Documentation Basis

Pacom Unison

BACnet Devices Integration in Unison

Documentation Basis UDB-2025-0001-EN
Revision 1.02

2025-06-18

Contents

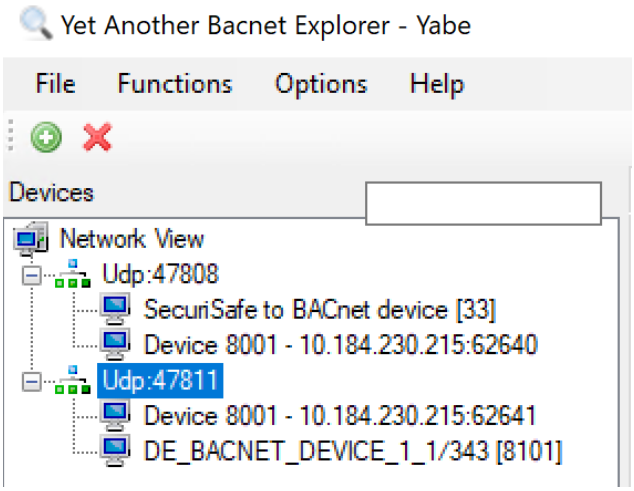
1 Introduction

This is the documentation for how a BACnet device can be integrated with Pacom Unison. This document is to be used as a preliminary documentation until the information is made available in Pacom Unison's official online documentation.
BACnet is a standardized communication protocol initially developed for building automation and control (BAC) networks. It has been an ASHRAE/ANSI standard since 1995 and is an ISO 16484-5 standard since 2003.

The BACnet integration in Unison is a generic integration, aimed to be possible to be use together with any BACnet compatible system. Due to its generic implementation, you will have to manually configure the BACnet objects imported to Unison to get the expected behaviour.

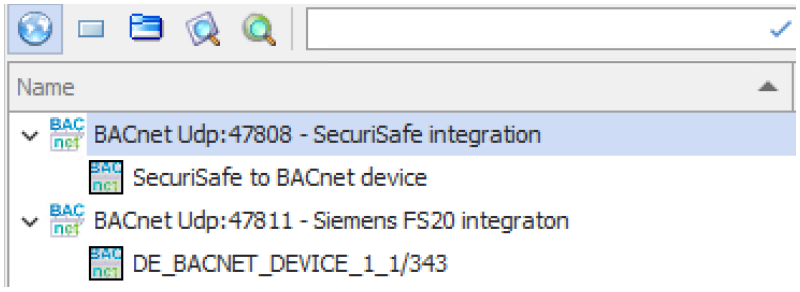
1.1 System overview

The Unison BACnet driver can use one single BACnet port/channel, so if you have different BACnet devices to connect to, you must create and use one Unison BACnet device for each BACnet port/channel. Looking at how the open source BACnet test client YABE works, it is easy to see the resemblance with how to configure this in Unison.



The YABE BACnet test client showing two started BACnet channels

Every channel as seen in the example from YABE above, that is the Udp:47808 and Udp:47811 objects, will be a separate device in Unison. Under each channel/port we will find the devices available.



Unison having the same two BACnet channels/ports seen in the YABE client, configured as two devices

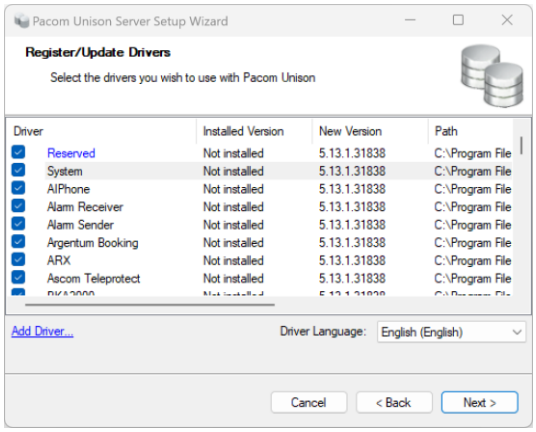
An example of how the same configuration looks in Unison can be found in the example above.

1 BACnet driver installation

The BACnet driver is currently being shipped as a separate driver package that you must install separately, either at the same time as you are installing Unison, or the driver can be installed and added to an existing Unison installation as well. If the driver is being installed during the first time installing Unison, please follow the instructions in chapter 1.1 and after the installation has been completed, continue to chapter 2. If you are installing the driver to an existing Unison installation, please follow the instructions from chapter 1.2.
The Unison BACnet driver is compatible with Unison v5.13.1.

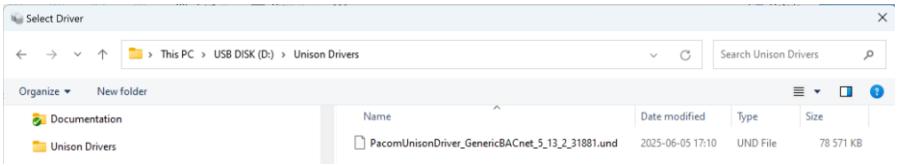
1.1 Installation during first time installing Unison

Install Unison according to instructions found in the Unison Installation Guide. When you come to the last part of the Unison Server Configuration wizard which is the **Register/Update Drivers** dialog.



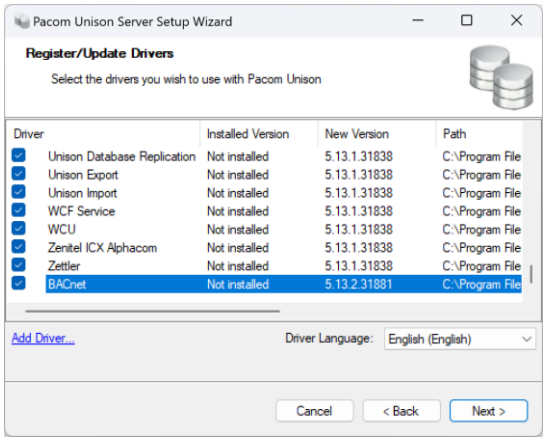
The Unison Server Configuration Wizard – Register/Update Drivers part

In this dialog select **Add Driver** found at the lower left-hand side and browse to find the BACnet driver package which is a file with the file extension *.und.
Select the file PacomUnisonDriver_GenericBACnet_5_13_2_31881.und and click OK.
Please note that the file can have different names depending on current build version, so the version number and build number can be different. Please ensure that you use the latest available version that is compatible with then version of Unison that you are using.



The BACnet driver und package

After returning to the Register/Update Driver dialog, you shall be able to find the BACnet driver at the bottom of the driver list.

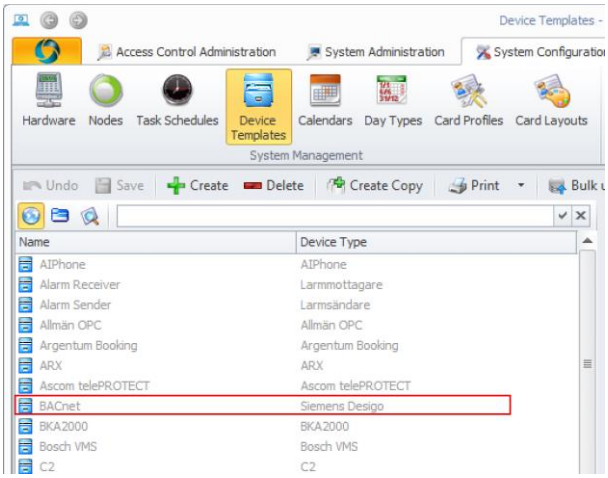


The BACnet driver added to the driver list

Please remember to select the correct driver language before completing the installation. It is also recommended to only install the drivers that you will use for the customer using this Unison. Once language and driver selections has been done, click Next and all the selected drivers will be installed.

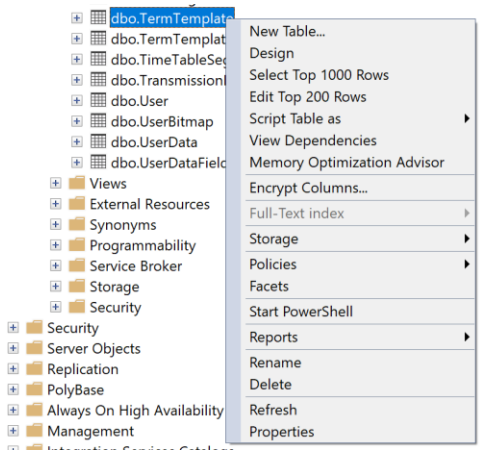
1.2 Preparation before installing in an existing Unison installation

If you are installing the BACnet driver in a Unison system that has been using a Unison version older than v5.11.5, you must probably change the name of a template in the database. If that is not done, the BACnet driver installation will fail due to a name conflict.
Start by checking the Device Templates dialog in Unison. If you have a template with the name BACnet, but the Device type is Siemens Desigo, you will have to rename this template before installation



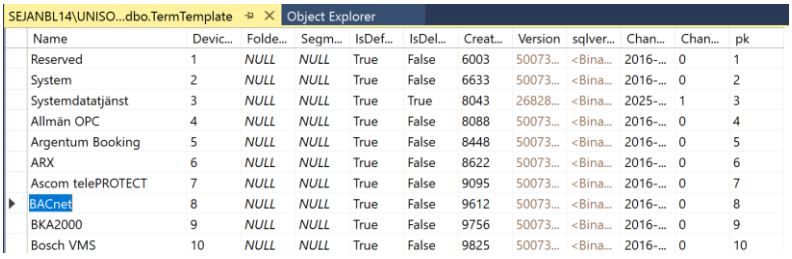
The Siemens Desigo template with wrong name

To change the template name, use the MS SQL Server Management Studio. Select the UnisonMain database and find the table dbo.TermTemplate. Right click this table and select **Edit Top 200 Rows**.



