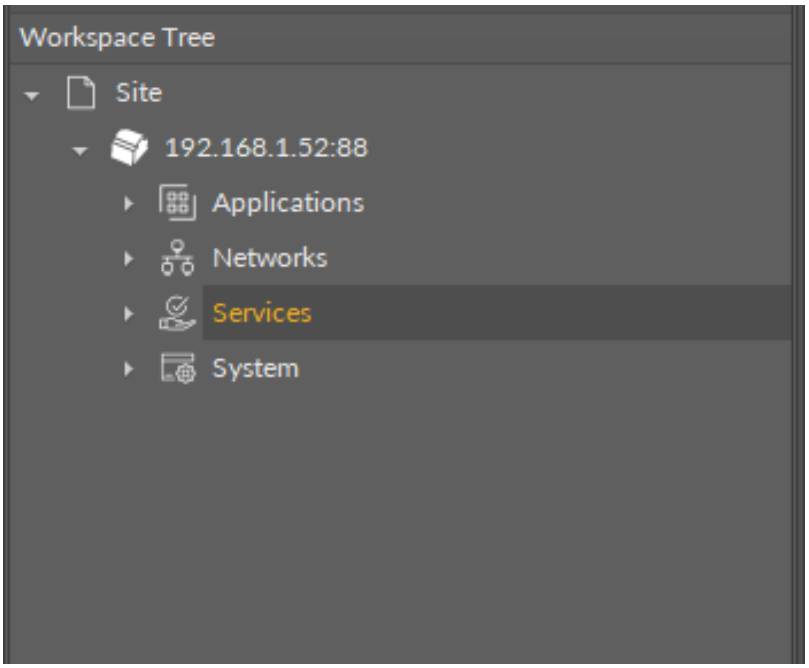


Figure 52. Services container

Available services:

- Configuration Data
- Trends
- Tagging
- Web service
- Haystack

7.1 Configuration Data

Applicable to OS V1.10

The Configuration Data is a service designed to save values of specific slots of Data Points and AnalogConstant and BinaryConstant components for the purpose of restoring them if changed or lost. The service is executed by adding the Configuration Data extension to components:

- AnalogDataPoint,
- BinaryDataPoint,
- MultistateDataPoint,
- AnalogConstant,
- BinaryConstant.

The service allows to save, load, or clear data from the following slots of components:

- Data Points' slots: In16, In1-In15 (if the Analog/Binary/MultistatePriorities extension has been added),
- AnalogConstant/BinaryConstant's slot: Out.

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.

