Powered by Niagara Framework®

SOFTWARE DOCUMENT

The BACnet EDE Workbench Plugin

Information and/or specifications published here are current as of the date of publication of this document. Tridium, Inc. reserves the right to change or modify specifications without prior notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia. Products or features contained herein are covered by one or more U.S. or foreign patents. This document may be copied by parties who are authorized to distribute Tridium products in connection with distribution fo those products, subject to the contracts that authorize such distribution. It may not otherwise, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior written consent from Tridium, Inc. Complete Confidentiality, Trademark, Copyright and Patent notifications can be found at http://www.tridium.com/galleries/SignUp/Confidentiality.pdf. © 2013 Tridium, Inc.

JACE, Niagara Framework, Niagara AX Framework and the Sedona Framework are trademarks of Tridium, Inc.

Getting Started

The following topics will get you started:

Introduction

The BACnet EDE Workbench Plugin extends the functionality of the NiagaraAX BACnet Driver. This document describes EDE files and the preparation, installation and operation of the plugin in the NiagaraAX Workbench.

The BACnet EDE Workbench Plugin provides functionality to allow BACnet EDE files to be used as a source of BACnet system data for the NiagaraAX BACnet driver which enables offline device and point discovery.

The plugin is installed in and operated from the Workbench only. It does not run in the remote JACE platform although it can operate on the JACE station.

Requirements

- Target platform running Niagara AX 3.7 or Niagara AX 3.8.
- Latest `bacnetEDE.jar` and `docBacnetEDE.jar` files in the `!modules` folder for the release of NiagaraAX you are using, where `!` replaces the folder path.
- Niagara AX license feature: “bacnetEde”.
- Four EDE files compatible with the EDE 2.2 specification, stored in the local file system and which are accessible to Workbench.

Using the Plugin

The following topics describe how to use the Plugin:

Adding an EDE Configuration

Offline discover operates on an EDE file set. You can select one or more different EDE file sets and each file set, along with its individual EDE file paths is stored as an EDE Configuration with a unique Configuration ID. You must create and add an EDE Configuration before undertaking offline discover.

Perform the following steps:

Step 1 Right-click the **BacnetNetwork** in the nav tree and select **View** > **Ede Bacnet Device Manager** to bring up the **Ede Bacnet Device Manager.**

Step 2 Click the **Discover** button to automatically learn what devices are defined in the EDE files. A popup **EDE Device Discovery** dialog appears.

Step 3 Click the **Add** button to bring up the **EDE Configuration** dialog.

Step 4 Enter a name for this configuration in the `Configuration ID` option.

Step 5 Click the down arrow next to the folder icon adjacent to the `EDE Master File Location` option and select `File Ord Chooser`.

Step 6 Navigate to and select the desired `MyProject_EDE.csv` EDE file.

Step 7 Repeat the previous procedure for the `Units File Location` and `State Texts File Location` options.

Step 8 Adjust, if necessary the `CSV Delimiter` option to match the configuration of your chosen EDE files. The default is Comma **,** .

Step 9 Click the **OK** button to save the **EDE Configuration** and return to the **EDE Device Discovery** dialog.

You have created an EDE Configuration which can be used for offline discover.

Editing an EDE Configuration

Each EDE Configuration can be edited if necessary to adjust the EDE file path details, EDE Configuration ID or the CSV delimiter.

Perform the following steps:

Step 1 From the Workbench **Tools** menu, select **Tools > Bacnet EDE**.

The default **Property sheet view** displays all the EDE Configurations.

Step 2 Make any required edits to the EDE Configurations and when finished, click the **Save** button.

Using offline Discover to add Bacnet Devices

The EDE files contain data that defines BACnet devices. You use the BacnetNetwork's **Ede Bacnet Device Manager** view to firstly select an EDE Configuration containing an EDE file set and then discover the BACnet Device components from it so that Bacnet devices can be added to the BacnetNetwork in a similar manner to Bacnet Online discovery.

Prerequisites:

- At least one EDE Configuration.

Perform the following steps:

Step 1 Right-click the **BacnetNetwork** in the nav tree and select **View** > **Ede Bacnet Device Manager** to bring up the **Ede Bacnet Device Manager.**

Step 2 Click the **Discover** button to automatically learn what devices are defined in the EDE files.

A popup **EDE Device Discovery** dialog appears.

Step 3 Click the down arrow in the `EDE Configuration Selection` and make a selection from the drop down list.

Step 4 Click the **OK** button to initiate the discovery process.