The UnisonMain table with template names

Find the name BACnet in the list of the device template names and select this row.



Name	Devic...	Folde...	Segm...	IsDef...	IsDel...	Creat...	Version	sqlver...	Chan...	Chan...	pk
Reserved	1	NULL	NULL	True	False	6003	50073...	<Bina...	2016-...	0	1
System	2	NULL	NULL	True	False	6633	50073...	<Bina...	2016-...	0	2
Systemdatatjänst	3	NULL	NULL	True	True	8043	26828...	<Bina...	2025-...	1	3
Allmän OPC	4	NULL	NULL	True	False	8088	50073...	<Bina...	2016-...	0	4
Argentum Booking	5	NULL	NULL	True	False	8448	50073...	<Bina...	2016-...	0	5
ARX	6	NULL	NULL	True	False	8622	50073...	<Bina...	2016-...	0	6
Ascom telePROTECT	7	NULL	NULL	True	False	9095	50073...	<Bina...	2016-...	0	7
BACnet	8	NULL	NULL	True	False	9612	50073...	<Bina...	2016-...	0	8
BKA2000	9	NULL	NULL	True	False	9756	50073...	<Bina...	2016-...	0	9
Bosch VMS	10	NULL	NULL	True	False	9825	50073...	<Bina...	2016-...	0	10

The template names as seen in MS SQL Server Management Studio

Change the name from BACnet to Siemens Desigo and press Tab. Once this is done, you will get a marking on that row that the data in that column has been changed.

SEJANBL14\UNISO...dbo.TermTemplate | Object Explorer

Name	Devic...	Folde...	Segm...	IsDef...	IsDel...	Creat...	Version	sqlver...	Chan...	Chan...	pk
Reserved	1	NULL	NULL	True	False	6003	50073...	<Bina...	2016-...	0	1
System	2	NULL	NULL	True	False	6633	50073...	<Bina...	2016-...	0	2
Systemdatatjänst	3	NULL	NULL	True	True	8043	26828...	<Bina...	2025-...	1	3
Allmän OPC	4	NULL	NULL	True	False	8088	50073...	<Bina...	2016-...	0	4
Argentum Booking	5	NULL	NULL	True	False	8448	50073...	<Bina...	2016-...	0	5
ARX	6	NULL	NULL	True	False	8622	50073...	<Bina...	2016-...	0	6
Ascom telePROTECT	7	NULL	NULL	True	False	9095	50073...	<Bina...	2016-...	0	7
Siemens Desigo	8	NULL	NULL	True	False	9612	50073...	<Bina...	2016-...	0	8
BKA2000	9	NULL	NULL	True	False	9756	50073...	<Bina...	2016-...	0	9
Bosch VMS	10	NULL	NULL	True	False	9825	50073...	<Bina...	2016-...	0	10

The template name has been changed

To save this information, press Tab several times until you reach the end of the row and the cursor moves to the next row. After that, the change of the name has been saved.

SEJANBL14\UNISO...dbo.TermTemplate | Object Explorer

Name	Devic...	Folde...	Segm...	IsDef...	IsDel...	Creat...	Version	sqlver...	Chan...	Chan...	pk
Reserved	1	NULL	NULL	True	False	6003	50073...	<Bina...	2016-...	0	1
System	2	NULL	NULL	True	False	6633	50073...	<Bina...	2016-...	0	2
Systemdatatjänst	3	NULL	NULL	True	True	8043	26828...	<Bina...	2025-...	1	3
Allmän OPC	4	NULL	NULL	True	False	8088	50073...	<Bina...	2016-...	0	4
Argentum Booking	5	NULL	NULL	True	False	8448	50073...	<Bina...	2016-...	0	5
ARX	6	NULL	NULL	True	False	8622	50073...	<Bina...	2016-...	0	6
Ascom telePROTECT	7	NULL	NULL	True	False	9095	50073...	<Bina...	2016-...	0	7
Siemens Desigo	8	NULL	NULL	True	False	9612	26828...	<Bina...	2025-...	1	8
BKA2000	9	NULL	NULL	True	False	9756	50073...	<Bina...	2016-...	0	9
Bosch VMS	10	NULL	NULL	True	False	9825	50073...	<Bina...	2016-...	0	10

The template name has been changed, and change is saved

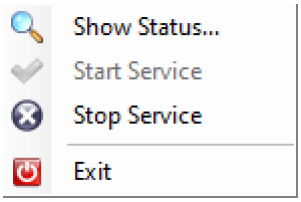
If you want to be really sure, you can check the Device Template dialog in the Unison client. The template name shall now be changed as seen in the example below. All of this can be done with Unison still running.

Name	Device Type
AIPhone	AIPhone
Alarm Receiver	Larmmottagare
Alarm Sender	Larmsändare
Allmän OPC	Allmän OPC
Argentum Booking	Argentum Booking
ARX	ARX
Ascom telePROTECT	Ascom telePROTECT
Siemens Desigo	Siemens Desigo
BKA2000	BKA2000
Bosch VMS	Bosch VMS
C2	C2

The Siemens Desigo template with the correct template name

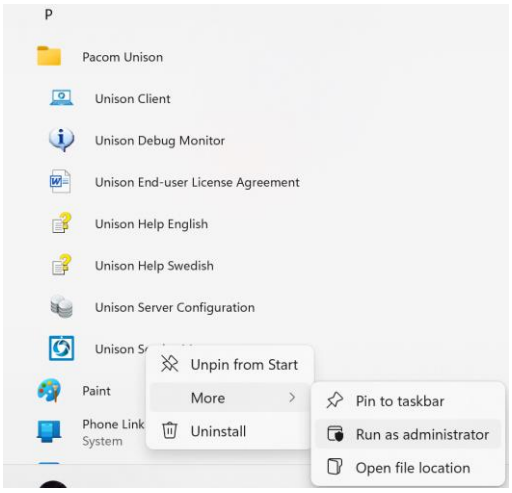
1.3 Installation in an existing Unison installation

To install the BACnet driver in Unison without shutting down the system, we can use the Unison Server Manager. To be able to do so, you must start the Unison Server Manager as an administrator. Start by stopping the Unison Server Manager by right clicking the tray icon and select **Exit**.



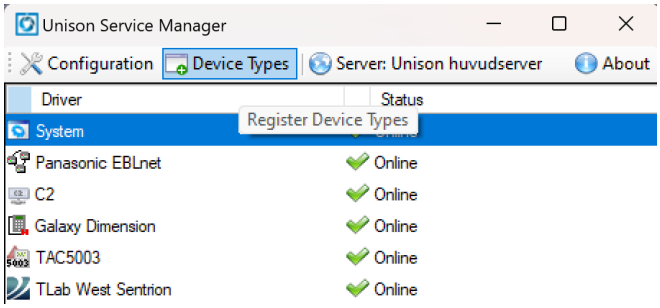
The Unison Server Manager

Find the Unison Server Manager from the Windows start menu and run the program as an administrator.



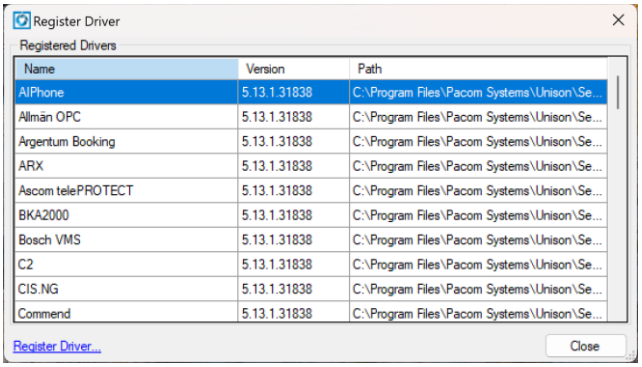
Run the Unison Server Manager as administrator

Open the Unison Server Manager UI by selecting **Show Status**. In this dialog, select **Device Types.**



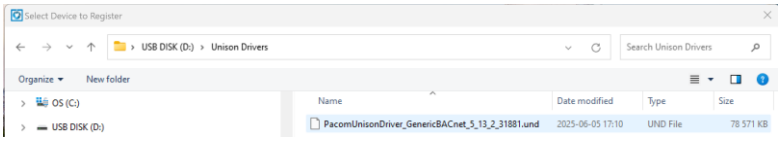
The Unison Server Manager user interface

In the register driver dialog, select the **Register Driver** button found at the lower left-hand corner.



