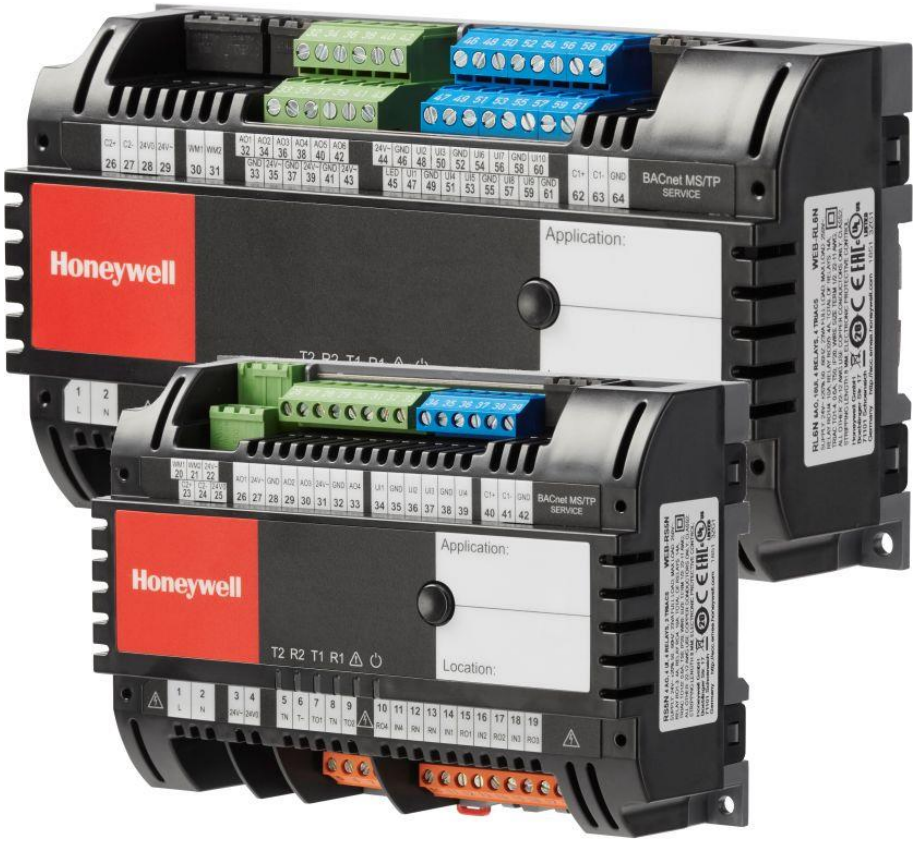
Spyder Model 5
ENGINEERING TOOL

Honeywell

USER GUIDE



IMPORTANT NOTE: Email your Host Id to Honeywell WEBs Customer Care (WEBsLicense@honeywell.com), so we can move the license to your organization. For additional queries contact to the distributor.

31-00282-01

TABLE OF CONTENTS

TABLE OF CONTENTS **2**

ABOUT THIS USER GUIDE **4**

Applicable Technical Literature 4

SYSTEM REQUIREMENTS **4**

Restrictions and Recommendations 4
Security Best Practices 4
Security Check List *5*

INTRODUCTION **6**

Basic Concepts 6
System Architecture 9

PREREQUISITES **10**

Adding IRM Application Template to Palette 10

SETTING UP CIPER MODEL 50 CONTROLLER **14**

Creating CIPer Model 50 Station *16*
Configuring CIPer Model 50 Controller as BACNET IP - MSTP Router *17*

ENGINEERING MODES **22**

Offline Engineering 22
Creating IRM BACnet Device *25*
Matching Devices *28*
Online Engineering 32
Synchronicity Check via Control Manager 35
Factory Device Handling 37
Detailed Offline Procedure 39
Detailed Online Procedure 43

WORKING WITH APPLICATIONS – THE IRM PROGRAM **45**

Preparing Niagara Work Environment 45
IRM Program Components 48
The Control Manager Procedure *50*
Periodic Program *53*
Event Program *54*
Onboard IO *55*
Alarms *58*
Application Engineering Guidelines 59
Commissioning and Station Copier Usage *59*
Final Step after Application Engineering *59*
Memory and Folder Usage *59*
IRM Function Blocks and External Application Components *60*
Application Templates Usage *60*
Reference Datapoints Usage 60
Manual Creation and Mapping of Reference Points (A) *65*
Automatic Creation and Mapping of Reference Points via Drag & drop (B) *76*
Application Management 82
Teaching and Learning *82*
Teaching to Controller *82*
Learning from Controller *83*
Clearing Controller *85*
Clearing Project *85*
Checking Hardware Compatibility *86*

Taking Snapshot *88*
Restoring Snapshot *89*
Swapping IRM Program *90*
MASTER SYNC 93
Cloning Application *94*
Applying Master Sync *95*
Excluding Function Block Items from Master Sync *97*
SPLITTING-OFF APPLICATION 99
CONTROLLER MAC ADDRESS ASSIGNMENT 102
Setting MAC Address of Controller *104*
SYNCHRONIZING DEVICE TIME 106

FIRMWARE DOWLOAD 107

ALARMING 108

MISCELLANEOUS 108

VALUE UPDATES AFTER DEVICE POWER FAILURE 108
HIDING SLOTS 108
SYLK WALL MODULE USAGE 110

ABOUT THIS USER GUIDE

This user guide describes the configuration and management of Spyder Model 5 controllers connected to a BACnet MSTP network via a CIPer Model 50 controller. Configuration and Management is done by using the Spyder Model 5 Engineering Tool based on the Honeywell WEBStation.

Applicable Technical Literature

- Spyder Model 5 Product Data: 31-00280ES-01
- Spyder Model 5 Installation Guide: 31-00281ES
- IRM-NX FUNCTION BLOCKS User Guide: EN2B-0415GE51
- IRM-NX APPLICATION Guide: EN2B-0416GE51
- Honeywell General Security Best Practices – System Engineering Guide: 31-00129
- CIPer Model 30 Hardening Guide: 31-00207EFS
- CIPer Model 50 Product Datasheet: 31-00197
- CIPer Model 50 Installation & Commissioning Instructions: 31-00233EFS
- CIPer Model 50 Mounting Instructions: 31-00234EFS
- CIPer Model 50 Controller User Guide: 31-00198

SYSTEM REQUIREMENTS

Niagara	:	Honeywell WEBStation N4.4.94.xx and higher
Firmware	:	IRMN4-IMG_V0.0.7.1.bin
Tools	:	Spyder5_Tool_1.0.0.37
DemoApplication	:	IRMN_H_0001_Ver_1.0.1.5
Firmware and Software Downloads	:	The firmware and the software can be downloaded from The Honeywell Buildings Forum

Restrictions and Recommendations

For successful and seamless engineering, it is recommended to note the following internal system restrictions:

Number of BACnet Devices	:	max. 64 devices per channel. Depending on performance needs of the application and bus traffic it is recommended to keep the number of devices below 64.
Controller Memory Usage	:	max. 80 % (recommended 70 %)
Function Blocks Usage	:	- max. 32 IRM folders overall - max. 100 function blocks per folder - max. 2000 function blocks overall - max 1 wall module per device
Baud rate	:	9600 through max. 76800 (default = 38400)

Security Best Practices

This section provides the necessary information about the requirements for configuring and managing the security when installing and maintaining a product or system.

Honeywell hereby expressly states that its controllers are not inherently protected against cyber-attacks from the Internet and that they are therefore intended solely for use in private networks. However, even private networks can still be subject to malicious cyber-attacks by skilled and equipped IT individuals and thus require protection. Customers should therefore adopt the installation and security best practices guidelines for Honeywell BACnet MS/TP-based products to mitigate the risk posed by such attacks.

The following check list describe the General Security Best Practices for Honeywell BACnet MS/TP-based products. They are listed in order of increasing mitigation. The exact requirements of each site should be assessed on a case-by-case basis. The vast majority of installations implementing all of the mitigation levels described here will be far in excess of that required for satisfactory system security.

Incorporating the security check list items 1-5 will generally meet the requirements for most automation control network installations.

Additional information can be obtained from:

- Honeywell General Security Best Practices – System Engineering Guide (31-00129)
- CIPer Model 30 Hardening Guide (31-00207-01)

Security Check List

1. Use the latest version of IRM software including firmware and software modules.
2. Include the WEBStation N4.x installation files, configuration files (including station backup), certificates and licenses in the disaster recovery plan.
3. Make sure that the PC running WEBStation N4.x, where possible, is secured against unauthorized physical access.
4. Make sure that the local ethernet network that the PC is connected to is secured, e.g. by the use of firewalls and intrusion detection systems.
5. The PC is running the latest version of the Windows operating system, with all updates and service packs.
6. The PC is running virus protection software.
7. Appropriate user accounts are set up on PC and access to files is restricted to only those who are authorized.
8. WEBStation N4.x is configured to use HTTPS using a certificate from a trusted Certificate Authority.
9. WEBStation N4.x users are configured as required.
10. WEBStation N4.x is configured to backup data regularly to a secure location as per your company's backup policy.
11. Ensure that complete commissioning is carried out in a closed local area network without connecting to the internet to avoid unauthorized sniffing of BACnet message packets.

INTRODUCTION

The Spyder Model 5 Engineering Tool provides the following functions:

- Setting up the IRM and CIPer Model 50 controllers for usage in a BACnet MSTP system architecture
- Creating BACnet devices offline and online
- Creating applications for IRM controllers
- Synchronizing applications between project and controllers
- Adjusting terminal layout deviations (hardware compatibility)
- Creating Master Sync groups
- Splitting applications
- Cloning applications
- Firmware download
- Alarming
- Printout

Basic Concepts

Engineering Modes

There are two kinds of engineering you can use for engineering an IRM project:

- Offline Engineering
  In this mode, you create an empty BACnet device manually, add an application (optional) and match it afterwards to a device discovered on the bus by using the service pin. This is normally applied when doing the engineering in the office without having the hardware available but knowing the hardware specification of the devices to be used later at the plant.

- Online Engineering
  In this mode, you discover the devices on the BACnet network in the first step and use the devices instantly for application engineering. This is recommended when doing the engineering directly at the plant with the devices already installed on the BACnet bus.

Synchronization Status of Application

The current content of the applications engineered in the project and running in the connected IRM controller is permanently monitored by the control manager (Online engineering only). Modifications can be any of the following:

- Control strategy icons (function blocks)
- Links between function blocks
- Notes
- Annotations
- Author
- Description
- Terminals
- Notification classes
- Etc.

Modifications can be detected in the controller, or in the project, or in both.
They can be synchronized by applying the following actions:

- Teach to controller
- Learn from controller
- Clear project (as required)
- Clear controller (as required)

Teaching and Learning

Synchronization of the application in the project and controller application can be performed in two ways:

- Teaching to Controller
  Downloads the application in the project to the controller
- Learning from Controller
  Uploads the application from the controller in the project

Teaching Modes

Any modifications in either the project or the controller can be synchronized in one of the following modes:

- on-demand will be performed explicitly as desired by the user
- immediate occurs instantly when the change is done

Clearing Controller / Project

The application in the project and in the controller, can be cleared independently.

IRM Program

The IRM program includes the control manager, the control strategy (periodic and event programs), the hardware configuration (on board I/O) and the alarming.

Control Manager

The control manager takes care of the major control functions and displays the current status of the IRM program accordingly. The following information is provided:

- Author of the application
- Description
- Application type
- Function block family, version, and numbers
- Number of folders and links
- Memory usage
- Hardware features and compatibility
- Controller connection type
- Engineering units (measurement) type
- Drop of BACnet point settings
- Communication status
- Synchronization status
- Last program change and commissioning dates

Periodic program

Includes the control strategy running in a cyclic manner on a fixed time base

Event program

Includes event-driven control strategy triggered by particular IO changes

On board I/O

Shows the hardware I/Os of the controller

Alarms

Provides the notification classes for establishing alarming

Hardware Compatibility Check

The hardware configurations of the used physical controller and the hardware defined for the controller in the application can be checked. Any differences are indicated graphically on the terminals on the wire sheet.

Taking / Restoring Snapshot

The current status of an application can be backed up and restored later. This allows restoring a changed application if these changes should be discarded.

Swapping IRM Program

Swapping saves RAM space and reduces processor load and bus traffic. The current state of a swapped device is frozen and saved to an IRM repository on the disk. Then, synchronization is no more possible. In order to synchronize swapped-out devices, the devices must be swapped in again. Swapping can be applied to a single or to all devices per step.

Spyder Model 5 Engineering Tool

The Spyder Model 5 Engineering Tool provides the work environment for engineering the controllers.

IRM Operations Monitor / Jobs Sidebar

The IRM Operations Monitor shows all actions in a popup window and all actions are summarized in a list in the *Jobs* window.

IRM Palette

The following palette is available for creating the application:

- honIrmControl - provides control function blocks and templates for IRM BACnet devices, IRM programs, and folders.
- honIrmAppl - provides standard FCU application with all commonly used functionalities.

System Architecture

The following schematic shows an example for a BACnet MSTP based system containing a CIPer Model 50 controller as a router and 3 IRM (Spyder Model 5) controllers for room control. The system is engineered using the Spyder Model 5 Engineering Tool based on the WEBStation N4.4 or higher.

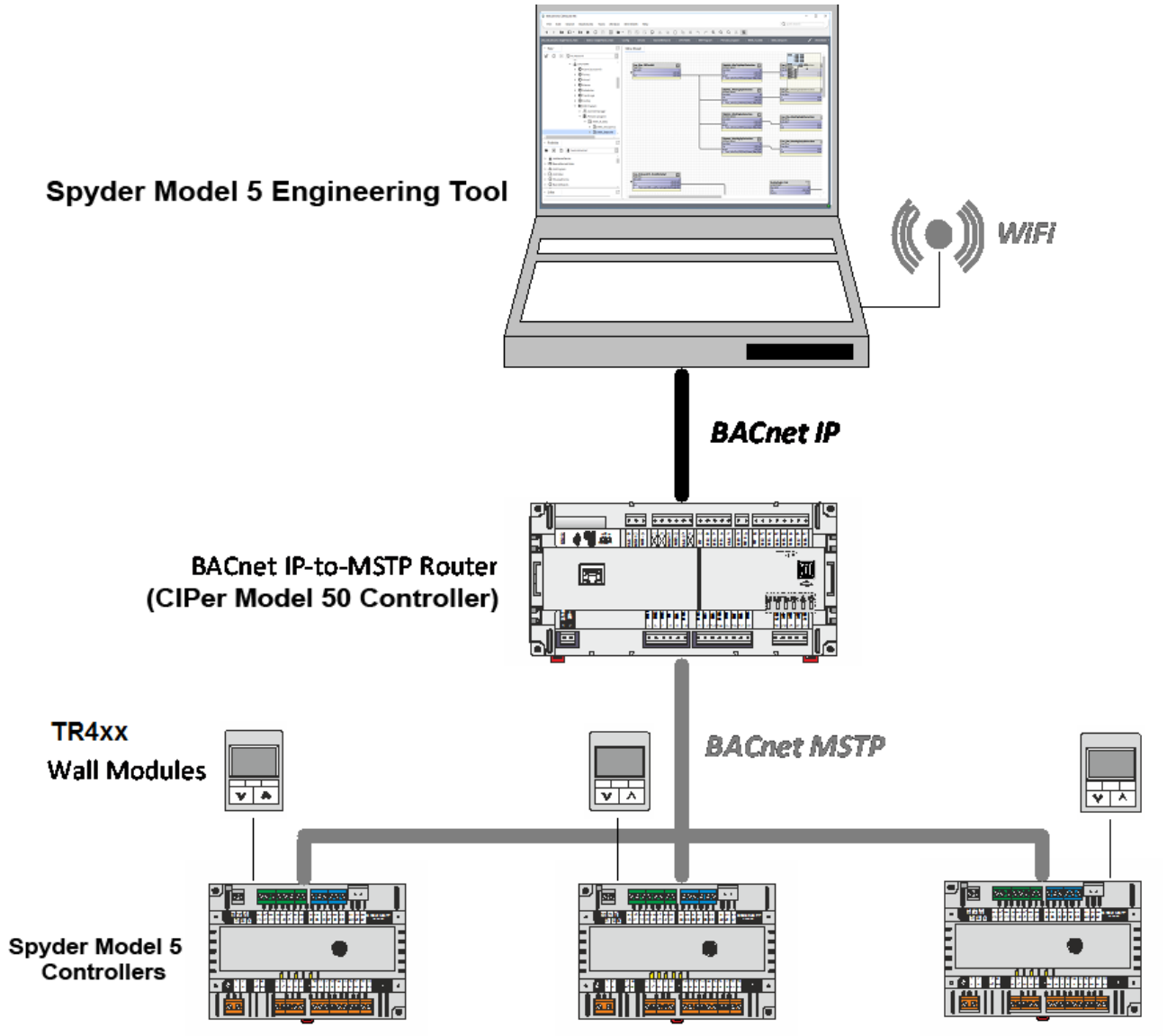


Fig. 1. System Architecture based on Spyder Model 5 Engineering Tool, Spyder Model 5 and CIPer Model 50 Controllers

PREREQUISITES

It is assumed that you are familiar with basic Niagara techniques and functions, such as creating platforms, stations, and networks etc.

Make sure that the Spyder Model 5 and CIPer Model 50 controllers are properly connected (see Spyder Model 5 Installation Guide, 31-00281ES and CIPer Model 50 Installation & Commissioning Instructions, 31-00233EFS.

Make sure that the following steps were done prior of working with the Spyder Model 5 Engineering Tool.

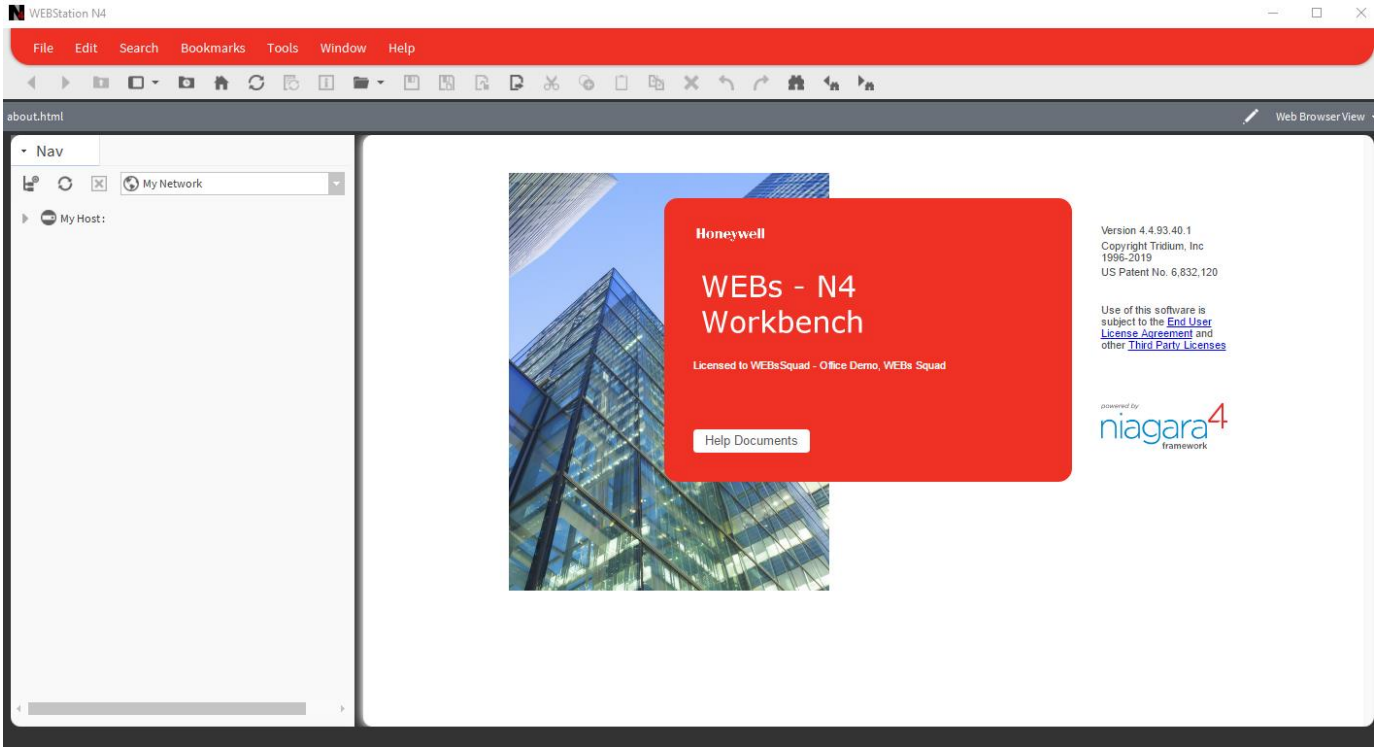
If not already available in the current WEBs N4 installation, copy the following files to the *Modules* folder:

- honIrmAppl-rt.jar
- honIrmAppl-rt.jar.sig
- honIrmConfig-rt.jar
- honIrmConfig-rt.jar.sig
- honIrmConfig-wb.jar
- honIrmConfig-wb.jar.sig
- honIrmControl-rt.jar
- honIrmControl-rt.jar.sig

Adding IRM Application Template to Palette

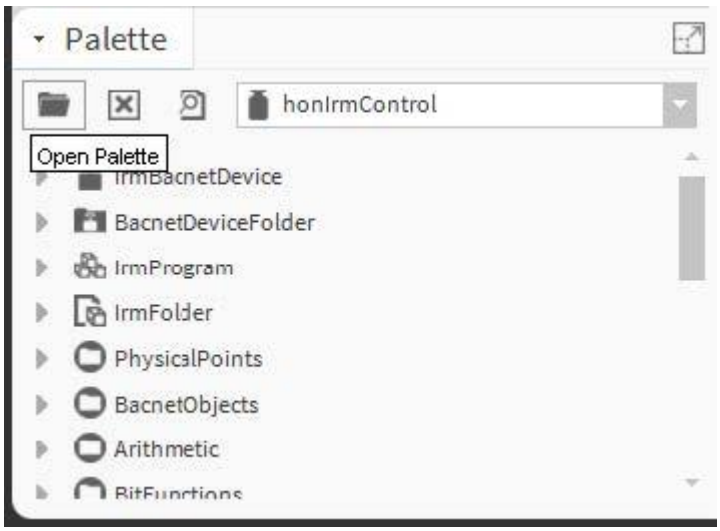
Procedure

1. Make sure that the ´honIrmAppl.jar` file is installed in the *Modules* folder of the Spyder Model 5 Engineering Tool installation.
2. Open the Spyder Model 5 Engineering Tool.

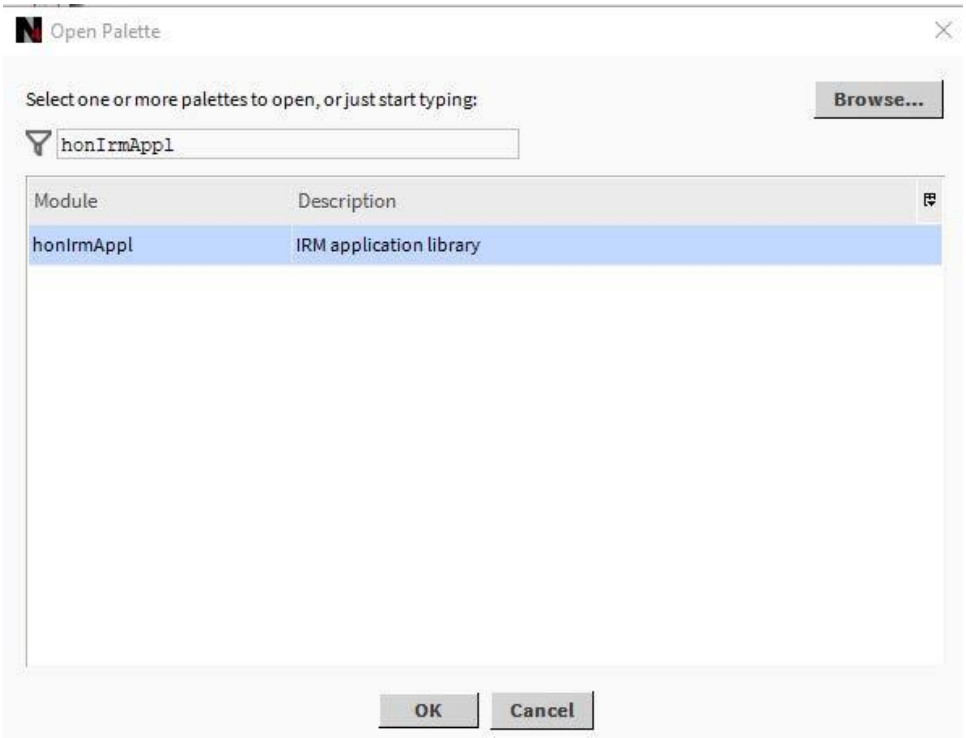


3. On the platform and connect to the station.

4. On the *Palette* pane, click **Open Palette** icon.

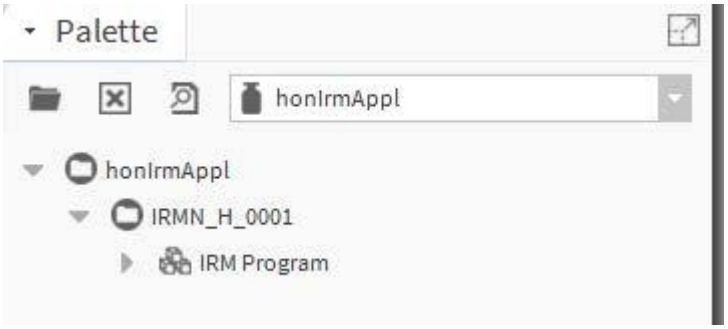


5. In the *Open Palette* dialog box, enter ´honIrmAppl.jar`.

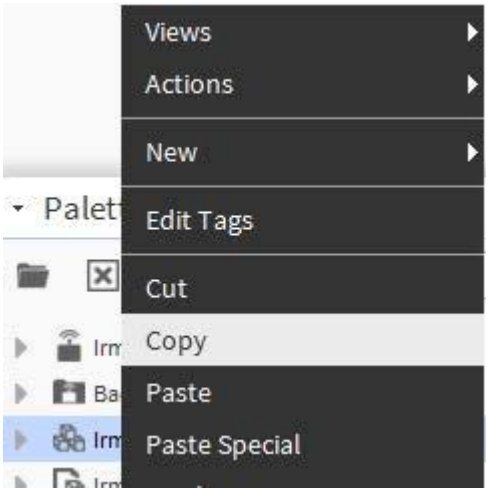


6. If not already selected in the list, select ´honIrmAppl.jar`, and then click **OK**.

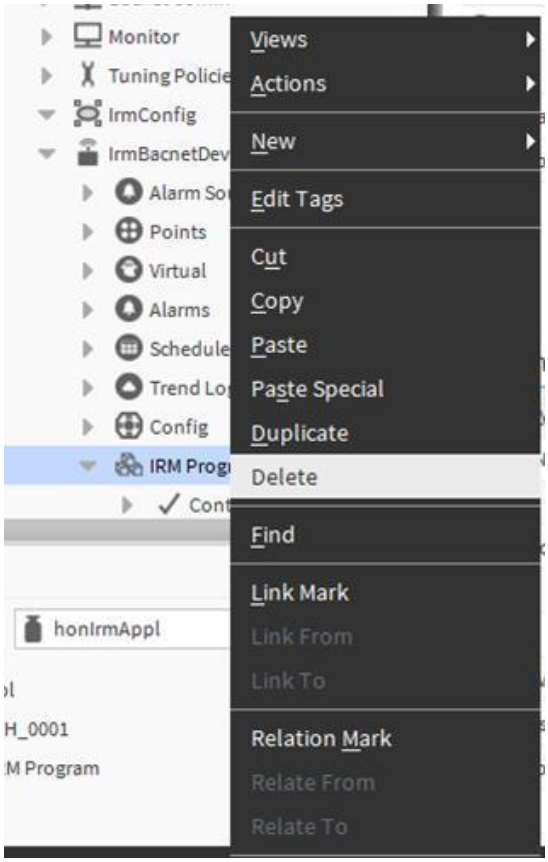
7. In the *Palette* pane, expand the *honIrmAppl* folder.



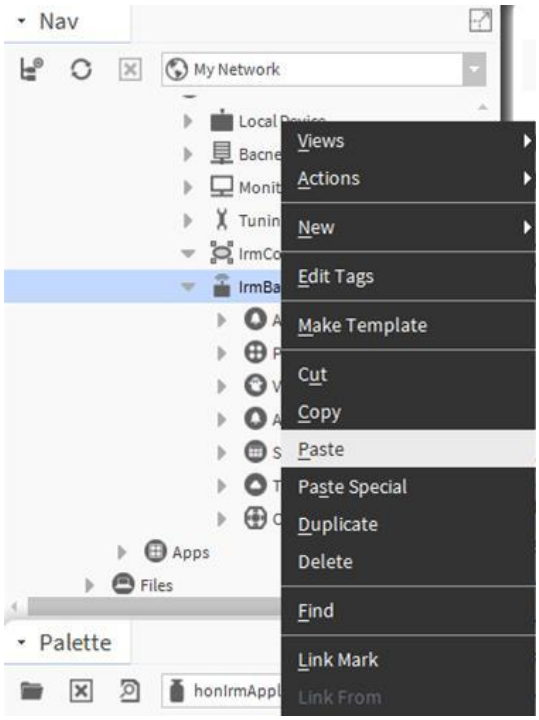
8. Right-click IRM Program and click **Copy** in the context menu.



9. In the *Nav* tree, expand the IRM device and delete the existing IRM Program.



10. Right-click the IRM device and click **Paste** in the context menu.



SETTING UP CIPer Model 50 CONTROLLER

In order to access the IRM controllers via Spyder Model 5 Engineering Tool or supervisor, a BACNET IP - MSTP router must be implemented. For this purpose, it is recommended to use the CIPer Model 50 NX controller which it can host and run a station for IRM engineering in parallel.

NOTE: The CIPer Model 50 controller can also be used as a router. But it can only be used as a router since it cannot host a station for IRM engineering.

The CIPer Model 50 controller is setup using standard Niagara techniques and functions such as creating platforms, stations, and networks. Hence, only the specialties for the CIPer Model 5, steps 2 and 5 are described in the following.

NOTE: For detailed information on standard Niagara techniques and functions, please refer to the Niagara online documentation and/or the CIPer Model 50 Controller User Guide: 31-00198-01.

Setting up the CIPer Model 50 controller as router and host for an IRM station (optional), includes the following main steps:

Offline Engineering

1. Open platform (PC)
2. Create CIPer Model 50 station
3. Start and connect to CIPer Model 50 station
4. Create BACnet Network
5. Configure CIPer Model 50 as BACNET IP - MSTP Router

Online Engineering

1. Commission Controller
   (see figure next page).

For detailed information on offline and online engineering, please refer to the “Engineering modes” section, pg. 22.

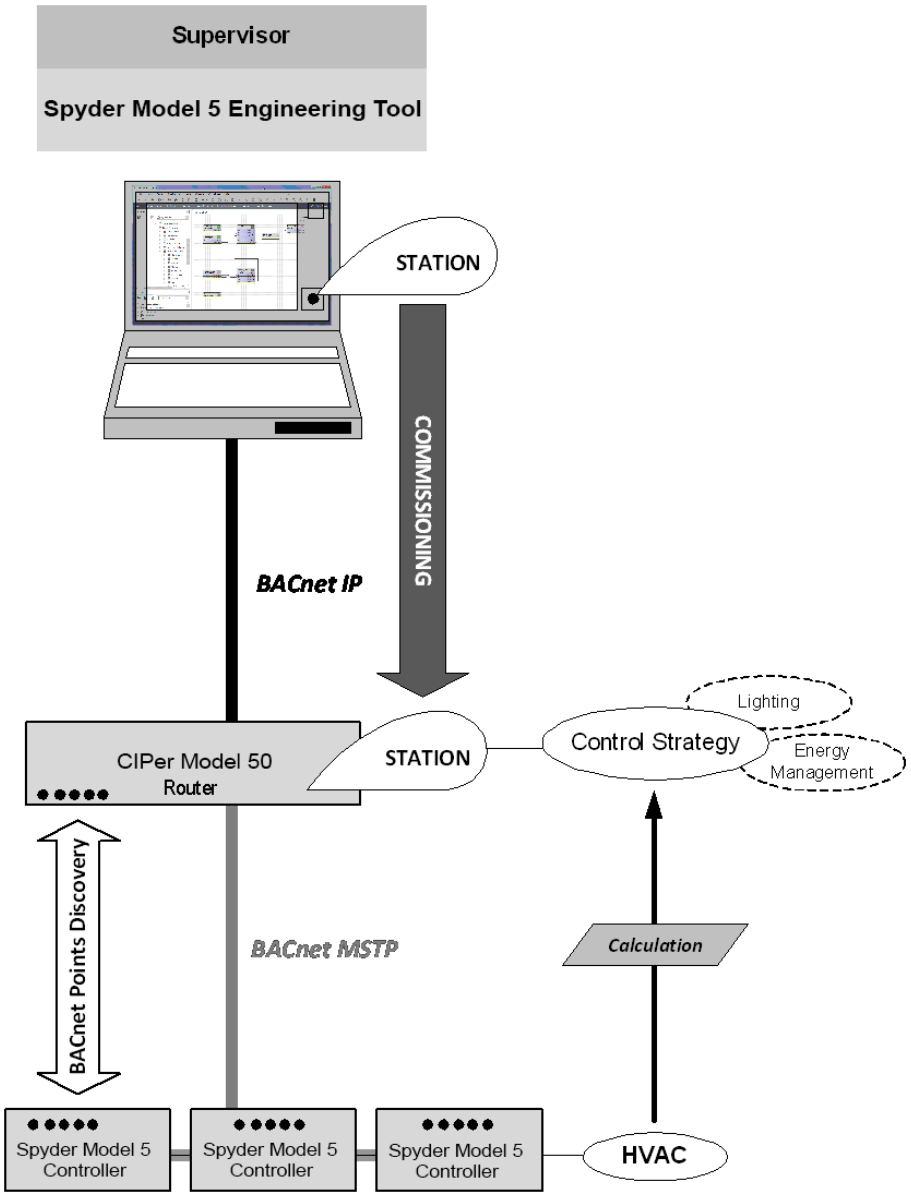


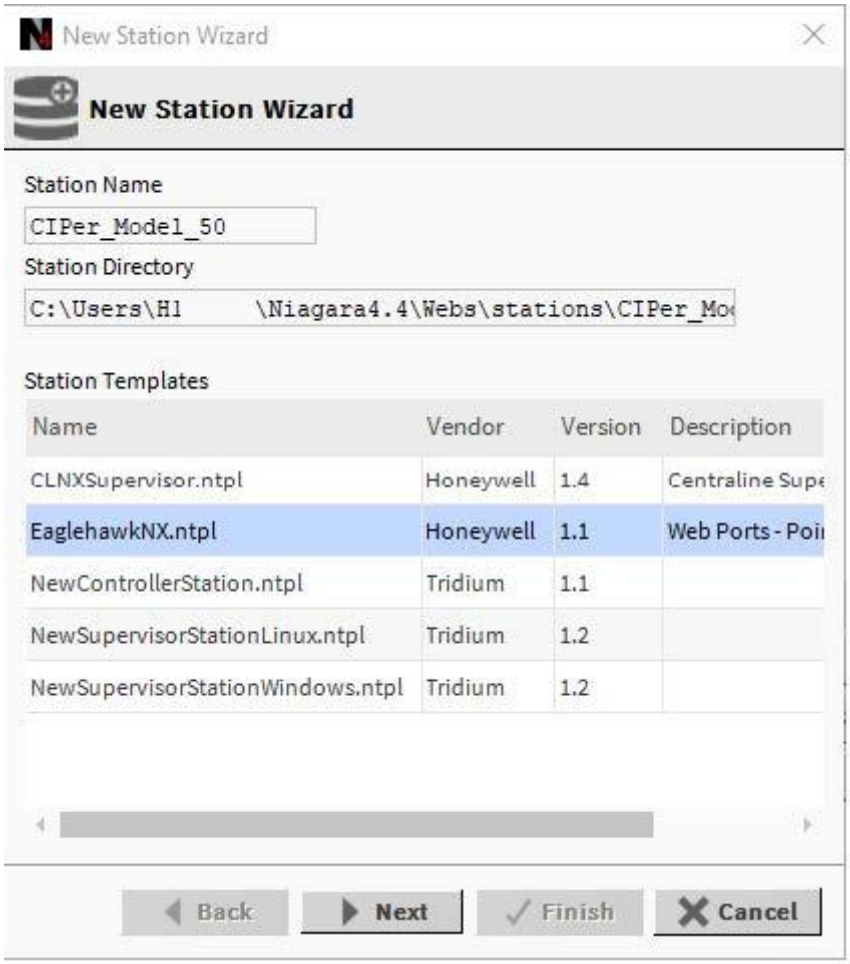
Fig. 2. Basic Engineering for Typical CIPer Model 50 and Spyder Model 5 Application Scenario

Creating CIPer Model 50 Station

For detailed information on standard Niagara techniques and functions, please refer to the Niagara online documentation and/or the CIPer Model 50 Controller User Guide: 31-00198-01.

Procedure

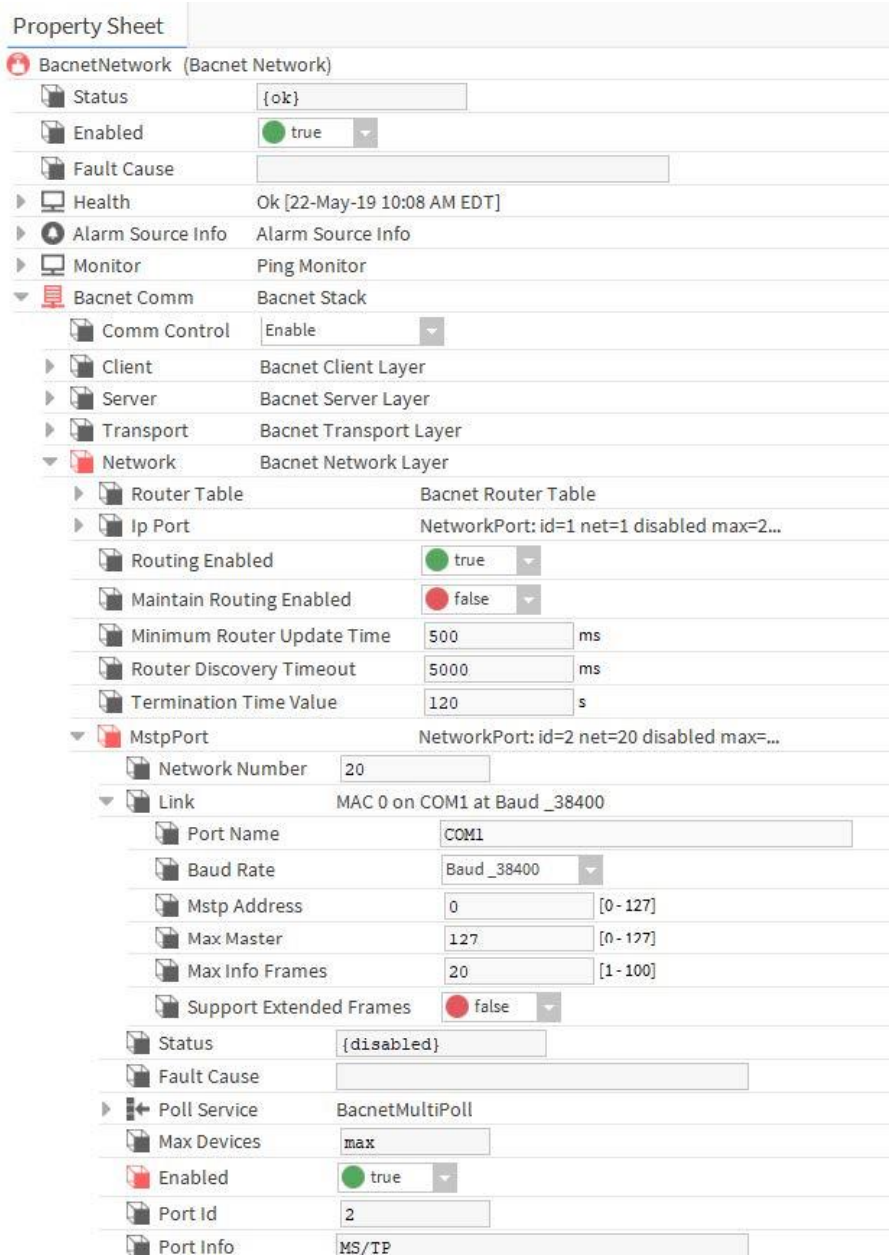
1. Open the platform on the PC.
2. Start creating the station using the New Station Wizard.
3. In the *New Station Wizard* dialog, select the ´EaglehawkNX.ntpl` template.



4. Continue with creating the station (standard Niagara procedure).

Configuring CIPer Model 50 Controller as BACNET IP - MSTP Router

In order to access the Spyder model 5 controllers via Spyder Model 5 Engineering Tool or supervisor, a BACnet IP – MSTP router must be implemented. For this, it is recommended to use the CIPer Model 50 controller which can host and run a station for Spyder Model 5 engineering in parallel.



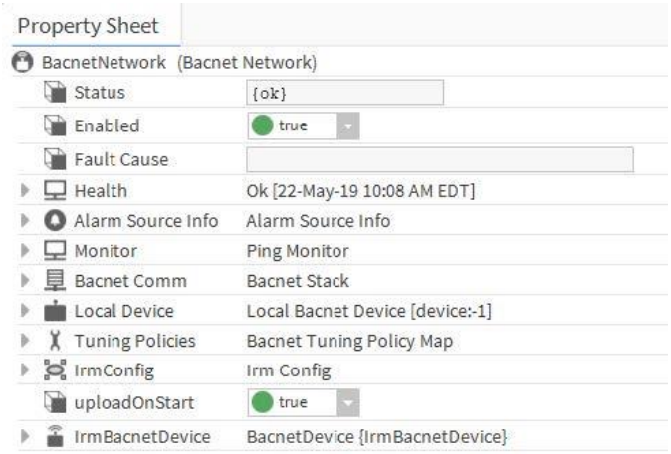
NOTE: For detailed information on standard Niagara techniques and functions, please refer to the Niagara online documentation and/or the CIPer Model 50 Controller User Guide: 31-00198-01.

