

Technical Document

Niagara 4 BACnet Driver Guide

May 31, 2023

niagara⁴

Niagara 4 BACnet Driver Guide

Tridium, Inc.
3951 Westerre Parkway, Suite 350
Richmond, Virginia 23233
U.S.A.

Confidentiality

The information contained in this document is confidential information of Tridium, Inc., a Delaware corporation ("Tridium"). Such information and the software described herein, is furnished under a license agreement and may be used only in accordance with that agreement.

The information contained in this document is provided solely for use by Tridium employees, licensees, and system owners; and, except as permitted under the below copyright notice, is not to be released to, or reproduced for, anyone else.

While every effort has been made to assure the accuracy of this document, Tridium is not responsible for damages of any kind, including without limitation consequential damages, arising from the application of the information contained herein. Information and specifications published here are current as of the date of this publication and are subject to change without notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia.

Trademark notice

BACnet and ASHRAE are registered trademarks of American Society of Heating, Refrigerating and Air-Conditioning Engineers. Microsoft, Excel, Internet Explorer, Windows, Windows Vista, Windows Server, and SQL Server are registered trademarks of Microsoft Corporation. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Mozilla and Firefox are trademarks of the Mozilla Foundation. Echelon, LON, LonMark, LonTalk, and LonWorks are registered trademarks of Echelon Corporation. Tridium, JACE, Niagara Framework, and Sedona Framework are registered trademarks, and Workbench are trademarks of Tridium Inc. All other product names and services mentioned in this publication that are known to be trademarks, registered trademarks, or service marks are the property of their respective owners.

Copyright and patent notice

This document may be copied by parties who are authorized to distribute Tridium products in connection with distribution of 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.

Copyright © 2023 Tridium, Inc. All rights reserved.

The product(s) described herein may be covered by one or more U.S. or foreign patents of Tridium.

Contents

| | |
|----------------------------------------------------------------------------------------|-----------|
| About this guide | 11 |
| Document change log | 11 |
| Related documentation | 12 |
| Chapter 1 Installation | 13 |
| Architecture | 13 |
| Prerequisites | 14 |
| Limits imposed by the BACnet licenses | 14 |
| Installing Workbench | 15 |
| AWS and OWS Supervisors | 16 |
| Requirements | 16 |
| Palette differences | 16 |
| Chapter 2 Network setup | 19 |
| Adding a BACnet network in a Supervisor station | 20 |
| Adding a BACnet network in a controller station | 24 |
| Configuring a network with an Ethernet port | 25 |
| Configuring a network with an MS/TP port | 26 |
| Setting up a controller under the Supervisor's network in the Supervisor station | 27 |
| Chapter 3 Controller device and point configuration | 29 |
| Adding a device using discovery | 29 |
| Matching a device | 30 |
| Adding a device using the New wizard | 30 |
| Adding a device from the palette | 31 |
| Configuring device IDs | 31 |
| Adding proxy points | 33 |
| Expanding polling to report more data | 34 |
| Adding a slot to poll for more data | 35 |
| Alarm reception configuration | 37 |
| Configuring a BACnet device to send alarms | 39 |
| Configuring a station to receive alarms | 39 |
| Alarm reporting configuration | 40 |
| Configuring the AlarmService for BACnet alarms | 40 |
| Exporting BACnet objects that generate alarms | 42 |
| Testing BACnet alarm generation | 42 |
| Backing up a station | 43 |
| Chapter 4 Offline device and point configuration | 45 |
| EDE device and point configuration files | 46 |
| EDE state texts support file | 48 |
| Configuring a station for EDE | 49 |
| Exporting controller device and point data | 49 |
| Validation criteria for data to be discovered | 50 |
| Discovering device data offline | 51 |

| | |
|---------------------------------------------------------|-----------|
| Discovering point data offline | 52 |
| EDE troubleshooting | 53 |
| Stack: Unresolved device address | 53 |
| Proxy point overwrite | 55 |
| Chapter 5 Device management..... | 57 |
| Searching for a specific device | 57 |
| Adding a time synchronization recipient | 58 |
| Synchronizing time across the network..... | 58 |
| Data gathering from devices..... | 59 |
| Improving performance by adding a tuning policy | 59 |
| Subscribing proxy points for COV data collection | 60 |
| Collecting device history | 61 |
| Performance improvements with virtual components..... | 63 |
| Configuring virtual device properties | 64 |
| Configuring virtual points in Px views | 66 |
| Importing a schedule from a remote station | 68 |
| Chapter 6 Server management | 69 |
| Dynamic creation of objects..... | 69 |
| Enabling the Object Handler..... | 70 |
| Creating object requests with initial values | 70 |
| Deleting a dynamically-created object..... | 74 |
| Types of objects to export | 74 |
| Discovering components to export as BACnet objects..... | 75 |
| Setting up a point export descriptor | 76 |
| Exporting a schedule..... | 78 |
| Exporting a file..... | 79 |
| History export..... | 81 |
| Exporting standard history | 82 |
| Configuring exported objects for write | 82 |
| Chapter 7 Internetworks and BACnet/IP | 85 |
| Example internetwork diagrams..... | 85 |
| Example BACnet/Ip diagrams | 88 |
| Internetwork rules | 89 |
| BACnet/IP and BBMDs | 90 |
| Installing a host on an IP subnet with a BBMD | 91 |
| Configuring a station as a BBMD..... | 91 |
| Configuring a station as a foreign device | 93 |
| New BACnet IP network BBMD configuration..... | 94 |
| Chapter 8 BACnet/SC (Secure Connect)..... | 95 |
| BACnet/SC and BACnet/IP | 95 |
| Message handling | 96 |
| Node certificates..... | 98 |
| Creating a site CA certificate | 100 |
| Creating operational certificates | 103 |

| | |
|-------------------------------------------------------------------------|------------|
| Creating a CSR for an operational certificate | 105 |
| Signing multiple certificates..... | 105 |
| Importing the signed certificates..... | 107 |
| Setting up a secure hub | 107 |
| Adding a hub function to a Secure Connect port..... | 111 |
| Configuring the WebService | 112 |
| Enabling a secure hub connection | 113 |
| Setting up a failover hub..... | 113 |
| Creating a direct node connection | 114 |
| Setting up a secure port | 115 |
| FAQ..... | 117 |
| Chapter 9 Troubleshooting..... | 119 |
| Manually adding a skipUploadslot | 120 |
| Rebooting a host..... | 120 |
| Chapter 10 Components in the bacnet module | 123 |
| BACnet EDE service (bacnetEDE-WbEdeService) | 123 |
| BacnetNetwork (bacnet-BacnetNetwork) | 124 |
| Local Device (bacnet-LocalBacnetDevice) | 125 |
| Bacnet Comm (bacnet-BacnetStack)..... | 133 |
| Tuning Policies (bacnet-BacnetTuningPolicyMap)..... | 165 |
| Bacnet Tuning Policy (Default Policy) (bacnet-BacnetTuningPolicy) | 165 |
| BacnetDeviceFolder | 168 |
| BacnetDevice (bacnet-BacnetDevice) | 168 |
| BacnetPointFolder (bacnet-BacnetPointFolder) | 170 |
| Virtual (bacnet-BacnetVirtualGateway) | 171 |
| BacnetVirtualObject (bacnet-BacnetVirtualObject)..... | 171 |
| BacnetVirtualProperty (bacnet-BacnetVirtualProperty)..... | 171 |
| Alarms (bacnet-BacnetAlarmDeviceExt) | 171 |
| Schedules (bacnet-BacnetScheduleDeviceExt)..... | 172 |
| Trend Logs (bacnet-BacnetHistoryDeviceExt) | 173 |
| Config container (bacnet-BacnetConfigDeviceExt)..... | 174 |
| Device Object (bacnet-BacnetDeviceObject) | 174 |
| BACnet array (bacnet-BacnetArray)..... | 176 |
| Bacnet Destination (bacnet-BacnetDestination) | 178 |
| Points | 179 |
| Points (bacnet-BacnetPointDeviceExt)..... | 179 |
| Proxy Ext (bacnet-BacnetBooleanProxyExt)..... | 179 |
| Proxy Ext (bacnet-BacnetNumericProxyExt) | 182 |
| Proxy Ext (bacnet-BacnetEnumProxyExt)..... | 185 |
| Proxy Ext (bacnet-BacnetStringProxyExt) | 189 |
| Out Of Service Extension (bacnet-OutOfServiceExt) | 192 |
| Schedule..... | 193 |
| Bacnet Schedule Import Ext (bacnet-BacnetScheduleImportExt) | 193 |

| | |
|-------------------------------------------------------------------------------------------|-----|
| Bacnet Schedule Export (bacnet-BacnetScheduleExport) | 194 |
| Trending | 197 |
| Bacnet History Import (bacnet-BacnetHistoryImport)..... | 197 |
| Bacnet Trend Log Multiple Import (bacnet- BacnetTrendLogMultipleImport) | 199 |
| Trending, Bacnet Numeric Cov Trend Log Ext (bacnet- BacnetNumericCovTrendLogExt)..... | 202 |
| Bacnet Numeric Interval Trend Log Ext (bacnet- BacnetNumericIntervalTrendLogExt) | 203 |
| Bacnet Numeric Trend Log Ext (bacnet- BacnetNumericTrendLogExt)..... | 205 |
| Bacnet Numeric Trend Log Remote Ext (bacnet- BacnetNumericTrendLogRemoteExt) | 207 |
| Bacnet Boolean Cov Trend Log Ext (bacnet- BacnetBooleanCovTrendLogExt) | 209 |
| Bacnet Boolean Interval Trend Log Ext (bacnet- BacnetBooleanIntervalTrendLogExt) | 210 |
| Bacnet Boolean Trend Log Ext (bacnet- BacnetBooleanTrendLogExt) | 212 |
| Bacnet Boolean Trend Log Ext (bacnet- BacnetBooleanTrendRemoteLogExt)..... | 213 |
| Bacnet Enum Cov Trend Log Ext (bacnet- BacnetEnumCovTrendLogExt)..... | 215 |
| Bacnet Enum Interval Trend Log Ext (bacnet- BacnetEnumIntervalTrendLogExt)..... | 216 |
| Bacnet Enum Trend Log Ext (bacnet- BacnetEnumTrendLogExt) | 218 |
| Bacnet Enum Trend Log Remote Ext (bacnet- BacnetEnumTrendLogRemoteExt) | 219 |
| Bacnet String Cov Trend Log Ext (bacnet- BacnetStringCovTrendLogExt)..... | 221 |
| Bacnet String Interval Trend Log Ext (bacnet- BacnetStringIntervalTrendLogExt) | 223 |
| Bacnet String Trend Log Ext (bacnet- BacnetStringTrendLogExt)..... | 224 |
| Bacnet String Trend Log Remote Ext (bacnet- BacnetStringTrendLogRemoteExt)..... | 226 |
| Bacnet Trend Log Alarm Source Ext (bacnet- BacnetTrendLogAlarmSourceExt) | 228 |
| Bacnet Bit String Trend Log Ext (bacnet- BacnetBitStringTrendLogExt) | 230 |
| config | 232 |
| ConfigFolder..... | 232 |
| BacnetObject | 232 |
| Config, AI (Bacnet Analog Input)..... | 232 |
| Config, AO (Bacnet Analog Output) | 234 |
| Config, AV (Bacnet Analog Value)..... | 236 |
| Config, BI (Bacnet Binary Input)..... | 237 |
| Config, BO (Bacnet Binary Output)..... | 239 |
| Config, BV (Bacnet Binary Value) | 241 |

| | |
|---------------------------------------------------------------------------------------|-----|
| Config, Cal (Bacnet Calendar) | 243 |
| Config, EE (Bacnet Event Enrollment)..... | 245 |
| EE (Event Parameter)..... | 247 |
| Config, File (Bacnet File) | 247 |
| Config, Grp (Bacnet Group) | 247 |
| Config, Loop (Bacnet Loop) | 249 |
| Config, MSI (Bacnet Multistate Input)..... | 251 |
| Config, MSO (Bacnet Multistate Output)..... | 253 |
| Config, MSV (Bacnet Multistate Value)..... | 255 |
| Config, NC (Bacnet Notification Class) | 256 |
| Config, Prog (Bacnet Program)..... | 258 |
| Config, Sched (Bacnet Schedule)..... | 259 |
| Config, TLog (Bacnet Trend Log)..... | 261 |
| Config, TLM (Bacnet Trend Log Multiple) | 263 |
| Server | 263 |
| Bacnet Analog Input Descriptor (ServerAIDesc) | 263 |
| Bacnet Analog Output Descriptor (ServerAODesc) | 265 |
| Bacnet Analog Value Descriptor (ServerAVDesc)..... | 267 |
| Bacnet Analog Value Prioritized Descriptor (ServerAVPDesc) | 269 |
| Bacnet Binary Input Descriptor (ServerBIDesc)..... | 271 |
| Bacnet Binary Output Descriptor (ServerBODesc)..... | 273 |
| Bacnet Binary Value Descriptor (ServerBVDesc) | 275 |
| Bacnet Binary Value Prioritized Descriptor (ServerBVPDesc) | 277 |
| Bacnet Boolean Schedule Descriptor (ServerBoolSchedDesc) | 279 |
| Bacnet Calendar Descriptor (ServerCalendarDesc)..... | 280 |
| Bacnet Enum Schedule Descriptor (ServerEnumSchedDesc)..... | 282 |
| Bacnet Event Enrollment Descriptor (bacnet- BacnetEventEnrollmentDescriptor) | 282 |
| Event Saver (Event Saver) | 284 |
| Bacnet File Descriptor (ServerFileDesc) | 285 |
| Bacnet Loop Descriptor (ServerLoopDesc) | 287 |
| Bacnet Multi State Input Descriptor (ServerMSIDesc)..... | 288 |
| Bacnet Multi State Output Descriptor (ServerMSODesc)..... | 289 |
| Bacnet Multi State Value Descriptor (ServerMSVDesc) | 291 |
| Bacnet Multi State Value Prioritized Descriptor (ServerMSVPDesc)..... | 293 |
| Bacnet Niagara History Descriptor (ServerNiagaraHistoryDesc)..... | 294 |
| Bacnet Notification Class Descriptor (ServerNCDesc) | 296 |
| Bacnet Numeric Schedule Descriptor (ServerNumSchedDesc) | 298 |
| Bacnet Dynamic Schedule Descriptor (bacnet- BacnetDynamicScheduleDescriptor)..... | 299 |

| | |
|------------------------------------------------------------------------|------------|
| Bacnet String Schedule Descriptor (ServerStringSchedDesc) | 301 |
| Bacnet Trend Log Descriptor (bacnet- BacnetTrendLogDescriptor)..... | 302 |
| Worker Pool..... | 304 |
| Worker Pool (bacnet-BacnetWorkerPool)..... | 304 |
| Chapter 11 Components in the bacnetAws module | 307 |
| BacnetAwsNetwork..... | 307 |
| Local Device..... | 309 |
| dynamicObjects folder (bacnetAws-BacnetDynamicObjects)..... | 312 |
| BacnetAwsDeviceFolder..... | 312 |
| BacnetAwsDevice..... | 313 |
| Config | 315 |
| AwsConfigFolder | 315 |
| BacnetEventLogImport | 315 |
| Elog..... | 317 |
| Door..... | 319 |
| ACC | 321 |
| CMD..... | 323 |
| LCO..... | 325 |
| PC | 327 |
| SVO..... | 329 |
| Chapter 12 Components in the bacnetOws module..... | 331 |
| BacnetOwsNetwork | 331 |
| Local Device..... | 332 |
| BacnetOwsDeviceFolder | 336 |
| Chapter 13 Components in the histories module | 337 |
| history-HistoryConfig | 337 |
| Chapter 14 Views | 339 |
| Bacnet Device Manager..... | 339 |
| EDE Bacnet Device Manager | 342 |
| Bacnet Point Manager | 344 |
| EDE Bacnet Point Manager..... | 349 |
| Bacnet Export Manager (bacnet-BacnetExportManager)..... | 351 |
| Svo Subordinate Manager view..... | 353 |
| Bacnet File Export Manager..... | 354 |
| Bacnet Niagara Log Export Manager | 356 |
| Bacnet Schedule Import Manager | 358 |
| Bacnet Schedule Export Manager | 361 |
| Bacnet History Import Manager | 363 |
| Device Histories view | 365 |
| Bacnet Trend Multiple View | 366 |
| Bacnet Config Manager..... | 366 |
| Bacnet Aws Device Manager..... | 368 |
| Bacnet Aws Config Manager..... | 370 |

| | |
|----------------------------------------------------------------|------------|
| Bdt Manager | 372 |
| Fdt Manager | 373 |
| Bacnet Ows Device Manager | 374 |
| HTML-5 Bacnet Ows Device Manager | 376 |
| HTML 5 Bacnet Device UxManager View | 378 |
| Chapter 15 Windows | 379 |
| Configure Device Discovery window | 379 |
| Device Communication Control window | 380 |
| EDE Export Window..... | 381 |
| EDE Configuration Window | 382 |
| Get Enrollment Summary window | 383 |
| New (and Edit) Bacnet Descriptors..... | 384 |
| New (and Edit) BBMD record | 386 |
| New (and Edit) device window | 386 |
| New (and Edit) device history window | 388 |
| New (and Edit) export folder window | 390 |
| Add (and Edit) export history window | 391 |
| New (and Edit) export objects window | 393 |
| New (and Edit) file export window | 395 |
| New (and Edit) foreign device record | 396 |
| New (and Edit) point window | 397 |
| New (and Edit) window for schedule import | 399 |
| New (Edit) window for schedule export..... | 401 |
| ReinitializeDevice window..... | 402 |
| Search for objects by name or id (who-has config) window | 403 |
| Synchronize Time window..... | 404 |
| Index..... | 405 |
| Glossary | 411 |

About this guide

This topic contains important information about the purpose, content, context, and intended audience for this document.

Product Documentation

This document is part of the Niagara technical documentation library. Released versions of Niagara software include a complete collection of technical information that is provided in both online help and PDF format. The information in this document is written primarily for Systems Integrators. To make the most of the information in this book, readers should have some training or previous experience with Niagara software, as well as experience working with JACE network controllers.

Document content

This document provides information about BACnet, components and plugins, license tools and other topics related to the BACnet.

CAUTION: Protect against unauthorized access by restricting physical access to the computers and devices that manage your building model. Set up user authentication with strong passwords, and secure components by controlling permissions. Failure to observe these recommended precautions could expose your network systems to unauthorized access and tampering.

Document change log

Changes to this document are listed in this topic.

May 31, 2023

Edited multiple topics to prepare the document for the 4.13 update

January 12, 2023

Updated "Bacnet Event Enrollment Descriptor (Bacnet-BacnetEventEnrollmentDescriptor)" property sheet.

Added new topic "HTML 5 Bacnet Device UxManager View" to the "plugins" chapter.

September 13, 2022

- Removed "History Name Format" property from all TrendLogExt property sheets in the BACnet module.
- Added new topic "HTML-5 Bacnet Ows Device Manager" to the "Plugins" chapter.
- Clarified a security statement in the BACnet/SC Message Handling topic.

November 9, 2021

Updated the "BACnet/SC (Secure Connect)" chapter.

May 29, 2021

- Updated image and description of tuning policy properties.
- Replaced Cov tuning policy example image.
- Created topics for Bacnet SC feature.

April 1, 2020

Created topics for EDE configuration utility and creation of dynamic objects with initial values.

October 25, 2019

In the topic, "About this guide", added a caution note alerting customers to restrict access to all computers, devices, field buses, components, etc., that manage their building model.

September 19, 2019

- Created BACnet Properties Dictionary.
- Edited multiple topics to prepare the document for the 4.9 update.

March 22, 2019

Changes for BACnet, AMEV profile AS-B updates, including the following:

- Updated properties in "Components" and "Windows" chapters.
- Added new topic "Dynamic creation of objects".
- Added new topics in "Components" chapters.
- Updated existing topics for the AMEV-AS-B updates.

November 27, 2018

Updated for functional changes in Niagara 4.6:

- Updated BACnet Virtual Component tuning policies.
- Included the StringPoint and StringWritable in the "Types of objects to export" topic.
- Updated the BACnet Status Flags status in the "Troubleshooting" topic.

August 09, 2018

Initial Release.

Related documentation

These documents contain related information.

- Multiple *Install and Startup Guides* for each remote host controller.
- *Niagara Drivers Guide*
- *Niagara Platform Guide*

Chapter 1 Installation

Topics covered in this chapter

- ◆ Architecture
- ◆ Prerequisites
- ◆ Limits imposed by the BACnet licenses
- ◆ Installing Workbench
- ◆ AWS and OWS Supervisors

This chapter documents how to install the BACnet driver to run in both a Supervisor (client) and remote station (server).

To use this driver, you must have a target host (remote controller) that is licensed for the feature **bacnet**, or a PC host, which acts as a BACnet Supervisor. The Supervisor must also be licensed for **bacnet**.

NOTE: All platforms support the link layer types BACnet/IP, BACnet/Ethernet, and BACnet/SC, however, BACnet MS/TP trunks are directly supported by QNX-based controllers. Separate licensing is required.

Modules

The palette provides four BACnet modules.

| Usage | Palette name | Network component name in the Nav tree | .jar file names in the modules folder | Comments |
|-----------------------------|--------------|----------------------------------------|----------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| BACnet functionality | bacnet | BacnetNetwork | bacnet-rt.jar, bacnetUtil-rt.jar, bacnet-ux.jar, bacnet-wb.jar | This module provides BACnet functionality to a server station running on a remote controller. |
| BACnet Advanced Workstation | bacnetAws | BacnetAwsNetwork | bacnetAws-rt.jar, bacnetAws-ux.jar, bacnetAws-wb.jar | This module provides BACnet functionality to a client station running on a PC. |
| BACnet Operator Workstation | bacnetOws | BacnetOwsNetwork | bacnetOws-rt.jar, bacnetOws-ux.jar, bacnetOws-wb.jar | This module provides BACnet functionality to a client station running on a PC. |
| BACnet Workstation | bacnetws | BacnetWsNetwork | bacnetMigrator-wb.jar | Support for this legacy module ended with NiagaraAX-3.7 and should not be used for new installations. |

Architecture

BACnet (Building Automation Control network) is a data communication protocol developed by ASHRAE (American Society of Heating, Refrigerating, and Air Conditioning Engineers) for managing building automation devices. The BACnet driver uses the standard Niagara Framework® network architecture to implement this protocol. The *Niagara Drivers Guide* documents the Framework's general approach.

Network types

A remote station's **BacnetNetwork** is unique because it simultaneously supports multiple BACnet link layer types using different BACnet communication protocols:

- BACnet/IP
- BACnet/SC
- BACnet/Ethernet
- BACnet MS/TP for host platforms that are based on QNX.

Thus, depending on the configuration, a **BacnetNetwork** may proxy BACnet devices that not only reside in different BACnet networks, but are also accessed using different physical ports on the host controller.

Client/server operations

BACnet is based on a client/server model. The client is in charge and tells the server what and how to deliver data. The server responds to client requests. A remote host platform may function as both a client and a server.

Server operations

If the host platform is licensed as a BACnet server, it can export any number of selected types of objects from the station (regardless of location) to appear as BACnet objects in a client. As such, these objects can service client requests from any networked BACnet device. This includes exporting schedules, calendars and histories. Exported histories appear externally as BACnet Trend Log objects.

NOTE:

BACnet AWS Supervisor (**BacnetAwsNetwork**) and BACnet OWS Supervisor (**BacnetOwsNetwork**) stations are not typically licensed for server side (export) operations.

The BACnet driver, running in a remote controller, is capable of managing both server and client operations concurrently. In special cases, you may configure a remote controller station to run as a BACnet server only.

Prerequisites

To use the BACnet drivers, your installation needs to meet specific requirements.

- A Niagara 4 license for the **bacnet** feature for each host including an optional Supervisor station running on a PC.
BACnet device and proxy point limits may exist in your license.
- One or more 64-bit computers to serve as Supervisors; the latest version of Niagara requires 64-bit computers.
- A target host controller running the Niagara 4 Framework or later.
- Workbench running on a PC and enabled to be used as an installation tool.
- A network of installed BACnet devices.

In addition, to use the EDE (Engineering Data Exchange) utility to discover BACnet devices and points offline:

- A license that includes **bacnetEde**.
- EDE configuration files that are compatible with the EDE 2.2 specification, stored in the local file system and accessible to Workbench.

Limits imposed by the BACnet licenses

As with other drivers, the **bacnet** feature of the license may define operational limits.

Quantity limits

The license for a single BACnet network across all BACnet devices and related child components in the station regardless of the BacnetComm network port used to reach any device (**Ip PortScPort**, **EthernetPort**, **MstpPort**) includes these maximum limits:

- number of devices
- number of proxy points
- number of histories
- number of imported schedules and calendars

- number of imported trend logs

Limits do not apply to the server operations of the BACnet driver, and any license entry value of `none`, indicates that no set limits apply for that item type.

Server limits

The export property in the `bacnet` license determines whether a station can provide BACnet server functions. When enabled (`true`), components, files, or histories are exported to BACnet.

NOTE: Server operation should be enabled for any controller ordered with the Enterprise Connectivity Pack, which supports communications to other controllers and/or a Supervisor.

If the export property is disabled (`false`), no components, files, or histories are exported to BACnet apart from the single BACnet device object that represents the station (as configured by the BACnet Local Device). Although you can still use the various views of the local device's **Export Table** to add BACnet export descriptors (and file and history descriptors), all descriptors report a fault status, with a fault cause of `Server capability not licensed`.

Bacnet MS/TP limits

Bacnet MS/TP is a separately-licensed feature, requiring a feature named `mstp` in the remote controller's license file.

For example:

```
<feature name="mstp"
  expiration="never"
  port.limit="5"/>
```

`port.limit` defines the number of MS/TP trunks (RS-485 ports) that can be used. This ranges from 1 to 6, varies based on the type of the host controller, and depends the license purchased.

Due to electrical considerations, the EIA-485 (or RS-485) load factor of connected MS/TP devices determines how many devices a trunk can physically support. This ranges from 31 (full load) to up to 127 (quarter load) devices.

Other device or platform limits in the license's `bacnet` feature also apply.

Installing Workbench

The BACnet drivers (`bacnet`, `bacnetAws` and `bacnetOws`) are available through Workbench palettes. No special driver installation is required. This procedure summarizes how to install the Niagara Framework and Workbench.

Prerequisites: Your network includes a Supervisor PC.

Step 1 Download the Niagara software image ZIP file to any folder on your hard drive.

Step 2 Extract all files from the ZIP file.

Step 3 Open Windows Explorer, navigate to the folder you just created and run the installer executable (`Installer_x64.exe`).

The installation wizard opens.

Step 4 Follow the wizard, confirming that the check box: "This instance of Workbench will be used as an installation tool" is enabled.

This option ensures that Workbench has the needed distribution files (.dist files) for commissioning various models of remote platforms. The .dist files are located under your install folder in a `sw\dist` subfolder.

When the wizard completes the installation it confirms that the software is licensed and launches Workbench.

AWS and OWS Supervisors

The driver provides two BTL-certified Supervisor stations. BTL stands for BACnet Testing Laboratories. BTL certification means that the driver has been tested by an official BACnet testing organization, and certified to comply with its own BACnet PICS (Protocol Implementation Conformance Statement).

Both the BACnet AWS (Advanced Workstation) Supervisor and BACnet OWS (Operator Workstation) Supervisor are specially-licensed Supervisor stations that run on a PC. Both provide client access to, and control of BACnet devices; both provide the supervisory functionality described in the BACnet Specification for B-AWS and B-OWS profiles; and, with the exception of being able to export to BACnet objects, both include the same functionality provided by the standard BACnet driver as described in this document.

NOTE:

Any Supervisor station that was created using the older `bacnetws` module (BacnetWsNetwork) needs to be converted using a utility that is available in both the `bacnetAws` and `bacnetOws` modules. This utility converts the saved station database (`config.bog`) to one compatible with one of the supported Supervisor modules.

Requirements

Each Supervisor requires the appropriate license and modules.

AWS requirements

An BACnet AWS Supervisor and any PC running Workbench with which to access a BACnet AWS Supervisor, requires these specific things:

- Niagara Framework® and the driver's `bacnet`, `bacnetOws`, and `bacnetAws` modules in the Framework's `modules` folder.
- A license that lists these features: `bacnet`, and `bacnetOws`, and `bacnetAws`. Without a license, the station's `BacnetAwsNetwork` remains in fault. The license may also have limits on the number of devices, points, and so forth.

OWS requirements

A BACnet OWS Supervisor and any PC running Workbench with which to access a BACnet OWS Supervisor, requires these specific things:

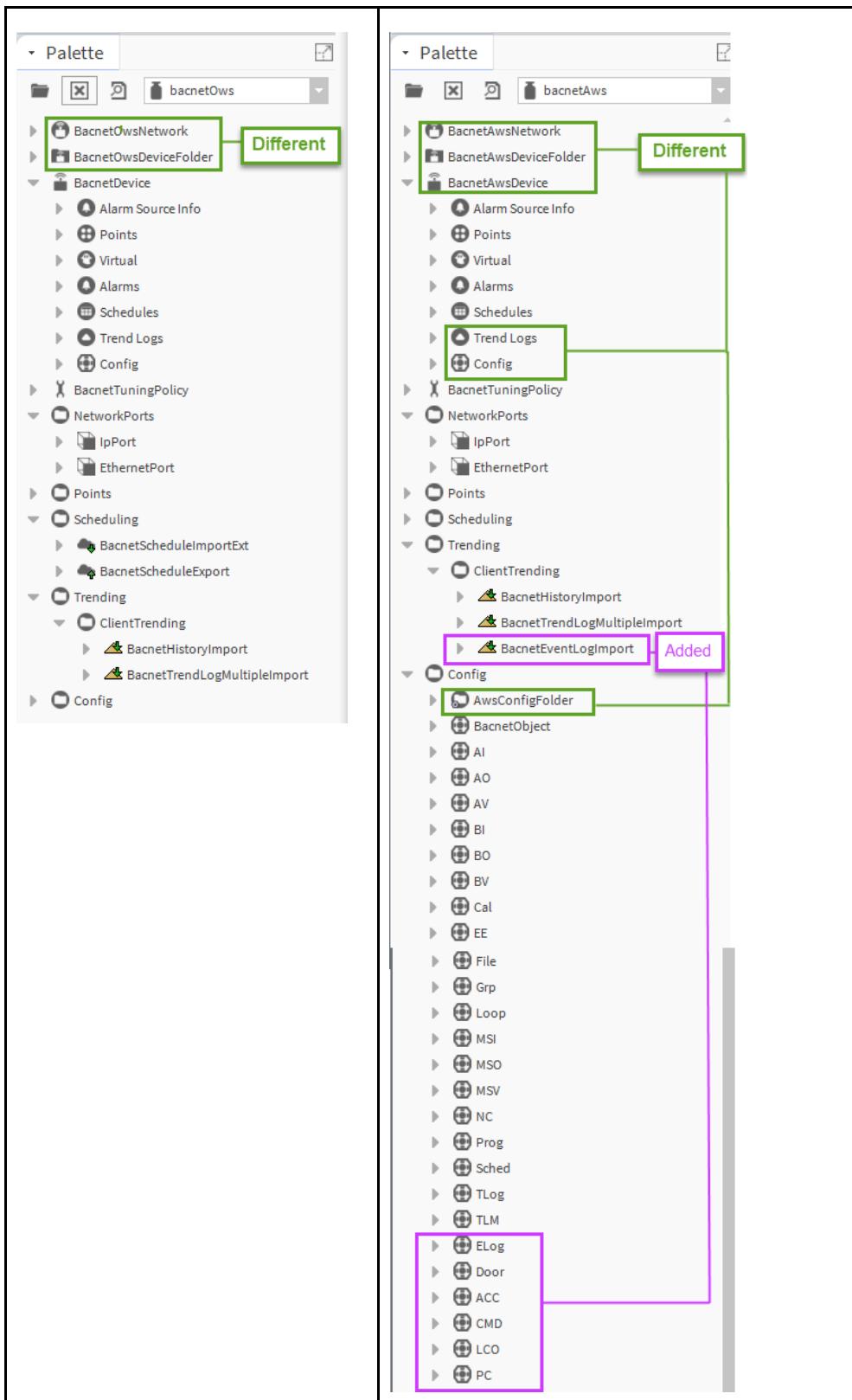
- Niagara Framework® and the driver's `bacnet`, and `bacnetOws` modules in the Framework's `modules` folder.
- A license that lists these features: `bacnet`, and `bacnetOws`. Without a license, the station's `BacnetOws-Network` remains in fault. The license may also have limits on the number of devices, points, and so forth.