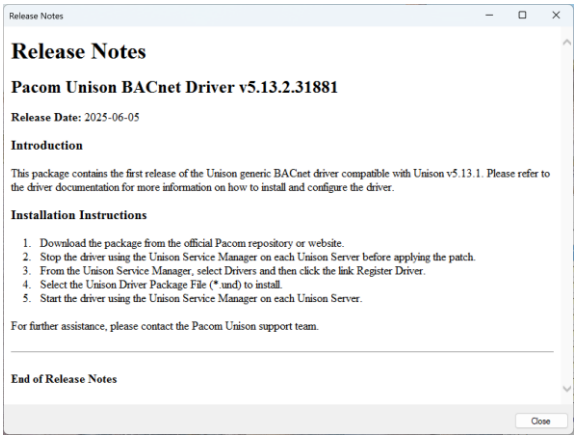
The Register Driver dialog

Browse to find the BACnet driver package which is a file with the file extension *.und.
Select the file PacomUnisonDriver_GenericBACnet_5_13_2_31881.und and click OK.
Please note that the file can have different names depending on current build version, so the version number and build number can be different from what is used in this documentation. Please ensure that you use the latest available version that is compatible with then version of Unison that you are using



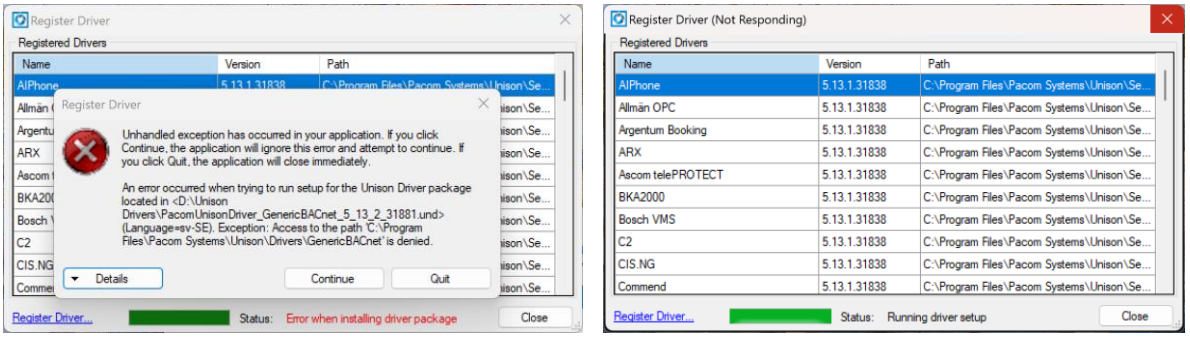
The BACnet driver und package

After selecting the driver package, a release notes information dialog is shown. Click Close to continue the installation.



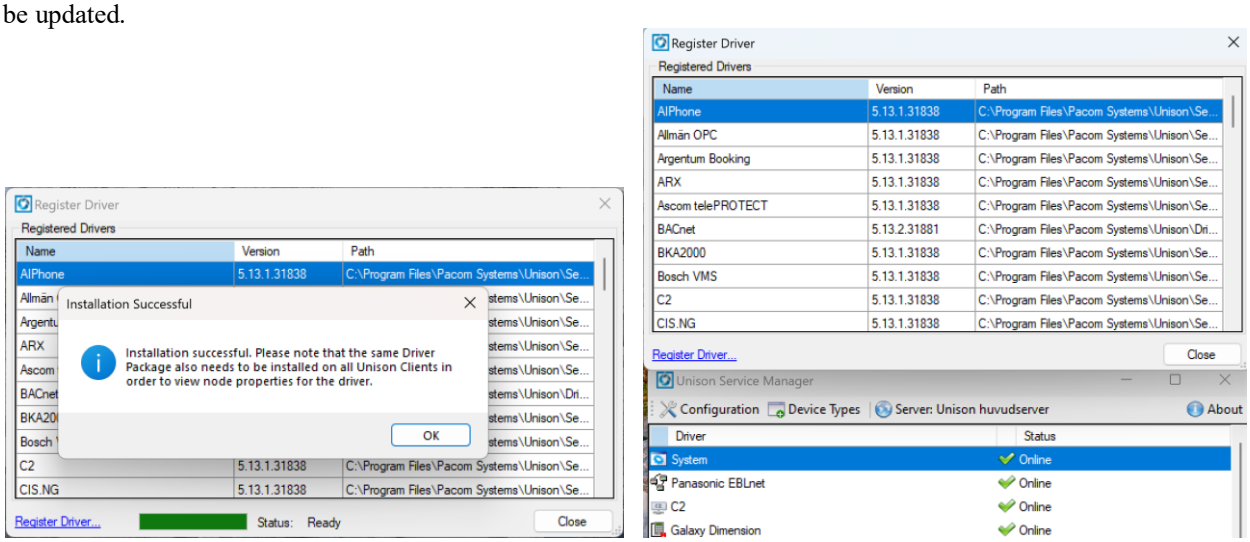
The driver release notes dialog

If you didn't start the Unison Server Manager as an administrator, you will at this point see the error message shown below. If this happens, please close the Unison Server Manager and try again. Otherwise, the driver will be installed as shown in the screenshot on the right-hand side below.



Driver installation failure and ongoing installation

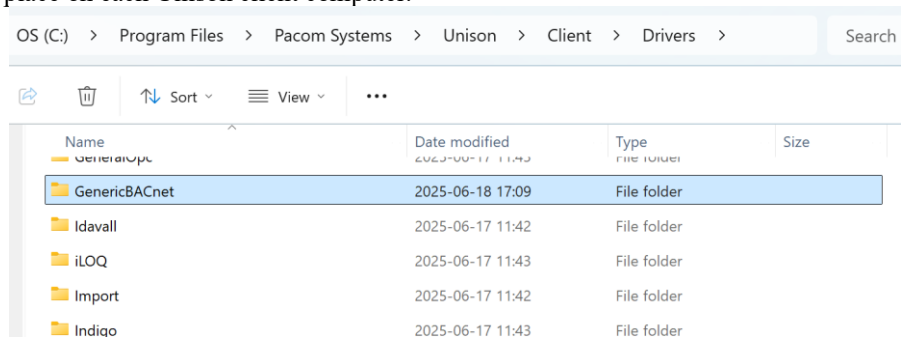
After a successful installation a message box dialog is shown. You can after this check that the driver has been added to the list of drivers. NB! Sometimes you must close Unison Server Manager and start it again for the list to be updated.



Successful installation message box and the BACnet driver listed among all other drivers

2 Client installation

Every computer having a Unison client installed, must also be updated with the client parts of the BACnet driver. Currently there is no installer or packages available for this, so the easiest way of doing this is to manually copy the driver client parts from the server installation and manually add them to each Unison client computer.
Copy the folder C:\Program Files\Pacom Systems\Unison\Client\Drivers\GenericBACnet and its content and paset this at the same place on each Unison client computer.



The client part of the BACnet driver

3 Driver Server installation

If the BACnet driver is to be used on a Unison driver server, the driver must be installed in the same ways as described in the previous chapters. Please see chapter 1.1 or 1.2 depending on whether it's a new installation or an upgrade installation.

4 BACnet driver configuration

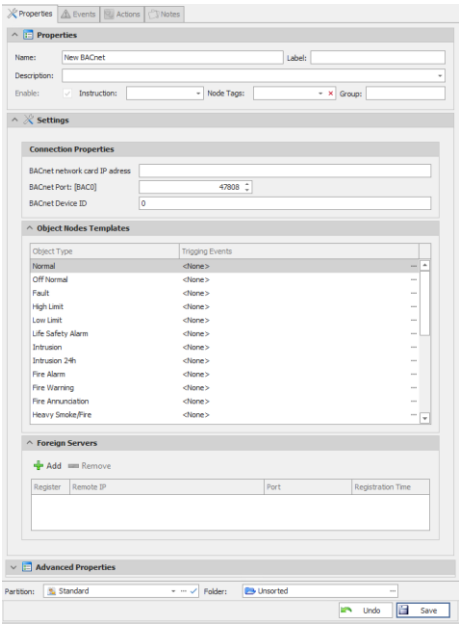
4.1 Step one– creating the driver

After creating a new BACnet driver in Unison, the driver property page as seen in the example below, is opened for the integration to be configured. Depending on what the BACnet integration is going to be used for and how the BACnet device that we are integrating with is configured, there are several things to prepare before the we can get the two systems to communicate and be able to execute a generate command to import data to Unison.

4.1.1 Connection Properties

The setting under this box must be configured to get the systems to communicate.

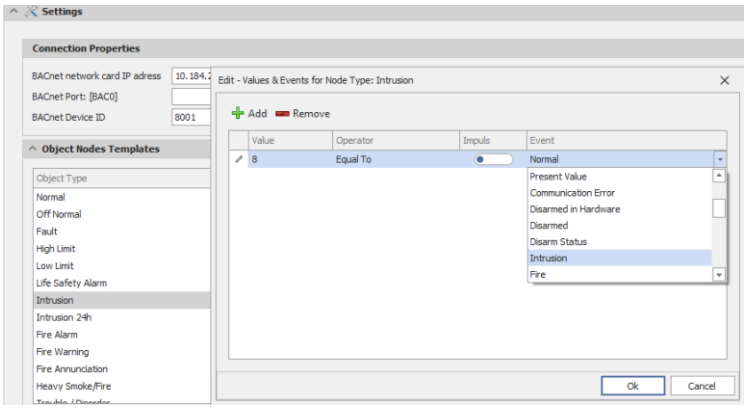
- BACnet network card address:
  Here we shall enter the IP address of the computer where the Unison BACnet driver is running.
  This is the local endpoint to which the BACnet device we want to integrate with, shall send it response to.
- BACnet Port:
  The BACnet port used for the communication. Must be configured according to how the BACnet device we are integrating with has been configured.
- BACnet Device ID:
  We shall assign Unison with a BACnet device ID that is unique in the system.
  Please note that some BACnet devices uses a whitelist configuration, so the BACnet device ID we enter her, must be part of that whitelist.