Prerequisite Steps

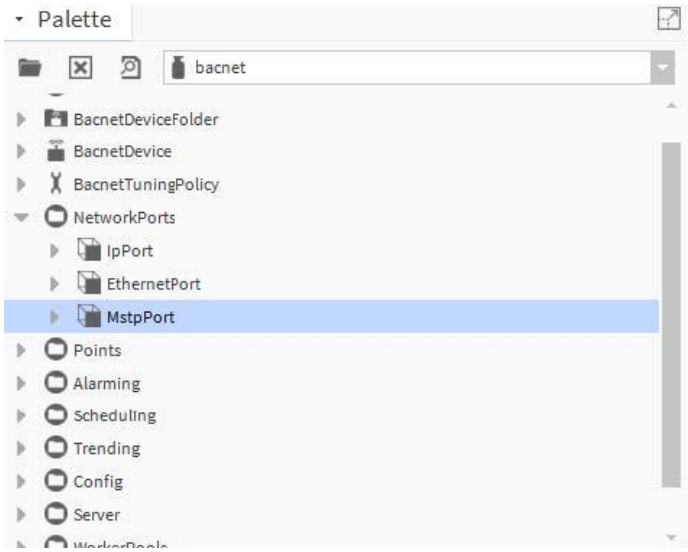
1. Open the platform (PC).
2. Create CIPer Model 50 station
3. Start and connect to CIPer Model 50 station

Procedure

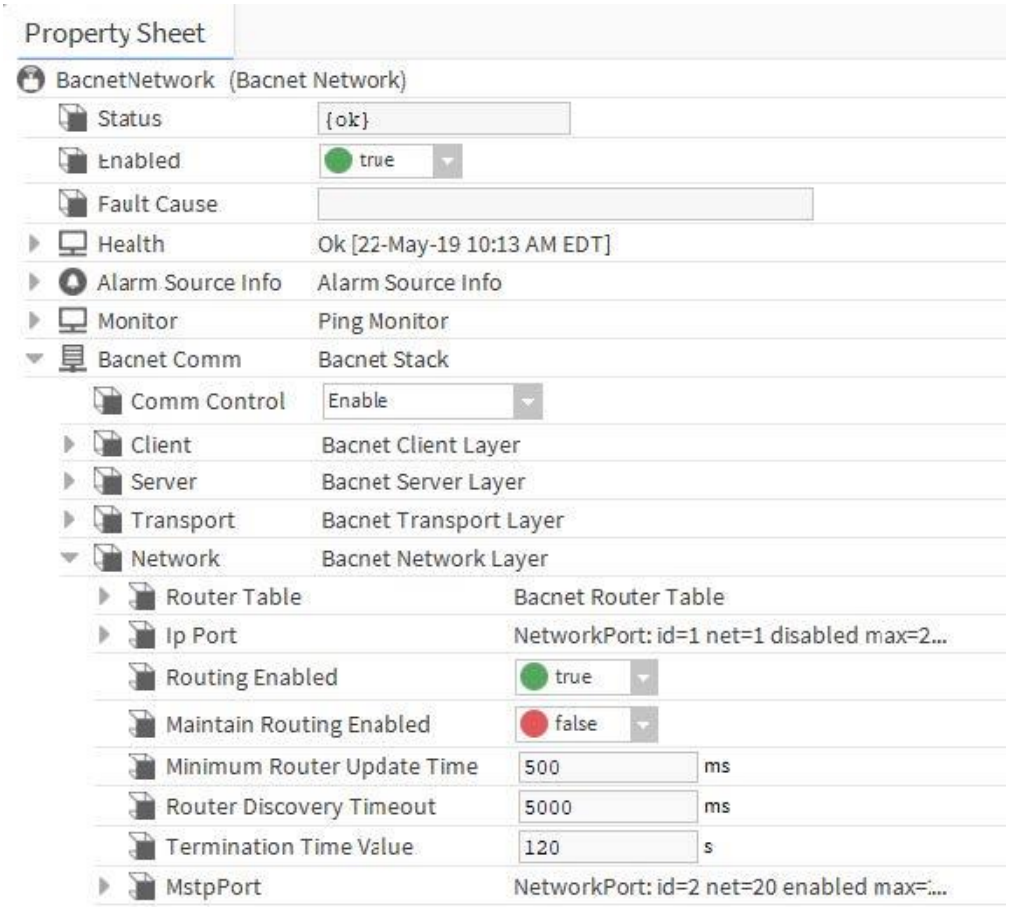
1. Create BACnet Network
2. Select the *Property Sheet* of the BACnet network.



3. Open the *BACnet* palette and expand *NetworkPorts*.

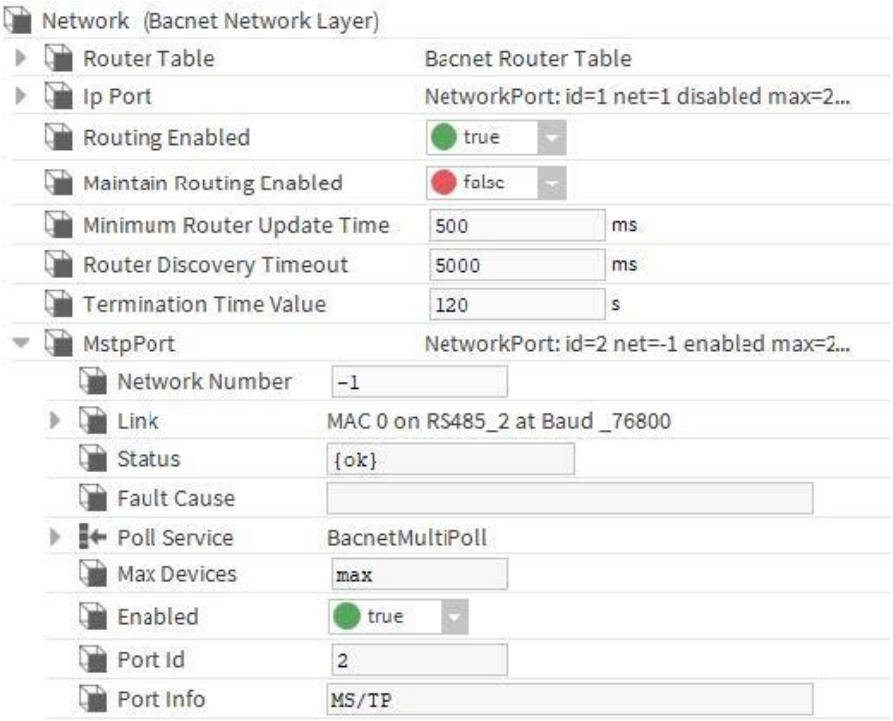


4. Expand **BACnet Comm** and **Network** on the *Property Sheet*



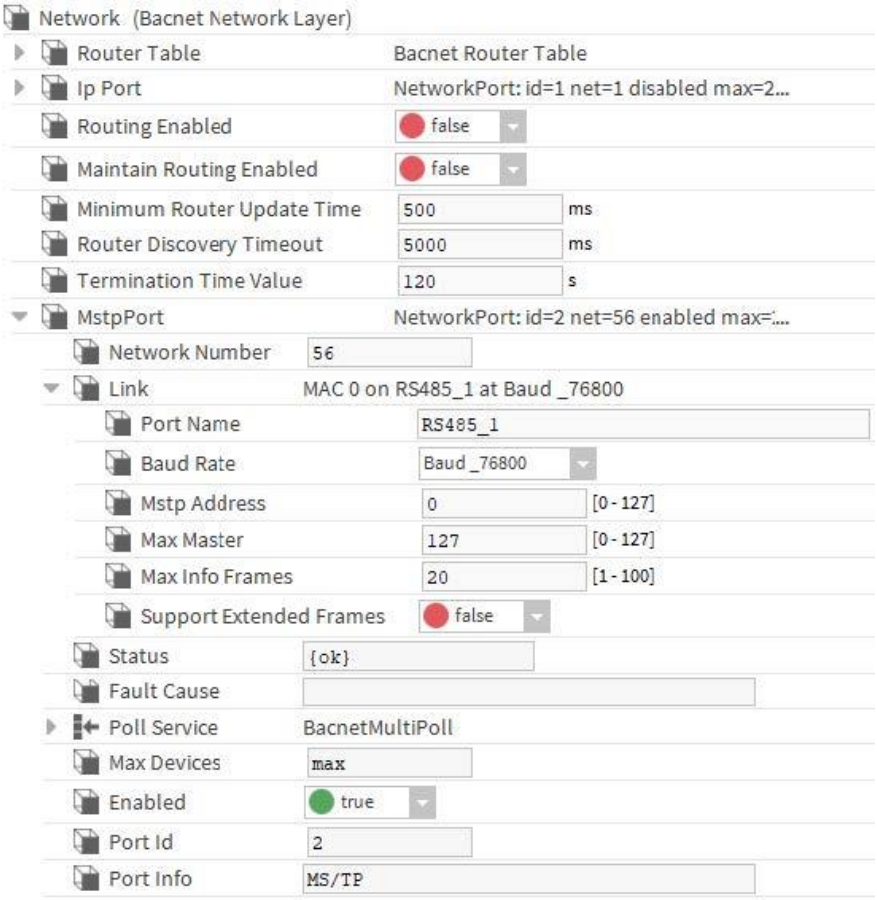
5. From the *BACnet* palette, add the **MstpPort** to **Network.**

6. Expand **MstpPort** and **Link.**

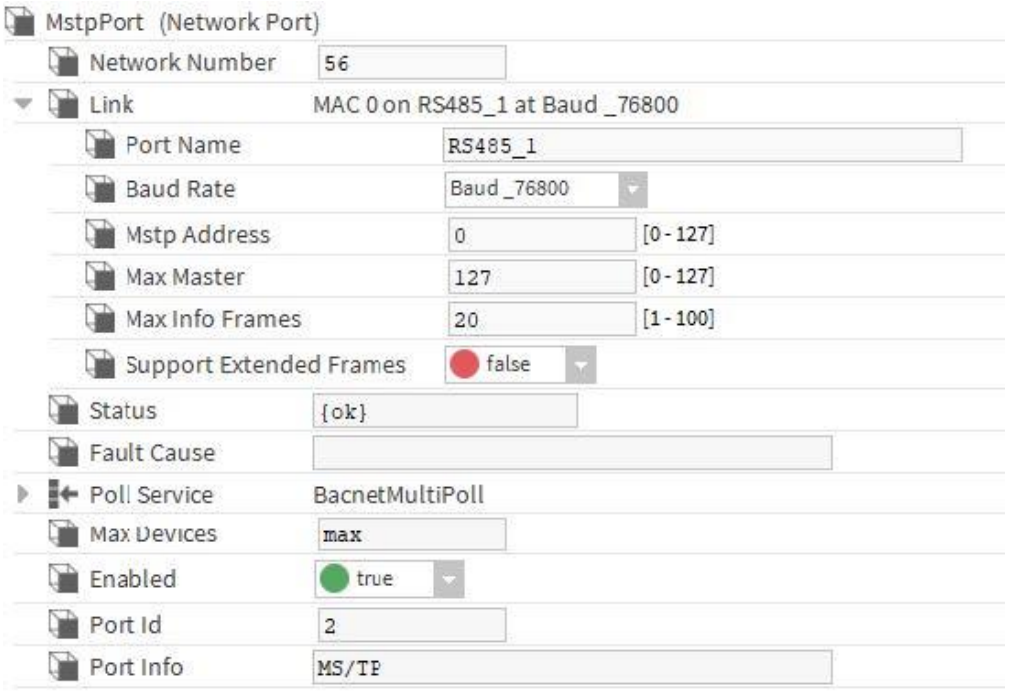


7. Set/enter the following:

- Network Number = any, e.g. 56 (must be less or equal than 65535)
- Port Name = RS485_1
- Baud Rate = Baud_38400 (see also “Baud rate note” below)
- Enabled = true



8. Click **Save**.



NOTE: If any of the configuration settings are changed during operation, you must restart the CIPer Model 50 controller (see also Baud rate Note).

9. Commission the CIPer Model 50 station to the CIPer Model 50 controller (standard Niagara procedure).

   NOTE: For detailed information on standard Niagara techniques and functions, please refer to the Niagara online documentation and/or the CIPer Model 50 Controller User Guide: 31-00198-01.

Baud rate Note

The following baud rates for the MSTP interface in the Spyder Model 5 controller are supported:

- 9600
- 19200
- 38400 (default)
- 57600
- 76800

The baud rate of each Spyder Model 5 controller on the BACnet MSTP bus is automatically set by the defined baud rate of the BACnet IP - MSTP Router (CIPer Model 5) controller after the Spyder Model 5 controller is powered up and connected to the BACnet MSTP bus.

Setting/changing the baud rate of a single Spyder Model 5 controller is not possible.

When changing the baud rate of the BACnet MSTP bus of a running system, any connected Spyder Model 5 controller must be power-cycled to adapt the changed baud rate.

ENGINEERING MODES

There are two kinds of engineering you can use for engineering an Spyder Model 5 project:

- Offline Engineering
  In this mode, you create an empty BACnet device manually, add an application and match it afterwards to a device discovered on the BACnet bus by using the service pin. This is normally applied when doing the engineering in the office without having the hardware available but knowing the hardware specification of the devices to be used later at the plant.

- Online Engineering
  In this mode, you discover the devices on the BACnet network in the first step and use the devices instantly for application engineering. This is recommended when doing the engineering directly at the site with the devices already installed on the BACnet bus.

Offline Engineering

Offline engineering includes the following steps:

- Create IrmBacnetDevice
- Create application for the IrmBacnetDevice (optional)
- Later at the site, do the following:
    - Connect to the BACnet network
    - Discover devices on the BACnet network
    - Match the empty device with a discovered device
    - Synchronize the application, if necessary, by applying any of the following actions appropriately:
        - Learn from controller
        - Teach to controller
        - Clear project
        - Clear controller

For detailed descriptions, please refer to the corresponding sections:

- Creating IRM BACnet Device, p. 25
- Working with Applications – The IRM Program, p. 45
-

Matching Devices , p. 28
- Teaching to Controller, p. 82
- Learning from Controller, p. 83
- Clearing Project, p. 85
-

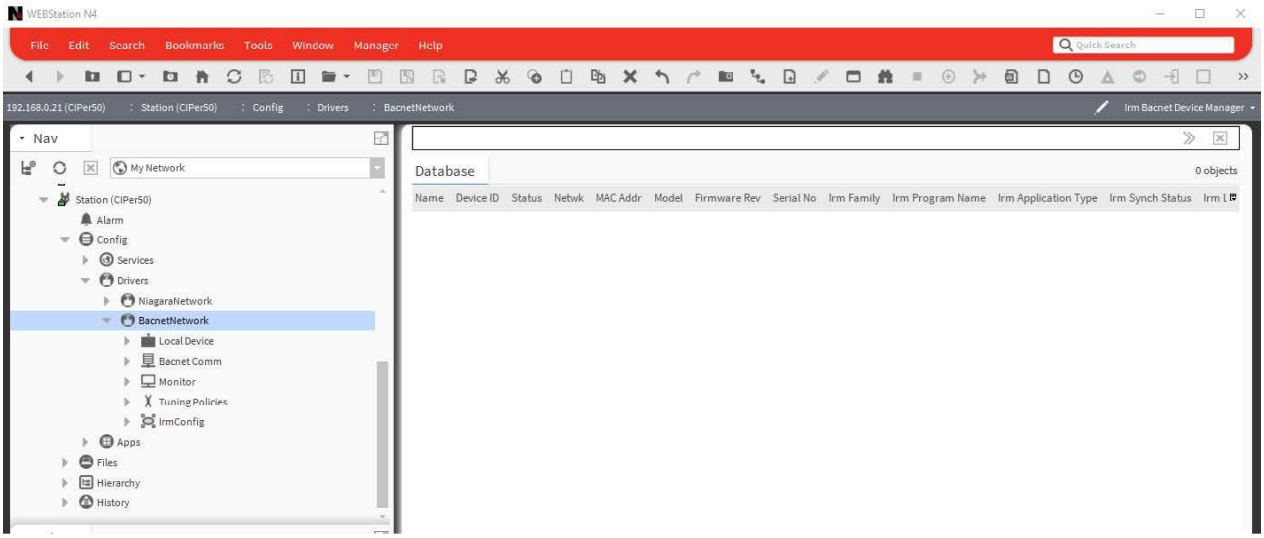
Clearing Controller, p. 85
- Checking Hardware Compatibility, p. 86.

For further information on offline engineering, please refer also to the “Factory Device Handling“ section, p. 37.

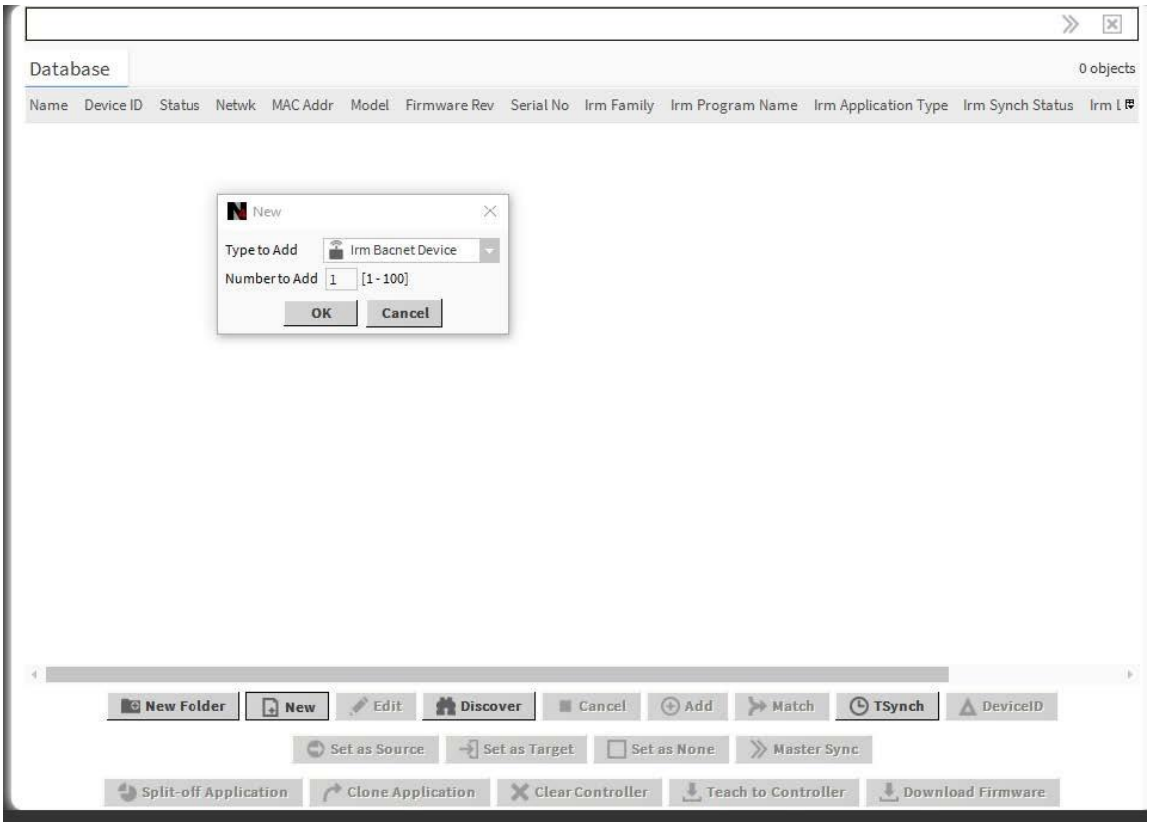
Creating IRM BACnet Device

Procedure

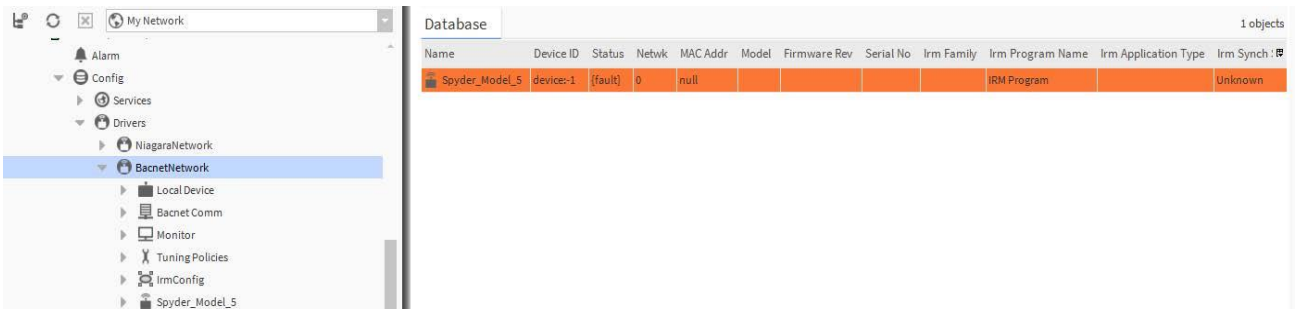
1. Double-click the BACnet network folder in the Nav tree, and then select Irm Bacnet Device Manager view.



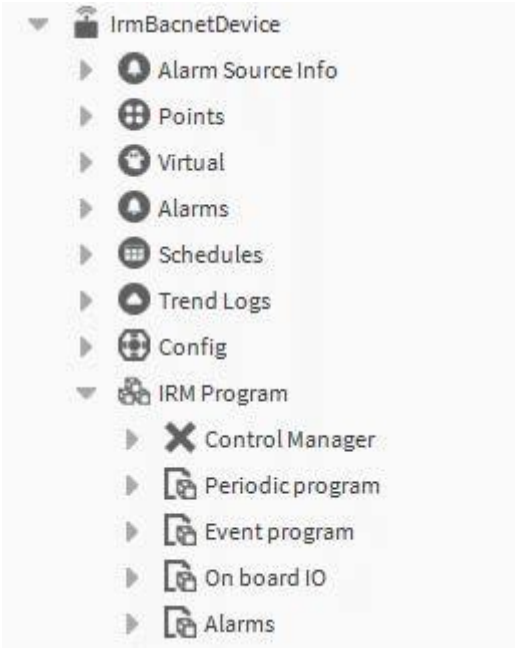
2. On the bottom, click **New**.



3. In the *New* dialog box, select ´Irm Bacnet Device`, and then click **OK**.
   RESULT: The Irm Bacnet Device is created and added to the *Database* pane and the *BacnetNetwork* tree.



4. Expand the device and create the IRM program by adding control strategy, hardware layout and alarm settings optional).



- To do so, please refer to the section “Working with Applications – The IRM Program“, p. 45.

5. After finishing the application engineering, match the offline disconnected device to the desired physical device discovered on the BACnet network (see “

Matching Devices “, p. 28).

6. Teach the application to the controller. If there is an application already running in the controller, clear the controller
7. Finally perform the hardware compatibility check to make sure that the application is properly designed for running seamlessly in the created device.
8. If software recommends, remove function blocks in the control logic.

   NOTE: Add the physical IO points. Make sure to select the right physical point template that matches the online device model.

   NOTE: It is allowed matching smaller offline device models with bigger online device models. It is strongly recommended not match devices vice versa, since hardware compatibility issues may arise. In this case, please execute the hardware compatibility check and rework the application accordingly. Or, install a bigger device with more hardware I/Os at the site.

For further information on offline engineering, please refer also to the “Factory Device Handling” section, p. 37.

Matching Devices

Purpose

This function matches an offline-configured, disconnected IRM device to an online discovered IRM controller on the BACnet network.

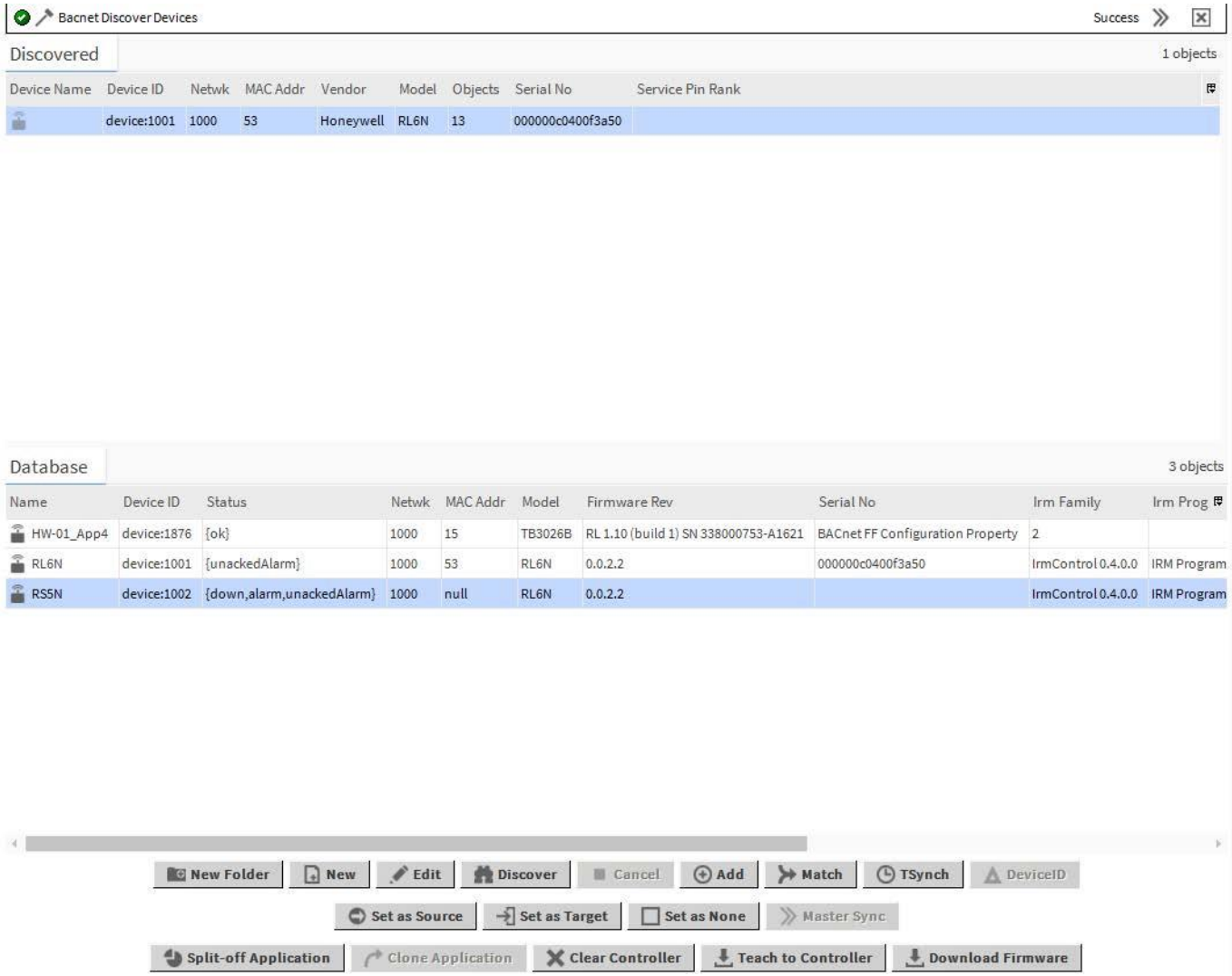
The offline-configured, disconnected IRM device will be configured according to the properties of the online discovered IRM device.

Inconsistencies may occur and can be solved by using the corresponding actions to establish synchronicity.

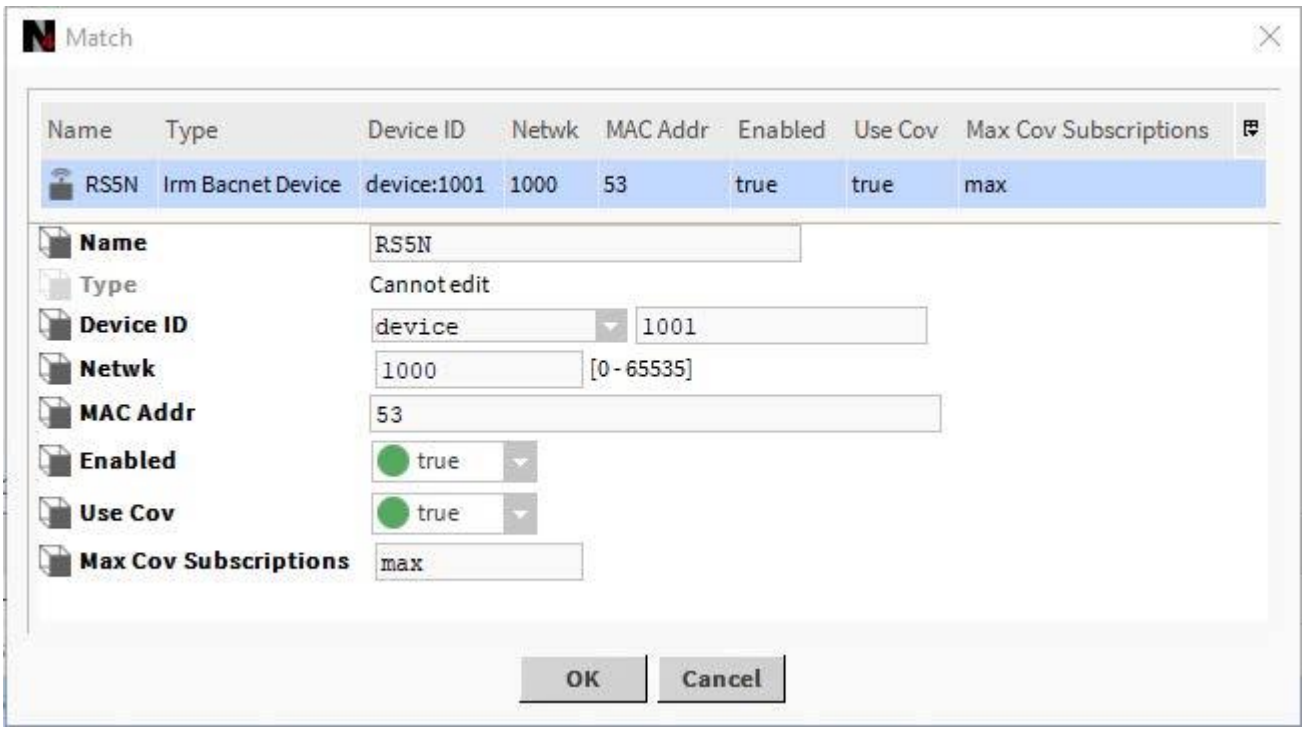
Procedure

1. Press the service pin at the device you want to match to the offline created IRM BACnet device. Do this for all devices, you want to match.
2. On the *Discovered* pane, click **Discover.**
   RESULT: All devices available in the BACnet network are discovered.

   RESULT: In the **Service Pin Rank** column, the service pin action is indicated by consecutive numbers depending on the time when the service pin was pressed at the device.
3. On the *Discovered* pane, select the online device and in the *Database* pane, select the offline created IRM BACnet device.



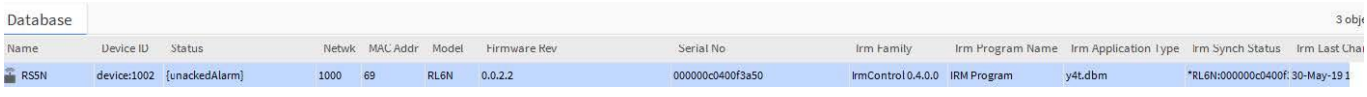
4. At the bottom, click **Match**.
   RESULT: The Match dialog box displays.



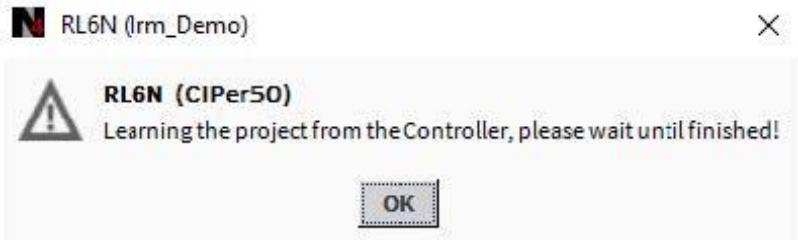
5. If desired, you can enter the device ID, Network address, and MAC Address.
6. Enable **Use Cov** by selecting ´true` from the drop-down List box.
7. Click **OK**.
   RESULT: The devices are matched as the properties of the disconnected offline IRM BACnet device indicate in the Database pane. The device gets the data of:

- Device ID
- Status
  e.g. Alarm, unacknowledged alarm.
- Model
- Firmware Revision
- Serial No.
- Irm Family
- Irm Program Name
- Irm Application Type
- Irm Sync Status
  synchronized or not synchronized
- Irm Last Change
- Irm Master Sync

The Irm Sync Status shows whether the device is part of a master sync group (source, target, none)



The application status is checked by learning the project from the controller as displayed in the following message:



8. Click **OK**.

   RESULT: If as a result, any inconsistencies of the application appear between the disconnected IRM BACnet device and the discovered Spyder Model 5 controller, a notification message will be displayed and the relevant synchronization status is indicated in the control manager (see section “Synchronicity Check via Control Manager”, p. 35).

9. To solve any inconsistencies, synchronize the applications by doing any of the following:

- Clear project
- Clear controller
- Teach to controller
- Learn from controller
- Checking Hardware Compatibility

For detailed descriptions, please refer to the corresponding sections:

- Clearing Project, p. 85
-

Clearing Controller, p. 85
- Teaching to Controller, p. 82
- Learning from Controller, p. 83
- Checking Hardware Compatibility, p. 86

For further information on offline engineering, please refer also to the “Factory Device Handling“section, p. 37.

Online Engineering

Online engineering includes the following steps:

- Connect to the BACnet network
- Discover devices on the BACnet network
- Create/change/ application and hardware layout if necessary
- Depending on the synchronization status, apply any of the following actions:
    - Learn from controller
    - Teach to controller
    - Clear project
    - Clear controller

For detailed descriptions, please refer to the corresponding sections:

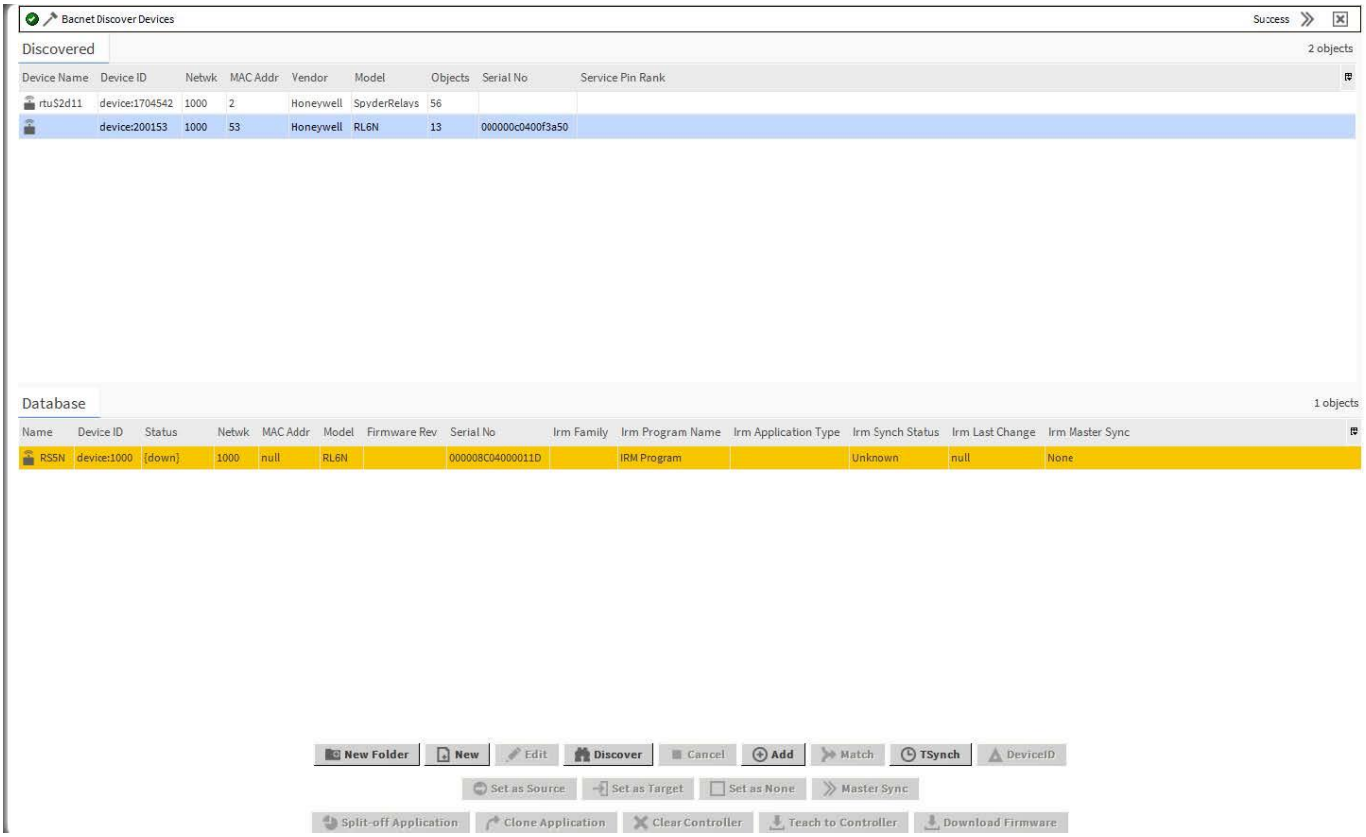
- Working with Applications – The IRM Program, p. 45
- Teaching to Controller, p. 82
- Learning from Controller, p. 83
- Clearing Project, p. 85
-

Clearing Controller, p. 85
- Checking Hardware Compatibility, p. 86

For further information on online engineering, please refer also to the “Factory Device Handling“ section, p. 37.

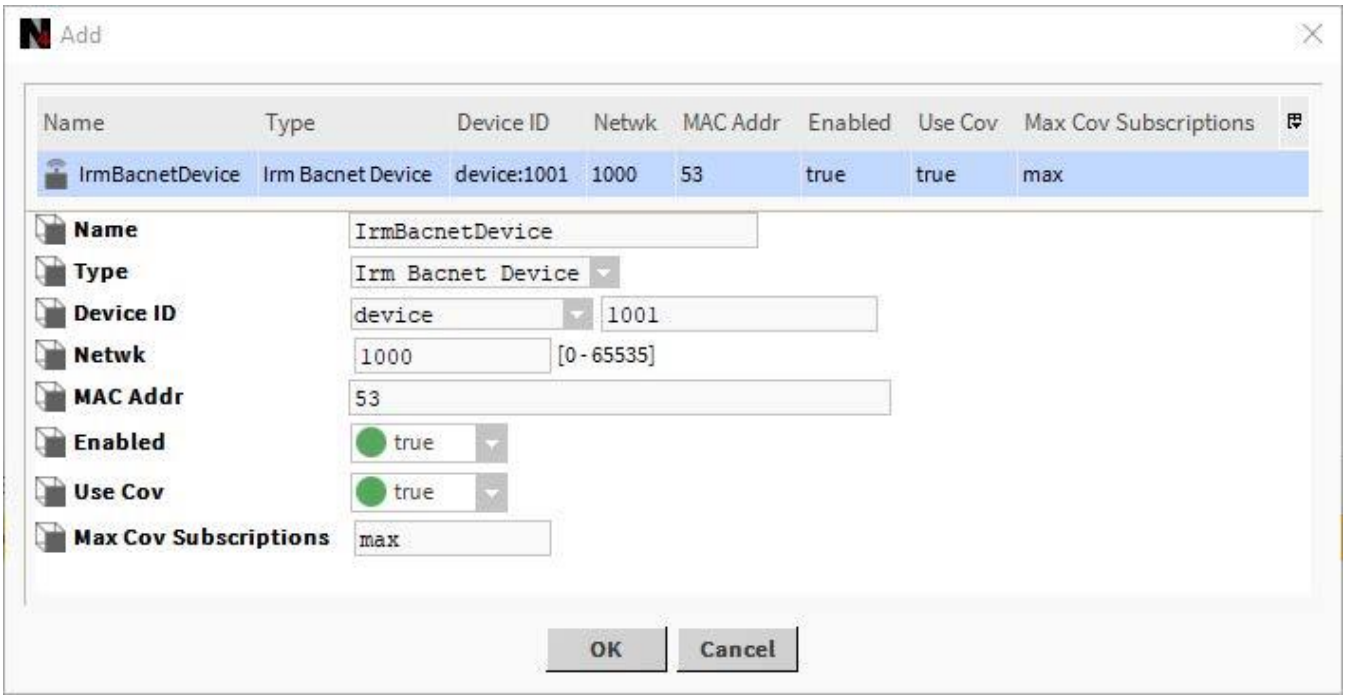
Procedure

1. For new factory devices, press the service pin at the devices.
2. On the *Discovered* pane, click **Discover.**
   RESULT: All devices available in the BACnet network are discovered.

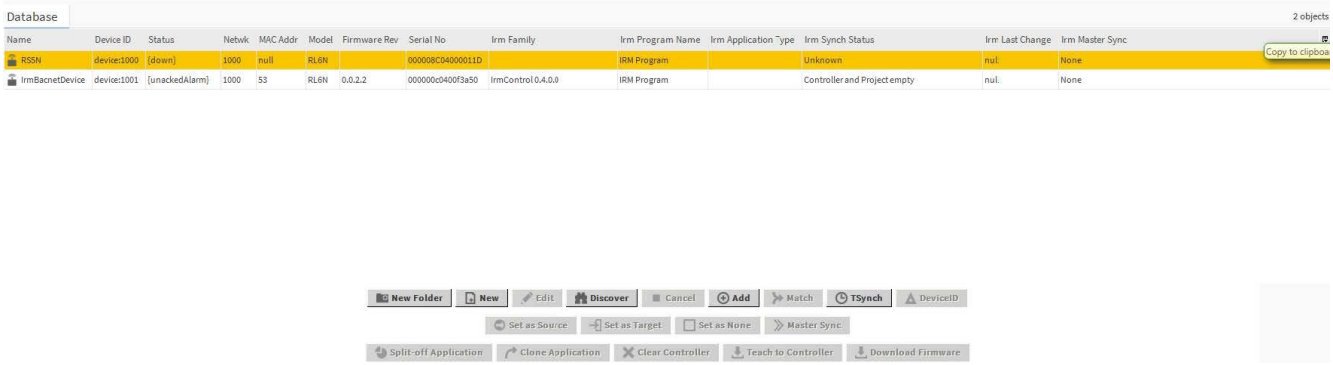


3. On the *Discovered* pane, select the devices you want to add to the database.

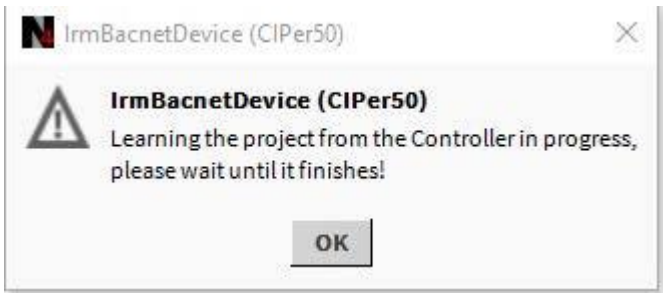
4. Click **Add**.
   RESULT: The Add dialog box displays. You can change properties of the devices before adding them to the database. To do so, select the device and change the desired property from:
   - Name
   - Device ID
   - Network
   - Use Cov
   - Max Cov Subscriptions

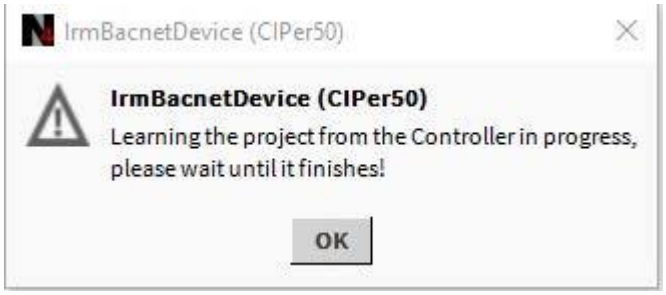


5. Click **OK** in *Add* dialog box.



RESULT: All devices are added to the database. The devices are learned from the project to the controller as indicated by the following messages.





6. Click **OK** to confirm.

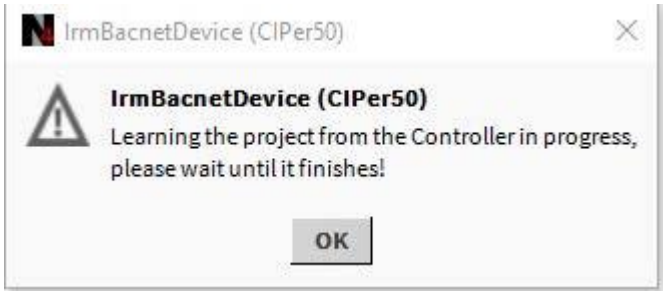
   RESULT: All devices are added to the database. The following properties are shown:
   - Name
   - Device ID
   - Status e.g. Alarm, unacknowledged alarm.
   - Model
   - Firmware Revision
   - Serial No.
   - Irm Family
   - Irm Program Name
   - Irm Application Type
   - Irm Sync Status synchronized or not synchronized
   - Irm Last Change
   - Irm Master Sync shows whether the device is part of a master sync group (source, target, none)

A synchronicity check is performed and particular messages are displayed if non-synchronicity is detected (see section “Synchronicity Check via Control Manager”, p. 35).

Synchronicity Check via Control Manager

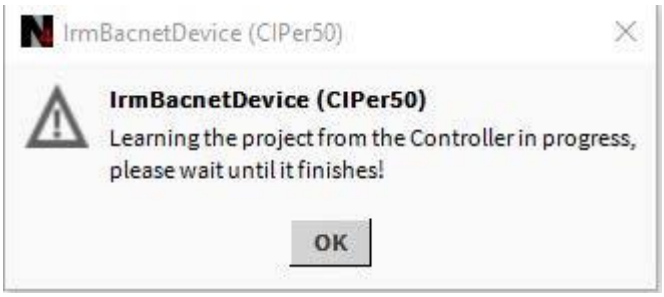
Whenever you connect to the BACnet network via device discovery (online mode) or when matching an empty IRM device (offline mode) to a discovered device of the BACnet network, the control manager starts working and checks the applications in the project and in the controller on synchronicity.

Example: Message when discovering device on the network



An application in the project can be either synchronous or not synchronous with the application running in the connected controller.

As soon as non-synchronicity is detected, notification messages about the synchronization status are displayed.