A progress bar appears at the top of the view and updates as the discovery occurs.

When the discovery job completes, discovered BACnet devices are listed in the *top pane* of the view, in the "Discovered" table.

NOTE: This works the same as most Device Manager views. For details, see the section "About Device Discover, Add and Match (Learn Process)" in the Drivers Guide. To add learned devices, see the section "To add discovered BACnet Devices" in the Bacnet Guide.

Using offline Discover to add Bacnet proxy points

The EDE files contain data that defines BACnet objects. You use the BacnetNetwork's **Ede Bacnet Point Manager** view to firstly select an EDE Configuration containing an EDE file set and then discover the BACnet Objects from it so that Bacnet proxy points can be added under any BacnetDevice in a similar manner to Bacnet Online discovery.

Perform the following steps:

Step 1 In the **Ede Bacnet Device Manager**, in the **Exts** column, double-click the `Points` icon in the row representing the device you wish to explore.

This brings up the **Ede Bacnet Point Manager**.

Step 2 Click the **Discover** button to automatically learn what BACnet objects are defined in the EDE files.

A popup **EDE File Chooser** dialog appears.

Step 3 Click the down arrow in the `EDE Configuration Selection` and make a selection from the drop down list.

NOTE: Ensure you select the same EDE Configuration as used previously for the EDE Device discovery.

Step 4 Click the **OK** button to initiate the discovery process.

A progress bar appears at the top of the view and updates as the discovery occurs.

When the discovery job completes, discovered BACnet objects are listed in the *top pane* of the view, in the "Discovered" table. Each BACnet object in the device occupies one row.

NOTE: This works the same as most Point Manager views. For details, see the section "About Point Discover, Add and Match (Learn Process)" in the Drivers Guide. To add learned devices, see the section "To add discovered BACnet objects as Bacnet proxy points" in the Bacnet Guide.

Using offline Discover to add Bacnet Histories

The EDE files contain data that defines BACnet Trend Logs. You use the BacnetNetwork's **Ede Bacnet History Import Manager** view to firstly select an EDE Configuration containing an EDE file set and then discover the BACnet Trend Log objects from it so that Bacnet history import components can be added to the BacnetDevice in a similar manner to Bacnet Online discovery.

Perform the following steps:

Step 1 In the **Ede Bacnet Device Manager**, in the **Exts** column, double-click the `Trend Logs` icon in the row representing the device you wish to explore.

This brings up the **Ede Bacnet History Import Manager**

Step 2 Click the **Discover** button to automatically learn what BACnet Trend Log objects are defined in the EDE files.

A popup **EDE File Chooser** dialog appears.

Step 3 Click the down arrow in the `EDE Configuration Selection` and make a selection from the drop down list.

NOTE: Ensure you select the same EDE Configuration as used previously for the EDE Device discovery.

Step 4 Click the **OK** button to initiate the discovery process.

A progress bar appears at the top of the view and updates as the discovery occurs.

When the discovery job completes, discovered BACnet Trend Log objects are listed in the *top pane* of the view, in the "Discovered" table. Each BACnet Trend Log object in the device occupies one row.

NOTE: For general information on this device extension, see "About the Histories extension" in the Drivers Guide and also see the section "BACnet Trend Log import notes" in the Bacnet Guide.

EDE and NiagaraAX

The following topics describe how NiagaraAX interacts with EDE:

Background to EDE

The BACnet Interest Group (BIG-EU), the European trade association for the application of the global BACnet standard ISO 16484–5, defined a format for distributing BACnet data-point lists. Here is a brief background to the concept of their definition.

In a multi vendor system, engineering data needs to be exchanged between interacting parties. In a BACnet system, it is normally expected that engineering data, such as data point object properties is exchanged by means of BACnet online discovery which is supported by most server and client devices. However this data exchange by definition, can only be accomplished online and is also dependent upon the BACnet discovery functionality being available in both the server and client. When this functionality is unavailable, It has proved necessary to obtain the engineering data in an offline condition. It is also needed when the server device has to be setup with its engineering data before it is installed and operational on a BACnet site network or when the client devices are unavailable for online discovery.

It was with this in mind that in 1999, the BIG-EU began a project to define a data format so that BACnet engineering data could be made available. The original concept was to provide only the BACnet data-point list in a simple human readable form via a Comma-Separated Value (CSV) file. The general format actually consists of four different CSV sheets and collectively they are commonly known as the EDE (Engineering Data Exchange) files, after the BIG-EU description document which defines them and is called *Engineering Data Exchange Template for BACnet Systems*.