The Unison BACnet driver

4.1.2 Object Node Templates

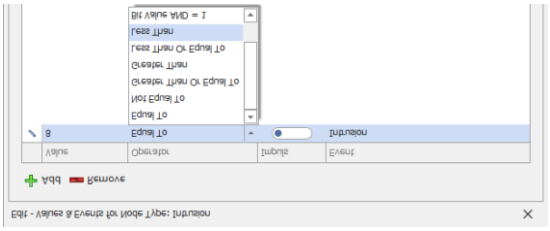
This part is used to pre-configure the event that shall trigger a specific Unison BACnet object. This can be done either before or after we execute the generate command to import data from the BACnet device to ease the configuration work. The event value to use for each object must be given from the BACnet system that we are integration with and in worst case scenario it must be investigated by testing the different objects.
It is possible to change the configuration manually for each individual BACnet object after the import has been done.
If you select an object type a new dialog will be opened where we can configure the event values to trigger this BACnet object and which Unison BACnet object event that shall be used.



Following parameters shall be configured:

- Value:
  The event value to use for the trigger

- Operator:
  How to evaluate the event value. Different options to choose from can be seen below.

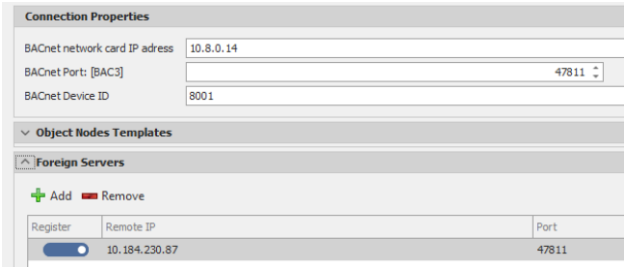


- Impulse
  If the event shall generate an impulse alarm in Unison or not.
  NB!
  If you use impulse, the event will generate an alarm, and if we acknowledge and reset the alarm from Unison, a new alarm will be triggered as long as the event is active at the BACnet device.
  This is the same behaviour we see when integrating with a fire alarm panel.
  The recommendation is to NOT use the impulse configuration. Downside to this is that an alarm must be manually reset at the BACnet device, before it can be reset in Unison.

- Event:
  Configuration of which Unison BACnet object node event the event value we have configure shall trigger.

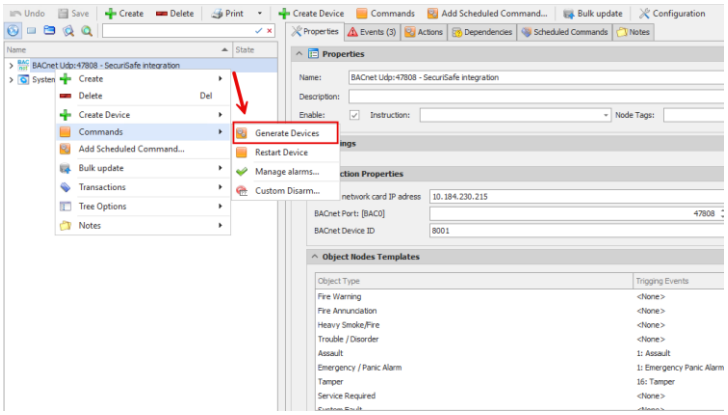
4.1.3 Foreign Servers

If the BACnet system we are integrating with uses Foreign Servers/ BBMD's, which stands for BACnet Broadcast Management Devices, we must configure the Unison BACnet driver with this information for the communication between the systems to work.
In the example below, the Unison server/local endpoint uses IP address 10.8.0.14, and the BACnet device, that we want to communicate with uses the address 10.184.230.87, which must be registered as a Foreign Server.



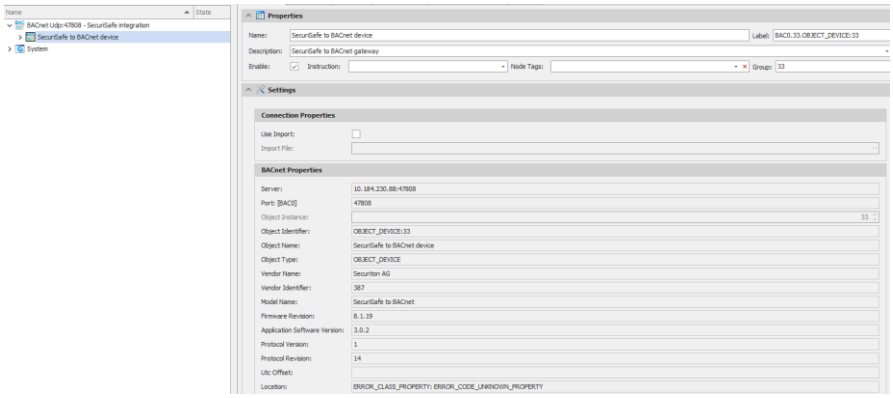
4.2 Step two– generate the server node

When the driver configuration has been completed adding all information needed and the Object Node Templates that we want to use have been configured, it is time to choose the command Generate Devices, to connect to BAC net device and import its data.



The BACnet driver with the Generate Device command

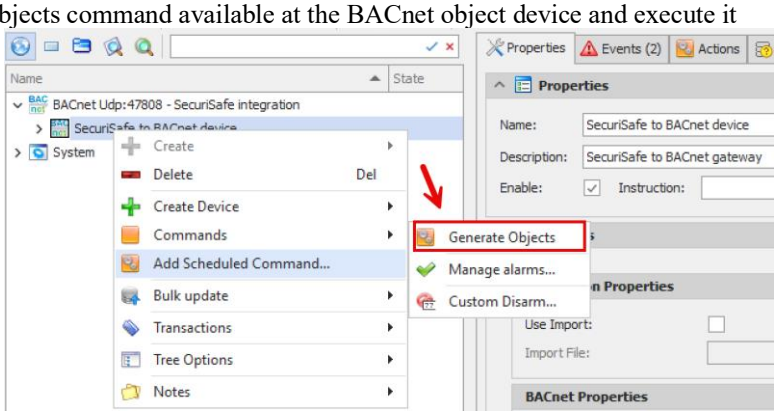
After selecting the Generate Device command, the driver will try and import the BACnet device server data found at the port we have configured to use. In the example below we can see that BACnet OBJECT_DEVICE has been found, imported and created in Unison. The import automatically creates a node label consisting of the Port, BACnet ID, Object type and Object ID. In this example, the ID is the same, but this differs when looking at other imported BACnet objects.
Under the Settings section, all available relevant BACnet properties for this object are presented. What kind of information we can find here depends on the brand of BACnet device we are integration with.



An example of how a BACnet object device can look.
Here exemplified with a Securiton SecuriSafe device

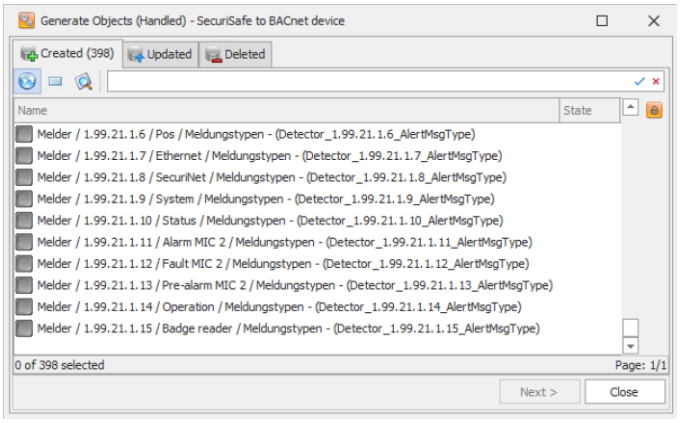
4.3 Step three– generate the BACnet object nodes

Once the BACnet object device has been imported, it is time for step thee in the generation and import of objects. Select the Generate Objects command available at the BACnet object device and execute it



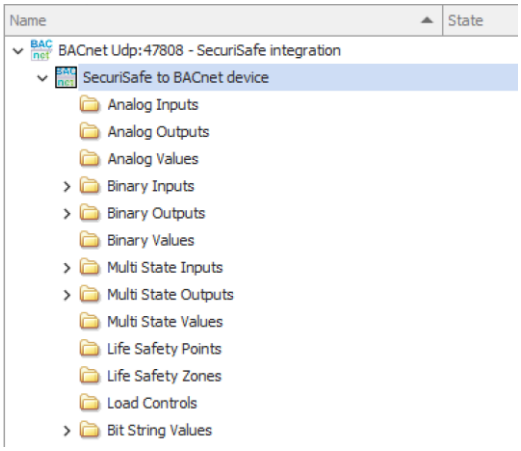
The BACnet object device node with the Generate Objects command

This command can take some time to execute and how long it takes depends on the number BACnet objects that we are going to import from the BACnet device. During the import of the BACnet objects the Generate Objects/Import data progress dialog will show the objects being imported and the progress.