NOTE: Software always give support by displaying appropriate messages which describe the current status/problem and how to proceed.

WARNING!

It is strongly recommended to read and note any messages in order to avoid derived misleading actions afterwards which may result in the deletion or loss of application data.

To solve any inconsistencies, synchronize the applications by doing any of the following:

- Clear project
- Clear controller
- Teach to controller
- Learn from controller
- Checking Hardware Compatibility

For detailed descriptions, please refer to the corresponding sections:

- Clearing Project, p. 85
-

Clearing Controller, p. 85
- Teaching to Controller, p. 82
- Learning from Controller, p. 83
- Checking Hardware Compatibility, p. 86

For detailed description of the control manager, see section “The Control Manager”, p. 50).

Factory Device Handling

Default Factory Device ID Settings

For successful communication via BACnet, each controller must have a unique device ID. By default, each new IRM controller has assigned the factory device ID = 4194302. But, when multiple controllers have the same device ID assigned, they cannot talk via the BACnet protocol (Who is? - I am broadcast messages).

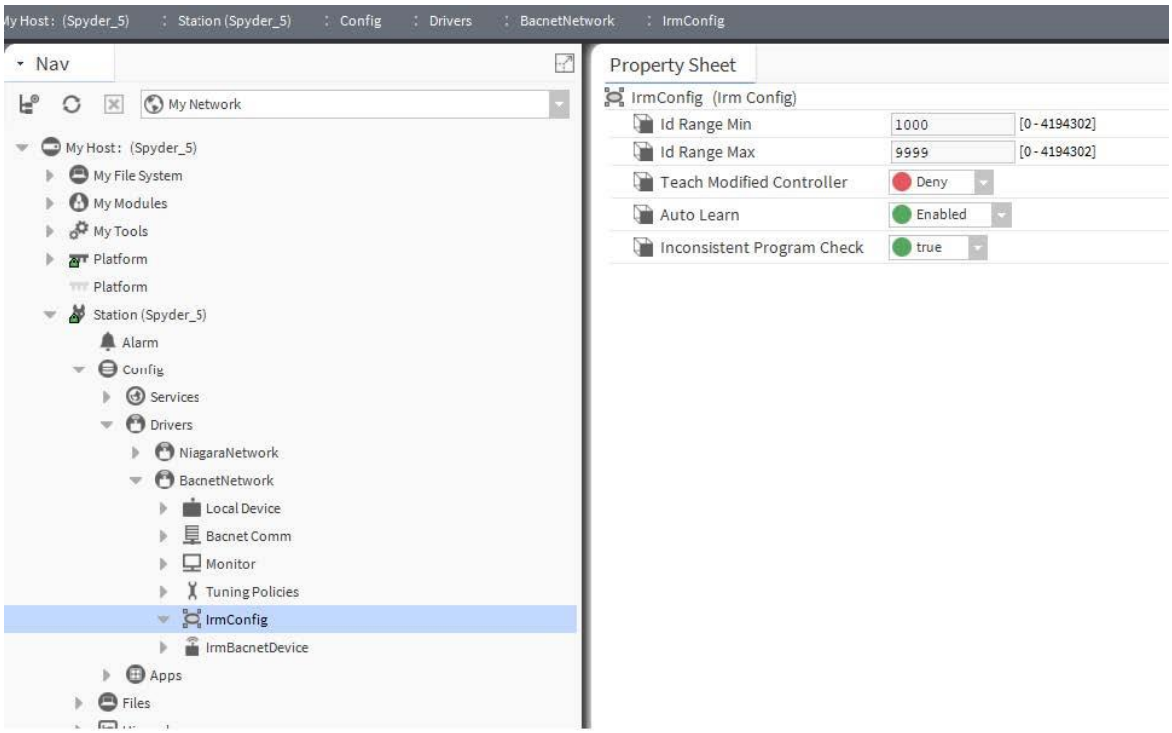
As a result, new controllers on the MSTP bus will not be visible after discovery.

IRM Configuration Settings

For engineering IRM controllers on a BACnet network, particular settings must be configured as default settings.

Procedure

1. To do this, expand the *BacnetNetwork* folder in the *Nav* tree.
2. Double-click the *IrmConfig* folder in the BACnet network to display the property sheet.



3. On the property sheet, do any of the following:

- Id Range Min Enter the value for the lowest drive ID that is assigned to a device during discovery.
- Id Range Max Enter the value for the highest device ID that is assigned to a device during discovery.

  Devices will get device IDs automatically assigned within the range of Id Range Min and Id Range Max settings

  NOTES: Actual device Id assignment is executed not till the controller is connected to the network.

  Then when adding the discovered controllers to the database, in the *Edit* dialog to each controller a unique device Id is issued within the range of the min and max Id limits configured in the IRM Config Properties (see above).

- Teach Modified Controller

IMPORTANT!

It is strongly recommended not to change this setting.

- Auto Learn If enabled (default), the project is automatically deleted and learned from the controller when a discovered controller is added to the database. If disabled, the project is kept and learning from the controller is not executed (see “Learning from Controller” section p. 74)
- Inconsistent Program Check If true (default), software checks the application for inconsistencies and shows a message if this is the case. Inconsistencies may occur if you have moved/copied the application to an external location, e.g. to the palette, changed it there and then include it again

  NOTE: Copying and pasting parts of an application from an external location such another controller does not generate inconsistencies in the application.

Assignment of Unique Device IDs

Depending on the requirements/scenarios (online at the site, offline in the office) new IRM controllers need to be engineered by using or by NOT using the service pin button.

Offline Mode (service pin not accessible)

Create a device (controller) by entering the device name and the serial number for the controller (use a barcode scanner software if available or enter the data manually). Do this for all controllers.

Discover the controllers. This will list all offline devices.

At the site, go online by connecting to the network.

Discover the controllers on the network. Now software matched the serial numbers issued for offline devices.

Once those controllers are added to the network, they will start communicating with the physical online controllers.

Then when adding the discovered controllers to the database, to each controller a unique device Id is issued within the range of the min and max Id limits configured in the IRM Config Properties (see above).

NOTE: In the *Add* dialog box, you can change the pre-defined device ID before the database addition.

Online Mode (Service pin accessible)

For each controller, press the service pin at the controller before executing the Discover command. The controllers will be discovered, but they still show the default factory Id.

Then when adding the discovered controllers to the database, to each controller a unique device Id is issued within the range of the min and max Id limits configured in the IRM Config Properties (see above).

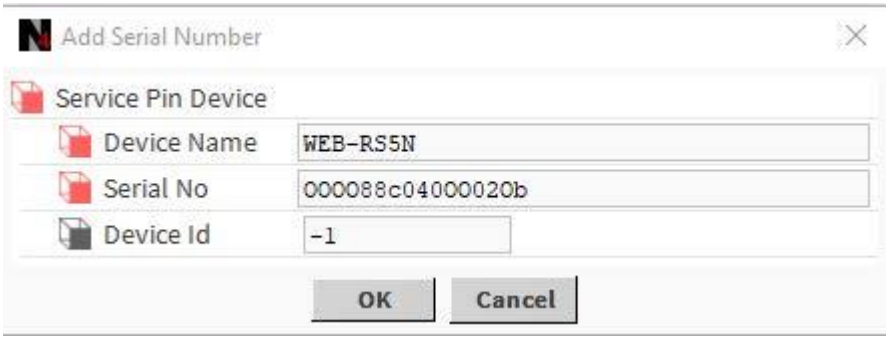
NOTE: In the *Add* dialog, you can change the pre-defined device ID before the database addition.

In both cases, device Ids will be assigned within the range of the min and max Id limits configured in the IRM Config Properties (see above).

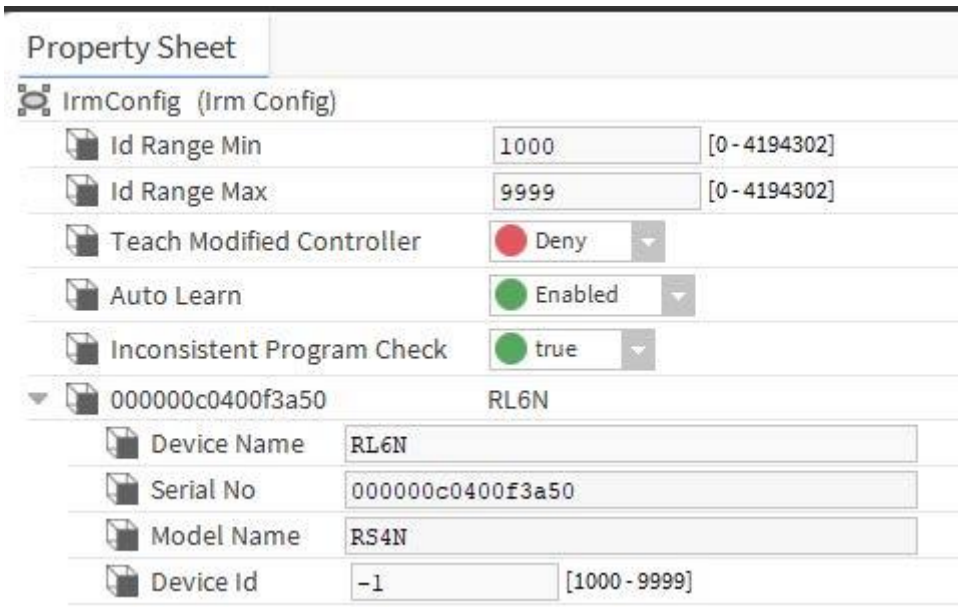
Detailed Offline Procedure

1. Expand the *BacnetNetwork* folder in the *Nav* tree.
2. Right-click the *IrmConfig* folder and then click **Add Serial Number** in the context menu.
   RESULT: The *Add Serial Number* dialog box displays.

3. Enter the **Device Name**, **Serial No** and **Device Id**. Use a barcode scanner software if available or enter the data manually.



4. Confirm by clicking **OK.**
   RESULT: The device is added to the *IrmConfig* Property Sheet on the right pane.



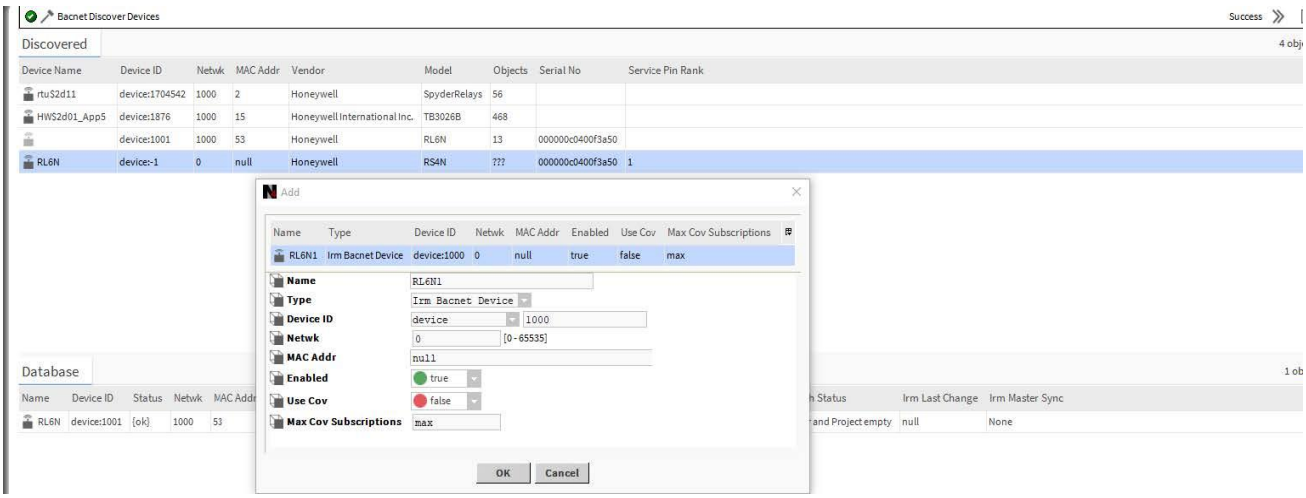
5. Repeat steps 2 through 4 for all controllers.
6. Discover the controllers. This will list all offline devices.

7. At the site, go online by connecting to the network.
8. Discover the controllers on the network.

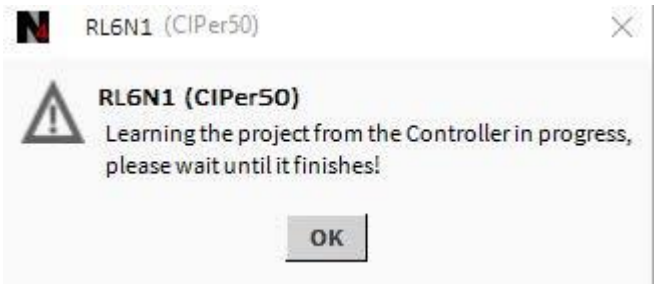
9. Now software searches for controllers that have the same serial numbers issued offline.
10. Add the controllers to the database.

RESULT: Once those controllers are added to the network, they will start communicating with the physical online controllers.

While adding the discovered controllers to the database, a unique device Id is issued to each controller within the range of the min and max Id limits configured in the IRM Config Properties (see screen above).

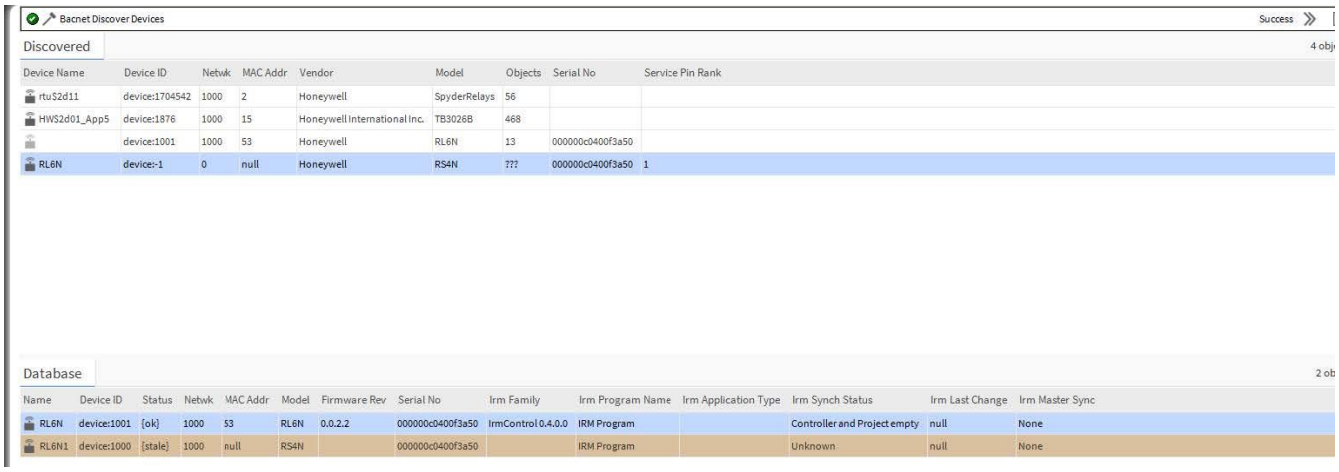


NOTE: In the *Add* dialog, you can change the pre-defined device ID before the database addition.



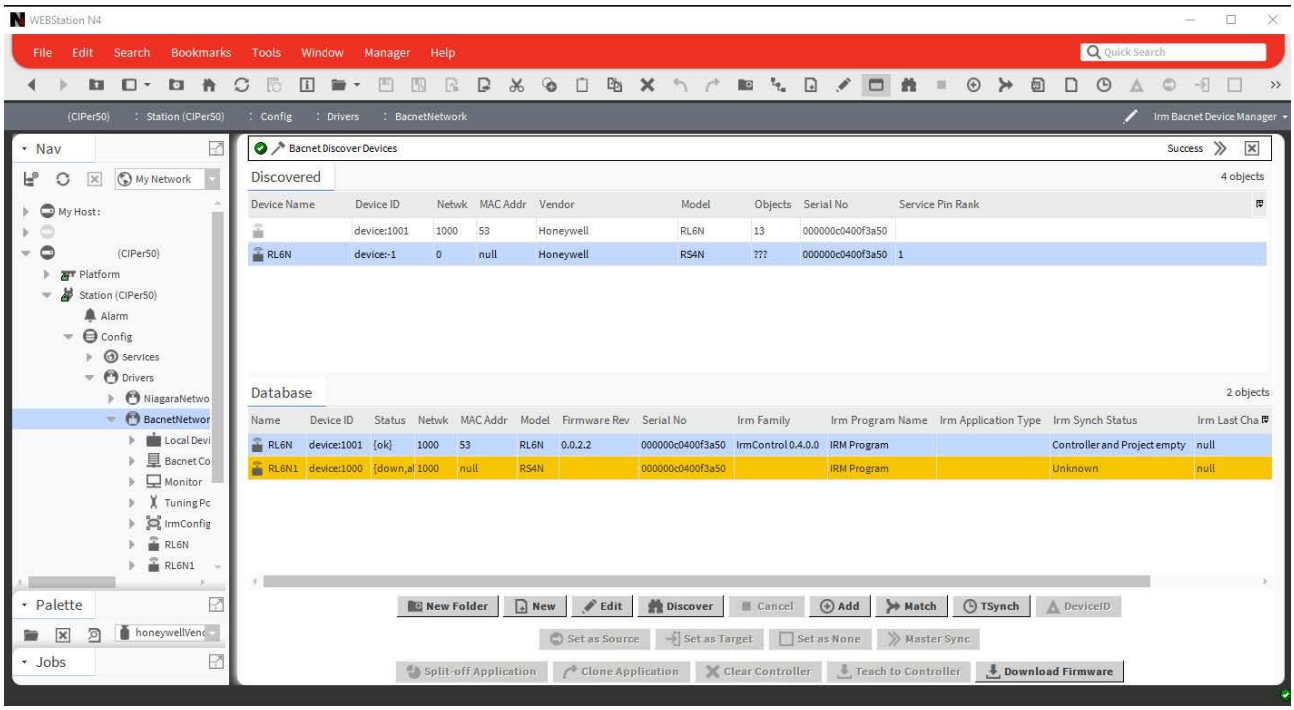
11. If any Sync messages are displayed, confirm by clicking **OK**.
12. Finish by matching the offline device with the online device in the database (see “

Matching Devices“, p. 28).

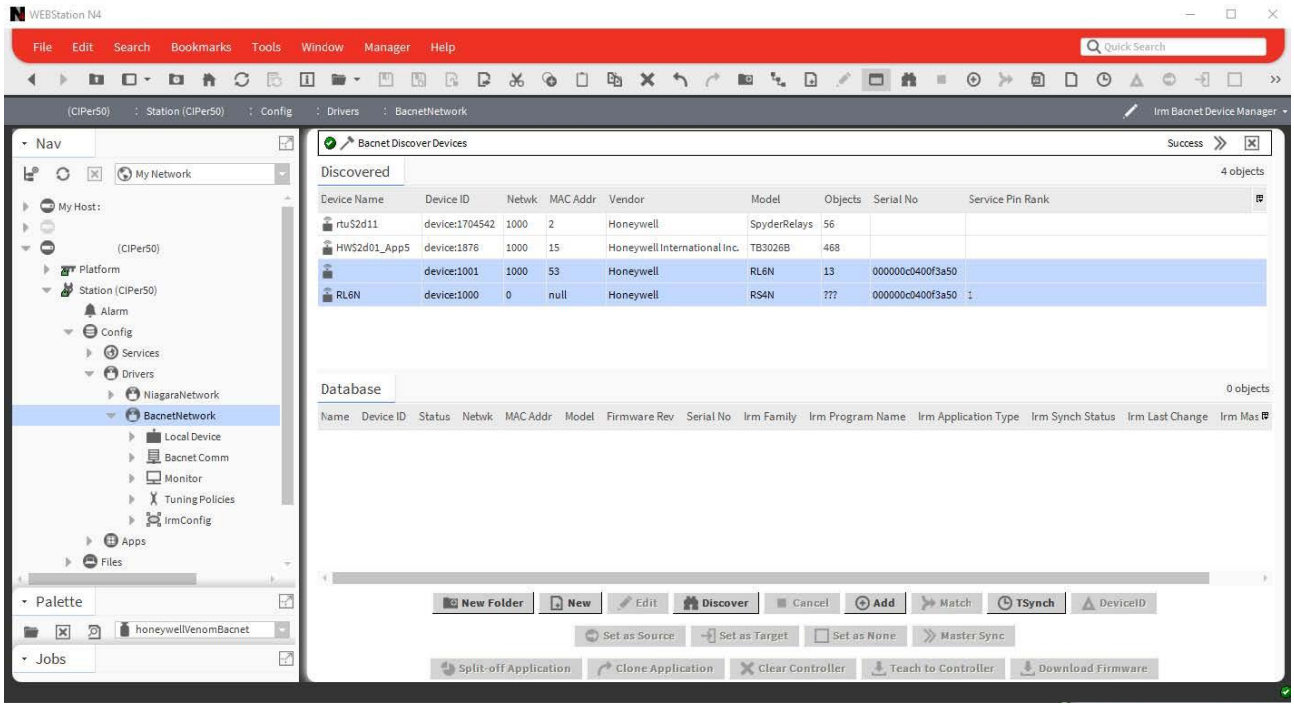


Detailed Online Procedure

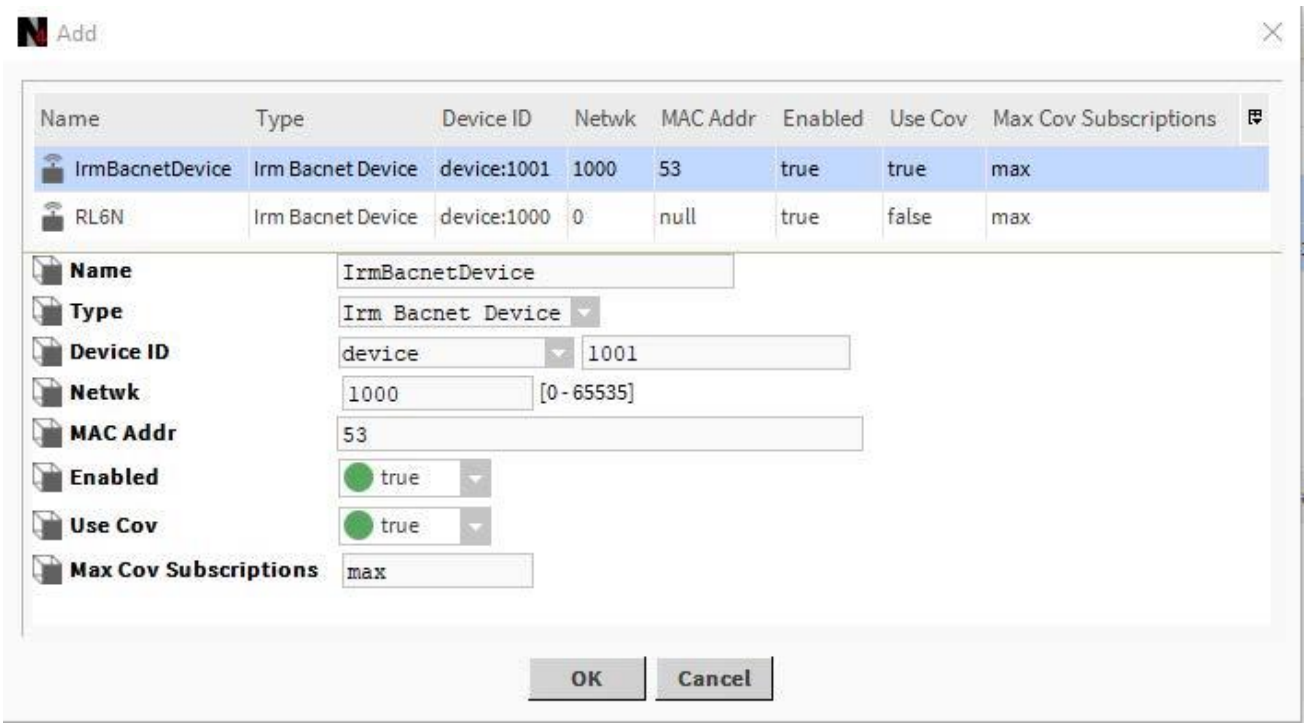
1. Double-click the *BacnetNetwork* folder in the *Nav* tree.
2. At the controller, press the service pin for 15 sec. Then click **Discover.**
   RESULT: The controller will be discovered, but it still shows the default factory Id ´4194302`.



3. Repeat step 2 for each controller.
4. Select the controllers.

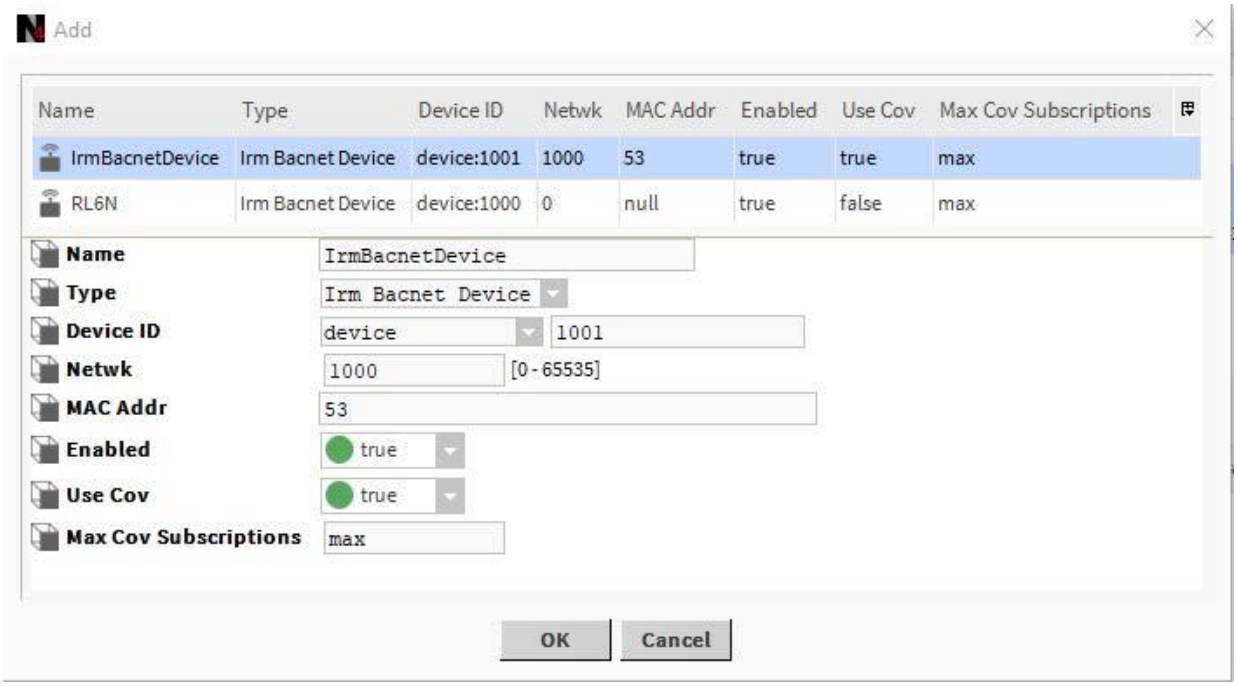


5. Add them to the database via drag & drop or by clicking **Add**.
   RESULT: The *Add* dialog box displays.



To each controller, a unique device Id is assigned within the range of the min and max Id limits configured in the IRM Config Properties.

NOTE: In the *Add* dialog box, you can change the pre-defined device ID before the database addition.



6. Click **OK**.

RESULT: The controllers are added to the database.

For further information, please refer also to the “Engineering modes” section, p. 22.

WORKING WITH APPLICATIONS – THE IRM PROGRAM

Application engineering includes the creation of:

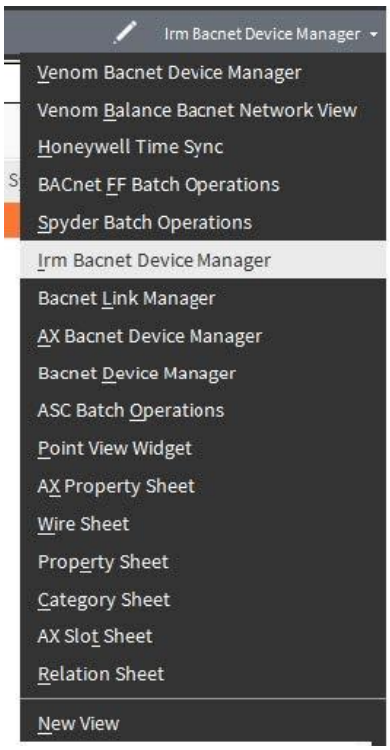
- Periodic control strategy and/or
- Event driven control strategy
- Hardware layout (onboard IO)
- Alarming settings

Preparing Niagara Work Environment

For working conveniently during application engineering, prepare the work environment by enabling the following functions in the Spyder Model 5 Engineering Tool:

BACnet Device Manager View

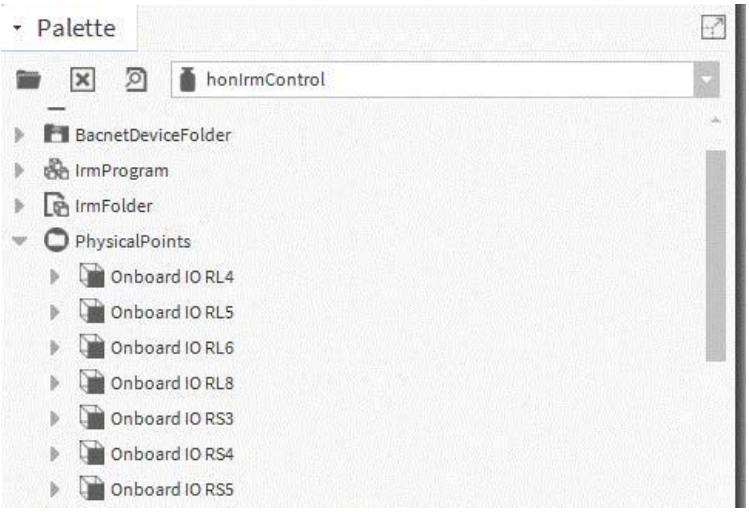
Double-click the *BACnet* network folder in the *Nav* tree, and then select the **IRM BACnet Device Manager** view.



IRM Palette

The following palette is available for creating the application:

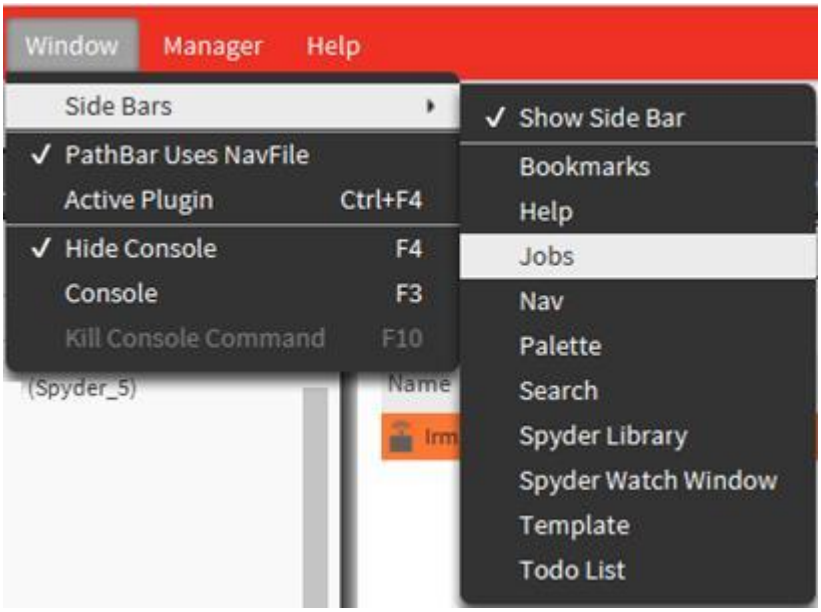
- honIrmControl - provides control function blocks and templates for IRM BACnet devices, IRM programs, and folders Open the palette in the *Palette* pane
- honIrmAppl - provides standard FCU application with all commonly used functionalities.



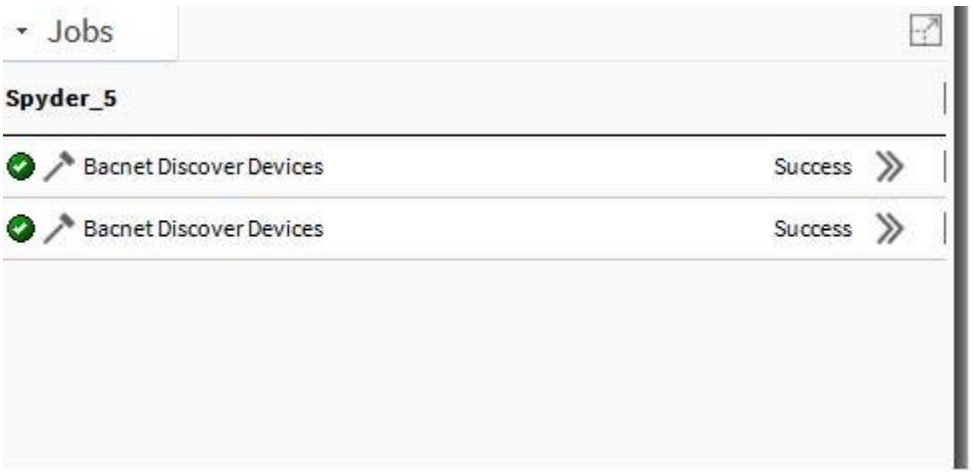
IRM Operations Monitor / Jobs Sidebar

The IRM Operations Monitor shows all actions in a popup window and all actions are summarized in a list in the *Jobs* window.

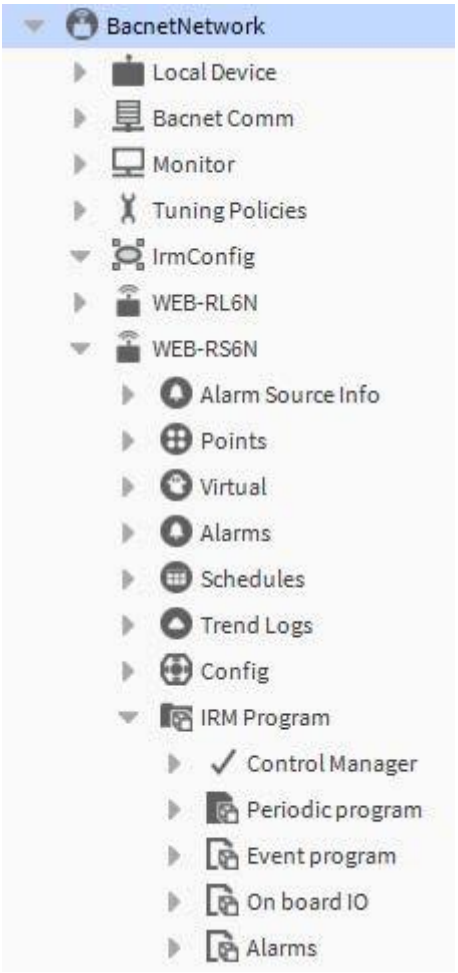
In the **Windows** menu, click **Side Bars**, and then click **Jobs**.



When actions are executed, they will be shown in the temporary Spyder_5 *Operations Monitor* popup window and listed permanently in the *Jobs* window.



You can also see the devices can also be seen in the nav tree as shown below.

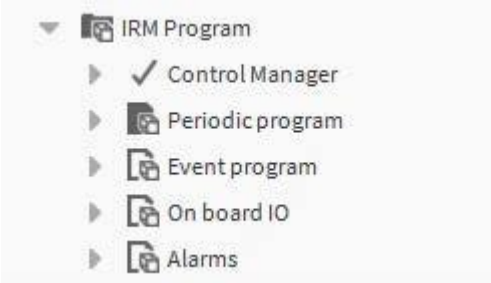


IRM Program Components

IRM Program in Nav tree

The IRM Program is part of the device and includes the following components:

- Control Manager
- Periodic program
- Event program
- Onboard IO
- Alarms



Wire Sheet and Property Sheet View

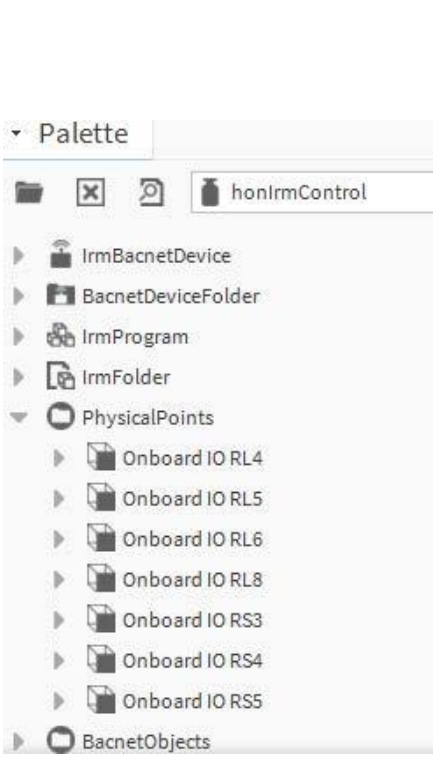
The component's work space can be shown by double-clicking it in the tree and selecting the *Wire Sheet* view. For the Control Manager, the *Property Sheet* displays after double-clicking on the item.

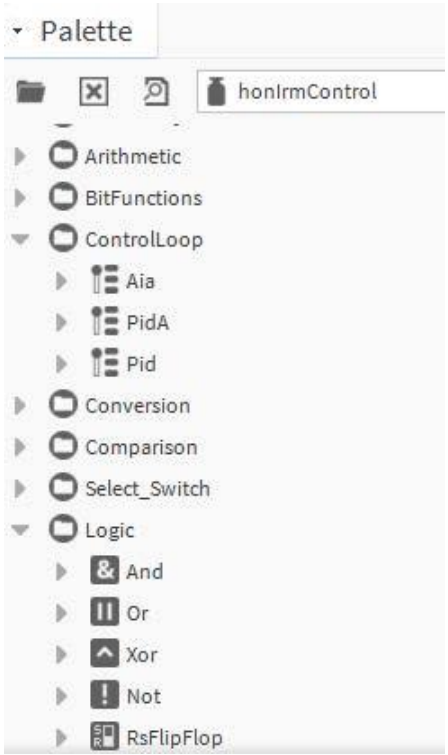
For detailed functional descriptions, please refer to the following sections:

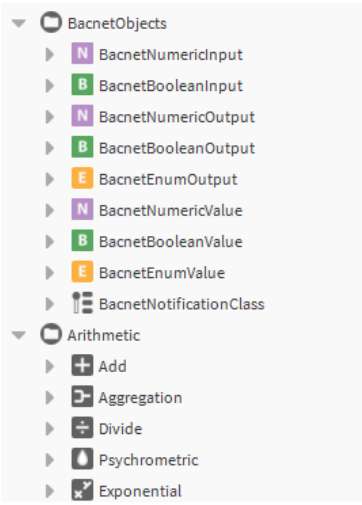
- The Control Manager, p. 50
- Periodic Program, p. 53
- Event Program, p. 54
- Onboard IO, p. 55
- Alarms, p. 58

Control Logic Source / Palette

For creating the application, the "honIrmControl" palette is used which includes all necessary components such as logic function blocks, terminals, and templates for IRM BACnet devices, IRM programs, and folders







Modifications and Consistency Check

Whenever working on the application, e.g. periodic program, on board I/O etc., every modification is detected by the control manager and indicated graphically by a yellow warning symbol that replaces the original symbols at the modified item on the wire sheet, and in the tree. This applies to teaching mode ´on demand` only. When working in teaching mode ´immediate`, changes are not indicated graphically since they are written instantly to the controller.

Example:

In addition, the IRM Program icon and the component´s icon gets a dark square background .

The control manager icon switches from synchronous state ✓ to non-synchronous state ..

On the property sheet of the controller manager, the synchronization status switches accordingly:

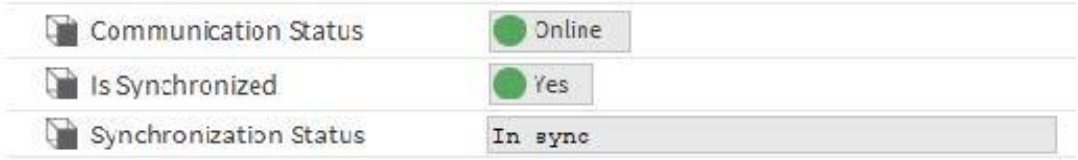


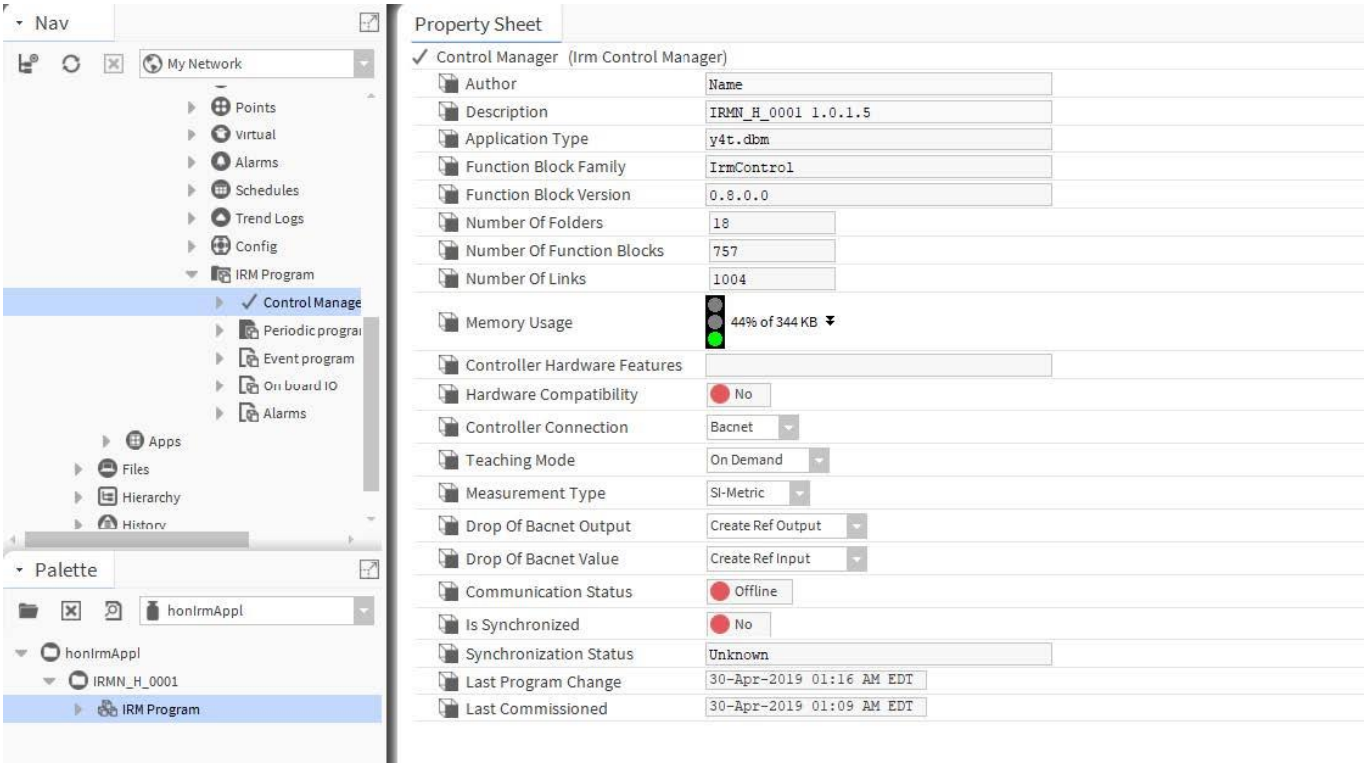
Fig. 3. Synchronization Status Information in Control Manager before Application Change

Fig. 4. Synchronization Status Information in Control Manager after Application Change

Depending on where the application is created (project) or changed (project or controller), the application must be taught to the controller, or must be learned from the controller to achieve synchronicity. In both cases, if meaningful and required, the controller or the project must be cleared before.

The Control Manager Procedure

1. In the *Nav* tree, expand the IRM Program folder.
2. Double-click on **Control Manager** icon.



RESULT: On the *Property Sheet*, the following information is displayed.

