iSMA**CONTROLLI**

RAC18-IP
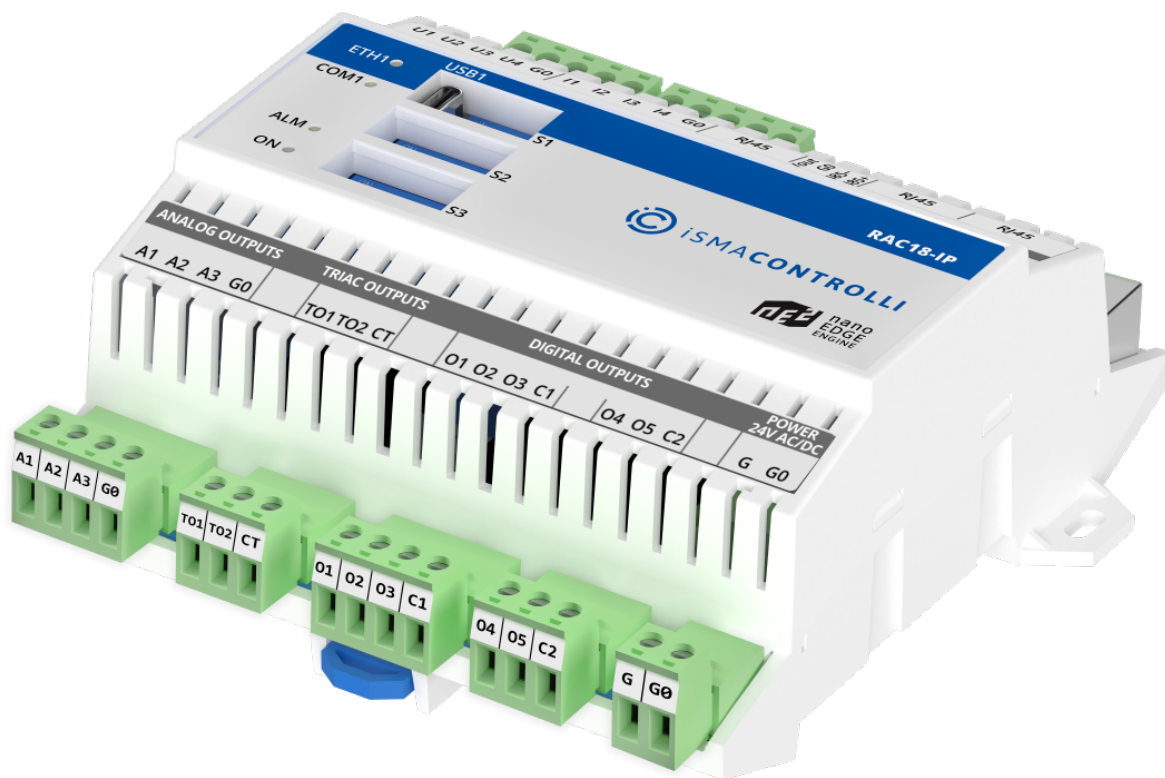
User Manual

Software

Table of Contents

1 Introduction

The RAC18-IP is a multifunctional device providing comfort in a single room or a zone in the building. 18 onboard I/O's with Ethernet and serial port make the controller a versatile system for comfort management, light and blind control, and integration and monitoring. The RAC18-IP controller is driven by the nano EDGE ENGINE–the cutting-edge software that allows for constructing multiple applications and provides a set of libraries and components tailored to create cycle-driven user applications. The device has an automatic BACnet IP and Modbus TCP/IP exposure and allows for real-time programming. Maintenance and programming of the device are possible from anywhere in real time with a dedicated, free-of-charge tool. The RAC18-IP ensures an easy replacement process with a system and applications on a removable SD card.

1.1 Revision History

Date	Rev.	Description
Oct 2023	1.4.0	Update in compliance with the nano EDGE ENGINE 1.4.1 release: • added functionality of the Modbus RTU server; • added functionality of the BACnet MS/TP client/server. (To learn about **nano EDGE ENGINE** structure, libraries, and components please refer to the nano EDGE ENGINE Programming User Manual.)
6 Dec 2022	1.2.0	Update in compliance with the **nano EDGE ENGINE** 1.2 release: • added functionality of the Modbus TCP/IP client; • added functionality of the Modbus gateway; • added functionality of the DNS server; • added functionality of schedules; • added functionality of the NTP time synchronization. (To learn about **nano EDGE ENGINE** structure, libraries, and components please refer to the nano EDGE ENGINE Programming User Manual.)
31 Aug 2022	1.1.0	Update in compliance with the **nano EDGE ENGINE** 1.1 release: • added functionality of the Modbus TCP/IP server; • added functionality of the DHCP. (To learn about **nano EDGE ENGINE** structure, libraries, and components please refer to the nano EDGE ENGINE Programming User Manual.)
21 Apr 2022	1.0.0	First edition

Table 1. Revision history

2 Licensing

The license for the new generation of iSMA CONTROLLI devices 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.

3 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.

3.1 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.
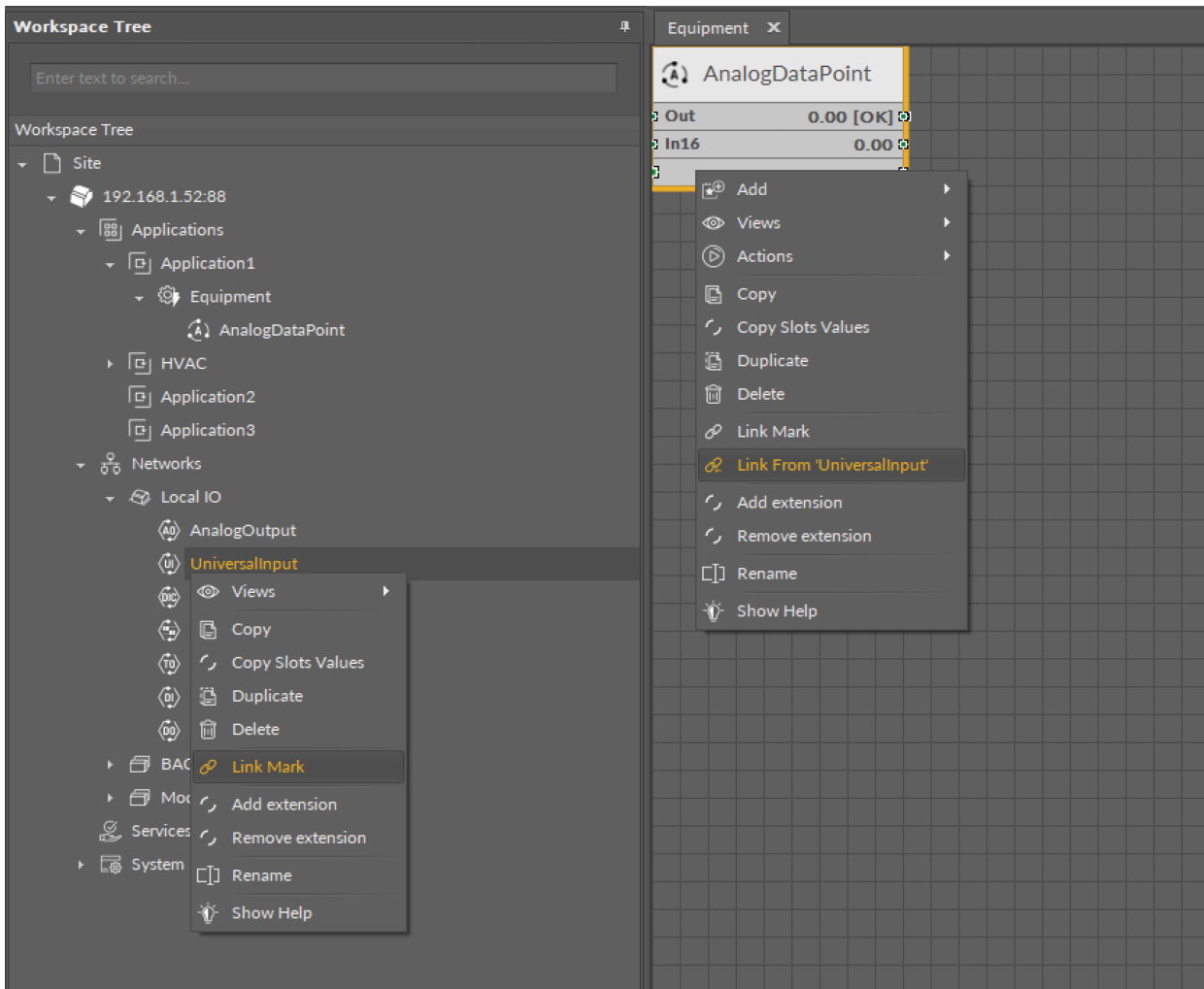
Figure 1. Link Mark-Link From options

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.

3.1.1 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.
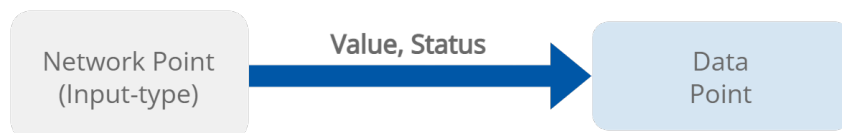

Figure 2. Data Point and the input-type network point links

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

3.1.2 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.
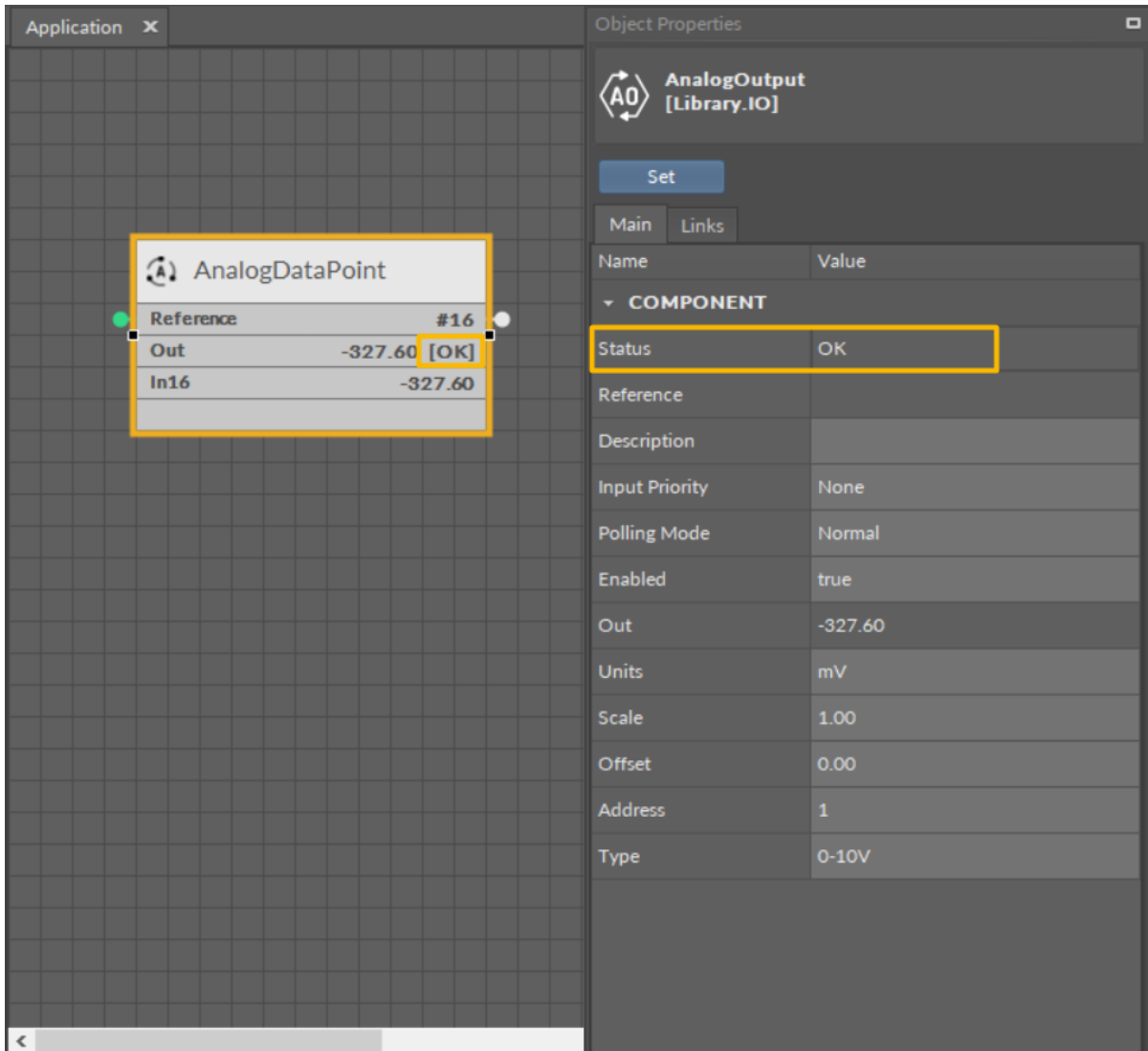

Figure 3. Output-type network point (status OK) referenced to the Data Point
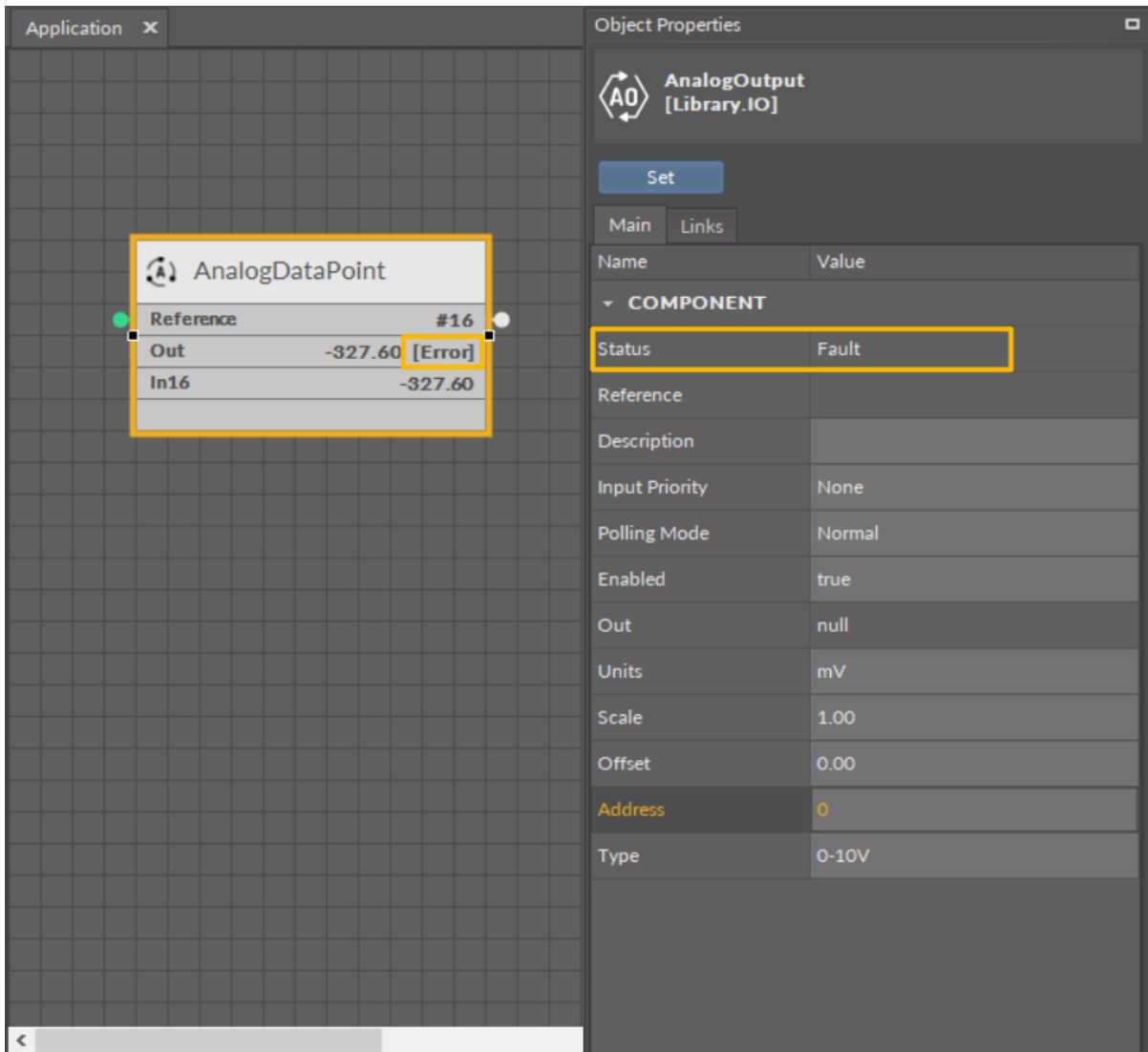
Figure 4. Output-type network point in the Fault status returning an Error status to the referenced Data Point

- 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.
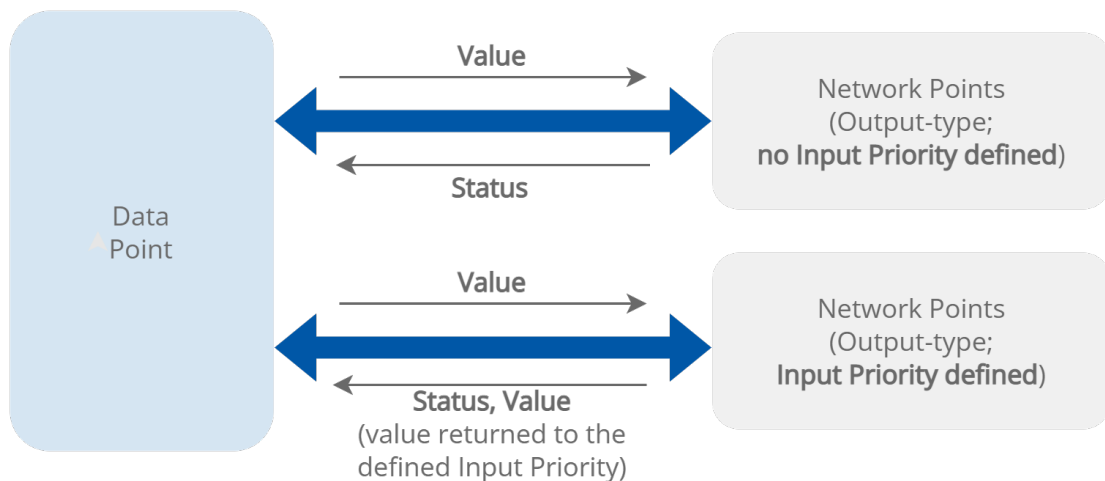