BIG-EU continued to develop the EDE concept and in February 2007 the BIG-EU Technical Working Group at the Milano-meeting, approved Version 2.2 of the EDE-layout and its description. This is the latest version of the EDE files definition. Over the years that the EDE definition has been available, many BACnet equipment vendors have adopted the EDE file format to define and distribute the point content of their devices and some vendors have included an EDE file import mechanism in their server device to provide offline engineering programming.

NOTE: It is noted that whilst to date there is no other generally adopted equivalent solution, the EDE it is not extensive in its data content and its reliability cannot be guaranteed. You should be cautious and aware that the data in the EDE is not as extensive as the data discovered online and data discrepancies can occur when discovering online after the EDE has been initially used offline to build the server engineering database. The original concept of a human readable file containing the BACnet data-point list was never intended to be a machine-to-machine (M2M) data exchange format but it has become adopted by BACnet product vendors as the de facto standard for M2M data exchange.

EDE File Set

Here are the definitions of the EDE (Engineering Data Exchange) files as specified in the BIG-EU description document called *Engineering Data Exchange Template for BACnet Systems*. The EDE file set is a collection of Comma-Separated Value (CSV) files.

NOTE: The definition document indicates that the *Unit-Texts* and *Object-Types* files may be *"extended by adding Proprietary Units and Objects, according to the rules defined in the BACnet Standard"*. However there is no facility within the Niagara driver framework to generate Niagara point types to represent *Proprietary Objects*, neither is it capable to generate Niagara Facets to represent *Proprietary Units* and therefore any such extensions made to the files are unsupported by this Plugin.

EDE

The CSV file which includes, in its name *EDE*, contains project information and also the list of data points chosen for interoperation.

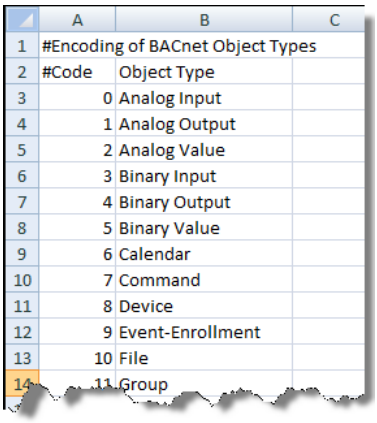
Figure 1. The EDE.csv file

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
6	#keyname	device obj.-	object-name	object-ty	object-ins	descrip	present-v	min-pres	max-pres	commanc	support	hi-limit	low-lir	state-text	unit-code	vendor
7	064 - UCU3213VAV	101064	064 - UCU3213VAV	8	101064											
8	064 - UCU3213VAVM	101064	MaintLimit	2	1										71	
9	064 - UCU3213VAVC	101064	ChwValve	2	9										98	
10	064 - UCU3213VAVR	101064	ReturnAirTemperat	0	3										62	
11	064 - UCU3213VAVS	101064	SetpointAdjust	0	4					Y					98	
12	064 - UCU3213VAVF	101064	FanSpeed	1	14										98	
13	064 - UCU3213VAVL	101064	LthwValveOutput	1	15										98	
14	064 - UCU3213VAVF	101064	FanRunStatus	5	2									1		
15	064 - UCU3213VAVS	101064	SupplyAirTemperat	5	3									2		
16	064 - UCU3213VAVR	101064	RunStatus	3	1									1		
17	064 - UCU3213VAVF	101064	FanEnable	4	9									1		
18	064 - UCU3213VAVP	101064	PIR	3	16									3		
19	064 - UCU3213VAVC	101064	COV 1	20	1											
20	064 - UCU3213VAVS	101064	Seconds from midr	20	4											
21	064 - UCU3213VAVS	101064	System Alarm 0	15	0											
22	064 - UCU3213VAVS	101064	Strategy Alarm 1	15	1											
23	064 - UCU3213VAVR	101064	ReturnAirTemperat	5	6									2		
24	064 - UCU3213VAVS	101064	SupplyAirTemperat	2	12										62	

Object-Types

The CSV file which includes, in its name *Object-Types*, contains a list of supported BACnet Object Types. The file *EDE* refers to entries in the file *Object-Types*.