- Author of the application.
- Description.
- Application Type.
- Function Block Family.
- Function Block Version.
- Number of Folders.
- Numver of function blocks.
- Number of Links.
- Memory usage shows the memory usage of the device, application and parameters in percentage and graphically.
- Controller Hardware features displays the I/O configuration (Online available in Online mode).
- Hardware compatibility indicates weather the terminal layout of the application is compatible with the hardware layout of the physical controller.
- Controller connection allows selecting the connection type from: network, offline, and simulation.
- Teaching mode allows selecting the teaching mode from on demand and immediate .
- Measurement Type allows selecting the type of engineering units from SI-metric or Imperial.
- Drop of BACnet Output defines which type of reference point, reference input or output is created when dropping a BACnet output from another device.
- Drop of BACnet Value defines which type of reference point, reference input or output is created when dropping a BACnet value point from another device.
- Communication Status displays the communication status of the device: online or offline.
- Is Synchronized indicates whether the project and the controller are synchronized (yes) or not (no).
- Synchronization status displays a message describing the cause of the current synchronization status.
- Last program change displays the date of the last program change.
- Last Commissioned displays the date of the last commissioning.

3. Do any of the following if desired:

    a. In **Controller Connection**, select the connection type for the devices from:

    - Network connects to the physical BACnet network.
    - Offline disables the connection to the network. This is useful in case you want to reduce bus traffic if the application has not been swapped out.

b. In **Teaching Mode**, select how the application changes in the project are written to the controller from:
- Immediate changes are written to the controller automatically and are effective immediately.
- on demand changes are written to the controller manually and explicitly by the Teach to Controller action.

NOTE: For both teaching modes, only the changes are written to the device, hence the process is very fast.

c. In **Measurement Type**, select the type of engineering units from:
- SI-metric
- Imperial

4. Click Save.

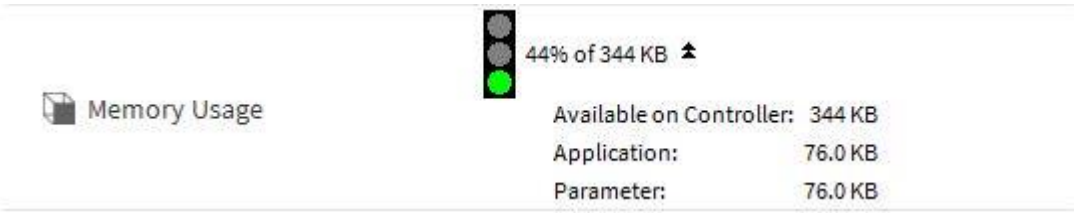
Memory Usage

The memory of the controller´s RAM is max. 344 KB and is consumed by the:

- Device
- Application
- parameters

The Spyder Model 5 Engineering Tool provides a memory consumption check of the current application in the tool. This is an approximate calculation of the application in the tool but not a real-time consumption within the controller.

The calculated memory usage is shown in the IRM Manager as percentage and graphically via traffic light symbol.



Memory Usage and Its Display

The max. number of memory in the controller is 344 KB.

Each folder added consumes 8 KB of memory. The max. number of folders is 36, 4 folders are reserved by default. Hence, user-defined individual folders can be 32 at max.

The calculated memory consumption will be indicated graphically and as percentage in the traffic light symbol as follows:

- Green < 80 %
- Yellow >80 %
- Red >90 %

The memory usage can be viewed on the property sheets of the following different levels:

- IRM Program (percentage)
- Control Manager (percentage and graphically)
- Periodic and Event Program and its subfolders (percentage)

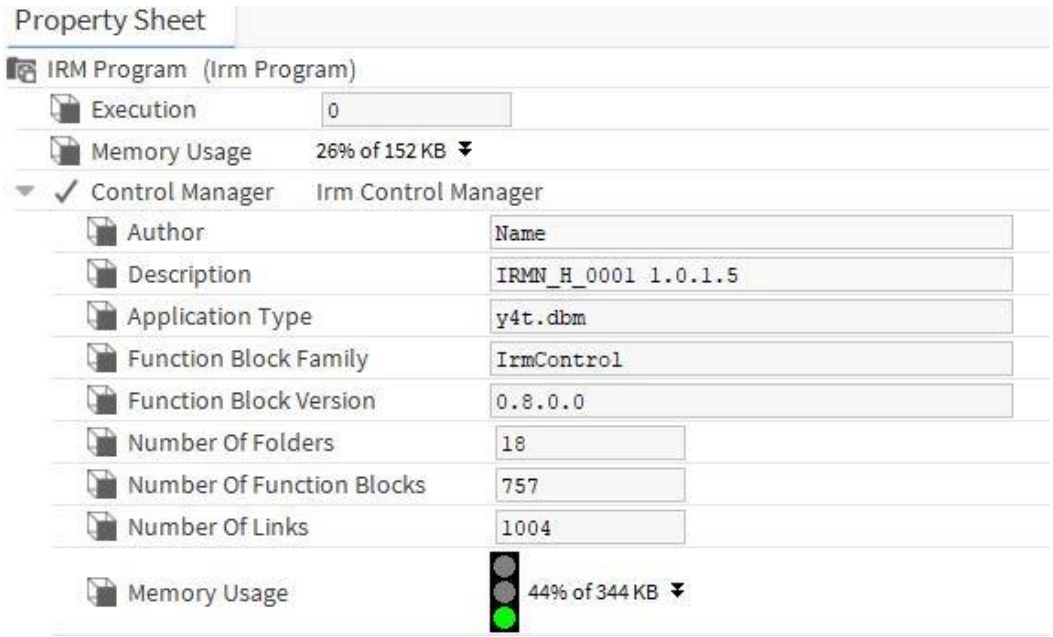


Fig. 5. Memory Usage Display on Control Manager Level

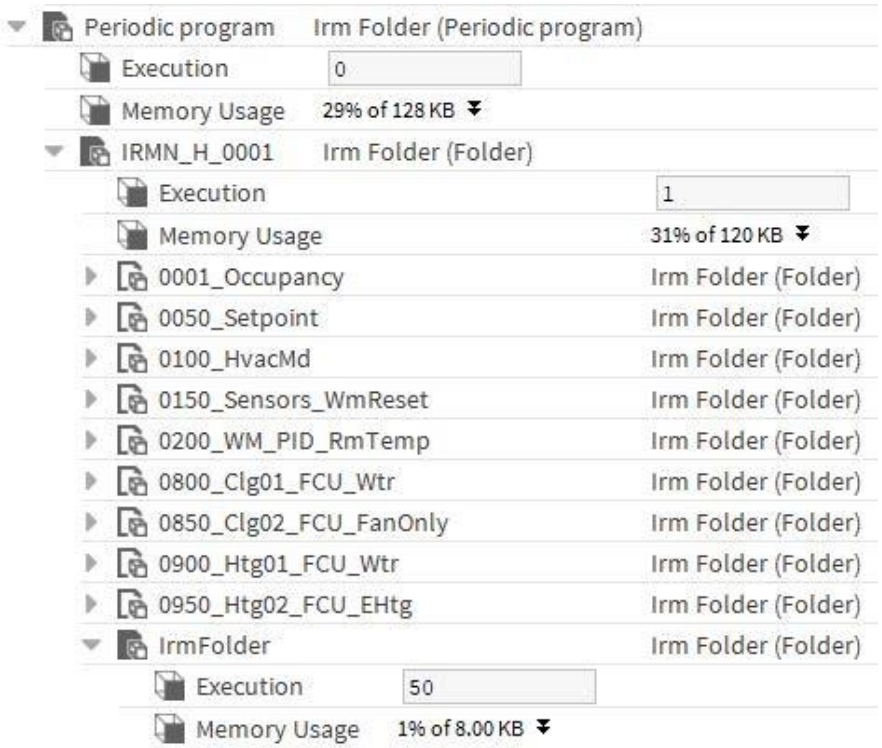


Fig. 6. Memory Usage Display on IRM Program Folder Level

Recommendations

It is recommended to monitor the memory usage on the Control Manager level and to avoid to exceed the limit of 80 %.

To keep the memory usage as low as possible, it is recommended to optimize application engineering by keeping the number of folders as low as possible and the number of added function blocks per folder as high as possible in relation to the required result of the logic.

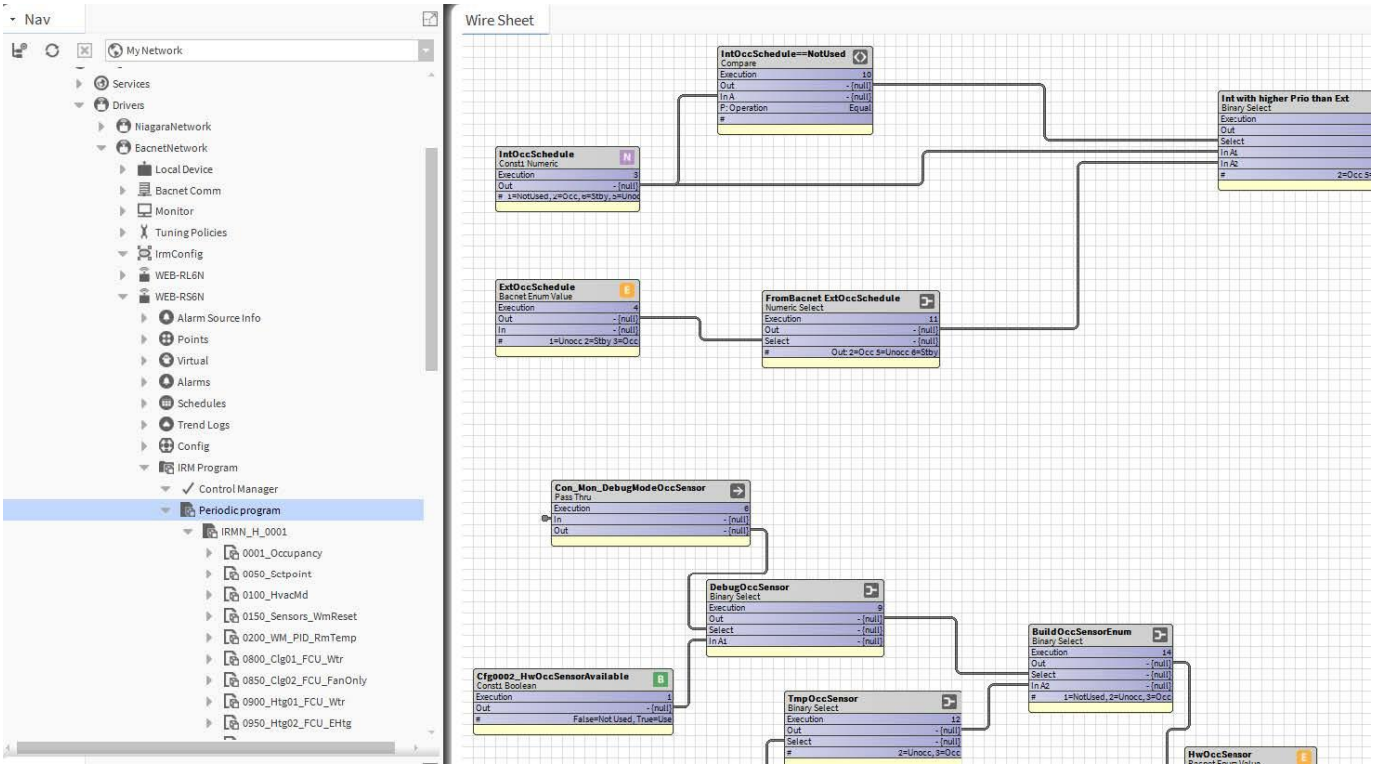
The max. number of function blocks per folder is 150. If you want to use more than 150 function blocks, please split them among two or more folders.

Periodic Program

Creates the control strategy running in a cyclic manner on a fixed time base. All function blocks in the root folder and its subfolders are processed. The periodic program is executed every 500 ms within the controller.

Procedure

1. Double-click *Periodic Program* in the tree, and then select the **Wire Sheet** view.
2. Open the “honIrmControl” palette.
3. From the palette, add control items to the wire sheet via drag & drop.
4. Create/change the control logic by applying desired steps such as connecting, adding, deleting, moving icons, and/or adding and deleting connections.



Synchronization Check

Any modifications on the Periodic Program wire sheet are detected by the control manager. Modifications can be any of the following:

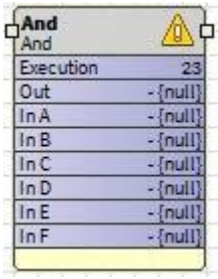
- Added item
- Deleted item
- Moved item
- Deleted connection
- Added connection
- etc.

As result, all modified items will be not in sync with the application in the controller and hence indicated by a yellow warning” symbol on the item.

The application can be synchronized according to the set teaching mode, on demand or immediate.

After synchronization, the warning symbols are removed and the items are indicated in its synchronized state.

See also the description of the “The Control Manager”, p. 50.



Event Program

Creates event-driven control strategy. All function blocks in the root folder and its subfolders are processed.

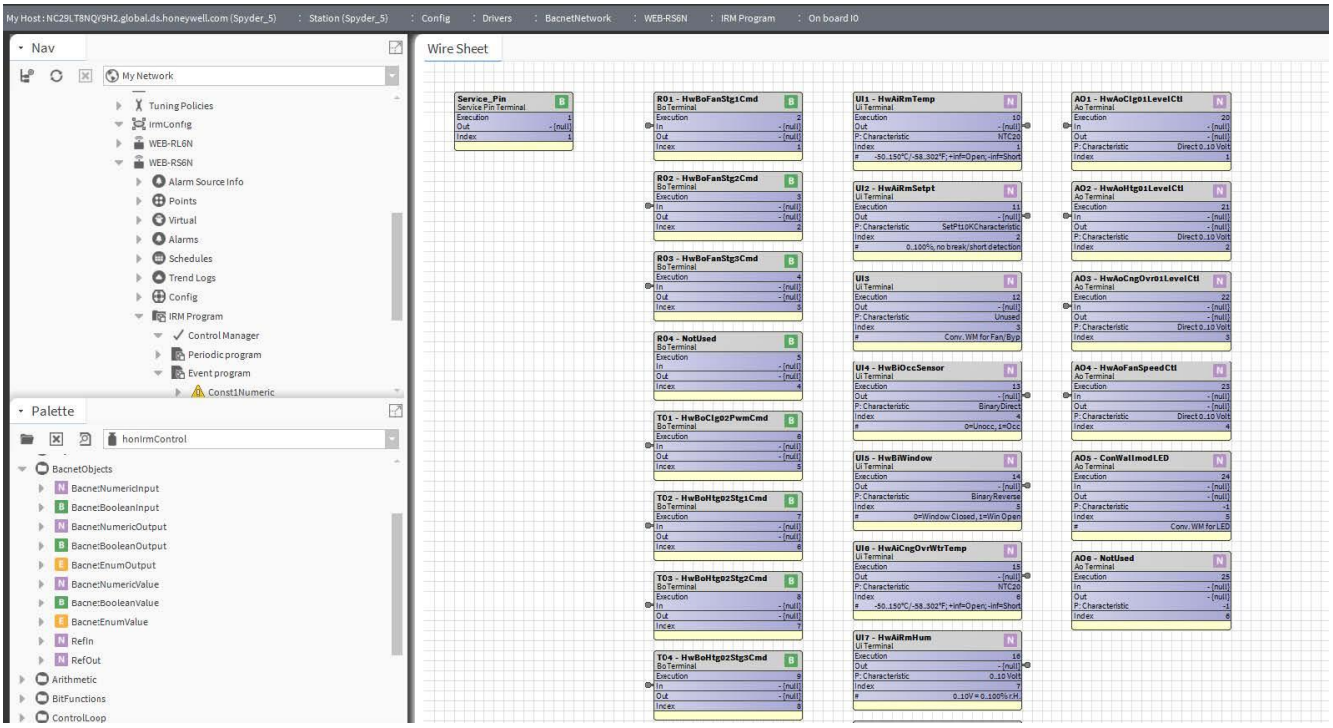
The event program is executed:

- when the time interval of 1000 ms has elapsed
- whenever the state of a hardware point configured as binary input that is used in the event program, changes and when this binary input point is used as an input slot to a function block

  You can configure BI and UI as binary inputs.

Procedure

1. Double-click *Event Program* in the tree, and then select the **Wire Sheet** view.
2. Open the “honIrmControl” palette.
3. From the palette, add control items to the wire sheet via drag & drop.
4. Create/change the control logic by applying desired steps such as connecting, adding, deleting, and moving icons, and/or adding and deleting connections.

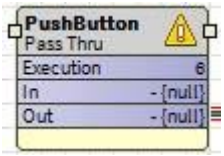


Synchronization Check

Any modifications on the Event Program wire sheet are detected by the control manager. Modifications can be any of the following:

- Added item
- Deleted item
- Moved item
- Added connection
- Deleted connection
- etc.

As result, if working in ´on demand` teaching mode, all modified items will be not in sync with the application in the controller and hence indicated by a yellow “warning” symbol on the item.

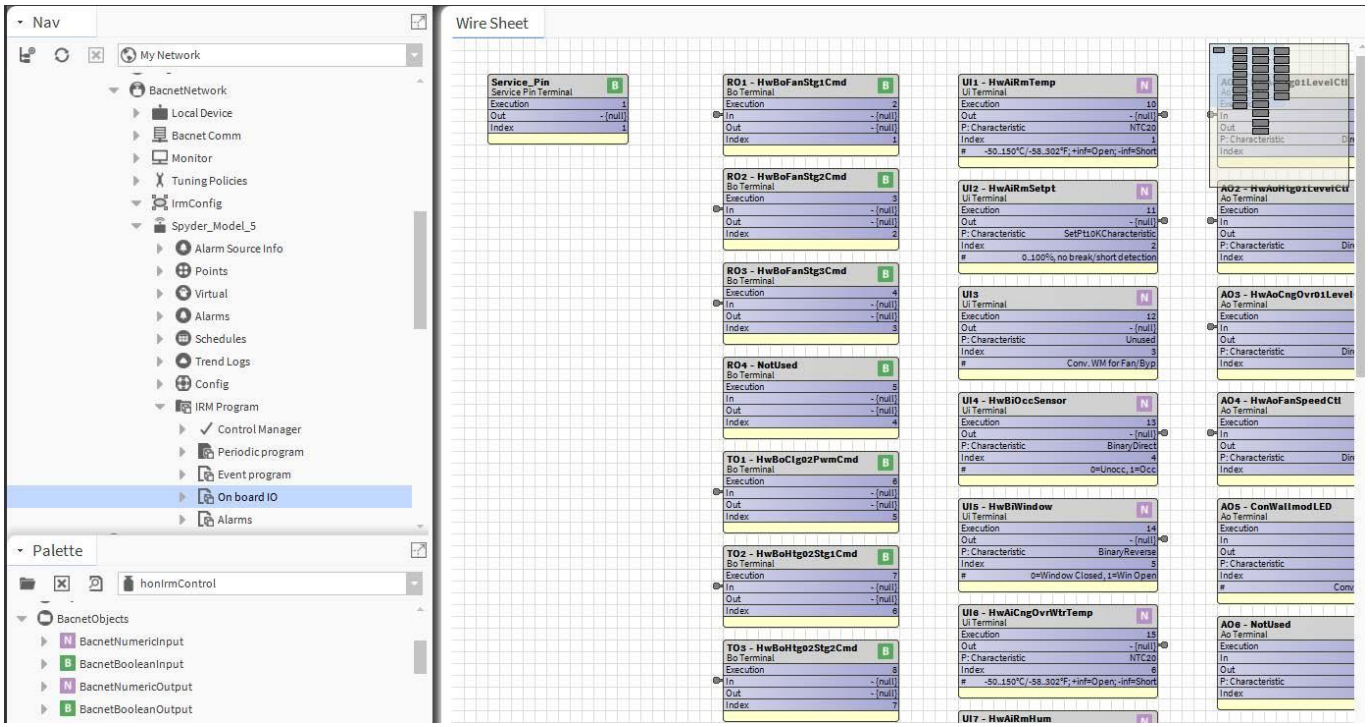


The application can be synchronized according to the active teaching mode, ´on demand` or ´immediate`. After synchronization, the warning symbols are removed and the items are indicated in its synchronized state. When working in teaching mode ´immediate` this happens immediately.

See also the description of the “The Control Manager”, p. 50.

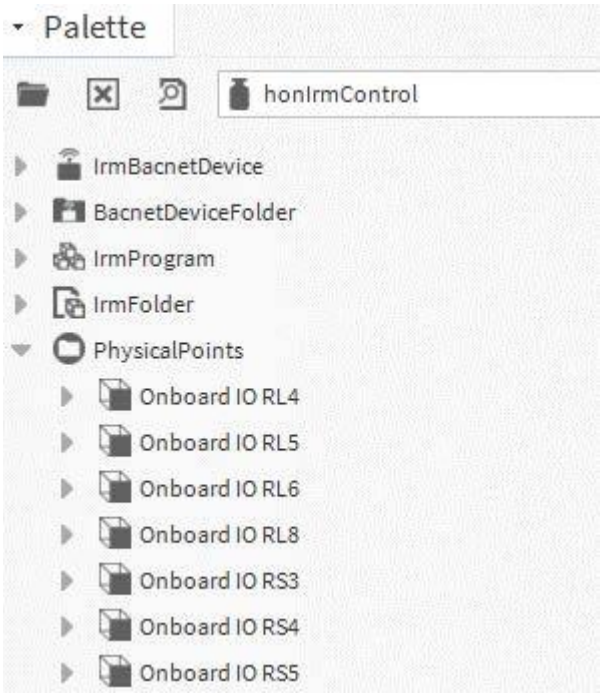
Onboard IO

Shows the hardware configuration to which the application is designed for. The hardware displayed here, can differ from the real hardware layout of the physical controller. For proper operation, the hardware of the project does not have to fit 100 % to the physical hardware of the used controller later. But, if the difference is too much, hardware compatibility is not guaranteed. The control manager will show incompatibilities and software give support and tells what to do for synchronicity.



Procedure

1. Double-click *On Board I/O* in the tree, and then select the **Wire Sheet** view.
2. Open the "honIrmControl" palette.
3. From the palette, add single physical terminals manually or a pre-defined template to the wire sheet via drag & drop. Pre-defined templates are matched to particular controller models. The following physical terminals and templates are available:



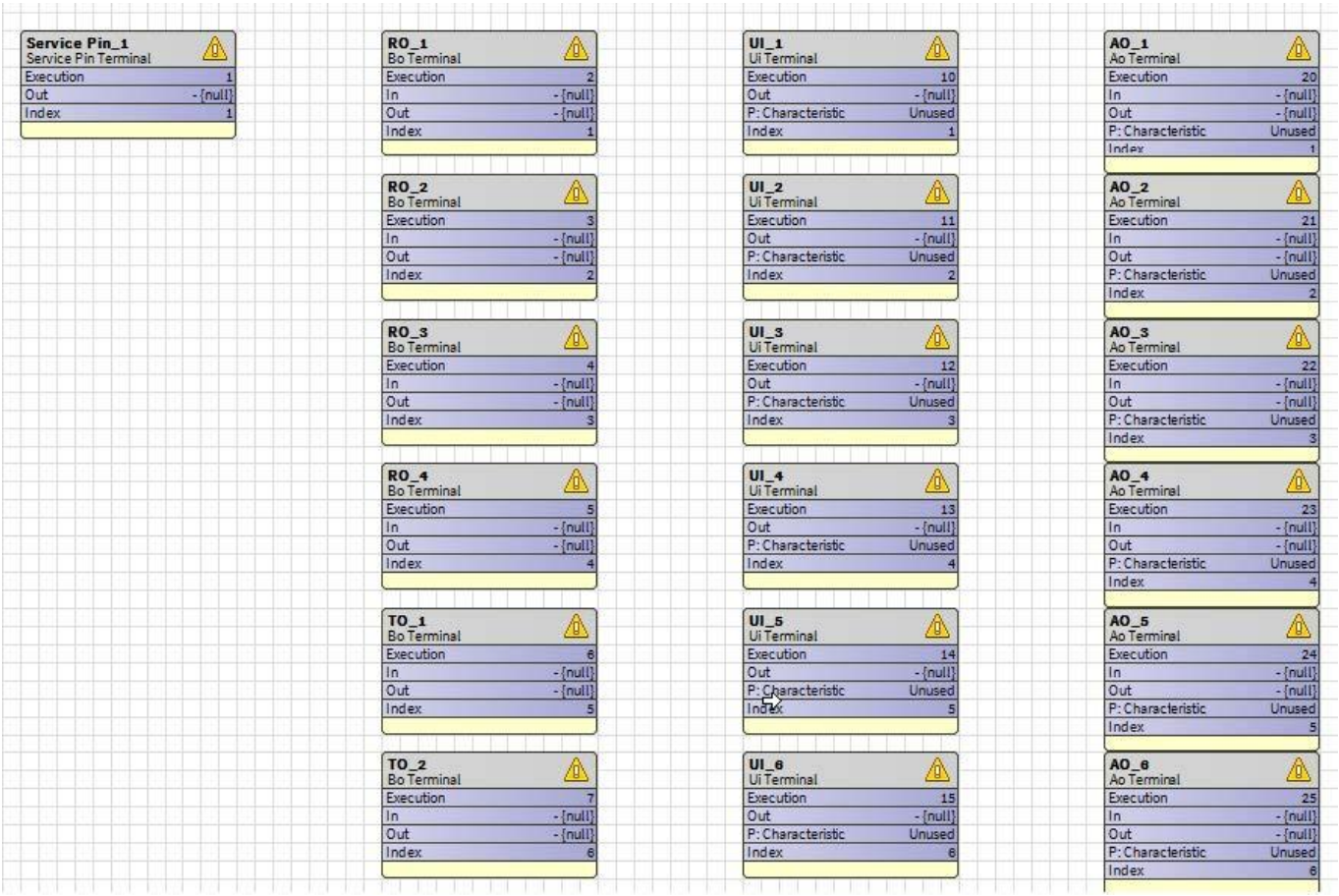
4. Create/change the layout by applying desired steps such as connecting, adding, deleting, and moving terminals, and/or adding and deleting connections.

Synchronization Check

Any modifications on the *On Board I/O* wire sheet are detected by the control manager. Modifications can be any of the following:

- Added item
- Deleted item
- Moved item
- Added connection
- Deleted connection
- etc.

As result, all modified items will be not in sync with the application in the controller and hence indicated by a yellow "warning" symbol on the item.



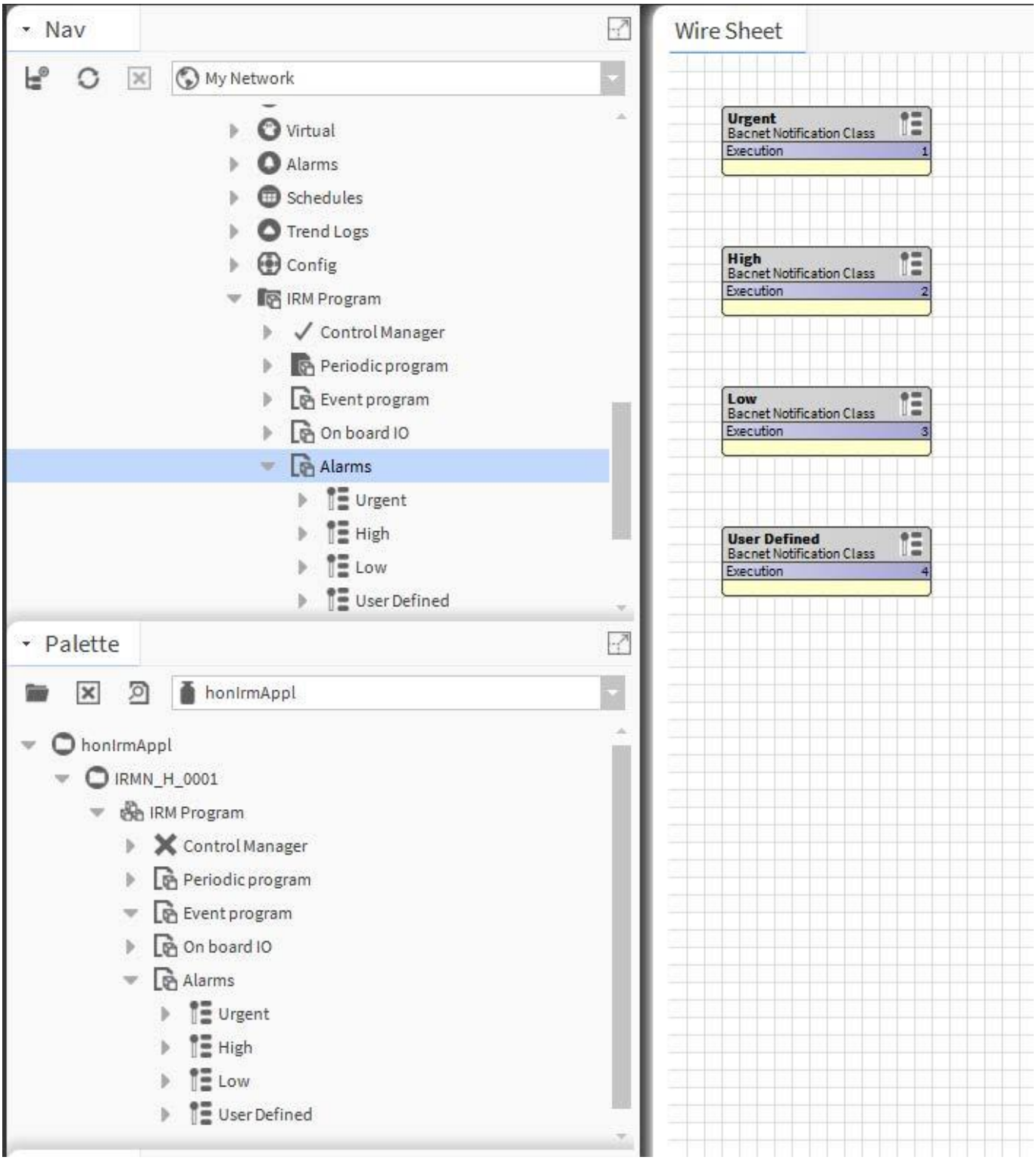
The application can be synchronized according to the active teaching mode, ´on demand` or ´immediate'. After synchronization, the warning symbols are removed and the items are indicated in its synchronized state.

See also the description of the “The Control Manager”, p. 50.

Alarms

Alarming

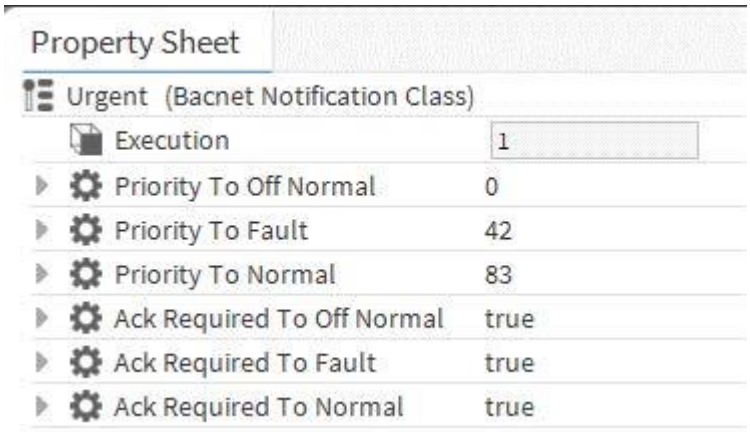
Provides the notification classes for establishing alarming



Synchronization Check

Any modifications such as adding, deleting or copying & pasting items on the *Alarms* wire sheet are not allowed.

You can only change the properties of the pre-defined notification class objects on the Property Sheet after you have double-clicked the icon.



Application Engineering Guidelines

Commissioning and Station Copier Usage

Before a controller can be used for the first time after initial installation or upgrade, it must be commissioned by loading the following components from the PC (localhost) into the controller (remote host):

- Niagara software
- License
- Application / station

When changes will be done later to the application on the PC, the station can be copied into the controller by using the Station Copier function. In this case, a new commissioning is not necessary.

For detailed description, please refer to the CIPer Model 50 Controller User Guide: 31-00198-01.

Final Step after Application Engineering

IMPORTANT!

It is strongly recommended to swap out the application in any case after finishing the engineering in online or offline mode.

Swapping out the application avoids additional communication between Spyder Model 5 Engineering Tool and controller and reduces bus traffic.

For details about Swapping, please refer to the “Swapping IRM Program“ section, p. 90.

Memory and Folder Usage

It is recommended to monitor the memory usage on the Control Manager level and to avoid to exceed the limit of 80 %.

To keep the memory usage as low as possible, it is recommended to optimize application engineering by keeping the number of folders as low as possible and the number of added function blocks per folder as high as possible in relation to the required result of the logic.

The max. number of function blocks per folder is 150. If you want to use more than 150 function blocks, please split them into two or more folders.

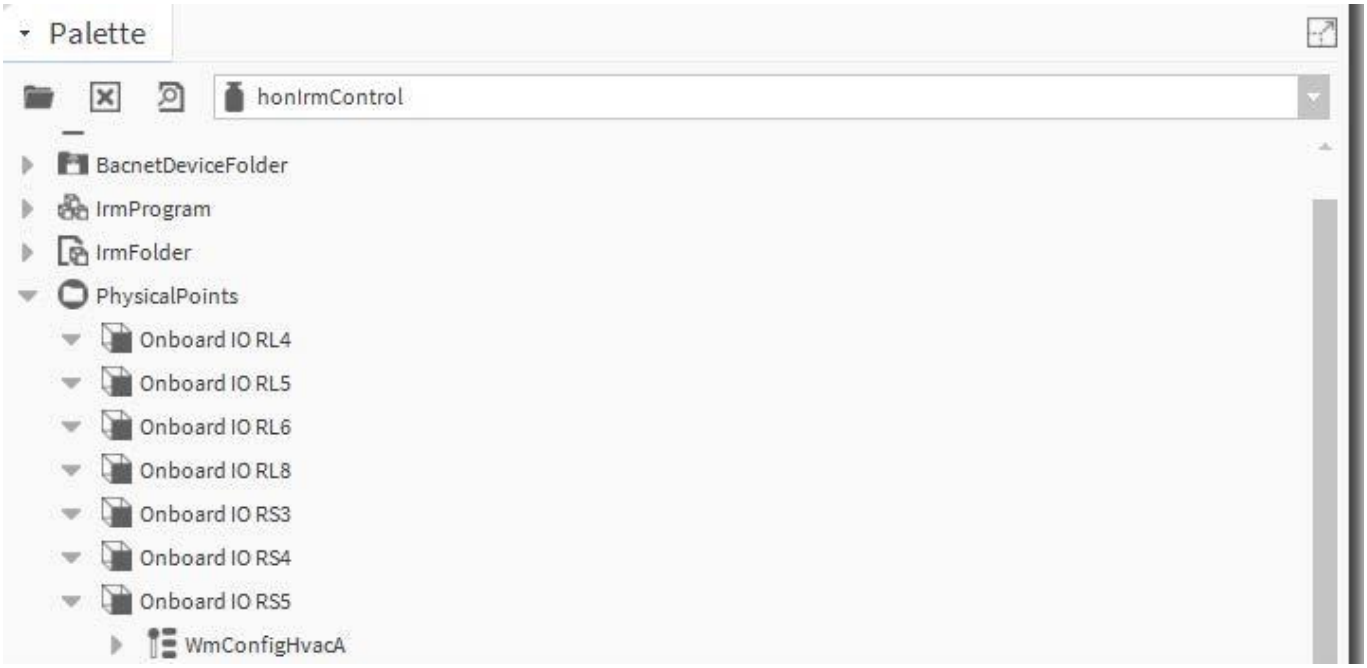
Memory Usage“ section, p. 51.

IRM Function Blocks and External Application Components

The IRM program accepts IRM control components only. Direct connection between IRM control function blocks with Niagara components by using the Niagara Link function does not work. External communication should be established via BACnet components which link the IRM function blocks to the Niagara components.

Application Templates Usage

For quick engineering and avoiding hardware compatibility problems, use pre-defined onboard I/O configurations (templates) provided in the *PhysicalPoints* folder of the **honIrmControl** palette. When using any of these, make sure that they fit to the physical device model.



Reference Datapoints Usage

General

If the control and monitoring system contains more than one BACnet device (controller), the devices communicate with one another via the BACnet MSTP protocol. This enables one controller to read values from other controllers and set values on other controllers.

This data communication is realized via so-called reference input and reference output points which are assigned to physical or value BACnet points.

Assignments can be done manually (manual creation and mapping) or automatically (automatic creation via drag & drop).

For manual creation and mapping, the device instance, object instance and object ID of the physical and value BACnet points are entered manually. For automatic creation via drag & drop, the device instance, object instance and object ID of the physical and value BACnet points of the physical and value BACnet points are carried over automatically.

NOTE: The reference input and output function provided by the Spyder Model 5 Engineering Tool is a proprietary Honeywell BACnet function.

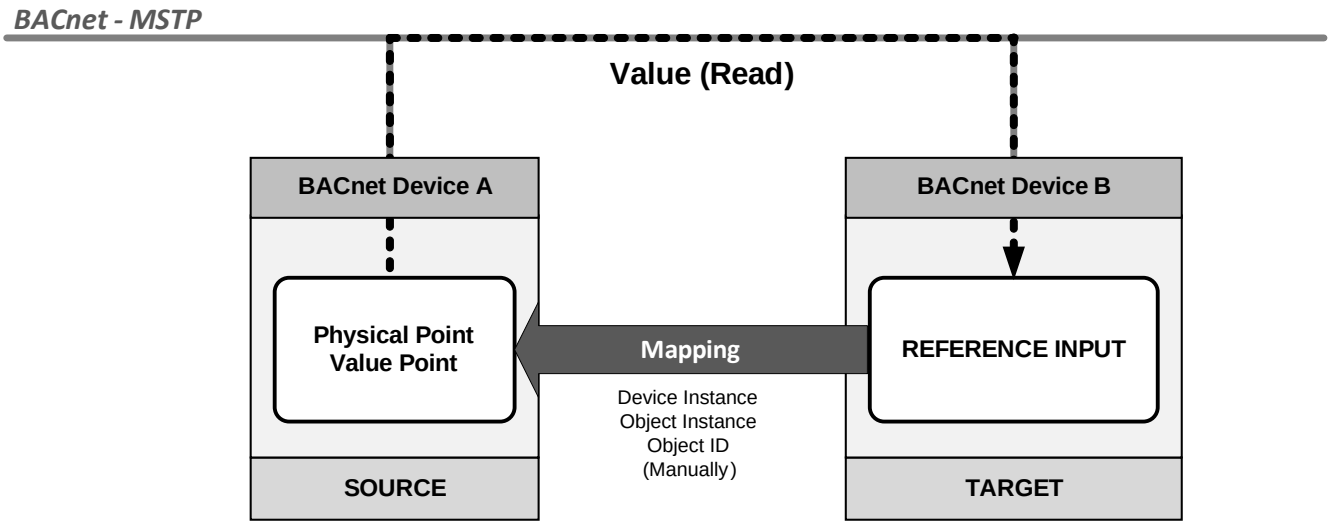


Fig. 7. Data Exchange via Manual Mapping of Reference Input Point to Physical / Value Point

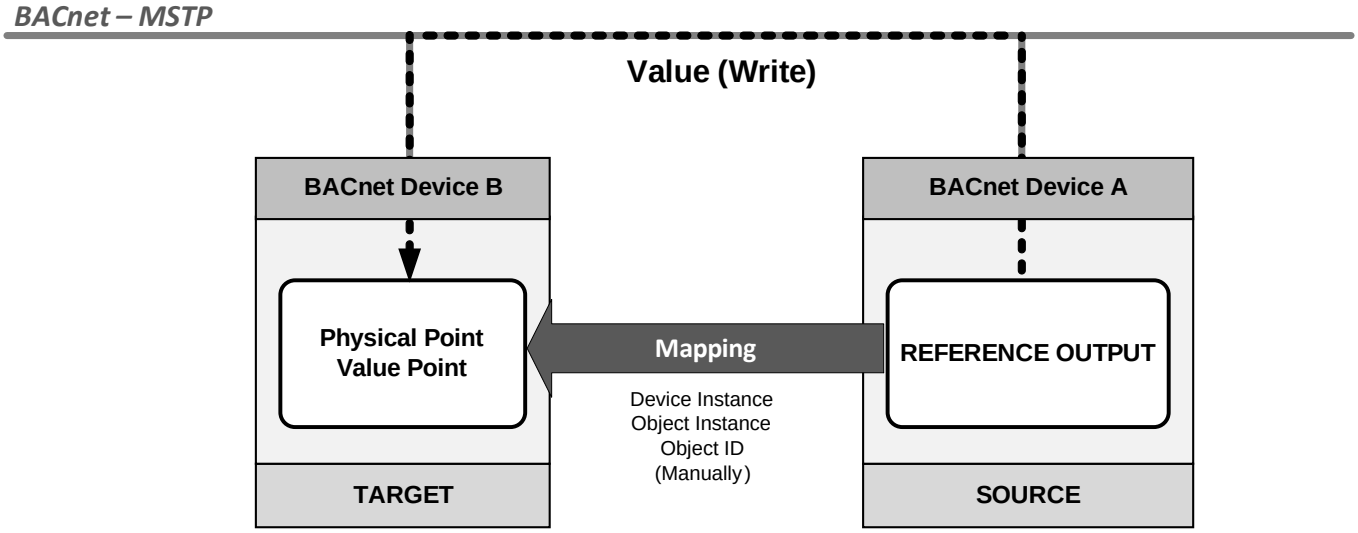


Fig. 8. Data Exchange via Manual Mapping of Reference Output Point to Physical / Value Point

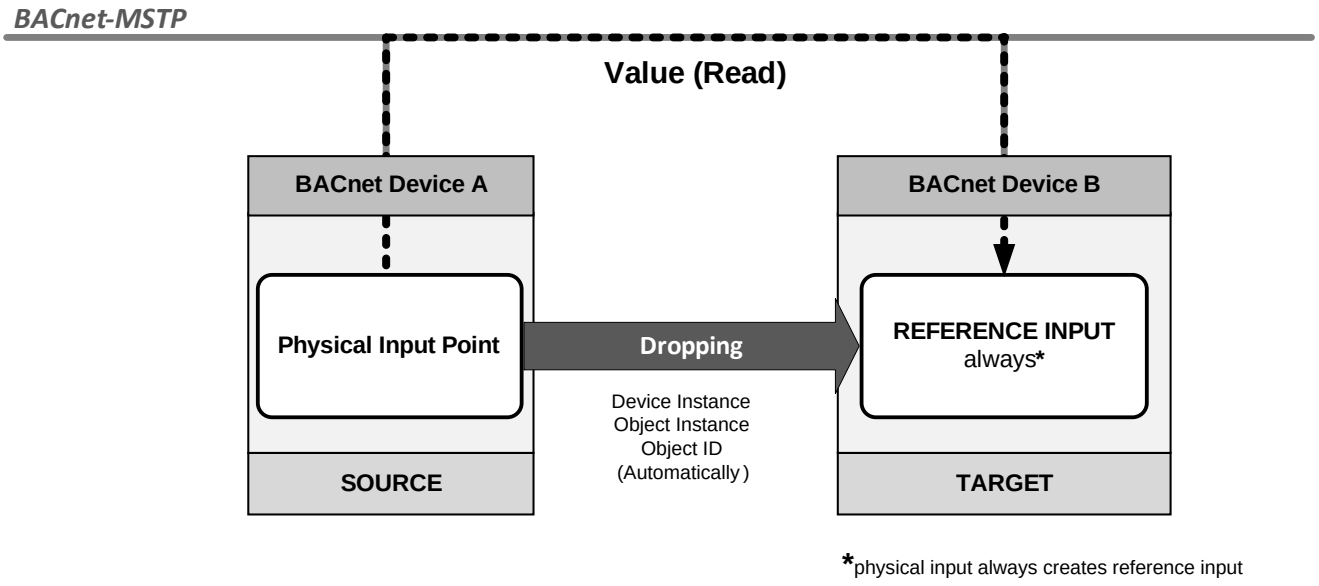


Fig. 9. Data Exchange via Dropping of Physical Input Point

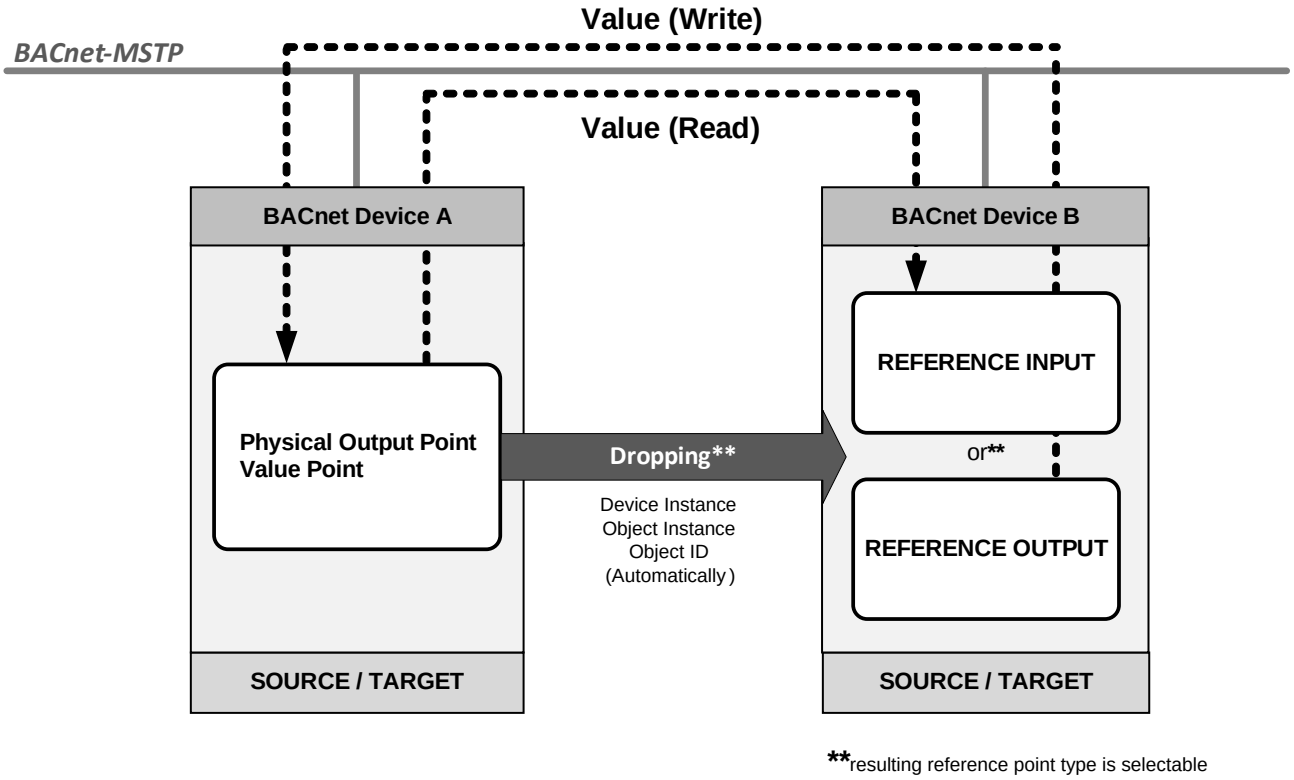


Fig. 10. Data Exchange via Dropping of Physical Output Point or Value Point

Example: There are two controllers (A and B) in two different rooms on the BACnet MSTP bus. Controller A has an outside air temperature sensor connected. The outside air temperature value sensed by controller A should be provided to controller B.