Figure 5. The Reference links between the Data Point and output-type network points
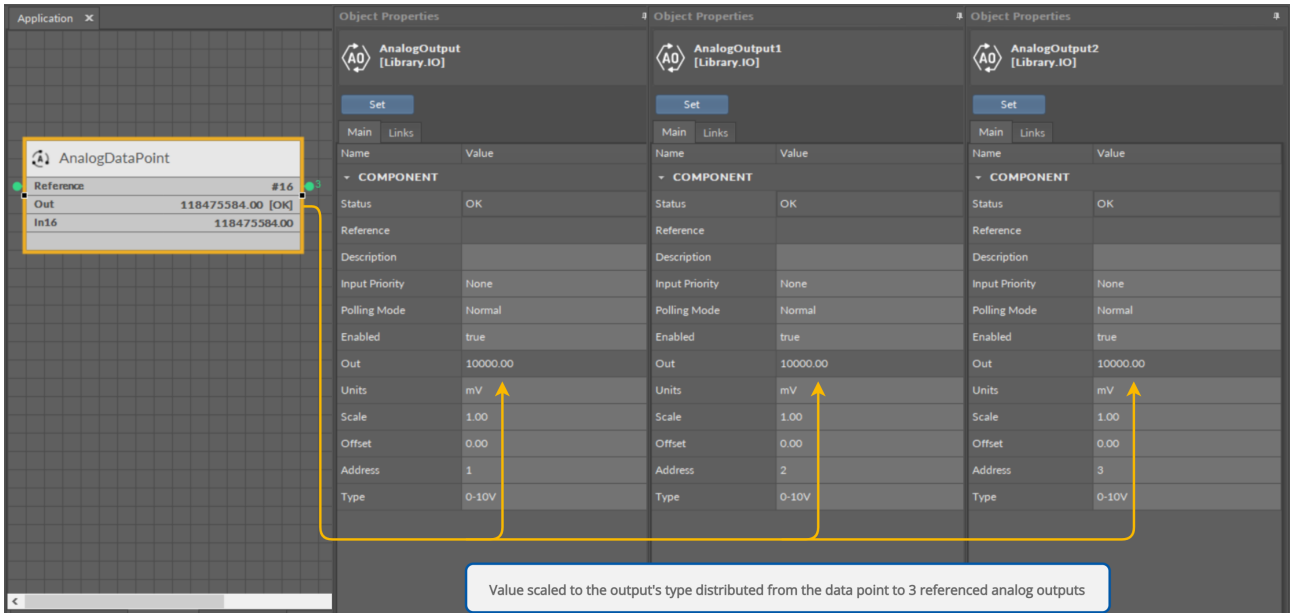
Figure 6. Values distributed from the Data Point to the referenced output-type network points
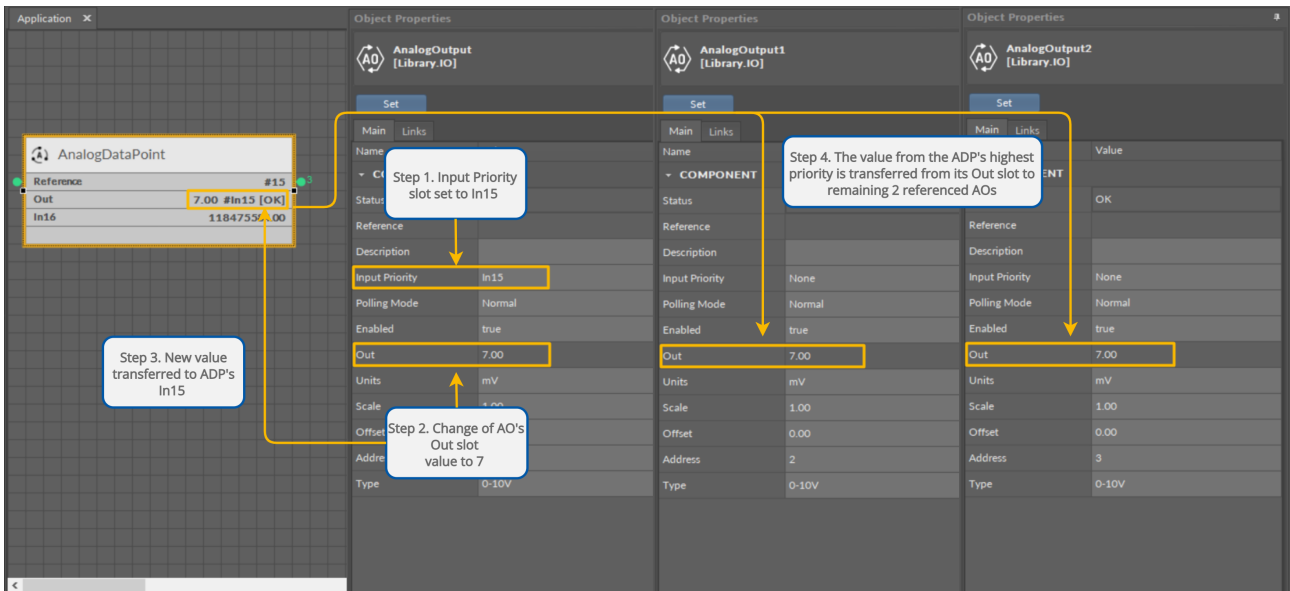

Figure 7. Value sent back to the specified input priority in the Data Point and distributed to remaining referenced output-type network points

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:
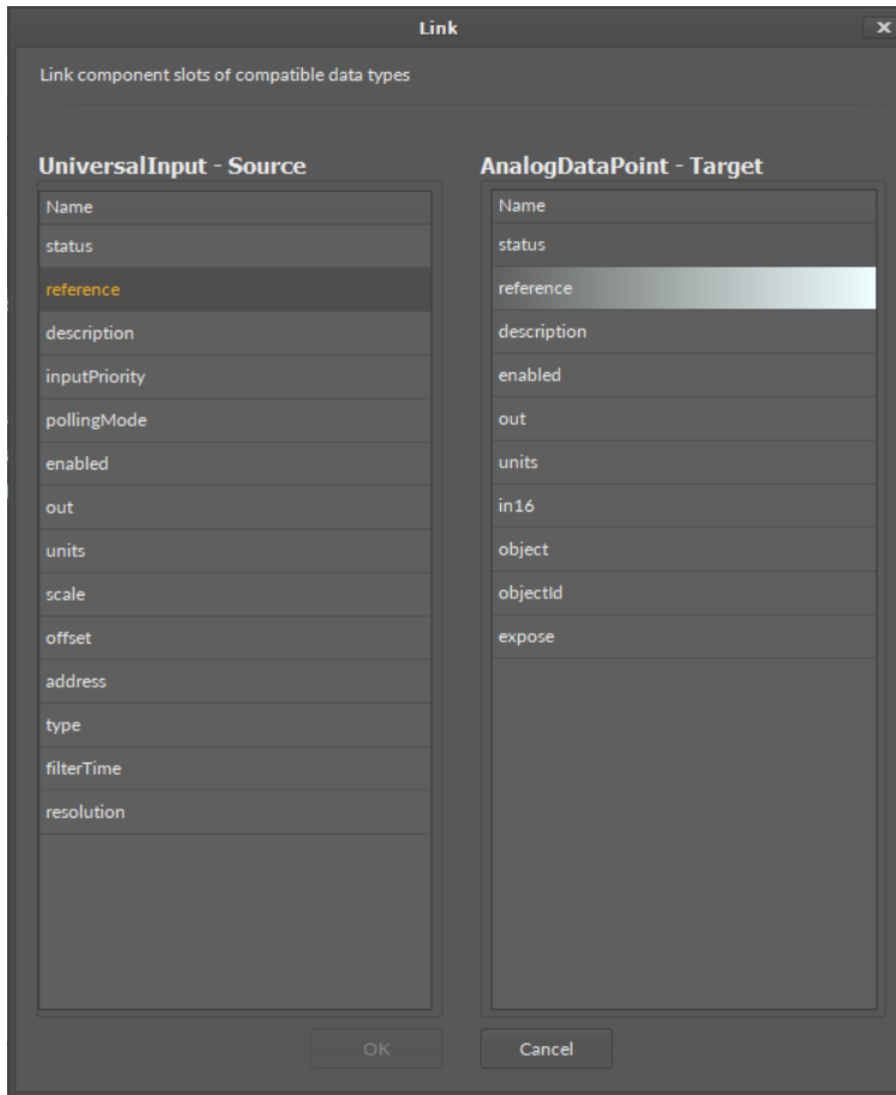
Figure 8. The linking dialog window

☑ 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:
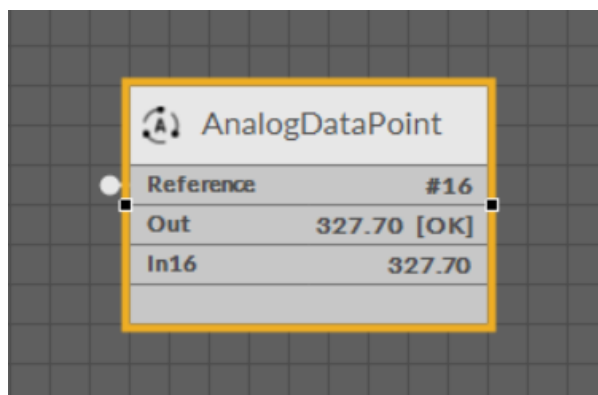

Figure 9. Analog Data Point linked from the Universal Input

☑ The details of the Reference links are always visible in the Links tab in the Object Properties window of the linked element:
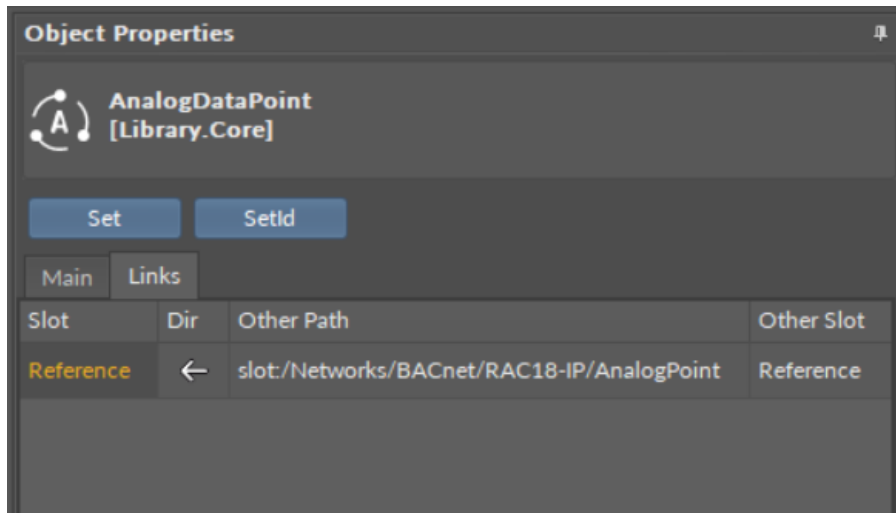
Figure 10. The Reference link path

4 Emergency Mode

The system and application(s) of the RAC18-IP controller are stored on an SD card. If the SD card is not detected in the device or the device detects frequent reboots (at least 5 times in 6 minutes), which prevent correct operation, the device enters an emergency mode.

What Causes the Emergency Mode?

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

4.1 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;
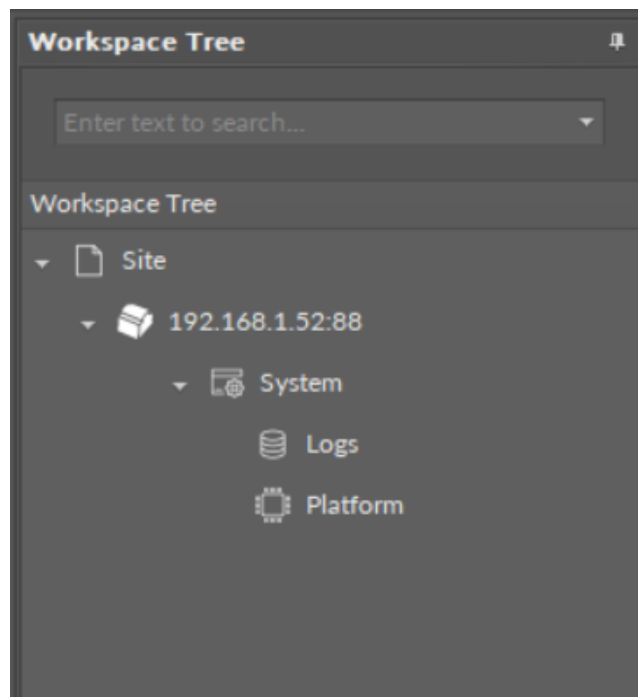

Figure 11. The device's Workspace Tree in the emergency mode

- 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:
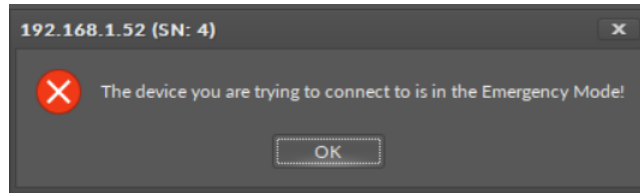
Figure 12. The emergency mode notification notice

4.2 Possible Actions

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

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

5 Applications Container

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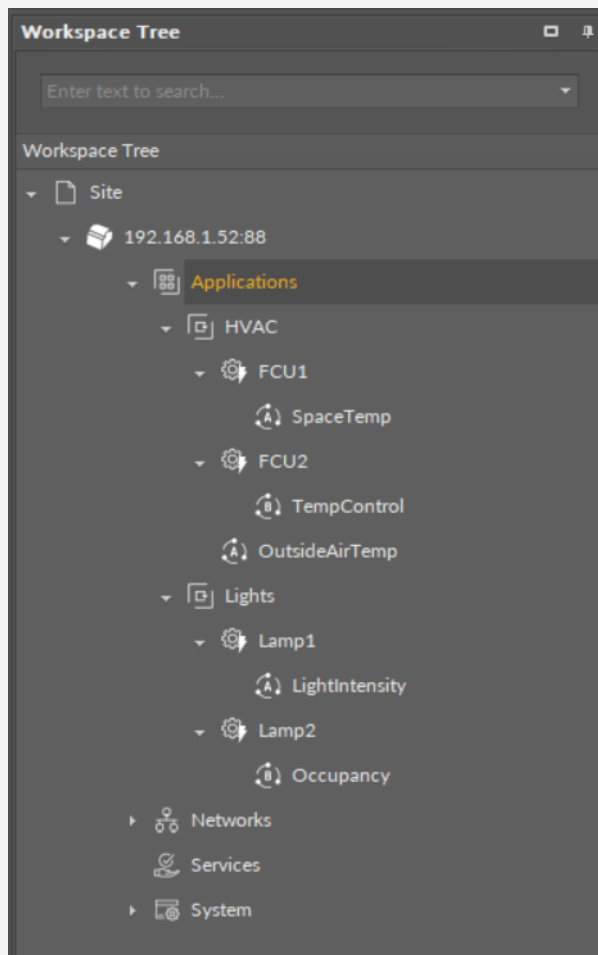
The user may create as many applications as needed—as long as the overall number of Data Points used within user applications does not exceed the device license.

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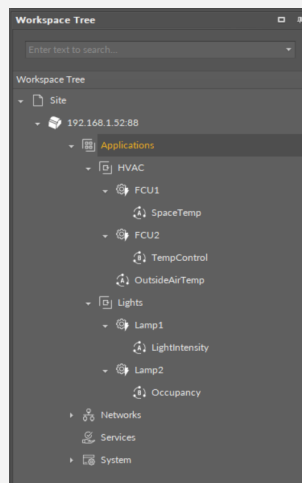
Good Practice Tip

In future developments, the RAC18-IP controller will have a Haystack functionality implemented in its firmware, which calls for some good practice mechanisms that can be introduced now, with the current functionalities.

The Haystack functionality will offer multiple advantages of using tags–it will help identify parts of equipment controlled by the application and used Data Points, filter data by equipment, sensor, or value, any many more.

In order to facilitate a future use of the full Haystack functionality, the Equipment component has been included in the Core library. It is therefore recommended 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

Such structure will be fully recognized in the Haystack functionality and will require minimum effort to use its full potential once updated.

Difference Between Equipment and Folder Components

The Equipment component with Haystack functionality will allow to identify the types and other characteristics of controlled devices by tags, which will be a main difference between the Equipment and Folder components–the Folder component is merely an organizing component; in the future, it will not carry any Haystack tags functionalities. It is therefore very important to use the Equipment components in the applications structure.

Also refer to: Equipment Manager

5.1 Applications Libraries

The nano EDGE ENGINE Applications libraries provide sets of components (service and basic) that allow to create HVAC 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;
- FCU library;
- Logic library;
- Math library;
- Other library;
- Process library;
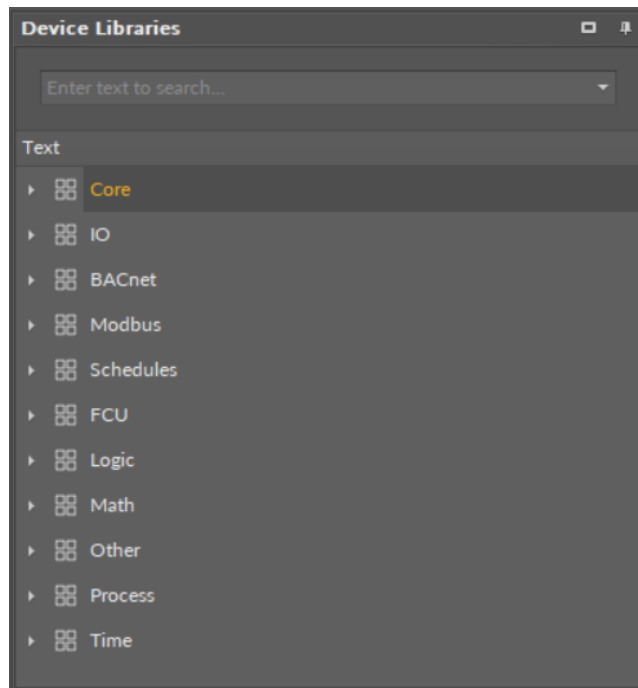- Time library;
- Schedules library.

Figure 13. nano Edge Engine libraries

5.2 Core (for Applications)

WARNING!

To learn about the components included in the Core library, please refer to the nano EDGE ENGINE Programming User Manual.

5.3 Logic

> **WARNING!**
>
> To learn about the components included in the Logic library, please refer to the nano EDGE ENGINE Programming User Manual.

5.4 Math

> **WARNING!**
>
> To learn about the components included in the Math library, please refer to the nano EDGE ENGINE Programming User Manual.

5.5 Process

> **WARNING!**
>
> To learn about the components included in the Process library, please refer to the nano EDGE ENGINE Programming User Manual.

5.6 Time

> **WARNING!**
>
> To learn about the components included in the Time library, please refer to the nano EDGE ENGINE Programming User Manual.

5.7 FCU

> **WARNING!**
>
> To learn about the components included in the FCU library, please refer to the nano EDGE ENGINE Programming User Manual.

5.8 Other

> **WARNING!**
>
> To learn about the components included in the Other library, please refer to the nano EDGE ENGINE Programming User Manual.

5.9 Schedules

> **WARNING!**
>
> To learn about the components included in the Schedules library, please refer to the nano EDGE ENGINE Programming User Manual.

5.10 Quick Start-up of Applications

The RAC18-IP controller offers smart and easy way of creating applications.

The application created with the **nano EDGE ENGINE** device refers to 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 will be the place to connect with additional functionalities.

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

Step 1: Make sure that the device is correctly connected and set up. Detailed instructions are available in the device's RAC18-IP Quick Start-up (Step 1-7).

Step 2: Go to the Networks container to configure the external communication settings (Modbus or BACnet). Configure all the network-specific parameters there in order to enable proper communication in Modbus RTU as a client, Modbus TCP/IP as a server device, or in BACnet IP as a client or server device.