NOTE: Neither Supervisor is licensed for the server-side, BACnet export of objects. This is not an issue, as a BACnet workstation's purpose is to be the ultimate "authoritative client" to all other BACnet devices.

Palette differences

Each BTL-certified BACnet Supervisor (OWS, AWS) uses much of the same software as the regular `bacnet` module. The `bacnetAWS` and `bacnetOws` modules extend the `bacnet` module providing enhanced operations.

For convenience, the palette in the `bacnetOws` and `bacnetAws` modules provide all available components, including all the shared `bacnet` components (except server-related ones).

Table 1 Palettes for bacnetOws (left) and bacnetAws (right), with differences from bacnet palette

The network component and a few others differ between the `bacnetOws` and `bacnetAws` modules, often with a different default view. For example, there is a **Bacnet Aws Device Manager** view and a **Bacnet Ows Device Manager** view. The `bacnetAws` palette provides additional components, which are not found in the `bacnet` module.

Server-side (BACnet export) components found in the `bacnet` palette are missing, as neither the BACnet AWS Supervisor nor the BACnet OWS Supervisor are licensed for server operation. Also, there is no `MsTpPort` under the **NetworkPorts** folder. (This port type is valid only for QNX-based hosts.)

The various Bacnet manager views in the palette simplify component creation, enforcing proper component hierarchy. For example, the **Bacnet Aws Config Manager** view on the **Config** device extension of a **BacnetAwsDevice** discovers the appropriate types of **Config** objects to represent BACnet objects (including added ones).

The **Bacnet Aws Config Manager** is also unique in that you can add BACnet objects and delete them from the selected device—providing that the device supports these BACnet services.

Chapter 2 Network setup

Topics covered in this chapter

- ◆ Adding a BACnet network in a Supervisor station
- ◆ Adding a BACnet network in a controller station
- ◆ Configuring a network with an Ethernet port
- ◆ Configuring a network with an MS/TP port
- ◆ Setting up a controller under the Supervisor's network in the Supervisor station

BACnet is a common and standard protocol with which you are probably very familiar. Even so, the BACnet driver contains multiple properties, which can easily become confusing.

This topic assumes that the Supervisor and controller station are using a standard Ip Port. These are the defaults. The topics that follow provide step-by-step procedures to perform each task.

Configure the BACnet network in a Supervisor station

Set up your Supervisor with a BACnet AWS or OWS network. As a minimum, configure the AWS/OWS network properties in this table.

| Property | Where to find this property (Supervisor station) | How to configure |
|-----------------|-----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Object ID | Expand the AWS/OWS network followed by expanding Local Device under the network node. | Select <code>device</code> (the Supervisor station is a BACnet device) and give the device a unique number. |
| Routing Enabled | Expand the AWS/OWS network followed by expanding Bacnet Comm and double-click Network . | Confirm that this property is set to <code>true</code> . If the Supervisor cannot connect to the controller station, it may automatically disable this property (set it to <code>false</code>), which prevents the station log from being filled with redundant messages. |
| Network Number | Expand or double-click Ip Port . | Change this from -1 (disabled) the network number. A network diagram may provide this information. |
| Adapter | Expand the AWS/OWS network followed by expanding Bacnet Comm → Network → Ip Port → Link . | Defines the Supervisor PC's network adapter or VPN connection. |
| Ip Address | Same location as Adapter . | Confirm that this address is correct. This identifies the Supervisor PC on the network. |

Set up the BACnet network in a controller station.

Add a **BacnetNetwork** to each controller station. As a minimum, configure these properties in each controller station.

| Property | Where to find this property (controller station) | How to configure |
|----------------|--------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Object ID | Expand BacnetNetwork and double-click Local Device . | Select device (the controller station is a BACnet device) and give the device a unique number. If this is a new installation, consider using the final two or three digits of the controller's IP address as its device ID. |
| Network Number | Expand BacnetNetwork→Bacnet Comm→Network→Ip Port . | Change this from -1 (disabled) the network number. A network diagram may provide this information. This should be the same network number as that for the Supervisor station. |
| Ip Address | Expand BacnetNetwork→Bacnet Comm→Network→Ip Port→Link . | Confirm that this address is correct. This identifies the controller on the network. |

If you are unable to connect from the Supervisor to a controller, use this topic as a checklist to make sure you configured each property in the Supervisor and each connector station.

Set up each controller under the Supervisor's network in the Supervisor station

Under the Supervisor's AWS or OWS network, each controller appears as a BACnet AWS or OWS device. These configurations in the Supervisor station make it possible to display points and their values from the controller stations in the Supervisor station.

You may discover the controller stations, add each controller configuration as a New AWS or OWS device, or you may drag a BacnetAwsDevice or BacnetOwsDevice from the appropriate palette to the BACnet network node in the Supervisor station.

As a minimum, if you created a new object or used the palette to create the object, configure these properties in the Supervisor station for each controller station.

| Property | Where to find this property (controller station) | How to configure |
|-----------|--------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
| Object Id | Expand the station name followed by expanding Config , then double-click Device Object . | This is the device ID assigned when you set up the controller station. |

When you define the **Object Id** the Supervisor makes the connection and populates the rest of the **Device Object** properties with information from the controller.

Additional options

If needed, in addition to an **Ip Port**, you can add one **EthernetPort** (per adapter), one or more **ScPorts**, and/or up to four **MstpPorts**. You can also add additional **Ip Ports** to enable the communications on additional UDP ports. In this case, the station acts as a BACnet/IP router between two logical BACnet/IP networks (that actually exist on the same physical network).

Adding a BACnet network in a Supervisor station

The driver supports only one BACnet network of any type in a Supervisor station. This procedure adds a network component from a BACnet palette and configures the minimum number of properties to establish the connection.

Prerequisites: You are using Workbench running in a PC and connected to the Supervisor station.

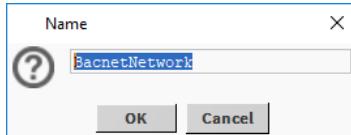
Step 1 To open the palette side bar in a new installation, click **Window→Sidebars→Palette** and open the palette for the type a Supervisor station:

- **bacnetaws** (Advanced Workstation) or
- **bacnetows** (Operator Workstation)

At this point there are several ways to add the network component. This procedure drags the network component from the palette to the station's Drivers container.

- Step 2** Expand the station's **Config** container and drag the network component (**BacnetAwsNetwork** or **BacnetOwsNetwork**) from its palette to the station's **Drivers** container.

The **Name** window opens.

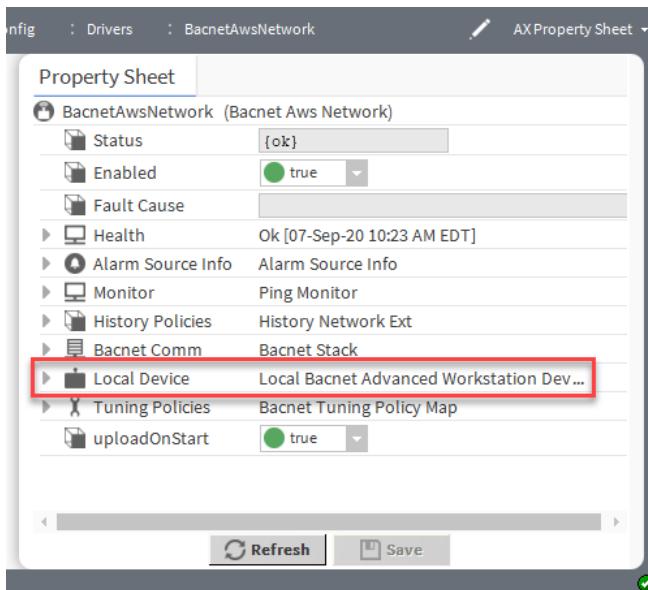


- Step 3** Rename the network, or use the default name, and click **OK**.

The added BACnet network is available under your **Drivers** container.

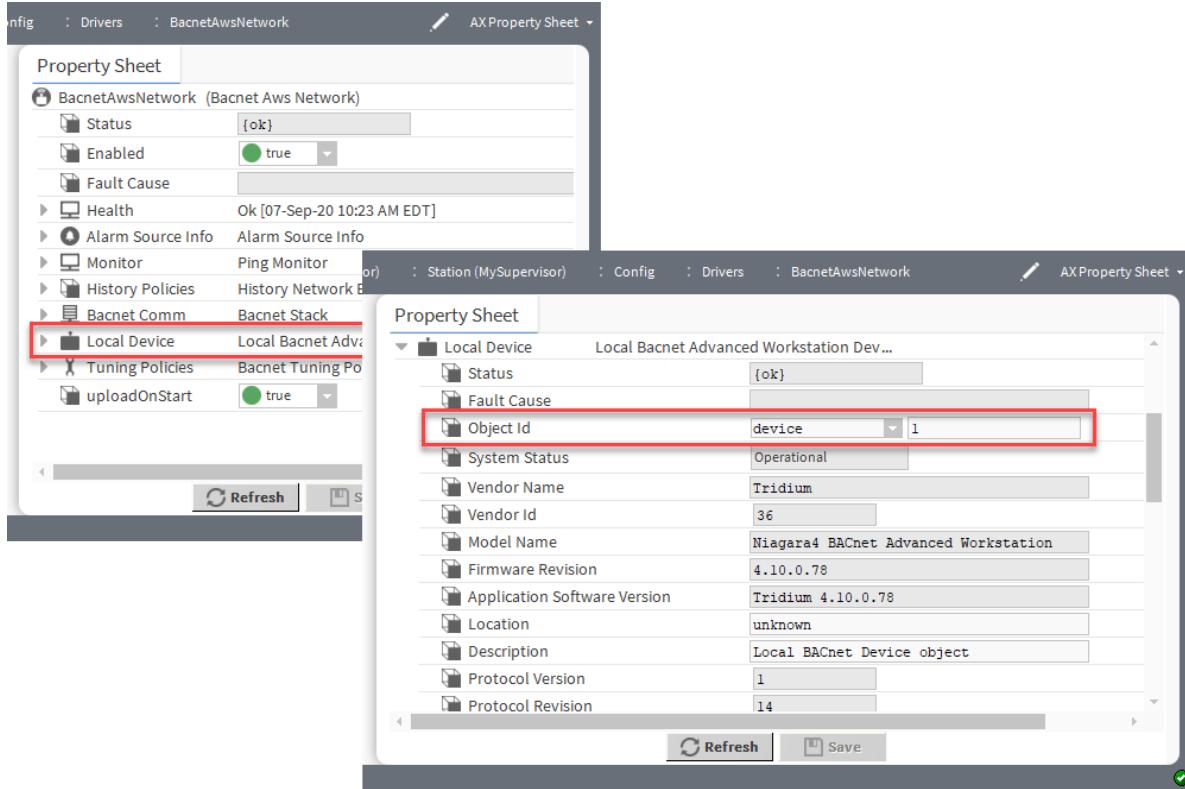
- Step 4** Right-click the BACnet network in the Nav tree and select **Views→Property Sheet**.

The **Property Sheet** opens.



- Step 5** To set up the Supervisor station as a local device, click the arrow symbol next to **Local Device** or use the Nav side bar and double-click **Local Device**.

The **Local Device** container opens.



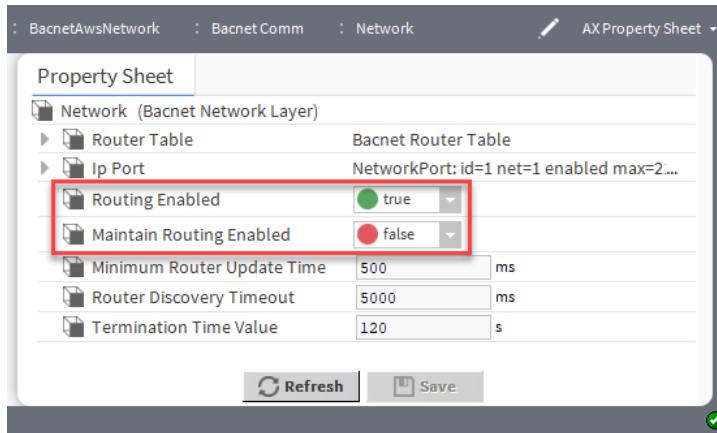
By default, a new BACnet network contains a **Bacnet Comm** (Bacnet Stack) component that supports Bacnet/IP

- Step 6** In the numerical **Object Id** property, change the entry from -1 to a valid, local device instance number, which must be unique across the entire BACnet internetwork (range is 0 to 4,194,302) and click **Save**.

NOTE: A Supervisor station is a device on a BACnet network. This instance number must be unique for this Supervisor device.

The **Local Device Status** should change from `{fault}` to `{ok}`, and the **Fault Cause** should change from `Invalid Object ID` to blank.

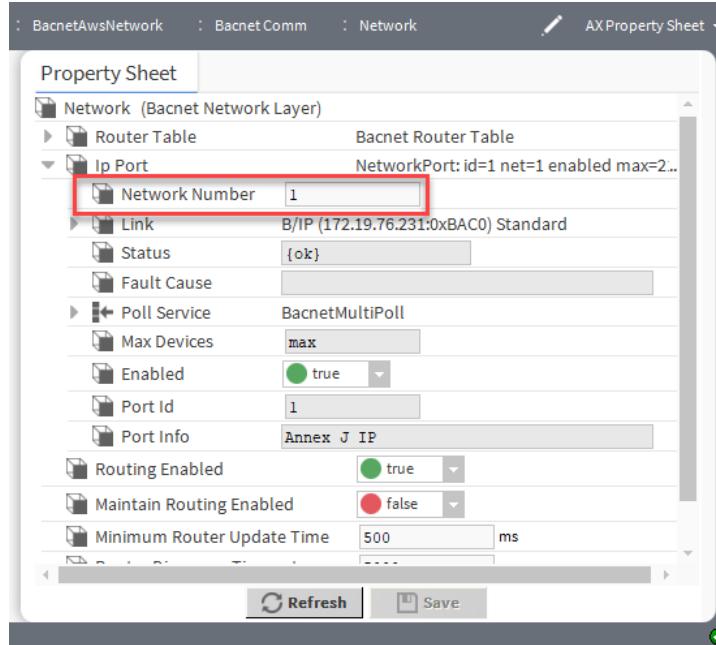
- Step 7** Expand **Bacnet Comm**, double-click **Network** and confirm that **Routing Enabled** is set to true. The **Network** (Bacnet Network Layer) properties become available.



If the Supervisor cannot connect to the controller station, it may automatically disable this property (set it to `false`), which prevents the station log from being filled with redundant messages. Setting **Maintain Routing Enabled** to `true` keeps routing enabled, but may flood the station log with messages.

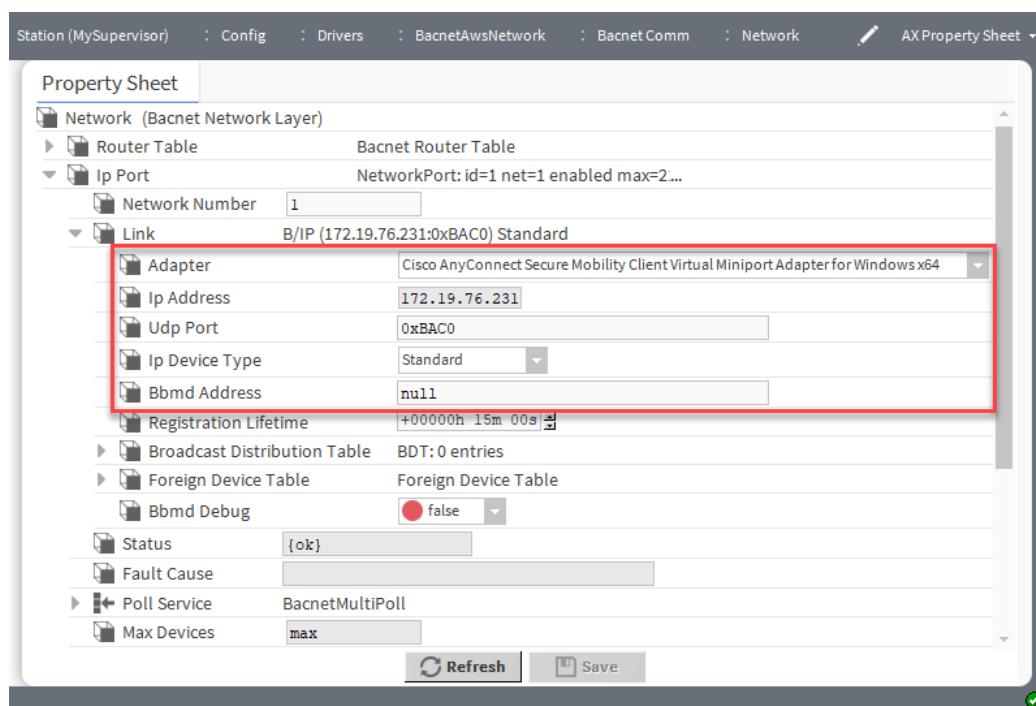
- Step 8** To set up the network port, expand or double-click **Ip Port** and change **Network Number** from -1 (disabled) to the BACnet segment number.

The **Ip Port** properties open.



- Step 9** To identify the type of adapter and define the IP address, expand **Link**.

The **Link** properties open.



Step 10 Select at least the **Adapter** and define the PC's **Ip Address** and click **Save**.

Notice the **Udp Port**. You will need this value later.

Adding a BACnet network in a controller station

The driver supports only a **BacnetNetwork** in a controller station. This procedure adds a network component and configures the station as a local device on a BACnet network segment.

Prerequisites: You are using Workbench and are connected to a controller station.

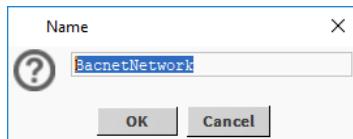
Step 1 Expand the station's **Config** container and double-click **Drivers**.

The **Bacnet Driver Manager** view opens.

At this point there are several ways to add the network component. This procedure uses the **New** button on the **Bacnet Driver Manager**.

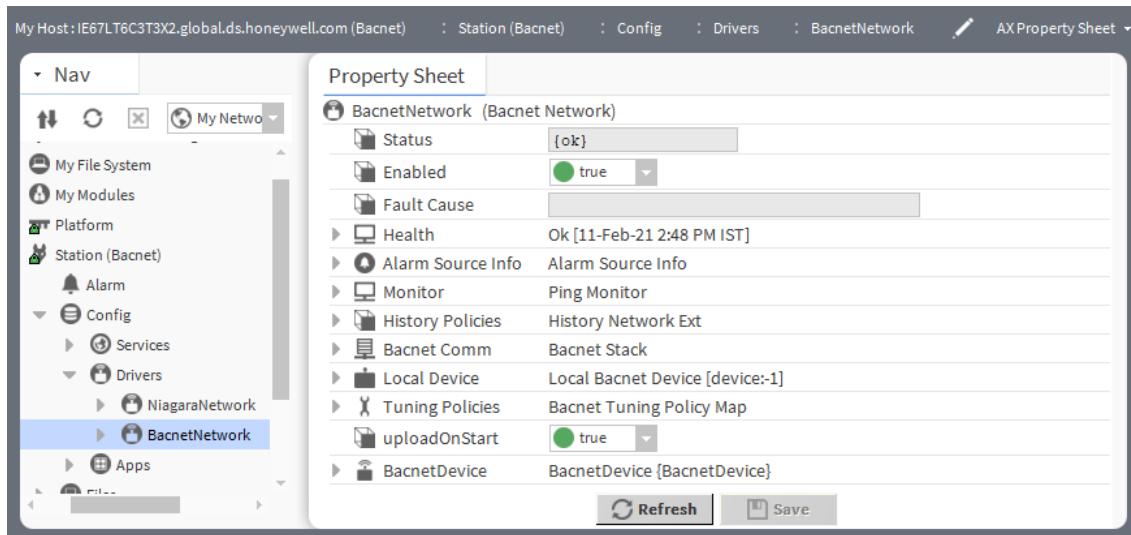
Step 2 Click the **New** button.

The **Name** window opens.



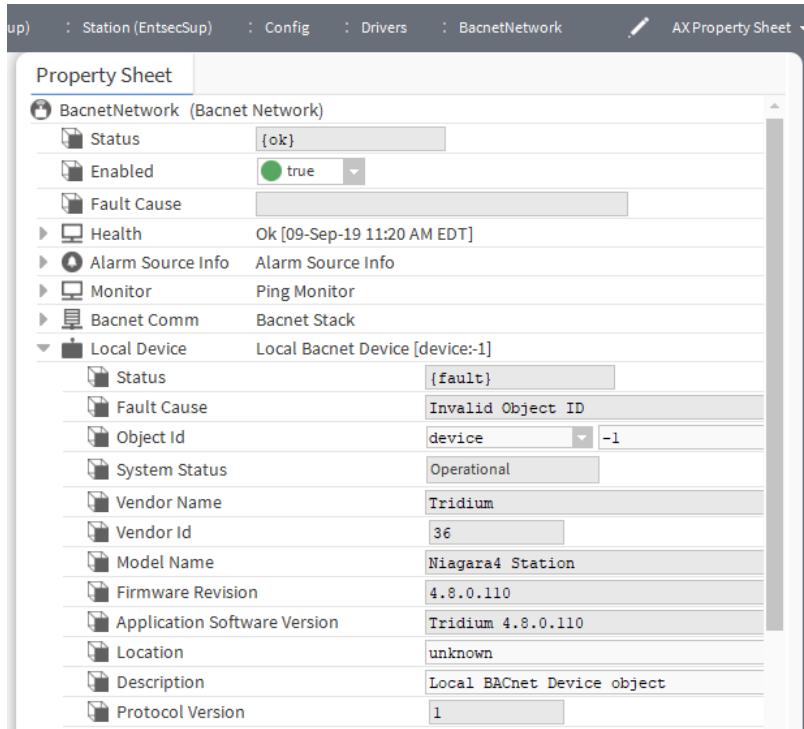
Step 3 Rename the network, or use the default name, and click **OK**.

The added **BacnetNetwork** is available under your **Drivers** container.



Step 4 Expand **BacnetNetwork** and double-click **Local Device**.

The **Local Device** properties become available.



- Step 5** In the numerical **Object Id** property, change the entry from **-1** to a valid, local device instance number, which must be unique across the entire BACnet internetwork (range is 0 to 4,194,302) and click **Save**.

NOTE: A controller is a device on a **BacnetNetwork**. An instance number must be unique for each device in the network.

The **Local Device Status** should change from **{fault}** to **{ok}**, and the **Fault Cause** should change from **Invalid Object ID** to blank.

- Step 6** Expand **Bacnet Comm**→**Network**→**Ip Port**

The **Ip Port** properties open.

- Step 7** Change **Network Number** from **-1** (disabled) to the BACnet segment number.

- Step 8** To define the IP address, expand **Link**.

The **Link** properties open.

- Step 9** Enter the IP address and click **Save**.

Configuring a network with an Ethernet port

The network needs an Ethernet port if all network connections use the Ethernet protocol and wired connections. This procedure configures the Ethernet port on a Supervisor and controller station. These properties include segment number and adapter type.

Prerequisites: You are working in a Supervisor or controller station that has a network component.

- Step 1** To add an Ethernet port, open one of the palettes, expand **NetworkPorts** and drag an **Ethernet-Port** component to the BACnet network **Bacnet Comm**→**Network** folder in the Nav tree.
- Step 2** Expand the BACnet network followed by expanding **Bacnet Comm**→**Network**→**EthernetPort**.
- Step 3** Change **Network Number** from **-1** to the BACnet network number for the network segment to which this local device connects and click **Save**.

- If this is an existing BACnet installation, make sure to use the same network segment number that is already in use.
- If this is a new BACnet installation, you can choose this number (for example: 2).

Step 4 Double-click the **Link** component in the **Property Sheet**.

Step 5 To configure the Ethernet adapter, use the drop-down list of **Adapter Title**, and click **Save**.

This discovers and validates the adapter automatically. The properties **Adapter Description** and **Adapter Name** provide additional information about the selected adapter.

Step 6 Right-click the **EthernetPort** node in the property sheet or Nav side bar and click **Actions→Enable**.

The **Status** property under **Ethernet Port** should report {ok}.

Step 7 When you are done, collapse the nodes you expanded in the Nav tree.

Configuring a network with an MS/TP port

A remote station, which runs on a QNX-based controller is the only device that supports a BACnet MS/TP port.

Prerequisites: You are working in Workbench running on a PC. You have added an MX/TP port to the **BacnetCom→Network node in a controller station**.

Step 1 Expand **Config→Drivers→BacnetNetwork→Bacnet Comm→Network** and double-click **MstpPort** in the Nav tree.

Step 2 Set the **Network Number** from -1 to the BACnet network number for the network segment to which you are connecting and click **Save**.

- If this is an existing BACnet installation, make sure to use the same network number that is already in use.
- If this is a new BACnet installation, configure this number as needed.

Step 3 Double-click the **Link** component in the property sheet.

Step 4 Set the **Port Name** to RS-485.

The station can have a total of three RS-485 ports:

- Set the **Port Name** to COM3 with a regular option card.
- Set the **Port Name** to COM3, COM4 or COM5 with a dual-RS-485 option card.

Step 5 Do one of the following:

- Leave the **Mstp Address** at 0 (the default), and verify that no other MS/TP device has a same address.
- Set the **Mstp Address** to a unique BACnet MAC address on the MSTP trunk.

The valid range is from 0 (default) to 127 decimal. Each BACnet device on the MS/TP network segment must have a unique MAC address. If there is ever a lost token, the device with the lowest MAC address regenerates the token (in this case it will be the station).

Step 6 Set the **Max Master** to the highest known master device on the network with possible room for expansion.

Step 7 Set the **Max Info Frames** to the range from 0 to 100.

This property controls how many messages are sent before passing the token. In some cases, increasing this value up to 50 may improve performance.

Step 8 To save the **MstpPort** changes, click **Save**.

Step 9 Right-click the **MstpPort** (in property sheet or Nav side bar), and select **Actions→Enable**.

The **Status** property under **MstpPort** should report {ok}.

Setting up a controller under the Supervisor's network in the Supervisor station

For the data in a controller station's database to be shared with the Supervisor station and beyond, the unique properties that configure a controller must be present as a node under the Supervisor's BACnet network. While you may discover the controller stations, add each controller configuration as a New AWS or OWS device, or drag a **BacnetAwsDevice** or **BacnetOwsDevice** from the appropriate palette to the BACnet network node in the Supervisor station, this procedure documents using the **New** button on the **Bacnet Aws Device Manager**.

Prerequisites: You are using Workbench running in a PC and connected to the Supervisor station.

Step 1 Expand **Config→Drivers** and double-click **BacnetAwsNetwork**.

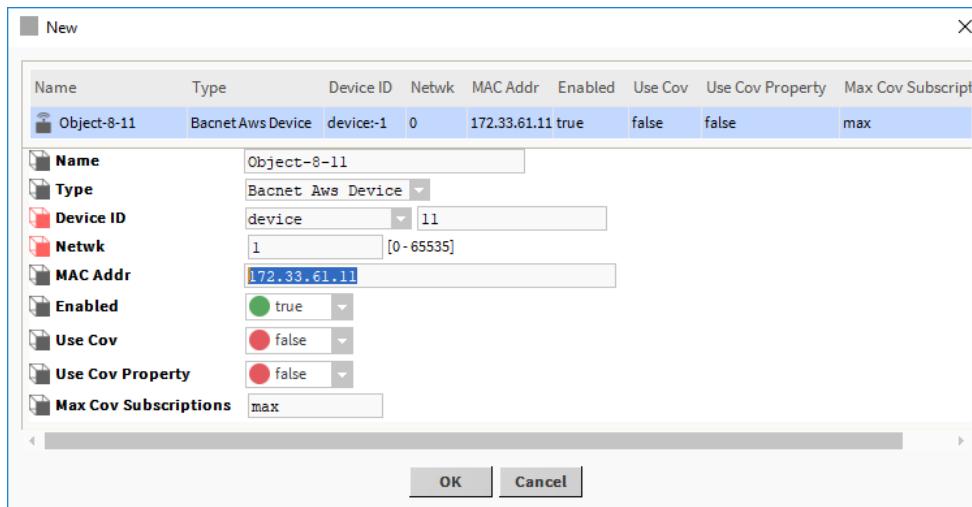
The **Bacnet Aws Device Manager** opens.

Step 2 Click **New**.

A **New** window opens.

Step 3 Add one or more BACnet AWS devices and click **OK**.

The second **New** window opens.



Step 4 Name each device and configure the **Device ID**, **Netwk** and **MAC Addr** (IP address) for each, then click **OK**.

As a best practice, consider naming devices for the controller IP addresses. You don't have to use the entire address, just the last two or three digits.

The driver creates a controller network record for each controller under the Supervisor station's **BacnetAwsNetwork**.