There is a 3rd party BACnet MSTP wall module connected to an IRM controller which should process values coming from the wall module.

Read / Write Directions

A reference input point (target) is used for reading a value from a physical or value BACnet point (source)

A reference output point (source) is used for writing a value to a physical or value point (target).

There are two ways for applying reference points in the application:

- Manual creation and mapping of reference points to BACnet points (A)
- Automatic creation and mapping of reference points via Drag & drop of BACnet points (B)

COV and Polling

The controller that contains the reference point subscribes to the mapped physical or value point if COV reporting possible. If COV reporting is not possible, it polls the mapped physical or value point.

Manual Reference Point Creation and Mapping

The reference input-BACnet point connection is established by manual creation and mapping of the reference input point (target) to the physical or value BACnet point (source).

The reference output connection is established by manual creation mapping of the reference output point (source) to the physical or value BACnet point (target).

In both cases, the following device and object information from the BACnet source or target points must be entered during the mapping:

- Device Instance (ID)
- Object type (BACnet point type) = object ID
- Object Instance = object ID

Automatic Reference Point Creation and Mapping via Drag & Drop

Reference points can be automatically created and mapped to physical or value BACnet points by dragging & dropping the physical or value BACnet points onto the wiresheet. In this case, the corresponding device and object information comes with the corresponding BACnet point and must not be entered manually.

Master – Slave Configurations

A typical application scenario for the usage of reference points is a master-slave configuration consisting of one master controller connected to multiple slave controllers and/or 3rd Party BACnet devices. The master incorporates the main control logic, writes values to the slave controllers and/or 3rd Party BACnet devices and receives values from the slave controllers and/or 3rd Party BACnet devices.

Example:

- The master controller:
  - reads the window contact status [EffWindow] from the slave controller via RefIn
  - reads the room temperature [ExtWmRmTemp] from the externa BACnet Wall Module via RefIn
  - writes the occupancy status [EffOccMd] to an external BACnet Wall Module via RefOut
- The slave controller:
  - reads the fan output status [FanStage] of the staged fan (slow, medium, high) from the master controller via RefIn (see figure next page).

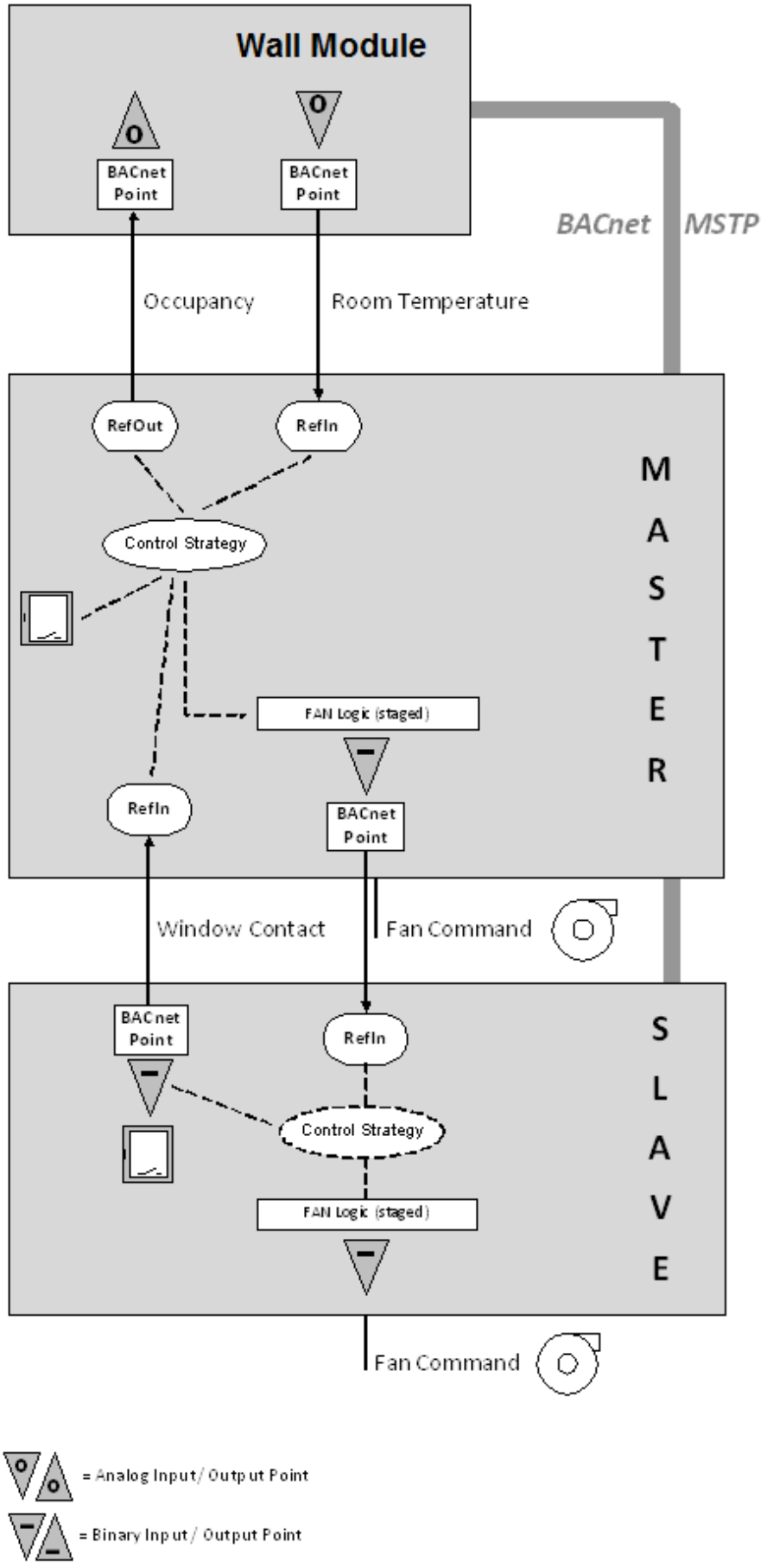


Fig. 11. Master-Slave Configuration using Reference Points

Manual Creation and Mapping of Reference Points (A)

Example:

Two IRM controllers and a wall module are on the MSTP bus in master-slave configuration:

- WEB-RL6N (Master)
- WEB-RS5N (Slave)
- External BACnet wall module
  - The WEB-RL6N master controller:
    - reads the window contact status [EffWindow] from the slave controller via RefIn
    - reads the room temperature [ExtWmRmTemp] from the BACnet Wall Module via RefIn
    - writes the occupancy status [EffOccMd] to an external BACnet Wall Module via RefOut
  - The RS5N slave controller:
    - reads the fan output status [FanStage] of the staged fan (slow, medium, high) from the master controller via RefIn

Reference Inputs

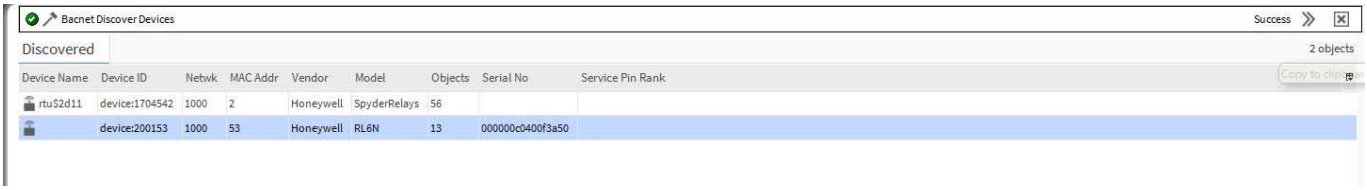
Based on the example above, the reference input functions of the WEB-RL6N master controller are described in the following.

- The WEB-RL6N master controller
  - reads the window contact status [EffWindow] from the slave controller via RefIn
  - reads the room temperature [ExtWmRmTemp] from the BACnet Wall Module via RefIn

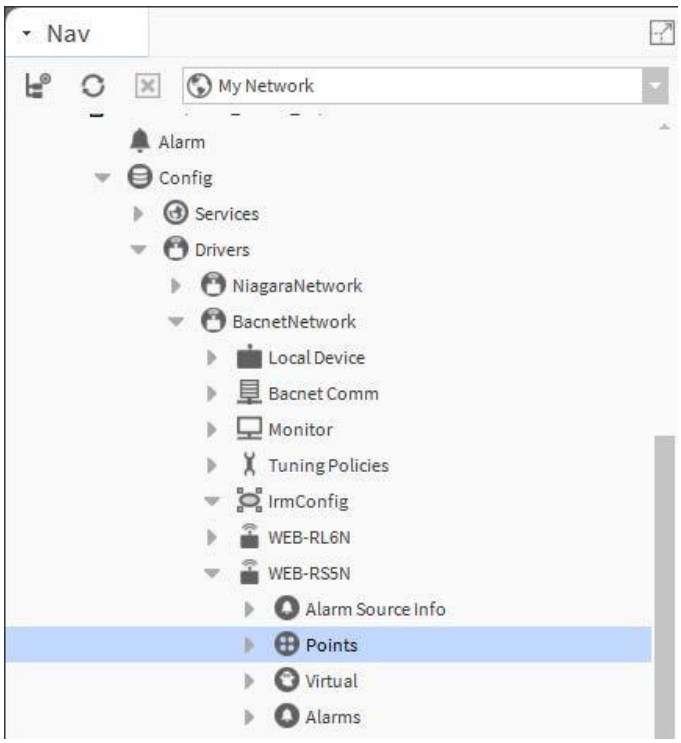
NOTE: The read function of the WEB-RS5N slave controller is described using the automatic creation via drag & drop (see “Automatic Creation and Mapping of Reference Points via Drag & drop (B)“ section, p. 76).

Procedure

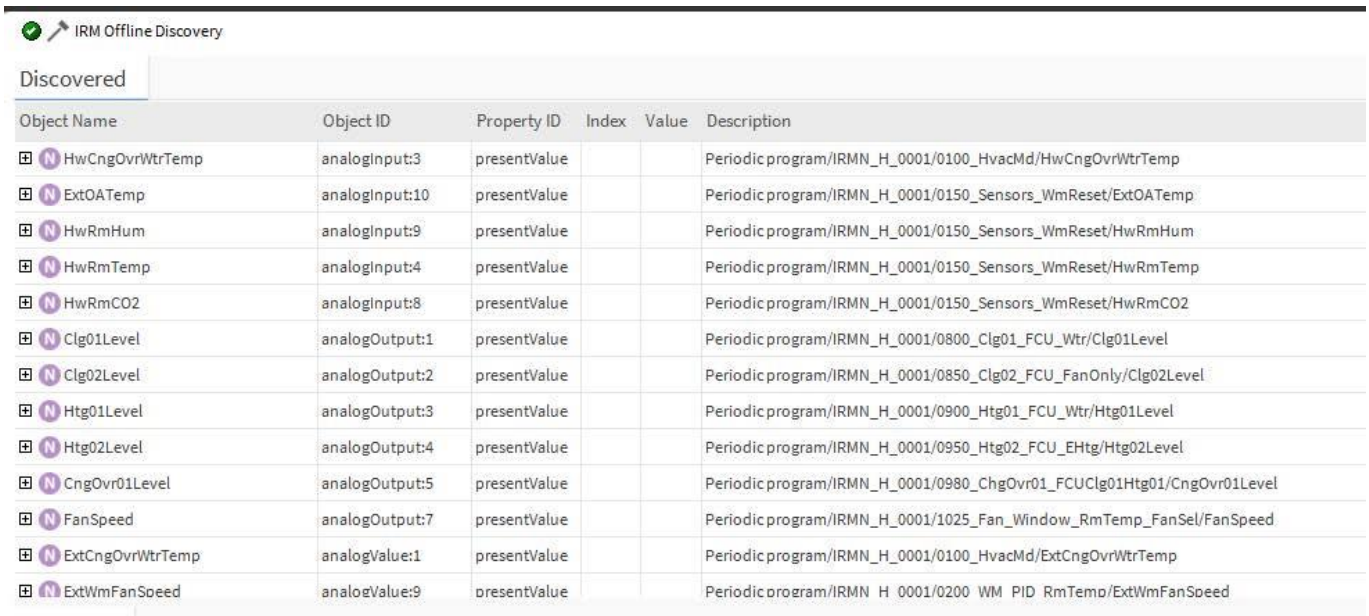
1. Note the device IDs of the WEB-RL6N master and WEB-RS5N slave controllers, in this case ´5003` and ´5001` (displayed in the Discovered and Database pane).



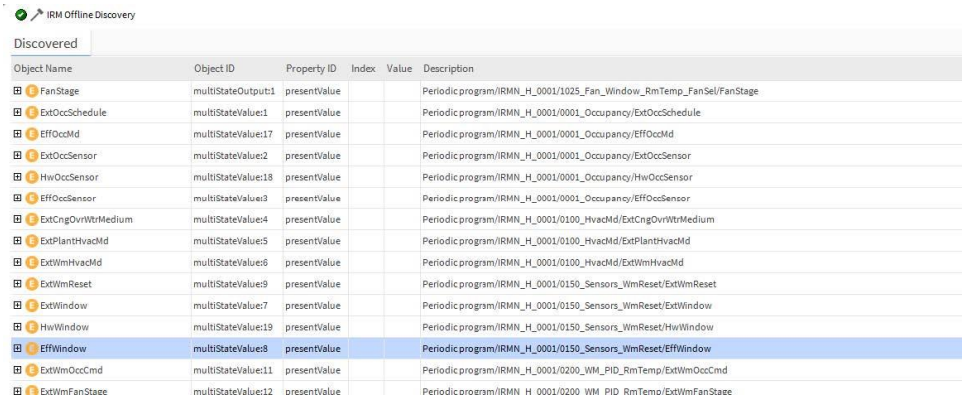
2. In the *Nav* tree, expand the BACnet network and browse to the *Points* folder of the RS5N slave controller.



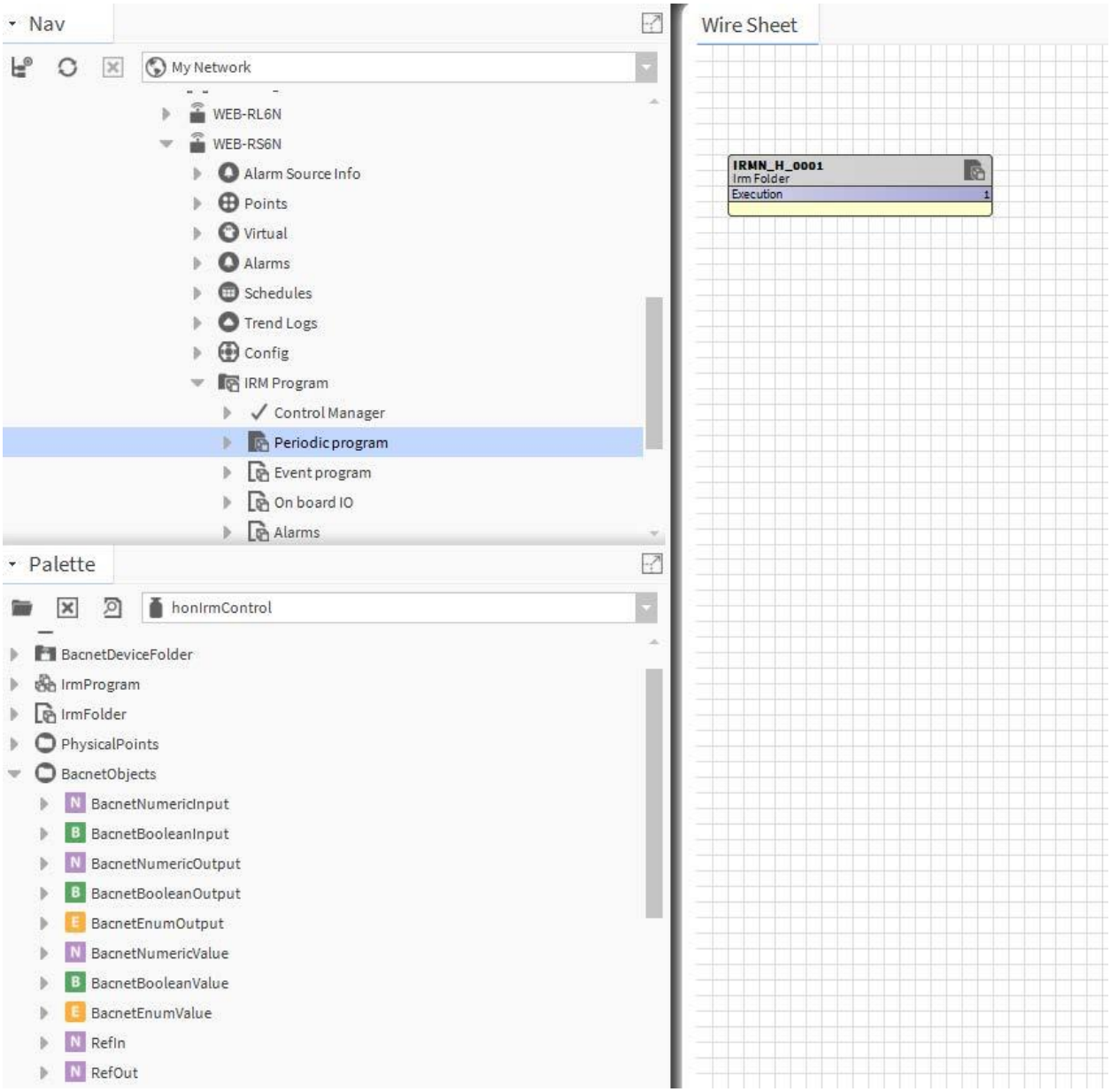
3. Double-click the *Points* folder and discover the points by clicking **Discover** on the bottom.



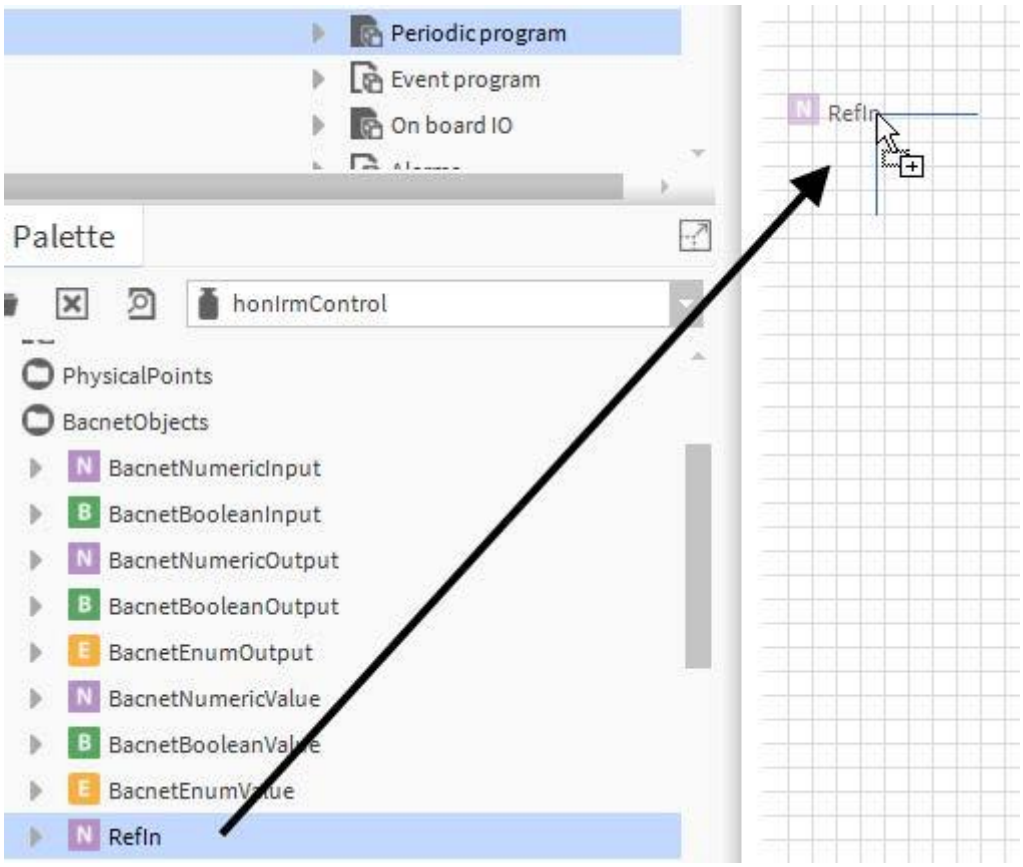
4. Sort the discovered points by clicking on the **Object Name** column.



5. In the point list, scroll to the ´EffWindow` BACnet point.
6. Note the Object ID, in this case ´multiStateValue:8`.
7. In the *Nav* tree, browse to the WEB-RL6N master controller.
8. Expand the IRM Program folder.
9. Double-click the control program folder, *Periodic Program* or *Event Program*, to which you want to add the reference input point.

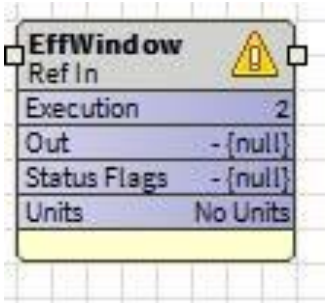


10. In the *honIrmControl* palette, expand BacnetObjects, and then drag & drop the **RefIn** BACnet object to the wire sheet.

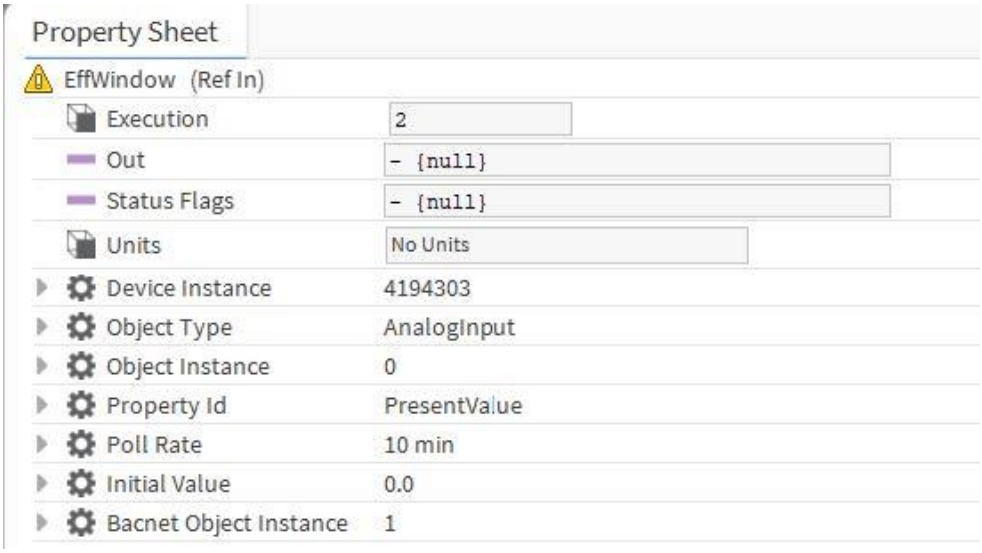


11. In the *Name* dialog box, change the name to ´EffWindow`, and then click **OK**.

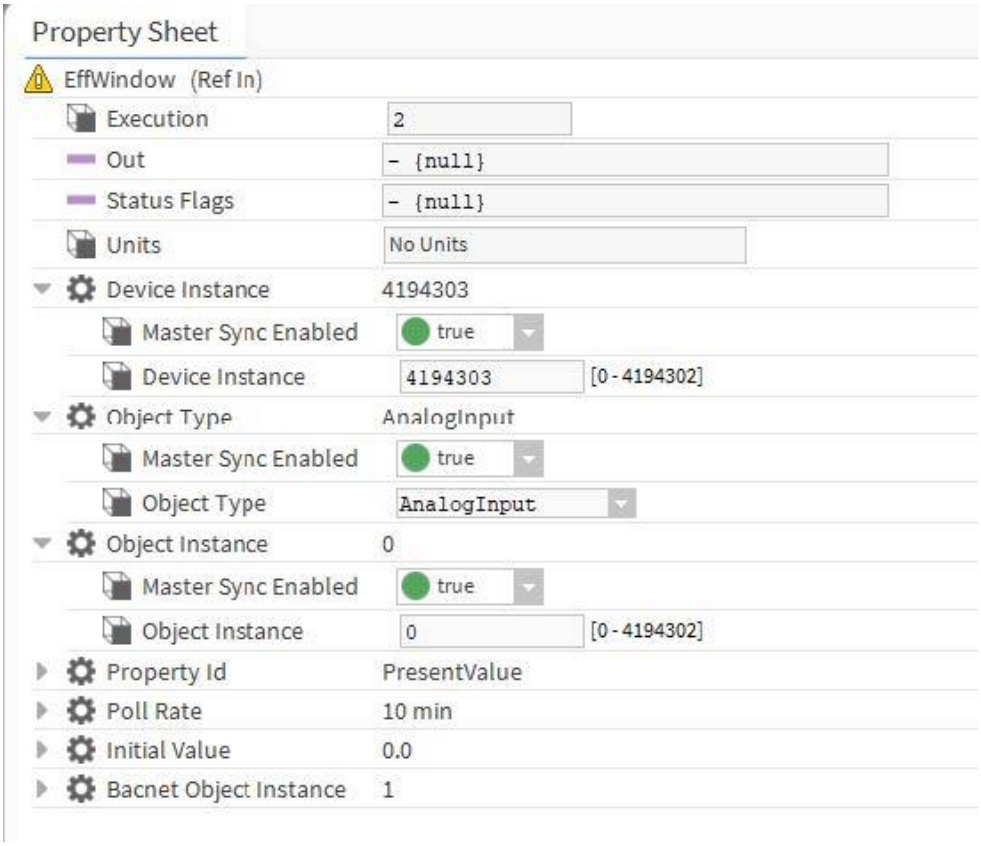
12. The reference input is added to the wire sheet.



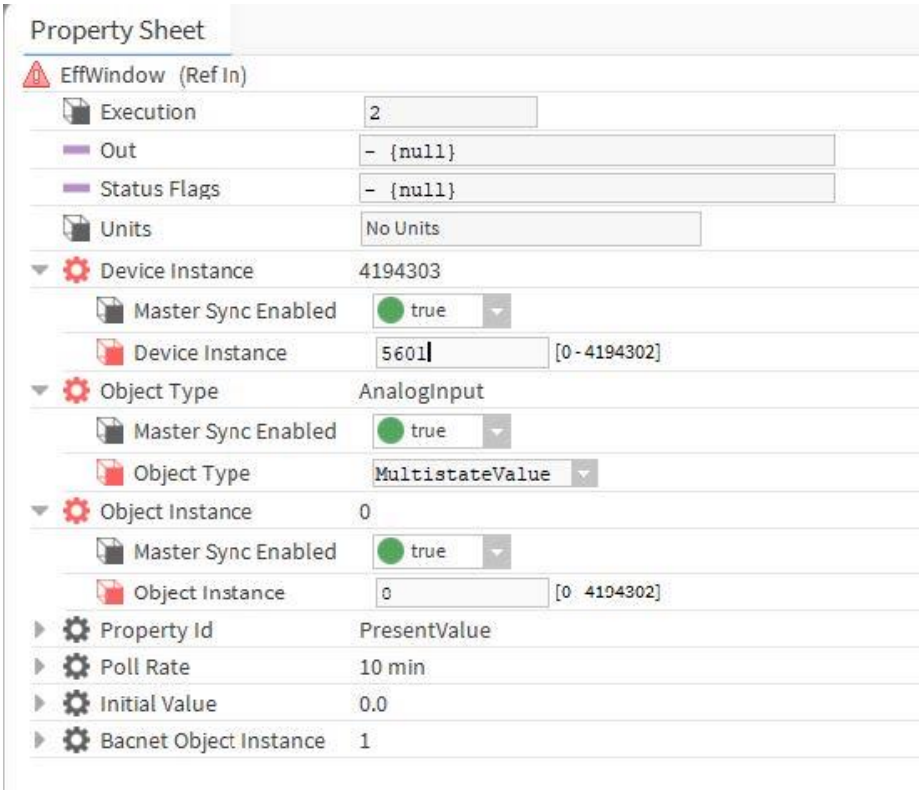
13. Double-click the symbol to display the property sheet for the reference input.



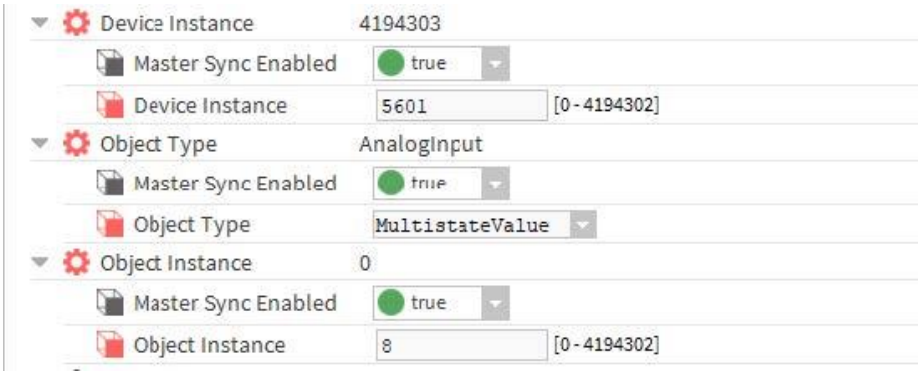
14. On the *Property Sheet*, expand **Device Instance**, **Object Type**, and **Object Instance**.



15. Enter the values of the wall module and the ´EffWindow` multistate value point as shown in step 1 and 6.
    - Device Instance: device ID = 5001
    - Object Type: object ID = multistateValue
    - Object Instance: object ID = 8



16. Click **Save**.



17. The reference input in the WEB-RL6N master controller is mapped to the multistate value point ´EffWindow` in the WEB-RS5N controller which provides the window contact status.

18. Apply the procedure in the same way for:

- Reading the room temperature by mapping a reference input from the master controller to the analog value [ExtWmRmTemp] of the external BACnet Wall Module
- Reading the fan stage by mapping a reference input from the WEB-RS5N slave controller to the multistate output [FanStage**]** of the WEB-RL6N master controller.

Alternatively, you can use the “automatic reference point creation and mapping via drag & drop” method which is described in the section “Automatic Creation and Mapping of Reference Points via Drag & drop (B)“, p. 76. This section uses fan stage reading as an example.

19. For writing the occupancy status from the master controller to the external BACnet wall module, please refer to the Reference Outputs“section, p. 71.
20. If desired, you can change the setting for master sync or other settings.

Master-Slave Synchronization

If Master Sync Enabled is true, this property will be synchronized between master and slaves in case a master sync operation is performed. The source device setting will be synced to a target device during master sync operation. If the source device has set the flag to false, then this parameter will not be synced to all target devices.

Reference Outputs

Example:

Two IRM controllers and a wall module are on the MSTP bus in master-slave configuration:

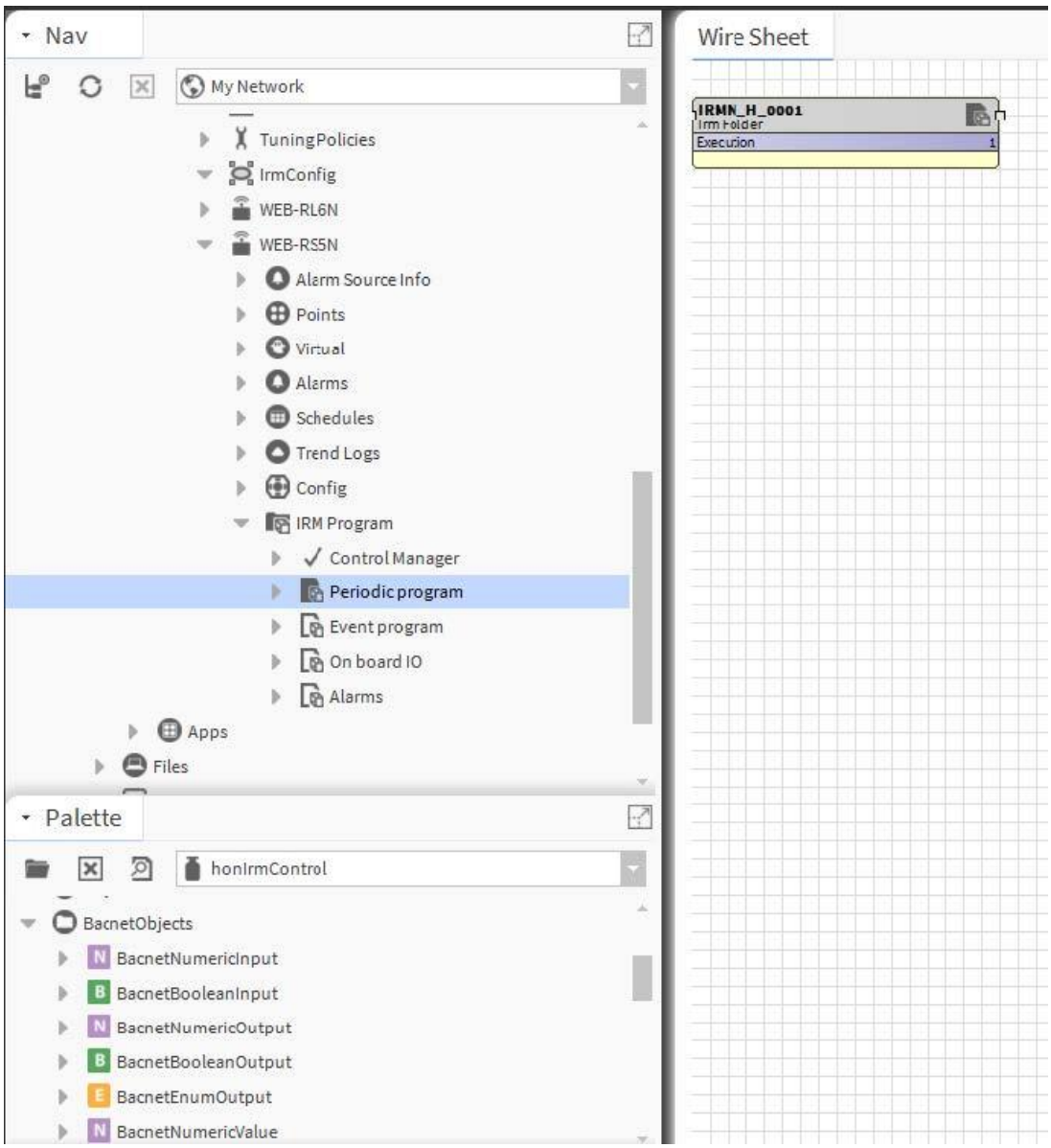
- WEB-RL6N (Master)
- WEB-RS5N (Slave)

In the following procedure, the reference output function of the WEB-RL6N master controller is described.

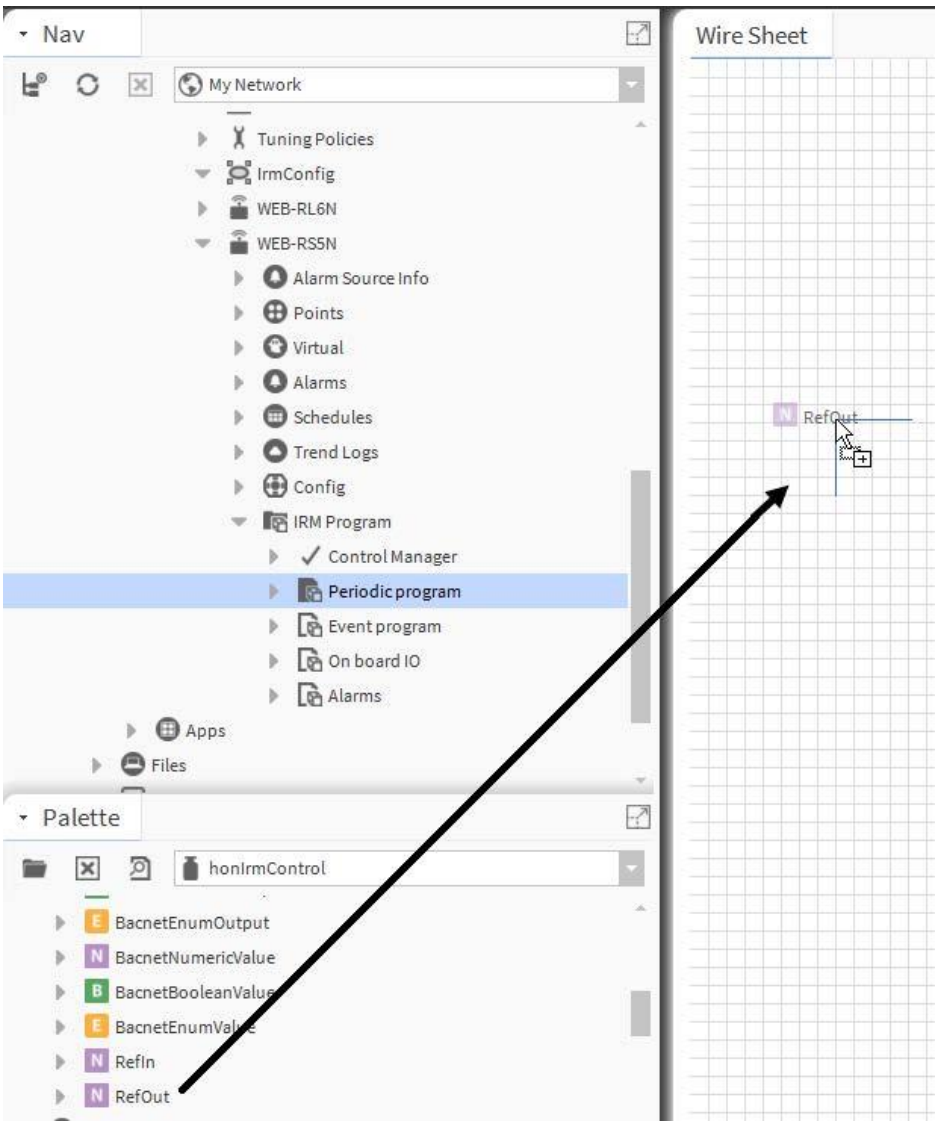
- The WEB-RL6N master controller writes the occupancy status [EffOccMd] to an external BACnet wall module via RefOut

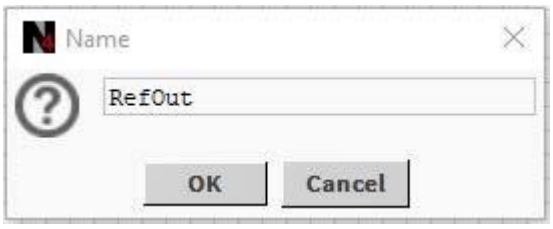
Procedure

1. Note the device ID of the external BACnet wall module, in this case, e.g. ´5008` (displayed in the *Discovered* and *Database* pane).
1. In the *Nav* tree, expand the BACnet network and browse to the *Points* folder of the wall module.
2. Double-click the *Points* folder and discover the points by clicking **Discover** on the bottom.
3. Sort the discovered points by clicking on the **Object Name** column.
4. In the point list, scroll to the ´EffOccMd` BACnet point.
5. Note the Object ID, in this case, e.g. ´multiStateValue:13`.
6. In the *Nav* tree, browse to the WEB-RL6N master controller.
7. Expand the IRM *Program* folder.
8. Double-click the control program folder, *Periodic Program* or *Event Program,* to which you want to add the reference output point.



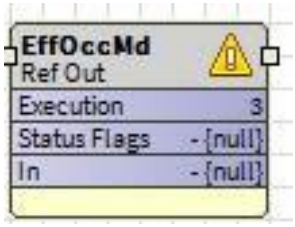
9. In the *honIrmControl* palette, expand BacnetObjects, and then drag & drop the **RefOut** BACnet object to the wire sheet.



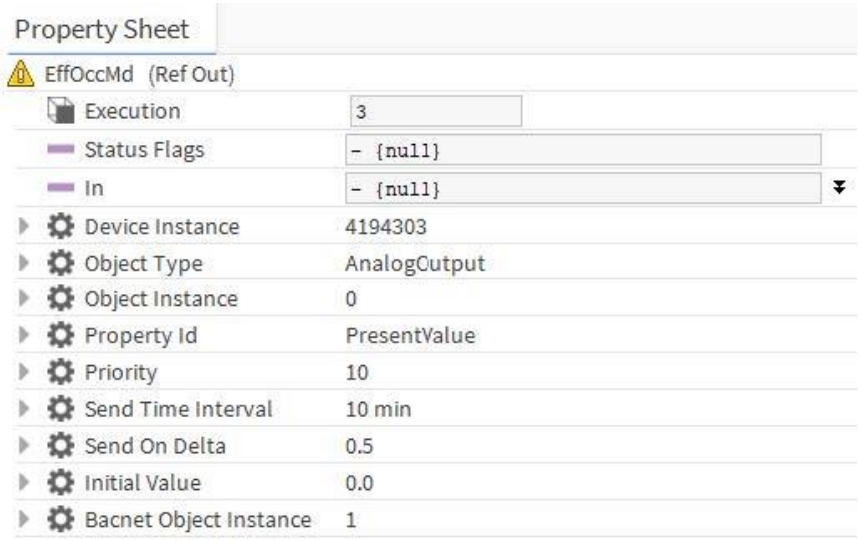


10. In the *Name* dialog box, change the name to ´EffOccMd`, and then click **OK**.

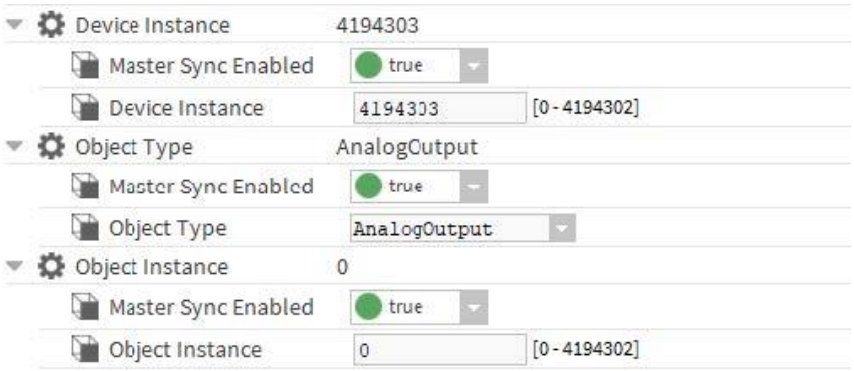
11. The reference output is added to the wire sheet.