Step 3: 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. The first and foremost is to 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 4: Go to the Applications container. This is the place to add Application components from the Core library–in fact, 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.

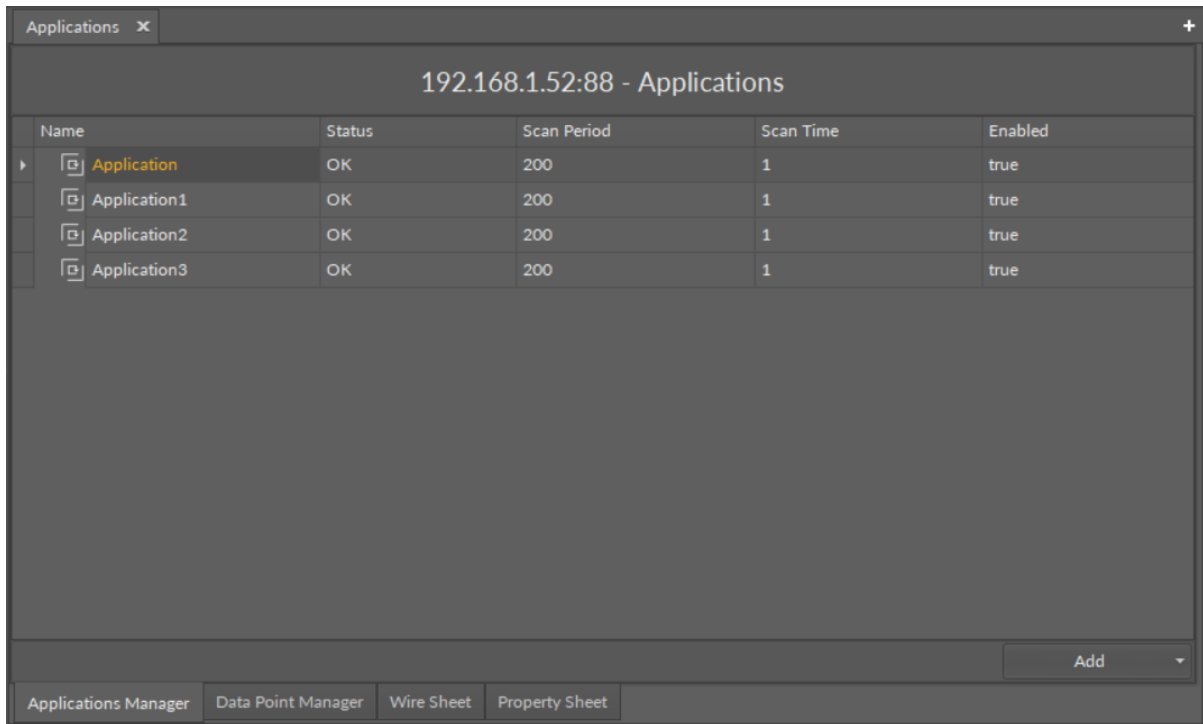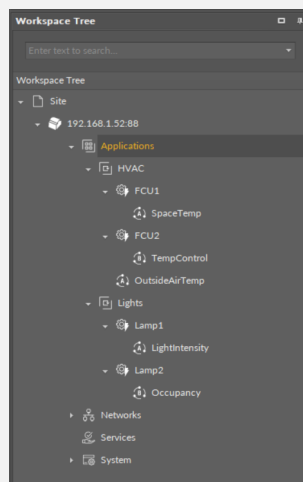*Figure 14. The Applications Manager*

Good Practice Tip

In future developments, the RAC18-IP controller will have a Haystack functionality implemented in its firmware, which calls for some good practice mechanisms that can be introduced now, with the current functionalities.

The Haystack functionality will offer multiple advantages of using tags–it will help identify parts of equipment controlled by the application and used Data Points, filter data by equipment, sensor, or value, any many more.

In order to facilitate a future use of the full Haystack functionality, the Equipment component has been included in the Core library. It is therefore recommended 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

Such structure will be fully recognized in the Haystack functionality and will require minimum effort to use its full potential once updated.

Difference Between Equipment and Folder Components

The Equipment component with Haystack functionality will allow to identify the types and other characteristics of controlled devices by tags, which will be a main difference between the Equipment and Folder components–the Folder component is merely an organizing component; in the future, it will not carry any Haystack tags functionalities. It is therefore very important to use the Equipment components in the applications structure.

Also refer to: Equipment Manager

Step 5: (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.
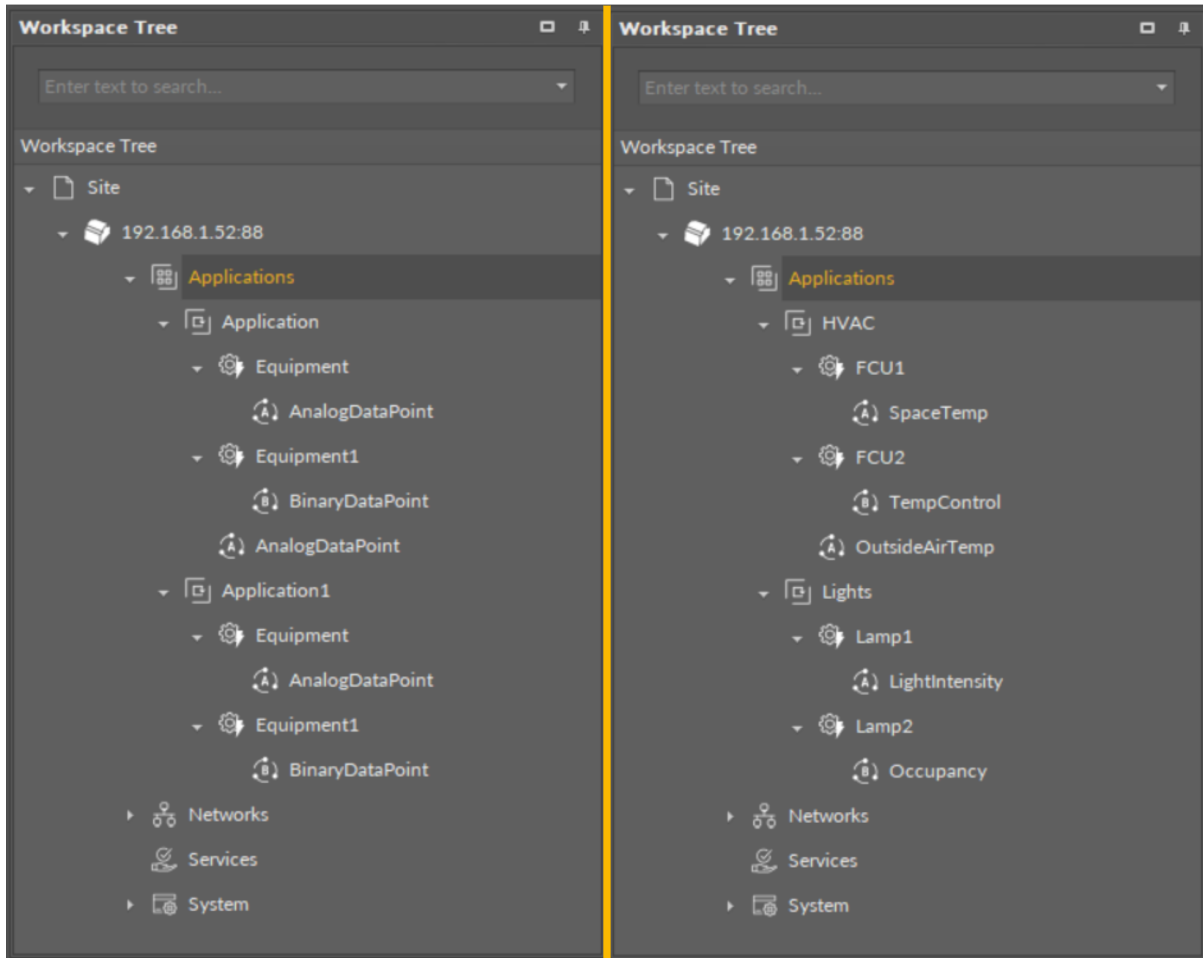

Figure 15. The recommended structure for applications

Step 6: (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 evoke 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 IO components.
>
> • Analog Data Point;
> • Binary Data Point.

Step 7: 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. Detailed technical information about linking in the iC Tool is available in the Linking Components 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.

5.10.1 Applications Manager

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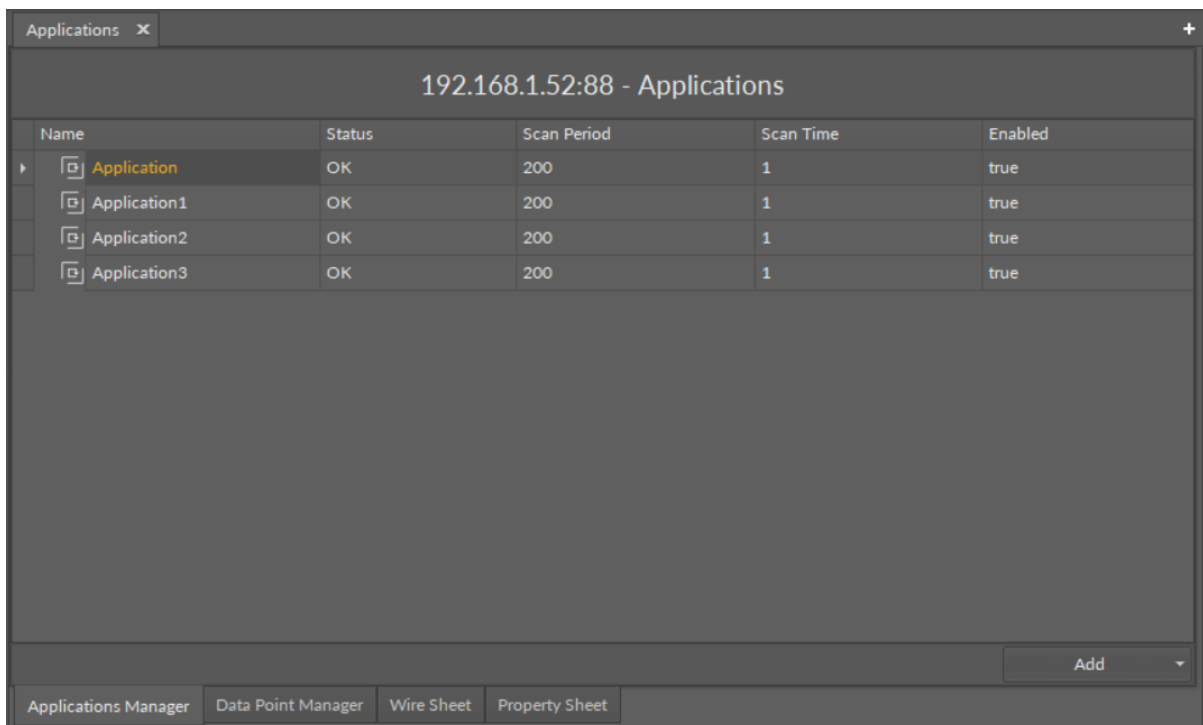

Figure 16. The Applications Manager

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.

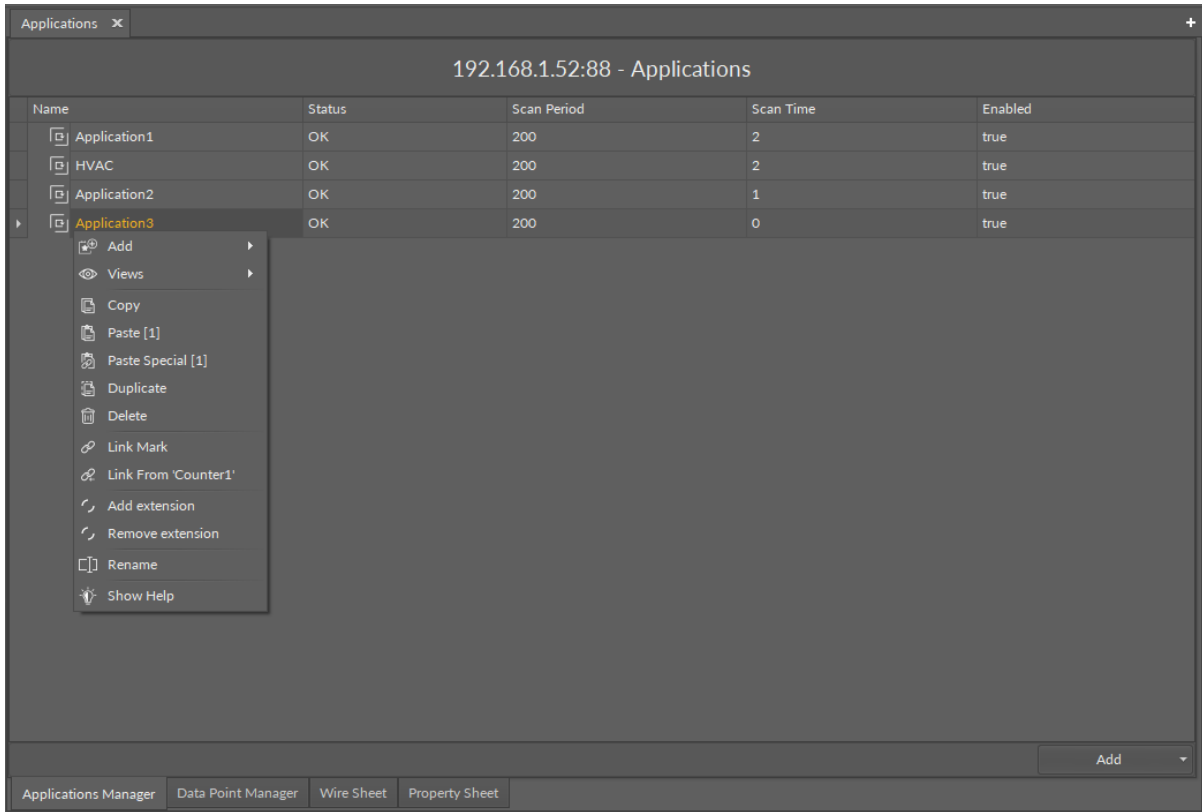In the Applications Manager, it is possible to add, remove, copy, or duplicate Application components.

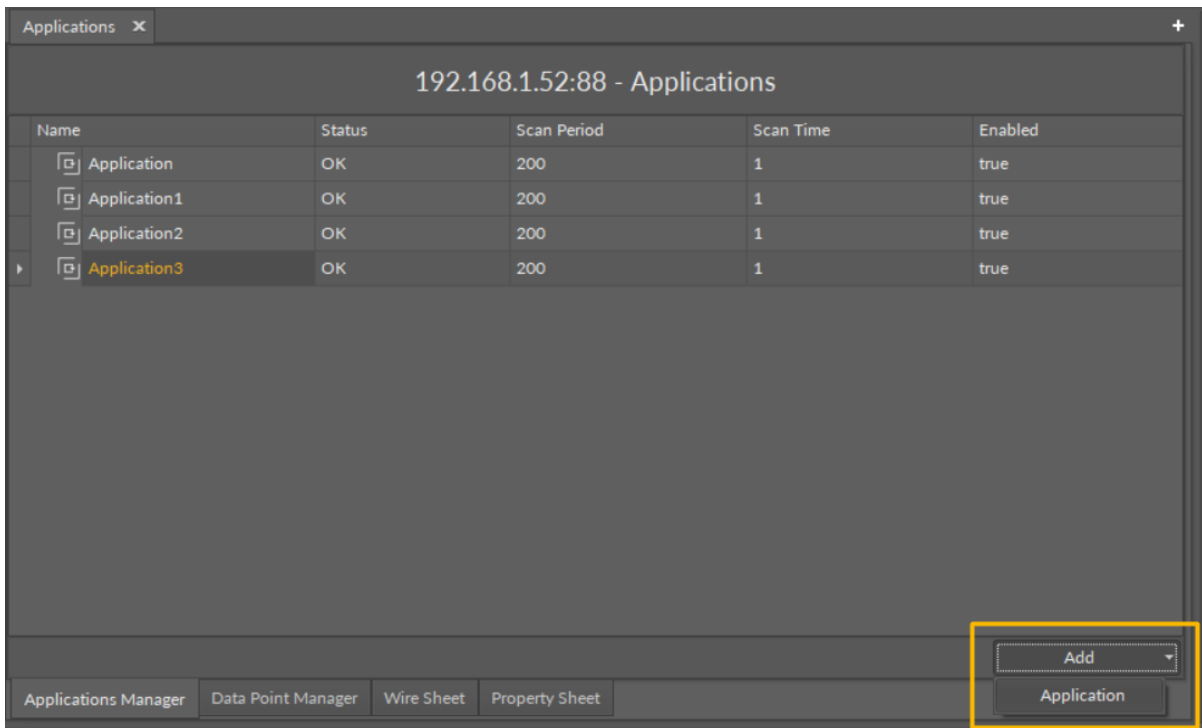

Figure 17. Options in the Applications Manager

Figure 18. Options in the Applications Manager

Opening the Applications Manager

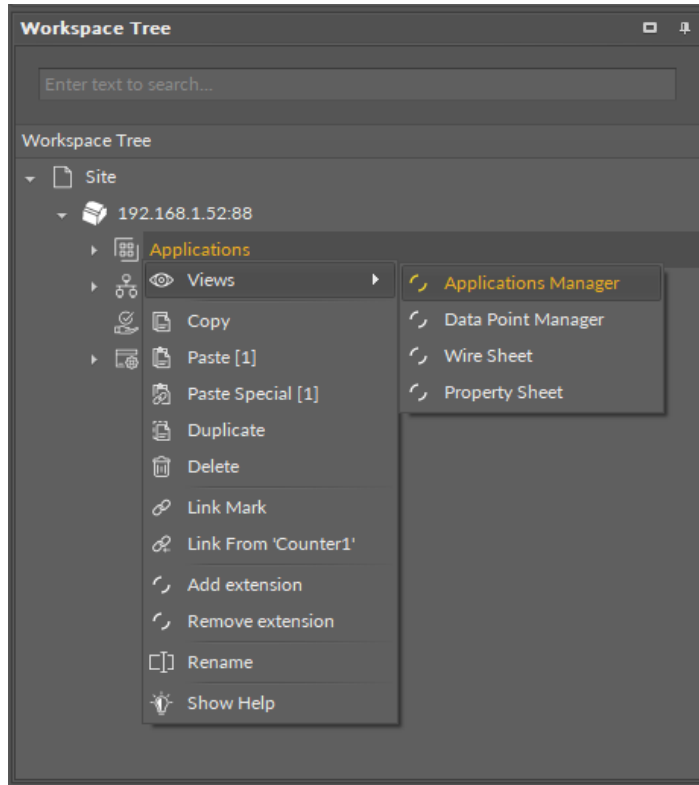The Applications Manager view is accessible in the context menu of the Applications container.