Chapter 3 Controller device and point configuration

Topics covered in this chapter

- ◆ Adding a device using discovery
- ◆ Matching a device
- ◆ Adding a device using the New wizard
- ◆ Adding a device from the palette
- ◆ Configuring device IDs
- ◆ Adding proxy points
- ◆ Expanding polling to report more data
- ◆ Adding a slot to poll for more data
- ◆ Alarm reception configuration
- ◆ Alarm reporting configuration
- ◆ Backing up a station

A remote controller can be a client or a server on the network. It functions as a client on its **BacnetNetwork** when it discovers devices and points, when it imports schedules from another station and when it receives alarms from devices. It functions as a server on a **BacnetAwsNetwork** or **BacnetOwsNetwork** when a Supervisor station requests data from its station.

Like other drivers, you configure controller devices and points using special manager views and property sheets in Workbench. After configuring the **BacnetNetwork** in the remote station, you discover devices and points.

Adding a device using discovery

You can use online discovery to find and create devices under a controller's **BacnetNetwork**. This is the easiest way to populate the controller station with the necessary configured devices.

Prerequisites: You are working in a controller station that has a **BacnetNetwork**.

Step 1 In the Nav tree, expand **Config→Drivers** and double-click the **BacnetNetwork**.

The **Bacnet Device Manager** view opens.

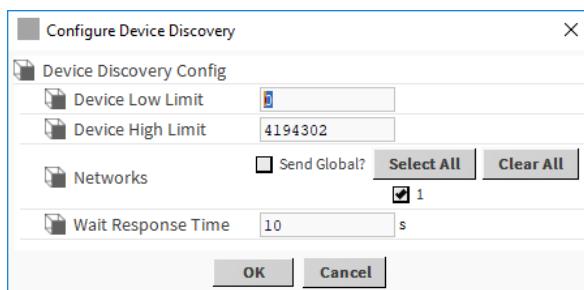
Step 2 To group multiple devices into manageable folders, click **New Folder**, give the folder a name and double-click the new folder name in the Nav tree.

You can discover devices as many times as needed, each time from a different folder.

Each folder opens the **Bacnet Point Manager** view.

Step 3 Click the **Discover** button.

The **Configure Device Discovery** window opens.



Step 4 Depending on the size of the BACnet internetwork, you may wish to change the discovery settings from their defaults to something else.

By default, discovery selects all known networks for the job (each network number that matches an entry in the router table). The **Send Global?** buttons (**Select All** or **Clear All**) are not selected; and the full range of device IDs (0 to 4194302) is specified.

Step 5 To initiate the discovery job, click **OK**.

A progress bar at the top of the view updates as the driver discovers the devices. When the discovery job completes, discovered devices display in the top **Discovered** pane. The bottom pane, labeled as **Database**, is a table of devices that are mapped into the station—initially, this table is empty.

Step 6 To sort the table by a column, click the column head.

Step 7 To map a discovered device in the station, do one of the following:

- Drag the device from the **Discovered** pane to the **Database** pane (an **Add** window opens).
- Double-click the device in the **Discovered** pane (an **Add** window opens).
- Click to highlight the device in the **Discovered** pane, then press “a” (for add).

If you dragged or double-clicked the device, the **Add** window opens.

Step 8 If needed, edit the configuration, and click **OK**.

The driver adds the device to the station database and displays it in the **Database** pane. It dims the device in the **Discovered** pane.

Matching a device

If you added devices to a station database offline, you need to bring the devices online, discover them, and match them to the database records.

Step 1 If the **Discovered** device table is empty, discover the device.

Step 2 Select the device in the **Discovered** pane.

Step 3 Select the device record in the **Database** pane.

Step 4 To match the device with its record in the database, click the **Match** button, or type **Ctrl + m**.

If you clicked the **Match** button, the **Match** window opens.

Step 5 If required, edit the properties in the **Match** window, and click **OK**.

The discovered device appears dimmed, which indicates that it is present in the station database.

Adding a device using the New wizard

Rather than discover devices, you may add them individually. This feature makes it possible to set up devices in the database even when the network is offline.

Prerequisites: You are working in a controller station that has a **BacnetNetwork**.

Step 1 In the Nav tree, expand **Config→Drivers** and double-click the **BacnetNetwork**.

The **Bacnet Device Manager** view opens.

Step 2 In the **Bacnet Device Manager**, click the **New** button.

The first **New** window opens.

Step 3 In **Type to Add** property, select the **BacnetDevice**.

Step 4 Enter the number of devices to add (default value is 1), and click **OK**.

The second **New** window opens.

- Step 5 If needed, edit the configuration of the **BacnetDevice** component(s).

This **New** window is like the **Add** and **Edit** device windows, but in this case, values are not pre-populated from any discovery. Configure the properties based on device connection requirements.

- Step 6 To complete configuration, click **OK**.

The driver adds the device to the station database, and it appears in the **Database** pane.

Once the station is online with the devices, run a discovery job to locate the devices, select each device in both the **Discovered** and **Database** panes and click **Match**. This associates the actual device with its database record.

Adding a device from the palette

You can add a device to the station database using the palette.

Prerequisites: You are working in a controller station that has a **BacnetNetwork**.

- Step 1 Open the **bacnet** palette in the palette side bar.

- Step 2 Expand the **Config→Drivers** node and double-click the BACnet network node.

The **Bacnet Device Manager** view opens.

- Step 3 Drag a **BacnetDevice** component from the palette to the BACnet network node in the Nav tree.

The device appears in the **Database** tab of the **Bacnet Device Manager** view.

- Step 4 To rename the device, double-click the current name, change the name and click **OK**.

The **Edit** window is like the **Add** device window.

The system updates the device in database.

Configuring device IDs

Each out-of-the-box BACnet controller or local device may be configured with the same device ID. Identical device IDs cause devices to appear in fault (orange) in the **Discover** pane. This procedure changes the device ID in one or more devices. Changing a device's **Object Id** property on a BACnet network **Property Sheet** does not write to the actual device, it only determines how the station communicates with the device.

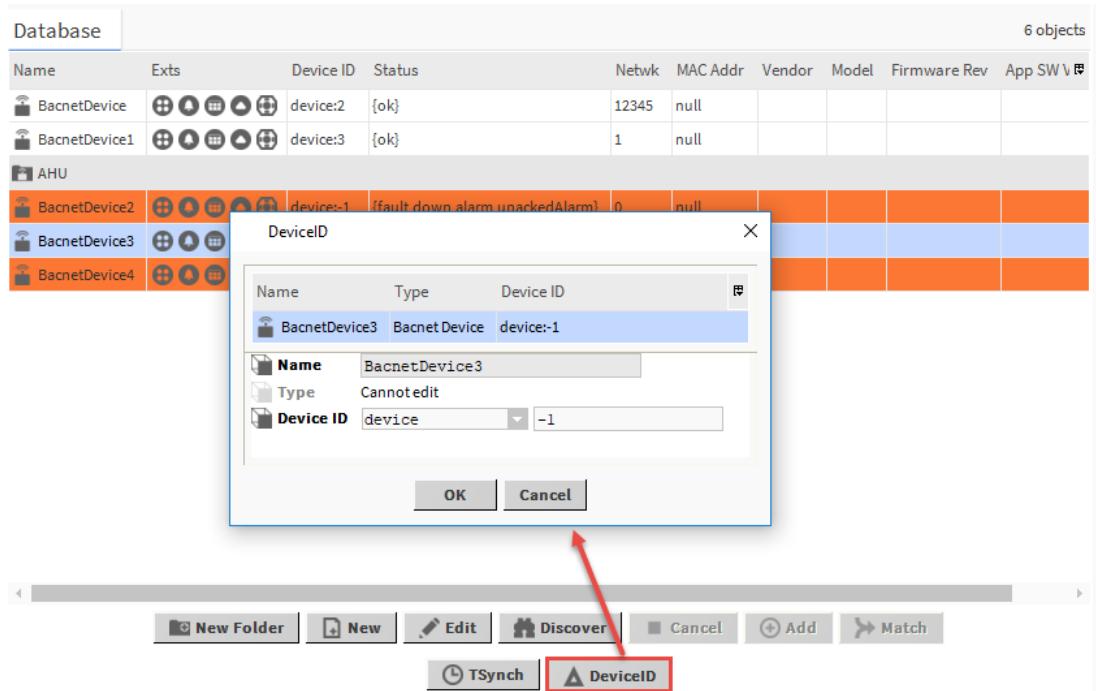
Prerequisites: The BACnet controller or device is capable of accepting a change to its device ID. The **Bacnet Device Manager** view is open with discovered devices displayed in the **Database** pane.

- Step 1 Select the devices that are in fault (orange) and click the **Device ID** button at the bottom of the view.

The driver prompts you to confirm this action.

- Step 2 To continue, click **Yes**.

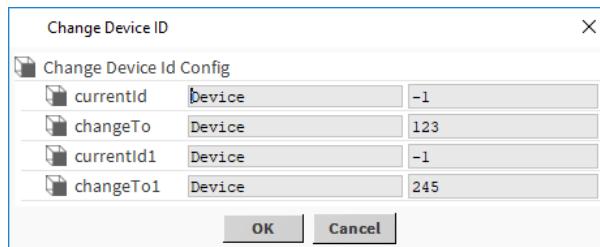
A popup **DeviceID** window opens.



Step 3 Do one of the following:

- If you selected a single device, fill in a unique **Device ID** for the device and click **OK**.
- If you selected multiple devices, fill in the first unique **Device ID** for the first device and click **OK**.

The **Change Device ID** window opens. The driver displays a summary of the proposed changes.

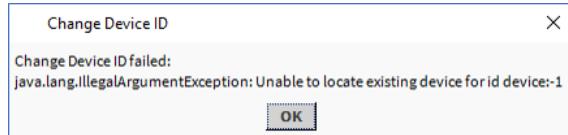


Step 4 To continue, click **OK**.

The driver launches a Change Device Id job, with an entry in the station's **JobService**.

If you selected multiple devices, the driver assigns new Device IDs sequentially, starting with the unique number for the first device. For example if you selected three devices and filled in 62, the driver changes the Device IDs of the three devices to 62, 63, and 64.

The job ends with a popup notification, if the Device ID change was not successful.



Step 5 To finish changing IDs, click **OK**.

Adding proxy points

As with device objects in other drivers, each remote controller station has a **BacnetDevice** with a **Points** extension that contains the device points. The default view for the BACnet points extension is the **Bacnet Point Manager**. If the controller is online, this is the easiest way to accurately add BACnet points under a **BacnetDevice**.

Prerequisites: You are working in Workbench connected to a remote host station.

Step 1 Expand **Config→Drivers** and double-click **BacnetNetwork**.

A **Bacnet Device Manager** view opens.

Step 2 Under the **Exts** column, double-click the Points icon (⊕) in the row.

The **Bacnet Point Manager** view opens.

Step 3 To discover the points, click **Discover**.

A progress bar appears at the top of the view. When the discovery job completes, the system lists the discovered points in the top pane of the view. Each point occupies one row.

Step 4 If needed, you can click to expand the objects (by default, **presentValue** is on top of any object).

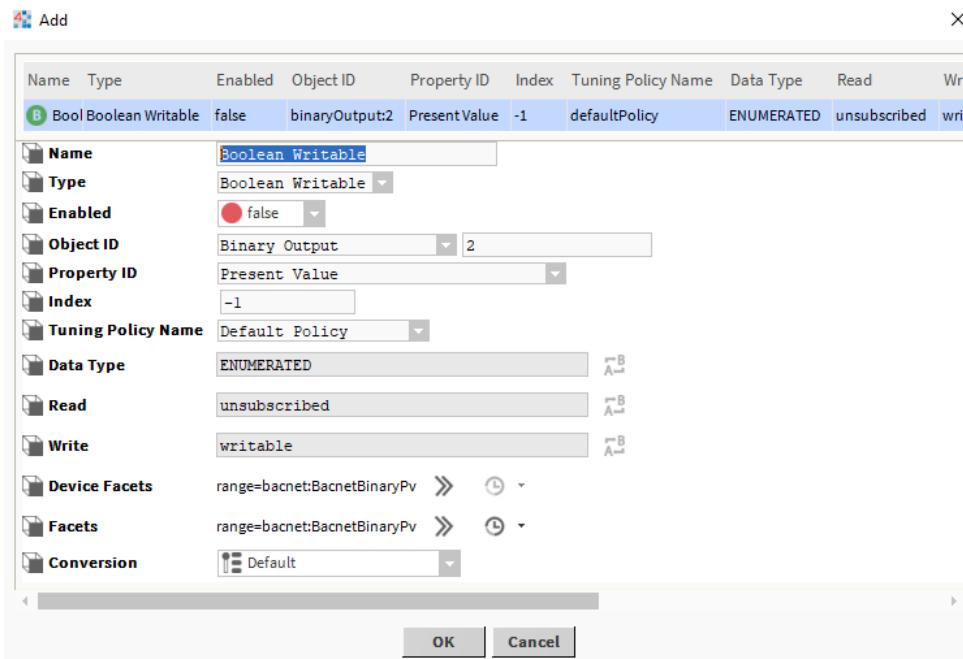
Step 5 Select the property of the discovered device you wish to proxy.

Typically, this is the default (top) **presentValue** property, but you may wish to create one or more additional proxy points for other properties (expand the discovered object). Possible examples include **eventState**, or, for a priority-type device, a particular **priorityArrayIndex** (level).

Step 6 To map selected items in the station database do one of the following:

- Drag from the **Discovered** pane to **Database** pane.
- Double-click an item in the **Discovered** pane.
- Click to select in **Discovered** pane, then press a.

The **Add** window opens.



The discovered object determines the initial property value. To deal with proprietary object types and properties, refer to the numerical codes supplied by the device vendor.

Step 7 Do one of the following:

- Click to highlight the **Object Id** or the **Property Id** and type in the needed numeric code. For example, a proprietary property 1000 may be included on a company's analog input objects.
- If you added the relevant enumeration value into the BACnet device's **Enumeration List** property, you should be able to select the **Object Id** or **Property Id** by name from the drop-down list.

Step 8 To complete the configuration, click **OK**.

The system adds the points to the station database and displays them in the **Database** pane.

Step 9 To manually add points, click the **New** button in the **Bacnet Point Manager** or drag points from the **bacnet** palette.

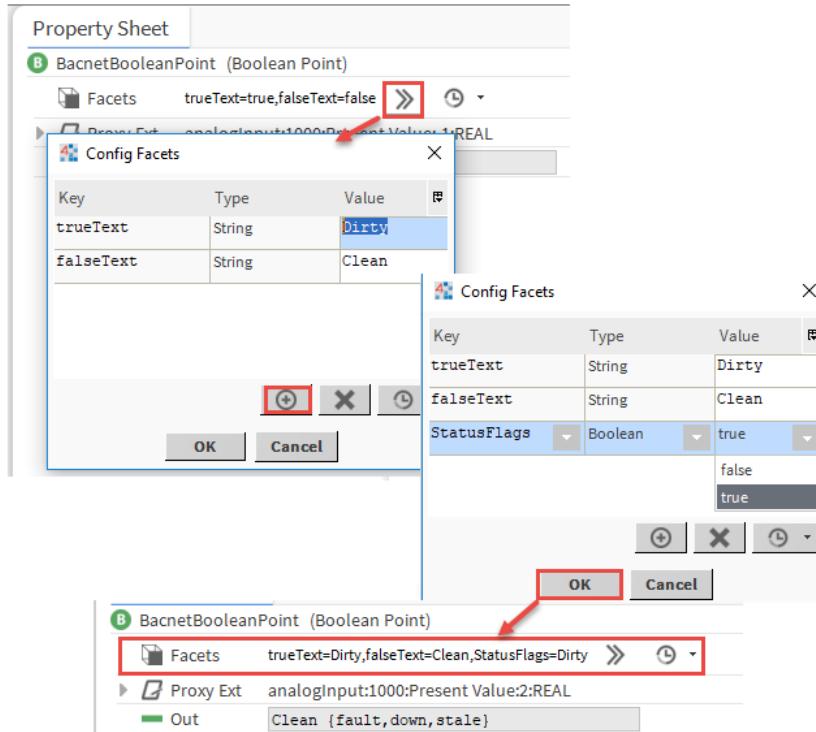
Expanding polling to report more data

By default, the **bacnet** driver polls or collects on COV (Change of Value), the data contained in a single property for each proxy point. On occasion, more information may be required. For example, if you configure the driver to poll a binary input object's **presentValue** using its **Property Id**, and the object goes into alarm, you receive no indication of the in-alarm condition because the driver does not poll for status. This procedure updates facets to configure the driver to return data for additional properties.

Prerequisites: The computer is connected to a remote controller station using Workbench.

Step 1 Expand **Config**→**Drivers**→**BacnetNetwork**→**BacnetDevice**→**Points** and double-click the point.

Step 2 Expand the **Proxy Ext** and click the chevron next to **Facets**.



Step 3 Edit the point facets to include additional properties (beyond the configured property) for polling—one of which could be **statusFlags**.

This facets edit applies to the main point's facets (and not device facets in its ProxyExt).

- Step 4 To add to the point poll, click the add button () and add Boolean facet(s) with any of these (Key) names:
- **statusFlags** for the **Status_Flags** property
 - **priorityArray** for the **Priority_Array** property
 - **eventState** for the **Event_State** property
 - **reliability** for the **Reliability** property

- Step 5 To complete the add, click **OK**.

After adding additional facet(s), the status of a proxy point reflects resulting metadata from additional polled properties in these ways:

- **statusFlags** merge with the proxy point status (bits of BStatus).
- **priorityArray** shows bac=X, where X is 1 to 16.
- **eventState**, as a facet, shows state=stateName, where stateName is the name of a BacnetEventState enumeration, such as normal, offnormal, highLimit, and so on.
- **reliability**, as a facet, shows reliability=reliabilityEnum, where reliabilityEnum is one of the BacnetReliability values, such as noFaultDetected, overRange, and so on.

Adding a slot to poll for more data

In addition to editing a proxy point's facets to poll additional properties, you can add a slot to the point's ProxyExt, and direct it to a specific property—even a property of a different object or device. This technique is based on a Bacnet Device Object Property Reference (DOPR) format, where numerical codes are required for processing.

Prerequisites: The computer is connected to a remote controller station using Workbench.

- Step 1 Expand **Config→Drivers→BacnetNetwork→BacnetDevice→Points** followed by expanding the point.

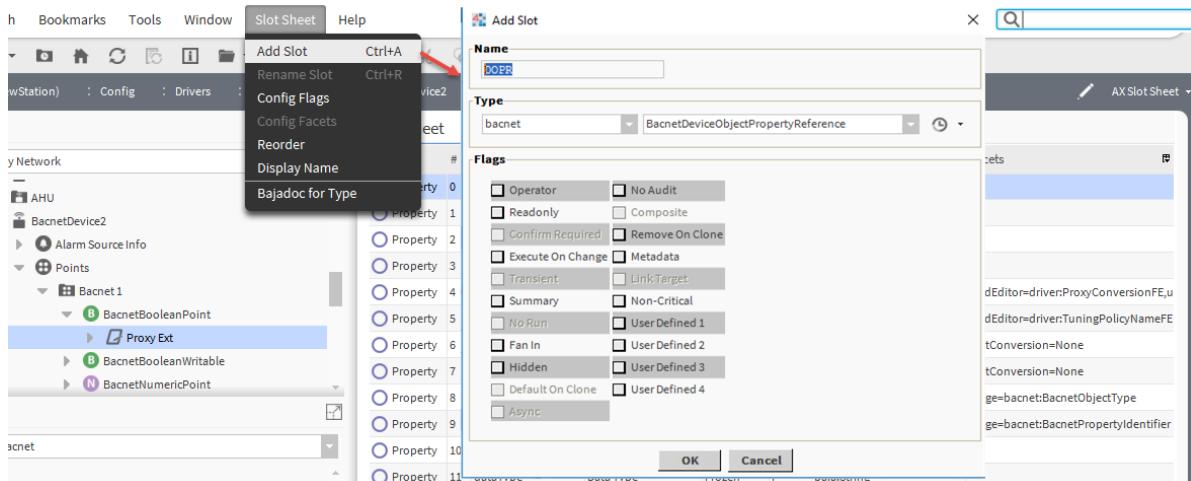
- Step 2 Right-click the point's **Proxy Ext** and click **Views→Slot Sheet**.

The **Slot Sheet** view opens.

| Slot | # | Name | Display Name | Definition | Flags | Type | Facets |
|----------|----|-----------------------|-------------------------|------------|-------|------------------|--------|
| Property | 0 | visible | Visible | Frozen | | baja:Boolean | |
| Property | 1 | enabled | Enabled | Frozen | | baja:Boolean | |
| Property | 2 | layout | Layout | Frozen | | bajau:Layout | |
| Property | 3 | styleClasses | Style Classes | Frozen | h | baja:String | |
| Property | 4 | styleId | Style Id | Frozen | h | baja:String | |
| Topic | 5 | keyEvent | Key Event | Frozen | | bajau:KeyEvent | |
| Topic | 6 | mouseEvent | Mouse Event | Frozen | | bajau:MouseEvent | |
| Topic | 7 | focusEvent | Focus Event | Frozen | | bajau:FocusEvent | |
| Property | 8 | title | Title | Frozen | | baja:String | |
| Property | 9 | ord | Ord | Frozen | | baja:Ord | |
| Property | 10 | location | Location | Frozen | rto | baja:String | |
| Property | 11 | progress | Progress | Frozen | rto | baja:Double | |
| Property | 12 | progressRunning | Progress Running | Frozen | rto | baja:Boolean | |
| Property | 13 | showProgressIndicator | Show Progress Indicator | Frozen | | baja:Boolean | |
| Property | 14 | contextMenuEnabled | Context Menu Enabled | Frozen | | baja:Boolean | |
| Action | 15 | reload | Reload | Frozen | | void(void) | |
| Topic | 16 | initialized | Initialized | Frozen | | baja:Value | |
| Topic | 17 | closing | Closing | Frozen | | baja:Value | |
| Topic | 18 | loaded | Loaded | Frozen | | baja:Boolean | |
| Topic | 19 | statusItem | Status Item | Frozen | | baja:String | |

Step 3 Right-click an empty area of the **Slot Sheet** and click the **Add Slot**.

The **Add Slot** window opens.



Step 4 Select **bacnet** from the first, and **BacnetDeviceObjectPropertyReference** from the second **Type** drop-down list and click **OK**.

Step 5 To open the **ProxyExt's Property Sheet**, double-click the **ProxyExt** in the Nav tree.

The **Property Sheet** opens.

The screenshot shows the Niagara 4 BACnet Driver configuration interface. At the top, there's a navigation bar with tabs like 'BacnetDevice2', 'Points', 'Bacnet1', 'BacnetBooleanPoint', and 'Proxy Ext'. To the right of the navigation bar is a button labeled 'AX Property Sheet' with a red box around it. Below the navigation bar is a table with various properties for a point, such as 'Object Id' (Analog Input), 'Property Id' (Present Value), 'Data Type' (REAL), 'Read Status' (Polled), 'Write Status' (Read Only), and 'newSlot' (analoginput-1_presentValue[-1]). A section labeled 'DOPR' is expanded, showing its properties: 'Object Id' (Analog Input), 'Property Id' (85), 'Property Array Index' (-1), and 'Device Id' (device). At the bottom of the main configuration area are 'Refresh' and 'Save' buttons. Below the main configuration area is a navigation tree titled 'Bacnet 1 (Bacnet Point Folder)'. It contains several entries: 'BacnetBooleanPoint' (Clean {fault,down,stale}), 'BacnetBooleanWritable' (false {fault,down,stale} @ def), 'BacnetNumericPoint' (0.0 {fault,down,stale}), and 'BacnetNumericWritable' (0.0 {fault,down,stale} @ def). An arrow points from the 'BacnetBooleanPoint' entry in the navigation tree to the 'DOPR' section in the main configuration area.

Step 6 Expand the new slot and edit its DOPR properties to poll and display the needed property.

Due to the complexity of this procedure, it is expected to be infrequently used. However, in certain applications, such as with Event Enrollment objects (a BACnet object that monitors other objects for the purpose of generating alarms based on its own algorithm), it may prove useful.

The resulting metadata are included with the name of the DOPR. If it is an Event Enrollment object, and the DOPR property is `Event_State` (the alarm state of the Event Enrollment object), the facet appears as follows:

```
<EEinstanceNumber>=<EE event state>
```

if another property, the facet is:

```
<EEinstanceNumber>_<EEpropId>=<prop value>
```

Alarm reception configuration

The `bacnet` driver supports intrinsic event reporting more commonly called alarming. Support is available for both client and server operations, which means that a station can both receive (and reroute) BACnet alarms from Bacnet proxy points, and also send BACnet alarms from components that have been exported as BACnet objects.

Figure 1 Example of a BACnet alarm received in an alarm console

The screenshot shows the AX Alarm Console interface. At the top, there is a header bar with a search field and a time range selector. Below this is a table titled "Open Alarm Sources" with four entries. The second entry is highlighted in blue. A modal window titled "Open Alarm Sources" is displayed over the main table, showing a single entry for "NiagaraNetwork SupplyAirResetNew1". Below this, a larger modal window titled "Alarm Record" provides detailed information about the selected alarm. The "Alarm Data" section is highlighted with a red border and contains the following details:

| | |
|--------------|----------------------------------------------------------------------------------------------------|
| Timestamp | 19-May-18 2:35:34 PM IST |
| Uuid | 9a5c17b4-720c-4397-b20b-7ca3122083f7 |
| Source State | Offnormal |
| Ack State | Unacked |
| Ack Required | true |
| Source | NiagaraNetwork SupplyAirResetNew1 local:station:slot:/Drivers/NiagaraNetwork/SupplyAirResetNew1 |
| Alarm Class | Default Alarm Class |
| Priority | 255 |
| Normal Time | null |
| Ack Time | null |
| User | Unknown User |
| Alarm Data | Escalated |
| Message Text | Ping Failed |
| Notes | » |
| Source Name | NiagaraNetwork SupplyAirResetNew1 |
| Time Zone | Asia/Calcutta (+5:30) |

At the bottom of the "Alarm Record" window, there are buttons for "Acknowledge", "Hyperlink", "Notes", and "Close".

Prerequisites to receive alarms

To be received, a BACnet alarm must be generated by a BACnet device that is represented in the remote station. Otherwise, the driver discards the alarm.

Ideally, the alarming BACnet object should be represented by a Bacnet proxy point, in which case the alarm record source provides the proxy point name and ord. If the alarming BACnet object is not proxied in the station, the alarm record shows the source as the Alarms device extension (ord) of the corresponding **BacnetDevice**. In either case, alarm details provide the source **Object Id**, for example `analogInput 3 current toState` and `statusFlags` as well as any alarm message text.

Restrictions

AlarmClass components in the station's **AlarmService** receive the BACnet alarms. Each device uses only one **AlarmClass**, as referenced in each **BacnetDevice**'s **Alarms** device extension. You can specify the same one (for example, the **DefaultAlarmClass**), or create and use separate alarm classes, as needed.

In the **Notification Class** object used by the sending BACnet device, any receiving station must be added in the **Recipient_List** property, using its numerical device ID (**Local Device Object Id**). You configure this using the BACnet device's native configuration tool, or, if the device supports it, directly from the driver using corresponding **Config** components for objects in that device.

Configuring a BACnet device to send alarms

Sometimes a remote BACnet device may not allow this configuration. In this case, you must use third-party tools to configure Notification Class objects in a BACnet device as described below.

Prerequisites: The station is open and a BACnet device has been added and is communicating in the station.

- Step 1 Expand **Config**→**Drivers**→**BacnetNetwork**→**BacnetDevice** and double-click on the **Config** node in the Nav tree.
The **Bacnet Config Manager** view opens.
- Step 2 Discover the objects in the device.
- Step 3 Add the **Notification Class** config object for the alarms you wish to receive.
To determine which one to add, add a config object for an object that will be alarming, and inspect its **Notification Class** property. This specifies which notification class object will route those alarms.
- Step 4 After determining the referenced notification classes, delete unused config objects.
- Step 5 Right-click the **Notification Class** config object, and select **Actions**→**Add Destination**.
The **Add Destination** window opens.
- Step 6 Expand the **Recipient** property and enter the station's device ID, replacing the **-1** with value of the station's **BacnetNetwork**→**Local Device, Object Id** property.
- Step 7 Specify the (integer) **Process Identifier** to use in sending alarms to the station (the default is **0**).
- Step 8 Configure other properties, and click **OK**.
- Step 9 To save the changes, click **Save**.

Alarms are now configured to send to the client station.

NOTE: For BACnet alarms from a device to be successfully received in a client station, the Process Identifier in the Recipient List entry for the station (in the BACnet notification class object of the sending device) must be matched when configuring the BacnetDevice's Alarms device extension, **Niagara Process Id** property.

Configuring a station to receive alarms

The BACnet driver converts incoming alarms from remote BACnet devices to Framework alarms, which can then be routed through the Framework's alarm subsystem to recipients (console, station, printer, email, and so on). A console operator acknowledges a BACnet alarm just as they acknowledge a native Framework alarm.

- Step 1 In the receiving station, create **BacnetDevice** components for all devices from which the station will receive alarms.
- Step 2 It is also recommended to create BACnet proxy points for all BACnet objects that will send alarms to the station.
- Step 3 Under the station's **AlarmService**, add **AlarmClass** components as needed, setting ack requirements and priorities.
Each **BacnetDevice** can use only one **AlarmClass** (either used by other devices, or unique to that device).
- Step 4 Link the **AlarmClass** components into whatever alarm recipient objects are needed (**AlarmConsole**, etc.).
- Step 5 For each **BacnetDevice** in the station, in its **Alarms** device extension:
 - Specify the **AlarmClass** component to use.

- Set the **Niagara Process Id** to match the BACnet process identifier (integer value) used in sending alarms to the station.

Step 6 Using third-party tools for each remote BACnet device, configure the **Notification Class** object that routes alarms to include (add) the device **Object Id** of the station to its **Recipient_List** property.

This is the numerical value seen in the station's **BacnetNetwork→Local Device Object Id** property.

Alarm reporting configuration

The BACnet driver supports intrinsic event reporting more commonly called alarming. Support is available for both client and server operations, which means that a station can both receive (and reroute) BACnet alarms from Bacnet proxy points, and also send BACnet alarms from components that have been exported as BACnet objects.

Any station that is exporting control points to BACnet can also route BACnet alarms from those points to BACnet devices. A **BacnetDestination** component in the Alarming folder of the **bacnet** palette is used to specify a receiving BACnet client device.

Prerequisites for exporting alarms

Only Framework components in the station that are exported as BACnet objects can generate BACnet alarms, where alarming capability is intrinsic to that object. For example, a control point exposed as an analog input object must use an out-of-range algorithm provided by an **OutOfRangeAlarmExt**. Another alarm extension, such as the **StatusAlarmExt**, is not supported.

In general, only components exported as BACnet object types: Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, Binary Value, Multistate Input, Multistate Output, or Multistate Value are good candidates for exporting alarms. In a station these include control points (often proxy points) and components from the **kitControl** module, at least those with a **NullProxyExt**. **Calendar** and **Schedule** components cannot export alarms.