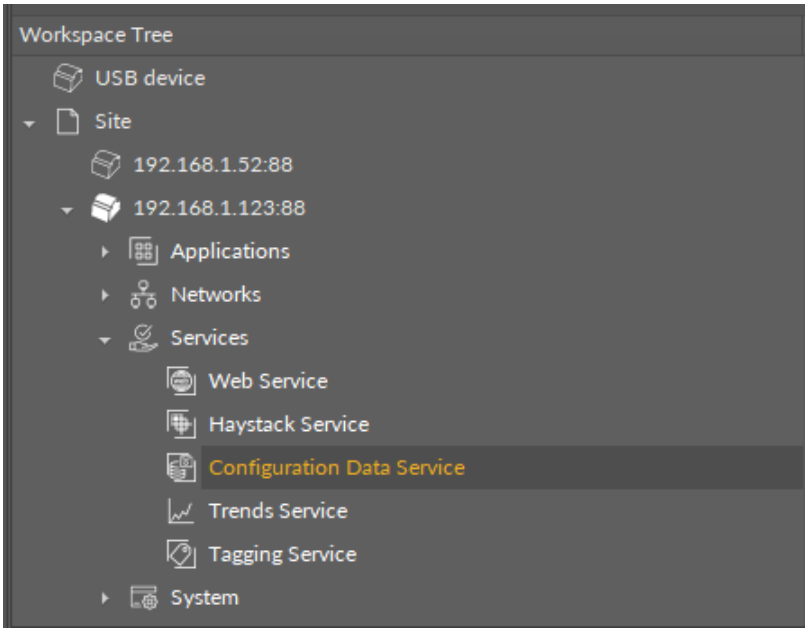


Figure 53. The Configuration Data Service in the workspace tree

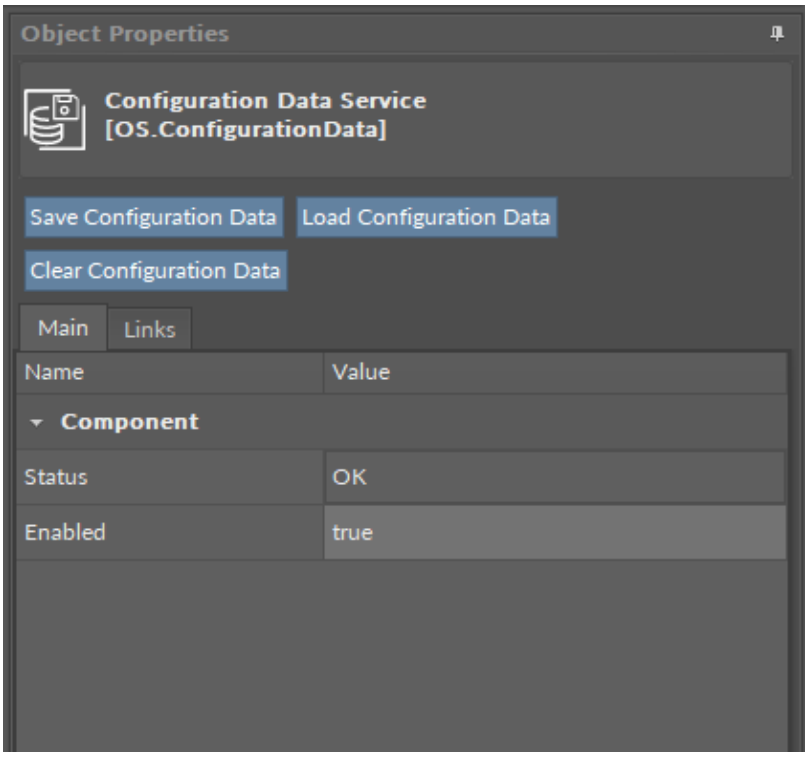


Figure 54. The Configuration Data service slots

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.

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**
> Saved/loaded values:
> - Data Points' slots: In16, In1-In15 (if the Analog/Binary/MultistatePriorities extension has been added),
> - AnalogConstant/BinaryConstant's slot: Out.
>
> The values can be loaded only to components, which had the ConfigurationData extension added at the point of saving values. Values will not be loaded if a link has been connected to the In slot of a saved configuration data.

- **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.