Figure 19. Accessing the Applications Manager from the Applications container context menu

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

5.10.2 Equipment Manager

The Equipment Manager is a special view that allows to manage the Equipment components, Data Points, and Folder components added to a specific Application component.
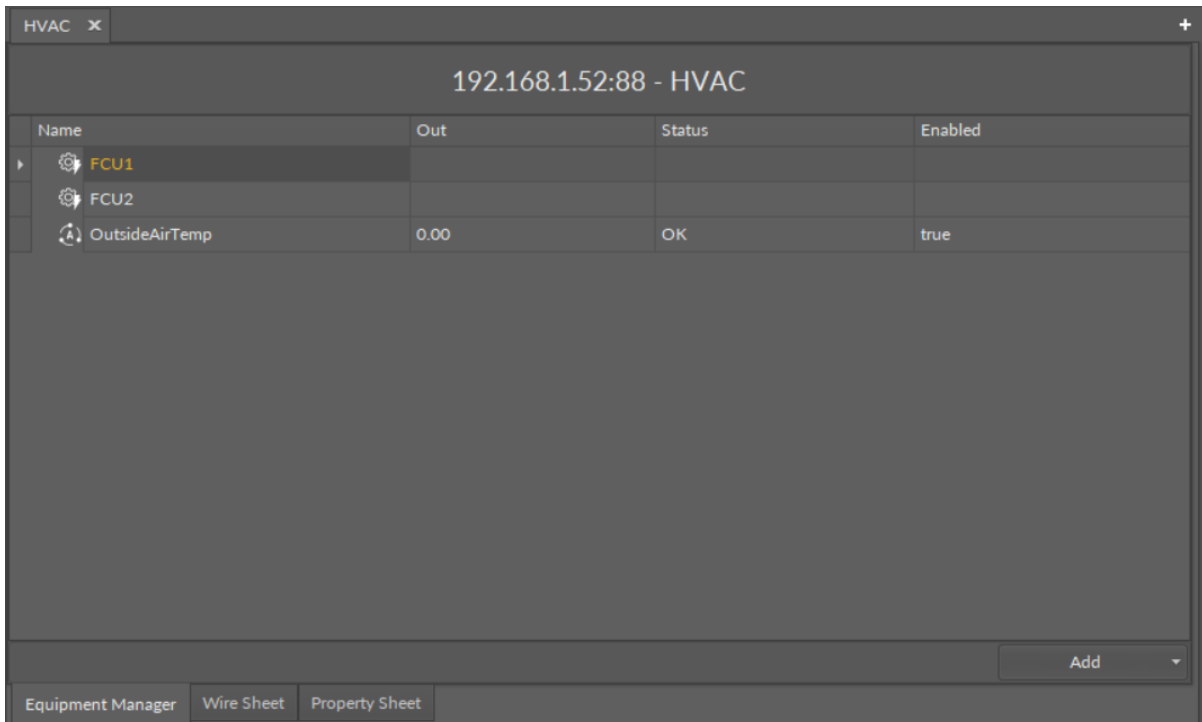

Figure 20. The Equipment Manager

The Equipment Manager lists all Equipment components and Data Points used in the specific Application component. The view shows the following fields:

- name of the Data Point;
- value on the Out slot;
- Data Point's status;
- enabled or disabled status.

The Equipment Manager view is not editable; however, it allows to add, remove, duplicate, and rename Equipment components, Data Points (withing the Equipment components and directly in the Application component), and Folder components, which help organize components in the application.

Once a specific Data Point is clicked in the Equipment 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 Application Manager, their common slots are available for multiedition in the Object Properties window.
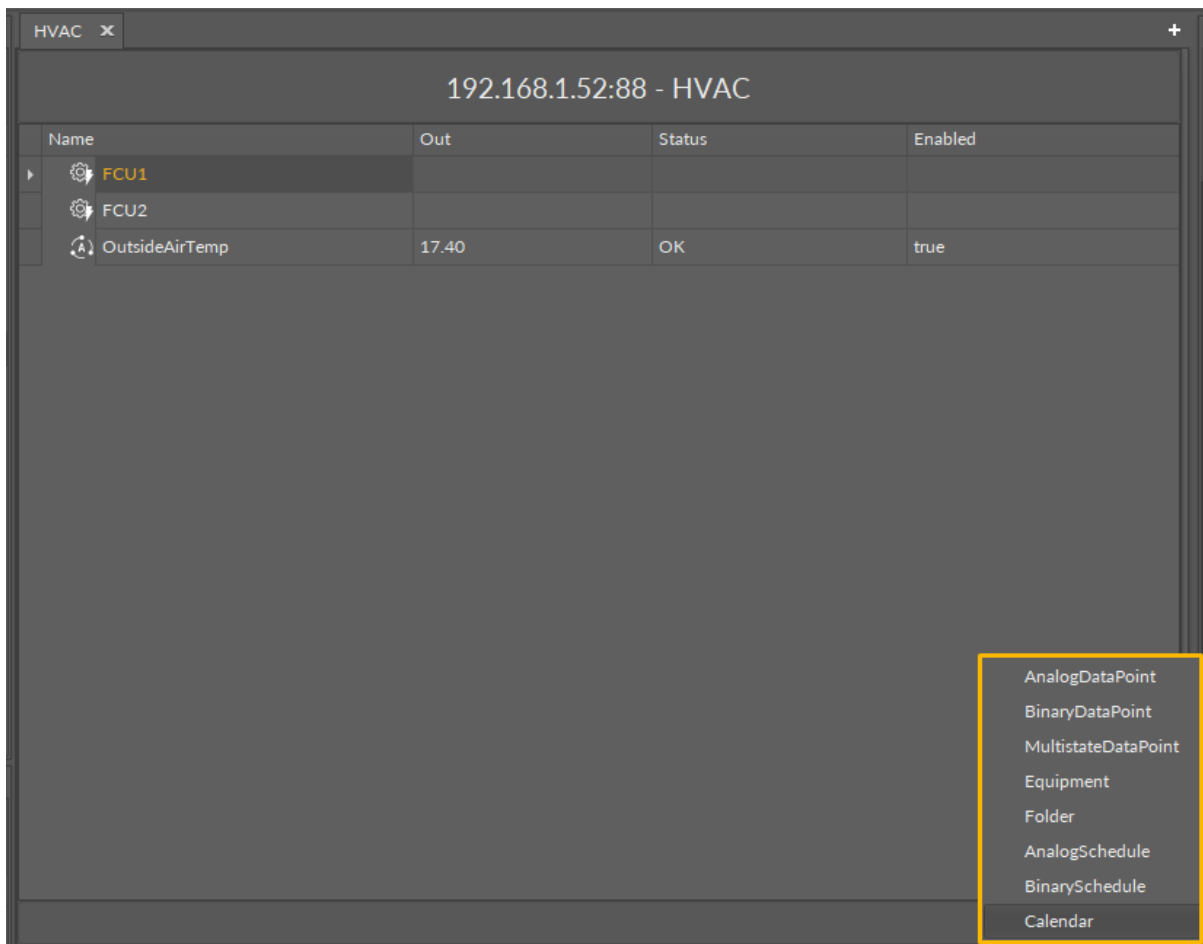

Figure 21. Options in the Equipment Manager

Opening the Equipment Manager

The Equipment Manager view is accessible in the context menu of the Application component.
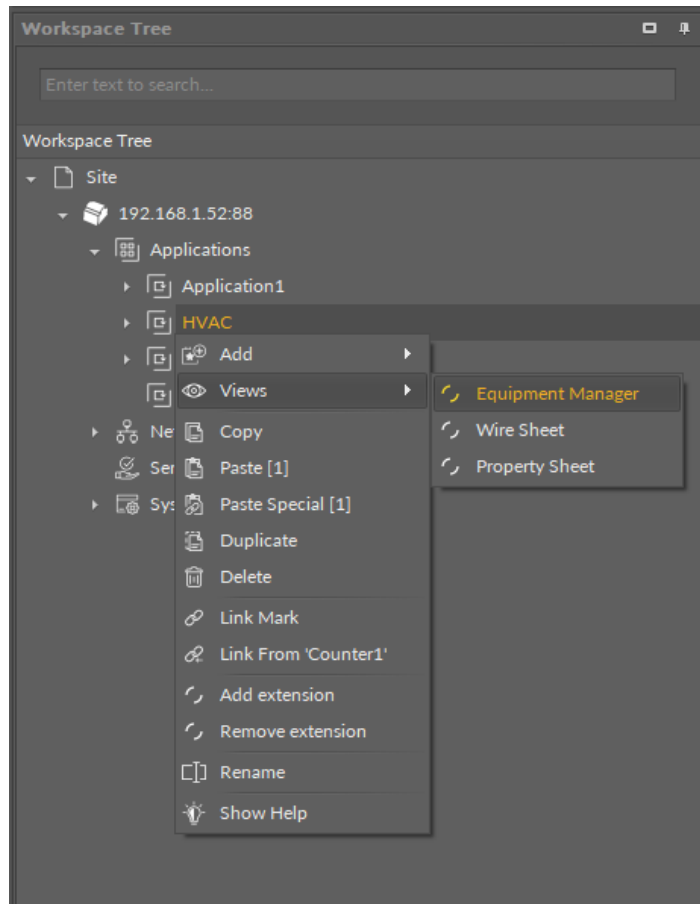
Figure 22. Accessing the Application Manager from the Applications container context menu

The Equipment Manager view is also automatically opened if either the Application component is double-clicked in the Workspace Tree window.

5.10.3 Data Point Manager

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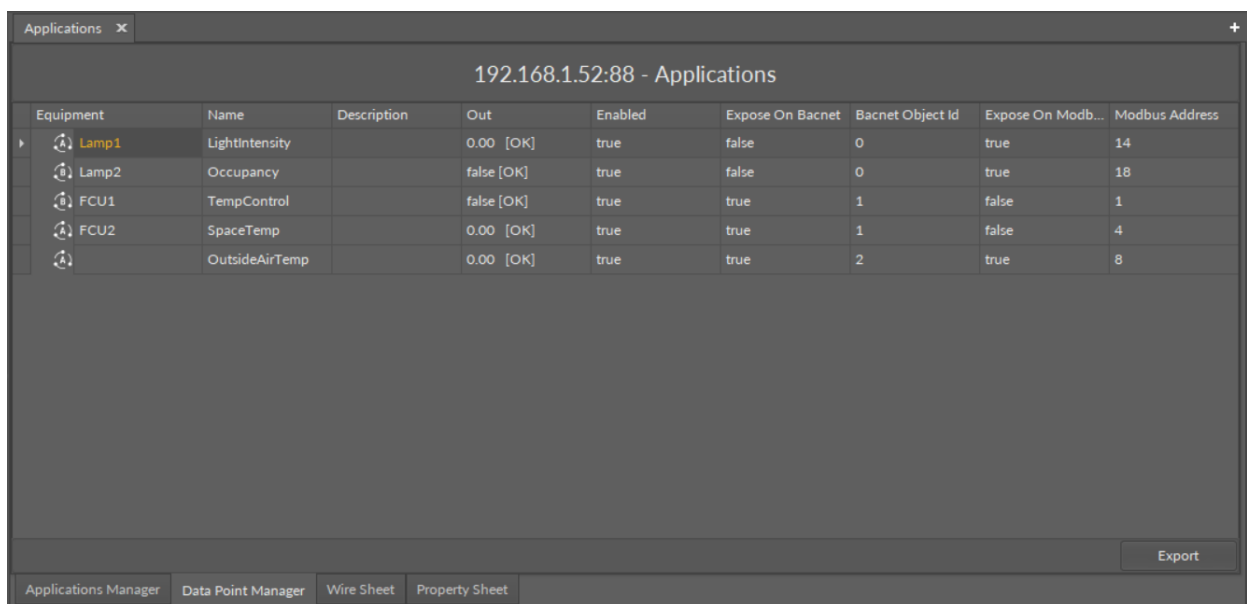

Figure 23. The Data Point Manager

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.

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 RAC18-IP (Device for BACnet) component;
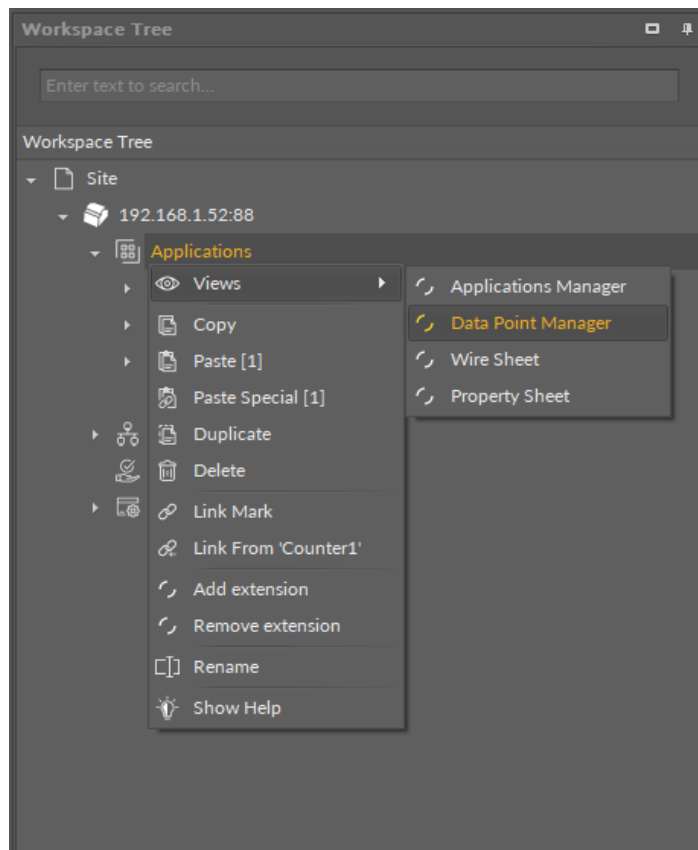- in the context menu of the RAC18-IP (Device for Modbus) component.

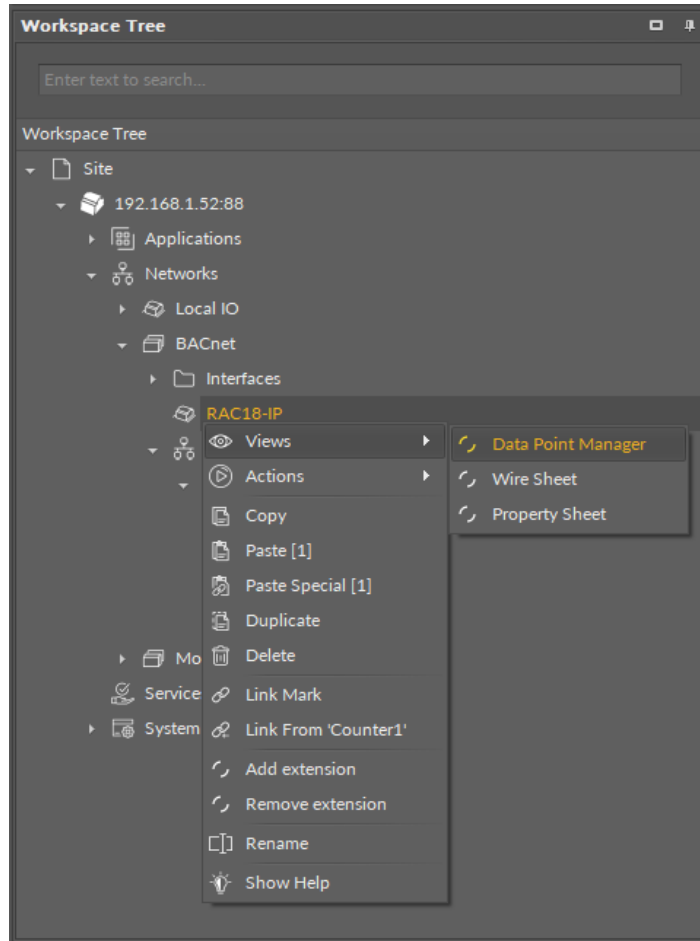*Figure 24. Accessing the Data Point Manager from the Applications container context menu*

Figure 25. Accessing the Data Point Manager from the context menu of RAC18-IP (LocalDevice) component in the Networks container

The Data Point Manager view is also automatically opened if either the Applications container or the RAC18-IP (Device 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 RAC18-IP Device showing information specific to the network:

Figure 26. The Data Point Manager for the RAC18-IP BACnet LocalDevice

Figure 27. The Data Point Manager for the RAC18-IP 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.

6 Networks Container

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.
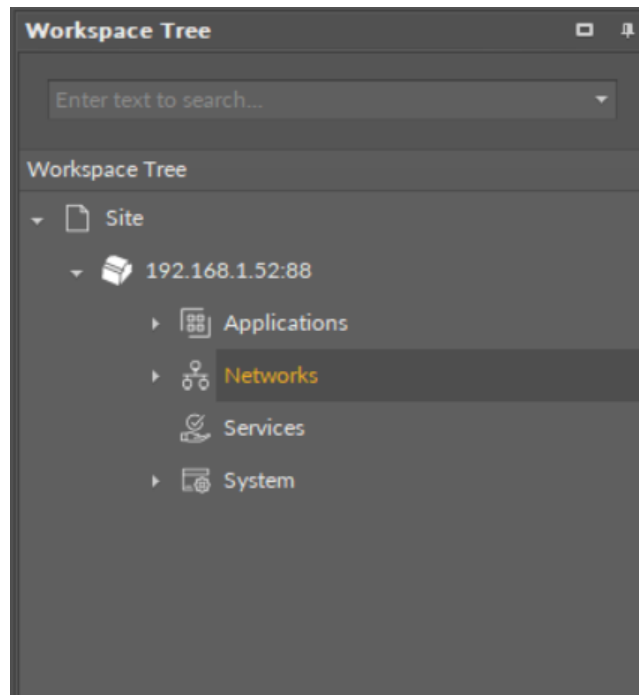

Figure 28. The Networks container

6.1 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;
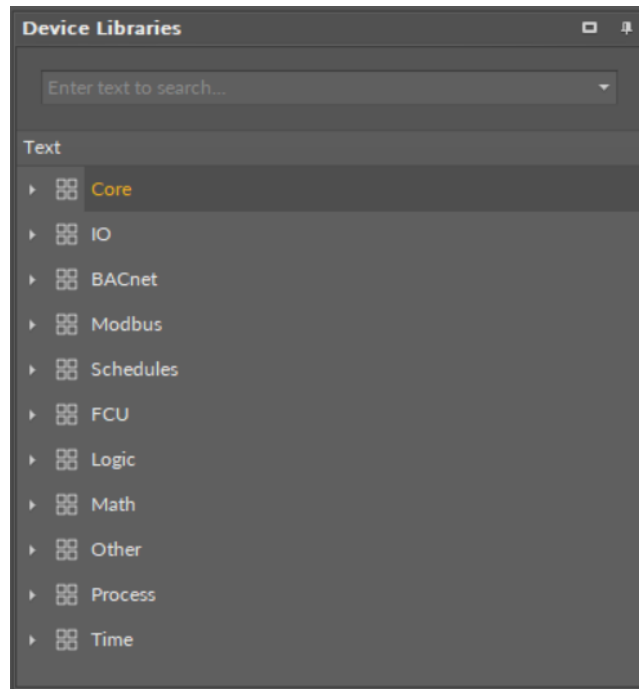· BACnet library;
· Modbus library.

Figure 29. nano Edge Engine libraries

6.2 Core (for Networks)

> **WARNING!**
>
> To learn about the components included in the Core library, please refer to the nano EDGE ENGINE Programming User Manual.

6.3 IO

> **WARNING!**
>
> To learn about the components included in the IO library, please refer to the nano EDGE ENGINE Programming User Manual.