12. Double-click the symbol to display the property sheet for the reference output.



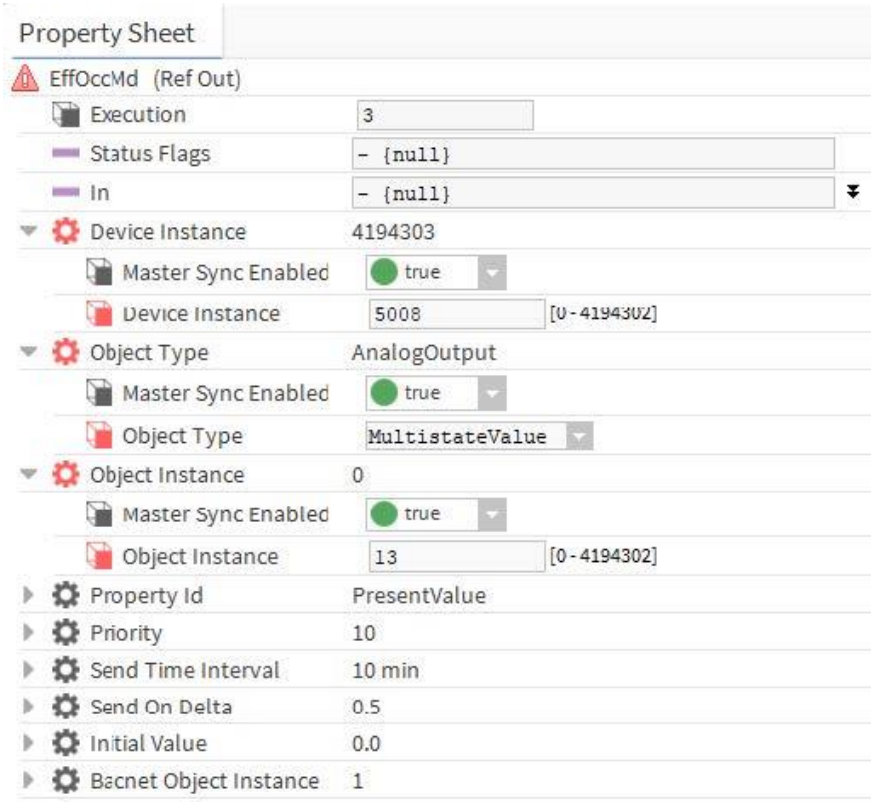
13. On the *Property Sheet*, expand **Device Instance**, **Object Type**, and **Object Instance**.



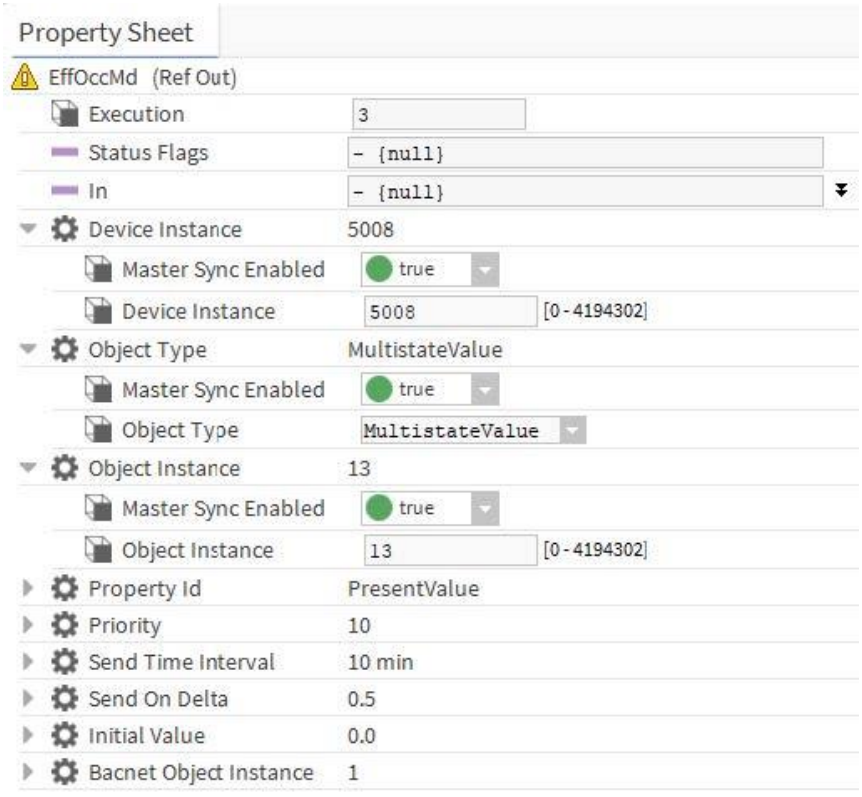
14. Enter the values of the external BACnet wall module and the ´EffOccMd` multistate value point as shown in step 1 and 4.

- Device Instance: device ID = 5008
- Object Type: object ID = multistateValue
- Object Instance: object ID = 13

15. Click **Save**.



16. The reference input in the WEB-RL6N master controller is mapped to the multistate value point ´EffOccMd` in the wall module.



17. If desired, you can change the setting for master sync or other settings.

Master-Slave Synchronization

If Master Sync Enabled is true, this property will be synchronized between master and slaves in case a master sync operation is performed. The source device setting will be synced to a target device during master sync operation. If the source device has set the flag to false, then this parameter will not be synced to all target devices.

Automatic Creation and Mapping of Reference Points via Drag & drop (B)

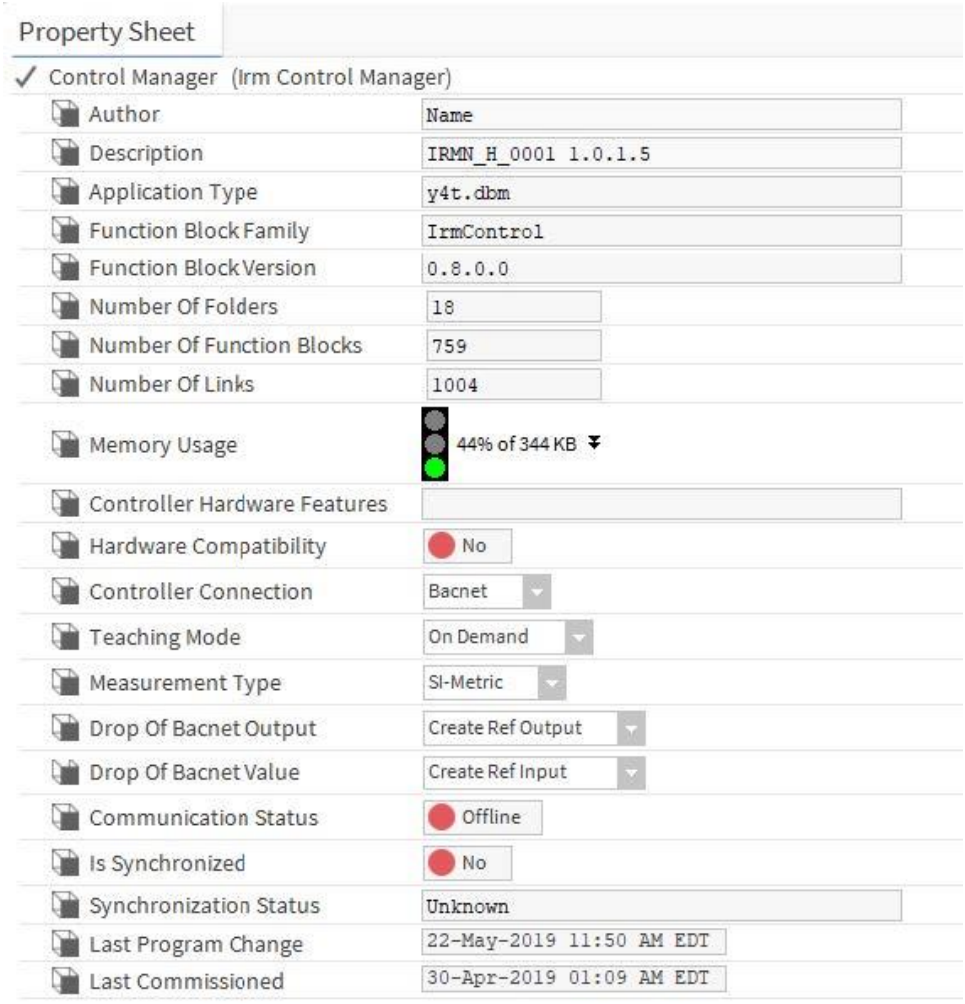
Niagara allows the automatic creation of reference points when BACnet points are dragged & dropped onto the wiresheet. Thus, you do not have to enter the values for device instance, object type, and object instance of the target controller and point manually, since they come with the BACnet point automatically.

This automatic creation is always executed whenever a BACnet point is dragged & dropped onto the wiresheet of another controller or external BACnet device.

The drop result, that is, which type of reference point, input or output, is to be created, can be defined in the control manager as described in the following.

Drop Settings in Control Manager

When using reference points in the application via drag & drop, the dropping result depends on the pre-setting for drop in the control manager of the device to which the BACnet point is dragged & dropped.



In the control manager, the result for dropping a BACnet output and a BACnet value point can be selected. For both point types, either a reference output or a reference input can be selected as drop result.

NOTE: Input points cannot be overridden by another controller. Hence when dropping an input onto the wiresheet, always a reference input is created and a drop setting for inputs in the control manager is not necessary.

Example:

Two IRM controllers and a wall module are on the MSTP bus in master-slave configuration:

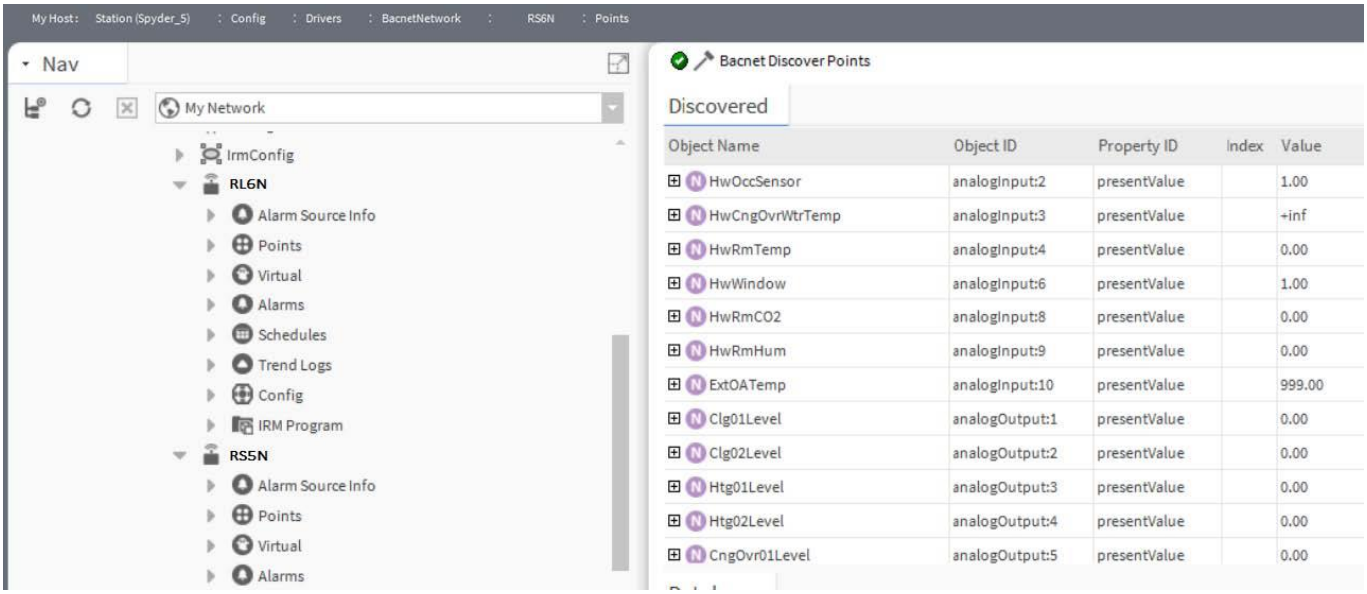
- WEB-RL6N (Master)
- RS5N (Slave)
- External BACnet wall modul
- The WEB-RL6N master controller:
  - reads the window contact status [EffWindow] from the slave controller via RefIn
  - reads the room temperature [ExtWmRmTemp] from the BACnet Wall Module via RefIn
  - writes the occupancy status [EffOccMd] to an external BACnet Wall Module via RefOut
  - The RS5N slave controller:
  - reads the fan output status [FanStage] of the staged fan (slow, medium, high) from the master controller via RefIn

Based on the example above, the reference input function of the RS5N slave controller is described.

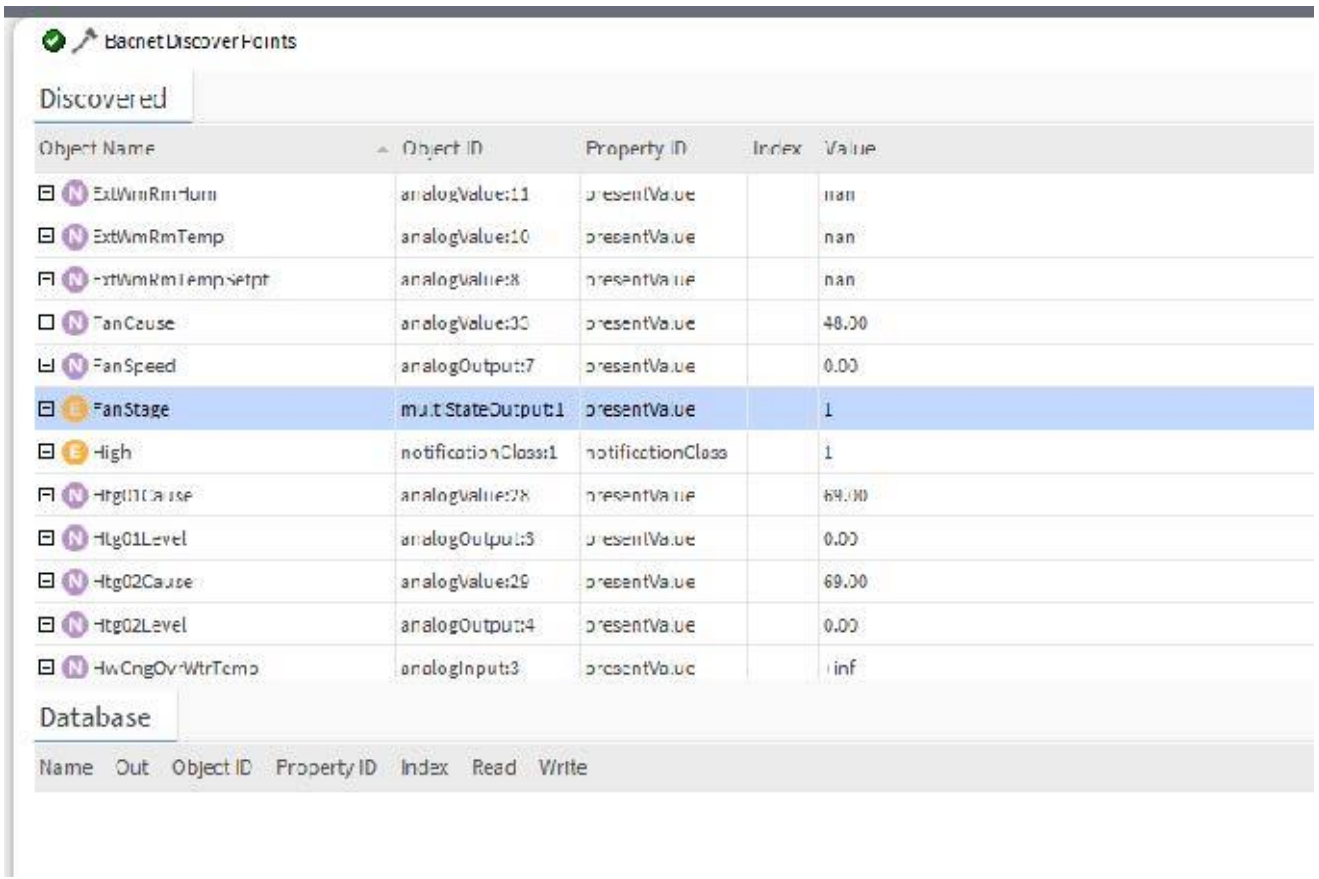
- The RS5N slave controller:
- reads the fan output status [FanStage] of the staged fan (slow, medium, high) from the master controller via RefIn.

Procedure

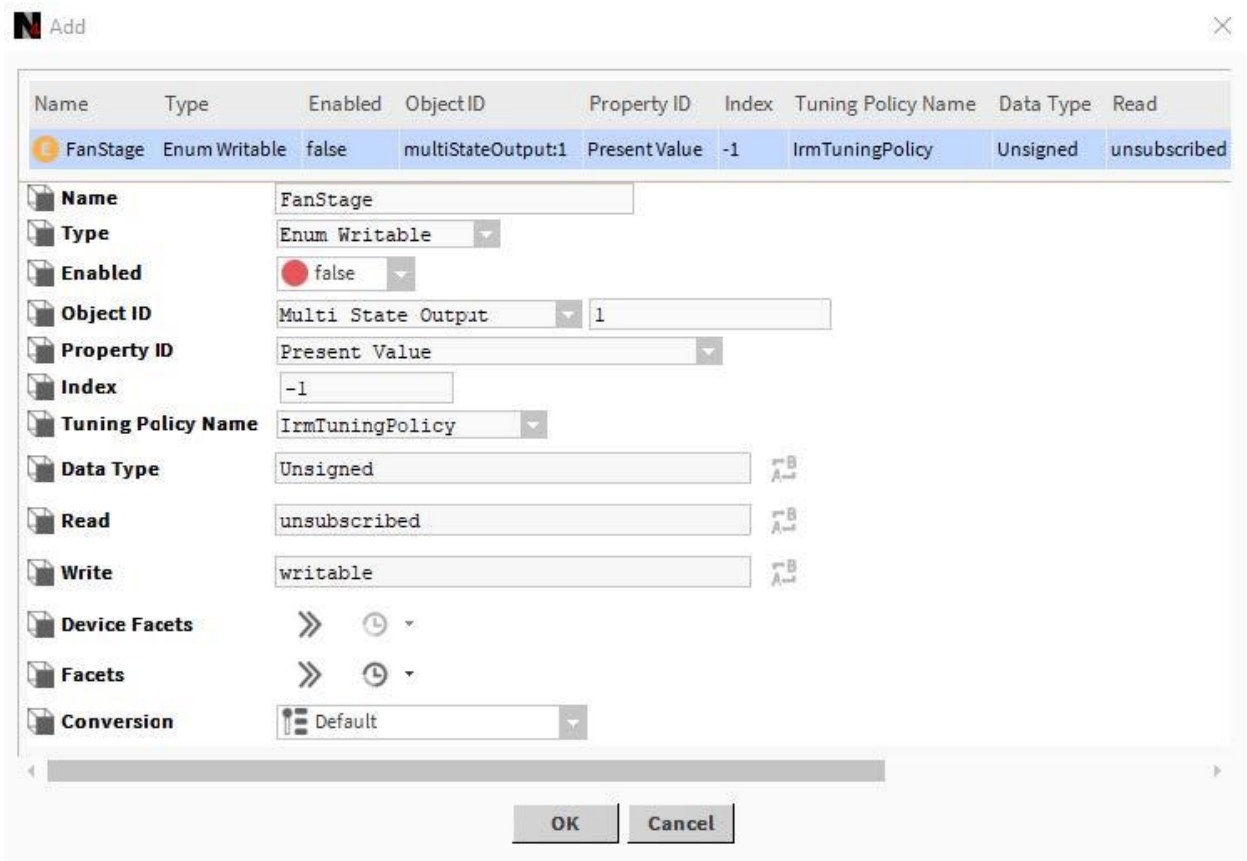
1. In the *Nav* tree, expand the BACnet network and browse to the *Points* folder of the WEB-RL6N master controller.
2. Double-click the *Points* folder and discover the points by clicking **Discover** on the bottom.



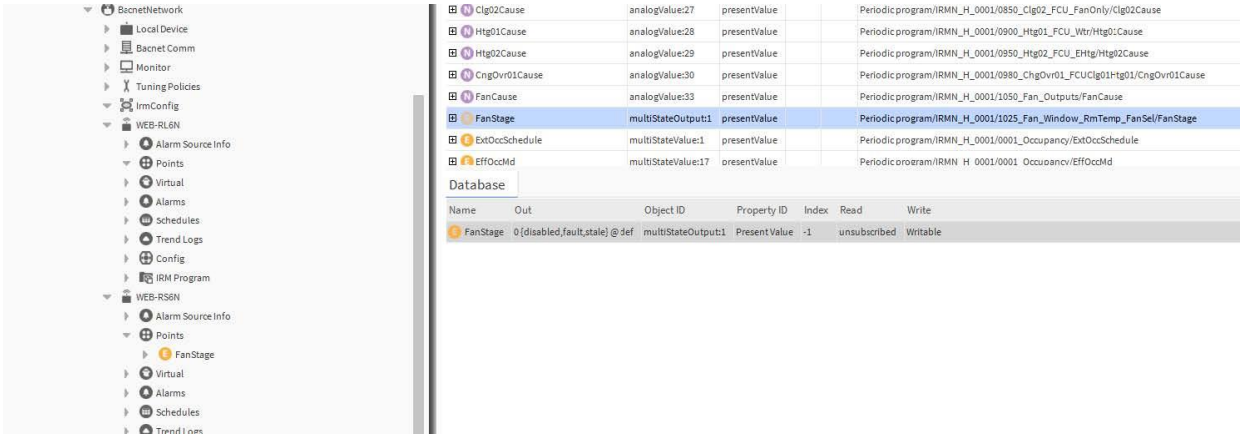
3. Sort the discovered points by clicking on the **Object Name** column.
4. In the point list, scroll to the ´FanStage` BACnet point.



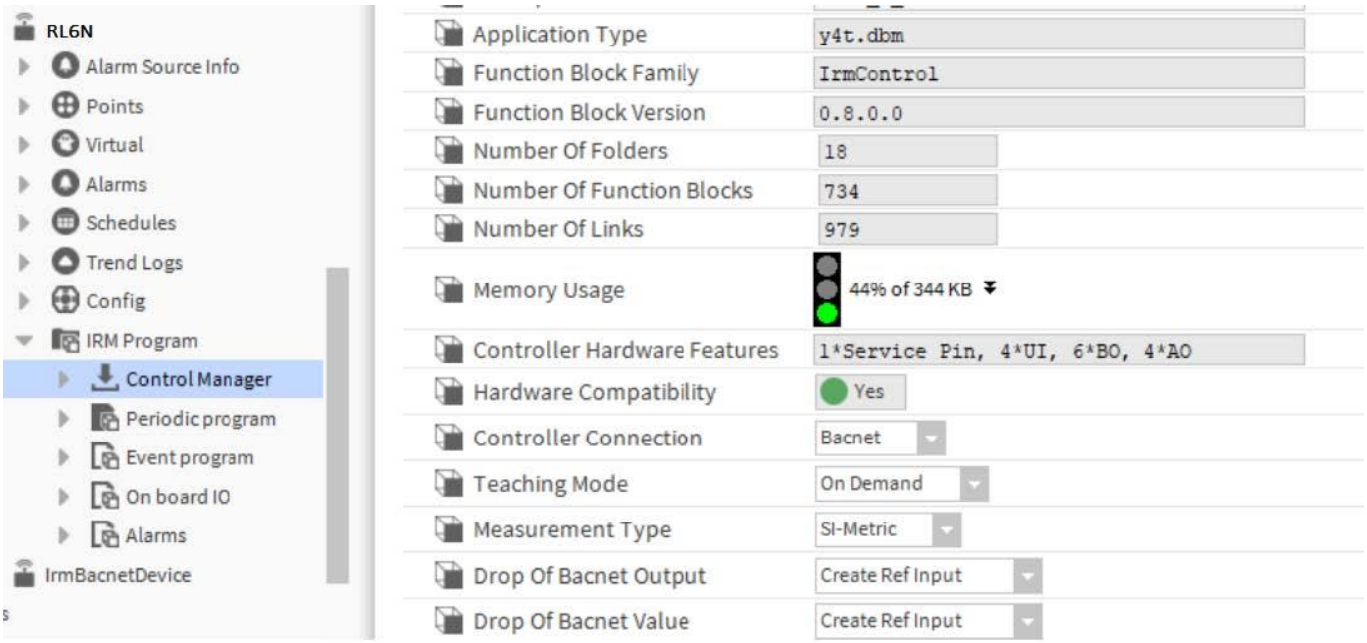
5. Add the ´FanStage` BACnet point to the database by clicking **Add** at the bottom.



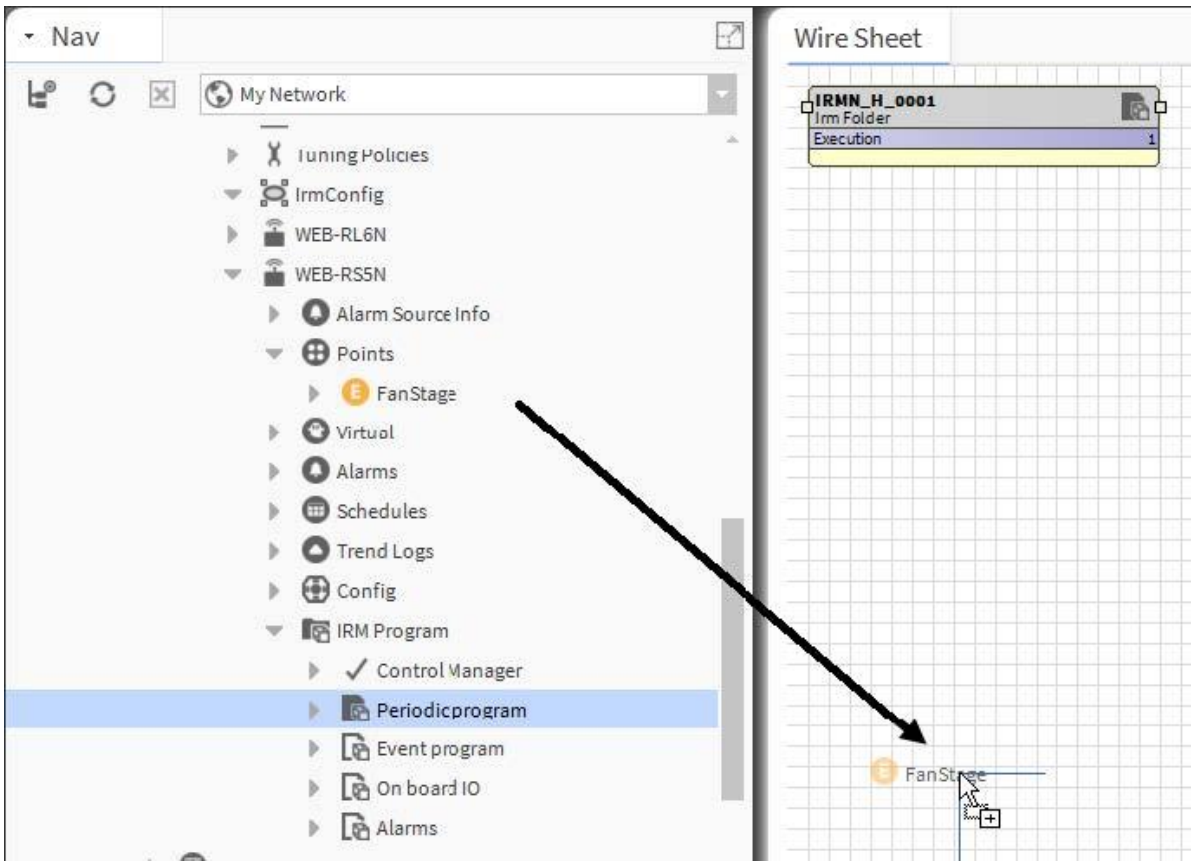
6. Click **OK** in the *Add* dialog box. The point will be added to the *Database* and the *Points* folder.



7. In the *Nav* tree, browse to the WEB-RS5N slave controller.
8. Expand the IRM Program folder.
9. In the control manager of the WEB-RS5N slave controller, set the drop option for BACnet output dropping (mandatory for outputs and value points, but not necessary for input points, see previous subsection). Based on the example, select ´Create RefInput' in **Drop of Bacnet Output.**

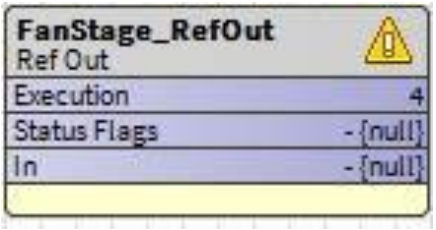


10. Double-click the control program folder, *Periodic Program* or *Event Program*, to which you want to add the ´FanStage` BACnet point.

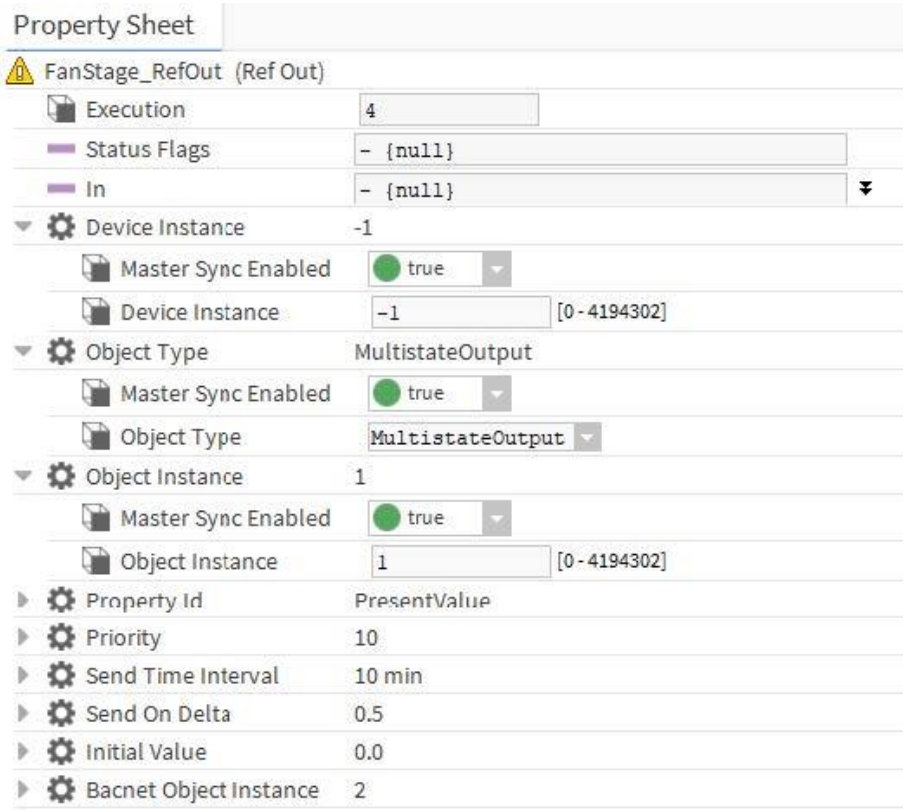


11. From the *Points* folder of the WEB-RL6N master controller, drag & drop the ´FanStage` BACnet point to the wiresheet of the WEB-RS5N slave controller.
12. In the *Name* dialog box, click **OK**.

A reference input point with the name of the dropped BACnet point, in this case ´FanStage`, is created.



13. Double-click the symbol of the reference input point to display the property sheet.



14. On the *Property Sheet*, you can see that the reference input is already mapped to the dropped BACnet multistate output point indicated by the values for device instance, object type, and object instance.
15. If desired, you can change the setting for master sync or other settings.

Master-Slave Synchronization

If Master Sync Enabled is true, this property will be synchronized between master and slaves in case a master sync operation is performed. The source device setting will be synced to a target device during master sync operation. If the source device has set the flag to false, then this parameter will not be synced to all target devices.

Application Management

Teaching and Learning

Synchronization can be performed in two ways:

- Teaching to Controller Downloads the changed application in the project to the controller
- Learning from Controller Uploads the changed application from the controller in the project

When in teaching mode, application changes of the project can be written to the controller in two ways:

- Immediate changes are written to the controller automatically and are effective immediately
- on demand changes are written to the controller manually and explicitly by the Teach to Controller action.

The explicit usage of the teaching and learning actions and the time when a particular action is used depends on the application status in the project and the connected controller, and the result you want to achieve.

Controllers can:

- be empty (factory delivery, cleared, no application)
- have a history (engineered with application)

Recommendations

Please be sure, which result you want to achieve and ask the following:

- Do you want to keep the application in the controller, or in the project?
- Do you want to change the application in the project and teach it to the controller?
- Do you want to learn an application from the controller to the project?

If you want to keep the application in the project, learning the application from the controller should be avoided for controllers with history. In this case, the controller should be cleared before, in order to avoid the destruction of the application in the project.

If you want to keep the application in the controller, teaching the application to the controller would not destroy the application. Only the changes are teached.

NOTE: If there are too many changes in the project or in the controller, learning from and teaching to the controller does not work.

NOTE: Software always give support by displaying appropriate messages which describe the current status/problem and how to proceed.

Teaching to Controller

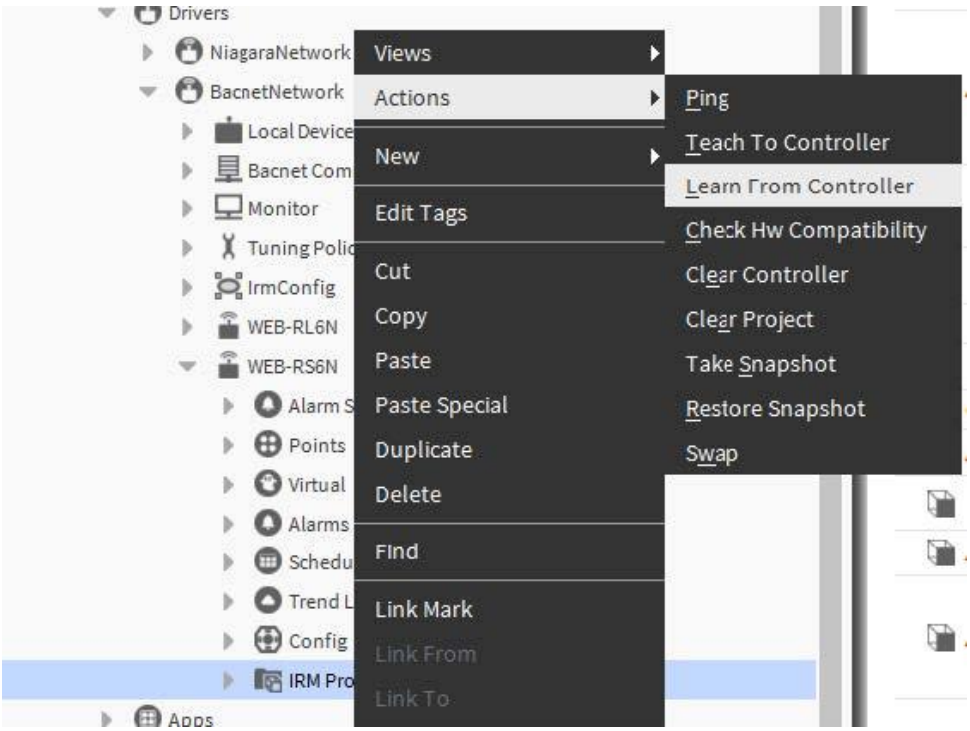
Teach to Controller" downloads the changed application of the project to the controller. This can happen in two ways:

- Immediate changes are written to the controller automatically and are effective immediately
- on demand changes are written to the controller manually and explicitly by the "Teach to Controller" action.

NOTE: When working in teaching mode ´immediate`, no messages display and no changes are indicated graphically when changing the application. The following procedure can be applied at any time when working in teaching mode ´on demand`.

Procedure

1. Right-click on IRM Program in the tree, then click **Actions**, and then click **Teach To Controller** in the context menu.



RESULT: The changes are written to the controller. The applications are synchronized. The successful action is displayed via IRM **Operations Monitor**.

Application Download Failure

Due to various reasons the download of an application can fail. This is shown via message in the **Jobs Sidebar**. In case of a failed application download, however parts of the application may have already been downloaded to the controller, and such application parts could start and operate the Spyder Model 5 controller. In this case, damage of the controlled equipment may occur.

IMPORTANT!

To avoid damaging the controlled equipment or environment due to partial application download failure, it is strongly recommended to clear the controller and perform the application download again.

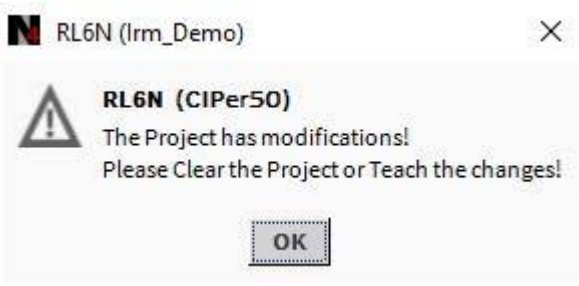
Learning from Controller

"Learn from Controller" uploads the current application from the controller in the project. The teaching mode does not care in this case.

WARNING!

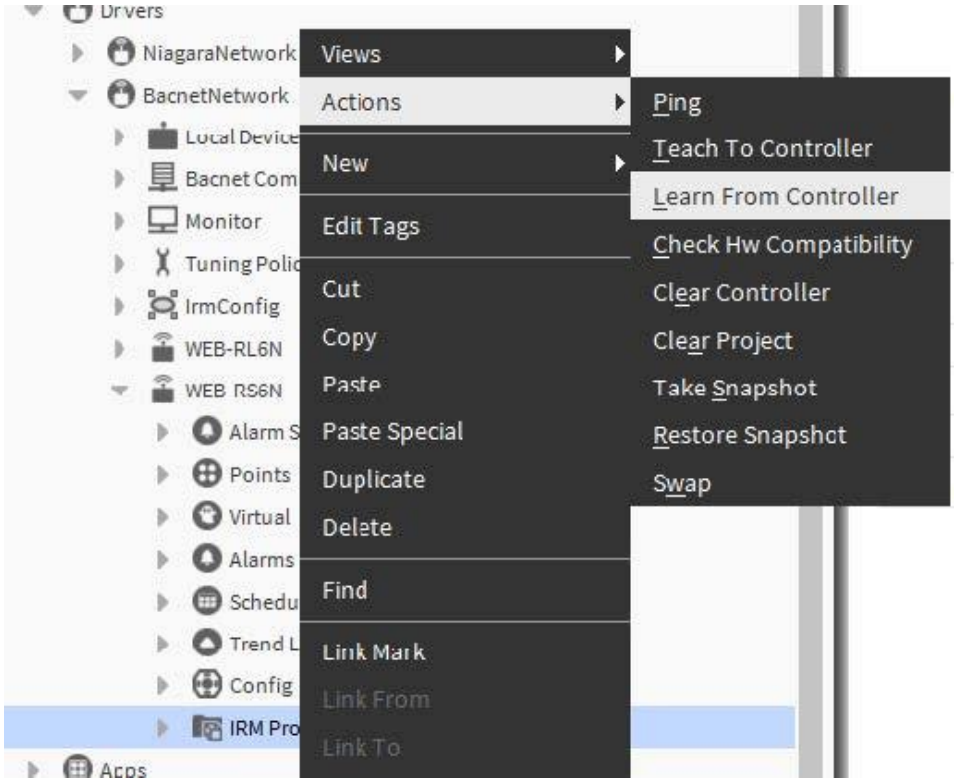
When learning from the controller, the application in the project will be deleted.

If the synchronization status is different, a message displays recommending either to clear the project (modifications are discarded) or to teach the controller (modifications in the controller are discarded).

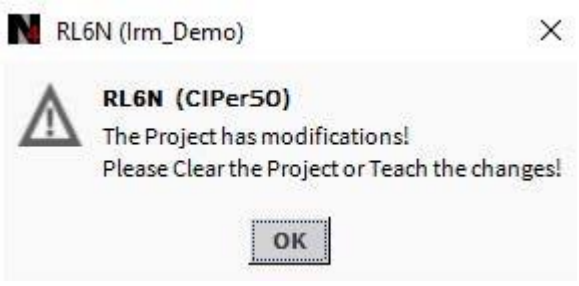


Procedure

1. Right-click on IRM Program in the tree, then click **Actions**, and then click **Learn From Controller** in the context menu.



RESULT: The application is to uploaded into the project. The applications are synchronized. The successful action is displayed via IRM **Operations Monitor**.

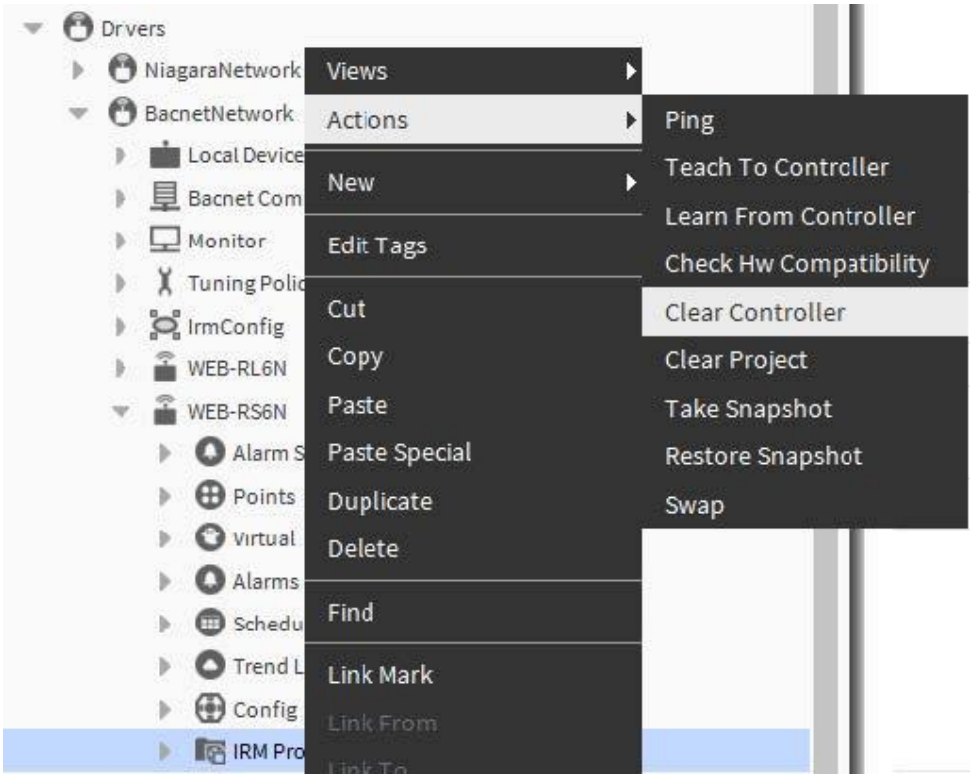


Clearing Controller

Clear Controller" deletes the application in the controller.

Procedure

1. Right-click on IRM Program in the tree, then click **Actions**, and then click **Clear Controller** in the context menu.