Figure 2. The Object-Types.csv file

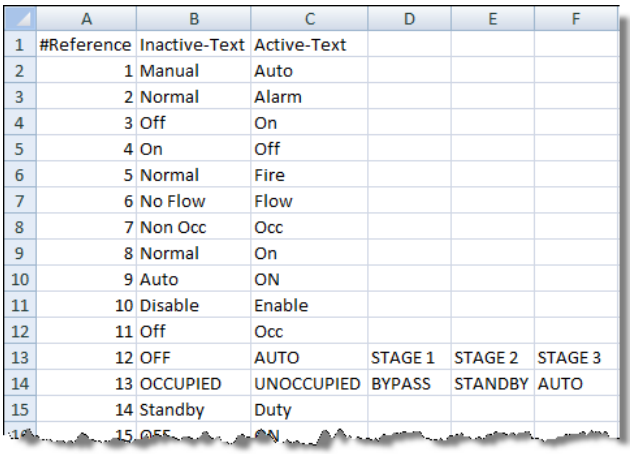


	A	B	C
1	#Encoding of BACnet Object Types		
2	#Code	Object Type	
3	0	Analog Input	
4	1	Analog Output	
5	2	Analog Value	
6	3	Binary Input	
7	4	Binary Output	
8	5	Binary Value	
9	6	Calendar	
10	7	Command	
11	8	Device	
12	9	Event-Enrollment	
13	10	File	
14	11	Group	

State-Texts

The CSV file which includes, in its name *State-Texts*, contains information about the state texts being used for binary objects and multi state objects. The file *EDE* refers to entries in the file *State-Texts*.

Figure 3. The State-Texts.csv file

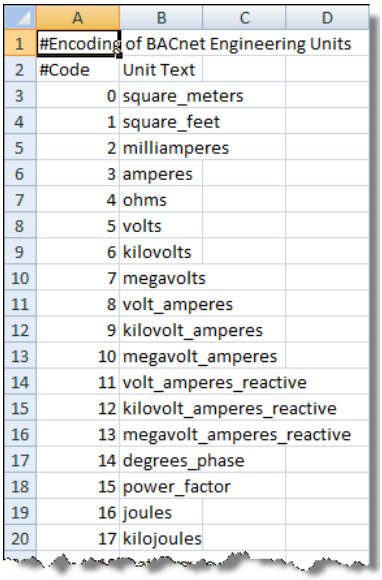


	A	B	C	D	E	F
1	#Reference	Inactive-Text	Active-Text			
2	1	Manual	Auto			
3	2	Normal	Alarm			
4	3	Off	On			
5	4	On	Off			
6	5	Normal	Fire			
7	6	No Flow	Flow			
8	7	Non Occ	Occ			
9	8	Normal	On			
10	9	Auto	ON			
11	10	Disable	Enable			
12	11	Off	Occ			
13	12	OFF	AUTO	STAGE 1	STAGE 2	STAGE 3
14	13	OCCUPIED	UNOCCUPIED	BYPASS	STANDBY	AUTO
15	14	Standby	Duty			
16	15	OFF	ON			

Unit-Texts

The CSV file which includes, in its name *Unit-Texts*, contains a list of supported *BACnetEngineeringUnits* which are being used for analog objects. The file *EDE* refers to entries in the file *Unit-Texts*.

Figure 4. The Unit-Texts.csv file



	A	B	C	D
1	#Encoding	of BACnet Engineering Units		
2	#Code	Unit Text		
3	0	square_meters		
4	1	square_feet		
5	2	milliamperes		
6	3	amperes		
7	4	ohms		
8	5	volts		
9	6	kilovolts		
10	7	megavolts		
11	8	volt_amperes		
12	9	kilovolt_amperes		
13	10	megavolt_amperes		
14	11	volt_amperes_reactive		
15	12	kilovolt_amperes_reactive		
16	13	megavolt_amperes_reactive		
17	14	degrees_phase		
18	15	power_factor		
19	16	joules		
20	17	kilojoules		

EDE File Set example

The four EDE files will be named in a similar fashion to those in the following `MyProject...` example:

- `MyProject_EDE.csv`
- `MyProject_ObjTypes.csv`
- `MyProject_StateTexts.csv`
- `MyProject_Units.csv`

EDE Object Types In NiagaraAX

The EDE definition document states that those Object Types defined in the `MyProject_ObjTypes.csv` file that have a code number which corresponds with the enumerated value of an object type already defined in the BACnet specification must remain consistent with the BACnet specified type for that code number. Here is a description of the supported and unsupported BACnet object types.

The EDE definition document does state that new Object Types could conceivably be added to the Object-Types CSV file with code values outside of the enumerated range of values reserved by the BACnet specification.