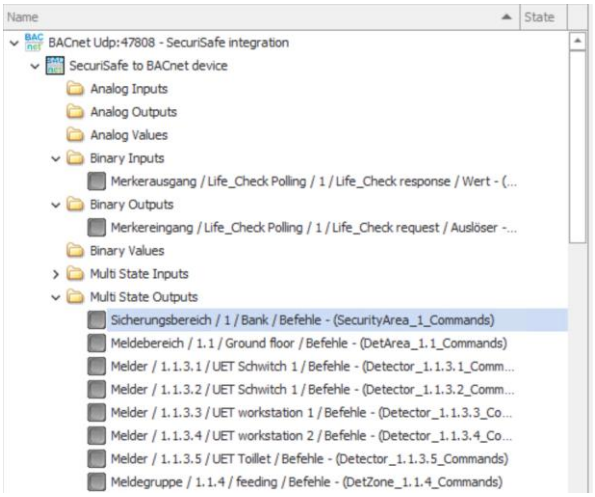


The Generate Objects progress dialog showing a completed data import

When the import of the BACnet objects have been completed a lot of folders have been created under the BACnet object device node. These folders contain different BACnet objects and what kind objects we will find here depends on the brand and type of BACnet device we are integration with.
NB! It's recommended to disable and enable the driver after importing the BACnet database to get communication between the systems stabilized. If not done, you can experience communication errors.



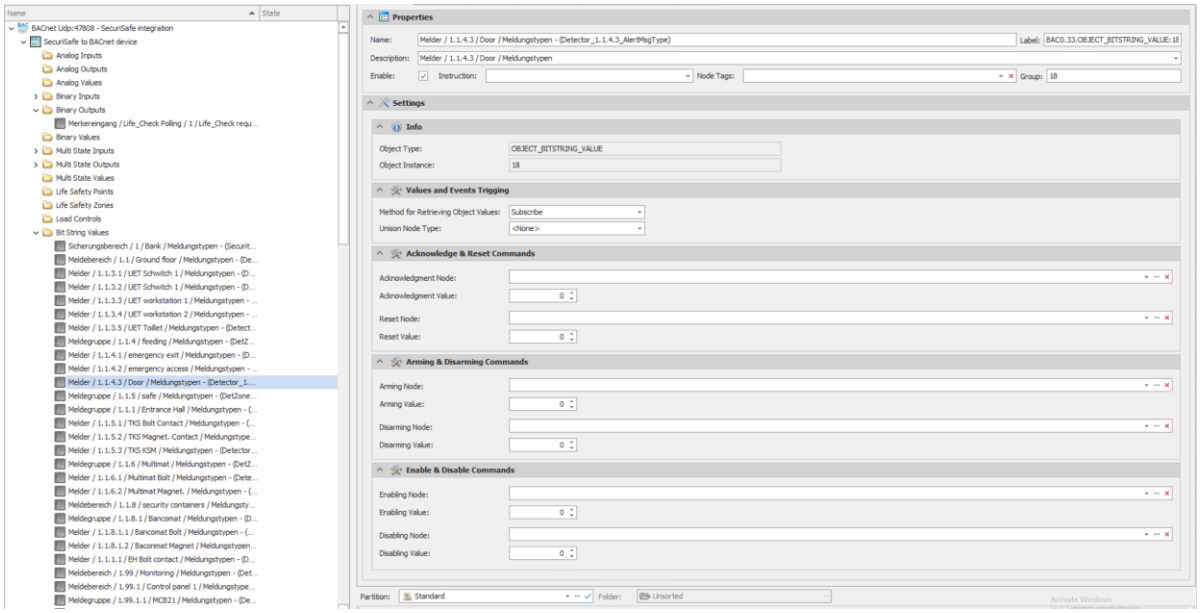
An example of how the folders for BACnet objects imported to Unison looks.



An example of how BACnet objects after importing objects from a Securiton SecuriSafe system can look like

4.4 Step four – Configuration of the BACnet nodes

Now it's time for the trickier part of the configuration of BACnet objects in Unison. How tricky this can be, depends on the brand and type BACnet device that we are integrating with. In the picture below we can see what kind of BAC net object that we are looking at right now and which BACnet ID this object has. We can also see the different configuration parameters that we can use for this BACnet object, and we here will try to describe how to use them.



These are the BACnet object node properties, where they come from and what it is used for:

Node Name	The node name is put together from the BACnet object description followed by the BACnet object name.
Node Label	The node label is put together with the Port, BACnet ID, Object type and Object ID
Node Description	The BACnet object description is saved as the node description, so we actually have the BACnet description saved both in the node name and the node description.
Object Type	Shows the BACnet object node type.
Object Instance	Shows the BACnet object identifier.
Method for Retrieving Object Values	Defines if Unison shall subscribe for events from this BACnet object. Normally Unison shall subscribe on events to be able to get notified of new events/alarms, but some objects does not support this, therefore you can disable this function. Please check what is valid for the system you are integrating with when configuring the system.
Unison Node Type	Here you define how Unison shall interpret and handle events from the integrated BACnet object. The configuration done here will use the setting done for the Object Node Templates as seen in chapter 4.1.2.

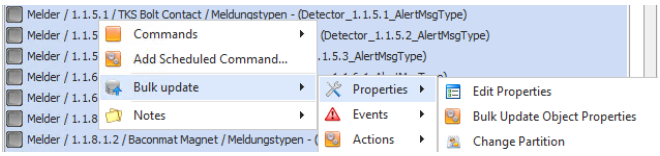
The following BACnet node properties are used to be able to configure which BACnet object to send an event value to, to acknowledge an alarm, reset an alarm, disarm or arm the object and so on. Which BACnet node to send this event value to differs from one BACnet device to another, so you must know how the BACnet device that you are integration with works to be able to configure this correctly. Please also note, that a BACnet node might not support all the commands that can be configured. Please check the BACnet device configuration manual for the specific BACnet device you are integrating with.

Acknowledgement Node	Defines which BACnet object node Unison shall use to send the value for acknowledging an alarm.
Acknowledgement Value	The event value to send for acknowledgment
Reset Node	Defines which BACnet object node Unison shall use to send the value for resetting an alarm.
Reset Value	The event value to send for alarm reset
Arming Node	Defines which BACnet object node Unison shall use to send the value for arming a BACnet object
Arming Value	The event value to send for arming a BACnet object
Disarming Node	Defines which BACnet object node Unison shall use to send the value for disarming a BACnet object
Disarming Value	The event value to send for disarming a BACnet object

Enabling Node	Defines which BACnet object node Unison shall use to send the value for enabling a BACnet object
Enabling Value	The event value to send for enabling a BACnet object
Disabling Node	Defines which BACnet object node Unison shall use to send the value for disabling a BACnet object
Disabling Value	The event value to send for disabling a BACnet object

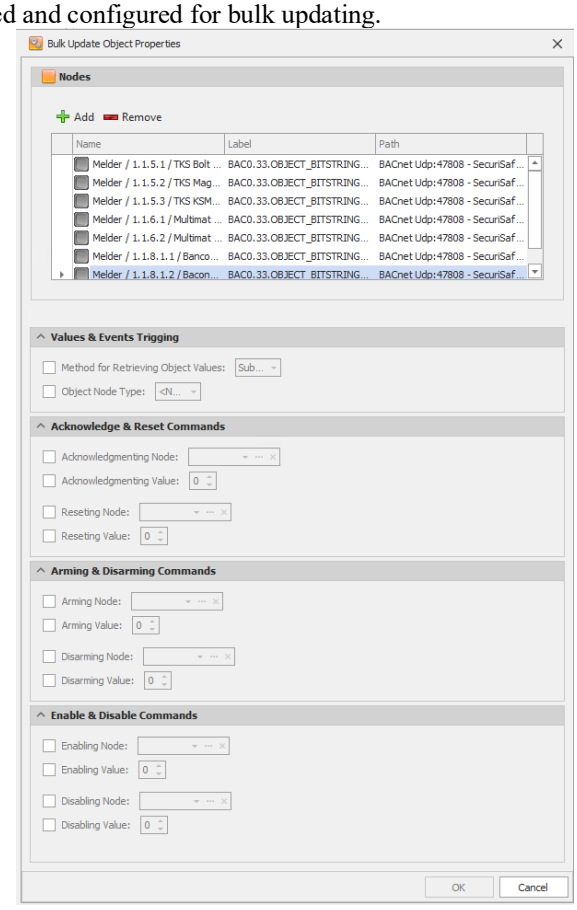
4.5 Bulk updating BACnet node properties

To make the configuration of a BACnet device easier, Unison supports bulk updating of the BACnet node properties as described in the previous chapter. Whether this is useful for the BACnet device being integrated, must be evaluated from device to device as each BACnet device differs in the way they are structured.
To use the bulk update feature, start by multiselecting the BACnet object nodes and select the command Bulk Update Object Properties.



The Builk Update Object Properties command

After the command has been selected, the following dialog is opened. Here all the properties as described in the previous chapter can be selected and configured for bulk updating.