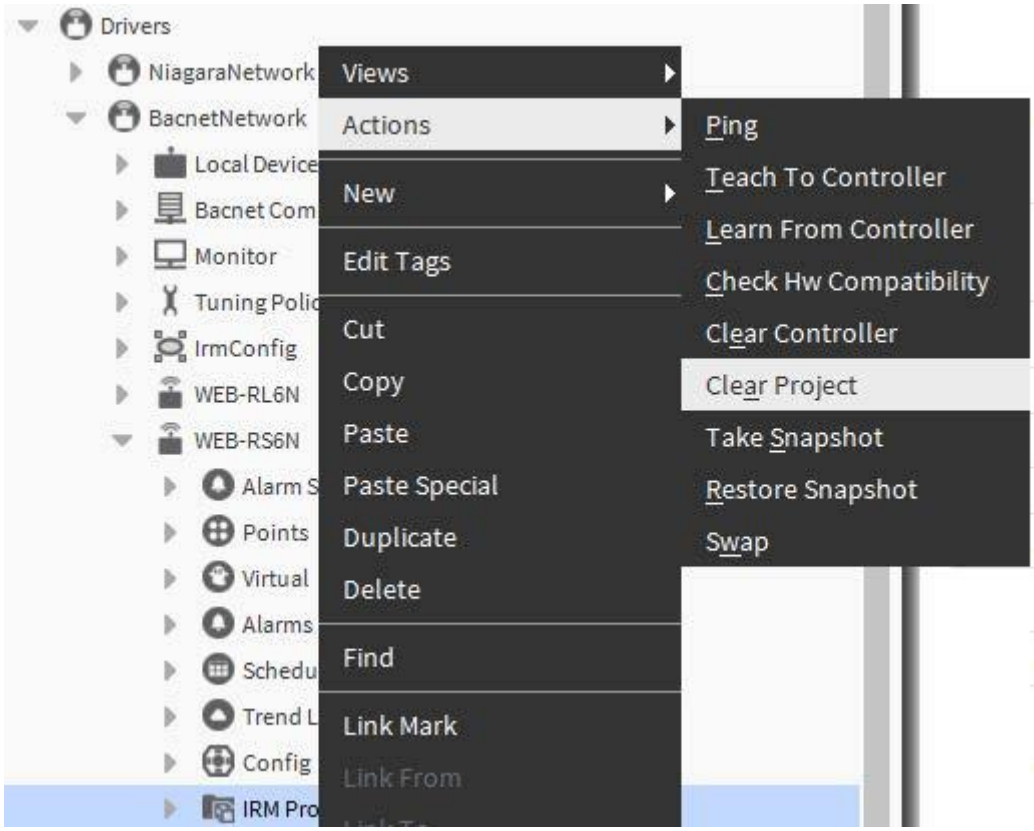
RESULT: The application is deleted in the controller. The successful action is displayed via IRM **Operations Monitor**.

Clearing Project

"Clear Project" deletes the application in the project.

Procedure

1. Right-click on IRM Program in the tree, then click **Actions**, and then click **Clear Project** in the context menu.



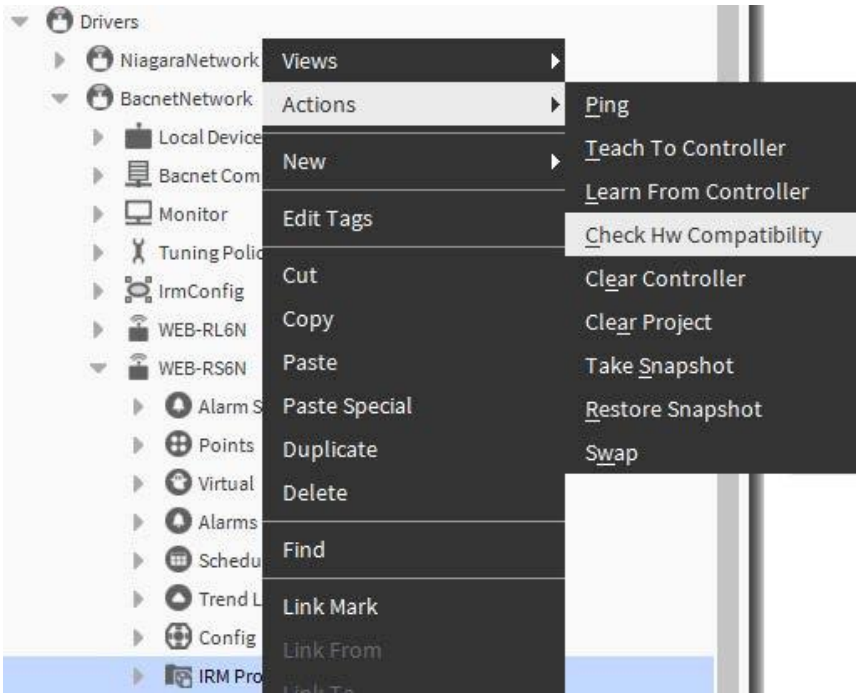
RESULT: The application is deleted in the project. The successful action is displayed via IRM **Operations Monitor**.

Checking Hardware Compatibility

The hardware configurations of the used physical controller and the hardware defined for the controller in the application can be checked. Any differences are indicated graphically via yellow warning symbol on the terminals in the wire sheet. In addition, a message displays that lists the affected terminals.

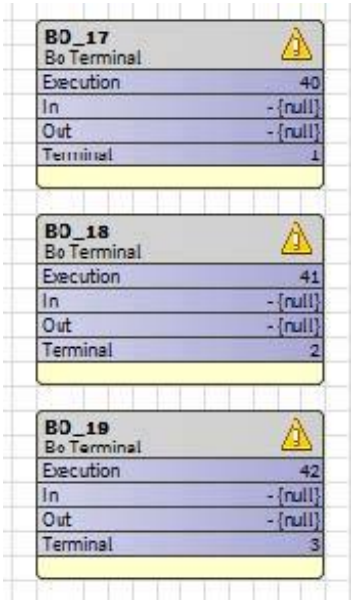
Procedure

1. Right-click on IRM Program in the tree, then click **Actions**, and then click **Check Hw Compatibility** in the context menu.



RESULT: In case of incompatibilities, as message box displays that lists the affected terminals:

On the wire sheet, the terminals are marked as “dirty” by a yellow warning symbol.



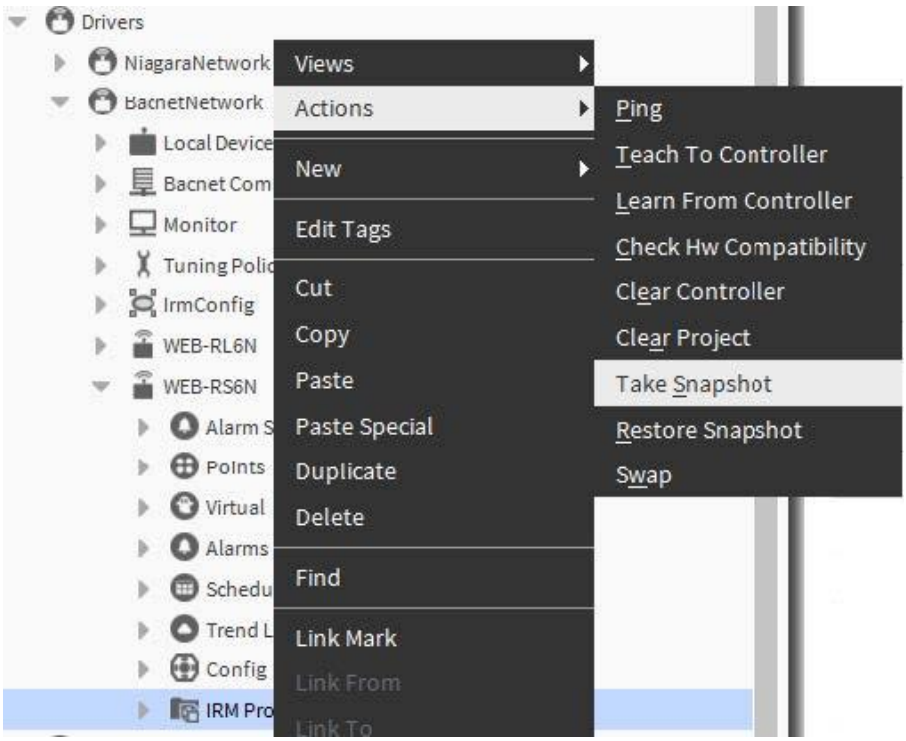
2. Confirm the message box by clicking **OK**, and then remove the dirty terminals.
3. Teach the controller.
   RESULT: The successful action is displayed via Irm **Operations Monitor**.

Taking Snapshot

The current status of an application can be backed up and restored later. This allows restoring a changed application if these changes should be discarded.

Procedure

1. Right-click on IRM Program in the tree, then click **Actions**, and then click **Take Snapshot** in the context menu.



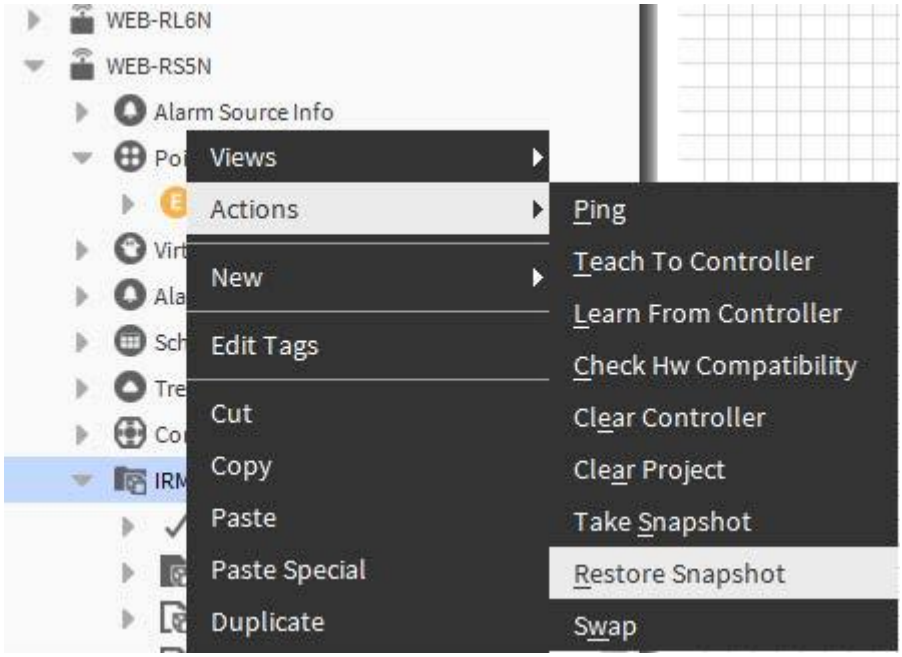
RESULT: The application is backed up and the successful action is displayed via IRM **Operations Monitor**.

Restoring Snapshot

The current status of an application can be backed up and restored later. This allows restoring a changed application if these changes should be discarded.

Procedure

1. Right-click on IRM Program in the tree, then click **Actions**, and then click **Restore Snapshot** in the context menu.



RESULT: The application is restored in the project and the successful restore is displayed via IRM **Operations Monitor**.

Swapping IRM Program

Purpose

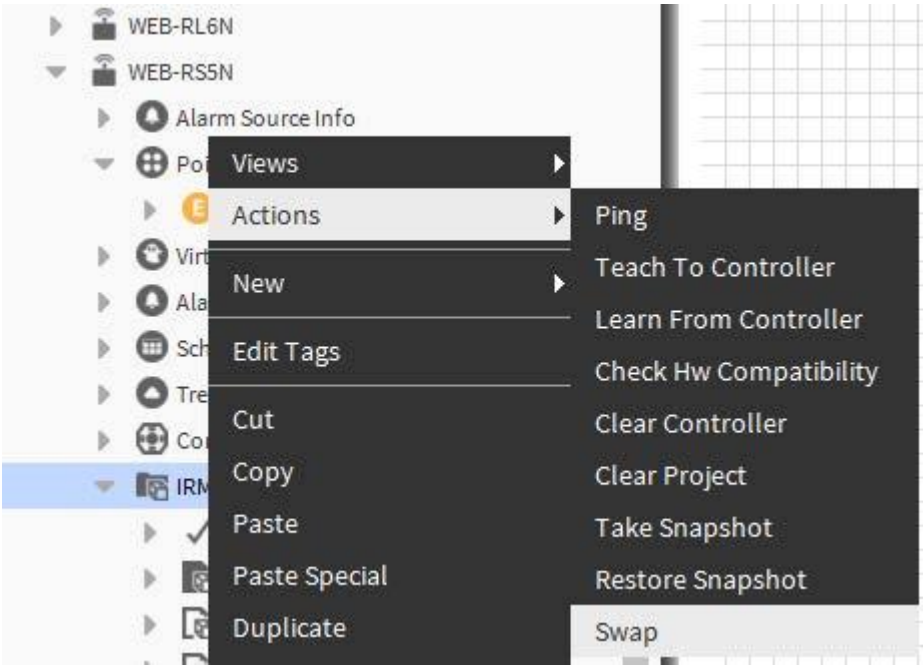
Swapping saves RAM space and reduces processor load and bus traffic. The current state of a swapped device is frozen and saved to an IRM repository on the disk. The swapped device is indicated by a proxy which inherits and shows the minimum information of the device necessary for swapping-in the device. For swapped-out devices, synchronization is no more possible. In order to synchronize swapped-out devices, the devices must be swapped in again. You can swap single devices one after another or multiple devices in one step.

IMPORTANT!

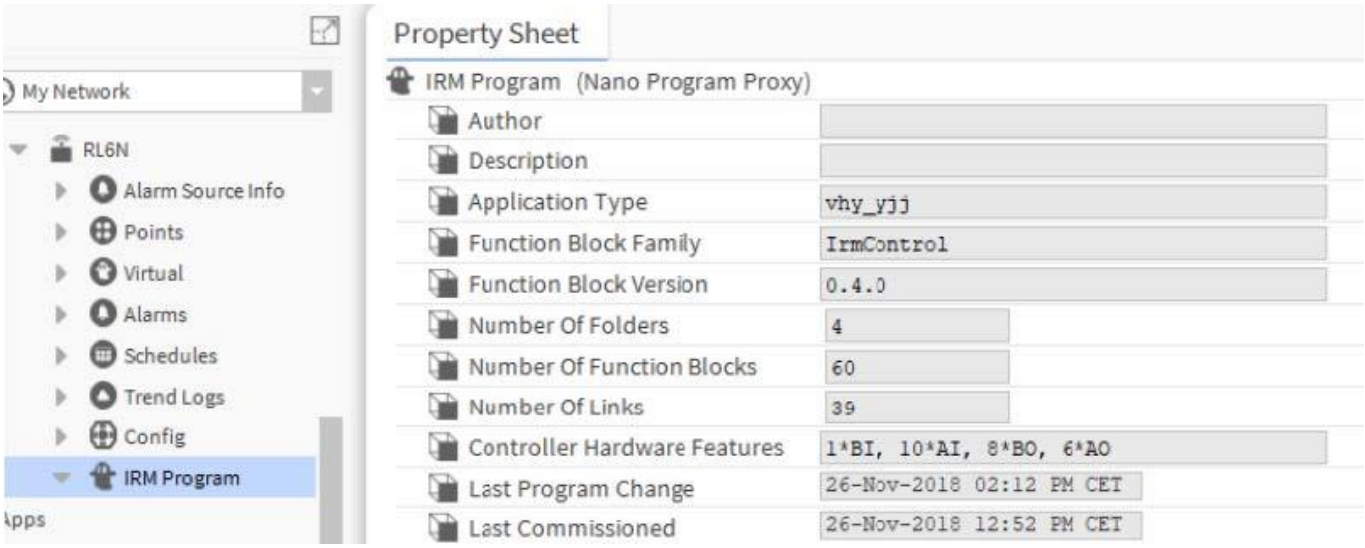
It is strongly recommended to swap out the application in any case after finishing the engineering in online or offline mode.

Procedure (Single Device)

1. To swap-out the device, expand the device.
2. Right-click on the IRM Program folder in the tree, then click **Actions**, and then click **Swap** in the context menu.



RESULT: The device is swapped out. On the *Property Sheet* the proxy is displayed. In the tree, the IRM program is shown without any subfolders.



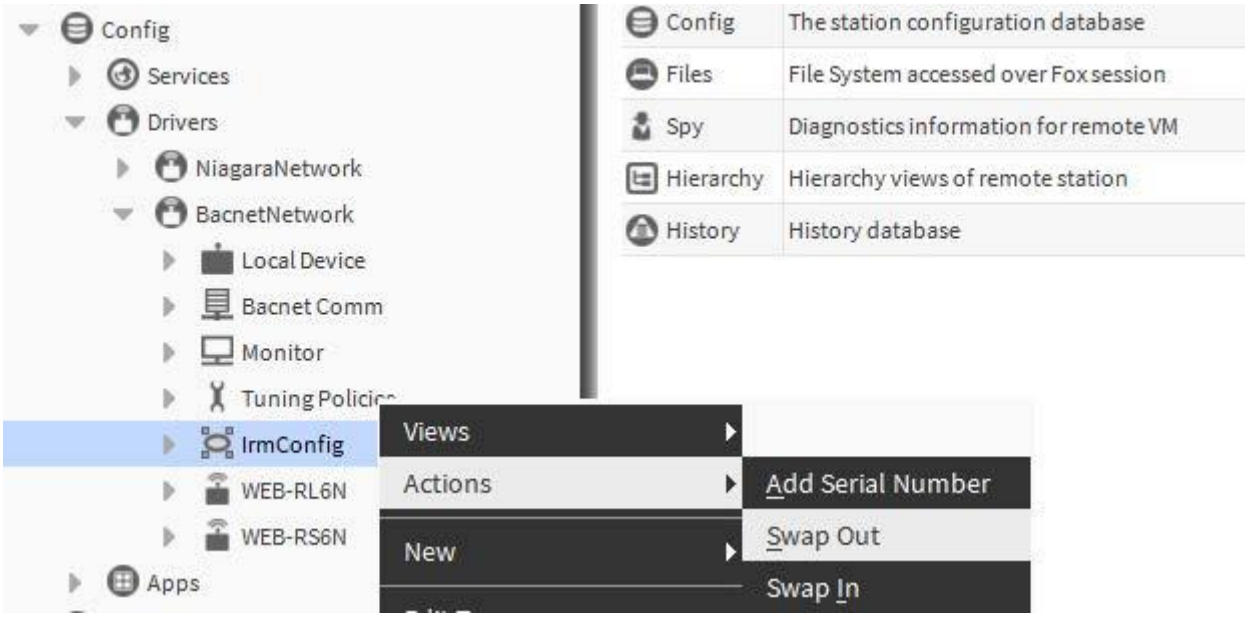
The successful swap-out is displayed in the **Irm Operations Monitor**.

3. To swap-in the device, right-click on the IRM Program folder in the tree, then click **Actions**, and then click **Swap** in the context menu.

RESULT: The device is swapped in. In the tree, the IRM program is restored showing all subfolders. The successful swap-in is displayed in **Irm Operations Monitor**.

Procedure (Multiple Devices)

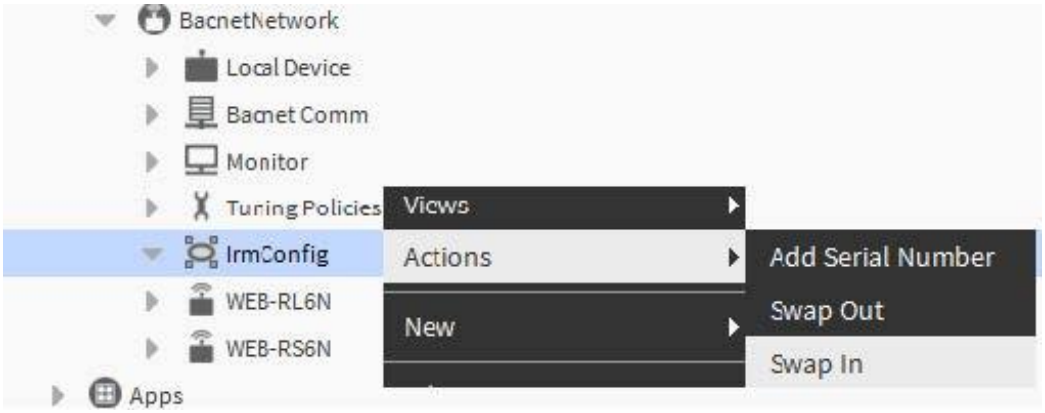
1. To swap-out the devices, expand the BACnet network.
2. Right-click on the IRM *Config* folder in the tree, then click **Actions**, and then click **Swap Out** in the context menu.



RESULT: The devices are swapped out.

The successful swap-out is displayed in the **Irm Operations Monitor**.

3. To swap-in the devices, right-click on the IRM *Config* folder in the tree, then click **Actions**, and then click **Swap In** in the context menu



RESULT: The devices are swapped in. The successful swap-in is displayed in **Irm Operations Monitor.**

Master Sync

The “Master Sync” function is based on a group of multiple devices that must have the same application type. This is called a master sync group. The master sync group is established via the “Clone Application” function which clones the application of one selected device (template) to multiple devices.

Among all devices, one device is defined as the source and the other devices are defined as targets. Then the master sync command allows rolling out the current application (changes) of the source device to all target devices per one step.

The master sync function works in online and offline mode on project level, but not in the devices. Hence, the applications in the project do not have to be synchronous with the applications in the devices when working in teaching mode ´on demand`.

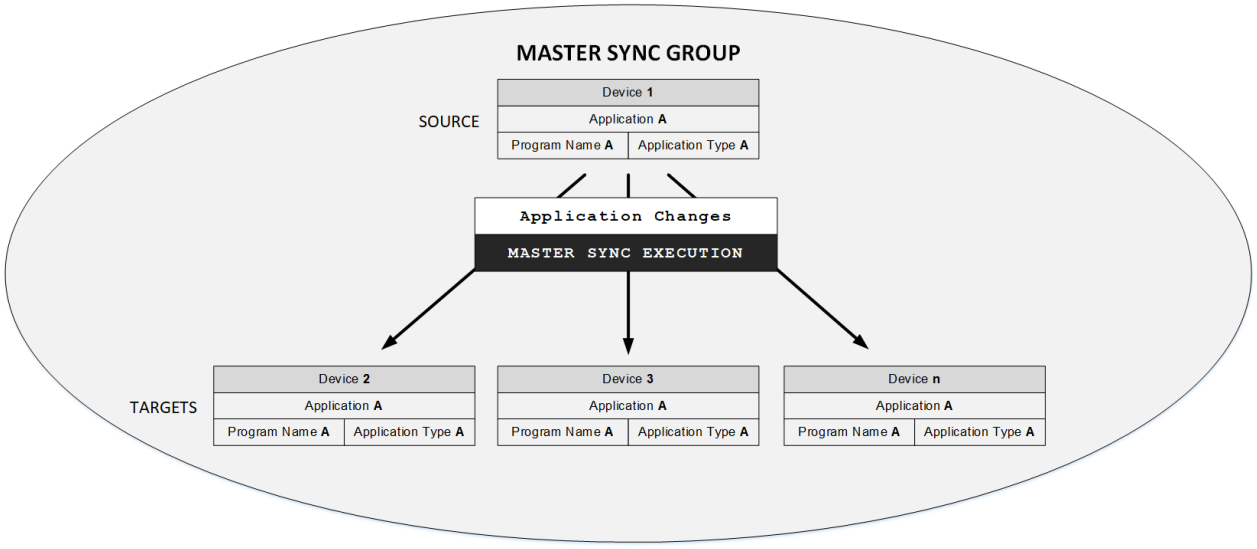


Fig. 12. Master Sync Group including Source and Target Devices

NOTE: Cloning of the application and the source-target definition for the devices can be done independently. That is, the source-target definition of the devices can be done before or after cloning of the device application and vice versa.

To start with cloning the application, see section “Cloning Application“, p. 94.

Cloning Application

Clones the device application based on a selectable template (device) to selectable devices. As result, all devices will receive the same application type which is the basis of a master sync group.

NOTE: Even if all devices will have the same application type after cloning, the master sync group is not finally established. This is done before or after the cloning via the **>>Master Sync** function (see “Applying Master Sync” section, p. 95).

Example:

The following schematic shows a master sync group consisting of 4 devices with the same application A which has the program name A and application type A. The application is cloned using device 1 as template and the devices 2 through n are selected for receiving this application. They are forming a master group defined by application Type A.

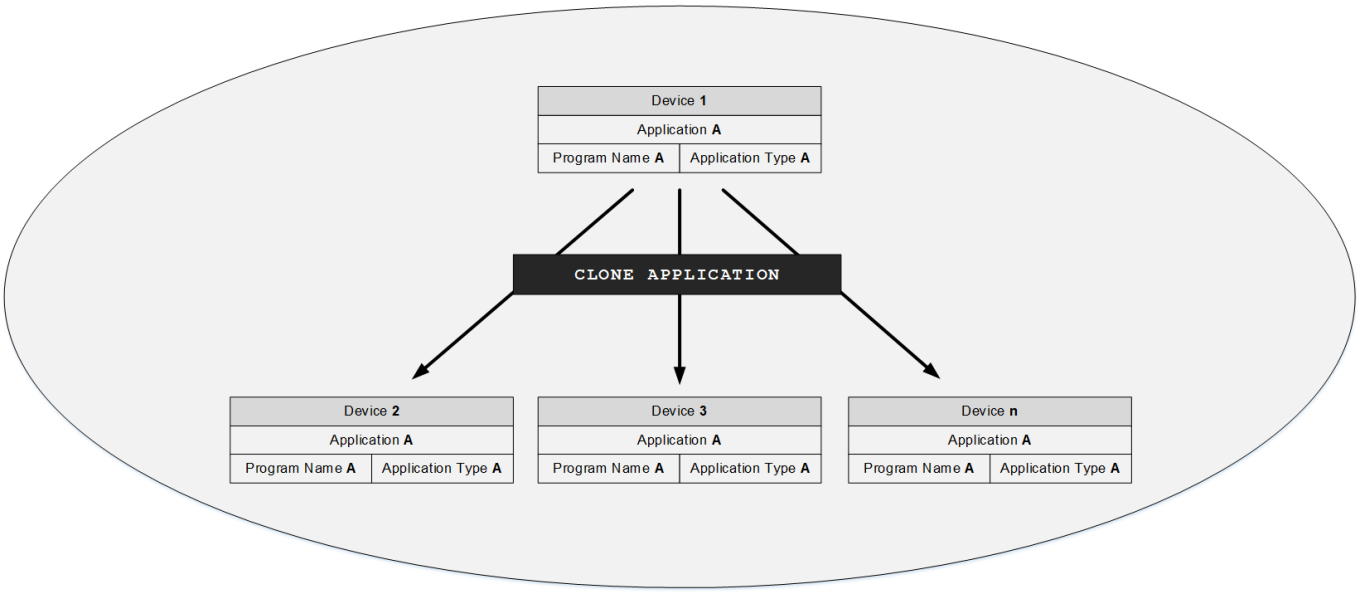
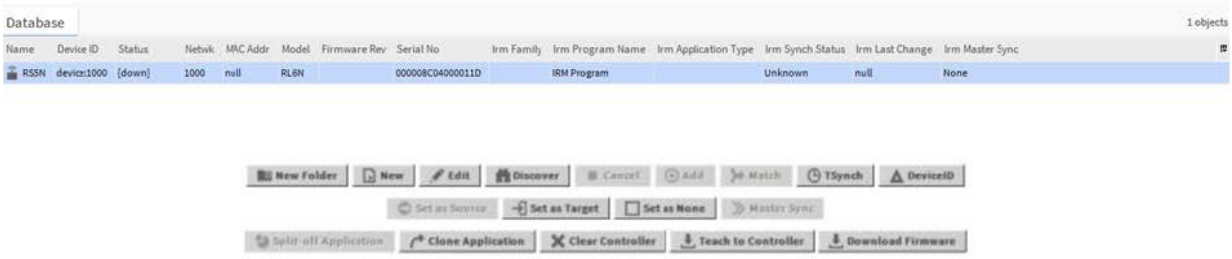


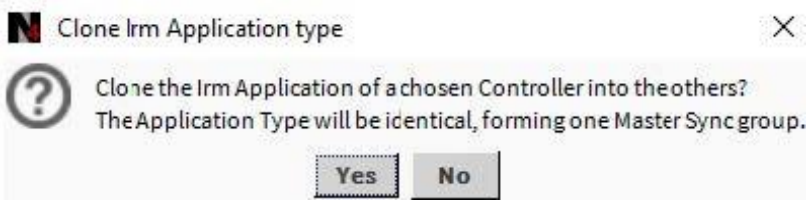
Fig. 13. Cloning Application

Procedure

1. In the *Database* pane, select all devices which should be included in the clone. In the next step, the application of one selected device will be used as template. All others will receive the application type of the selected template device.



RESULT: The following message box displays.



2. Confirm the message by clicking **OK.**

RESULT: The Select Template dialog box displays.

3. Select the template (device) from the drop-down listbox.
4. Confirm the message by clicking **OK.**

   RESULT: The selected template will be used for all devices selected in the *Database* pane. The **Irm Application Type** column in the *Database* pane indicates that all devices have now the same application type. The master sync group is formed.

NOTE: Cloning of the device application can be done before or after the source-target definition of the devices and vice versa.

Applying Master Sync

Allows rolling out the current application (changes) of the source device to all target devices per one step. The differences in all target device applications are synchronized with the application of the source device.

NOTES: The source-target definition of the devices can be done before or after cloning of the device application and vice versa.

At least one device must include an application in order establish a master sync group via cloning. If not already done, clone the application as described in the section “Cloning Application“, p. 94.

The master sync function can be applied in online and offline mode.

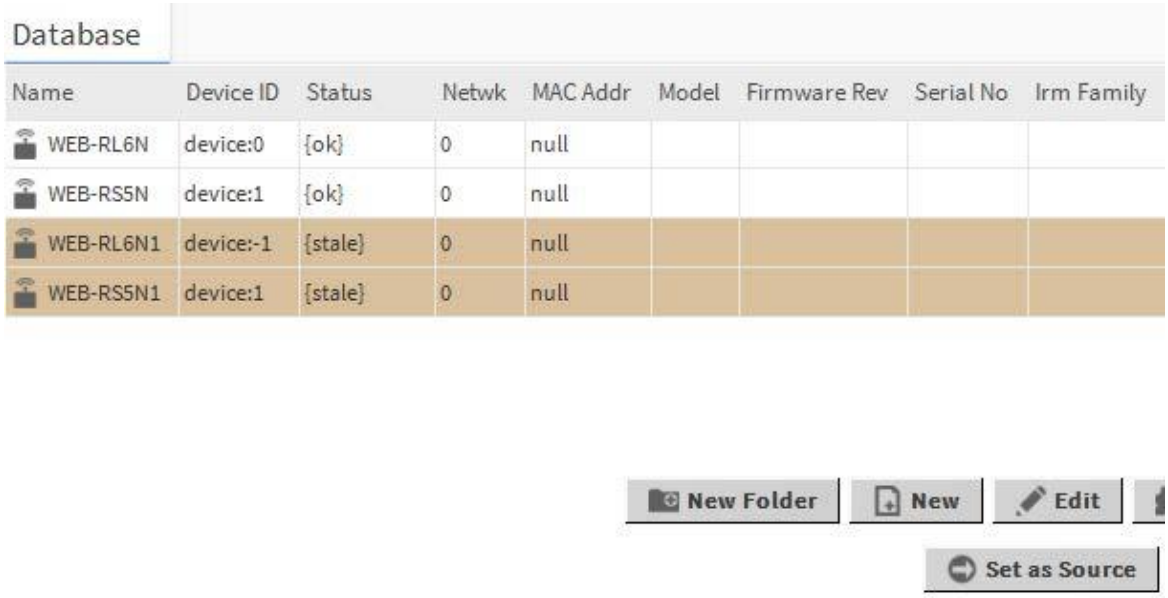
Example:

The following procedure shows an offline engineering example with 6 IRM devices created manually using the standard Niagara **New** command. The application has been cloned.

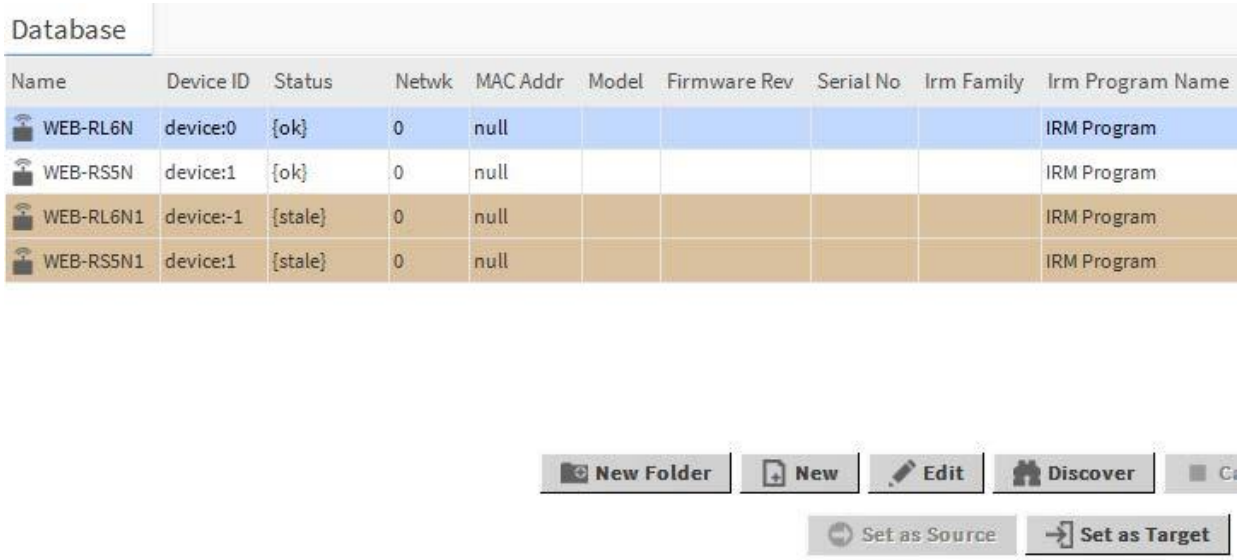
Procedure

1. Invoke the *Database* pane to display the offline devices. In the IRM **Application Type** column, the same application type is shown for all devices forming the master sync group.

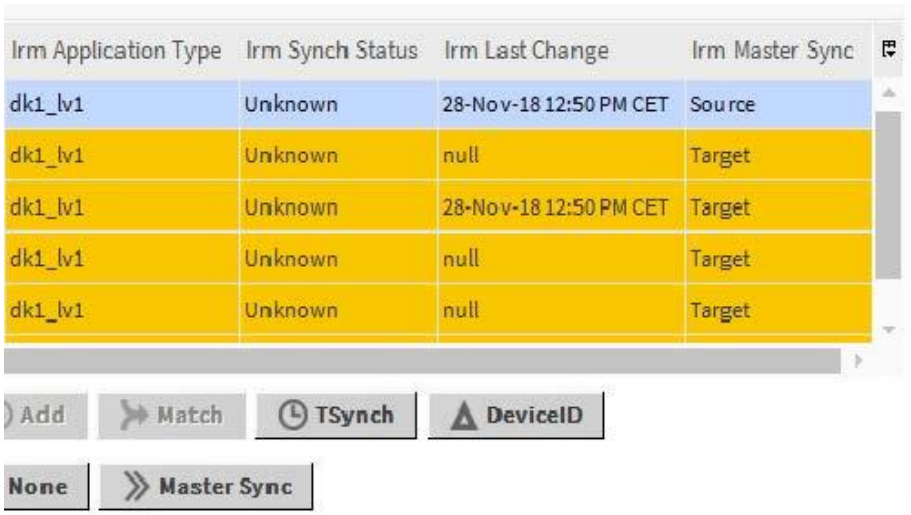
2. Select the device you want to define as the source, and then click **Set as Source**.



3. Select the devices you want to define as the targets, and then click **Set as Target**.



4. Select the source device, and then click **>>Master Sync**.



RESULT: The Synchronize *differences* dialog box displays.

5. Confirm the message by clicking **OK.**

RESULT: The differences in all target device applications are synchronized with the application of the source device.

Excluding Function Block Items from Master Sync

As desired, particular values of function blocks can be excluded from the Master Sync function. Then, when performing the master sync command, the excluded values of the function block will be kept and not affected by the updated application. This applies to the periodic and event programs.

Excluding function block values from master sync updates can be applied in the master device and in the target device(s).

When excluding a function block value in the master device, this particular value will be kept in all target devices.

When excluding a function block value in a target device, only this particular target device will be excluded from the value update.

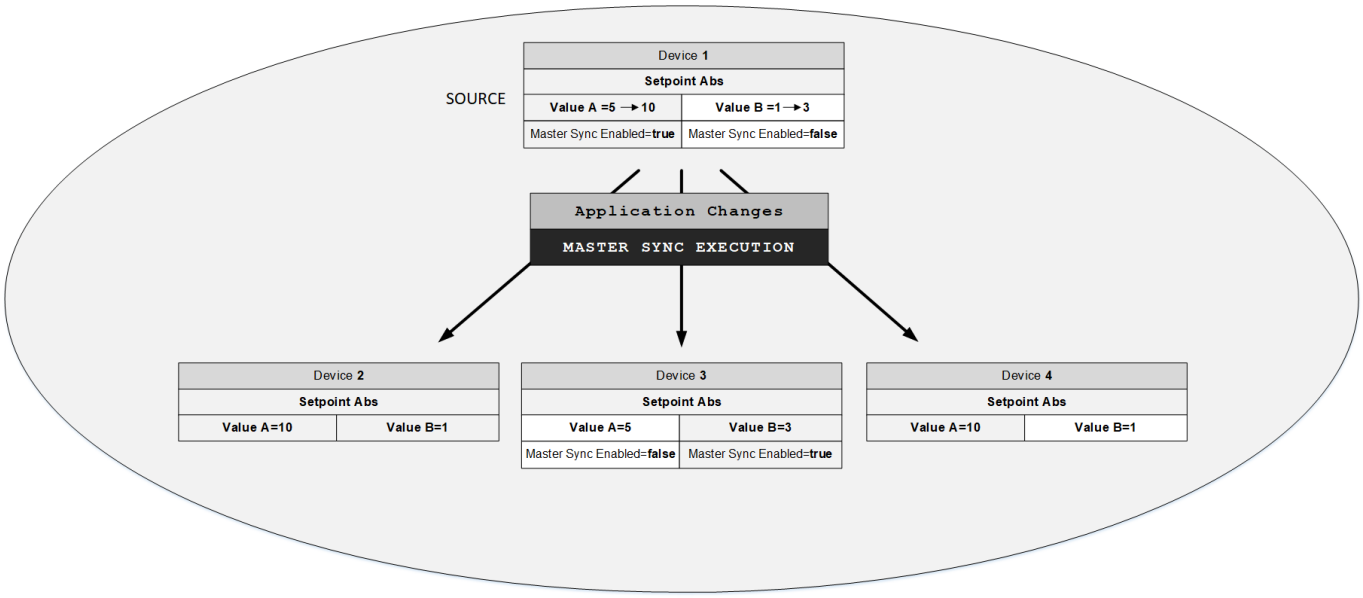
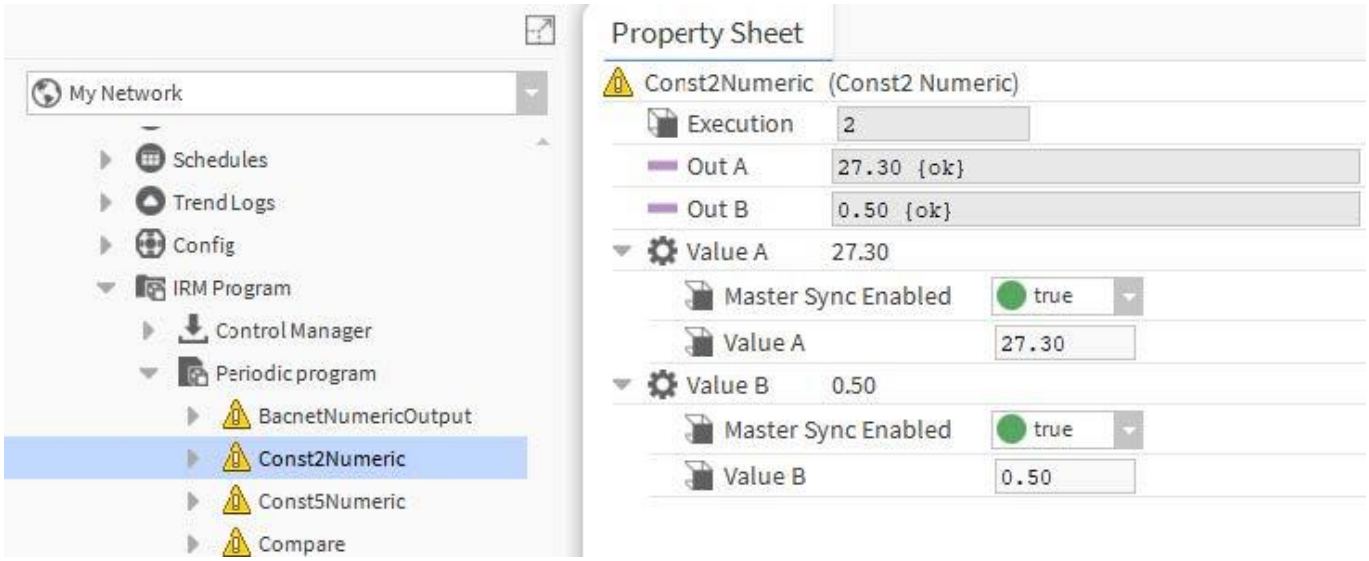


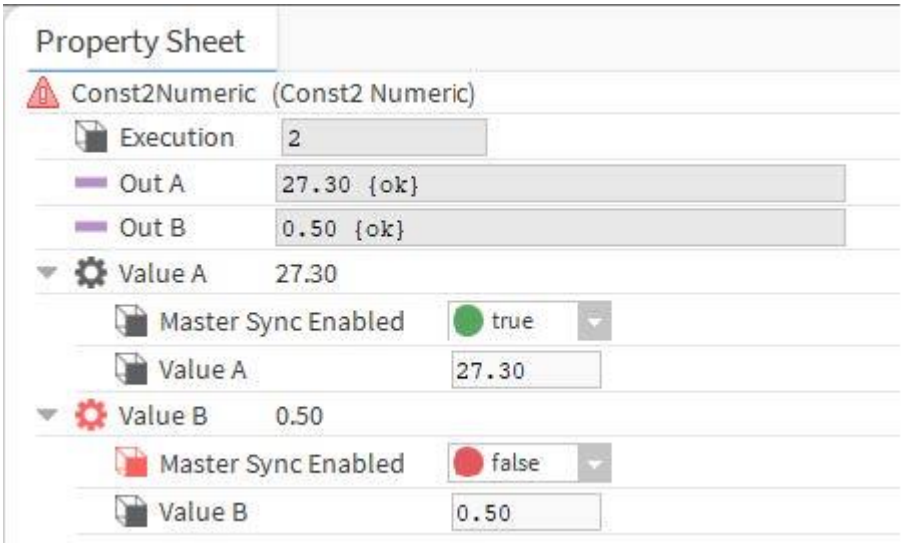
Fig. 14. "Master Sync Enabled" Configuration and Function Block Value Updates

Procedure

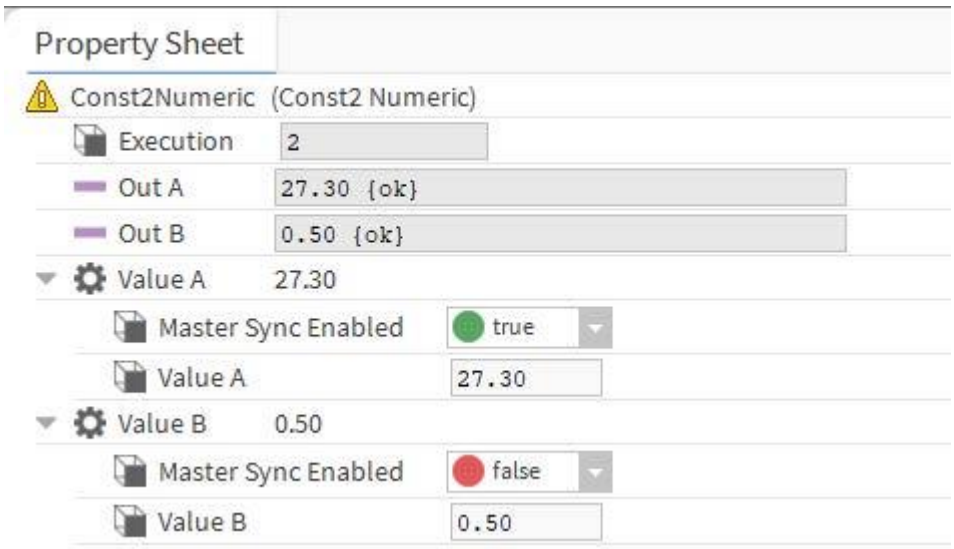
1. Double-click the item in the *Periodic program* or *Event program* to display the Property Sheet.