NOTE: Object Type Values that are outside of the BACnet specification will be ignored because it is not possible to determine what appropriate NiagaraAX object type should be used to represent the new Object Type given only an Object Type name.

EDE File (*Object-Types.csv*) references

NOTE: When any of the EDE files are parsed, the *#Comment* row is ignored. Instead, the column number relative to the start of each row is used by the parser. Column 0 is the first column of each row followed by Column 1, Column 2 etc.

EDE File (*Object-Types.csv*)		NiagaraAX		
Column 0	**Column 1**			
Code	**BACnet Object Type**	**Proxy Point Type**	**Property ID**	**Write**
0	Analog Input	Numeric Point	Present Value	readonly
1	Analog Output	Numeric Writeable	Present Value	Writable
2	Analog Value	Numeric Point	Present Value	readonly
3	Binary Input	Boolean Point	Present Value	readonly
4	Binary Output	Boolean Writeable	Present Value	Writable
5	Binary Value	Boolean Point	Present Value	readonly
6	Calendar	Unsupported		
7	Command	Enum Point	Present Value	readonly
8	Device	Device		
9	Event-Enrollment	Enum Point	Event-Enrollment	readonly
10	File	Unsupported		
11	Group	Unsupported		
12	Loop	Unsupported		
13	Multistate Input	Enum Point	Present Value	readonly
14	Multistate Output	Enum Writeable	Present Value	Writable
15	Notification Class	Unsupported		
16	Program	Unsupported		
17	Schedule	Unsupported		
18	Averaging	Numeric Point	Average Value	readonly
19	Multistate Value	Enum Point	Present Value	readonly
20	Trend Log	History Import		
21	Life Safety Point	Enum Point	Present Value	readonly
22	Life Safety Zone	Enum Point	Present Value	readonly
23	Accumulator	Enum Point	Present Value	readonly
24	Pulse Converter	Unsupported		

EDE Device Discovery In NiagaraAX

The EDE files contain data pertaining to the BACnet Device object. Here is a description of the relevant EDE file data which is used to define the NiagaraAX Bacnet device.

EDE File (*EDE.csv*) references

NOTE: When any of the EDE files are parsed, the *#Comment* row is ignored. Instead, the column number relative to the start of each row is used by the parser. Column 0 is the first column of each row followed by Column 1, Column 2 etc.

EDE File (*EDE.csv*)		NiagaraAX
Column*	**#Comment**	**Device Property**
0 (M)	keyname	Bacnet Device Display Name
3 (M)	object-type	(object-type = 8 which determines the BACnet object type = Device)
4 (M)	object-instance	Object Id

NOTE: *The EDE definition document indicates that some fields are *Mandatory* and some are *Optional*. These are indicated (M) or (O)

EDE Point Discovery in NiagaraAX

The EDE files contain data pertaining to the BACnet Device object. Here is a description of the EDE file data which is used to define the NiagaraAX Bacnet point.

EDE File (*EDE.csv*) references

NOTE: When any of the EDE files are parsed, the *#Comment* row is ignored. Instead, the column number relative to the start of each row is used by the parser. Column 0 is the first column of each row followed by Column 1, Column 2 etc.

EDE File (*EDE.csv*)		NiagaraAX
Column*	**#Comment**	**Point / Proxy Ext Property**
2 (M)	object-name	Display Name
3 (M)	object-type	Point Type: As specified by the BACnet specification
4 (M)	object-instance	Object Id
5 (O)	description	Facets key: description
6 (O)	present-value-default	Facets key: presentValueDefault
7 (O)	min-present-value	Facets key: min
8 (O)	max-present-value	Facets key: max
9 (O)	settable	Point type: Writable
10 (O)	supports COV	Facets key: cov
11 (O)	hi-limit	Facets key: hiLimit
12 (O)	low-limit	Facets key: lowLimit

EDE File (*EDE.csv*)		NiagaraAX
Column*	**#Comment**	**Point / Proxy Ext Property**
13 (O)	state-text reference (reference: State-Texts.csv)	Facets key: stateTextRange [EnumRange]
14 (O)	unit-code (reference: Unit-Texts.csv)	Facets key: units [Unit]
15 (O)	vendor-specific-address	Unsupported

NOTE: *The EDE definition document indicates that some fields are *Mandatory* and some are *Optional*. These are indicated (M) or (O)