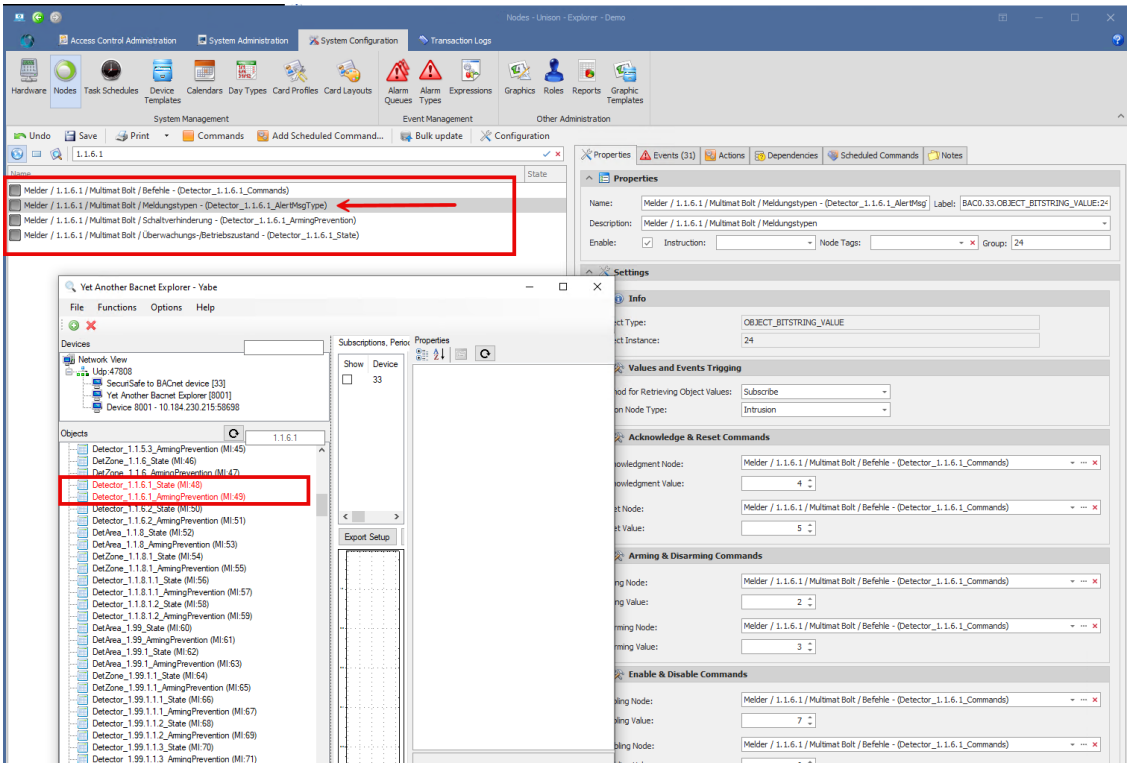


The bulk updating selection dialog

5 Integration Example – Securiton SecuriSafe

5.1 BACnet objects

The SecuriSafe BACnet device represents any of its devices such as a detector/input, area/zone and so on with four BACnet objects imported as four different nodes in Unison.
The Bitstring Value object which for SecuriSafe have name ending with “AlertMsgType” is the node that represents and handles the alarm. In the picture below we see an example with the four nodes filtered by using the address “1.1.6.1” and on top of the Unison client we see the YABE test client, where two objects have been marked in red as we are filtering that list with the same address.



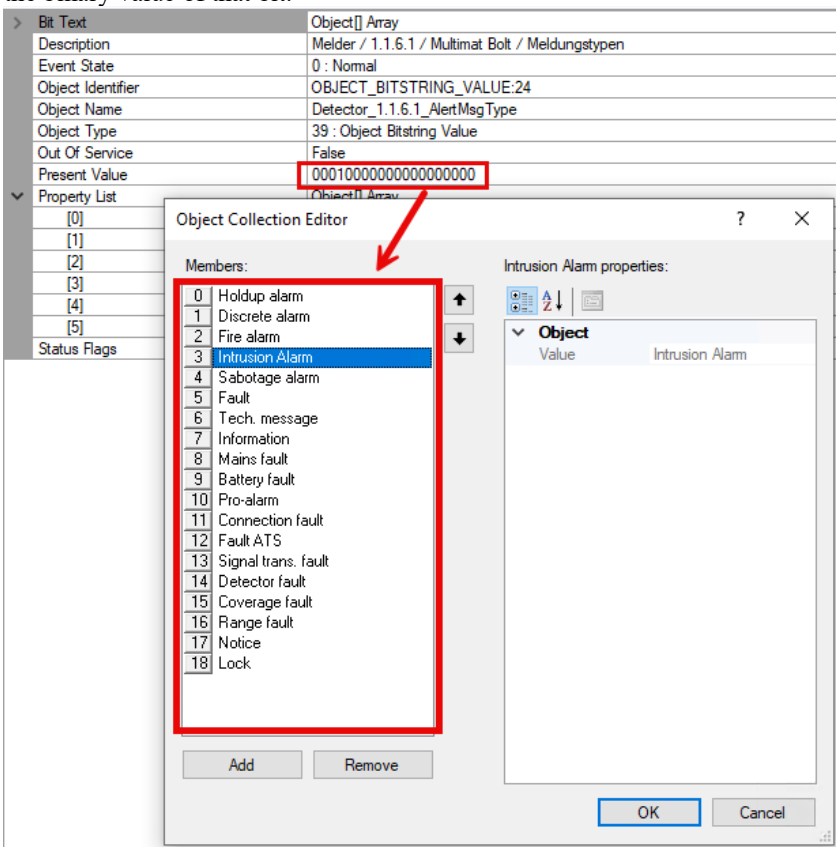
In Unison the BACnet object nodes will be used for this:

- The Bitstring_Value object node with “**AlertMsgType**” in its name (pointed out with an arrow above):
  This node shall be configured to present any alarm events from this object, and we shall also configure which object to use when writing values back to the BACnet device and configure which values to use to be able to acknowledge and reset an alarm. We can also define the BACnet object to use when for writing values for arming and disarming of the object as well as which values to use for enabling and disabling of the BACnet object.
- The Multi_State_Output object node with “**Commands**” in its name:
  This is the BACnet object node we shall use for all the commands we want to send to the BACnet device to be able to acknowledge and reset an alarm, arm and disarm the object and enable/disable the object.
- The Multi_State_Input object node with “**State**” in its name:
  Not really used in Unison but can be used for presenting the “Present Value” value which is a represents the current state.
- The Multi_State_Input object node with “**ArmingPrevention**” in its name:
  Currently not used in Union.

On the right-hand side of the screenshot above, we can see how the “**AlertMsgType”** node has been configured in Unison to be able to handle alarm events, acknowledge and rest the alarm, arm and disarm the object and enable/disable the object. Please see the following chapters for information on how to configure this.

5.2 Alarm events

If we investigate a Bitstring_Value object in the YABE test client, we can identify the alarm states supported for this object. The current state is represented by one single bit set at the "Present Value parameter and what kind of alarm event this represents is shown if you check the Bit Text Object Array. All of this seen in the example below. The value of each bit is the binary value of that bit.



Event values and what they represent

These are the values representing each event state, which also can be seen in the screenshot above.

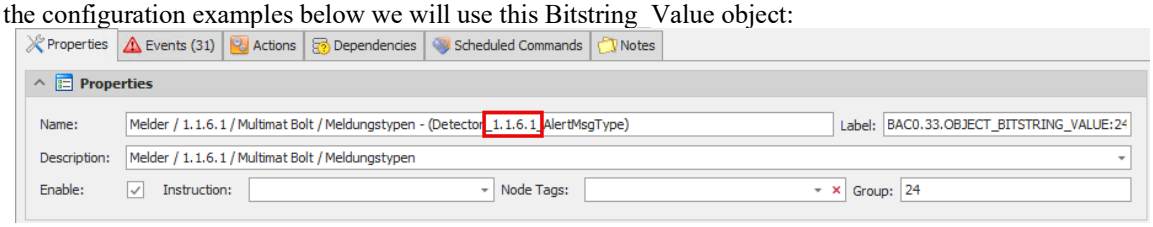
Bit	Value	Represents
0	1	Holdup alarm
1	2	Discrete alarm
2	4	Fire alarm
3	8	Intrusion alarm
4	16	Sabotage alarm
5	32	Fault
6	64	Tech. message
7	128	Information
8	256	Mains fault
9	512	Battery fault
10	1 024	Pro-alarm
11	2 048	Connection fault
12	4 096	Fault ATS
13	8 192	Signal trans. fault
14	16 384	Detector fault
15	32 768	Coverage fault
16	65 536	Range fault
17	131 072	Notice
18	262 144	Lock

The values above shall be configured as described in chapter 4.1.2 - Object Node Templates and after that the correct template shall be linked to the Unison Node Type parameter at the Bitstring_Value object node.