6.4 Quick Start-up of LocalIO

In order to have the components available in the IO library working properly, it is required to follow these steps:

Step 1: Having the device added correctly in the iC Tool, expand the Networks container, and go to the LocalIO component.

Step 2: 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 Network container.

Step 3: 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.

> **Worth to Notice:**
>
> 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.

6.4.1 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);
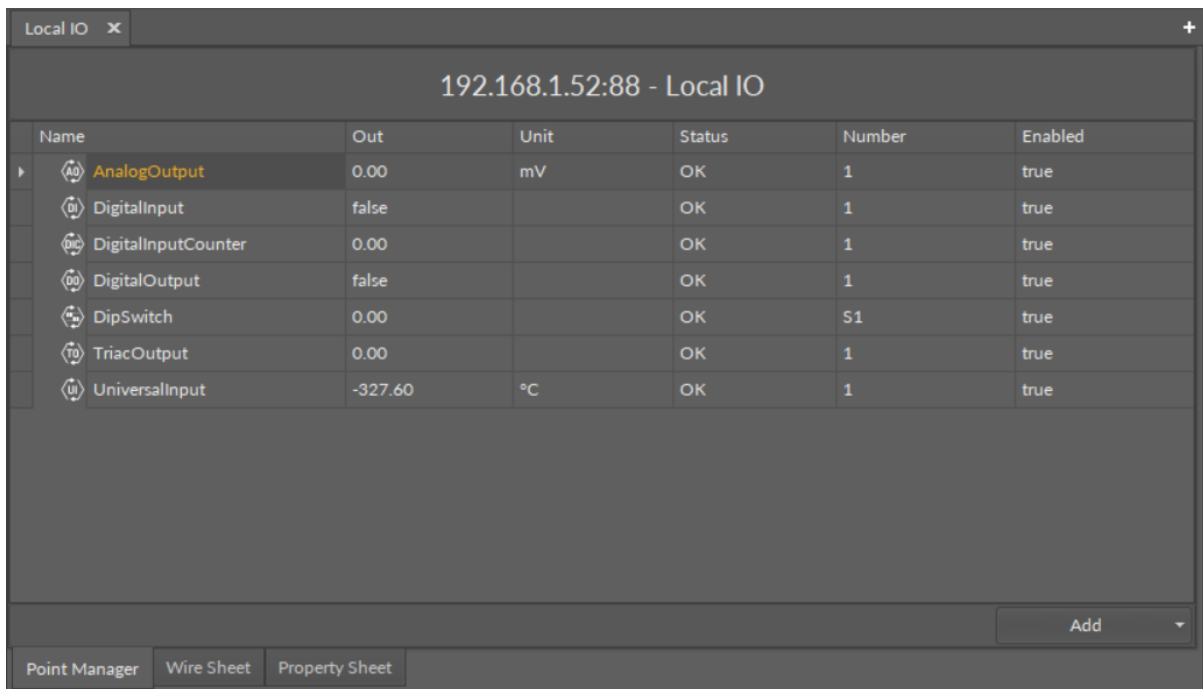- status;
- number;
- enabled or disabled state.

Figure 30. The Point Manager view

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.
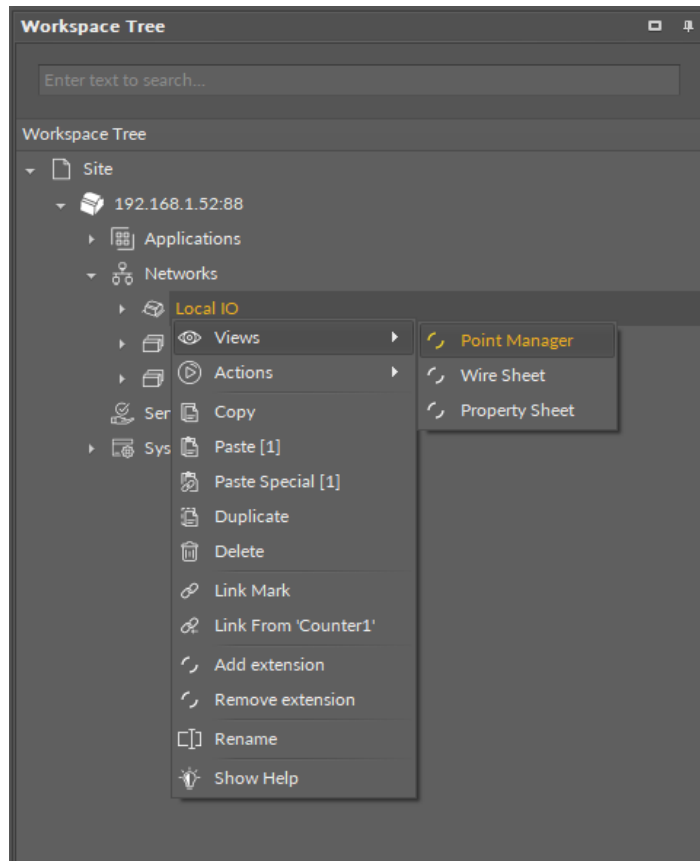
Figure 31. Opening the Point Manager view

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.
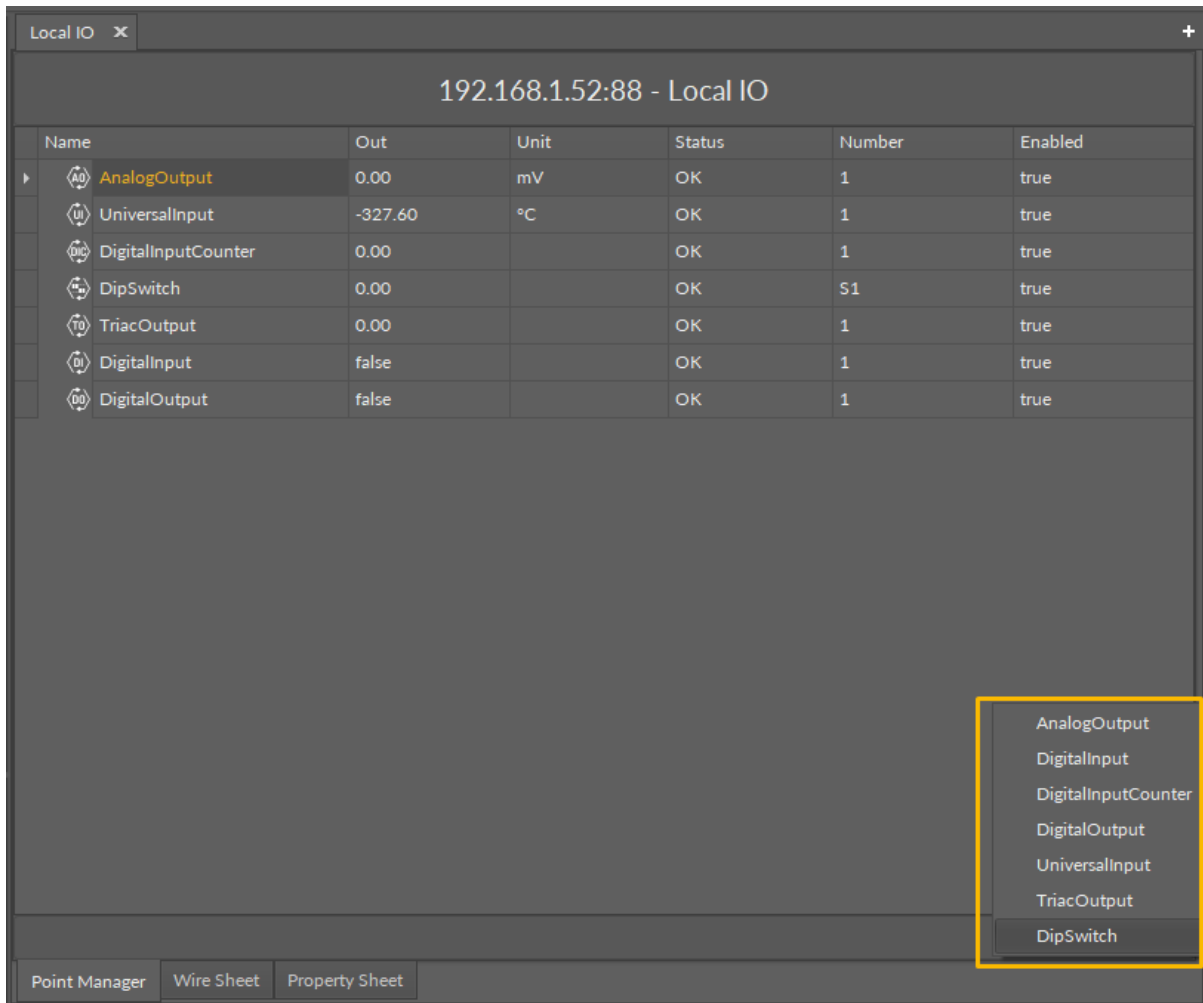
Figure 32. The Add button in the Point Manager

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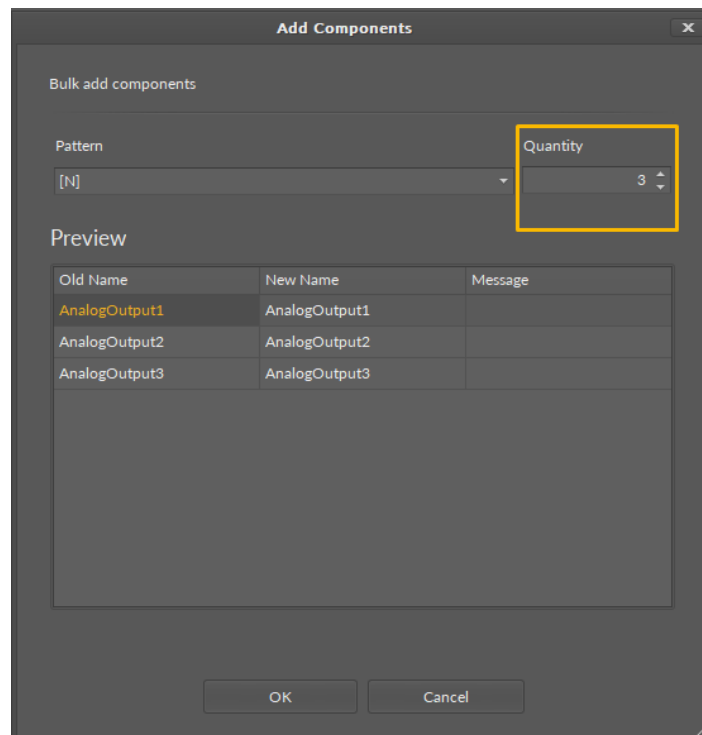

Figure 33. Adjusting quantity of added IO points

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.

6.5 BACnet

> **WARNING!**
>
> To learn about the components included in the BACnet library, please refer to the nano EDGE ENGINE Programming User Manual.

6.6 Quick Start-up of BACnet

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

Step 1: Having the device added correctly in the iC Tool, expand the Networks container, and go to the BACnet component.

Step 2: Go to the Interfaces component:

(a) The Ethernet component is added by default. Go to the Ethernet component to configure the IP port for BACnet IP communication if necessary.

(b) Add the SerialPort component from the Core library for serial communication if necessary. Configure the RS485 port for BACnet MS/TP communication.

> **Worth to Notice:**
>
> 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.

Step 3: The RAC18-IP controller allows to communicate as the BACnet IP or MS/TP client device and BACnet IP or MS/TP server device.

(a) The RAC18-IP component (LocalDevice) for BACnet IP server device is also added by default. In order to configure the BACnet server device, go the the component.

(b) In order to configure the BACnet IP or MS/TP client network, the Network and Device components need to be added, along with AnalogPoint, BinaryPoint, or MultistatePoint components, as necessary. Go to the Device Libraries and expand the Core library for the Network component and BACnet 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 BACnet component in the Network container.

> **Worth to Notice:**
>
> 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 AnalogPoint/BinaryPoint/ MultistatePoint 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.

Step 4: 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.).

> **Worth to Notice:**
>
> In order to facilitate working with BACnet component, 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.

6.6.1 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 the network. Once the network, listed in the Network Manager, is double-clicked, the respective Network component is opened.
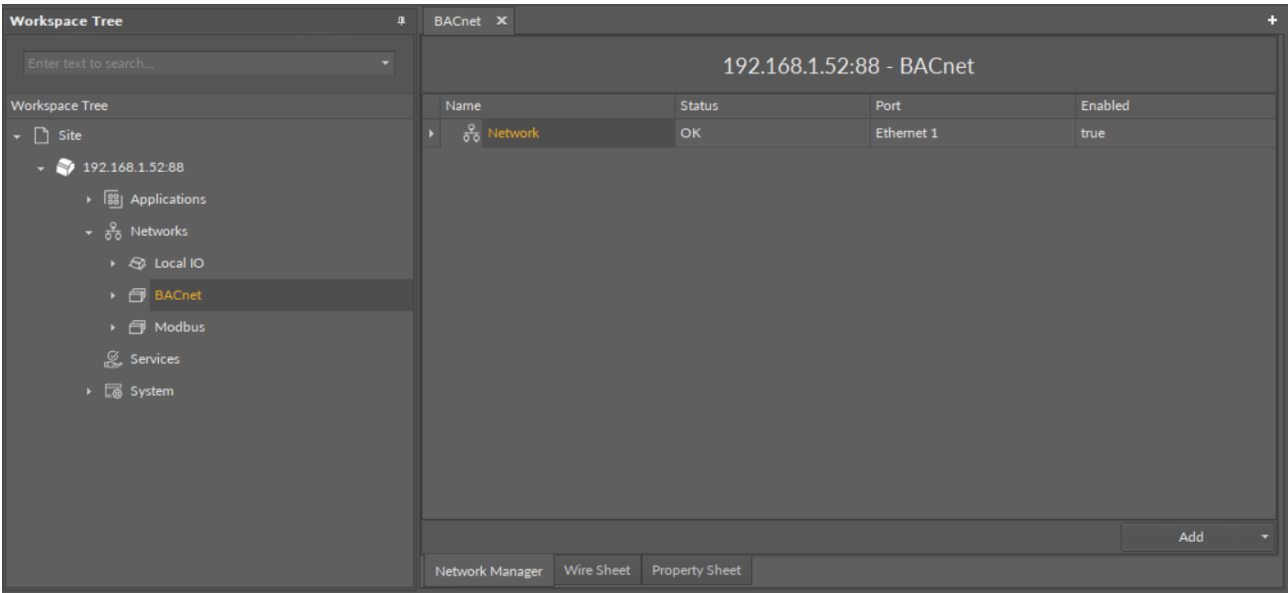
Figure 34. The Network Manager for BACnet

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.
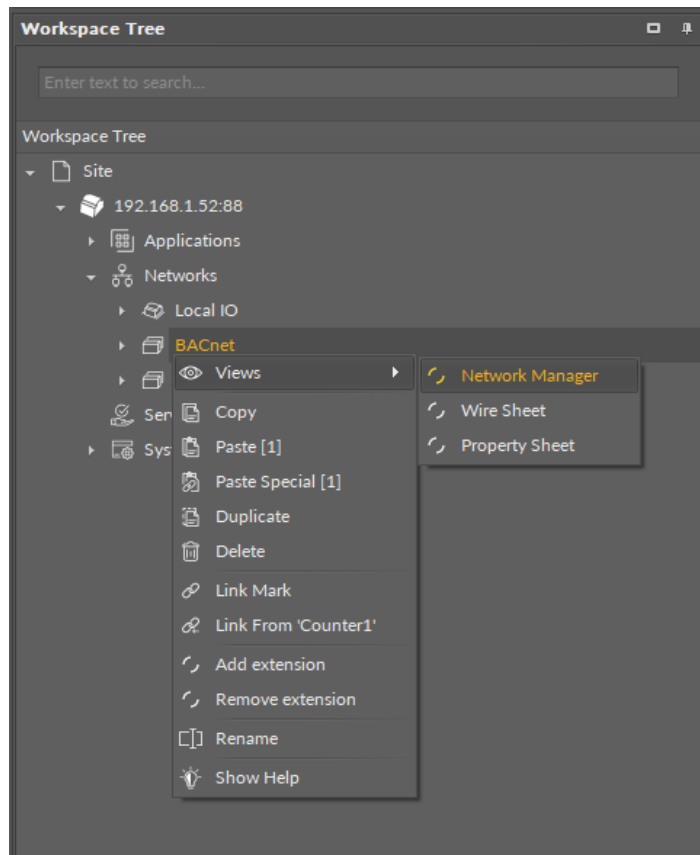

Figure 35. Opening the Network Manager

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.
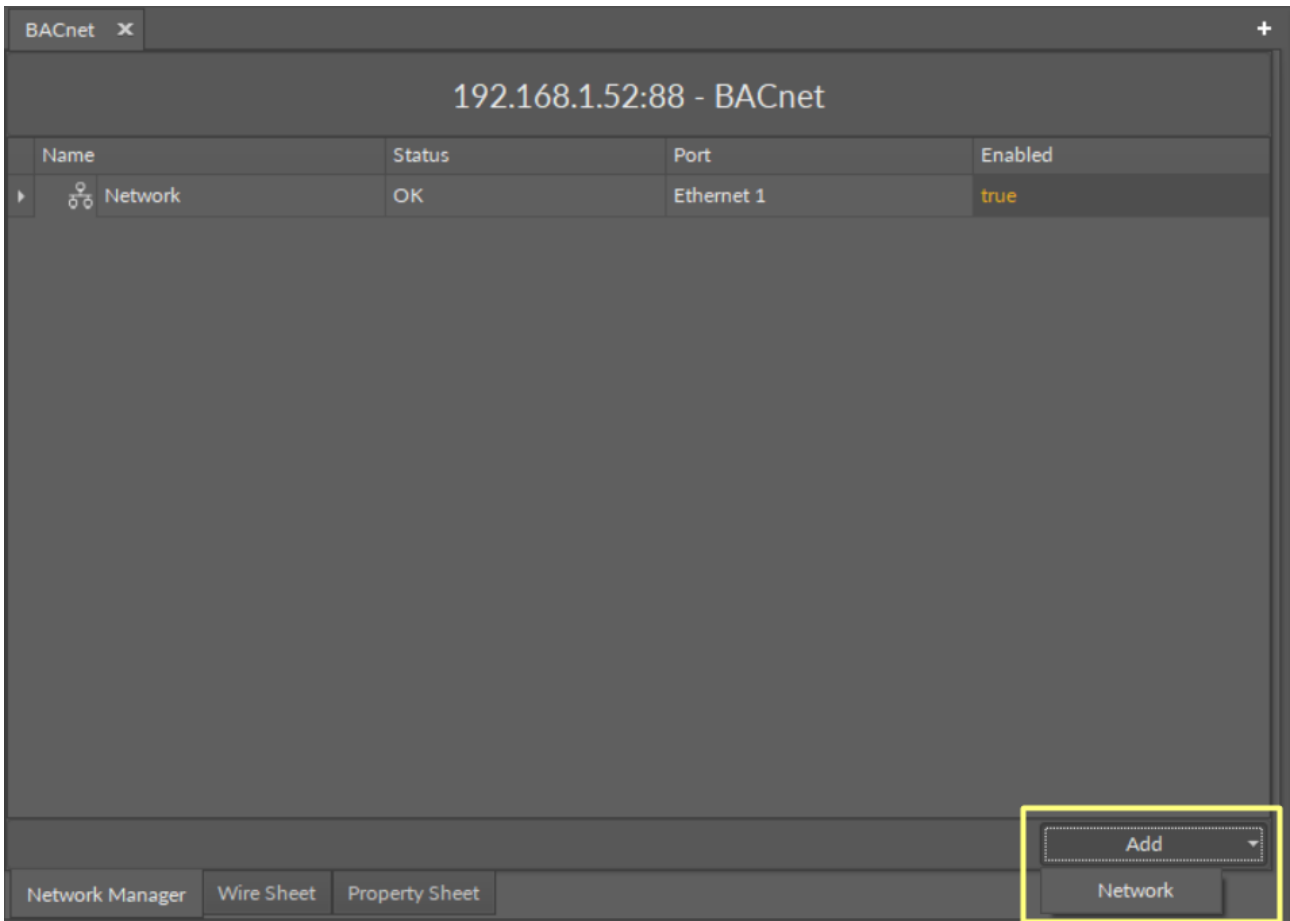

Figure 36. The Add button

Using this Add button opens the dialog window, which allows to adjust the quantity of networks to be added.
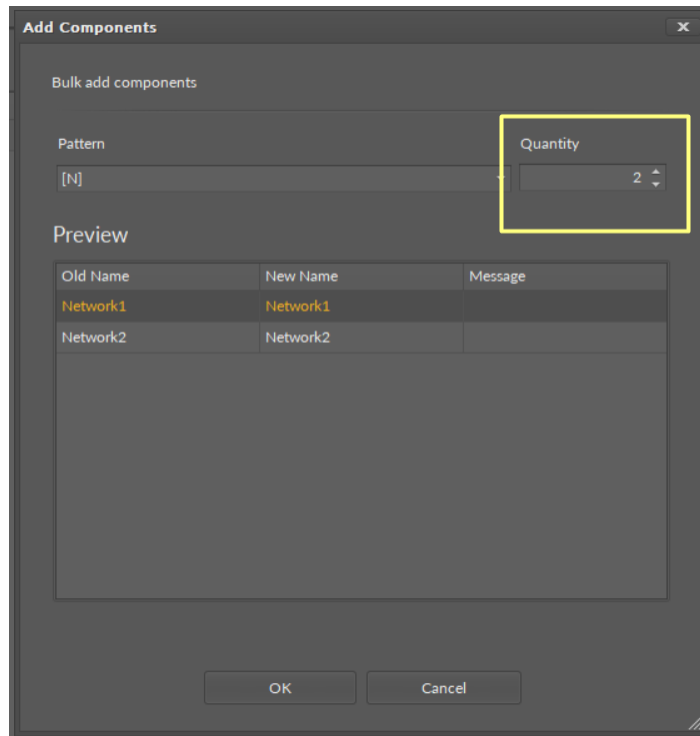
Figure 37. The dialog window

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.