Troubleshooting

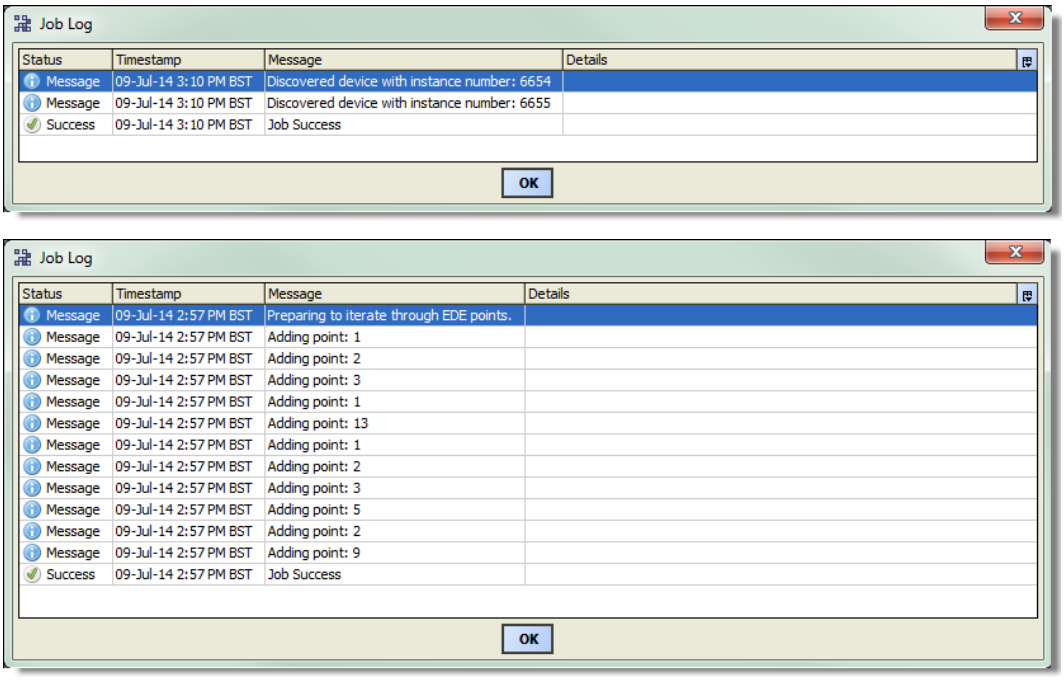
Here is a collection of Troubleshooting topics:

Job Log

When an EDE discover is initiated, the NiagaraAX **JobService** records a **Job Log**. This can be inspected if necessary and used in troubleshooting.

You can click on the **Job Log** control near the top of the manager to see the progress of the EDE discover.

Figure 5. The Job Log shows successful device and point discover

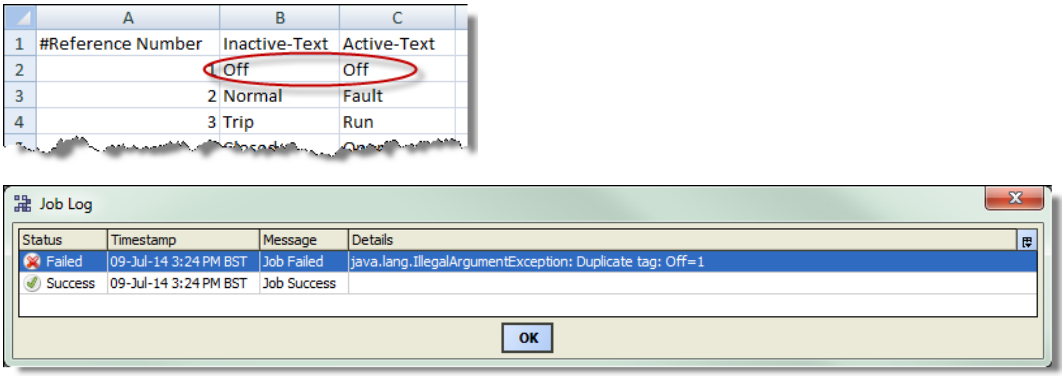


State-Texts Error

If the EDE *State-Texts* file is in error, the EDE discover will fail.

In this example, the EDE *State-Texts* file contains duplicated texts and the **Job Log** `Status` has indicated `Failed`.

Figure 6. A duplicated text and Job Log error



NOTE: The EDE discover will fail to complete successfully and you will need to correct the error in the EDE *State-Texts* file before continuing.