7.1.1 Configuration Data Service View

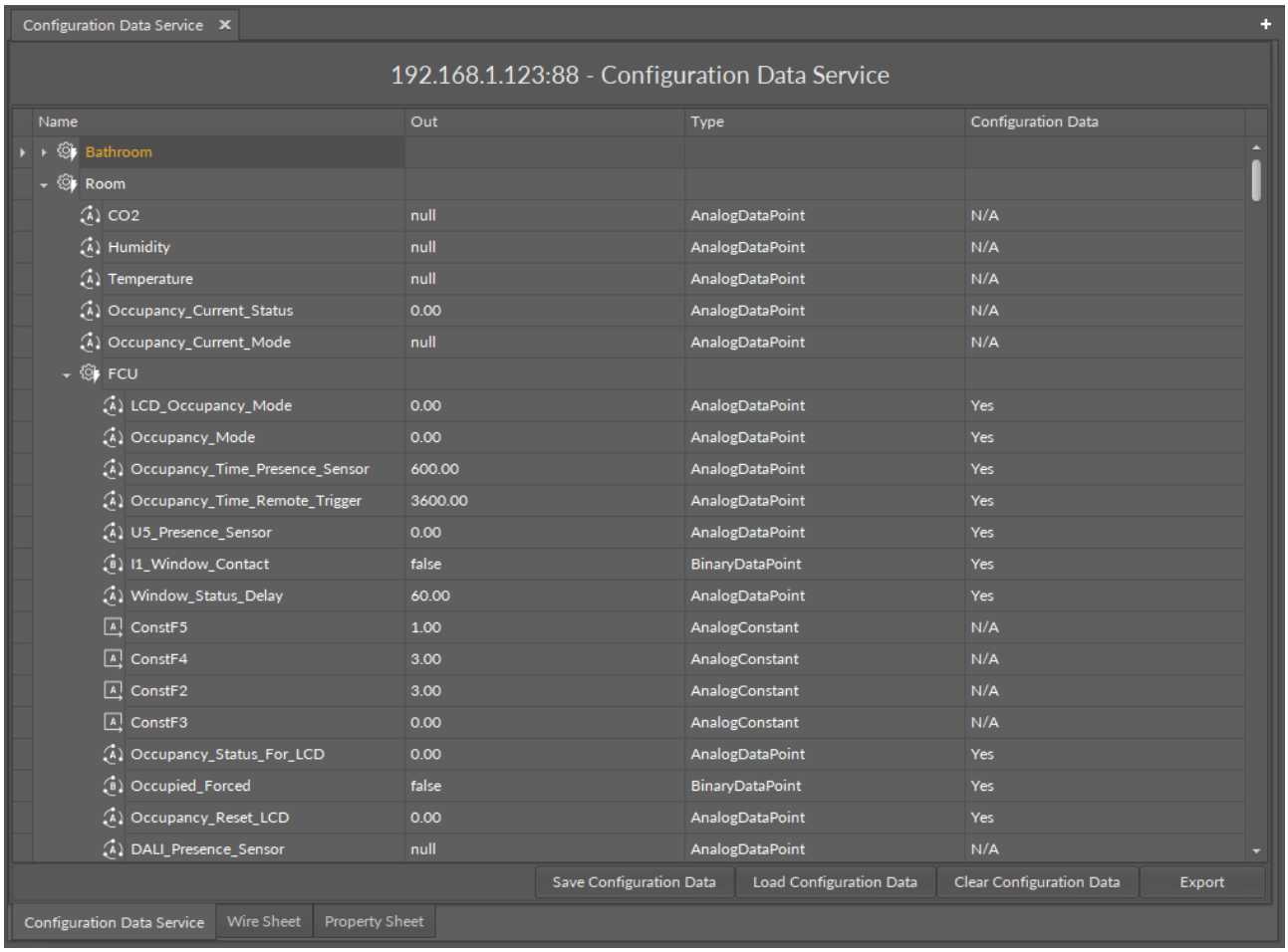


Figure 55. The Configuration Data service view

The Configuration Data service view is a simple table view showing which Data Points or AnalogConstant/ BinaryConstant components have the Configuration Data extension added, along with their type and Out slots.

The components marked N/A in the view are components added to applications but without the Configuration Data extensions, as the service collects data only from components with added extension. The view shows components from all applications executed in the Applications container. Additionally, it allows to export the gathered data with the Export option in iC Tool.

7.1.2 Configuration Data Extension

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

7.2 Trends

Applicable to OS V1.8-1.10

The Trends service is designed to save values from Data Points in a database. It allows to manage historical data coming from Data Points, enable or disable saving data from a specific Data Point, and depict data in a graphical or listed form.

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

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

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

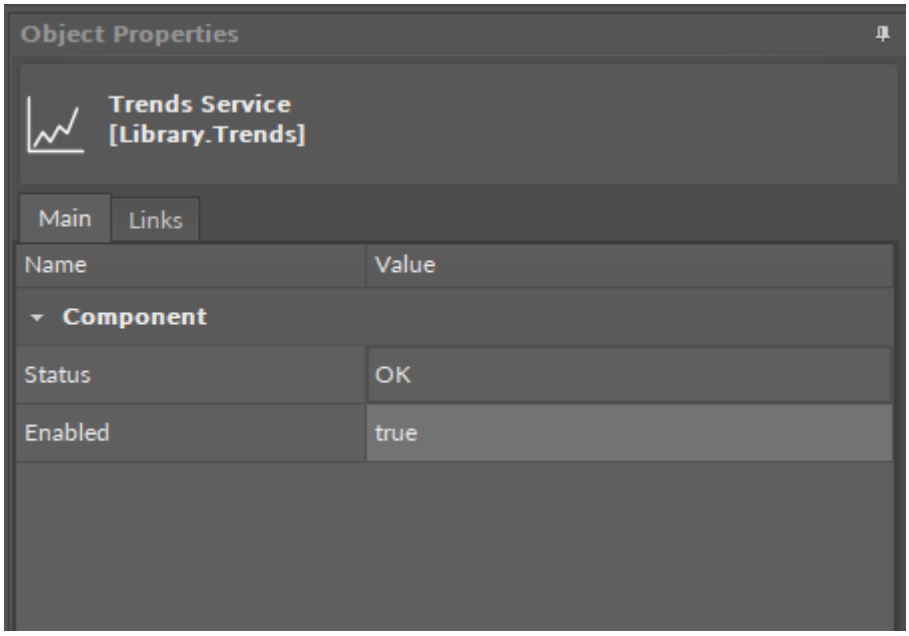


Figure 56. Trends service

The Trends service has the following slots:

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

Dedicated views
Views dedicated to the Trends service are available in the iC Tool: iC Tool - Trends Viewer and iC Tool - Trends Manager, and nE2 Link module: nE2 Link - Trends.

7.3 Tagging

Applicable to OS V1.9-1.10

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

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

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

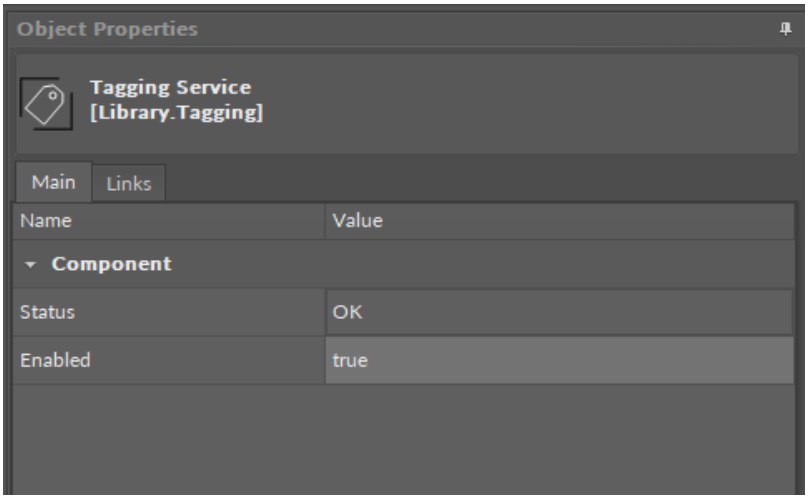


Figure 57. Tagging service

The Tagging 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.

7.3.1 Tag Dictionaries

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

There are two tag dictionaries available: Haystack and Webserver.

Haystack

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

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

Webserver

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

Tag	Auto-tag	Description
web:application	Yes (ADP, BDP, MDP)	Shows which application the Data Point belongs to (for the nanoWebUI order)
web:expose	Yes (ADP, BDP, MDP)	Defines the Data Point's visibility on the nanoWebUI (editable)
web:widgetType	Yes (ADP, BDP, MDP)	Defines the widget type to display on the nanoWebUI (editable)
web:order	Yes (ADP, BDP, MDP)	Defines the order of widgets displayed on the nanoWebUI
web:decimalPoint	Yes (ADP)	Allows to configure the display of decimal point on the nanoWebUI (editable)
web:step	Yes (ADP)	Allows to configure the step value change for the nanoWebUI (editable)
web:homepage	No (manually added)	Sets the nanoWebUI as a web browser's homepage

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

7.3.2 Using Tags

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

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

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

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

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

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

Auto-tagging

Auto-tagging is a feature required for a correct and efficient display of the nanoWebUI and is available for Data Points and the Equipment component. The list of auto-tags and exact details of applying auto-tags to Data Points and Equipment are available here:

- AnalogDataPoint
- BinaryDataPoint
- MultistateDataPoint
- Equipment

Manually Added Tags

Apart from the auto-tagging feature, it is possible to manually select tags from the available tag dictionaries.
Tag dictionaries are libraries containing lists of tags, which can be applied to Data Points and the Equipment component. Tag dictionaries are searchable and contain labels, which allow for the clear and efficient structure management, data modelling, and faster data processing by local control systems.
There are two tag dictionaries available: Haystack and Webserver (for details, see the above Tag Dictionaries section).
The exact details of manually applying tags to Data Points and Equipment are available here:

- AnalogDataPoint
- BinaryDataPoint
- MultistateDataPoint
- Equipment

Dedicated views
Views dedicated to the Tagging service are available in the iC Tool: iC Tool - Tag Manager and iC Tool - Tags Tab, and nE2 Link module: nE2 Link - Tagging.