5.3 Command object and values

The SecuriSafe system uses the Multi_State_Output object node with "**Commands**" in its name as the object that we shall use to be able to send/write event values back to the BACnet device object for all the commands we want to support. As the SecuriSafe system is organized having a Bitstring_Value object corresponding to its Bitstring_Value object, this configuration is a bit tedious as it only partly can be bulk updated. Below is a description on the values to use for the different commands and an example on the easiest way to do this configuration

For the configuration examples below we will use this Bitstring_Value object:

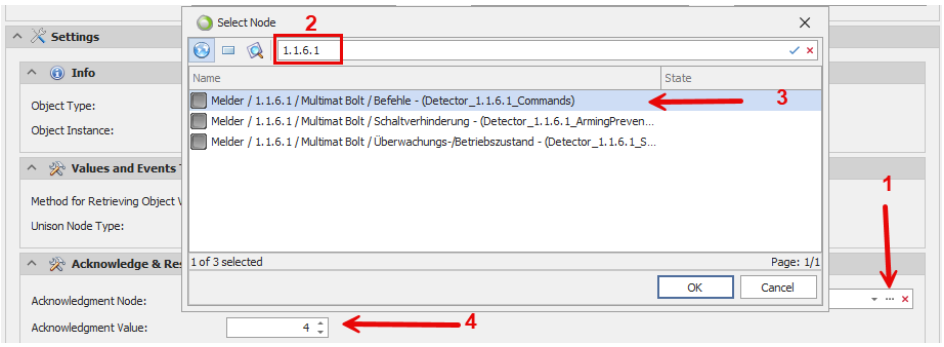


Especially not the address marked with a red square in the picture above. This address will be used to make it easier to link the correct Multi_State_Output "**Commands**" node to this Bitstring_Value object node.

5.3.1 Alarm Acknowledgment

The first thing to do is to select the correct "**Commands**" node.

1. Click the three buttons at the Acknowledgment Node field to open a node selection dialog.
2. Enter the address that the Bitstring_Value "**AlertMsgType**" object node uses in the search field to filter the node list.
3. Select the "Commands" node and click OK.
4. Enter the Acknowledgement value 4.



5.3.2 Alarm Reset

Select the same "Commands" node as the Reset node.
Reset value = 5

5.3.3 Arming

Select the same "Commands" node as the Arming Node.
Arming Value = 2

5.3.4 Disarming

Select the same "Commands" node as the Disarming Node.
Disarming Value = 3

5.3.5 Enabling

Select the same "Commands" node as the Enabling Node.
Enabling Value = 7

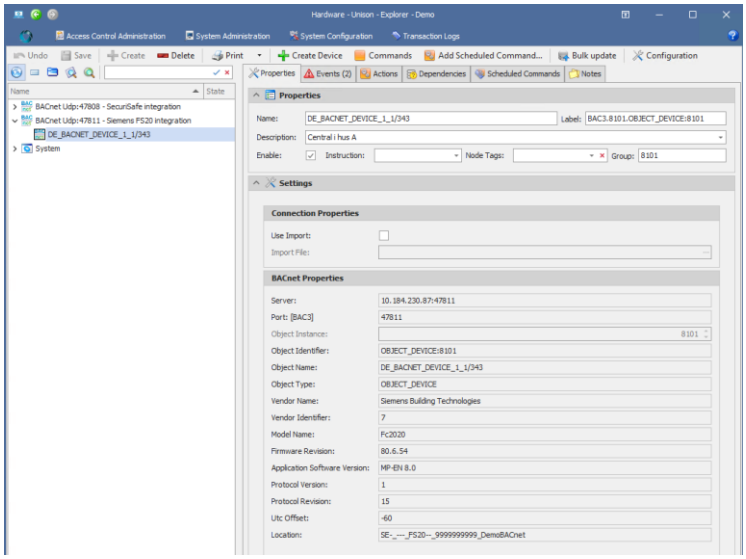
5.3.6 Disabling

Select the same "Commands" node as the Disabling Node.
Enabling Value = 6

Integration Example – Siemens FS20

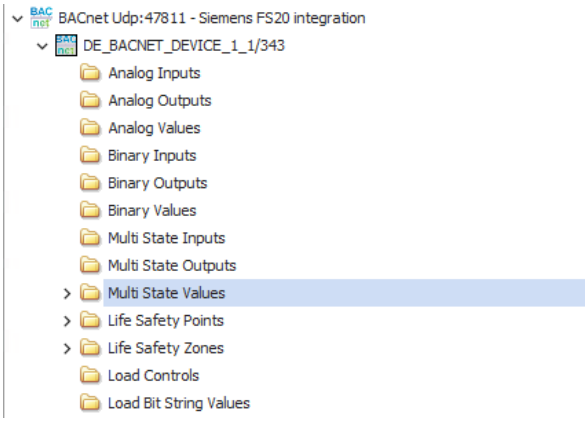
5.1 BACnet objects

Importing data from a Siemens BACnet device is done in several steps as described in the beginning of this document. After creating the BACnet driver node and generating the server node, a result as can be seen in the picture below shall be available.



A Siemens BACnet server node in Unison

Next step is to execute the generate command from the server node. After importing the BACnet objects, a tree structure as seen in the picture below shall be available.



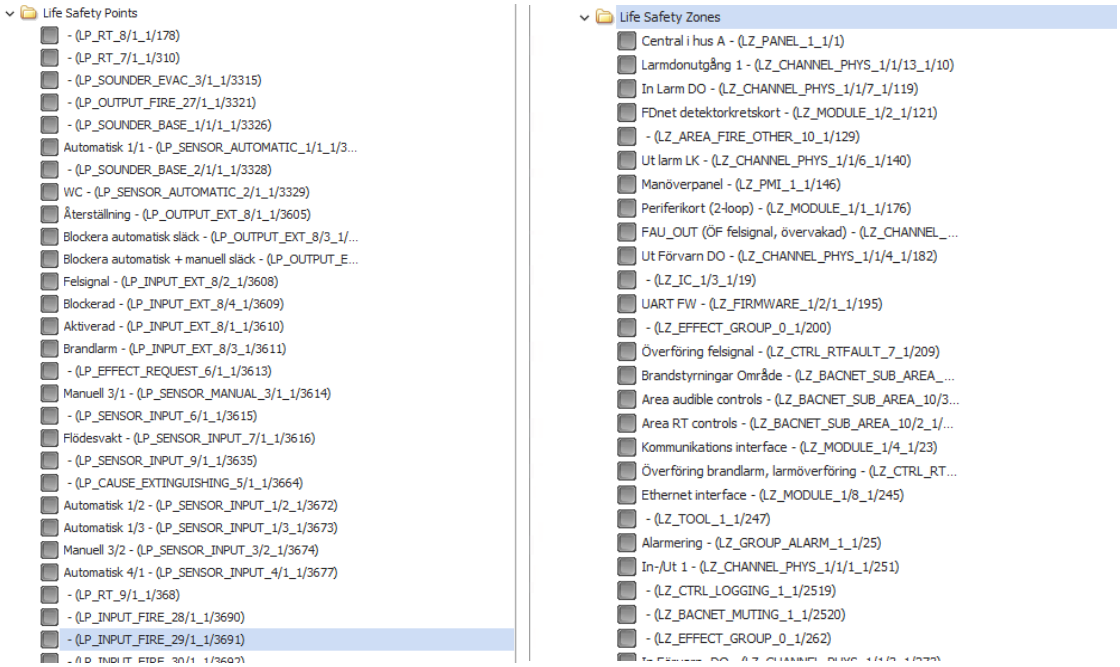
The BACnet object tree structure after importing data from a Siemens system

A Siemens BACnet device only uses three different BACnet objects

- Multi State Values:
  This/these objects are used to be able to send commands to the fire alarm system to be able to acknowledge and reset alarm.
- Life Safety Points:
  This folder contains objects that are used to represent fire alarm detectors and a lot of other different devices representing technical parts of the fire alarm system. Unison is only interested in using the objects that will be presenting and handling alarms, but all the other objects are imported and available anyway.
- Life Safety Zones:
  This folder contains object that are used to represent a fire alarm zone (section) and this folder also contains a lot of other different objects representing technical parts of the fire alarm system.

5.1.1 Life Safety Points and Life Safety Zones

The picture below shows example of how Life Safety Zone objects, and Life Safety Points can look like after being imported to Unison. The recommendation is to use the nodes found under the Life Safety Points folder in Unison as they represent a single detector. How to exactly setup the system must be decided together with personnel handling the Siemens fire alarm panel configuration.

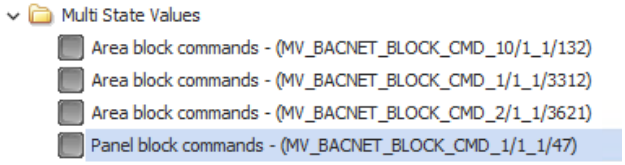


BACnet node objects in Unison imported from a Siemens fire alarm panel

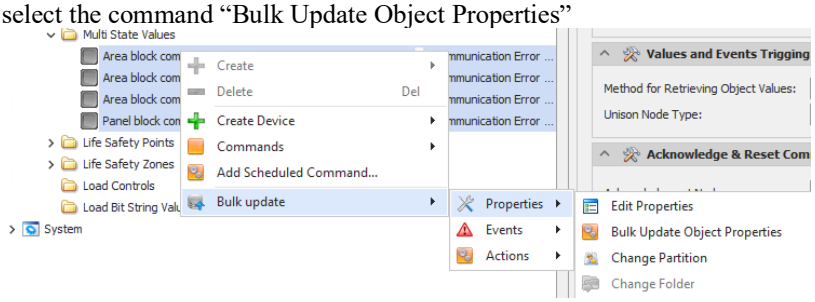
Looking at the examples above on the left-hand side, we see Life Safety Points that we want to configure to be used in Unison. The node object name consists of two parts from the BACnet object, the BACnet description and the BACnet object name. An example is the node “Manuell 3/1 - (LP_SENSOR_MANUAL_3/1_1/3614)” which is a manual fire alarm call point. All these kinds of object should be configured. If a node object name only contains the BACnet object name, such as the “- (LP_RT_9/1_1/368)”, these objects can be left unconfigured as they won’t be used in Unison. You can identify the objects that you can ignore as their node name all start with “-“.