Stack:Unresolved device address

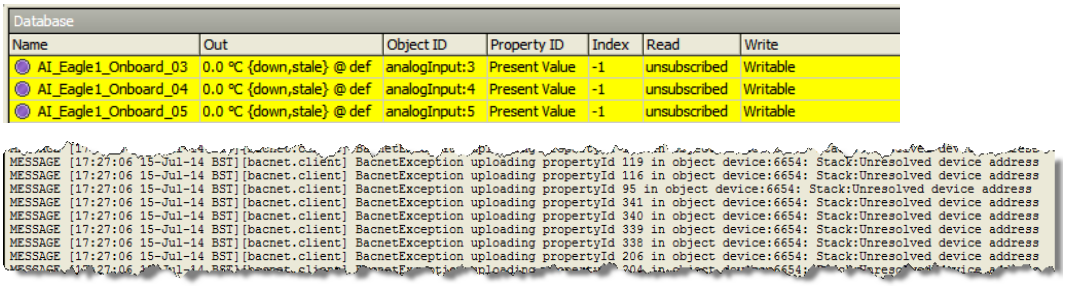
Using offline discover to add Bacnet proxy points into the driver database whilst the BacnetDevice is offline will create a number of `BacnetException:Stack:Unresolved device address` debug messages in the **Application Director** view. The `status flag type` and `status flag colours` will indicate the condition of the proxy point.

In these examples, offline discover has taken place to add Bacnet proxy points into the driver database. The BacnetDevice is initially offline and then afterwards brought into an online condition. Inspection of the **Application Director** view shows a number of messages which occur during the discover while the BacnetDevice is offline.

NOTE: The behaviour of the Bacnet proxy point `status flag type` and `status flag colours` are different between the Niagara AX 3.7.x and 3.8.x releases.

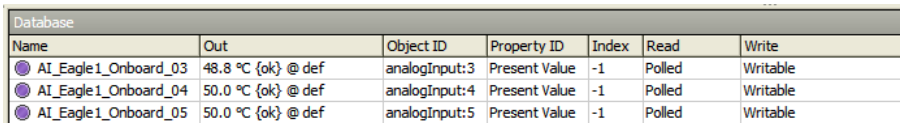
Proxy points added while the BacnetDevice is initially offline at Niagara AX 3.7.x

Figure 7. BacnetDevice offline



NOTE: The Bacnet proxy points are shown `down` indicating driver communications to the parent BacnetDevice are currently lost, based upon the device status (Monitor) configuration for that network.

Figure 8. BacnetDevice online



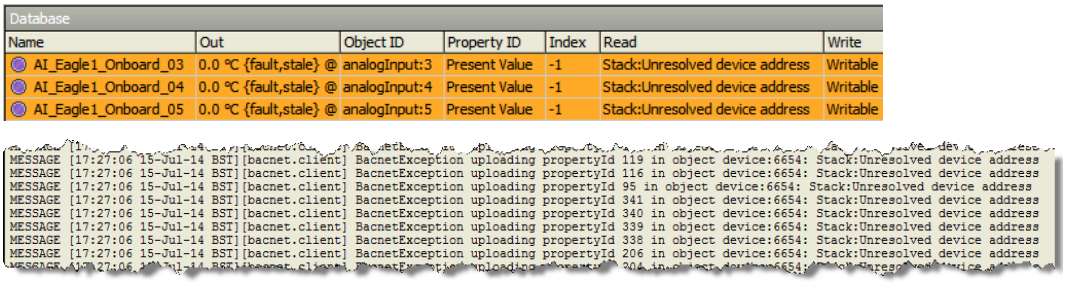
Database

Name	Out	Object ID	Property ID	Index	Read	Write
AI_Eagle1_Onboard_03	48.8 °C {ok} @ def	analogInput:3	Present Value	-1	Polled	Writable
AI_Eagle1_Onboard_04	50.0 °C {ok} @ def	analogInput:4	Present Value	-1	Polled	Writable
AI_Eagle1_Onboard_05	50.0 °C {ok} @ def	analogInput:5	Present Value	-1	Polled	Writable

NOTE: When driver communications to the parent BacnetDevice are subsequently established, based upon the device status (Monitor) configuration for that network, the Bacnet proxy points are shown `ok`

Proxy points added while the BacnetDevice is initially offline at Niagara AX 3.8.x