A separate **BacnetDestination** component must be added under the station's **AlarmService** for each remote BACnet client to which alarms need to be sent (one-to-one **BacnetDestination** to device). In this component, you specify the unique BACnet device identifier for this device, and possibly make other non-default configuration changes.

Restrictions on exporting alarms

Only one alarm extension is supported on a point configured for BACnet alarming (referencing an alarm class linked to a **BacnetDestination**).

Each alarm class component that you link to a **BacnetDestination** component must be exported to BACnet, as a notification class object. Linked **BacnetDestination** components automatically appear as entries in the **Recipient List** of these notification class objects.

Unless the alarm classes you are using define escalation levels, alarms are not regenerated. If a BACnet device is down when the component sends an alarm, the alarm the driver does not save the alarm for retransmission. Of course, the station maintains a record of alarms in the alarm database.

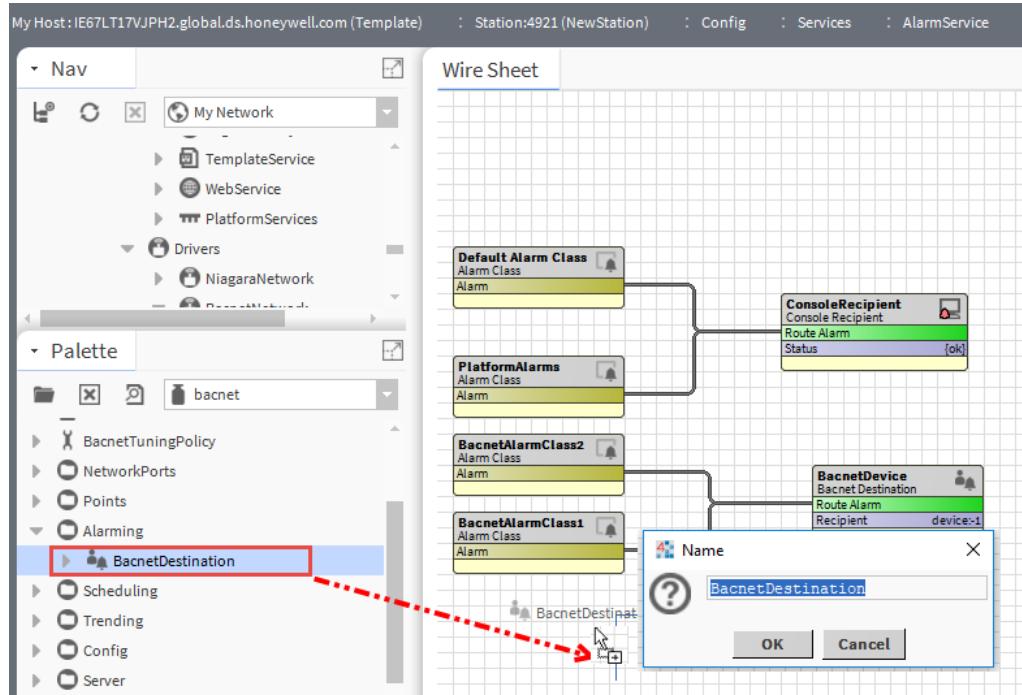
Configuring the AlarmService for BACnet alarms

Two procedures are required to configure BACnet alarms. First, you configure the **AlarmService** of each station to process BACnet alarms. Next, you configure and export the BACnet objects that are capable of generating alarms. This procedure configures the **AlarmService**.

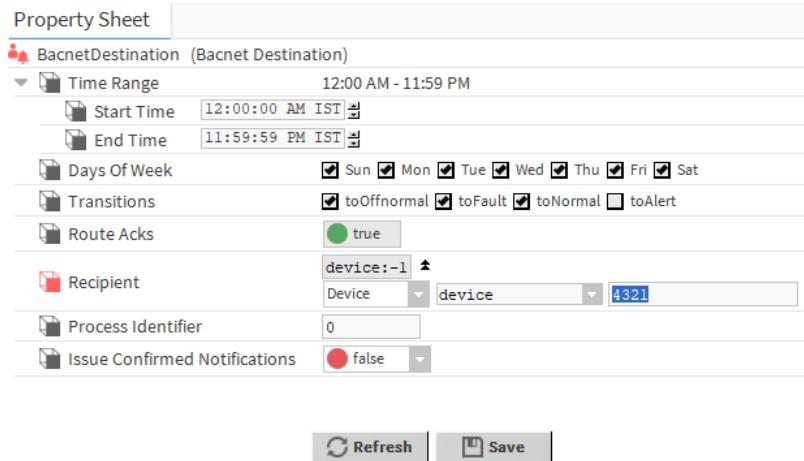
Prerequisites: You have not exported the objects yet. The **bacnet** palette is open. The **AlarmService** wire sheet is open.

Step 1 Under the station's **AlarmService** wire sheet, add the **AlarmClass** components to use with alarmable control points exported to BACnet.

- Step 2** Set ack requirements and priorities as needed, but do not export yet.
- Step 3** Drag a **BacnetDestination** recipient from the **bacnet** palette (Alarming folder) to the wire sheet.
- The **Name** window opens.



- Step 4** Name the destination component and click **OK**



- Step 5** Double-click the **BacnetDestination** component and set the **Recipient** property to the BACnet device ID for the BACnet client that will receive alarms.
- Step 6** Create a separate **BacnetDestination** for each BACnet client to receive alarms.
- Step 7** Link the **AlarmClasses** to the **BacnetDestination** recipients (and other recipients as needed).
- The **AlarmService** is prepared to process BACnet alarms.

Exporting BACnet objects that generate alarms

Two procedures are required to configure BACnet alarms. First, you configure the **AlarmService** of each remote host station to process BACnet alarms. Next, you configure and export the BACnet objects that are capable of generating alarms. This procedure configures and exports the BACnet objects that generate alarms.

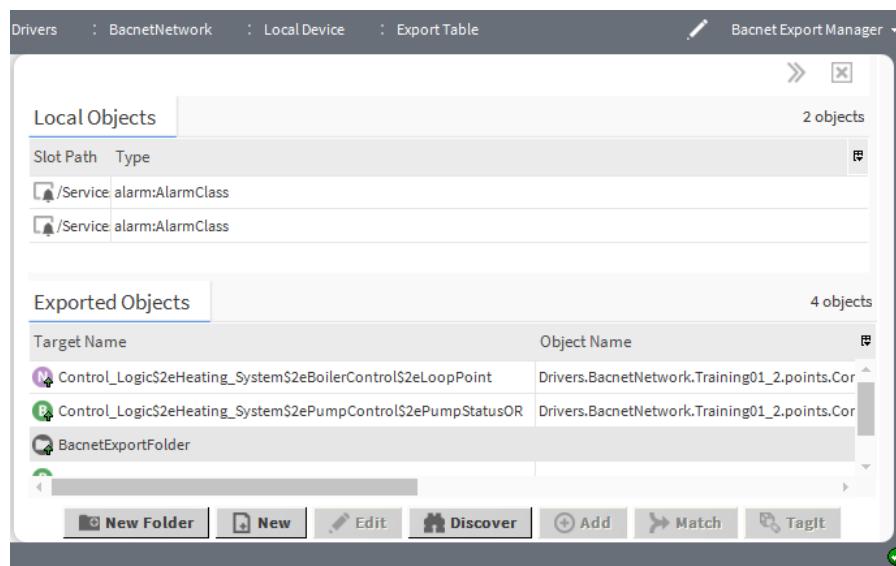
Prerequisites: The AlarmService has been configured to process BACnet alarms.

Step 1 Expand **BacnetNetwork→Local Device** and double-click **Export Table**.

The **Bacnet Export Manager** opens.

Step 2 Discover the objects capable of generating alarms.

Step 3 Export all **AlarmClass** components that are linked to **BacnetDestination** recipients.



These export as only one object type: Notification Class.

NOTE: For the discover job to find them quickly, choose **Of Type: for Alarm Class**.

A BACnet Notification Class object instance number automatically starts at 0 for the first exported alarm class, then increments by one for each alarm class.

Step 4 Add the appropriate alarm extension to control points that alarm to BACnet, populating the **Alarm Class** for each with an **AlarmClass** component.

Step 5 Do one of the following:

- Select and drag discovered objects from the **Local Objects** pane to the **Exported Objects** pane.
- To identify a single object to export, click **New**.
- To export one or more object, select the object(s) and click **Add**.

The **New** or **Add** object export window opens.

The station is now configured to generate BACnet alarms for the exported control points.

Testing BACnet alarm generation

This procedure verifies that alarms are working properly.

Prerequisites: You should have access to third-party BACnet tools for the remote BACnet clients. In addition, the AlarmClasses involved are linked to a standard AlarmConsole recipient.

- Step 1** Force into an alarm condition one of the exported control points that is exported to BACnet (and assigned to an exported **AlarmClass**).
For example, you can do this by adjusting an alarm high limit or low limit.
The station routes the alarm via the exported **AlarmClass** to the linked **BacnetDestination** (client) devices. This translates the alarm from a standard format to a BACnet format. At the same time, the alarm should be available in the standard alarm console.
- Step 2** Using a third-party tool to access the remote BACnet client, verify that the remote client receives the alarm.
- Step 3** Open the linked alarm console and verify that it received the alarm.
- Step 4** Using the third-party BACnet tool, issue an acknowledgment for the alarm.
- Step 5** Verify that the station received the acknowledgment. The alarm acknowledgment will be saved in the alarm database.
- Step 6** Remove the test alarm from the list of unacknowledged alarms in the alarm console.

Backing up a station

This procedure summarizes the steps for backing up a remote BACnet station.

Prerequisites: You have admin-level write permissions on the **BacnetNetwork** and **BacnetLocalDevice**, at a minimum. The password is not blank, not the default value, and you know what the password is.

- Step 1** Un-hide the normally hidden **Reinitialize Allowed** property of the **Server** layer component (child of the **BacnetComm** container under the **BacnetNetwork**) and set its value to `true`.
This value defaults to `false`. Once the backup is complete, you can hide it again.
- Step 2** Right-click the station node and click **Backup Station**.

The **Local Device's Backup and Restore State** property changes from `Idle` to `Preparation` for backup and the station's **BackupService** begins preparing a backup `.dist`, which contains the following:

- Station database — `config.bog`
- Alarm history database
- History database
- Module metadata — information about which version of each module is loaded. The modules themselves are not backed up.

The system exports the `.dist` file as a **File** object—part of the **configurationFiles** property of the device object. The backup client then reads the **File** object using BACnet file access services.

Finally, the **Local Device's Backup and Restore State** property changes back to `Idle` and normal station operation resumes.

Chapter 4 Offline device and point configuration

Topics covered in this chapter

- ◆ EDE device and point configuration files
- ◆ EDE state texts support file
- ◆ Configuring a station for EDE
- ◆ Exporting controller device and point data
- ◆ Validation criteria for data to be discovered
- ◆ Discovering device data offline
- ◆ Discovering point data offline
- ◆ EDE troubleshooting

The EDE (Engineering Data Exchange) configuration utility, running on either a controller or PC platform, can discover BACnet devices and points from EDE configuration files, creating an offline BACnet network of corresponding devices and points with the same hierarchy and type information as described in the EDE configuration files.

Support for EDE offline discovery is built into the Framework. It does not run in the remote controller platform although it can operate on the controller station.

A bit of background

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.

In a multi-vendor system, interacting parties need to exchange engineering data. In a BACnet system, BACnet online discovery, which most server and client devices support, exchanges data, such as data point object properties, however, this data exchange, by definition, can only be accomplished online and depends on BACnet discovery functionality being available in both the server and client.

These conditions prevent this discovery and require an offline solution:

- Lack of discovery functionality in both server and client
- A server that was not set up with engineering data before it was installed and operational on a BACnet network
- Lack of client devices for online discovery

It was with these conditions 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 offline. The original concept was to provide only the BACnet data-point list in a simple human-readable form via a CSV (Comma-Separated Value) file. The general format actually consists of four different CSV spreadsheets, which collectively are known as the EDE (Engineering Data Exchange) configuration files. The BIG-EU document that defines them is the *Engineering Data Exchange Template for BACnet Systems*.

The latest version of the standard is 2.3 (2017) introduced the EDE table. Since this version was published, many BACnet equipment vendors have adopted the EDE file format to define and distribute device point content. To provide offline engineering programming, many include an EDE master configuration file-import mechanism in their server device.

NOTE: The data contained in the EDE configuration files are not as extensive as the data discovered online and discrepancies can occur when discovering devices and points online after initially using the EDE configuration files offline to build the engineering database in a server station. 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 BACnet product vendors have adopted it as the *de facto* standard for M2M data exchange.

When to use this utility

Within the Niagara Framework you should use this utility:

- When upgrading to a new controller. You can export EDE data from the station running on the old controller and use the exported EDE configuration files with a discovery job to update the station database in the new station before connecting devices.
- To configure a Supervisor station. You can export EDE data from each controller station and use the exported EDE configuration files to discover and update the database in the Supervisor station.

Offline configuration

This utility configures a station that is offline, where offline means that no BACnet devices, which will be imported into the station, are currently connected to the station's controller. The station itself is running and accessible using Workbench.

To facilitate network configuration, the station requires the Bacnet EDE Service component, which handles the appropriate job submissions for device and point discovery. Standard manager views on the BACnet network and BACnet device components manage these job submissions.

The discovery process follows the standard system process of discovery on the network for all devices and on each device for the underlying points.

Licensing

This utility does not require a separate license. A BACnet license is sufficient to use this feature.

Object export not supported

This utility supports the automatic creation of a BACnet network. Device and point data are intended to configure connections to real-world BACnet devices and points. The purpose is to import state information from the points into a running station. This utility does not support the configuration of existing components for export as point data to BACnet devices.

EDE device and point configuration files

The EDE configuration utility relies on CSV files. You can create these files using Notepad or Excel in preparation for discovering devices and points.

Two CSV (Comma Separated Variable) configuration files provide a limited amount of information to identify each device and point:

- One or more EDE master configuration files that contain the device and point information
- A state texts support file

Both files must conform to the EDE 2.3 specification.

The EDE specification for this utility does not support two additional configuration files: the Unit Text File and Object Type File because unit codes and object types are already defined in the BACnet specification.

The utility supports a single BACnet network instance. Multiple EDE master configuration files for a single BACnet network allow the discovery of devices and points defined in each EDE master configuration file for addition to the current BACnet network instance.

CAUTION: Data security is your responsibility. Store EDE configuration files in a secure location where they cannot be modified by unknown users.

EEDE master configuration file device data specification

The CSV file includes EDE in its name and contains project information as well as the list of data points chosen for inter-operation.

Figure 2 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-typ | object-ins | descrip | present-v | min-pres | max-pres | comand | support | hi-limit | low-lir | state-text | unit-code | vendor | |
| 7 064 - UCU3213VAV | 101064 064 - UCU3213VAV | 8 | 101064 | | | | | | | | | | | | |
| 8 064 - UCU3213VAVN | 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 |

Due to the limited information provided by the EDE format specification, these properties are included for each discovered device:

| Property | Required? | Description |
|-------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Display Name | mandatory | Identifies the device with a system-wide unique name. |
| Object Name | mandatory | Provides the device address, which must be a unique within a BACnet network. The utility makes exceptions for dynamically-generated objects (for example, for a file to back up and restore data). |
| Object Type | mandatory | Contains a decimal value to represent the device. For example, an Object Type of Calendar requires a code number of 06. The utility ignores values that are outside of the BACnet specification. The utility ignores Object Type values that are outside of the BACnet specification because it is not possible to determine what appropriate Niagara object type to use to represent the new Object Type given only an Object Name. |
| Device Object Instance | mandatory | Identifies the device that contains the data object. |
| Object Instance | mandatory | Defines a decimal number with which to identify the object. Object Type and Object Instance together provide a unique identifier in each device. |
| Description | optional | Provides important additional information about the object and its function. |
| Vendor Specific Address | optional | May identify an address used in the server device, such as a terminal number. This information provides an internal data-point identification or reference. |

EDE master configuration file point data specification

The EDE master configuration file for point discovery contains this information:

| Property | Required? | Description |
|--------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Display Name | mandatory | Defines a system-wide unique name for the point. |
| Object Name | mandatory | Provides the point address, which must be a unique within the device. |
| Object Type | mandatory | Contains a decimal value to represent engineering units as defined by the BACnet specification (square-meters, volts, joules, hertz, etc.). The utility supports only unit codes specified in the BACnet specification. The utility ignores Object Type values that are outside of the BACnet specification because it is not possible to determine what appropriate |

| Property | Required? | Description |
|------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | Niagara object type to use to represent the new Object Type given only an Object Name . |
| Device Object Instance | mandatory | Identifies the parent device that contains this point. |
| Object Instance | mandatory | Defines a decimal number with which to identify the object. Object Type and Object Instance together provide a unique identifier in each point. |
| Description | optional | Provides important additional information about the object and its function. |
| Present Value | optional | Displays the current value for the point. The default for this value corresponds to a BACnet control point's Fallback value. |
| Min | optional | Displays the Present Value specified as a MIN facets value on the point. It can be obtained from or written to the Present Value property. |
| Max | optional | Displays the Present Value specified as a MAX facets value on the point. It can be obtained from or written to the Present Value property. |
| Settable | optional | Indicates if the writable Present Value property is controlled by an automated process (device, program) making it a read-only point representation, or if it can be set by a client, which makes it a writable point. All BACnet points are writable unless Settable is included in the EDE master configuration file and its value is false . |
| Support COV | optional | Defines a Boolean facets value on the point. This value controls if the system supports COV or not. |
| High Limit | optional | Defines a numeric facets value on the point—the upper alarm limit. Limits imply conditional alarming. The EDE does not include alarm classes. The utility passes this information to Niagara as facet information only. |
| Low Limit | optional | Defines a numeric facets value on the point—the lower alarm limit. Limits imply conditional alarming. The EDE does not include alarm classes. The utility passes this information to Niagara as facet information only. |
| State Text | optional | Provides information as an enumeration facets value on the point. It is valid for binary and multistate objects only. |
| Vendor Address | optional | Provides a string facets value on the point. It may identify an address used in the server device, such as a terminal number. This information provides an internal data-point identification or reference. |

EDE state texts support file

This file provides a reference number and lists enumeration state texts for multi-state and binary objects.

This CSV file includes "State-Texts" in its name.

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 |
| 14 | 13 | OCCUPIED | UNOCCUPIED | BYPASS | STANDBY |
| 15 | 14 | Standby | Duty | | AUTO |
| | 15 | OFF | ON | | |

You cannot mix multi-state and binary texts under the same reference number, that is, in the same row.

Each row consists of **Inactive-Text**, **Active-Text** or **Text 1** and so on. An empty column indicates unused or not-supported states.

Configuring a station for EDE

Before you can use the EDE configuration utility to discover devices and points offline, the EDE tool must be enabled in the station. Both the controller and Supervisor stations require this configuration.

Prerequisites: The Supervisor or controller station is running and connected using Workbench.

Step 1 Click **Tools**→**Bacnet EDE**.

Bacnet EDE appears as a node under the **My Tools** folder in the station. This component has a single property, **Store Time**, which defaults to 30 minutes.

Step 2 If needed, adjust the time and click **Save**.

Exporting controller device and point data

A remote controller connected to the BACnet network functions as a BACnet Controller Device (BCD). You would export device and point data from this controller's station when upgrading the controller or preparing an EDE master configuration file to use to configure the Supervisor station.

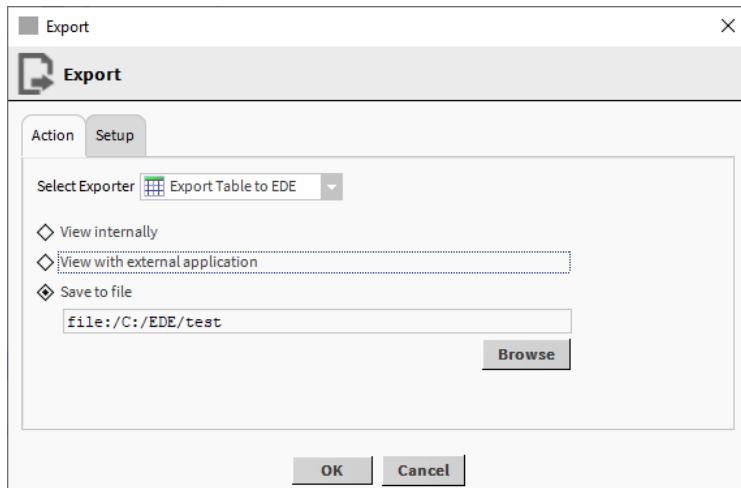
Prerequisites: The controller station is running and all the devices and points are configured properly.

The export function can collate all exported files into a single EDE master configuration file or you may prepare this file manually. Later, offline, you import the EDE master configuration file into the new controller station or the Supervisor station, BACnet AWS (Advanced Work Station).

Step 1 Open the station and navigate to **Config**→**Drivers**→**BacnetNetwork**→**Local Device**.

Step 2 Right-click the **Export Table** and click **Export**.

The **Export** window opens.

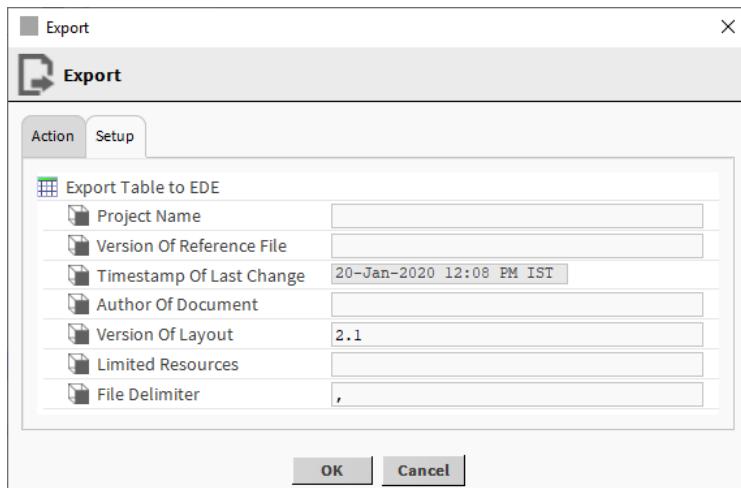


Step 3 Select **Save to file**, click **Browse** to a secure location in which to save the EDE master configuration file, give the file a name and click **Save**.

CAUTION: Choose a secure location for the exported EDE master configuration file(s) where it/they cannot be modified by unknown users.

The utility exports the EDE master configuration file(s). There is no option to export the state texts file, which must be prepared manually.

Step 4 Click the **Setup** tab and enter the following properties.



Step 5 To export the data, click **OK**.

The station exports the file(s) to the desired location.

If you exported the EDE master configuration file(s) for the purpose of upgrading the controller, you may replace the controller now, then import data from the EDE master configuration file(s) to set up the station database. Your BACnet devices come online and work as expected. Once online, the driver updates the point values in the Supervisor station overriding the point read values that came from the EDE master configuration file.

Validation criteria for data to be discovered

You can create EDE comma-delimited EDE master configuration files by exporting them from another station (perhaps a station running older software that you are upgrading) or by using Microsoft™ Excel™ or Notepad to create them from scratch.

Before you discover devices and points from your EDE configuration files, validate the data contained in the files for accuracy. Following are some validation criteria to give you an idea about what to check before importing an EDE master configuration file into a station database.

- string value and string length: make sure all names are spelled the way you want them to appear in the interface and the string value is large enough to meet your needs.
- integer/float range: make sure this value is large enough to meet your needs.
- enum range: make sure that your enum values are spelled correctly and that you have one for each important characteristic.

Discovering device data offline

Offline configuration does not require a connection to the network. This procedure uses the utility to discovery device data from the EDE master configuration file and add them to the database.

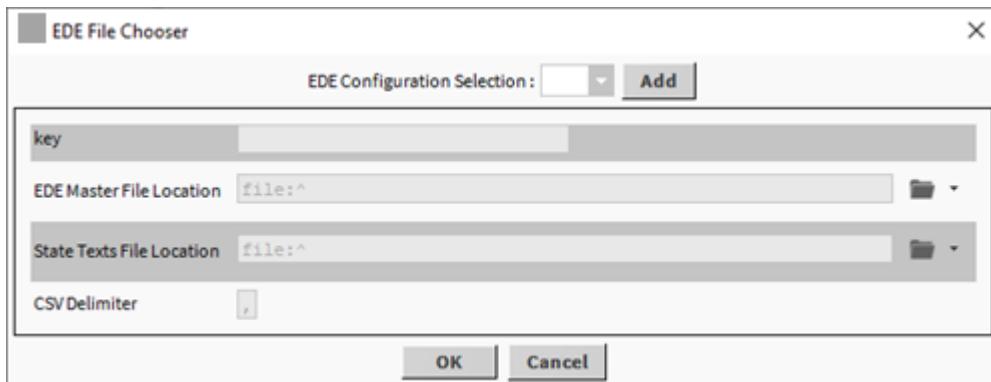
Prerequisites: You have the EDE master configuration file that was exported from the remote station as well as a state text file, which you created manually. The station is connected to the BACnet network and running Workbench.

Step 1 To discover devices, expand **Config→Drivers** and double-click the **BacnetNetwork**.

The **EDE Bacnet Device Manager** opens.

Step 2 To continue, click **Discover**.

The discovery job opens the **EDE File Chooser** window.



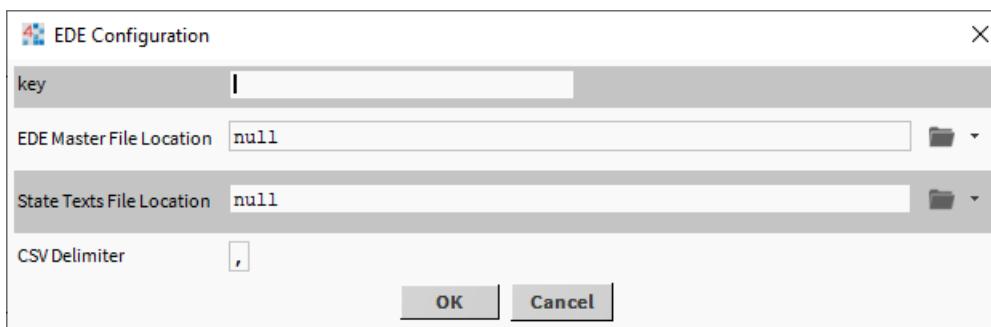
A key serves as an ID associated with the EDE master configuration file and state texts file.

Step 3 To populate the location properties, select the key from the **EDE Configuration Selection** drop-down list.

The **EDE File Chooser** automatically populates the location properties with file paths.

Step 4 If this set of configuration files does not have a key yet, click **Add**.

The **EDE Configuration** window opens with the **key** property available for editing.



Step 5 Enter a **key** value, use the browser folder icons to the right of the location properties to locate the file paths and click **OK**.

The next time you use the EDE configuration utility with this set of EDE configuration files, you will not need to browse to the file locations.

Step 6 With all properties configured, click **OK**.

The discovery job finds all the devices present in the EDE configuration files.

Step 7 To add these data to the database, select one or more devices in the **Discovered** pane and click **Add**.

The devices move from the **Discovered** pane to the **Database** pane.

Discovering point data offline

You discover point data from the EDE master configuration file and add them to the database. Once a controller or Supervisor station is on line, running a second discovery job matches the point records in the database with the actual points.

Prerequisites: You have the EDE master configuration file that was exported from the remote station as well as a state text file, which you created manually. The station is connected to the BACnet network and running Workbench.

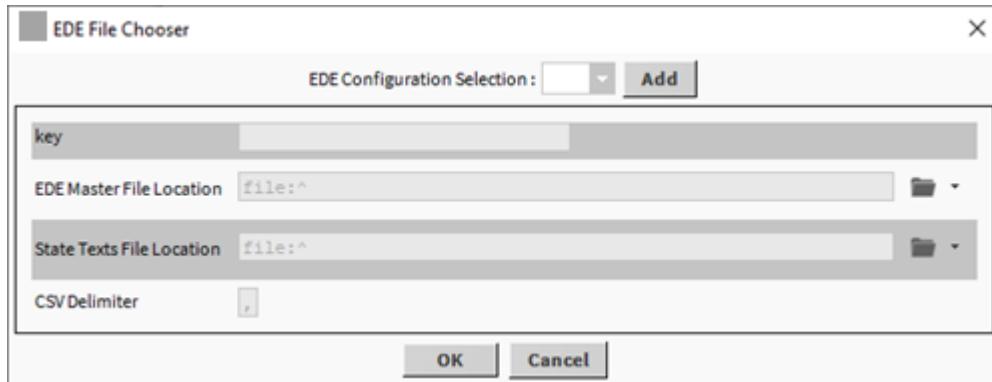
A single EDE master configuration file can have many points for each device. Inside a specific points folder, discovery finds only points that belong to the current device even though the EDE master configuration file may contain many points that belong to other devices.

Step 1 To discover points expand **Config→Drivers→BacnetNetwork** and double-click the **Points** folder.

The **Points Manager** view opens.

Step 2 To continue, click **Discover**.

The discovery job opens the **EDE File Chooser** window.



A key serves as an ID associated with the EDE master configuration file and state texts file.

Step 3 To populate the location properties, select the key from the **EDE Configuration Selection** drop-down list.

The **EDE File Chooser** automatically populates the location properties with file paths.

Step 4 If this set of configuration files does not have a key yet, click **Add**.

The **EDE Configuration** window opens with the **key** property available for editing.



Step 5 Enter a **key** value, use the browser folder icons to the right of the location properties to locate the file paths and click **OK**.

The next time you use the EDE configuration utility with this set of EDE configuration files, you will not need to browse to the file locations.

Step 6 With all properties configured, click **OK**.

The discovery job finds all the analog, binary and multistate points present in the EDE configuration files.

Step 7 To add these point data to the database, select the points in the **Discovered** pane and click **Add**.

The point data move from the **Discovered** pane to the **Database** pane.

EDE troubleshooting

When the station initiates an EDE discovery job, the JobService records it in the Job Log. You can review this log and use it in troubleshooting.

To see the progress of the EDE discovery job, click on the **Job Log** control near the top of the manager view. This job displays only objects that failed to discover. It also specifies the invalid property due to which the discovery job failed.

Figure 4 The Job Log shows failed device and point discovery

| Status | Timestamp | Message | Details |
|---------|------------------------|------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
| Message | 18-Feb-20 11:14 AM IST | Preparing to iterate through EDE points. | |
| Failed | 18-Feb-20 11:14 AM IST | Failed to parse line : multiStateValue_3,1125,Points.EnumW | javax.baja.sys.BajaRuntimeException: Value is not valid for EDE property type "State Text Range" |
| Failed | 18-Feb-20 11:14 AM IST | Job Failed | javax.baja.sys.BajaRuntimeException: Failed to parse all lines in EDE file. |

Stack: Unresolved device address

Issue: Using offline discovery to add BACnet points into the station database while the BACnet device is offline creates a number of `BacnetException:Stack:Unresolved device address` debug messages in the **Application Director** view. The status flag type and status flag colors indicate the condition of the point.