7.4 Web service

Applicable to OS V1.9-1.10

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

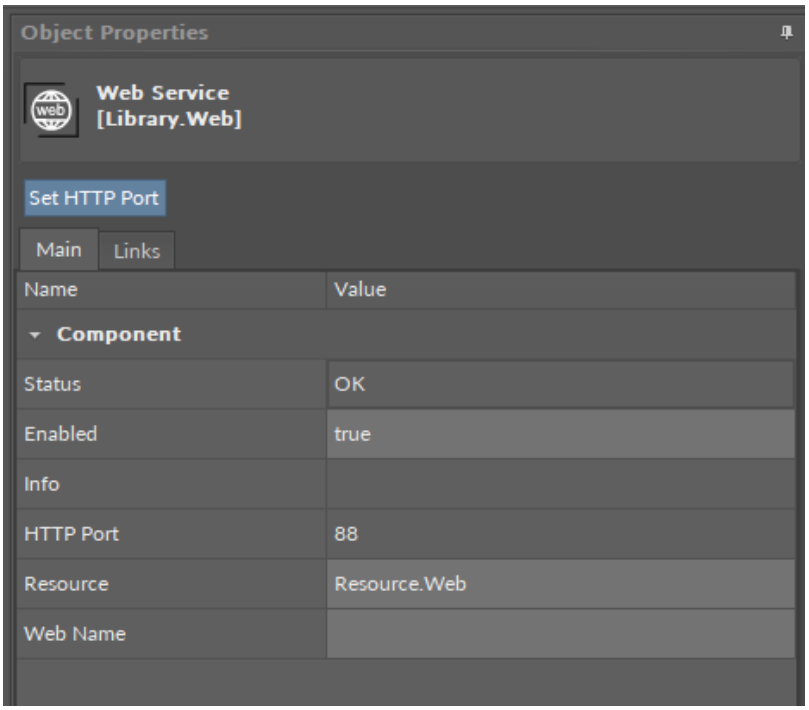


Figure 58. Web service

The Web service has the following slots:

- Status: indicates the current status of the component. If the component works properly, its status is OK; however, it changes accordingly when values in other slots are adjusted.
    - Available information: disabled (the Enabled slot is set to false), OK;
- Enabled: change of the slot's value enables or disables the component;
- Info: informs about a non-OK status of the component;
    - Available information:
        - Couldn't start file server (status Error): occurs when the file server cannot not be started due to a memory shortage or other exception,
        - No resource manager (status Error): occurs when the resource file manager cannot be opened due to a memory shortage or other exception,
        - No resource (status Error): occurs when the resource file is not loaded,
        - The port change will take effect after the device restart (status OK): occurs when the HTTP port number has been changed with the Set HTTP port action;

Numerical value	Displayed information
0	No information displayed in the Info slot

Numerical value	Displayed information
1	Couldn't start file server
2	No resource manager
3	No resource
4	The port change will take effect after the device restart

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

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

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

The Web service has the following action:

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

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

Warning!