6.6.2 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.
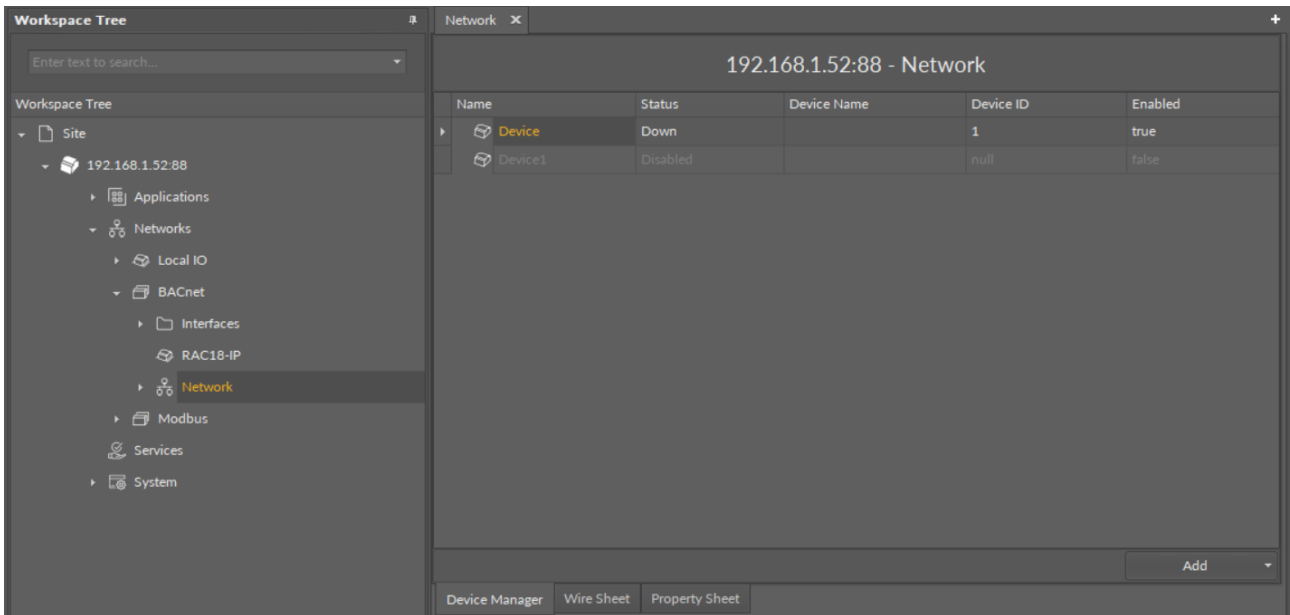
Figure 38. The Device Manager

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.
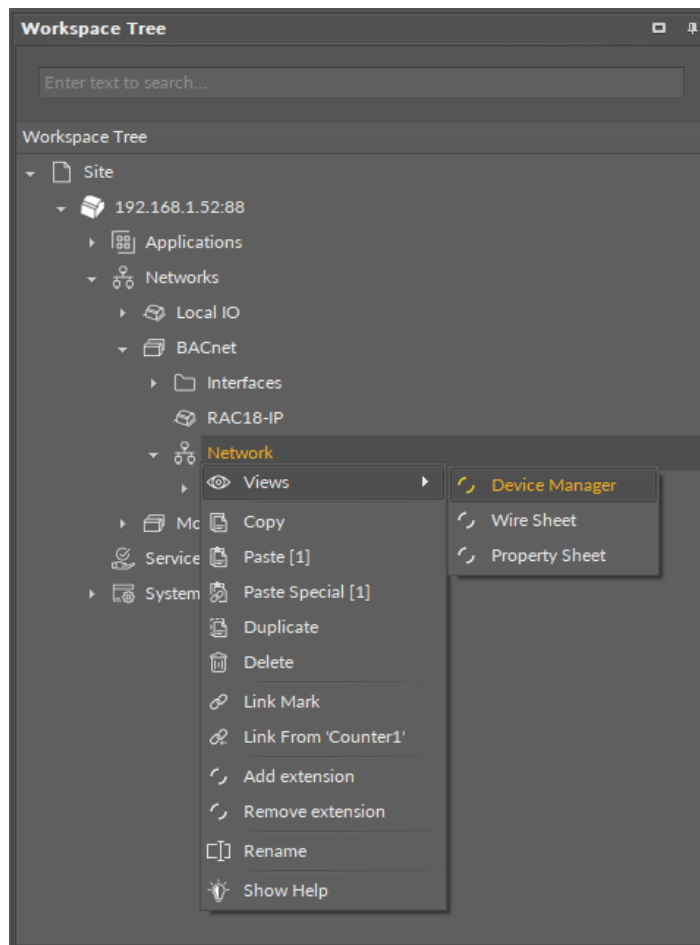

Figure 39. Opening the Device Manager

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.
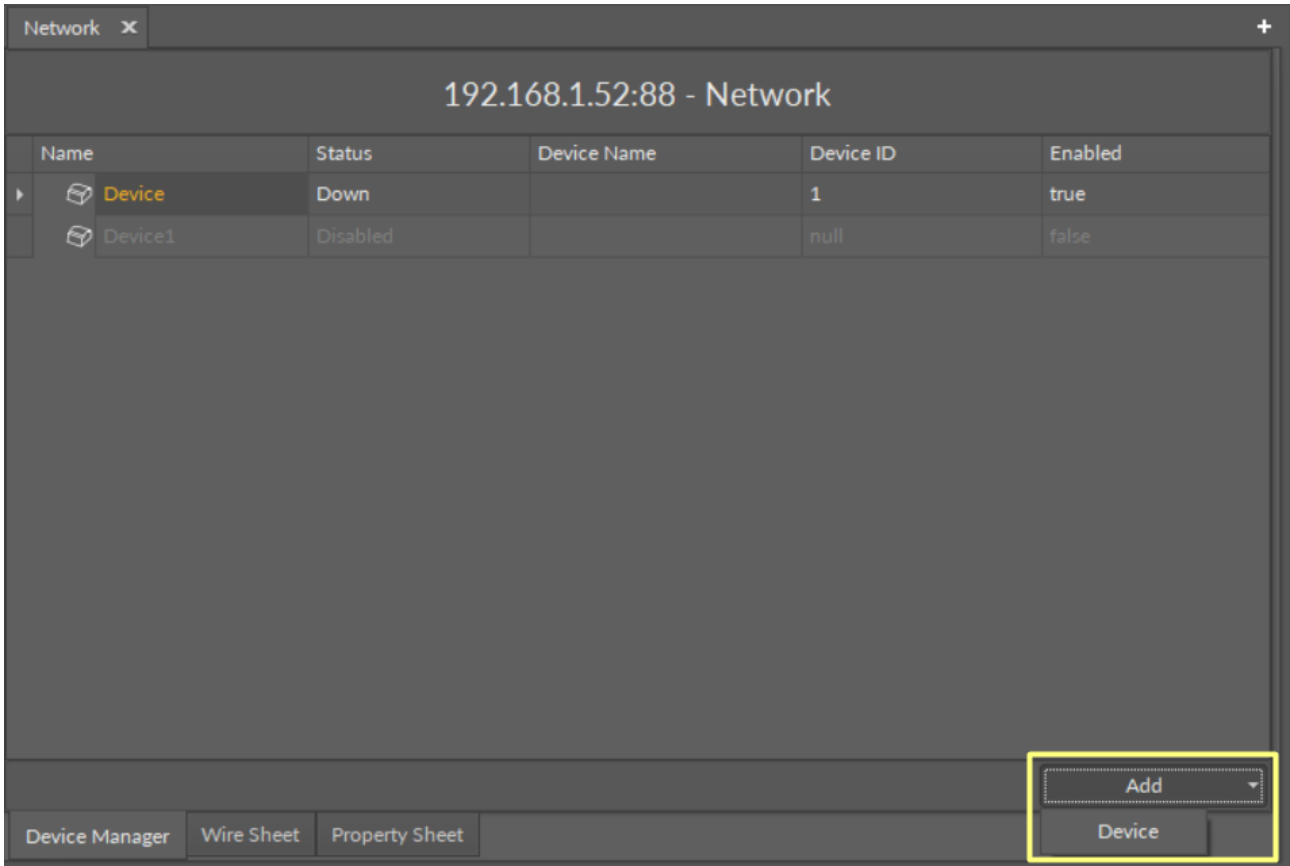

Figure 40. The Add button in the Device Manager

Using this Add button opens the dialog window, which allows to adjust the quantity of devices to be added.
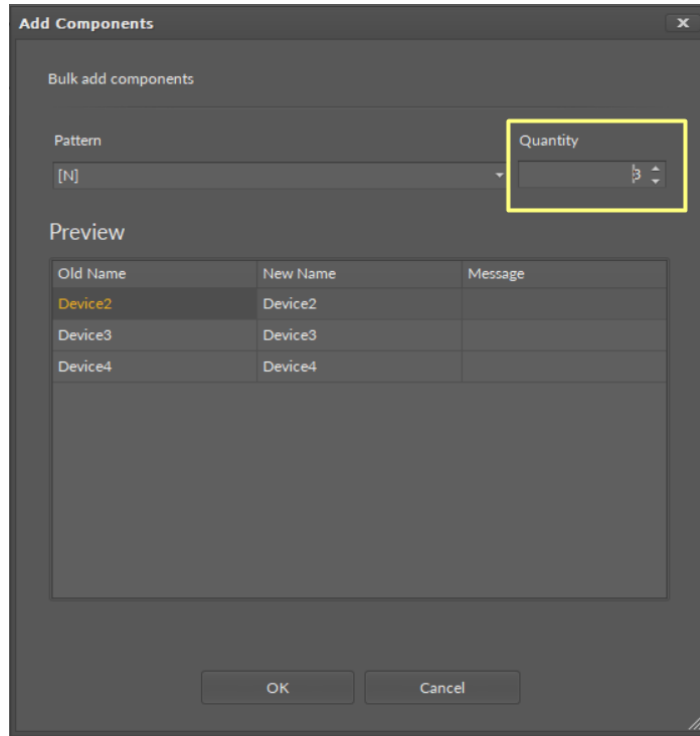
Figure 41. The dialog window

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.

6.6.3 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.
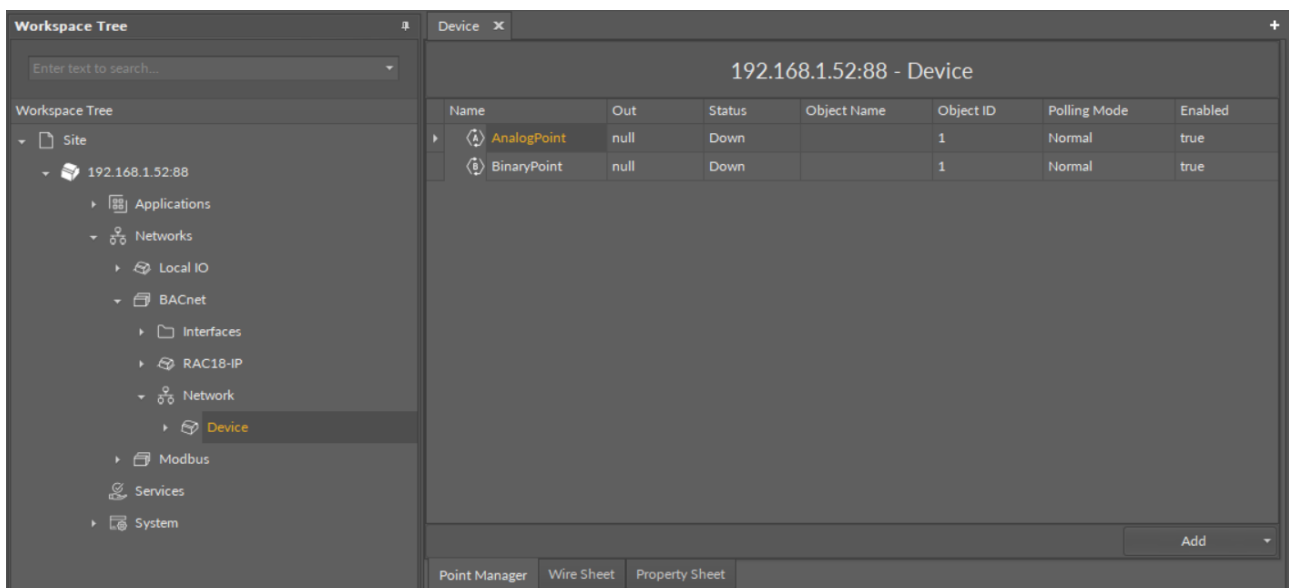

Figure 42. The Point Manager

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.
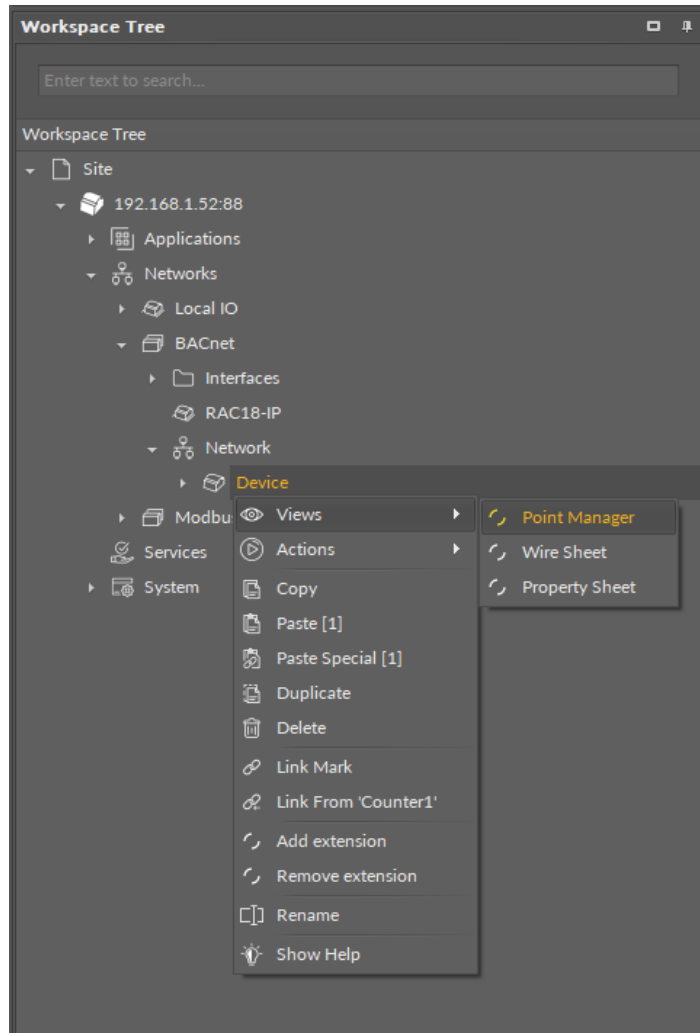