In these examples, offline discovery attempted to add BACnet proxy points to the station database. The BACnet device is initially offline and then afterwards brought online. Inspection of the **Application Director** view shows a number of messages, which occurred during the discovery job while the `BacnetDevice` was offline.

Figure 5 Proxy points added while the BacnetDevice is offline

| Database | | | | | | |
|----------------------|---------------------------|---------------|---------------|-------|--------------|----------|
| Name | Out | Object ID | Property ID | Index | Read | Write |
| AI_Eagle1_Onboard_03 | 0.0 °C {down,stale} @ def | analogInput:3 | Present Value | -1 | unsubscribed | Writable |
| AI_Eagle1_Onboard_04 | 0.0 °C {down,stale} @ def | analogInput:4 | Present Value | -1 | unsubscribed | Writable |
| AI_Eagle1_Onboard_05 | 0.0 °C {down,stale} @ def | analogInput:5 | Present Value | -1 | unsubscribed | 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 |
| MESSAGE [17:27:06 15-Jul-14 BST] [bacnet.client] BacnetException uploading propertyId 304 in object device:6654: Stack:Unresolved device address |

The BACnet proxy points report `{down}` indicating driver communications to the parent device are currently lost based upon the device status (Monitor) configuration for the network.

Figure 6 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 |

When driver communications to the parent device are subsequently established, based upon the device status (Monitor) configuration for the network, the BACnet proxy points report `{ok}`

Figure 7 Proxy points added while the BacnetDevice is initially offline

| Database | | | | | | |
|----------------------|----------------------------|---------------|---------------|-------|---------------------------------|----------|
| Name | Out | Object ID | Property ID | Index | Read | Write |
| AI_Eagle1_Onboard_03 | 0.0 °C {fault,stale} @ def | analogInput:3 | Present Value | -1 | Stack:Unresolved device address | Writable |
| AI_Eagle1_Onboard_04 | 0.0 °C {fault,stale} @ def | analogInput:4 | Present Value | -1 | Stack:Unresolved device address | Writable |
| AI_Eagle1_Onboard_05 | 0.0 °C {fault,stale} @ def | 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 |
| MESSAGE [17:27:06 15-Jul-14 BST] [bacnet.client] BacnetException uploading propertyId 304 in object device:6654: Stack:Unresolved device address |

The BACnet proxy points report `{fault}`, which typically indicates a configuration error or a native fault in the parent BACnet device.

Figure 8 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 |

| |
|-----------------------------------------------------------------------------------------------------------------------------------------------|
| NAME [17:27:06 15-Jul-14 BST] [bacnet.client] BacnetException uploading propertyId 119 in object device:6654: Stack:Unresolved device address |
| NAME [17:27:06 15-Jul-14 BST] [bacnet.client] BacnetException uploading propertyId 116 in object device:6654: Stack:Unresolved device address |
| NAME [17:27:06 15-Jul-14 BST] [bacnet.client] BacnetException uploading propertyId 95 in object device:6654: Stack:Unresolved device address |
| NAME [17:27:06 15-Jul-14 BST] [bacnet.client] BacnetException uploading propertyId 341 in object device:6654: Stack:Unresolved device address |
| NAME [17:27:06 15-Jul-14 BST] [bacnet.client] BacnetException uploading propertyId 340 in object device:6654: Stack:Unresolved device address |
| NAME [17:27:06 15-Jul-14 BST] [bacnet.client] BacnetException uploading propertyId 339 in object device:6654: Stack:Unresolved device address |
| NAME [17:27:06 15-Jul-14 BST] [bacnet.client] BacnetException uploading propertyId 338 in object device:6654: Stack:Unresolved device address |
| NAME [17:27:06 15-Jul-14 BST] [bacnet.client] BacnetException uploading propertyId 206 in object device:6654: Stack:Unresolved device address |
| NAME [17:27:06 15-Jul-14 BST] [bacnet.client] BacnetException uploading propertyId 304 in object device:6654: Stack:Unresolved device address |

When driver communications to the parent BACnet device are subsequently established, based upon the device status (Monitor) configuration for the network, the BACnet proxy points report `{ok}`. When you refresh the database view, the system updates the Read: Stack:Unresolved device address.

Proxy point overwrite

The station may discover BACnet devices and proxy points from online devices and add them to its database after offline discovery from the EDE configuration files.

Online discovery that overwrites offline-discovered proxy point data may break any previously made object or graphical link.

Chapter 5 Device management

Topics covered in this chapter

- ◆ Searching for a specific device
- ◆ Adding a time synchronization recipient
- ◆ Synchronizing time across the network
- ◆ Data gathering from devices
- ◆ Performance improvements with virtual components
- ◆ Importing a schedule from a remote station

Once the BACnet network is configured, devices and proxy points added to the database, network communication may require adjustments from time to time, including synchronizing time and configuring tuning policies. You can search for devices on the network and set up virtual components, which can improve network performance.

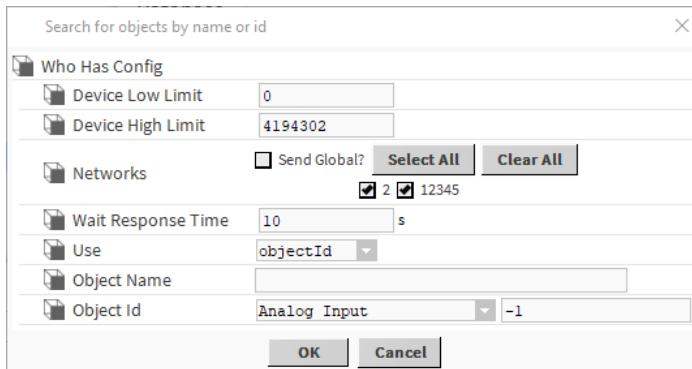
Searching for a specific device

You use the **Who Has** service to locate objects by name or object ID (type and instance number).

Prerequisites: The **Bacnet Device Manager** view is open.

Step 1 Go to the menu bar and select **Manager→Who Has**.

The **Search for objects by name or id** window opens.

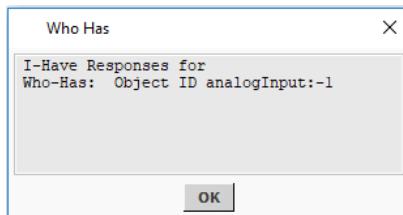


As in the device discover window, you can set device low and high limits, as well as select which BACnet networks (by number) will receive the Who-Has broadcast (or all networks).

Step 2 Set the **Device Low Limit** and **Device High Limit** properties.

Step 3 In **Networks** property, select the BACnet network, which you want to receive the Who-Has broadcast (or you can select All Networks), and click **OK**.

The system starts a BACnet who-has job. When the job completes, the result displays in the **Who Has** window.



Adding a time synchronization recipient

The driver sends periodic BACnet time synchronization messages from the station to all defined time synchronization recipients (devices) or to UTC time synchronization recipients (devices), at the interval specified in the properties. This procedure documents how to add these recipients. This adds controllers to the list of devices whose time can be reset.

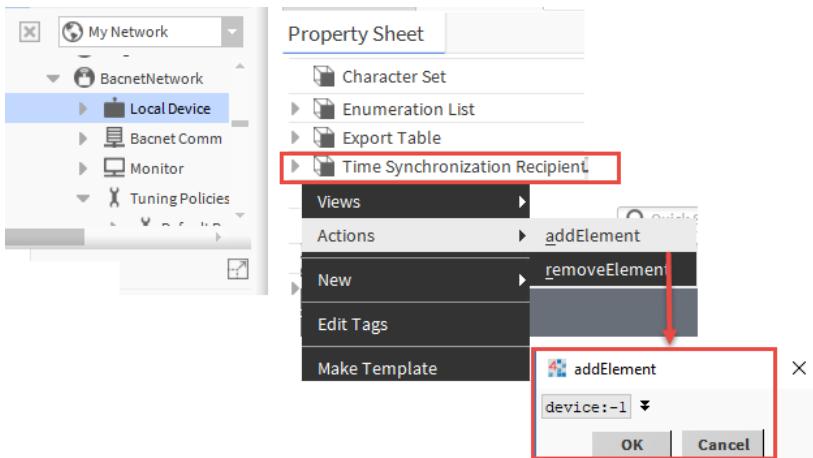
Step 1 Expand **Config→Drivers→BacnetNetwork** and double-click **Local Device**.

The **Local Device Property Sheet** opens.

Step 2 Scroll down to **Time Synchronization Recipients**.

Step 3 Right-click this property and click **Actions→addElement**.

The **addElement** window opens.



Step 4 Do one of the following:

- To add a device as the recipient, select **Device** from the left-hand drop-down box, select the name of the device in the middle drop-down box, fill in the device ID (**Object Id**), and click **OK**.



- To add the address of another location, select **Address** from the left-hand drop-down box (the window changes), fill in the **Network Number** and **MAC Address**, and click **OK**.

The driver adds the recipient name under **Time Synchronization Recipients**.

Synchronizing time across the network

This procedure sends a manual (one-time) time synchronization message from the Supervisor station to all BACnet devices on the network. This resets the time in all host controllers to the same time.

Step 1 Double-click the BACnet network node in the Nav tree.

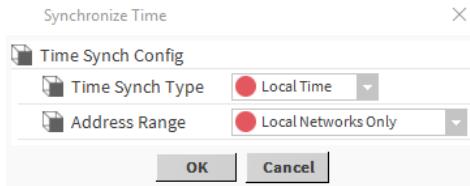
The **Bacnet Device Manager** opens.

Step 2 Click **TSynch**.

The driver asks you to confirm this action as it will send the current station time to all controllers in the network.

Step 3 To continue, click **Yes**.

The **Synchronize Time** window opens.



Step 4 Select the your local time or UTC (Coordinated Universal Time), select local networks or all networks, and click **OK**.

The driver broadcasts a time synchronization request to the network. This is a one-time message.

To schedule a regular time synchronization request from this station, configure the Local Bacnet Device properties.

NOTE: LocalDevice configuration applies to a **BACnet AWS Supervisor** (BacnetAwsNetwork) or **BACnet OWS Supervisor** (BacnetOwsNetwork).

Data gathering from devices

Two mechanisms import data from BACnet devices: device polling and, for devices that support COV (Change of Value), a COV subscription, which is more efficient than polling for data. Each network's tuning policies control these mechanisms.

Tuning policies configure how a network evaluates write requests (for example, requests to write to writable proxy points) and the acceptable freshness of read requests. They associate with one of three poll rates in a network's Poll Service.

The BACnet driver is unique among drivers in that it supports multiple communication port types: Ip port, Ethernet port, Secure Connect port, and MS/TP port. All port types can coexist under one network-level component (**BacnetNetwork**). The driver uses a separate BacnetMultiPoll (**Poll Service**) under **Bacnet-Comm→Network** for each port.

Device polling

A multi-thread capability exists in all BACnet poll services where point polling might involve a BACnet router (say, going between B/IP and slower MS/TP devices). This capability is sometimes needed even in a (typically speedy) Ip port or Ethernet port link layer.

Unique to BACnet polling is the interplay with BACnet COV services, where some server devices may offer COV for point updates—a desirable alternative to polling. This affects tuning policy configurations, and how you should assign tuning policies for points.

By default, any BACnet proxy point polls support only one configured polling property. Thus, if you specify `presentValue`, which displays all of a point's out values, no native object status or currently-active level (`bacn` if a commandable object) affects the point display. However, as needed you can add either or both of these extra polled properties, and/or others, by editing the facets of your proxy points.

COV subscription

For devices that support COV, the driver offers client support for BACnet COV services, which receive notifications when proxy point data change.

Improving performance by adding a tuning policy

Tuning policies configure how a network evaluates write requests (for example, requests to write to writable proxy points) and the acceptable freshness of read requests. They associate with one of three poll rates in a network's Poll Service. A BACnet network benefits from multiple tuning policies, especially if you enable multiple BACnet network ports.

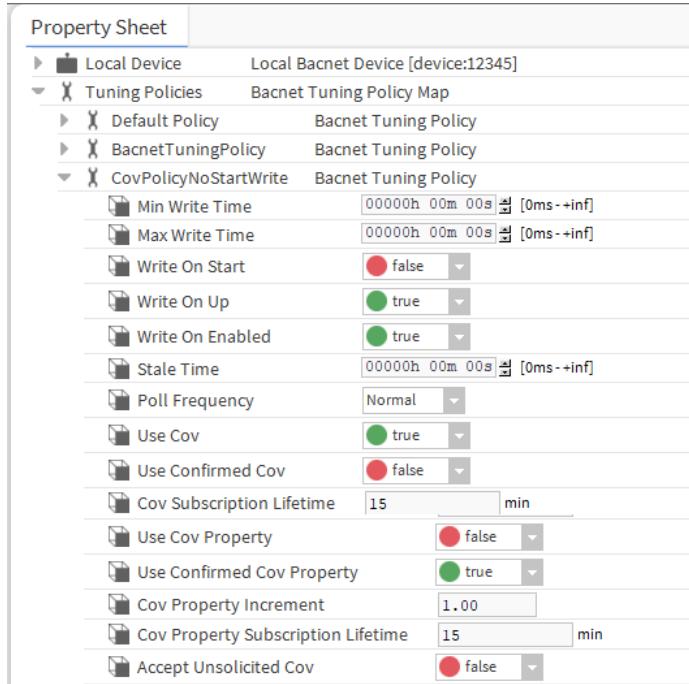
Prerequisites: An appropriate palette is open.

Configuring BACnet proxy points to reference different tuning policies under different networks (often with vastly different performance capabilities) allows you to improve overall performance. For example, points under a BACnet MS/TP network would likely use different tuning policies than those under a BACnet IP network.

Step 1 Do one of the following:

- Right-click an existing policy and choose **Duplicate** from the shortcut menu.
- Drag a **BacnetTuningPolicy** from the palette to the network's **Tuning Policies** container.

Step 2 Give the policy a name.



The screen capture shows the expanded view of one of five different tuning policies.

CovPolicyNoStartWrite configures COV (Change of Value) data collection. If you have client **BacnetDevices** that provide support for COV notifications, you can add at least one tuning policy that has its **Use Cov** property set to `true` (possibly with other adjustments).

Step 3 Double-click the policy in the Nav tree, edit policy properties and click **Save**.

Subscribing proxy points for COV data collection

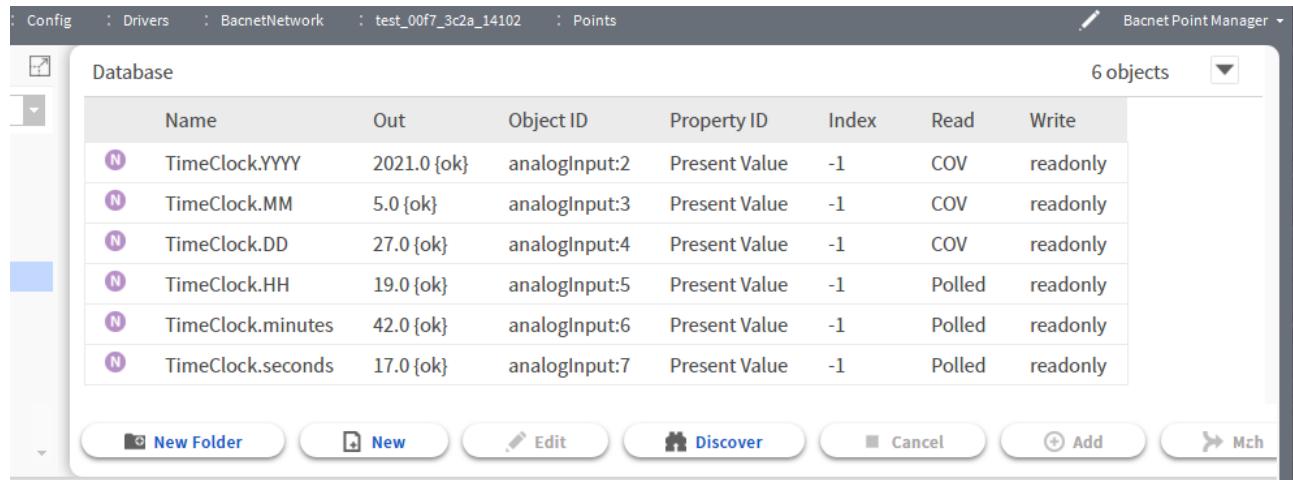
The driver offers client support for BACnet COV services for devices that support COV. Receiving COV (Change-of Value) notifications for proxy point data is more efficient than polling for data.

Prerequisites: The device(s) with the points you are subscribing support COV subscriptions.

Step 1 Under the **BacnetNetwork** you need to add at least one additional **TuningPolicy** (apart from the **DefaultPolicy**).

Step 2 Set its **Use Cov** property to `true`.

Step 3 Assign Bacnet proxy points to the CovPolicy under a COV-capable **BacnetDevice**.

Figure 9 Example proxy points subscribed for COV


The screenshot shows the Bacnet Point Manager interface. The title bar reads 'Bacnet Point Manager'. The main area is titled 'Database' and contains a table with the following data:

| Name | Out | Object ID | Property ID | Index | Read | Write |
|-------------------|-------------|---------------|---------------|-------|--------|----------|
| TimeClock.YYYY | 2021.0 {ok} | analogInput:2 | Present Value | -1 | COV | readonly |
| TimeClock.MM | 5.0 {ok} | analogInput:3 | Present Value | -1 | COV | readonly |
| TimeClock.DD | 27.0 {ok} | analogInput:4 | Present Value | -1 | COV | readonly |
| TimeClock.HH | 19.0 {ok} | analogInput:5 | Present Value | -1 | Polled | readonly |
| TimeClock.minutes | 42.0 {ok} | analogInput:6 | Present Value | -1 | Polled | readonly |
| TimeClock.seconds | 17.0 {ok} | analogInput:7 | Present Value | -1 | Polled | readonly |

Below the table are several buttons: New Folder, New, Edit, Discover, Cancel, Add, and Match.

If successful, such a proxy point list in the **Bacnet Point Manager** Read column shows COV instead of Polled. If the COV subscription is unsuccessful, the point falls back to polling and shows Polled, like the points configured with a polling **TuningPolicy**.

Collecting device history

The **Trend Logs** device extension under the **BacnetDevice** imports BACnet **Trend Log** objects from the device to the station as Framework histories. For general information on this device extension, refer to the *Niagara Drivers Guide*.

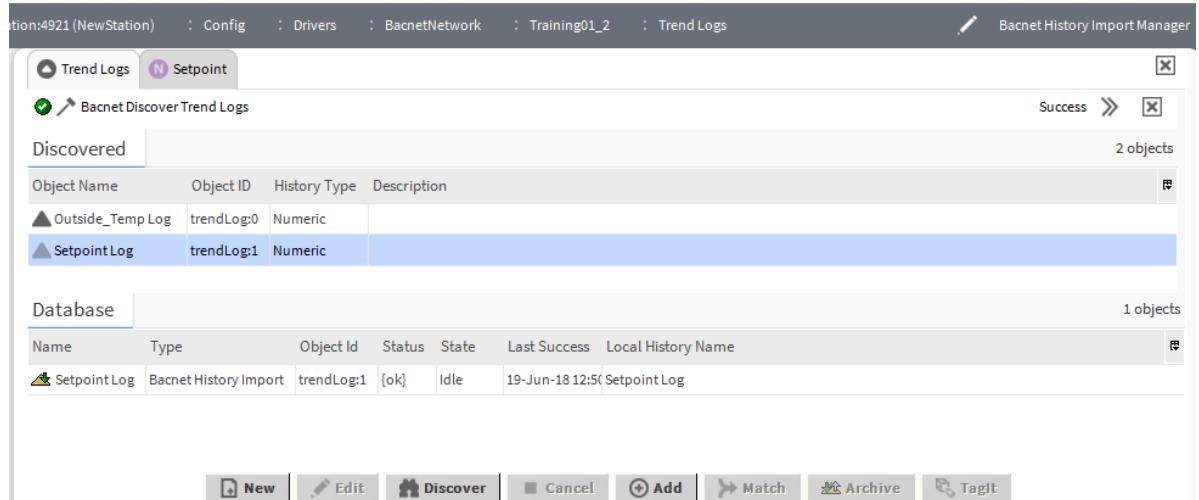
Prerequisites: Device history exists.

Step 1 Double-click the **BacnetNetwork**→**BacnetDevice**→**Trend Logs** node in the Nav tree.

The **Bacnet History Import Manager** view opens.

Step 2 Click the **Discover** button.

The driver discovers all trend logs (histories).



The screenshot shows the Bacnet History Import Manager interface. The title bar reads 'Bacnet History Import Manager'. The main area has two panes: 'Discovered' and 'Database'.

Discovered:

| Object Name | Object ID | History Type | Description |
|------------------|------------|--------------|-------------|
| Outside_Temp Log | trendLog:0 | Numeric | |
| Setpoint Log | trendLog:1 | Numeric | |

Database:

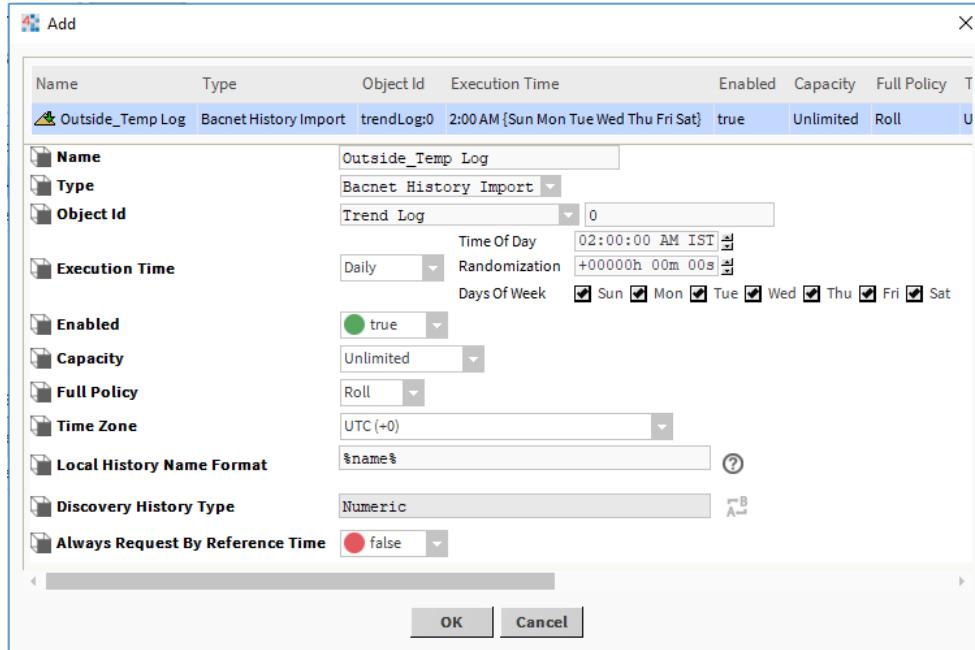
| Name | Type | Object Id | Status | State | Last Success | Local History Name |
|--------------|-----------------------|------------|--------|-------|-----------------|--------------------|
| Setpoint Log | Bacnet History Import | trendLog:1 | {ok} | Idle | 19-Jun-18 12:56 | Setpoint Log |

At the bottom are buttons: New, Edit, Discover, Cancel, Add, Match, Archive, and Tagit.

The screen capture shows three trend logs already imported into the station.

Step 3 To import trend logs or trend log multiples, select them in the **Discovered** pane and click **Add**.

The **Add** window opens.



This window shows some of a Bacnet History Import component's properties.

After adding, you can access additional Bacnet History Import properties from the **Property Sheet** of any **BacnetHistoryImport** and **BacnetTrendLogMultipleImport** components.

Step 4 Edit properties and click **OK**.

Step 5 If you are using the **Bacnet Trend Multiple View**, double-click a record.

The **Record** window opens.

This example shows a trend log multiple with three trend items: two properties of object AI1 (Present_Value and Status_Flags) and one property of object AI100 (Present_Value).

| Timestamp | Trend Flags | Status | Sequence Number | Log Event | Value |
|--------------------------|-------------|--------|-----------------|-----------|-------|
| 19-Jun-18 7:15:15 AM GMT | {} | {ok} | 1202 | | 19.34 |
| 19-Jun-18 7:15:16 AM GMT | {} | {ok} | 1203 | | 19.68 |
| 19-Jun-18 7:15:17 AM GMT | {} | {ok} | 1204 | | 20.02 |
| 19-Jun-18 7:15:18 AM GMT | {} | | | | |
| 19-Jun-18 7:15:19 AM GMT | {} | | | | |
| 19-Jun-18 7:15:20 AM GMT | {} | | | | |
| 19-Jun-18 7:15:21 AM GMT | {} | | | | |
| 19-Jun-18 7:15:22 AM GMT | {} | | | | |
| 19-Jun-18 7:15:23 AM GMT | {} | | | | |
| 19-Jun-18 7:15:24 AM GMT | {} | | | | |

Record dialog box details:

- Timestamp:** 19-Jun-18 12:45 PM IST
- Trend Flags:** {}
- Status:** {ok}
- Sequence Number:** 1205
- Log Event:**
- Value:** 20.36

The separate histories created by importing a trend log multiple objects are also recognizable in the station's Nav tree (below).

| Name | Type | Object Id | Status | State | Last Success | Local History Name |
|------------------|-----------------------|-----------|--------|-------|------------------------|--------------------|
| Setpoint Log | Bacnet History Import | trendLog1 | {ok} | Idle | 19-Jun-18 12:53 PM IST | Setpoint Log |
| Outside_Temp Log | Bacnet History Import | trendLog0 | {ok} | Idle | 19-Jun-18 12:53 PM IST | Outside_Temp Log |

By default, the name of each history associated with a trend log multiple import uses the format of TrendLogMultipleName_objectName_propertyName, for example TM1_AI1_presentValue.

Performance improvements with virtual components

These components can improve the performance of embedded controllers with limited resources.

Most drivers contain a device network, devices, and proxy points (control points with proxy extensions). Linking proxy points to control logic makes it possible to monitor and control building assets. The problem is that every piece of driver data used to visualize and configure a sophisticated model requires that a persistent component (that is, a proxy point) exist somewhere in a station's component space. This overhead becomes prohibitive when running on small embedded platforms with limited memory.

Virtual components solve this problem because they exist in a station database only as long as needed. These transient, on-demand entities reside in their own virtual space with their own virtual Ords (Object Resolution Descriptors), which provide the information needed to create the components at runtime. A virtual gateway links each virtual component space to an actual component space in a one-to-one relationship. A station may have multiple virtual gateways and virtual component spaces.

Each BACnet device includes a child component that serves as its virtual gateway () providing access to the device's virtual component space. The device's virtual points are under this gateway. There are two primary uses for BACnet virtual points:

- To poll values in multiple Px view bindings. Only the Ords persist (using virtual syntax) in the Px widget bindings (including the BacnetVirtualGateway within the Ord). With replicated device applications, this allows a station to graphically monitor many more devices than it could monitor otherwise.
- To efficiently review and adjust device property values. Configuring the Workbench property sheets of virtual components can speed system setup. Without virtual components you would need to create a persisted configuration-type object for each device, change object properties, and then delete the persisted object.

You cannot link to virtual components and they do not support point extensions (alarm, history). Links and extensions require BACnet proxy points that persist in the station database.

Virtual Ord syntax

The syntax for a virtual Ord uses a specialized form as follows:

<ord to VirtualGateway>|virtual:/objectType_Instance[A] /propertyName[B]

where:

- <ord to VirtualGateway> identifies the path to the virtual gateway.

- `objectType_Instance` is the object type instance.
- [A] may be a semicolon-separated list of modifiers. Refer to [\[A\] Modifiers, page 64](#).
- `propertyName` identifies the name of the property.
- [B] may be a semicolon-separated list of modifiers. Refer to [\[B\] Modifiers, page 64](#)

For Example: `ip:xx.xx.xx.xx|foxs:|station:|slot:/Drivers/BacnetNetwork/StationName/virtual|virtual:/device_deviceID`

Or in the special case of an arrayed property:

`<ord to VirtualGateway>|virtual:/objectType_Instance/propertyName/elementN`

where N is the property array index.

[A] Modifiers

Valid modifiers include:

- `priority=X`, which indicates the priority at which the point shall be written. Any set write to the polled `present_value` property of a Binary Output object occurs at priority level 10:`station:|slot:/Drivers/BacnetNetwork/BnetDev_99/virtual|virtual:/binaryOutput_2;priority=10/presentValue`
- `policy=X` indicates the name of a tuning policy. Virtual points do not use COV. Polling of this Analog Input object's `present_value` occurs using a Bacnet tuning policy named `FastP:station:|slot:/Drivers/BacnetNetwork/BnetDev_99/virtual|virtual:/analogInput_1;policy=FastP/presentValue`

[B] Modifiers

Valid modifiers include:

- `status=DOPR` (Device Object Property Reference) is encoded numerically, ignore line wrap:
- `objectType_instanceNumber_propertyId_[propertyArrayIndex]_[deviceObjectType_deviceInstance]`, where [options in brackets]
 - `status=0_0_85_8_10` (AI0, PresentValue in Dev10)
 - `status=0_0_111` (AI0, StatusFlags in local device)
 - `status=1_3_87_10` (AO3, PriorityArray[10] in local device)

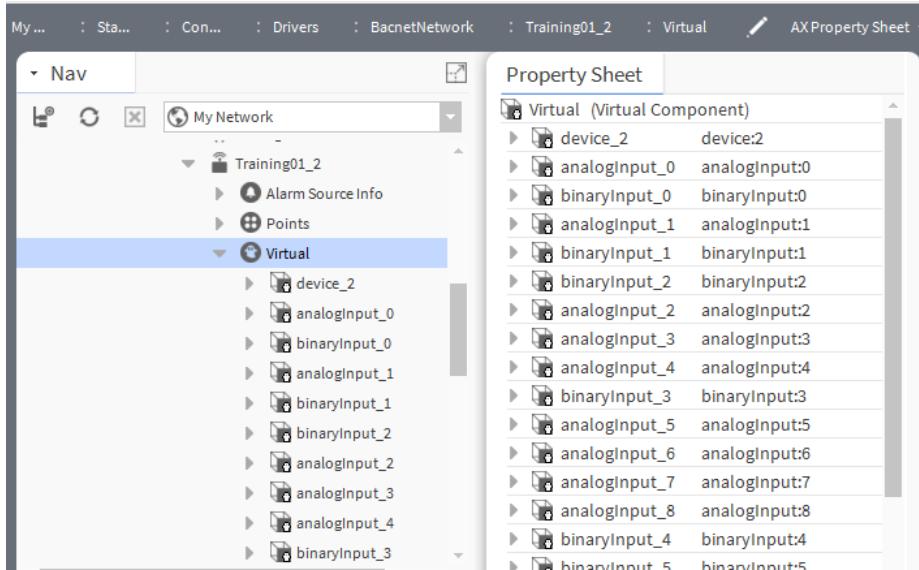
The system merges properties that are BACnetStatusFlags with the bits in the status portion of the BStatusValue. It adds the value of other properties to the facets of the status portion.