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

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

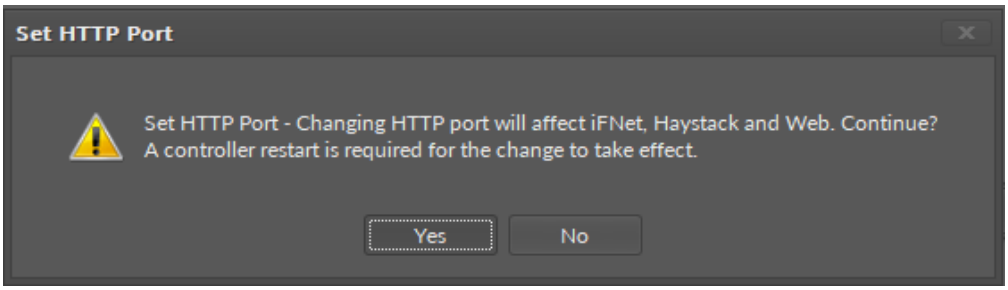


Figure 59. HTTP port change notification

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

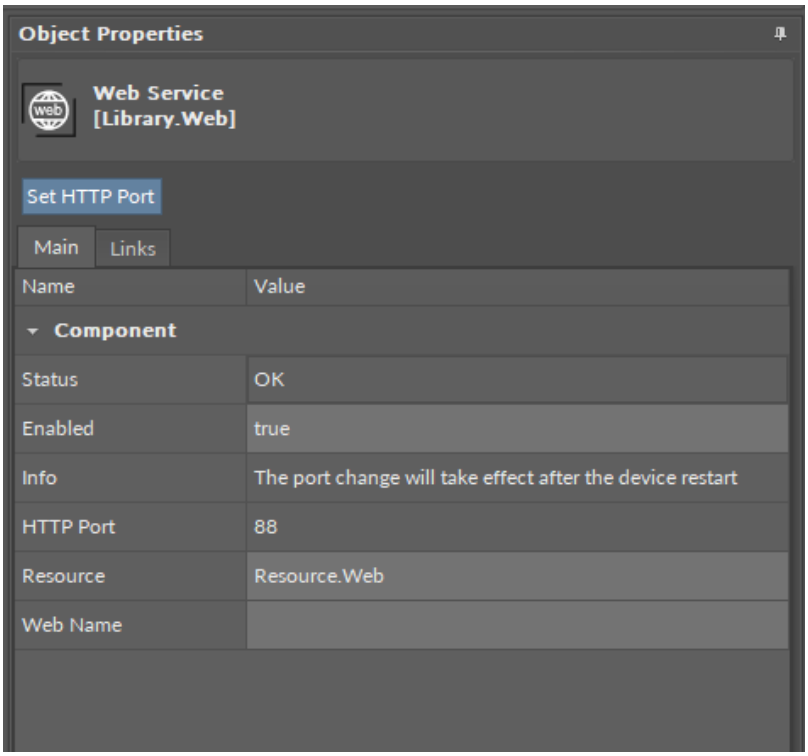


Figure 60. Info slot notification after HTTP port change

Dedicated views

Views dedicated to the Web service are available in the iC Tool: iC Tool - Web Tags, and nE2 Link module: nE2 Link - Web service.

7.5 Haystack

Applicable to OS V1.9-1.10

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

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

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

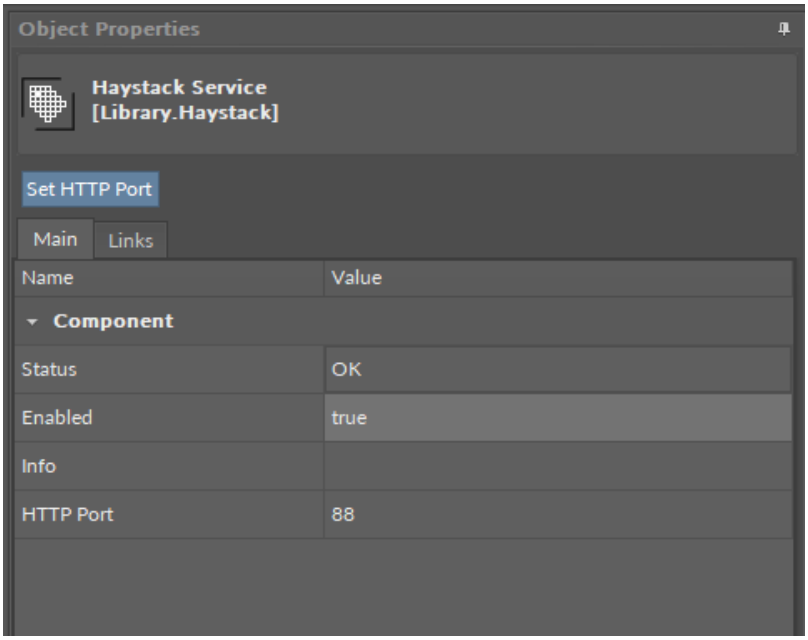


Figure 61. Haystack service

The Haystack service has the following slots:

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

The Haystack service has the following action:

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

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

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

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

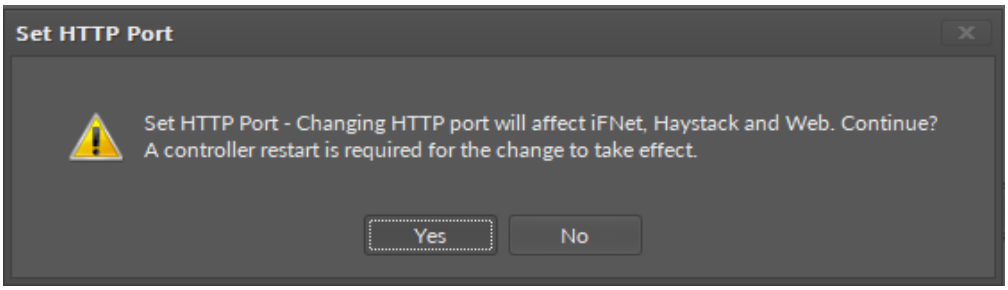


Figure 62. HTTP port change notification

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

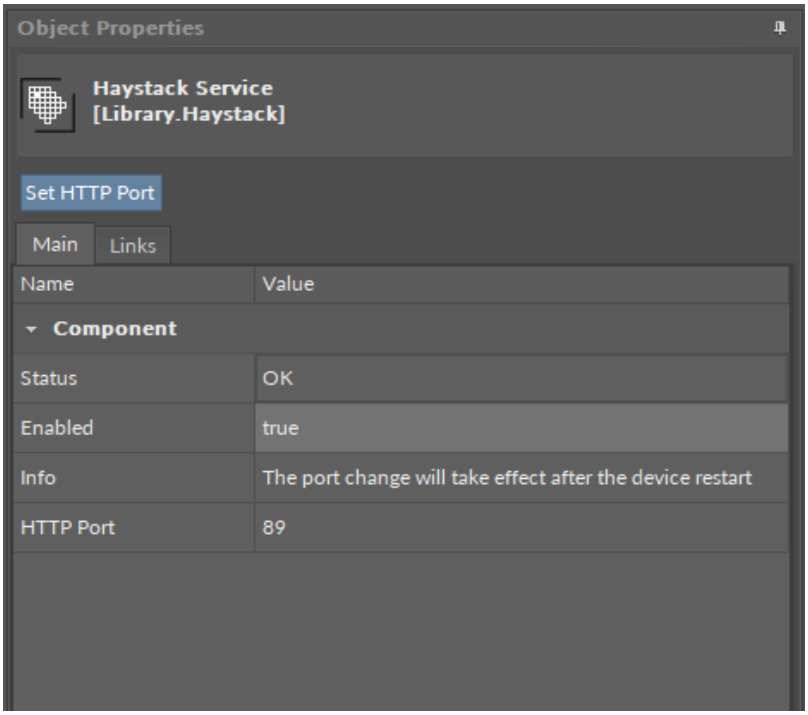


Figure 63. Info slot notification after HTTP port change

7.5.1 HTTP API

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

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

The nano EDGE ENGINE implementation supports the following version:

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

Supported Requests

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

The auto-refresh mechanism is set to 5 minutes.

8 System

The System container is a configuration center of a device; it includes information about the operating system and settings. It provides a hardware characteristic of the device, informing about the OS version, CPU performance, IP address, port number, or number of inputs and outputs.

The System container is a starting point to work with the device–it is a place to configure the device and connect it to the network. One of the most useful features of the System container is a possibility to set a unique IP address of the device.

In further operations it provides tools for monitoring and troubleshooting such as the Logs component, which provides an adjustable register of events taking place during the operation of the device. The System container includes a tree of components providing categorized information about the device and its settings.