Figure 9. BacnetDevice offline



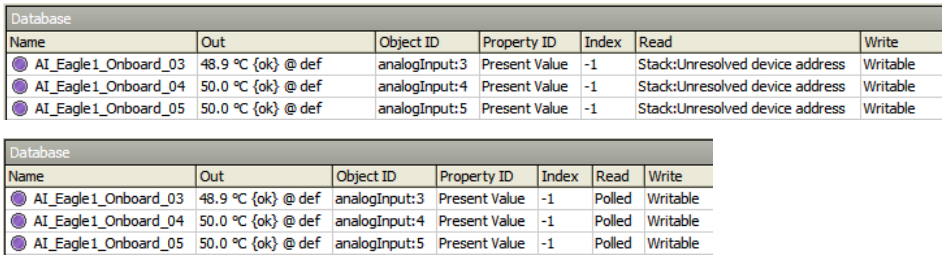
Database

Name	Out	Object ID	Property ID	Index	Read	Write
AI_Eagle1_Onboard_03	0.0 °C {fault,stale} @	analogInput:3	Present Value	-1	Stack:Unresolved device address	Writable
AI_Eagle1_Onboard_04	0.0 °C {fault,stale} @	analogInput:4	Present Value	-1	Stack:Unresolved device address	Writable
AI_Eagle1_Onboard_05	0.0 °C {fault,stale} @	analogInput:5	Present Value	-1	Stack:Unresolved device address	Writable

```
MESSAGE [17:27:06 15-Jul-14 BST][bacnet.client] BacnetException uploading propertyId 119 in object device:6654: Stack:Unresolved device address
MESSAGE [17:27:06 15-Jul-14 BST][bacnet.client] BacnetException uploading propertyId 116 in object device:6654: Stack:Unresolved device address
MESSAGE [17:27:06 15-Jul-14 BST][bacnet.client] BacnetException uploading propertyId 95 in object device:6654: Stack:Unresolved device address
MESSAGE [17:27:06 15-Jul-14 BST][bacnet.client] BacnetException uploading propertyId 341 in object device:6654: Stack:Unresolved device address
MESSAGE [17:27:06 15-Jul-14 BST][bacnet.client] BacnetException uploading propertyId 340 in object device:6654: Stack:Unresolved device address
MESSAGE [17:27:06 15-Jul-14 BST][bacnet.client] BacnetException uploading propertyId 339 in object device:6654: Stack:Unresolved device address
MESSAGE [17:27:06 15-Jul-14 BST][bacnet.client] BacnetException uploading propertyId 338 in object device:6654: Stack:Unresolved device address
MESSAGE [17:27:06 15-Jul-14 BST][bacnet.client] BacnetException uploading propertyId 206 in object device:6654: Stack:Unresolved device address
```

NOTE: The Bacnet proxy points are shown `fault` which typically indicates a configuration error or a “native fault” in the parent BacnetDevice.

Figure 10. BacnetDevice online



Database

Name	Out	Object ID	Property ID	Index	Read	Write
AI_Eagle1_Onboard_03	48.9 °C {ok} @ def	analogInput:3	Present Value	-1	Stack:Unresolved device address	Writable
AI_Eagle1_Onboard_04	50.0 °C {ok} @ def	analogInput:4	Present Value	-1	Stack:Unresolved device address	Writable
AI_Eagle1_Onboard_05	50.0 °C {ok} @ def	analogInput:5	Present Value	-1	Stack:Unresolved device address	Writable

Database

Name	Out	Object ID	Property ID	Index	Read	Write
AI_Eagle1_Onboard_03	48.9 °C {ok} @ def	analogInput:3	Present Value	-1	Polled	Writable
AI_Eagle1_Onboard_04	50.0 °C {ok} @ def	analogInput:4	Present Value	-1	Polled	Writable
AI_Eagle1_Onboard_05	50.0 °C {ok} @ def	analogInput:5	Present Value	-1	Polled	Writable

NOTE: When driver communications to the parent BacnetDevice are subsequently established, based upon the device status (Monitor) configuration for that network, the Bacnet proxy points are shown `ok`. Note also that the `Read: Stack:Unresolved device address` is updated when the database view is Refreshed.

Online Discover following Offline Discover

Bacnet Devices and Bacnet proxy points may be discovered from online devices and added to the driver database after the driver database has been initially discovered offline from an EDE Configuration.

NOTE: If you discover online and overwrite proxy points which have been added from offline discover from an EDE Configuration you are likely to break any Object or Graphical `Links` which you may have previously made.

Related Documentation

These documents provide more information for understanding how the NiagaraAX implementation of the BACnet EDE Workbench plugin works.

- ***NiagaraAX BACnet Guide***
- ***NiagaraAX Drivers Guide***

Document change log

This log identifies updates made to this document.

- Initial publication: July 18th, 2014