Configuring virtual device properties

To speed configuration, you can access a device's properties using the virtual gateway. Unlike the virtual device extension, there is no special view for a virtual gateway.

Step 1 To access a device's virtual property sheet, double-click its virtual gateway or expand it in the Nav tree.

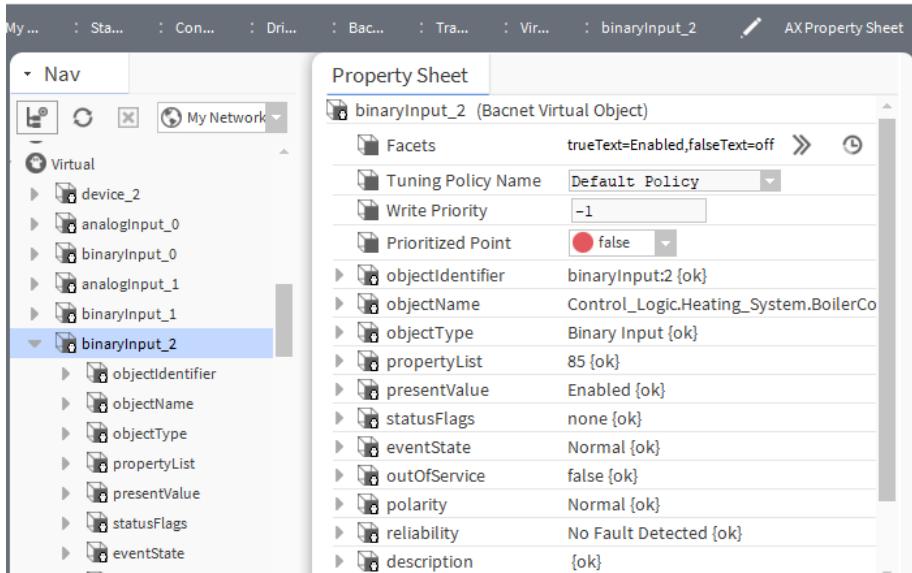
The system makes a call to the device to discover its BACnet objects. This request returns the device's object list. Each object appears as a virtual object (slot) under the gateway.



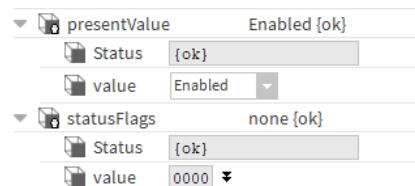
The screen capture shows BACnet virtual gateway functions as a BACnet object list. The gateway lists the virtual objects by object ID, that is, <objectType>_<instanceNumber>, for example `analogInput_2` or `trendLog_1`.

Step 2 Expand any BACnet virtual object to see the object's properties.

Each property is a child of the virtual device.



Properties of a **BacnetVirtualObject** are **BacnetVirtualProperty** components.

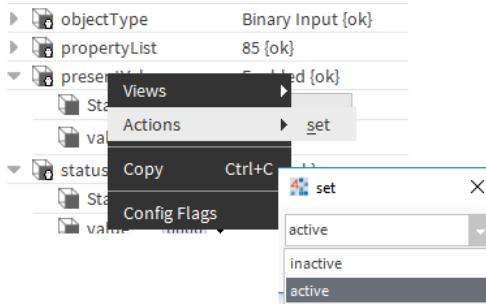


Each **BacnetVirtualProperty** has two properties: **Status** and **Value**. **Status** reflects poll status, but does not reflect any intrinsic BACnet status (such as alarm). Initially, status changes from `stale`

to `ok` upon the first polled read, and typically remains `ok`. Status could possibly change to either `stale` or `down` in certain scenarios.

By default, `value` automatically includes the object's units abbreviation, `state_text`, and so on.

Step 3 Right-click a virtual property.

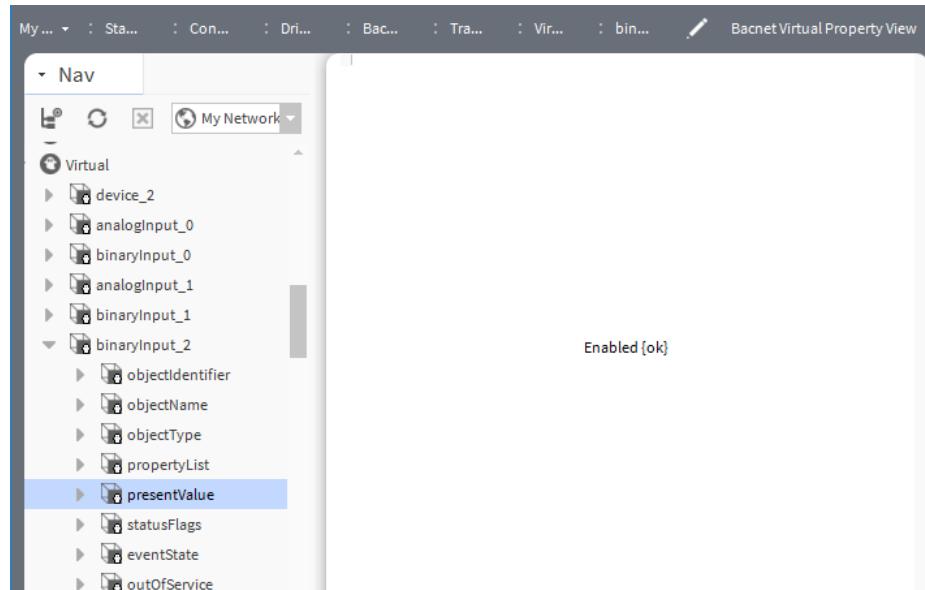


Each **BacnetVirtualProperty** also has a set action for right-click access. This allows you to easily do one time configuration tweaks from the Workbench **Property Sheet** of the virtual property—providing that the remote BACnet device permits writes. This write action exists for every property of the BACnet object, even read-only and status properties. This action is also available on a Px widget bound to any **BacnetVirtualProperty**.

NOTE: You can globally rename the set action (and resultant popup window) to something else by editing the `bacnet lexicon` value for the key named `BacnetVirtualProperty.set`.

Step 4 Double-click a virtual property in the Nav tree.

A virtual property view opens.



Each **BacnetVirtualProperty** has a default **Bacnet Virtual Property View** that shows both value and status. Although it is not an exciting view, it is an option when creating Px bindings and selecting Workbench views, and it also provides a quick preview outside of the Px editor.

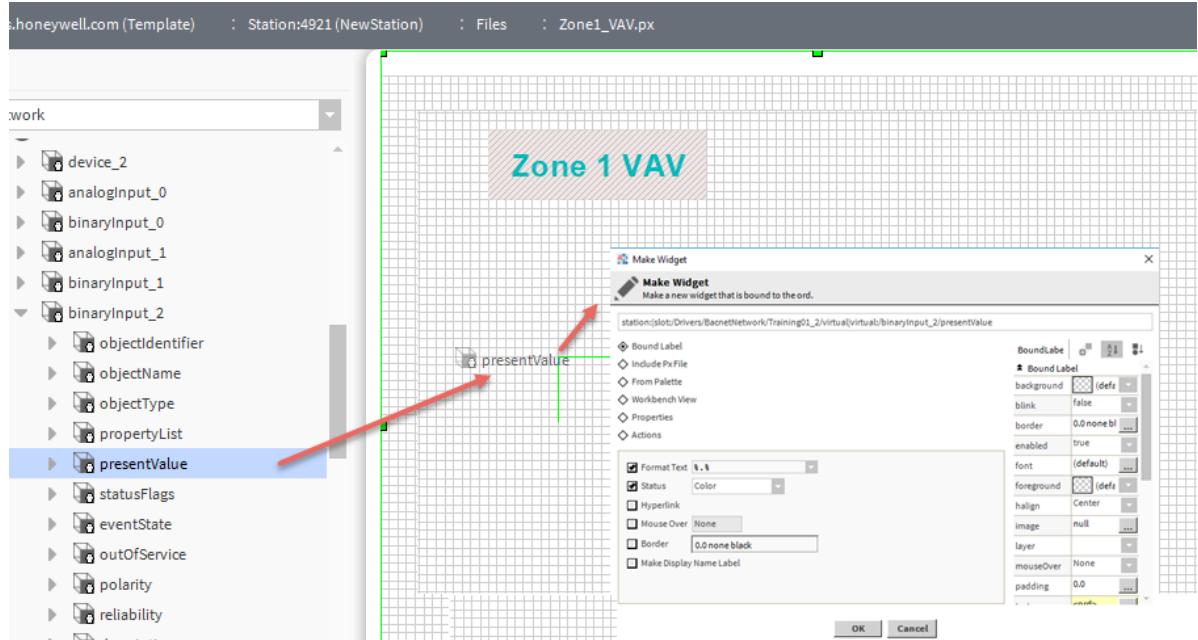
Configuring virtual points in Px views

BACnet virtual points can show real-time values in Px views. This includes an available right-click set action, as well as the poll status (`ok`, `stale`, `down`) of the polled property. A **BacnetVirtualGateway** cannot have

its own Px view, but you can use its child virtual components in Px views for other components, for example on the BACnet device itself, or its points extension, and so on.

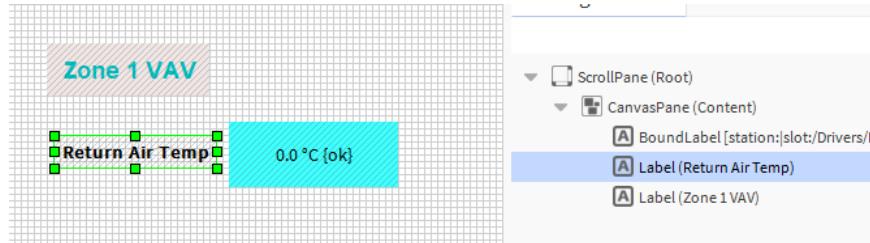
- Step 1** Drag a BACnet virtual component (**BacnetVirtualProperty**) to the Px page, and make your selection in the popup Px **Make Widget** window.

The system automatically resolves the Ord.



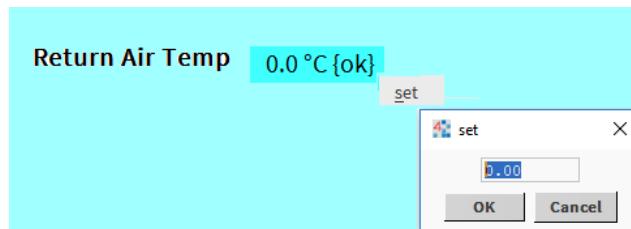
Often you may wish to include the **presentValue** property, as shown here.

- Step 2** In the **Make Widget** window, select either a **Bound label** (as shown) or **Workbench view** for the **Bacnet Virtual Property View**.
- Step 3** If you are using a bound label, edit the accompanying display name label (if that option was selected) to another text string.



The screen captures shows binding to a **BacnetVirtualProperty** in a Px view.

- Step 4** Every **BacnetVirtualProperty** provides an action to write (set is default name), which may be useful in some scenarios.



The screen capture shows the default set action and resulting popup for write to a present_value property.

By default, in the case of a write to the presentValue property of a BACnet object with a priority_array (such as an Analog Output or Binary Output), the write action is issued at priority level 16. You can specify another priority level by editing the ord used in the Px binding (to the Bacnet virtual component).

Importing a schedule from a remote station

You would import a schedule from a remote device to a Supervisor station. Imported schedules and calendars are standard schedule components with a **BacnetScheduleImportExt**. Events in an imported schedule (or calendar) are obtained from that BACnet device, and are read-only. By default, the **Schedules** device extension also contains a **Retry Trigger**, for automatic usage as needed. For details refer to the *Niagara Drivers Guide*.

Prerequisites: You are connected to the controller station that contains the schedule. The **bacnet** palette is open.

Step 1 Drag a **BacnetScheduleImportExt** from the palette to a **BacnetDevice→Schedules** node in the Nav tree.

Step 2 Double-click the **Schedules** node.

The **Bacnet Schedule Import Manager** view opens automatically.

Step 3 Click the **Discover** button.

The system discovers all available schedules in the station.

The default import (read synchronization) from the device's object is **Manual**, which means that to refresh the schedule, you must use the **Import** button in the **Bacnet Schedule Import Manager** view. Alternatively, if you anticipate ongoing changes to the object's configuration, you can set the trigger time of the **BacnetScheduleImportExt** to be either **Daily** or **Interval**.

Step 4 Select the schedule in the **Discovered** pane, do one of the following, and click **OK**.

- Drag the selected schedule to the **Database** pane.
- Click the **Add** button.
- Click the **Import** button.

The driver uploads the object configuration to the read-only schedule component.

If an imported schedule contains a reference to a calendar object in the device, the driver imports the calendar object as well as the schedule. This keeps the schedule self-complete.

Chapter 6 Server management

Topics covered in this chapter

- ◆ Dynamic creation of objects
- ◆ Types of objects to export
- ◆ Discovering components to export as BACnet objects
- ◆ Setting up a point export descriptor
- ◆ Exporting a schedule
- ◆ Exporting a file
- ◆ History export
- ◆ Configuring exported objects for write

Stations function as servers when they export objects at the request of a client station. These include exporting schedules, calendars, files, and histories.

You configure almost all BACnet server operations under the BACnet **Local Device** node of the station's **BacnetNetwork**. The local device's **Export Table** provides special manager views to centrally manage server functions, via a right-click on the **Export Table** node, or in its view selector. In some cases, when operating as a BACnet server, you may need to change properties under the **BacnetNetwork→Bacnet Comm→Server** component.

NOTE: The Batch Search and Replace feature is useful when adding/editing multiple exports.

Dynamic creation of objects

Starting in Niagara 4.7U1, the BACnet driver provides a BACnet create-object service.

BACnet devices may request the creation or deletion of BACnet objects dynamically. Creation may dynamically initialize property values as part of the creation process or a separate write service may populate property values. New objects receive a unique object identifier and are placed in a **dynamicObjects** folder, which is visible in the Workbench Nav tree.

NOTE: Do not manually delete (remove) any dynamically-created descriptor or extension components. Always use a Delete-Object request to remove any dynamically created objects.

Once the create and delete services are enabled, a valid request from a BACnet device may dynamically create or delete five types of descriptors. These properties are valid for initial values:

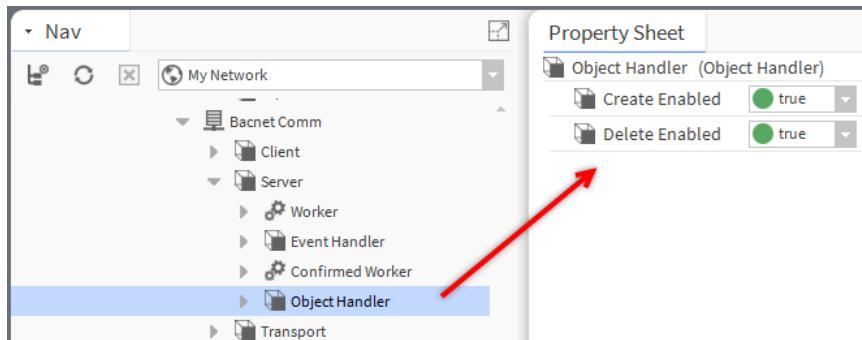
- **Calendar** objects define a list of dates, such as holidays or special events, for scheduling.
- **Event Enrollment** creates an event. Events are changes in the value of any property concerning any object that meets a certain defined criterion. The Event Enrollment object works with information related to the management of BACnet systems event. For example, this object can create: alarms, external alarms, and algorithmic alarms.
- **Notification Class** sends event notifications. If a device has a fault, for example, a notification object may communicate that there is something wrong. Dynamic Notification Class objects can be created and initial values populated simultaneously.
- **Schedules** provide periodic timetables of events that may repeat within a range of dates. The schedule object is categorized into normal days in a week and exception days. Dynamic Schedules are available for creation and initial property value configuration.
- **Trend Logs** support the dynamic creation of remote trending logs, with a unique trending log extension for each data type: numeric, Boolean, enum, and string. A client sends a request to a BACnet controller to create the Trend Log object using either **Object Id** or **Object Type**. The request may or may not include initial property values. Remote trending requests support only the present-value property.

Enabling the Object Handler

Enabling the Object Handler enables the dynamic creation and deletion of objects.

Prerequisites: You are working in Workbench connected to your Supervisor station that has been configured with a BacnetAWSNetwork.

Step 1 Expand **Config→Drivers→BacnetAWSNetwork→Bacnet Comm→Server** and double-click the **Object Handler** node in the Nav tree.



Step 2 Set the **Create Enabled** and the **Delete Enabled** property values to **true**.

This prepares the station for dynamic object creation.

Creating object requests with initial values

You can create five objects and configure them with initial values: Calendar, Event Enrollment, Notification Class, Schedule and Trend Log.

Prerequisites: You are working in Workbench connected to your Supervisor station that has been configured with a BacnetAWSNetwork.

Step 1 Expand **Config→Drivers→BacnetAWSNetwork**.

The **Bacnet Aws Device Manager** opens.

Step 2 If you have not already done so, click the **Discover** button.

The driver runs a discovery job to locate all remote controllers and devices.

Step 3 Expand a device and double-click its **Config** node in the Nav tree.

The **Bacnet Aws Config Manager** view displays.

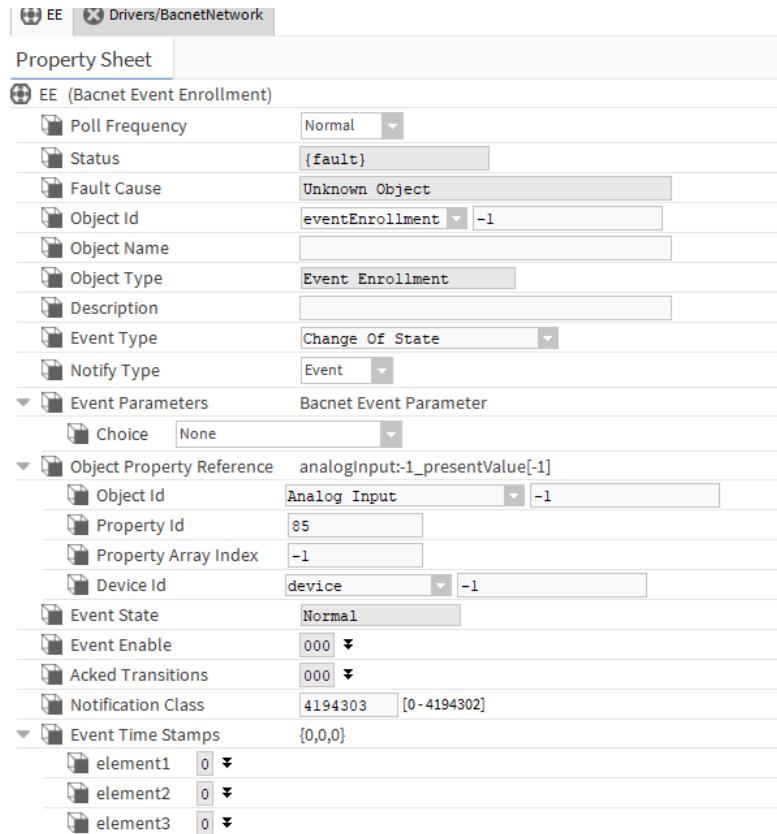
Step 4 To create a dynamic object, open the **bacnet** palette and drag a BACnet object from the **Config** folder in palette to the remote controller's **Config** folder under the **BacnetAWSNetwork**.



NOTE: This **Config** folder is not the Supervisor station's **Config** folder. It is the controller station's **Config** folder, which is under the Supervisor station's **BACnetAwsNetwork**.

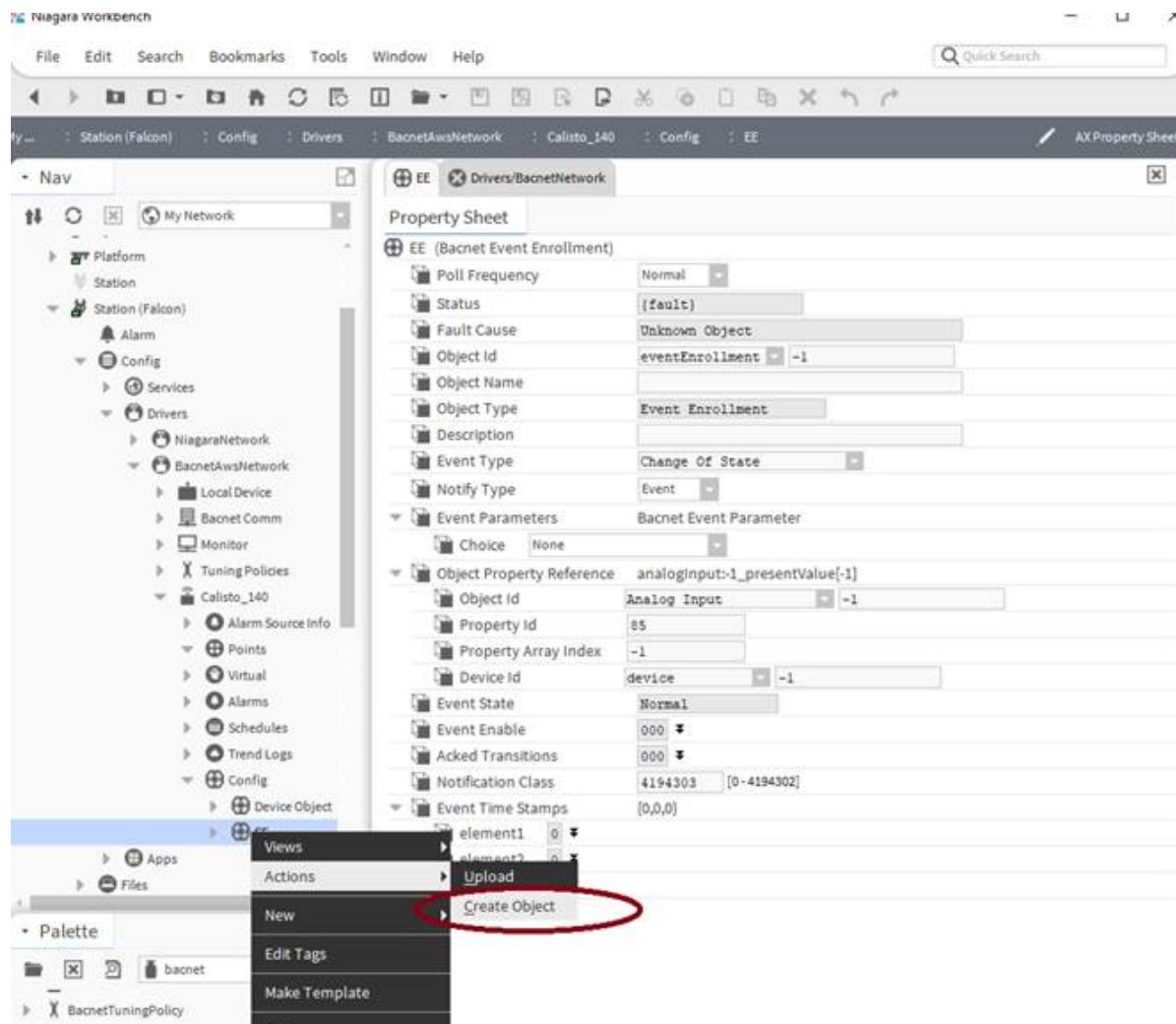
Step 5 Right-click the object in the **Database** pane and click **Views→AX Property Sheet**.

The object's **Property Sheet** opens.

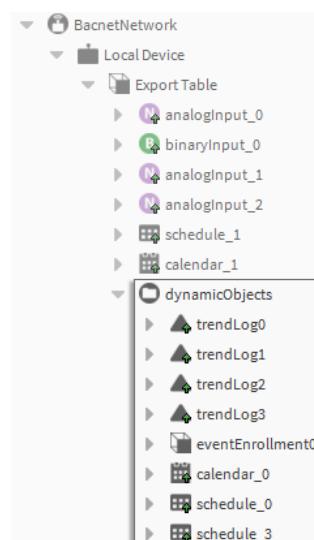


Step 6 Enter initial properties and click **Save**.

Step 7 Right-click the object and click **Actions→Create Object**.



The driver creates each object, giving each a unique object identifier and automatically places the object in a **dynamicObjects** folder, under the **BacnetAwsNetwork→Local Device→Export Table** in the Supervisor station's Nav tree.



NOTE: The **Create Object** action is not available under a **BacnetNetwork**. It requires the **BacnetAWSNetwork**.

Deleting a dynamically-created object

This topic explains to delete the dynamically created object.

Prerequisites: You are working in Workbench connected to your Supervisor station that has been configured with a BacnetAWSNetwork.

Step 1 ExpandConfig→BacnetAWSNetwork.

The **Bacnet AWS Device Manager** opens.

Step 2 Click **Discover** button.

The driver runs discovery job to locate all remote controllers and devices.

Step 3 Expand a device and double-click its **Config** node in Nav tree.

The **Bacnet AWS Config Manager** view opens.

Step 4 To delete the object, select the object and click **Delete**.

Types of objects to export

To export components you use the **Export Table's Bacnet Export Manager** view. To export files, you use the **Bacnet File Export Manager** view. To export histories you use the **Bacnet Niagara Log Export Manager** view.

Table 2 Objects for export as BACnet objects

| Station component, file, or history type | BACnet object choices | Notes |
|------------------------------------------------------------------------------------------------------------|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Alarm Class | Notification Class | Alarm Class components (under the station's AlarmService) export as BACnet Notification Class objects. |
| String Point, String Writable | String Input | Proxy points under any BACnet driver are supported. |
| BooleanPoint, BooleanWritable, various kitControl components (that is, Logic), which have a null ProxyExt. | Binary Input | Proxy points under any driver are supported. Acceptable kitcontrol components to export include all Logic components. Typically, you export these as value-type objects (Analog Value, Binary Value, Multistate Value). |
| | Binary Value | |
| BooleanWritable (additional choices) | Binary Output | |
| | Binary Value Prioritized | |
| EnumPoint, EnumWritable | Multi State Input | Proxy points under any driver are supported. The facets, range, ordinal (integer) in the exported EnumPoint or EnumWritable must begin with 1, and not 0. The range must also be contiguous up to the maximum value (gaps in the range are not allowed). Otherwise, the object does not export to BACnet. |
| | Multi State Value | |
| EnumWritable (additional choices) | Multi State Output | |
| | Multi State Value Prioritized | |
| NumericPoint, NumericWritable, various kitControl components (that is, Math) | Analog Input | Proxy points under any driver are supported. Acceptable kitcontrol components to export include all Math components. |
| | Analog Value | |
| NumericWritables (additional choices) | Analog Output | |
| | Analog Value Prioritized | |
| LoopPoint | Loop | Loop object export is default. You can export a LoopPoint as a BACnet Loop object. |

| Station component, file, or history type | BACnet object choices | Notes |
|----------------------------------------------------------------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Analog Input | |
| | Analog Value | |
| BooleanSchedule, EnumSchedule, NumericSchedule, StringSchedule | Schedule | In addition, you can selectively export the configuration of a standard schedule or calendar into an existing Schedule object or Calendar object in a client BACnet device, using the Bacnet Schedule Export Manager view of that BacnetDevice's Schedules extension. |
| CalendarSchedule | Calendar | You cannot export a TriggerSchedule because BACnet has no object equivalent. |
| <file> (any) | File | Files under the station folder can be exported. |
| <history> (any) | Trend Log | Two different methods exist for exporting histories. |

Discovering components to export as BACnet objects

You export station objects to BACnet objects so that client devices, such as your Supervisor station, can make requests of the objects. This includes exporting schedules, calendars and histories. Discovery vs. new and/or matching is the usual way to locate and export objects.

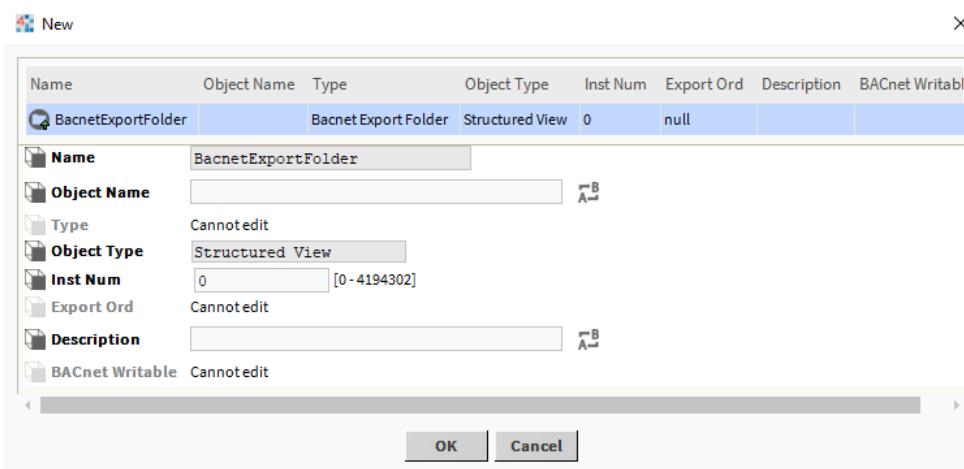
Prerequisites: You are working in Workbench connected to a remote host station.

Step 1 Expand Config→Drivers→BacnetNetwork→Local Device and double-click **Export Table**.

The **Bacnet Export Manager** opens.

Step 2 To create an export folder to hold your server descriptors, click the **New Folder** button.

The **New** export folder window opens.