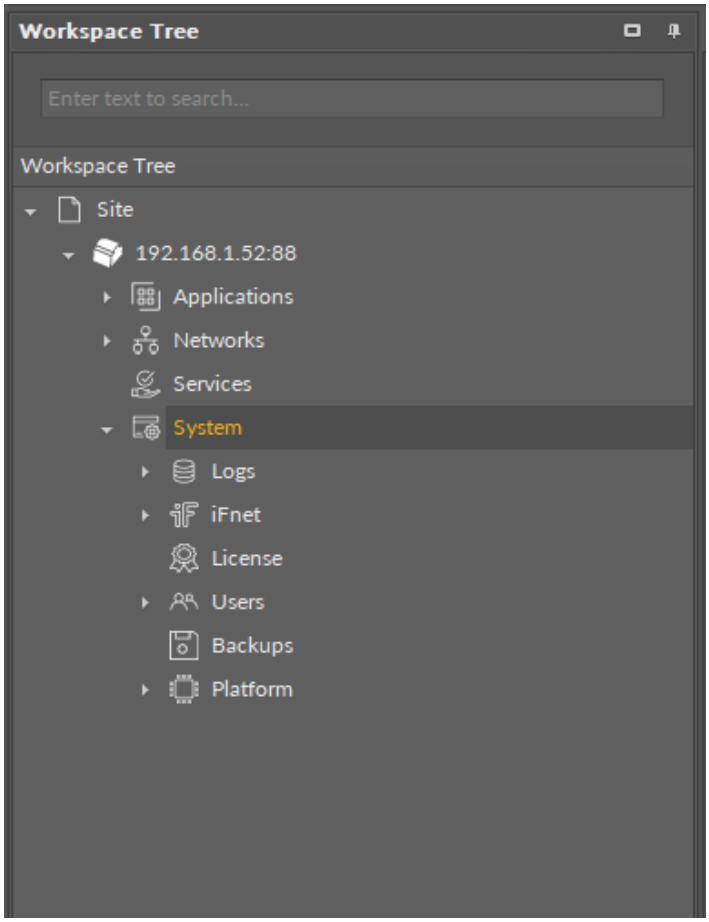


Figure 64. System container

Note
To learn about the components in the System container, please refer to the nano EDGE ENGINE Programming User Manual.

9 Troubleshooting

9.1 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.

9.2 Operation in Emergency Mode

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 Workspace Tree;

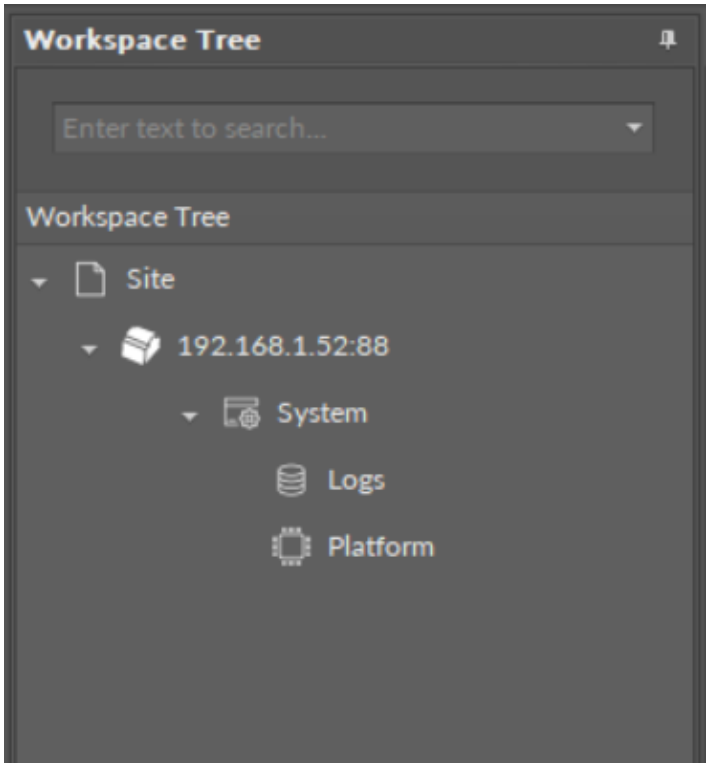


Figure 65. The device's Workspace Tree in the emergency mode

- 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).

When connecting to a device, which is in the emergency mode, a notification is displayed:

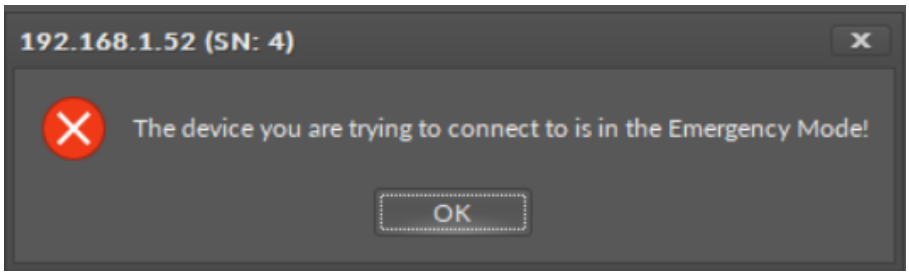


Figure 66. Emergency mode notification notice

9.3 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 (restore in the iC Tool): remove files from the SD card (if available and formatted) excluding only files with IP, port, and credentials (libraries must be also removed);
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.