5.1.2 Multi State Values

After importing a Siemens system, we will find four objects that can be used to send commands to the fire alarm panel. Unison will use one of these command objects as the node to send info to when acknowledge and reset an alarm. The BACnet object node that we shall use is the Panel block command as seen in the example below:

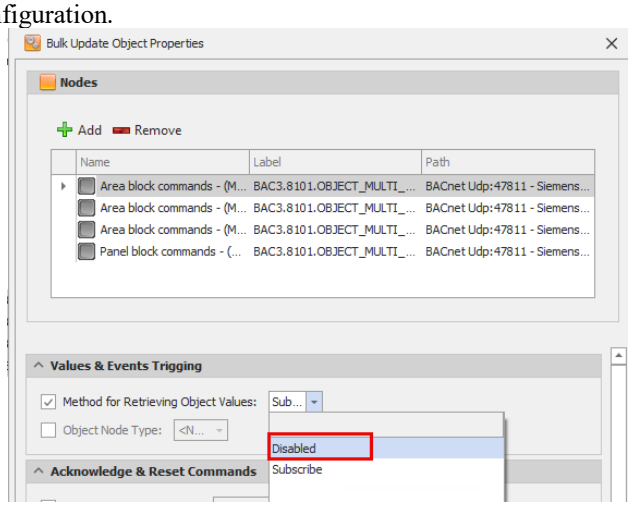


We must also change the configuration for these four objects, as we don’t want to subscribe for events from them. Multi select the and select the command “Bulk Update Object Properties”



Bulk update the Multi State Value objects

The BACnet object bulk update dialog is opened. Select the “Method for Retrieving Object Values”. choose “Disabled” and save this configuration.

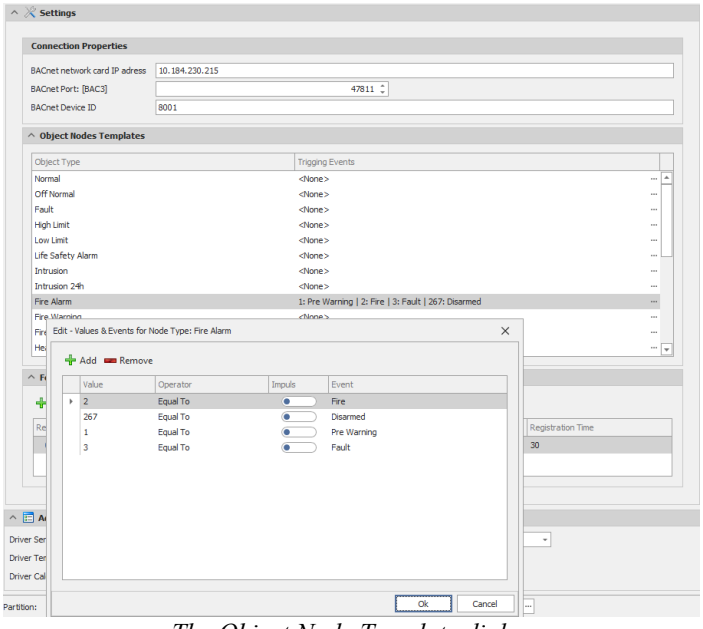


Bulk updating the “Method for Retrieving Object Values” parameter.

5.2 Alarm and status events

Next step is to configure the values we can receive from the Siemens system that represents different alarms and states. The Siemens system is easy to configure, as it uses different values for different alarms/states. In the example below we can see the four event values we will use. Configure this in the Object Node Template dialog for the Siemens BACnet device node at the Object Type **Fire**. These are the values and events we shall use:

Value	Represents
1	Pre Warning
2	Fire
3	Fault
267	Disarmed



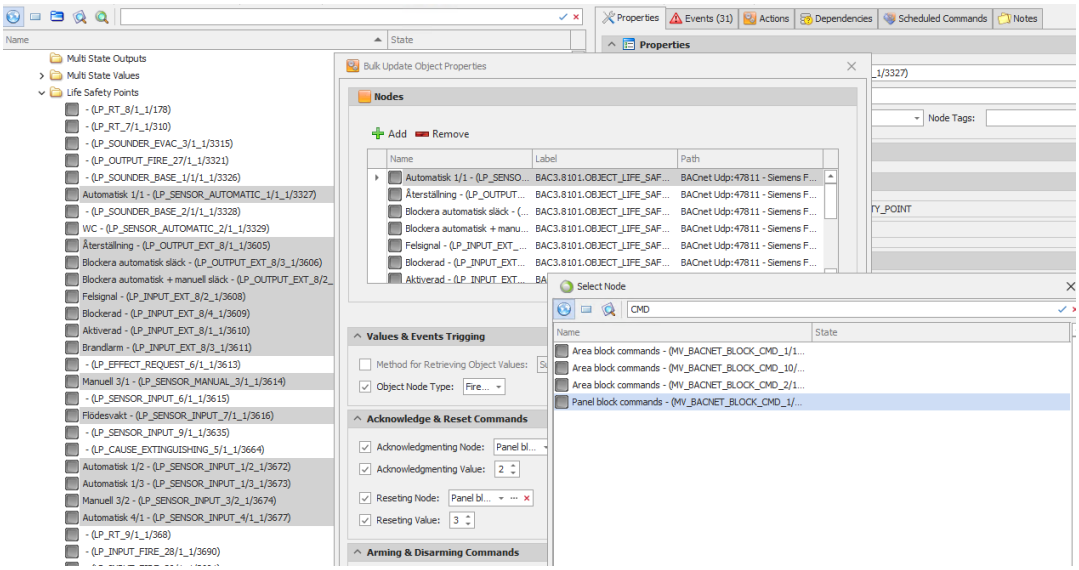
The Object Node Template dialog

Please note that the Impulse setting decides whether we will be able to reset and alarm from Unison or not. Impulse not activated, as seen in the picture above, means that you must reset an alarm from the fire alarm panel before it can be rest in Unison. Enabling the impulse functions means that you can reset the alarm immediately in Unison, but if the alarm still active, it will be activated in Unison again. We recommend not using the impulse function.

5.3 Command object and values

To be able to acknowledge and reset the alarms, we must configure the Multi State Value object to use for send command to the Siemens system on each of the Life Safety Point objects. We must also select the values to send/write to the Siemens BACnet system to do the acknowledge or the reset. This configuration is best done by using the bulk update feature available. To do this, do as follows:

1. Multi select the BACnet object nodes that you want to configure
2. Select the “Bulk Update Object Properties” command
3. The Bulk Update Object Properties dialog is opened
4. Make sure that the “Method for Retrieving Object Values” is configured for “Subscribe”. This is the default configuration, but you can bulk update this parameter at this stage if you want to be sure.
5. Mark the “Object Node Type” to be updated and select the “Fire alarm” type
6. Mark the “Acknowledging Node” to be changed.
7. Click the three dots on the right-hand side of the “Acknowledging Node” field and open the node selection dialog.
8. Search and select the Panel block command that we shall use to be able to send commands to the Siemens BACnet system.
9. Click OK
10. Set the “Acknowledging Value” = 2
11. Mark the “Reset Node” to be changed.
12. Click the three dots on the right-hand side of the “Reset Node” field and open the node selection dialog.
13. Search and select the Panel block command that we shall use to be able to send commands to the Siemens BACnet system.
14. Click OK
15. Set the “Reset Value” = 3
16. Click OK and execute the bulk update to save the selected configuration



Bulk updating the objects and values to use for acknowledging and reset of alarms

After this configuration has been completed, you can start testing the integration.

6 Known Issues

6.1 Securiton SecuriSafe

6.1.1 Arming and disarming

Arming and disarming a BACnet node object is done at the Bitstring_Value object, but the state presenting this is seen at the corresponding Multi_State_Input object, which is not the normal Unison behaviour.

6.2 Siemens FS20

6.2.1 Arming and disarming

It is not possible to arm/disarm a node from Unison as the Siemens system uses a “Mode” property for this that doesn’t align with the BACnet standard. Unison can though present if a zone/section detector has been disarmed/disabled.

Appendix A – Document Revisions

A.1 Revision 1.00 (2025-05-27)

Action	*Sign*
Initial version.	JB

A.2 Revision 1.01 (2025-06-16)

Action	*Sign*
Swap Ack and Reset values in text for Siemens	FE

A.3 Revision 1.02 (2025-06-18)

Action	*Sign*
Added information on how to install the driver using the und driver installation package and things to check before installation if you are going to use the driver in a Unison system with a database initially created using an Unison version older than v5.11.5.	JB