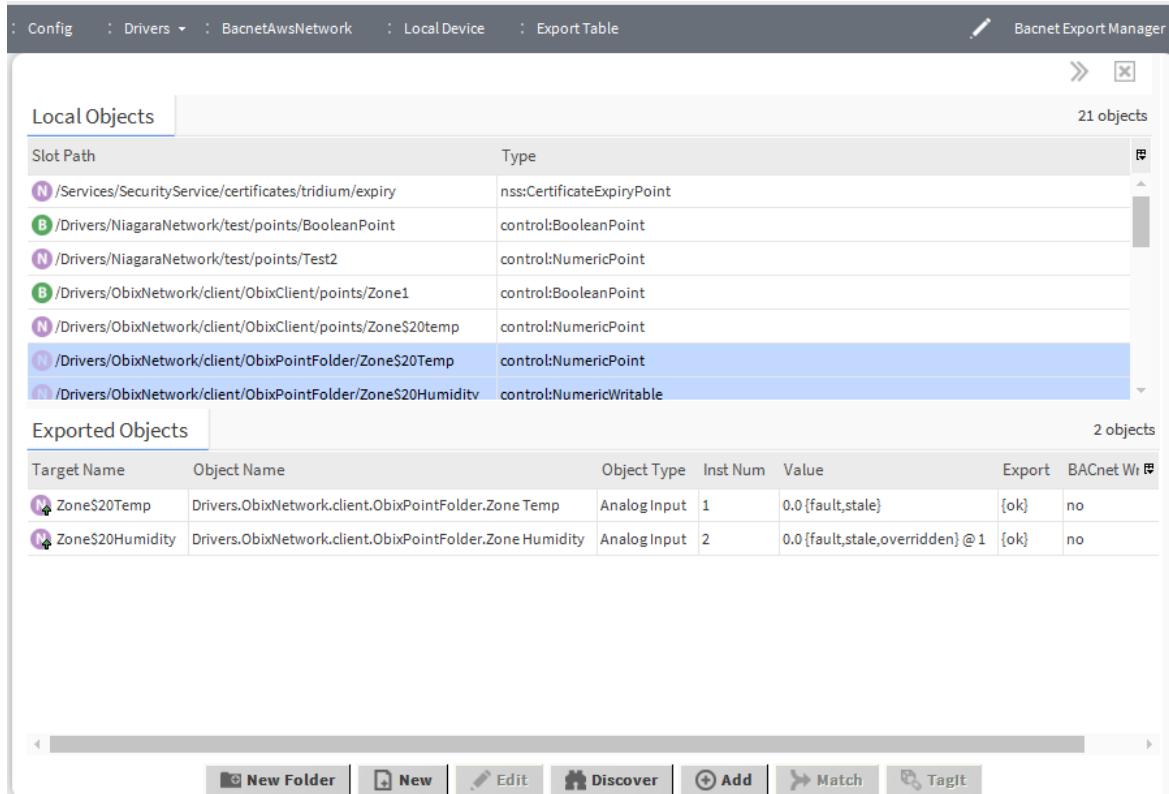
Step 3 Edit any other folder properties and click **OK**.

Step 4 To rename the folder, select it, click **Ctrl + R**, and enter a new name.

This should be a name related to the objects contained in the folder.

Step 5 Click the **Discover** button.

The **Bql Query Builder** window opens.



Step 6 Using this interface, locate the station objects to export (expose) as BACnet objects.

In the **Bql Query Builder**, type selections other than Control Point, Boolean Point, Enum Point, Numeric Point, Schedule, and Alarm Class, which have no practical application.

The **Local Objects** (discovered) pane lists the found components. The **Export Objects** (database) pane lists currently-exported objects. These special export descriptor components point to the exposed component. Additional slots determine how the driver handles BACnet access and writes.

Setting up a point export descriptor

You export point values (Analog Value, Binary Value, Multistate Value) from a server (remote controller) station to a client, such as a Supervisor station so that the client can view the current value of each point, and process historical trends. This is sometimes referred to as "exporting to BACnet" and "exposing objects in the host station." This procedure sets up export descriptors in the station's **Export Table**.

Prerequisites: You are using Workbench connected to a remote host station with points to export. If this point is capable of generating alarms, you configured the AlarmService before exporting point values. If you are exporting large numbers of records, you are ready to set up a folder structure to facilitate export management.

Step 1 Expand **Config**→**Drivers**→**BacnetNetwork**→**Local Device** and double-click **Export Table**.

The **Bacnet Export Manager** view opens.

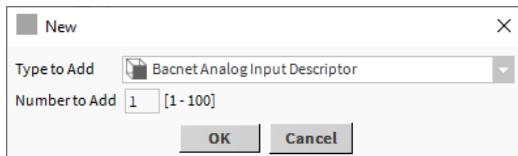
Step 2 To create a folder, click **New Folder**, give the folder a name and click **OK**.

Logically organizing multiple BACnet export descriptors into folder containers saves time. For example, one or more folders may represent a driver network, and another folder may contain schedules. Each export folder provides the same set of export manager views as the Export Table.

The station exposes export folders to BACnet as Structured View Objects, allowing a hierarchical organization of BACnet objects.

Step 3 To create a descriptor, click **New**.

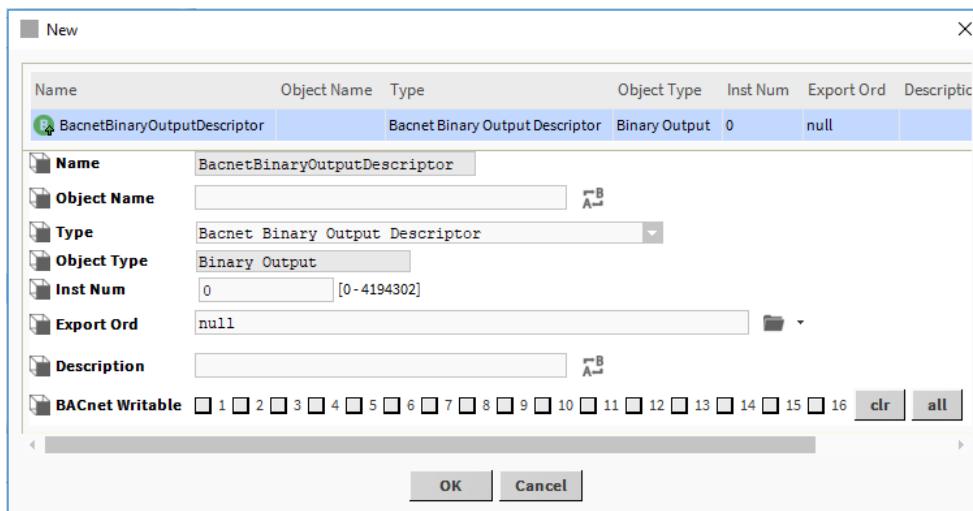
The **New** window opens.



Step 4 Select the type of descriptor to add using the **Type to Add** drop-down list and **Number to Add**, then click **OK**.

Type to Add is the type of point to configure for export. By increasing the **Number to Add** value you can create multiple instances of this point type in this single step.

The **New** point window with properties opens.



Step 5 Use the batch search and replace feature ($\text{Ctrl} + \text{F}$) to identify each point by its **Object Name** or **Description**.

Step 6 To configure the descriptor to export a group of points at once, select them all in this window, enter the **Inst Num** for the first one.

The driver assigns each descriptor successive instance numbers as available.

Step 7 Configure the other properties and click **OK**.

The manager automatically performs object ID maintenance on exported components, which ensures that the driver creates no duplicate combinations of object type and instance number. It also adds to each exported object an **OutOfServiceExt**.

Any exported points or components that contain an alarm extension are automatically exposed to BACnet with properties related to alarming available. For example, if you export a NumericPoint with an **OutOfRangeAlarmExt**, its exposed BACnet object (say, Analog Value) contains properties **High Limit**, **Low Limit**, **Deadband**, and so on.

NOTE: Before deleting any source (exported) component from the station, use the **Bacnet Export Manager** to delete its corresponding BACnet export descriptor. Otherwise, the export descriptor may remain orphaned in the **Export Table**, showing **Invalid Ord!** in the Target Name and Value columns. If this occurs, delete the orphaned export descriptors manually.

Exporting a schedule

You would export a schedule from a remote station to the Supervisor station connected to the BACnet network. A **BacnetScheduleExport** descriptor pushes schedules or calendars components to specific BACnet Schedule or Calendar objects in the Supervisor station.

Prerequisites: You are using Workbench connected to a remote host station that contains a schedule or calendar object to export.

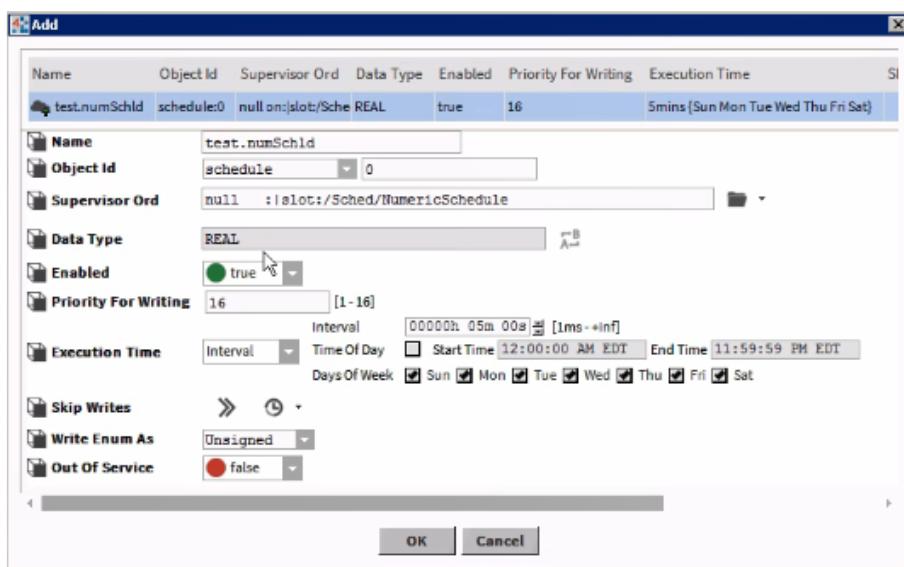
The **BacnetNetwork's Bacnet Device→Schedules** contains a **Bacnet Schedule Export Manager** view, which permits exposing schedule components in the station as either BACnet schedule objects or calendar objects. This server-type export makes exported components available to any networked BACnet device. It does not write to specific objects in a BACnet device, which act as clients.

- Step 1 To create a schedule export descriptor, expand **Config→Drivers→BacnetNetwork→BacnetDevice**, right-click **Schedules** and click **Views→Bacnet Schedule Export Manager**.

The **Bacnet Schedule Export Manager** opens.

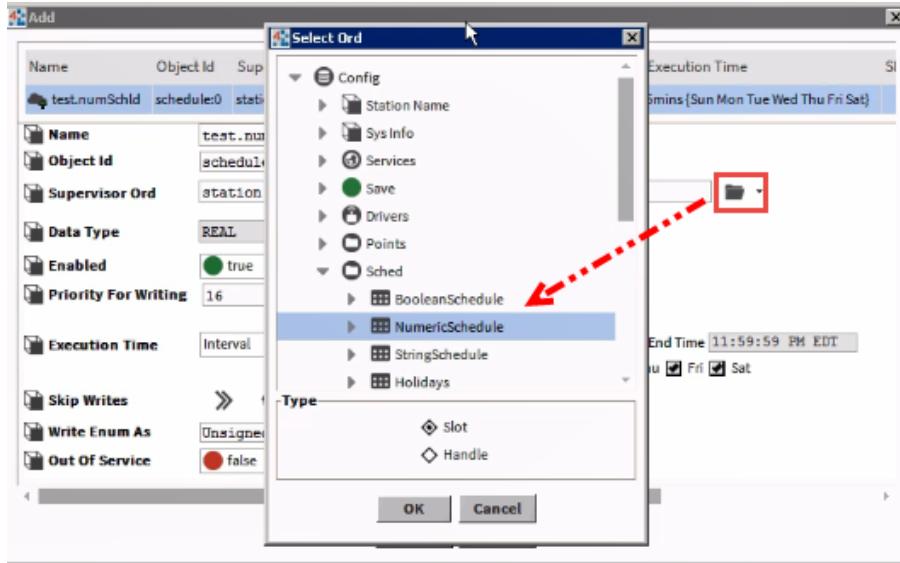
- Step 2 Click **New**, accept the default **Type to Add** and click **OK**.

The **New** window opens.



- Step 3 To define the location of the schedule, select the **Supervisor Ord**.

The **Select Ord** window opens.



- Step 4 Set the **Priority For Writing** as needed by the application in the BACnet device.
- Step 5 If needed, adjust the **Execution Time** (write synchronization) or set to Daily or even Manual (which requires an export from the **Bacnet Schedule Export Manager**).
- Step 6 To create the schedule export job, click **OK**.

The Framework adds the descriptor to the database, and makes an attempt to write the (source) configuration into the (target) BACnet schedule or calendar object. If successful, the status of the **BacnetScheduleExport** component remains `ok`, and its row in the **Database** table of the export manager remains white.

However, if any portion of the BACnet write failed, the **BacnetScheduleExport** reports a fault status, and its row in the export manager's **Database** table appears orange.

- Step 7 In the case of a fault, access the **BacnetScheduleExport** properties and examine the **Fault Cause**. If necessary adjust the **Skip Writes** property.
- Step 8 To export the schedule, select the row in the table and click **Export**.

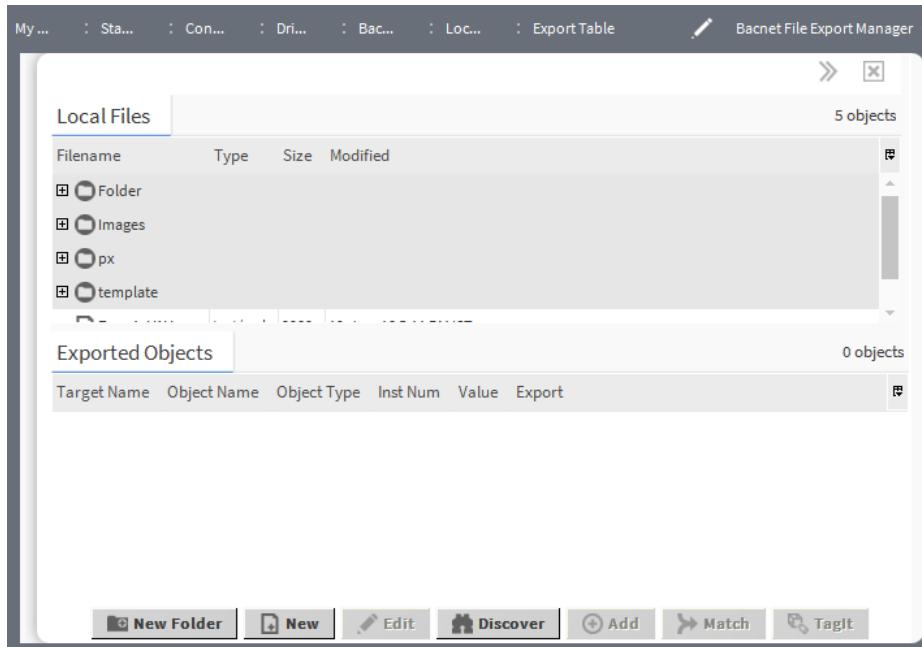
The manager automatically performs object ID maintenance on exported components, which ensures that the driver creates no duplicate combinations of **Object Type** and **Instance Number**. It also adds to each exported object an **OutOfServiceExt**.

Exporting a file

You typically discover files to export using the **Bacnet File Export Manager**, as opposed to using the **New** and/or **Match** buttons.

Prerequisites: You are working in Workbench connected to a remote host station. The **Bacnet Export Manager** view is open.

- Step 1 From the right most corner, select the **Bacnet File Export Manager** from the drop-down list.
The **Bacnet File Export Manager** view opens.
- Step 2 Click **Discover**.
The top pane shows files under the station's folder.



As with other manager views featuring online discovery, learn mode in **Bacnet File Export Manager** has two panes:

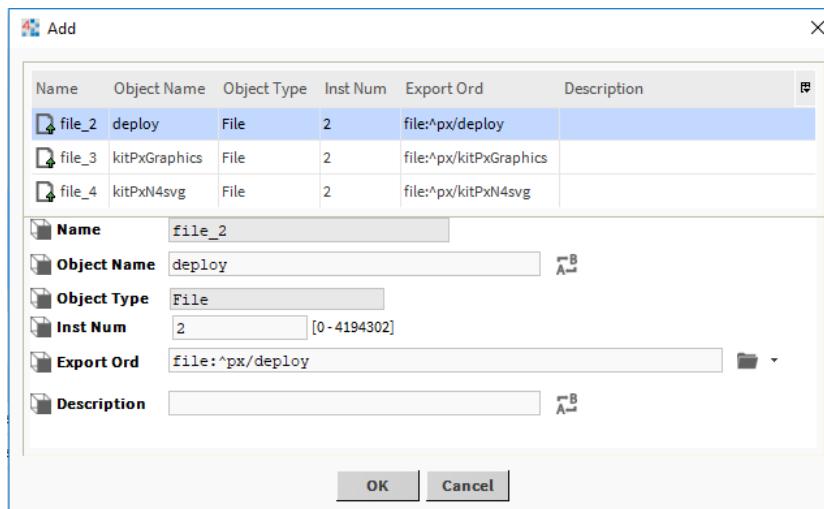
- Top (discovered) is the **Local Files** pane. It lists all files and subfolders under the station folder.
- Bottom (database) is the **Exported Objects** pane. It lists files currently exported to BACnet. These are special file descriptor components—each essentially a pointer to the exposed station file, with additional slots that determine how the system handles BACnet access and writes.

When exporting, the manager automatically performs object ID maintenance on exported files, ensuring that it creates no duplicate combinations of object type and instance number.

Step 3 Do one of the following:

- Select and drag discovered objects from the **Local Files** pane to the **Exported Objects** pane.
- To identify a single file to export, click **New**.
- To export one or more files, select the file(s) and click **Add**.

The **New** or **Add** file export window opens.



Step 4 If you used the **New** or **Add** buttons, use the batch search and replace feature () to locate the file and click **OK**.

The manager automatically performs object ID maintenance on exported components, which ensures that the driver creates no duplicate combinations of object type and instance number. It also adds to each exported object an **OutOfServiceExt**.

History export

BACnet trend log objects closely resemble standard histories in many ways. However, a fully-compliant BACnet trend log includes additional data, such as sequence numbers, as well as record entries for log events, such as log enable, log disable, and buffer purge within the log data.

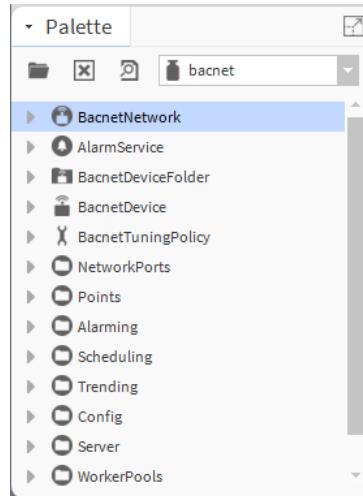
Therefore, depending on requirements, there are two methods to export a history:

- When a fully BACnet-compliant trend log object is needed, you must drag **BacnetTrendLogExt** extension from the **bacnet** palette into the source point in the station, instead of the typical history extension. In this case, the resulting history provides the additional BACnet-compliant data, such as sequence numbers and log events.
- If retroactively exporting to BACnet, and a trend log object that provides data by time and by index, access only is sufficient. Export by selecting any standard history (created by a standard history extension). Do this in the **Bacnet Niagara Log Export Manager** using a discover job. In this case, the exposed trend log object does not provide sequence numbers or by sequence number access, because this information was not stored within the original history.

BacnetTrendLogExt extensions

The **BacnetTrendLogExts** reside in the Trending folder of the **bacnet** palette.

Figure 10 BacnetTrendLogExts in the bacnet palette



You can use one of these instead of a standard history extension, pasting it in any point in which you need to export its history as a fully BACnet-compliant trend log object. For example, if you have a Lon proxy point of type NumericPoint, for nvoSpaceTemp representing room temperature, you could copy a **BacnetNumericIntervalTrendLogExt** into that NumericPoint. Its resulting history (created only after you enable it, as with any history extension) will now be fully BACnet-compliant.

By default, when you add one of the extensions to a component, note that a corresponding Bacnet log descriptor of type **BacnetTrendLogDescriptor** is automatically created in the root of the Export Table. You still define the normal collection properties in the **BacnetTrendLogExt** history extension (of the source component), however, there are additional properties in its associated BACnet trend log descriptor.

Differences between BACnet-exposed histories and standard histories

A history created by a BacnetTrendLogExt has extra properties for sequence numbers and log events. A standard history does not include these properties.

Exporting standard history

You typically use **Discover** in the **Bacnet Niagara Log Export Manager** for any standard history you wish to export to BACnet as trend log object—that is, without the ability for sequence number access from BACnet. Using a discovery job is not necessary for the history of any component in which you have already added one of the BacnetTrendLogExts—as the BACnet driver automatically added a BacnetTrendLogDescriptor under the root of the **Export Table**.

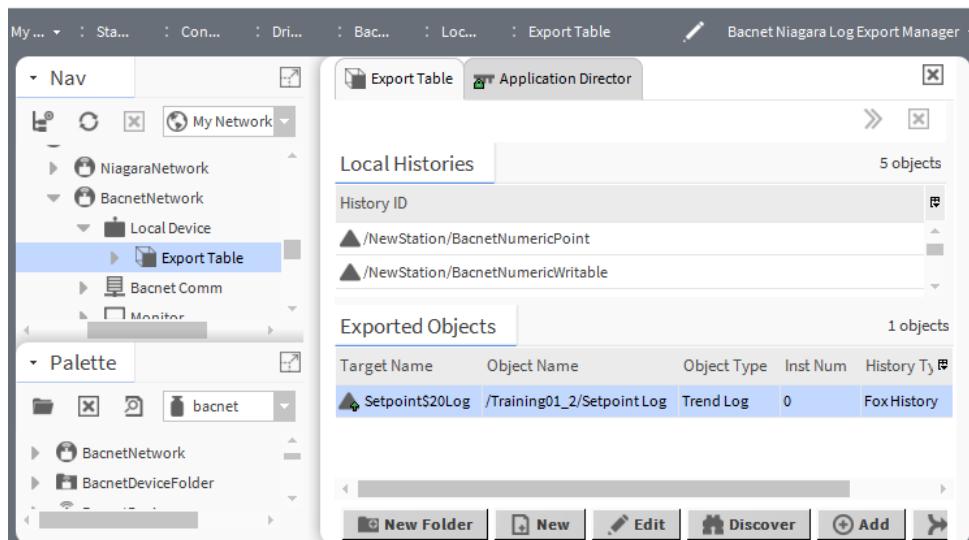
Prerequisites: You are working in Workbench connected to a remote host station.

Step 1 Expand **Config→Drivers** followed by expanding the BACnet network node in the Nav tree, right-clicking the **Export Table**, and clicking **Views→Bacnet Niagara Log Export Manager**

Step 2 Click **Discover**.

The top pane shows the histories under the station's folder.

As with other manager views featuring online discovery, learn mode in **Bacnet Niagara Log Export Manager** has two panes:



- Top (discovered) **Local Histories** pane—listing all histories in the local station.
- Bottom (database) **Exported Objects** pane—listing histories currently exported to BACnet. These include both types of log descriptor components—each essentially a pointer to the exposed history, with additional slots that determine how BACnet access/writes are handled.

Step 3 Select a history and click **Add**.

The system exports the history object.

When exporting, the manager automatically performs object ID maintenance on exported files, ensuring that no duplicate combinations of object type and instance number are created.

Configuring exported objects for write

By default, the BACnet driver provides external BACnet (client) devices read-access to all exposed (exported to BACnet) objects in the station. Access depends on a station user named **BACnet**. If the BACnet user does not already exist in the station, the BACnet driver automatically creates it upon startup. However, write-

access from the **BACnet** user to the properties of exported components, including invoking commands (actions), requires that the user have the necessary permissions for the components.

The station may need to write to a property like **Out Of Service** or **Notify Type**. If an exported NumericWritable has an alarm extension, and you want to permit external BACnet writes to its alarm limit values. In both cases you need to configure the **BACnet** user to have admin write permissions on the exported BooleanWritable and NumericWritable points.

The **BACnet** user is initially created without any permissions. This procedure documents how to assign the necessary permissions to the BACnet user.

Step 1 Expand **Config→Drivers→BacnetNetwork→BacnetDevice→Points** and double-click a writable point that the system will export.

The point's **Property Sheet** opens.

Step 2 To allow an invoked Active action from BACnet to an exported BooleanWritable, make the point **BACnet Writable** at priority.

Step 3 Next, expand the **Station→Config→Services→UserService** and double-click the **BACnet** user.

The BACnet (User) property sheet opens.

| BACnet (User) | | | | | | | | | | | | | | | | | | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|----------|-------|---------------------------|--------------------------------------|--------------------------|--------------------------------------|------------------------|--------------------------------------|-------------------------|--------------------------------------|----------------------|--------------------------------------|--------------------|--------------------------------------|-------------------|--------------------------------------|---------------------|--------------------------------------|
| Full Name | | | | | | | | | | | | | | | | | | | |
| Enabled | <input checked="" type="radio"/> true | | | | | | | | | | | | | | | | | | |
| Expiration | <input type="checkbox"/> Never Expires <input type="checkbox"/> Expires On 22-Jun-2018 11:59 PM IST | | | | | | | | | | | | | | | | | | |
| Lock Out | <input checked="" type="radio"/> false | | | | | | | | | | | | | | | | | | |
| Language | | | | | | | | | | | | | | | | | | | |
| Email | | | | | | | | | | | | | | | | | | | |
| Password | | | | | | | | | | | | | | | | | | | |
| Authenticator | Confirm | | | | | | | | | | | | | | | | | | |
| Force Reset At Next Login <input checked="" type="radio"/> false | | | | | | | | | | | | | | | | | | | |
| <input type="checkbox"/> Never Expires <input type="checkbox"/> Expires On 22-Jun-2018 11:59 PM IST | | | | | | | | | | | | | | | | | | | |
| Facets | Time Format (default) | | | | | | | | | | | | | | | | | | |
| Unit Conversion None | | | | | | | | | | | | | | | | | | | |
| Nav File | null | | | | | | | | | | | | | | | | | | |
| Prototype Name | | | | | | | | | | | | | | | | | | | |
| Network User | <input checked="" type="radio"/> false | | | | | | | | | | | | | | | | | | |
| Cell Phone Number | | | | | | | | | | | | | | | | | | | |
| Authentication Scheme Name | DigestScheme | | | | | | | | | | | | | | | | | | |
| Roles | admin >> | | | | | | | | | | | | | | | | | | |
| Allow Concurrent Sessions | <input checked="" type="radio"/> true | | | | | | | | | | | | | | | | | | |
| Auto Logoff Enabled <input checked="" type="radio"/> true | | | | | | | | | | | | | | | | | | | |
| Auto Logoff Settings | Use Default Auto Logoff Period <input checked="" type="radio"/> true | | | | | | | | | | | | | | | | | | |
| Auto Logoff Period 00000h 15m | | | | | | | | | | | | | | | | | | | |
| Type <input type="button" value="HTML5 Hx Profile"/> <table border="1"> <tr> <td>Hx Theme</td> <td>Zebra</td> </tr> <tr> <td>Enable Hx Workbench Views</td> <td><input checked="" type="radio"/> Yes</td> </tr> <tr> <td>Enable Nav Tree Side Bar</td> <td><input checked="" type="radio"/> Yes</td> </tr> <tr> <td>Enable Search Side Bar</td> <td><input checked="" type="radio"/> Yes</td> </tr> <tr> <td>Enable Palette Side Bar</td> <td><input checked="" type="radio"/> Yes</td> </tr> <tr> <td>Enable Nav File Tree</td> <td><input checked="" type="radio"/> Yes</td> </tr> <tr> <td>Enable Config Tree</td> <td><input checked="" type="radio"/> Yes</td> </tr> <tr> <td>Enable Files Tree</td> <td><input checked="" type="radio"/> Yes</td> </tr> <tr> <td>Enable History Tree</td> <td><input checked="" type="radio"/> Yes</td> </tr> </table> | | Hx Theme | Zebra | Enable Hx Workbench Views | <input checked="" type="radio"/> Yes | Enable Nav Tree Side Bar | <input checked="" type="radio"/> Yes | Enable Search Side Bar | <input checked="" type="radio"/> Yes | Enable Palette Side Bar | <input checked="" type="radio"/> Yes | Enable Nav File Tree | <input checked="" type="radio"/> Yes | Enable Config Tree | <input checked="" type="radio"/> Yes | Enable Files Tree | <input checked="" type="radio"/> Yes | Enable History Tree | <input checked="" type="radio"/> Yes |
| Hx Theme | Zebra | | | | | | | | | | | | | | | | | | |
| Enable Hx Workbench Views | <input checked="" type="radio"/> Yes | | | | | | | | | | | | | | | | | | |
| Enable Nav Tree Side Bar | <input checked="" type="radio"/> Yes | | | | | | | | | | | | | | | | | | |
| Enable Search Side Bar | <input checked="" type="radio"/> Yes | | | | | | | | | | | | | | | | | | |
| Enable Palette Side Bar | <input checked="" type="radio"/> Yes | | | | | | | | | | | | | | | | | | |
| Enable Nav File Tree | <input checked="" type="radio"/> Yes | | | | | | | | | | | | | | | | | | |
| Enable Config Tree | <input checked="" type="radio"/> Yes | | | | | | | | | | | | | | | | | | |
| Enable Files Tree | <input checked="" type="radio"/> Yes | | | | | | | | | | | | | | | | | | |
| Enable History Tree | <input checked="" type="radio"/> Yes | | | | | | | | | | | | | | | | | | |
| <input type="button" value="Refresh"/> <input type="button" value="Save"/> | | | | | | | | | | | | | | | | | | | |
| Default Web Profile | | | | | | | | | | | | | | | | | | | |

Step 4 Configure the station's BACnet user to have operator write permissions by right-click the chevron to the right of the **Permissions** property.

The Permissions grid opens.

Step 5 With the Admin category selected, click the cell under Operator, W.

This configures the BACnet user with operator write permission.

BACnet user permissions also apply to writes of any exported files and histories. While a password for the BACnet user is technically not needed (for external BACnet access), you should assign one anyway, because of the write permissions typically assigned.

Step 6 Configure a password for the BACnet user.

Make it a non-blank password, and guard this password carefully!

Step 7 To complete the configuration, click **OK**.

Chapter 7 Internetworks and BACnet/IP

Topics covered in this chapter

- ◆ Example internetwork diagrams
- ◆ Example BACnet/Ip diagrams
- ◆ Internetwork rules
- ◆ BACnet/IP and BBMDs

The BACnet standard defines an internetwork (inter-network) as a set of two or more BACnet networks connected by BACnet routers. This chapter explains topics relating to BACnet internetworks and station configuration. Also included is information related to BBMDs (BACnet/IP Broadcast Management Device).

Link layer types

All BACnet drivers support more than one link layer type using multiple network ports under the BACnet network's **Bacnet Comm→Network** child container. A new BACnet network defaults to BACnet/IP (**Ip Port**). The other port types (**EthernetPort**, **MstpPort**) are available in the **bacnet** palette. The **MstpPort** component, which supports a BACnet MS/TP device trunk, requires a station hosted by a QNX-based host.

Network numbers

Each port must reference a unique BACnet network number across an internetwork. The range for BACnet network numbers is 1-65535. You define this **Network Number** property in each port component's **Property Sheet**. This property defaults to -1, which means that it is undefined or inactive.