Figure 43. Opening the Point Manager

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.
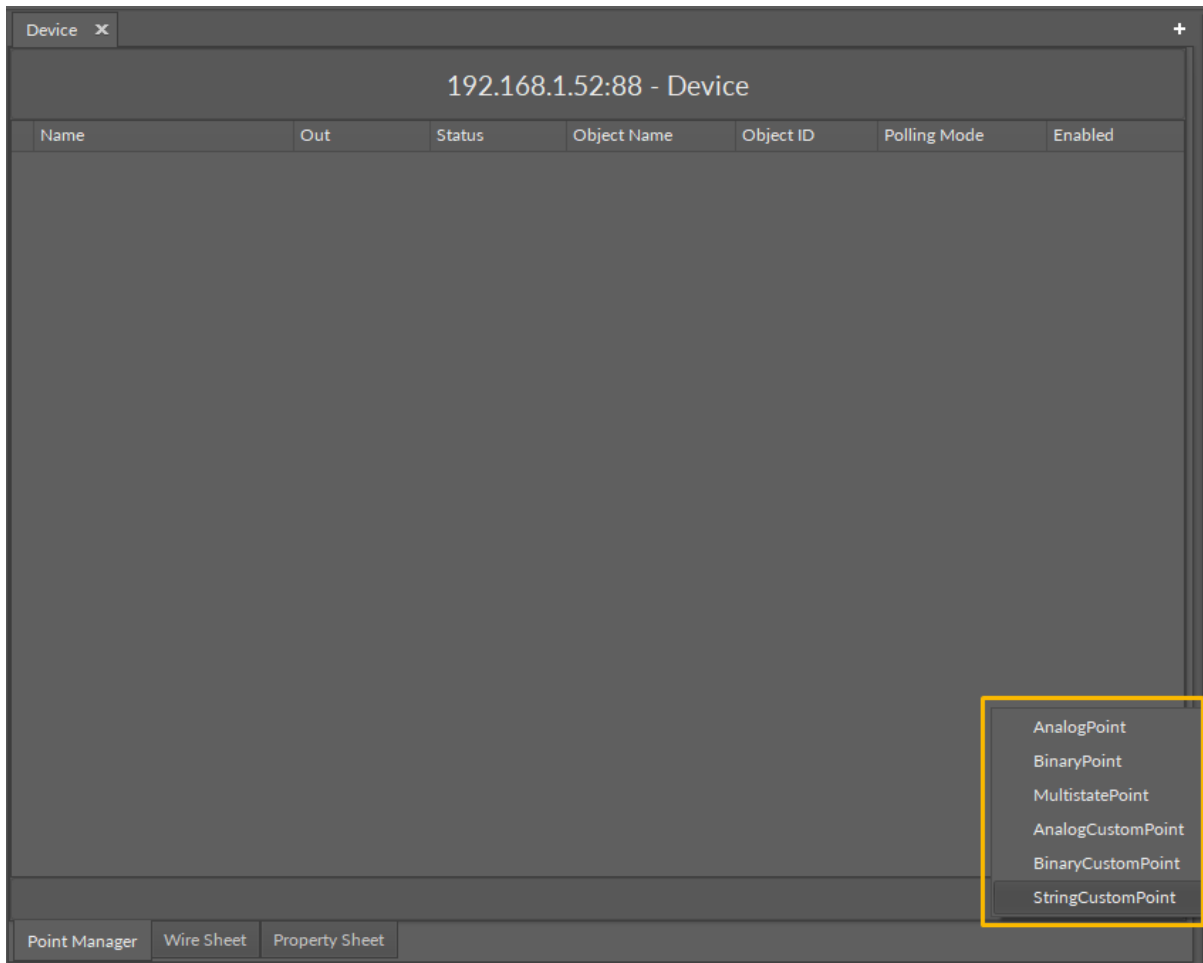
Figure 44. The Add button

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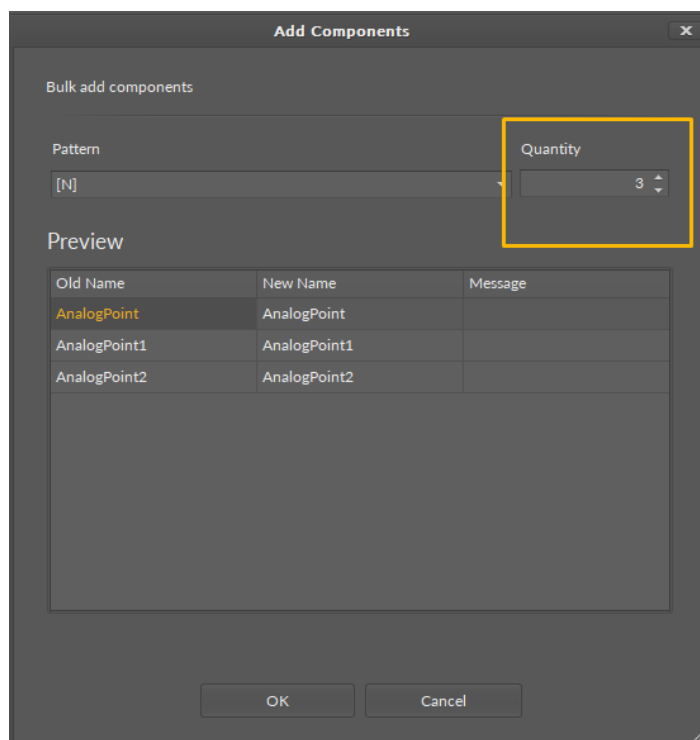

Figure 45. The dialog window

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.

6.7 Modbus

> **WARNING!**
>
> To learn about the components included in the Modbus library, please refer to the nano EDGE ENGINE Programming User Manual.

6.8 Quick Start-up for Modbus

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

Step 1: Having the device added correctly in the iC Tool, expand the Networks container, and go to the Modbus component.

Step 2: Go to the Interfaces component:

(a) The Ethernet component is added by default. Go to the Ethernet component to configure the IP port for Modbus TCP/IP communication if necessary.

(b) Add the SerialPort component from the Core library for serial communication if necessary. Configure the RS485 port for Modbus RTU communication.

> **Worth to Notice:**
>
> 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.

Step 3: The RAC18-IP controller allows to communicate as the Modbus RTU client or server device and Modbus TCP/IP client or server device.

(a): The RAC18-IP component (LocalDevice) for Modbus TCP/IP or RTU server device is also added by default. In order to configure the Modbus server device, go the the component.

> **Worth to Notice:**
>
> The RAC18-IP component (LocalDevice) 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 (for example, a RAC18-IP).

(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.

> **Worth to Notice:**
>
> 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.

Step 5: 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.

> **Worth to Notice:**
>
> In order to facilitate working with Modbus component, special views have been developed:
>
> - Network Manager, available in the Modbus 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.

6.8.1 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.
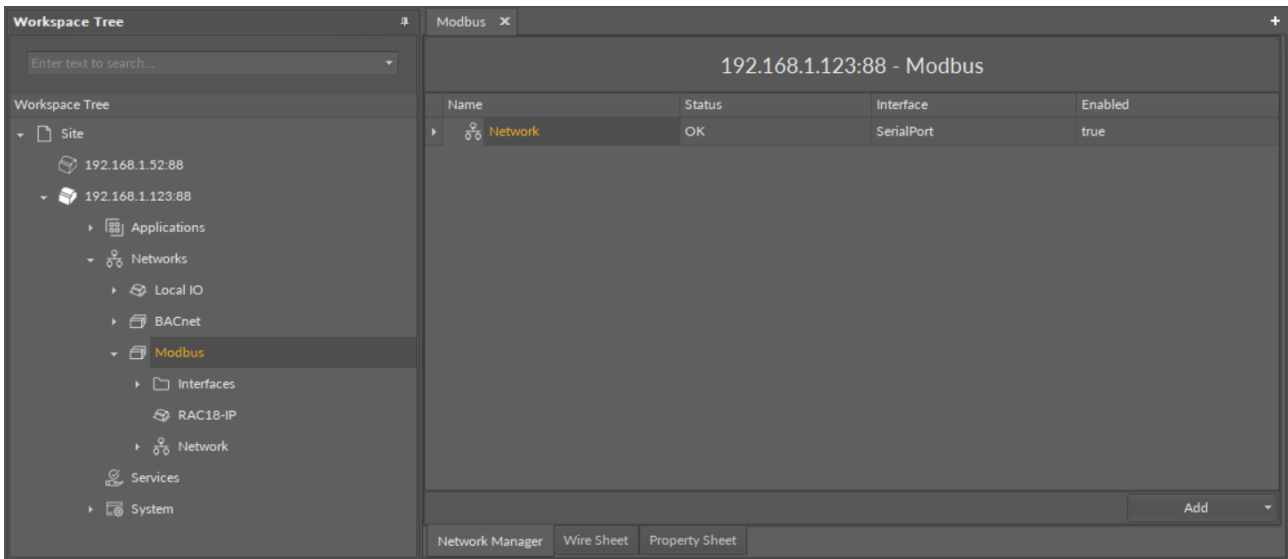

Figure 46. The Network Manager view

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.
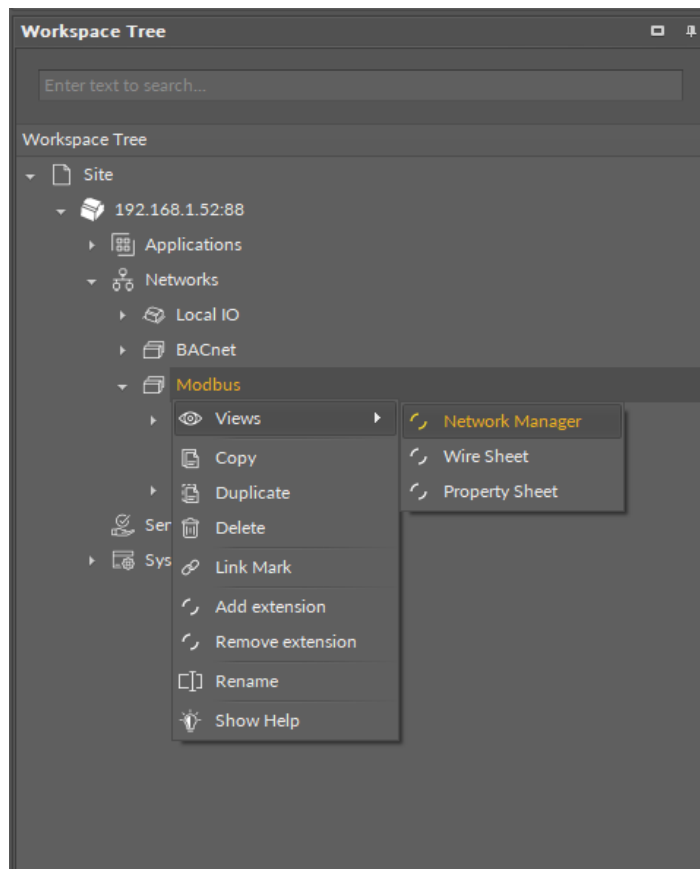

Figure 47. Opening the Network Manager

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.
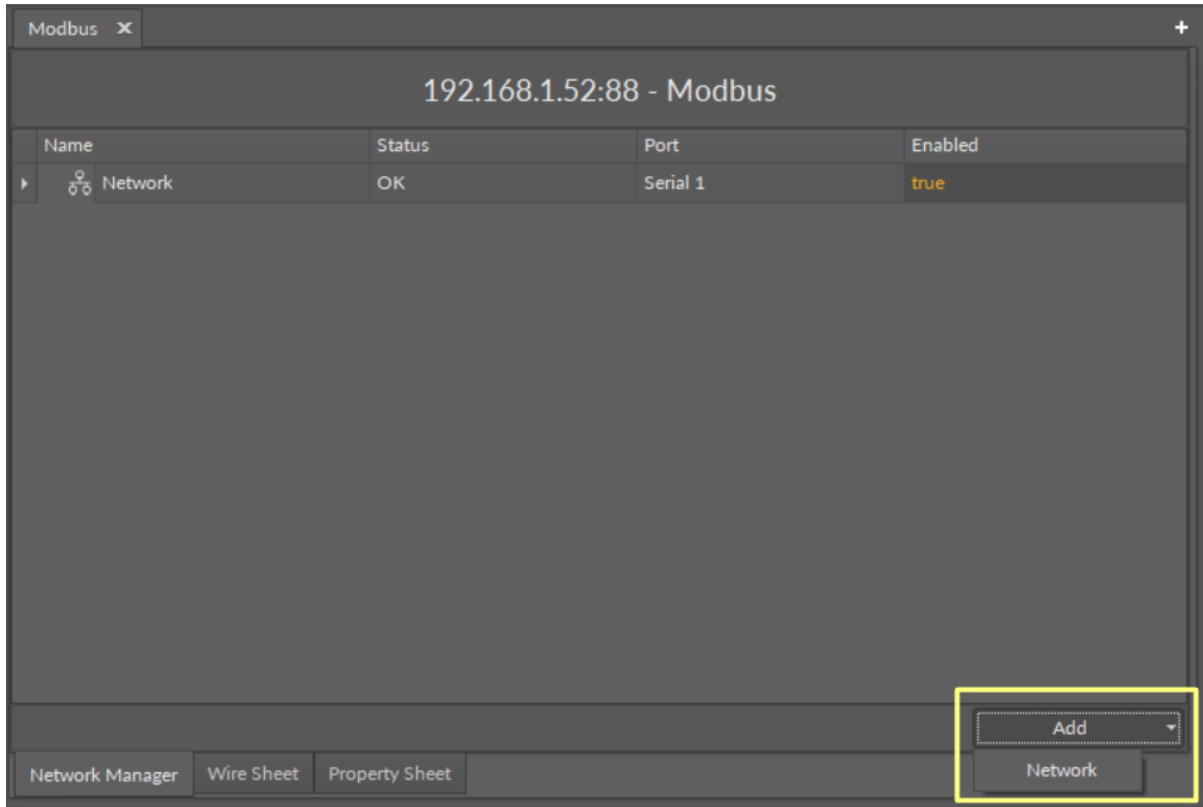

Figure 48. Adding the Modbus network

Using this Add button opens the dialog window, which allows to adjust the quantity of networks to be added.
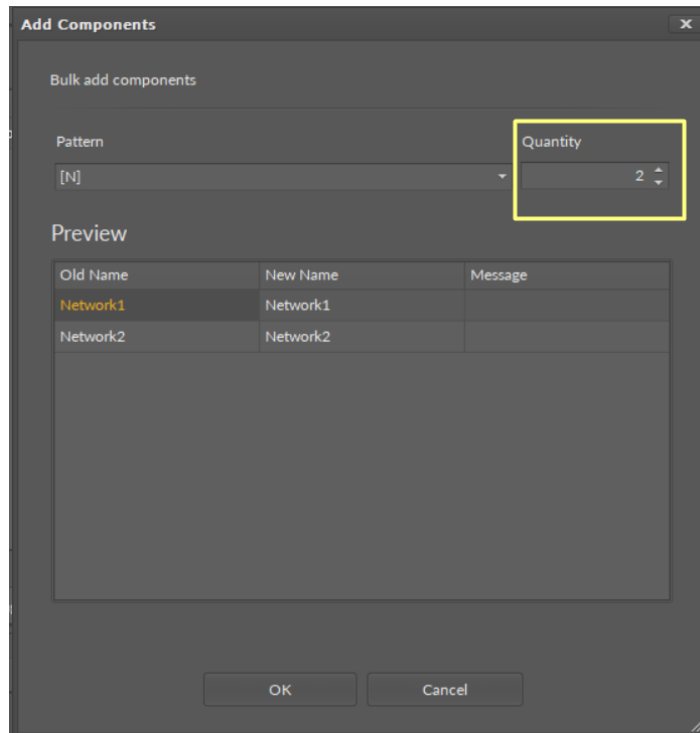
Figure 49. The dialog window

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.

6.8.2 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.
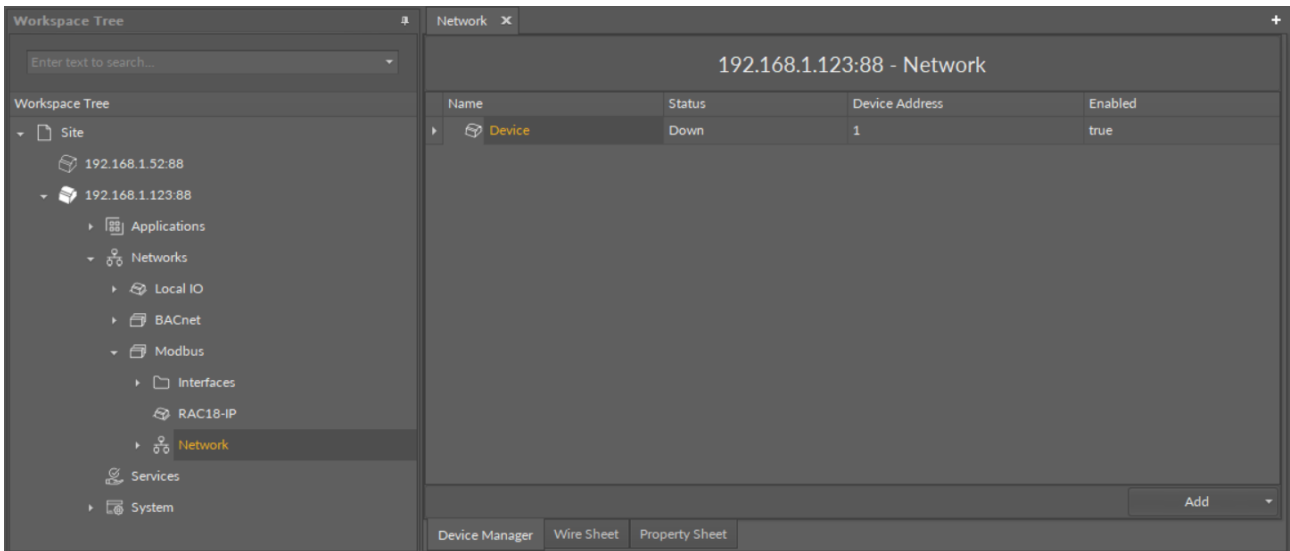
Figure 50. The Device Manager view

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.
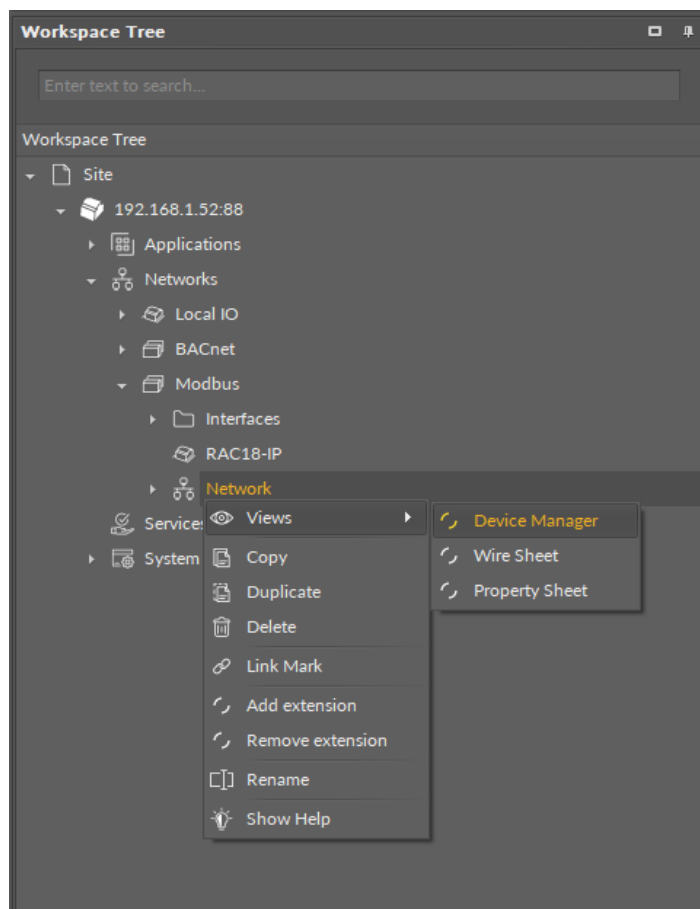