2. Expand the values area by clicking ⚙. The **Master Sync Enabled** option is set to ´true` by default.
3. To exclude a function block value from master sync updates, set the option to ´false'.



4. Click **Save**.



Splitting-Off Application

Splits the unique application of the master sync group into a new application and keeps the existing application. For the new application, you can enter a different IRM program name. A new IRM application type is issued automatically by the software. The new application can then be cloned to form the new master sync group.

This function can be applied in order to extend an existing application with new features for the usage in a similar environment, e.g. the application of a small-sized office will be used as basis for creating an application for a mid-sized office.

(see figure next page).

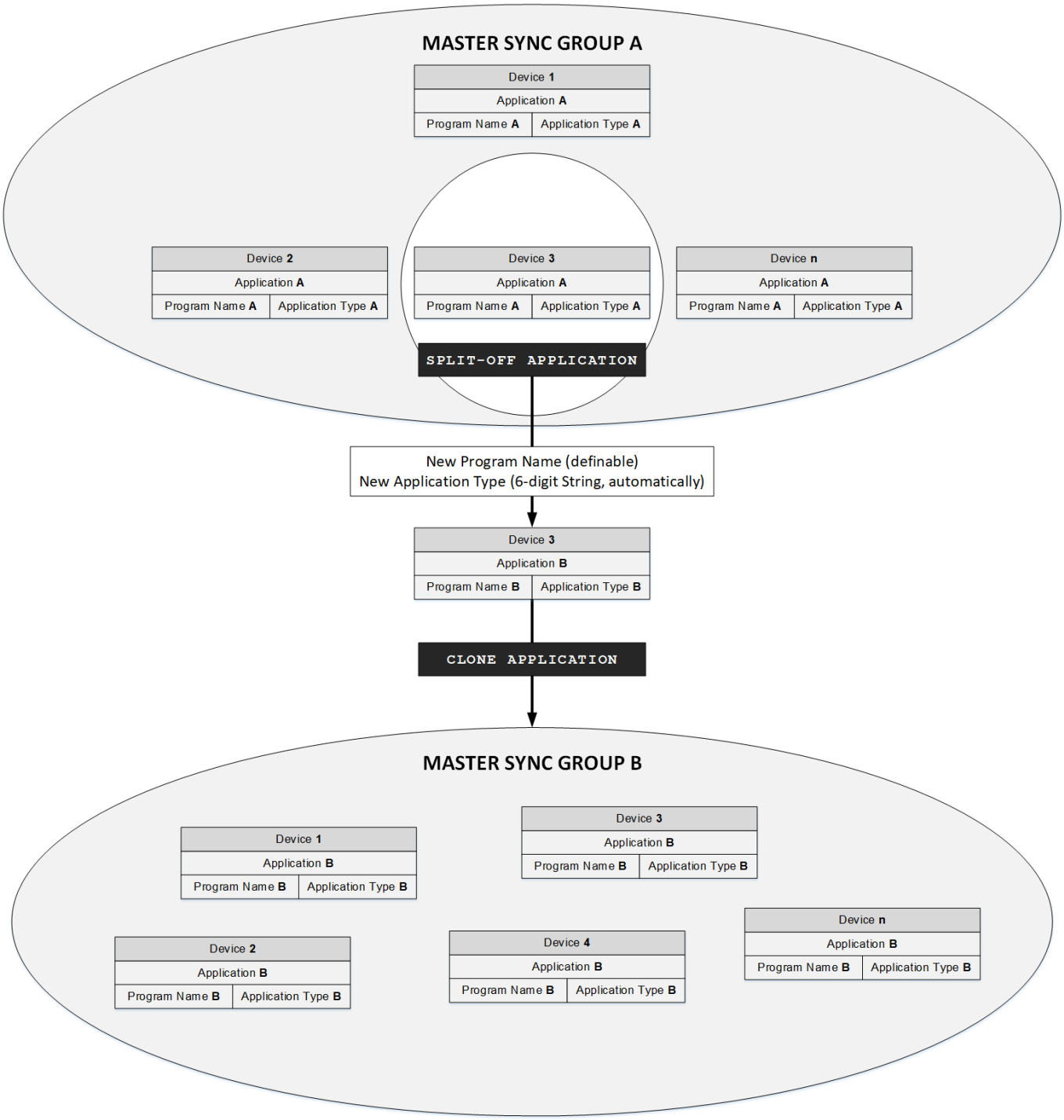
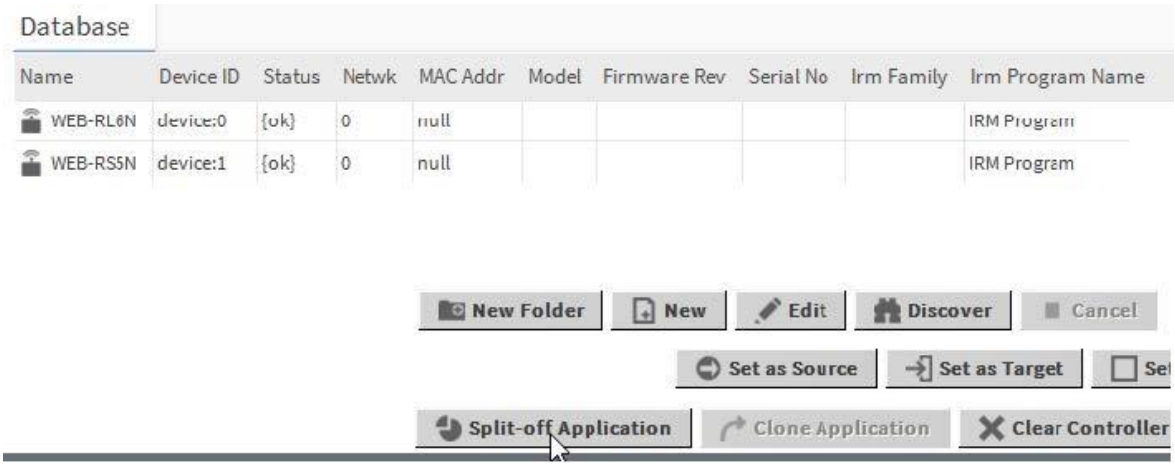


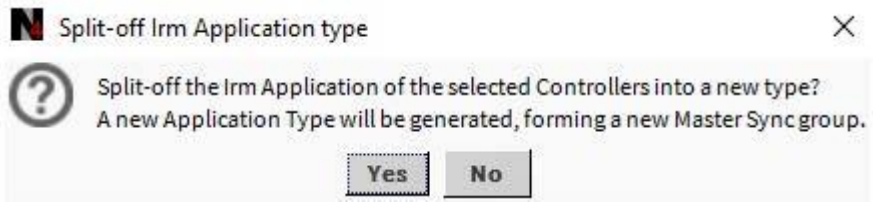
Fig. 15. Splitting-Off and Cloning Application

Procedure

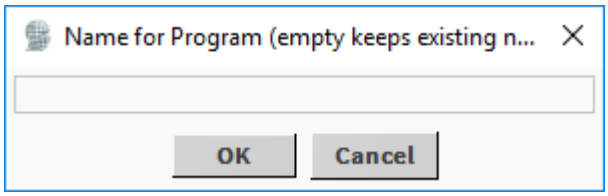
1. In the *Database* pane, select the device you want to split-off, and then click **Split-Off Application**.



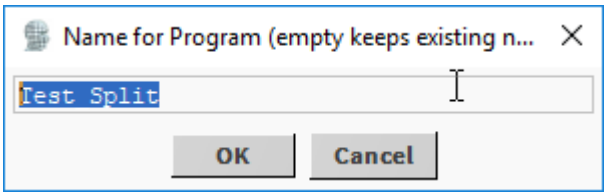
RESULT: The following message box displays.



2. Confirm the message by clicking **OK.**
   RESULT: The following dialog box displays.



3. Enter a new name for the IRM program.



4. Confirm by clicking **OK.**
   RESULT: In the *Database* pane, in the IRM **Program Name** column the defined program name is displayed and in the IRM **Application Type** column the new application type is displayed.

With the device based on this split-off application type, you can now form a new master sync group by cloning the application and defining the source and targets.

Irm Program Name	Irm Application Type
Test_Split	psi_o1q
IRM Program	ni2+goe

NOTE: Cloning of the device application can be done before or after the source-target definition of the devices and vice versa.

Controller MAC Address Assignment

Automatic MAC Address Assignment

The factory setting of the MAC address of an IRM controller is 0xFF by default. On first power up, a controller will automatically assign itself a unique MAC address within the range of assignable MAC addresses. The range of assignable MAC addresses is defined by the Min MAC and Max MAC settings within the controller. For new factory controllers, the default value for Min MAC is 1 and for Max MAC it is 32 (= Max Master setting in Niagara).

NOTE: The Max Master setting can be extended in Niagara to 127 at maximum and saved into the controller.

Once the controller has found a valid MAC address, it is saved in the controller permanently. From now on and on every power up or system reset, this MAC address is used for MSTP communication.

The permanent MAC address changes automatically if any of the following conditions happen:

- the Auto MAC process is re-triggered by Niagara via “You are” command using the MAC address 255 (0xFF)
- there is a conflict caused by devices in the network with the same MAC address.
- reset to factory defaults by pressing the service pin during power on

Manually Changing MAC Addresses

In some scenarios, you might intend to change the MAC addresses:

Example:

If a small number of controllers are connected to a network, the Auto MAC function executed in the controller may result in huge gaps of the assigned MAC addresses, e.g. 2, 7, 16, 23 and 31. Such a MSTP network is not optimized in terms of bandwith usage.

To optimize the network, you can manually change the MAC addresses of the controllers in Niagara (see also section “

Setting MAC Address of Controller“, p. 104). Make sure that unique MAC addresses are assigned.

IMPORTANT!

Once you have changed all desired MAC addresses for the controllers, it is recommended to change the Max Master setting of the controllers to the highest MAC address (=Max MAC) among all controllers. This results in an optimum bus performance.

NOTE: If the Max Master setting is not set to the Max MAC value, it will not impact the MSTP functionality, but only the MSTP performance.

Swapping MAC Addresses

If you want to assign to a controller (A) a MAC address which is already assigned to another controller (B), the following steps must be applied:

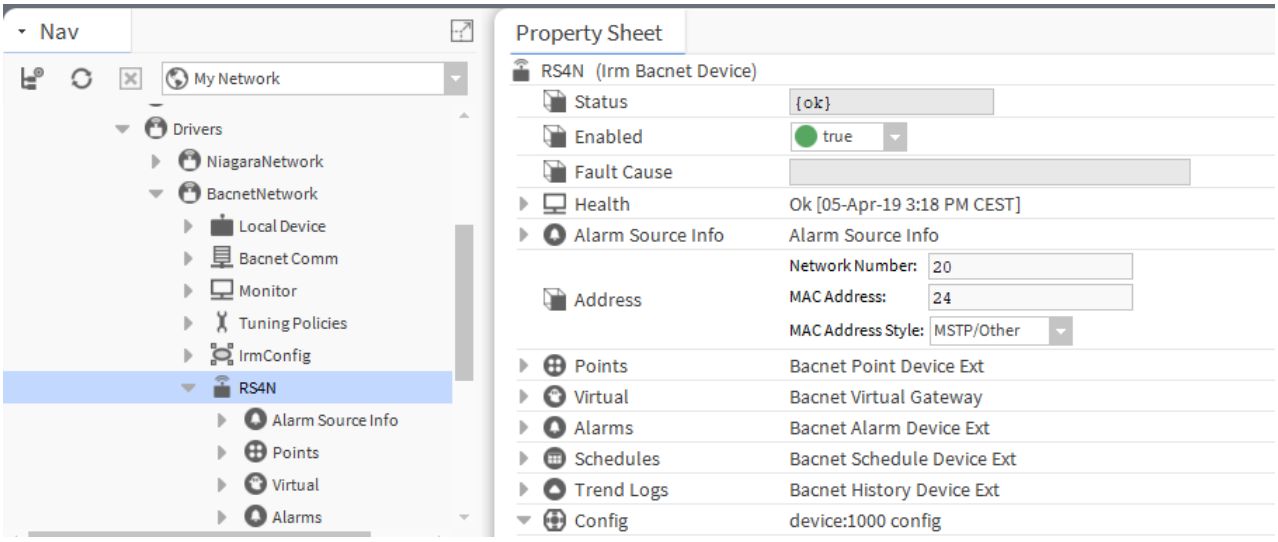
1. Re-assign a free MAC address in the range of 1-32 to the controller B which blocks the needed MAC address.
2. Now, assign the freed MAC address of controller B (step 1) to the controller A.
3. Assign the next needed MAC address to the controller B (optional, e.g. in case of particular MAC address requirements).
4. Above steps can be continuously applied for assigning desired MAC addresses to further controllers.
5. Change the Max Master setting to the highest MAC address (=Max MAC) among all controllers (see Important Note above).

Setting MAC Address of Controller

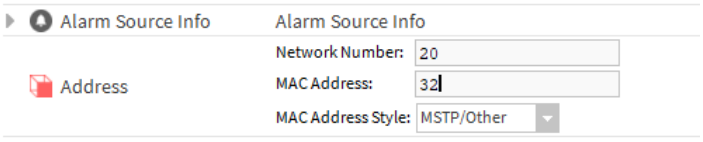
At any point, the automatically or manually assigned MAC address of an IRM controller and the Max Master setting can be changed in Niagara.

Procedure

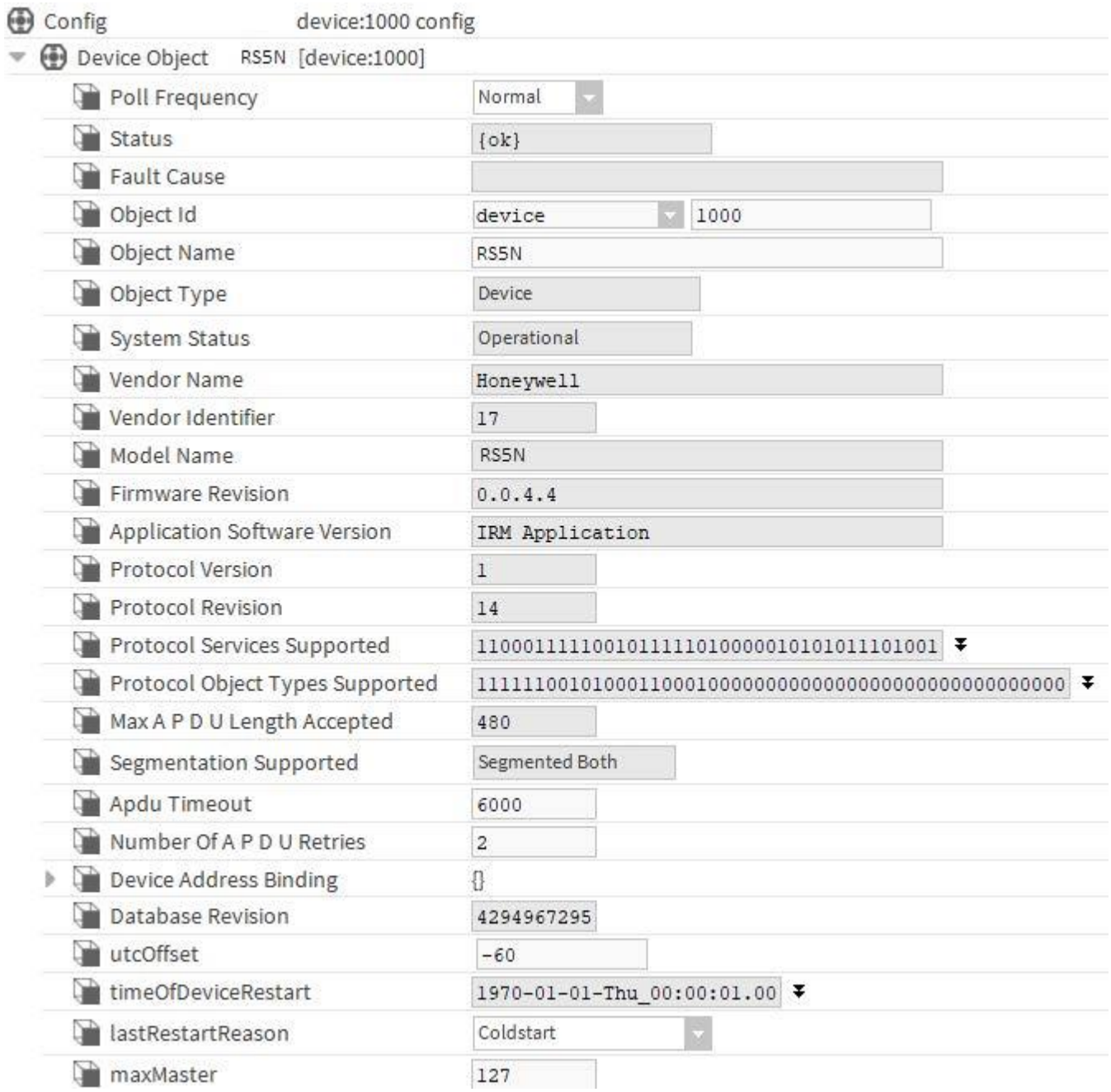
1. Open the *Property Sheet* of the IRM controller.



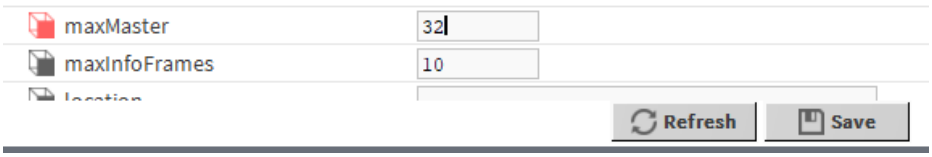
2. In **Address**, change the MAC Address as desired.



3. Confirm by clicking **Save**.
4. If the changed address is the highest address on the MSTP network, change the Max Master setting as follows.
5. Expand the **Config / Device Object** area.



6. In **maxMaster**, enter the highest MAC address of all controllers of the network.



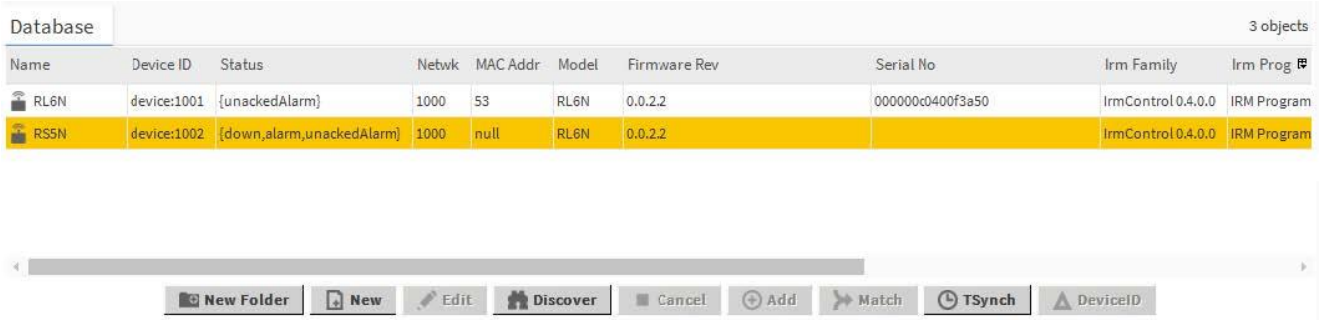
7. Confirm by clicking **Save**.
8. Enter the same number of ´maxMaster` in all controllers of the network.

Synchronizing Device Time

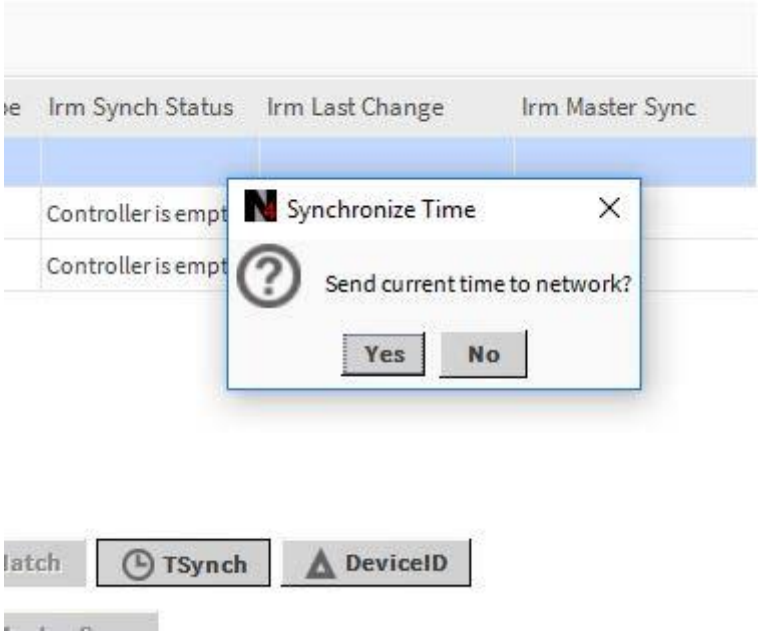
Synchronizes the time of the devices with the time either from the work bench PC that is used for device engineering or from the supervisor. This depends from where the function is performed.

Procedure

1. In the *Database* pane, select **TSync**.



   RESULT: The *Synchronize Time* dialog box displays.



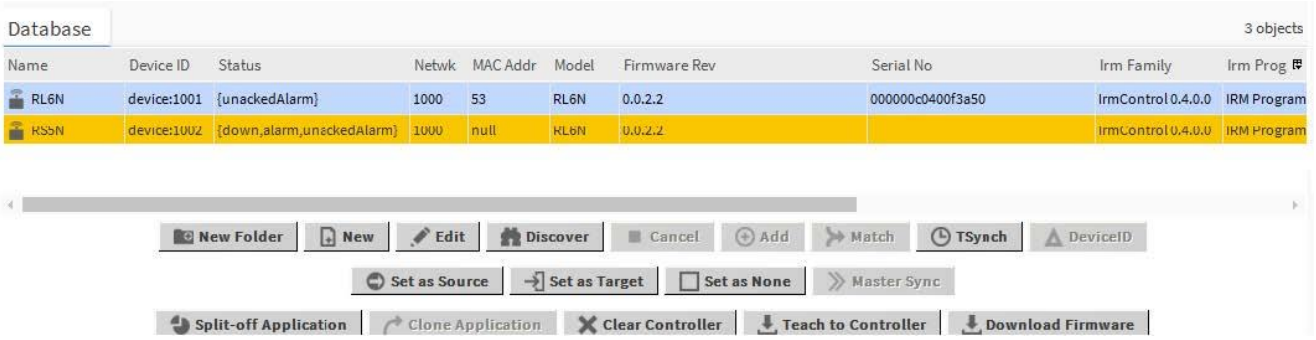
2. Confirm the message by clicking **OK.**
   RESULT: The time of the devices are set to time of the supervisor.

FIRMWARE DOWLOAD

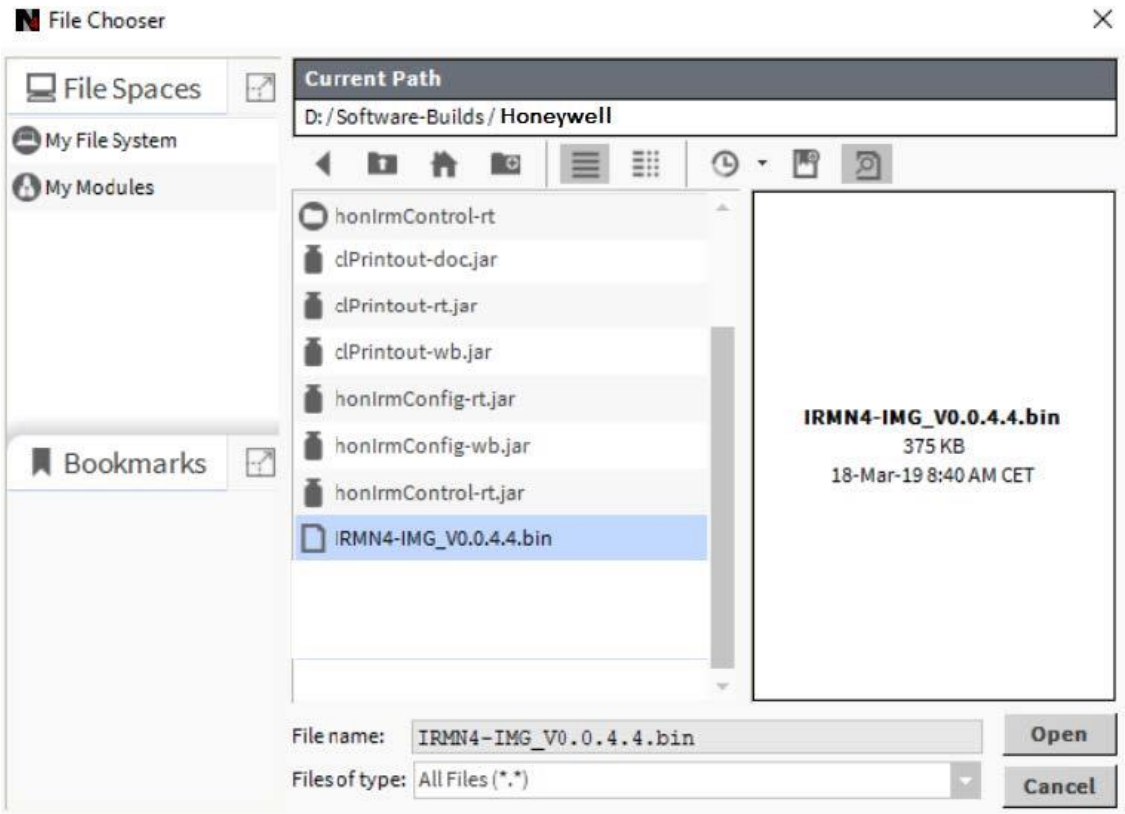
Allows updating the firmware in the controller via download.

Procedure

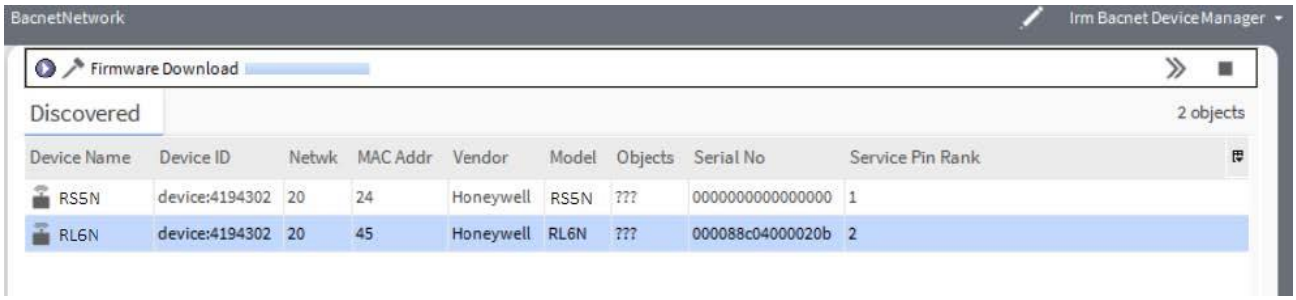
1. In the *Database* pane, select the controller.



2. Click **Download Firmware** button.
   RESULT: The *File Chooser* dialog box displays.



3. Navigate to the folder where the firmware file is located, then select the firmware file (IRMN4-IMG_Vxxxx.bin), and then click the **Open** button.



RESULT: The firmware download process is started as indicated on the top in the **Firmware Download** progress bar. When the firmware download is finished successfully, it is indicated by the ´Success` message.

ALARMING

Alarm configuration and details on how to use NC objects in BACnet points and generate alarms and get it in Niagara.

To use the alarming function, please refer to the N4.x BACNET UTILITIES DRIVER User Guide, form no. EN2Z-1020GE51.

MISCELLANEOUS

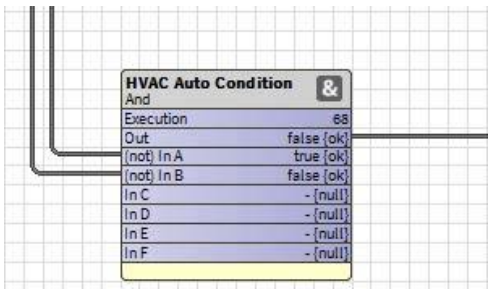
Value Updates after Device Power Failure

If a device had a power failure and the device is restarting, it will take 3 - 5 minutes until the values will be updated in the wire sheet. To accelerate the value update on the wire sheet, please manually refresh the wire sheet by navigating to another page and returning to the wire sheet.

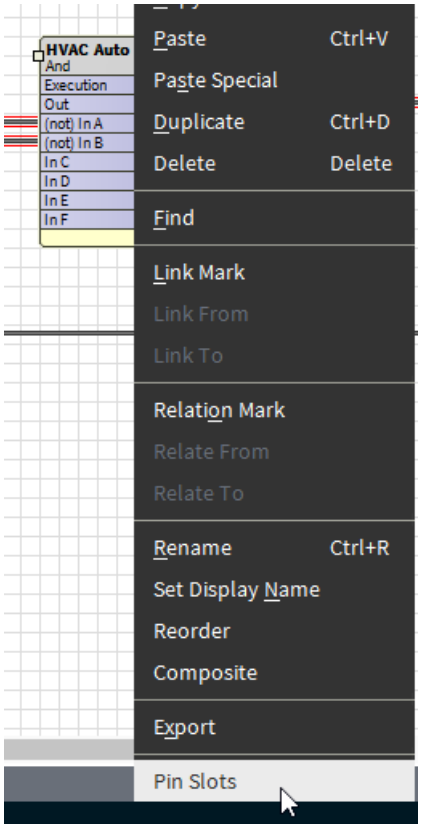
Hiding Slots

Procedure

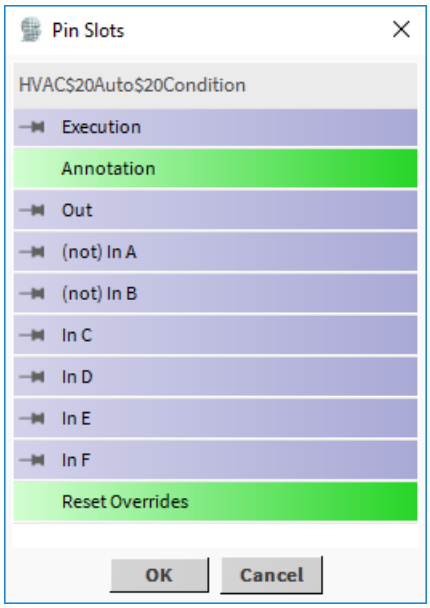
1. Right-click the function block icon, of which you want to hide slots.



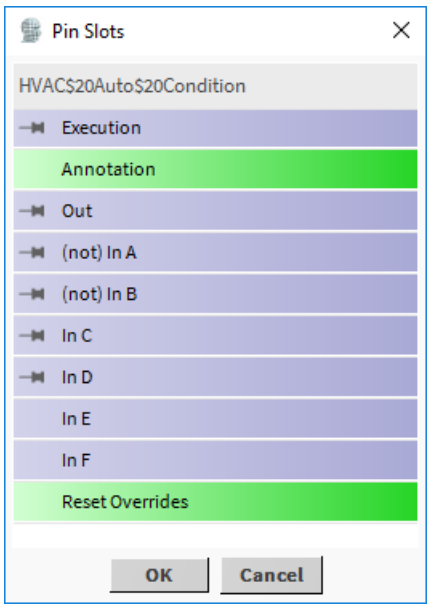
2. In the context menu, click **Pin Slots**.



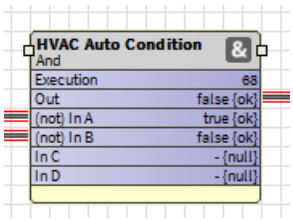
RESULT: The *Pin Slots* dialog box is displayed.



3. Click the slot you want to be hidden.
   RESULT: The pin icon at the selected slot will be removed.



4. Click **OK**.
   RESULT: The function block symbol redisplays. The hidden slots are removed from display.



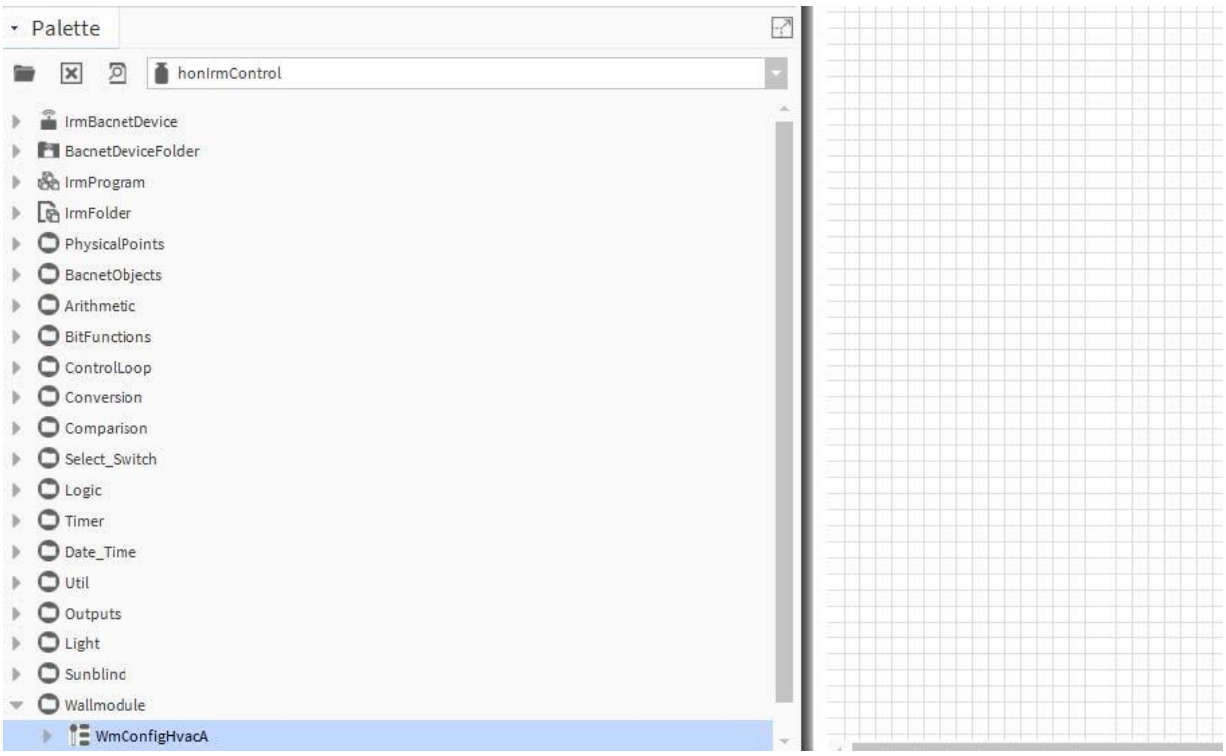
   NOTE: Any slots manually hidden in the Pin Slots dialog are not stored in the controller. After clearing the project and learning the application back from the controller, the hidden slots will be visible again.

Sylk Wall Module Usage

When using a Sylk or external wall module, only one wall module per controller can be added and its address is fixed to 1.

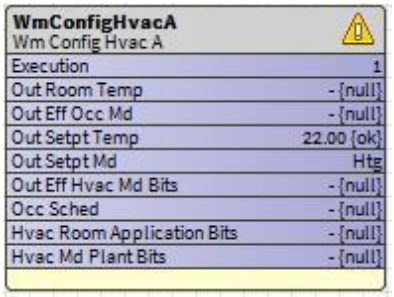
Procedure

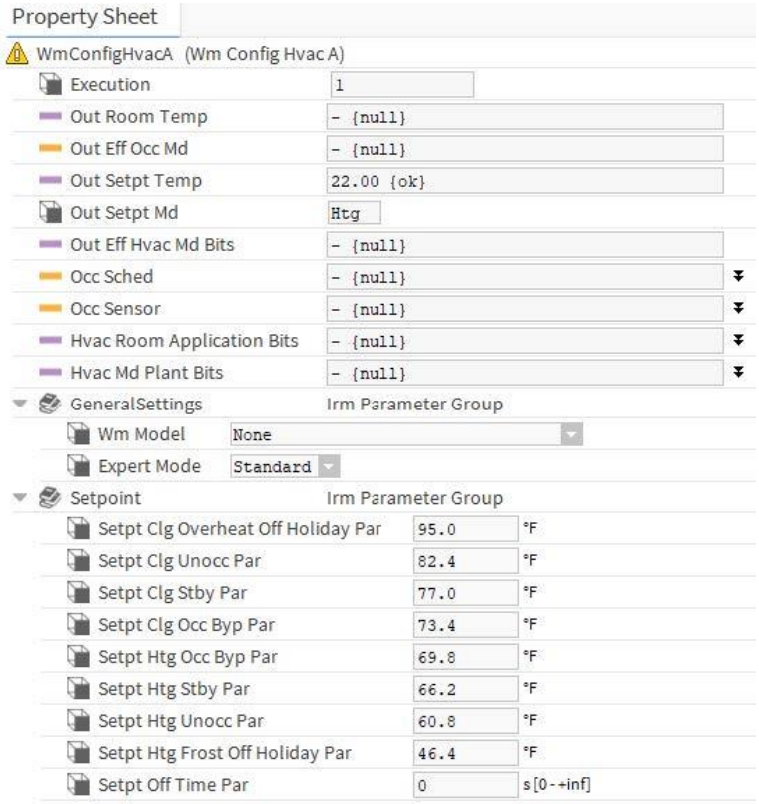
1. Double-click *Event Program* or *Periodic Program* in the tree.
2. Open the **honIrmControl** palette.
3. Scroll down to the *Wallmodule* group, and then expand it.
4. Select the ´WmConfigHvacA` entry and drag & drop it onto the wiresheet.



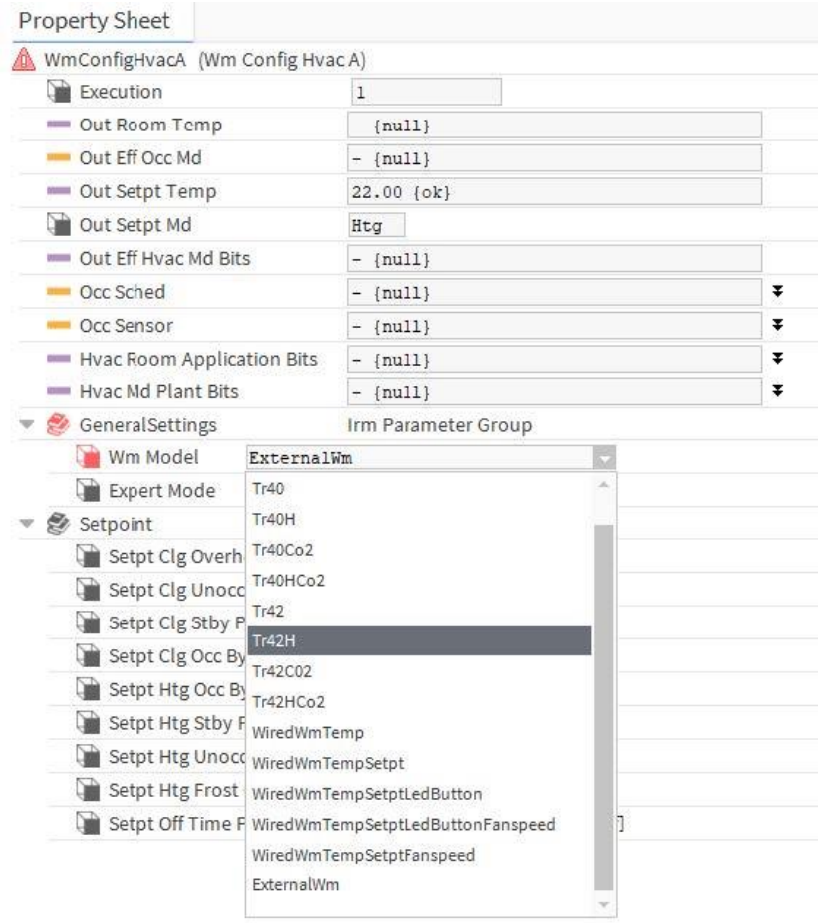
5. In the *Name* dialog box, change the name if desired and then click **OK**.

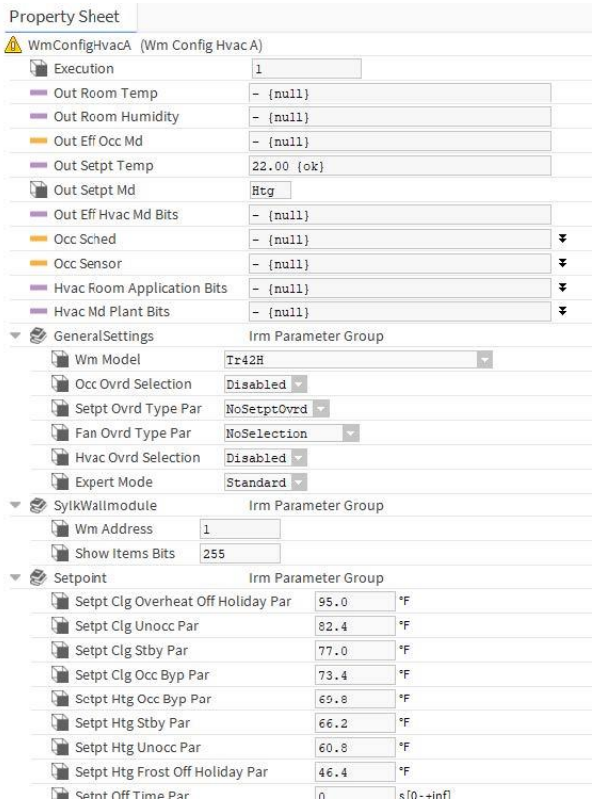
6. Double-click the **WmConfigHvacA** icon to display the Property Sheet.



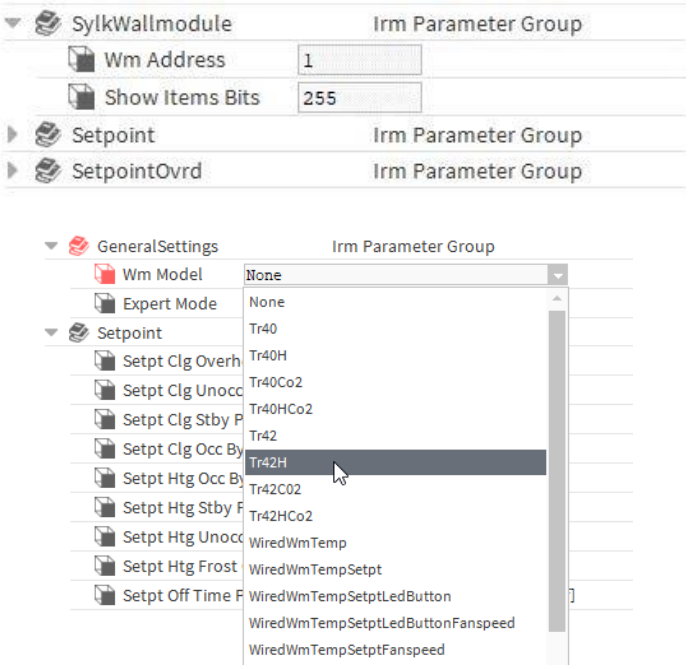


7. Once TR42H is saved, the window will pop up to configure the SylkwallModule address.





8. On the property sheet, select the sylk wall module type in **Wm Model**.
   RESULT: Under **SylkWallmodule**, the fixed Wm Address can be viewed.



For detailed information on configuration of wall modules, please refer the IRM-NX Application User Guide, form no. EN2B-041.

Manufactured for and on behalf of the BMS Division of Honeywell Products and Solutions SARL, Z.A. La Pièce, 16, 1180 Rolle, Switzerland by its Authorized Representative:

Honeywell Building Technologies
Honeywell International Inc.
1985 Douglas Drive North
Golden Valley, MN 55422
customer.honeywell.com

31-00282ES | Rev. 01 08-19

Honeywell