Figure 51. Opening the Device Manager

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.
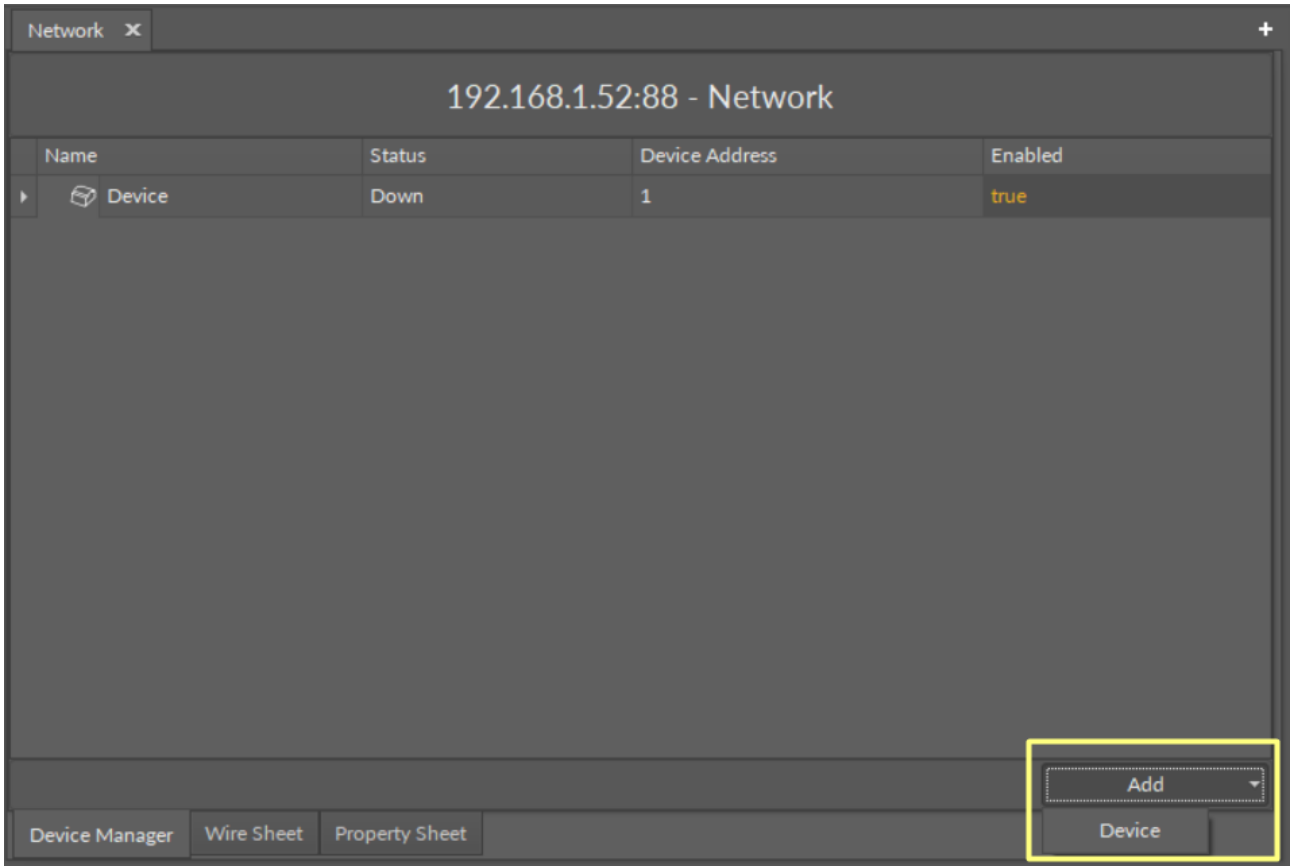

Figure 52. Adding Modbus devices

Using this Add button opens the dialog window, which allows to adjust the quantity of devices to be added.
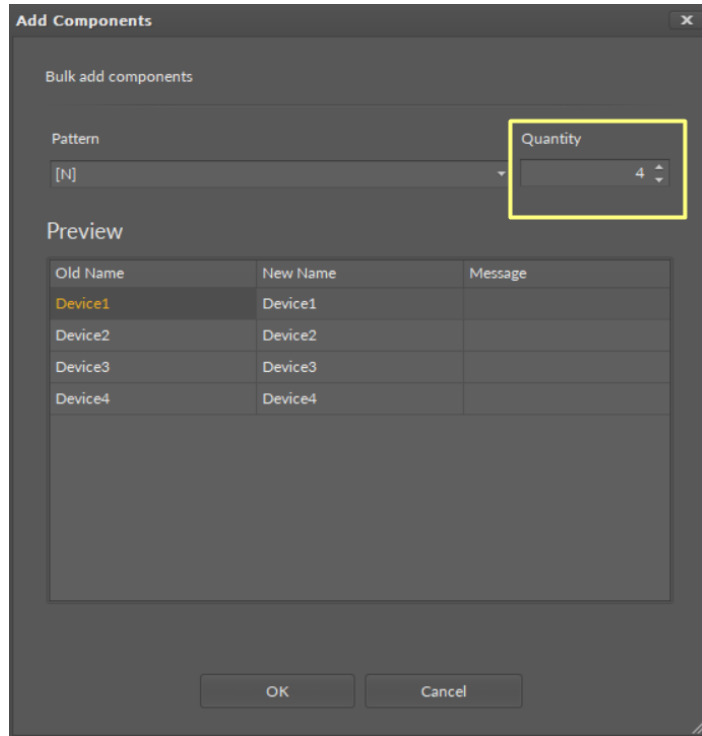
Figure 53. The dialog window

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.

6.8.3 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.
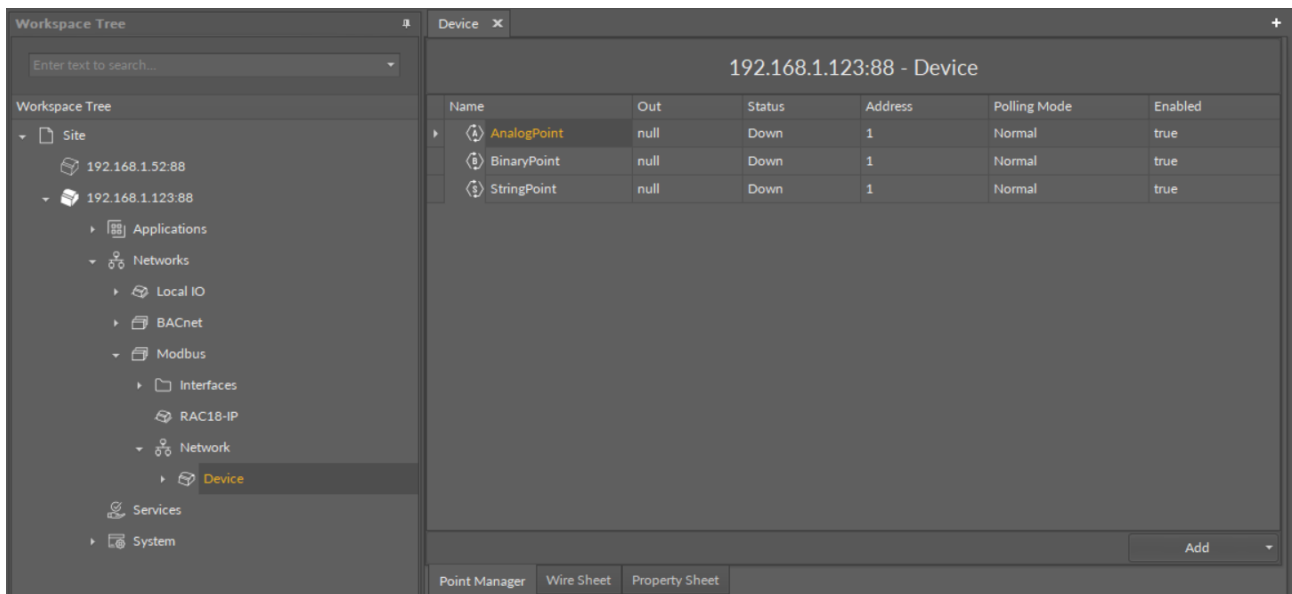

Figure 54. The Point Manager

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.
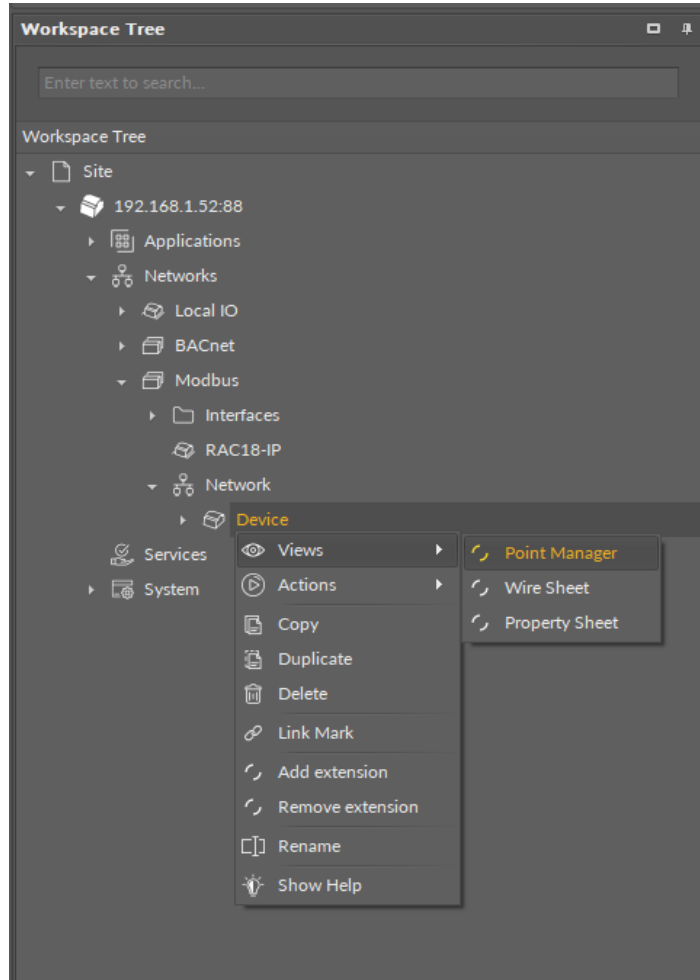

Figure 55. Opening the Point Manager

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.
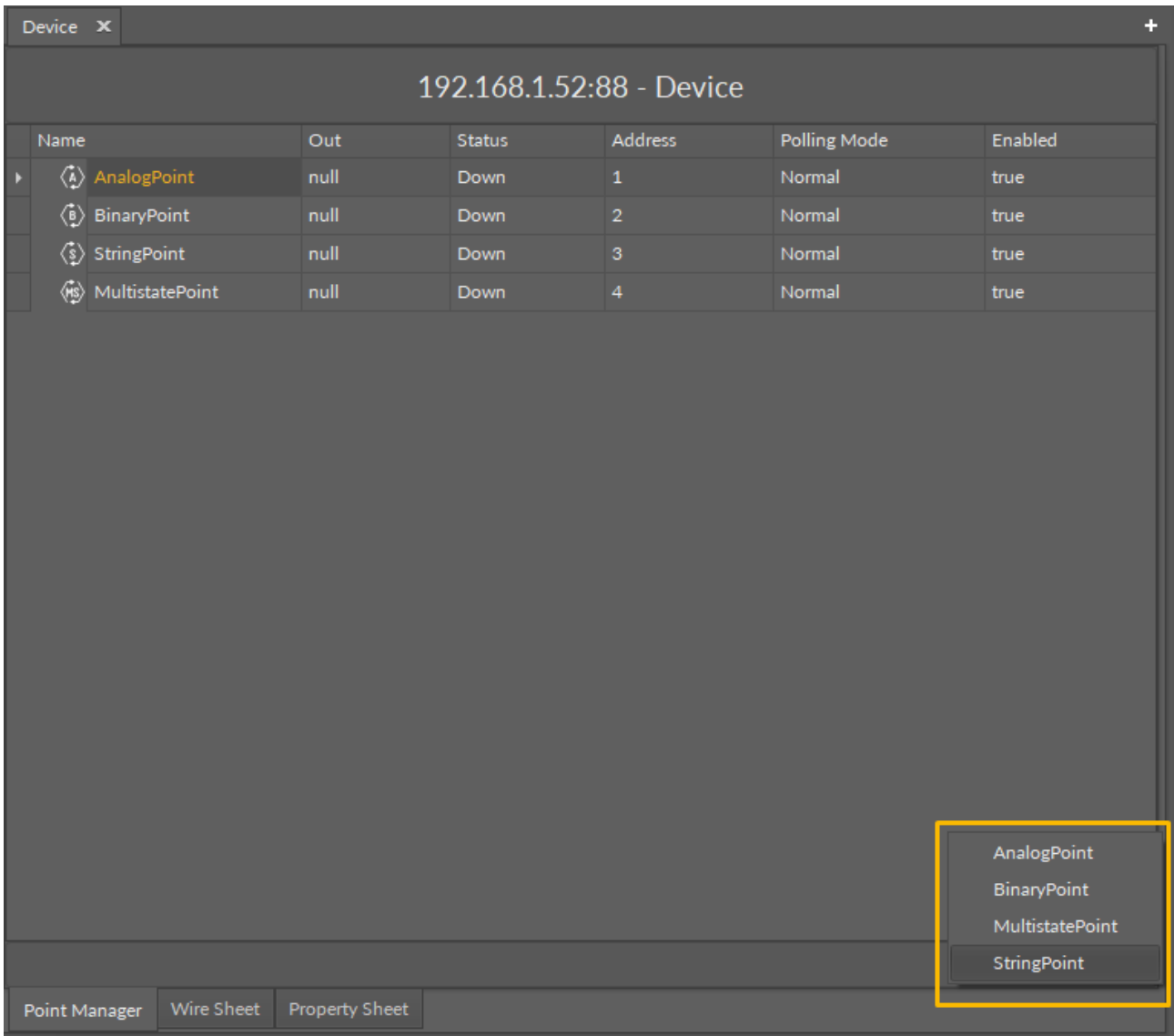
Figure 56. The Add button

Using this Add button opens the dialog window, which allows to adjust the quantity of Modbus Points to be added.
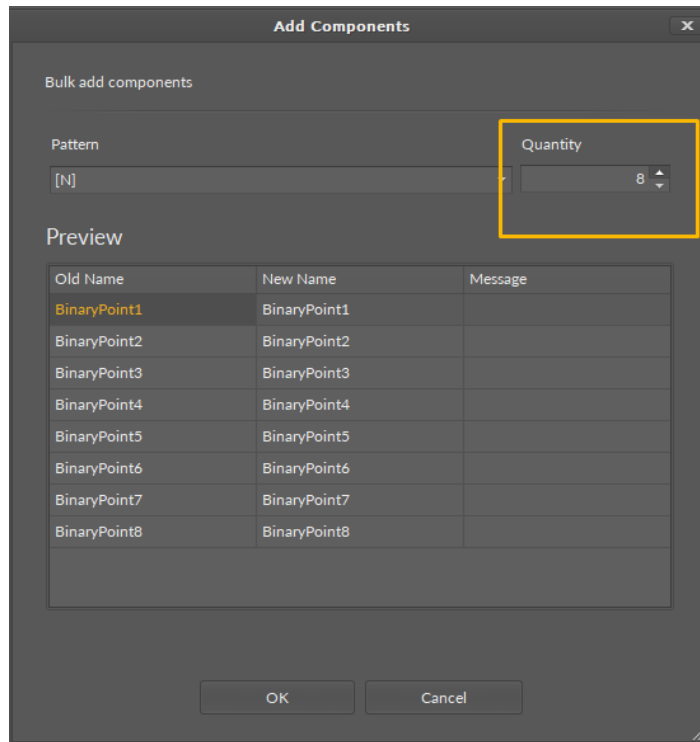
Figure 57. The dialog window

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.

7 Services Container

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.
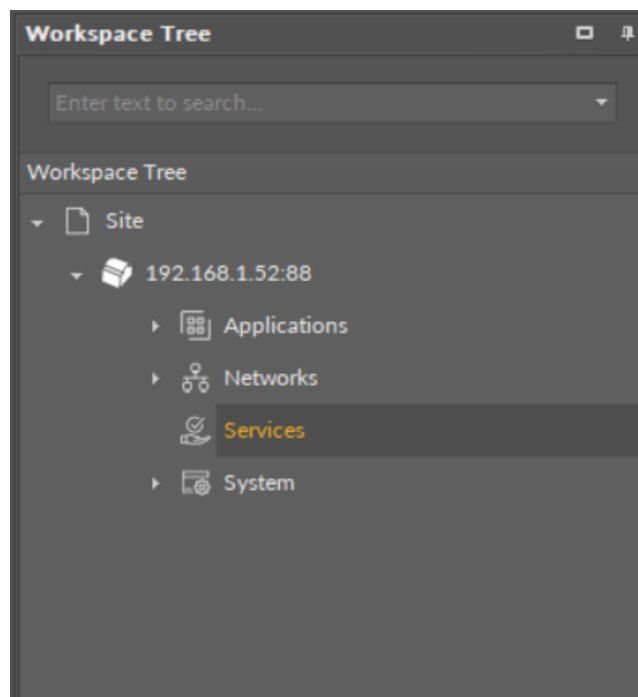

Figure 58. The Services container

8 System Container

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.
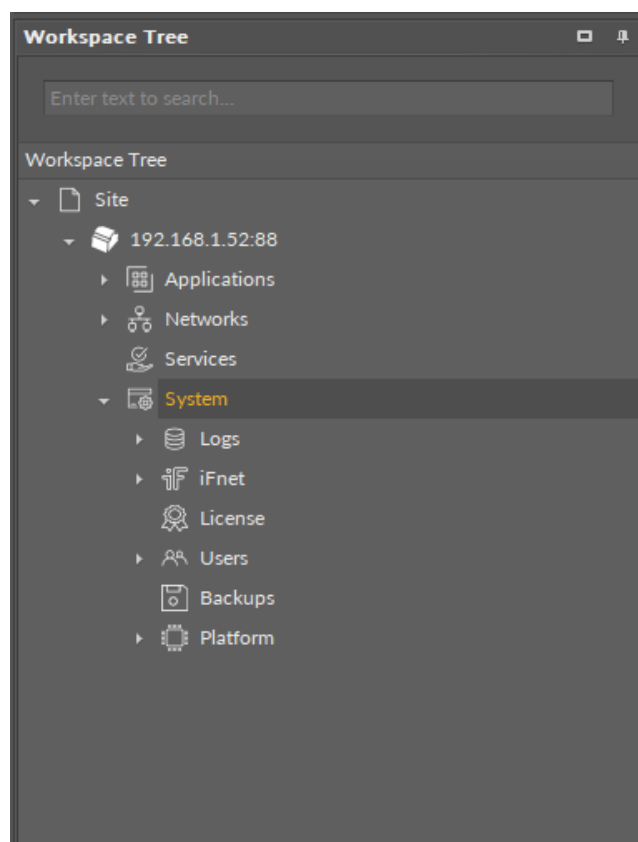

Figure 59. The System container

WARNING!

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

9 RAC18-IP Quick Start-up

Start working with your RAC18-IP device in just 8 easy steps:

Step 1: Unbox the device.

Step 2: Connect power supply. The RAC18-IP device requires 24 V AC/DC power supply.

Step 3: Connect the device to the network. The RAC18-IP device is equipped with the Ethernet IP fail-safe protected ports, RS485 to connect with other devices in the network, and RJ45 to connect with external panels.

Step 4: Make sure to download the iC Tool software.

Step 5: Open the iC Tool, and 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). In this window, the type of device cannot be changed (it is the nano EDGE ENGINE type). 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.

Step 6: Connect with the device. Once the device is added, the Authentication dialog window pops up. The default credentials are:

- username: admin
- password: admin

> **Worth to Notice:**
>
> When logging in to a factory-set or reset device, it is required to change password. Please refer to First Logging In.

Step 7: Go to the System container in the Workspace 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 8: The device is now ready to work. Start creating applications: go to the Applications container, and add as many Application components as there are user applications to be created. Then go to the Device Libraries, and drag the needed components to the Application components. The Core library is the one containing the Application component itself, licensed Data Points, and Equipment component, which helps to organize the contents in the Applications container.