If you are installing a station on an existing BACnet internetwork, where one or more BACnet routers already exist, you need to know the assigned network number(s), and enter them, as appropriate, in the network port(s) under the **Bacnet Comm→Network** container. You can find the existing network number information in the configuration setup of the BACnet routers.

A station's BACnet network is not limited to only BACnet networks defined by its own network ports. The BACnet network automatically learns other remote networks, from global BACnet I-Am messages received from the devices (and routers) on other remote networks. The station maintains a table of known BACnet networks under the network's **Bacnet Comm→Network** child in a **Router Table** component.

By default, the station automatically performs BACnet router functions across multiple BACnet networks. Often this is the desired behavior, especially when there are MS/TP trunks that attach to RS-485 ports on a remote controller. However, multiple BACnet routes to the same network segment result in message flood issues, and are considered a mis-configuration. If the driver detects this configuration, it automatically disables BACnet routing by setting the **Routing Enabled** property (found in the BACnet network's **Bacnet Comm→Network** component slot) to **false**.

Internetwork BACnet routers

The BACnet network maintains a table of BACnet routers in the **Bacnet Comm→Network** component. Included is the ability to modify, add, and remove entries that represent BACnet routers.

A full listing of all BACnet router functions is extensive—refer to the BACnet specification for complete details on routers. From a general perspective, BACnet routers pass BACnet messages among BACnet networks. This applies to all directed messages as well as broadcast messages.

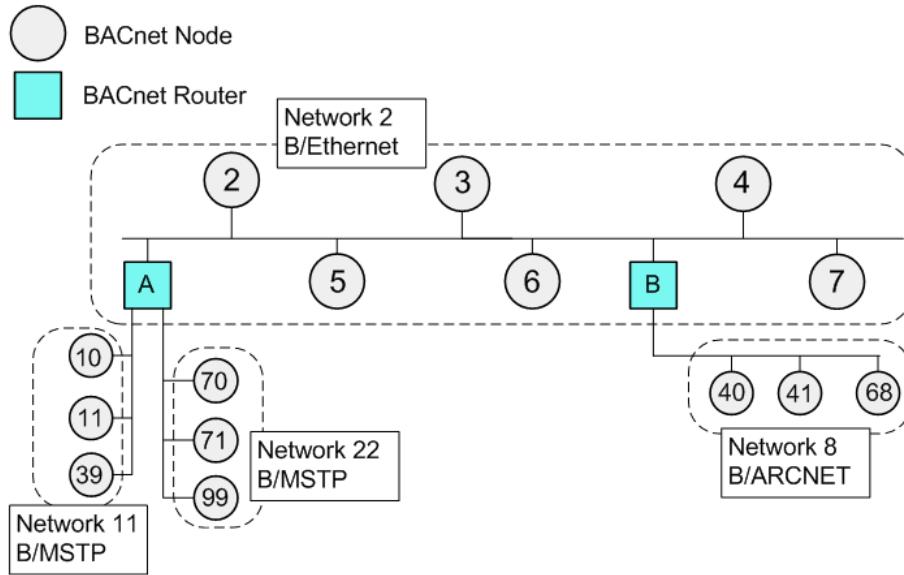
Usually, a BACnet router joins networks of different media/link-layer types, for example a router with RS-485 MS/TP port(s) and an Ethernet-BACnet/IP port.

Example internetwork diagrams

The diagrams in this topic illustrate internetwork configurations.

Four physical networks, two routers

Figure 11 Ethernet, MS/TP and ARCnet

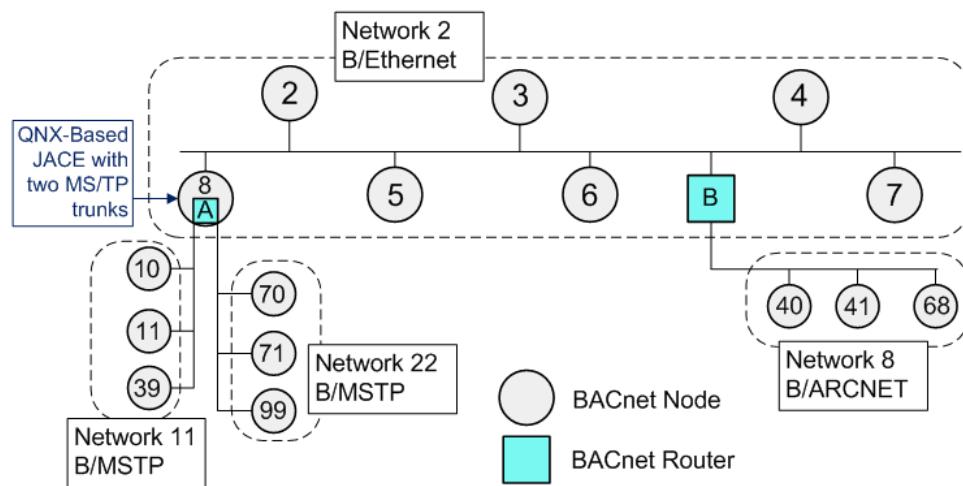


Routers A and B join together three media and link-layer types: BACnet over Ethernet (network 2), BACnet MS/TP (networks 11 and 22), and BACnet ARCnet (network 8). Each BACnet node has a unique Device ID number (shown inside the node circle). Router A could actually be a remote station that is licensed for MS/TP (next example).

BACnet Ethernet to MS/TP router functions provided by the station in a remote host

This internetwork includes an Ethernet/IP backbone.

Figure 12 Ethernet and MS/TP

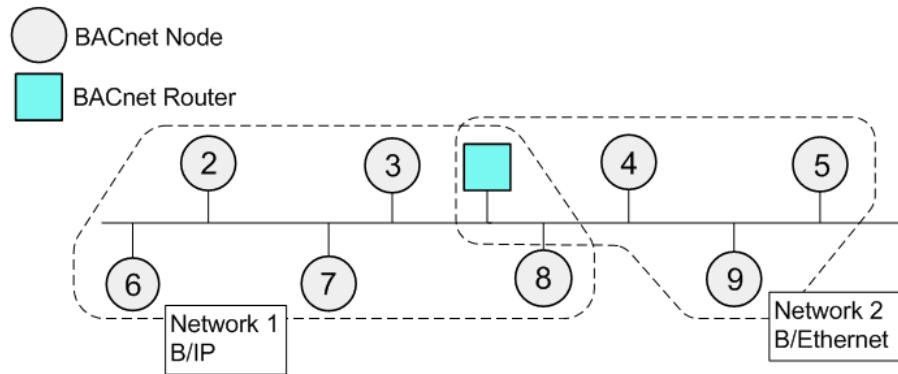


In this case, the host's station provides BACnet routing between the two MS/TP networks and an Ethernet network. On an Ethernet LAN, there may be two separate BACnet networks sharing the same media: BACnet over Ethernet and BACnet/IP.

Same physical Ethernet LAN, router joins two BACnet networks

Each BACnet node has a unique Device ID number (shown inside the node). This router function could actually be provided by a host station.

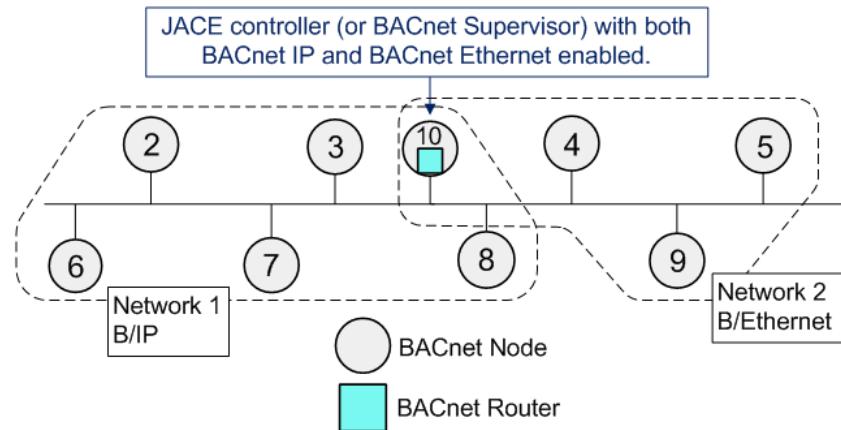
Figure 13 Two networks



BACnet/IP to BACnet/Ethernet router function provided by a station

The controller or BACnet Supervisor manages this network with both BACnet/IP and BACnet Ethernet enabled.

Figure 14 Two networks



In this case, be especially careful that no other router exists between the BACnet Ethernet and BACnet/IP networks. If the host station detects this illegal (loop) configuration, it stops its routing functions. This condition is indicated on the **Property Sheet** of the **Bacnet Comm→Network** component, where its **Routing Enabled** property is automatically set back to `false` (from `true`). Corresponding entries are generated in the station's LogHistory, which records the detected mis-configuration and disabled BACnet router functionality.

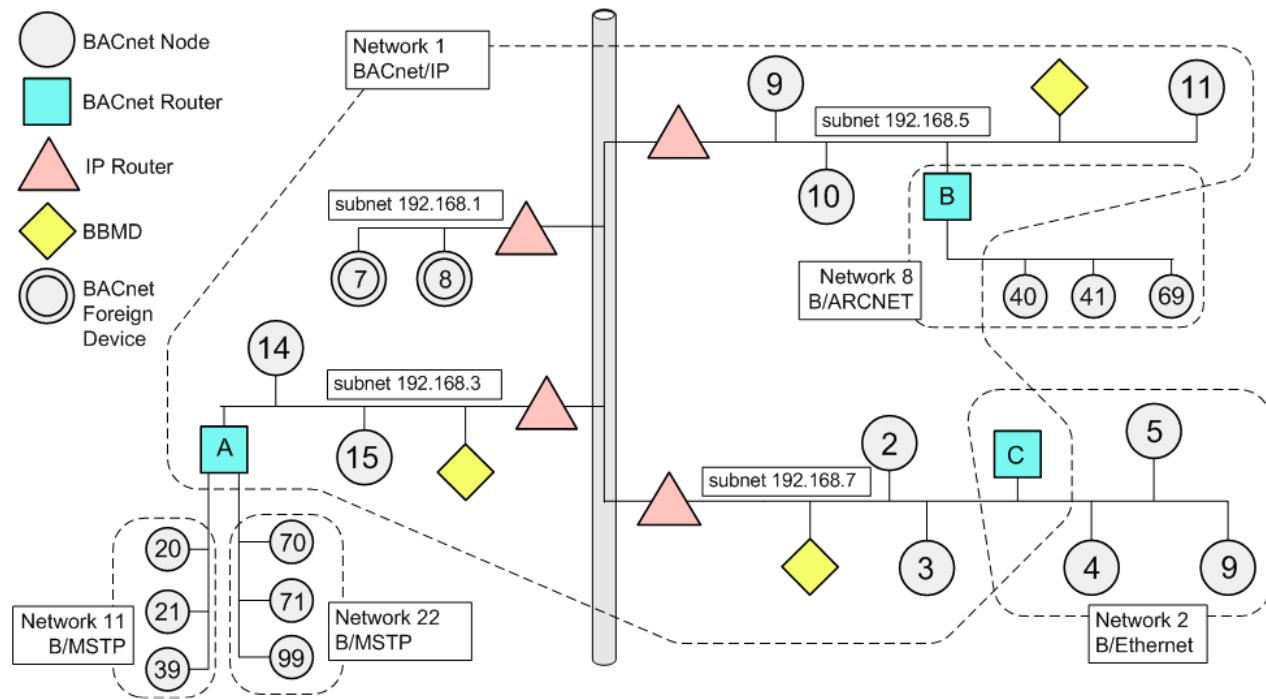
Example BACnet/IP diagrams

BACnet/IP networks can be complex, with devices designated as BBMDs and foreign devices, plus standard TCP/IP (IP) routers. BBMDs are used to avoid having separate B/IP networks for each subnet, which would require a BACnet/IP router to exist on each subnet (yet even more complexity).

Internetwork that includes a single Bacnet/IP network spanning multiple IP subnets, plus other BACnet networks

A Bacnet/IP network may span multiple IP subnets. Use of BBMDs provides BACnet broadcast message delivery through IP routers.

Figure 15 Multiple networks



On subnet 192.168.1, there is no BBMD. Both BACnet/IP nodes on that subnet (Device IDs 7, 8) are registered as foreign devices with a remote BBMD.

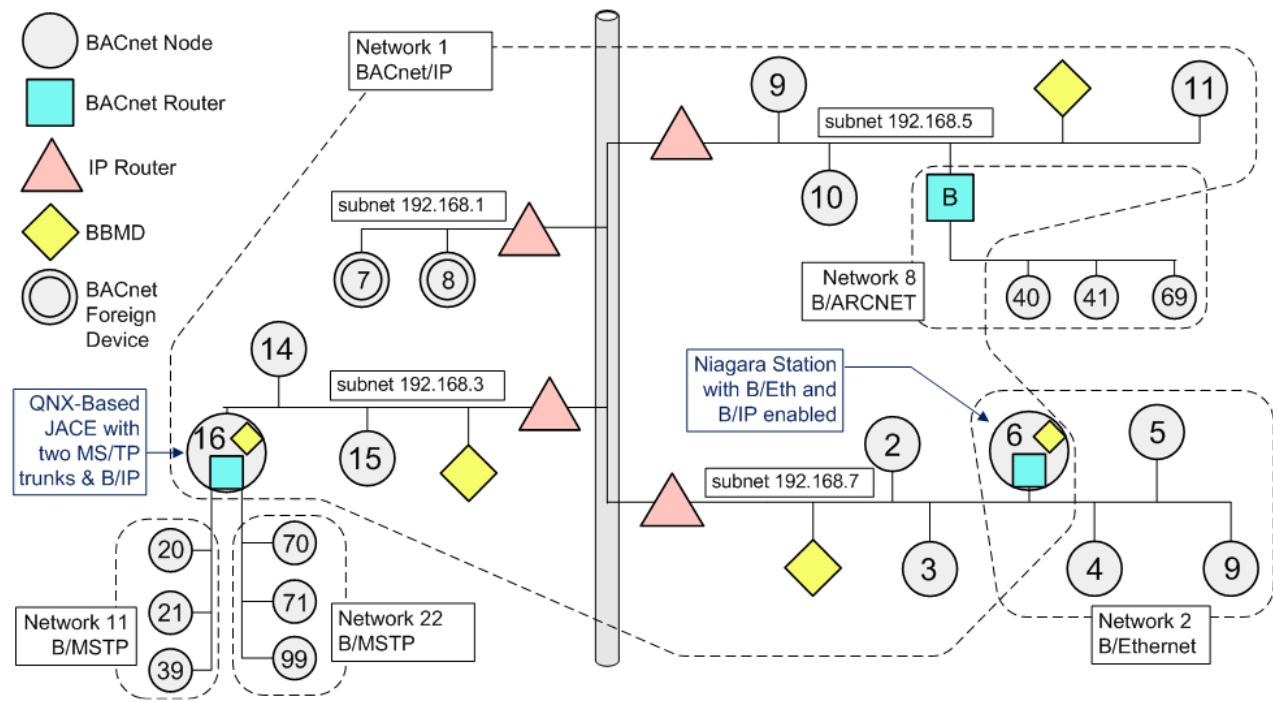
On the physical Ethernet segment used by subnet 192.168.7, there are both BACnet/IP devices (Device IDs 2 and 3) and BACnet over Ethernet devices (Device IDs 4, 5, and 9). Router C provides communications between these two BACnet networks.

With the exception of router B (B/IP to B/ARCNET), all BACnet routing and BBMD functions can be performed by host stations.

- On subnet 192.168.3, a QNX-based host with two MSTP trunks and BACnet/IP enabled performs router A functions (B/IP to MS/TP) and BBMD functions.
- On subnet 192.168.5, BBMD a host station performs functions with BACnet/IP enabled.
- On subnet 192.168.7, any host station with both B/Ethernet and BACnet/IP enabled can perform router C functions (B/IP to B/Eth) and BBMD functions.

A similar internetwork with stations providing BBMD and routing functions

The figure shows a similar internetwork with functions provided by Framework stations.

Figure 16 Framework station provide internetwork functions

In the internetwork above, the station operating as BACnet device (node 6) could be hosted by either a remote controller or a BACnet (OWS or AWS) Supervisor.

Internetwork rules

These rules ensure communication success among networks and devices on an internetwork.

Rule: There must be only one message path between any two BACnet devices on an internetwork. No communication loops are allowed.

Example: Do not configure multiple BACnet devices on the same LAN with both BACnet/IP and BACnet/Ethernet enabled. This includes host (remote controller) stations.

Rule: A BACnet router must exist between two BACnet networks (different network numbers) for BACnet messages to pass between the devices on the two networks. This applies to any link-layer types.

The important concept here is that a BACnet router can be a single-purpose device, or an application layer device (controller) that also includes router functionality, such as a station running in a remote host.

The BACnet Specification requires that all devices communicating on the BACnet network have an application layer entity, including a BACnet Device object.

By default, if a station is configured for multiple link-layers, it automatically acts as a router between two (or more) BACnet networks to which it is directly attached. However, a station does not act as a plain IP router.

For example, if an EthernetPort and two MstpPorts are enabled in a remote station, the station operates as a BACnet router between the three networks. If Ethernet, IP, and a MS/TP ports are enabled, the station operates as a BACnet-to-BACnet/IP router between the three networks, and so forth. If you add a second IpPort, bound to a different UDP port, the station routes BACnet messages among the networks, just like other data link layers.

Rule: Within any given internetwork, each BACnet network must have a unique network number, from 1 to 65534.

The concept is that each link-layer (network port), which is enabled in the BACnet network's **Bacnet Comm→Network** container corresponds to a specific BACnet network, specified by the **Network Number** property.

If you establish a new BACnet internetwork, you can specify any network number for each port. For example, BACnet/IP (IpPort) = 1, BACnet/Ethernet (EthernetPort) = 2, and (if a QNX-based host), MstpPort (1) = 3, MstpPort (2) = 4, and so on.

If you add a host on an existing BACnet internetwork, you should specify the established BACnet/IP network number and/or BACnet/Ethernet network number currently in use. If you have a host with one or more enabled MstpPorts, you must also specify a previously-unused network number for each network.

Rule: Every BACnet object, including the Device object in each BACnet device, must have a unique numeric Object_Identifier.

In the specific case of the Device object, this Object_Identifier must be unique, internetwork-wide. Valid values for this identifier range from 0 to 4194302.

The concept is that BACnet devices respond to a Who-Is broadcast message with an I-Am reply that includes each Device object's unique numeric Object_Identifier. If you receive duplicate I-Am messages (multiples showing the same device, by number), either more than one device has been assigned the same identifier (number) or a message loop exists.

BACnet/IP and BBMDs

The driver's BACnet/IP support includes the ability to host BBMDs (BACnet/IP Broadcast Management Device). If a station is configured for BBMD operation, you can review (and if necessary, modify) the broadcast distribution table common to all BBMDs on the BACnet/IP network, as well as the foreign device table used in the local host station.

Use of BBMDs is the answer to a BACnet/IP network that needs to span multiple IP subnets. Each IP subnet with BACnet/IP devices requires one (and only one) BBMD.

A BBMD delivers BACnet broadcast messages, such as Who-Is and Who-Has. As a rule, standard IP routers inherently block globally broadcast messages used to connect separate IP subnets. This blockage does not exist for directed messages between devices on different subnets, such as a common ReadProperty request, which is not a broadcast message.

The BBMD resolves this issue by acting as a broadcast manager for its subnet, working in coordination with other peer BBMDs. Each other IP subnet that contains BACnet/IP devices has one BBMD. Each BBMD stores a table with the IP address and distribution mask of all BBMDs, itself included. This is called the BBMD's BDT (Broadcast Distribution Table), and is identical in each BBMD for that entire BACnet/IP network. (The BDT in each BBMD is required to be identical in the original specification. However, Addendum 135-2008o (not currently implemented in the Framework) relaxes this requirement to allow BACnet/IP communication in the Network Address Translation (NAT) environment.)

A BBMD may be a device operating solely as a BBMD, or more typically, include BBMD functions in addition to other application/controller duties. This is the case with a host station running a BACnet network with its IpPort, Link container's **Ip Device Type** property set to **Bbmd** (instead of the default **Standard**, or **Foreign Device**).

NOTE:

Often an installation with BBMDs is not an internetwork, but a single BACnet/IP network. If so, host stations on different subnets should specify the same network number for their associated IpPort.

When a device sends a global broadcast message, all devices on its local subnet automatically received it, including the BBMD. (Local devices reply as needed to the broadcast device without BBMD involvement.) The local BBMD forwards the broadcast message to the other subnets, using one of two methods (as defined for each remote subnet):

- One-hop: the BBMD sends the message using a directed broadcast, whereby the IP router for the destination subnet broadcasts the message on its local subnet. The router must support directed broadcasts, otherwise the two-hop method must be used. One-hop is unusual, as IP routers rarely pass directed broadcasts from or to the local subnet.
- Two-Hop: the BBMD sends the message to its peer BBMD on the remote subnet. The receiving BMMD broadcasts the message on its local subnet.

Replies to BACnet broadcast messages may or may not require BBMD involvement. Occasionally, these are directed messages back to the particular BACnet/IP device that generated the message. However, replies to a Who-Is broadcast are often broadcast I-Am messages, as is the case with the BACnet driver.

Installing a host on an IP subnet with a BBMD

If you have an existing BACnet/IP network that spans IP subnets, chances are that it has at least one existing BBMD.

Step 1 Expand **BacnetNetwork**→**BacnetComm**→**IpPort**→**Link**.

Step 2 In the **Link** property sheet of the **Ip Port** leave the **Ip Device Type** property set to **Standard**.

Configuring a station as a BBMD

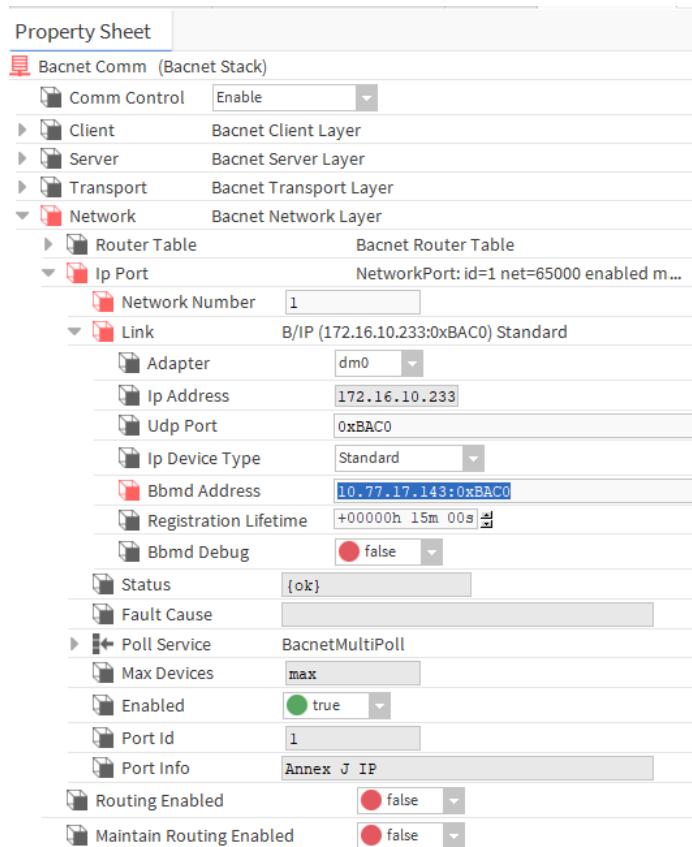
Unless the host for the station is located on that same IP subnet as an existing BBMD, you must configure it as a BBMD or a foreign device. Configuring the station as a BBMD is often the best option if the local subnet is receiving other new BACnet/Ip devices, and/or existing BACnet/Ip devices on this subnet and these devices are not currently supported by a BBMD.

Prerequisites: The IP router for the local subnet supports directed broadcasting.

Step 1 Expand the BACnet network's **BacnetComm** container to reveal **Network**→**IpPort**→**Link** component's property sheet .

Step 2 In the **Ip Device Type** property, select **Bbmd** from the drop-down list.

Step 3 To have the driver read from another BBMD, in the **Bbmd Address** property, enter the address of a known BBMD on a remote subnet using its full BACnet/IP MAC address, that is: **IP_Address**:
UDP_port, where **IP_Address** is the number that identifies the station, and **UDP_port** is a number that identifies the port, which conforms to the User Datagram Protocol.



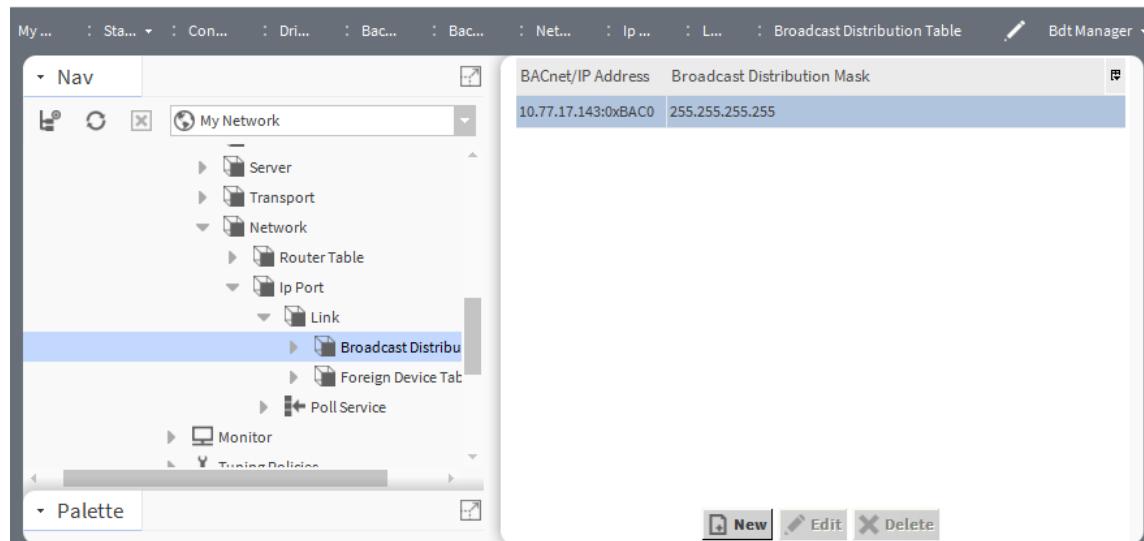
For example, for a BBMD at 10.77.17.143 using the standard UDP port 0xBAC0 (decimal 47808), either entry format is accepted: 10.77.17.143:47808 or 10.77.17.143:0xBAC0.

Step 4 To enable this configuration, click **Save**.

This should update all other BBMDs on the BACnet/IP network with the same BDT information.

Step 5 Under the **Link** container, expand the **Broadcast Distribution Table** slot to see child entries.

The **Bdt Manager** view opens.



This view allows you to manually add new BBMD entries if necessary, or edit or delete existing BBMD entries. There should be one local Device entry and at least one BdtEntry for the remote BBMD entered.

NOTE: You can also double-click the **Broadcast Distribution Table** component for its default **BDT Manager** view. The entry for the local device is indicated in a pale blue color.

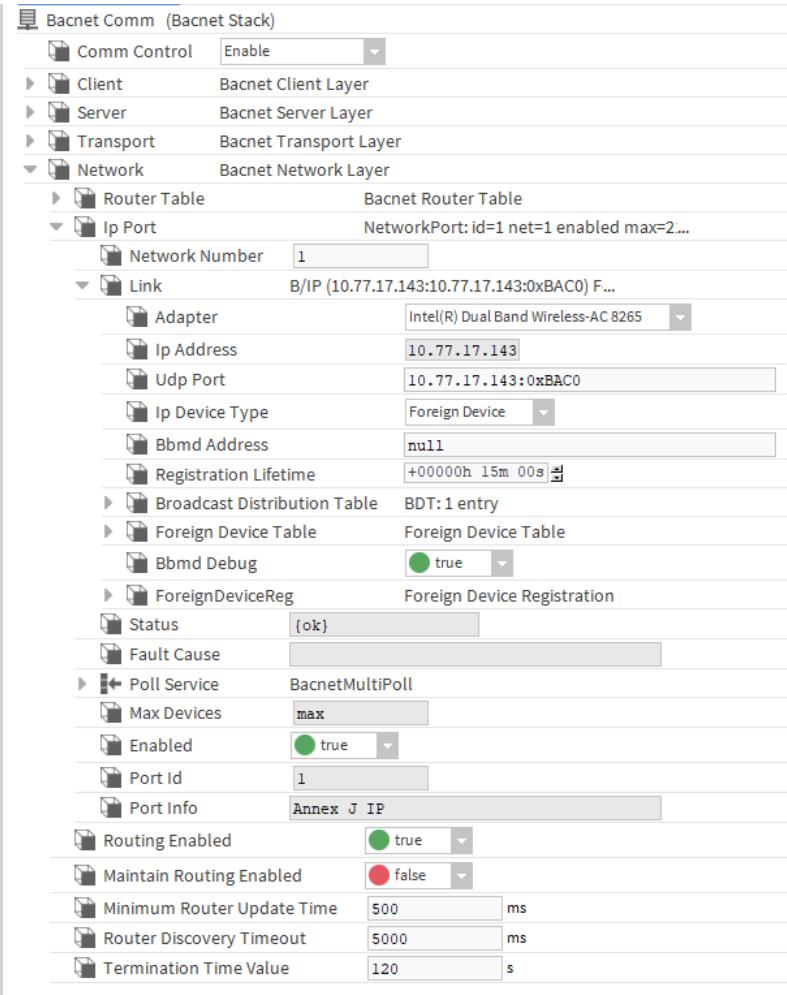
- Step 6 If the router for the local subnet supports directed broadcasting, edit the localDevice entry's subnet mask (in decimal) in the Distribution Mask field. Otherwise, if the router does not support directed broadcasting, or if you are not sure, leave 255:255:255:255 as the Distribution Mask.
- Step 7 From the **Bacnet Device Manager**, perform a device discover. If the BBMD configuration was successful, you should see devices from other subnets respond.

Configuring a station as a foreign device

If the station is configured as a BBMD, in addition to the **Broadcast Distribution Table** component, the IpPort's Link container also has a **Foreign Device Table** component. It accumulates entries for remote BACnet/IP devices that have registered with the station as a foreign device, and includes their registration time to live and upcoming purge time.

- Step 1 Expand the BACnet network's **Bacnet Comm**→**Network**→**IpPort**→**Link** node and open the property sheet for the **Link** component.
- Step 2 In the **Ip Device Type** property, select **Foreign Device** from the drop-down list.
- Step 3 In the **Bbmd Address** property, enter the address of a known BBMD on a remote subnet using its full BACnet/IP MAC address: **IP_Address:UDP_port**, where **IP_Address** is a number that identifies the station, and **UDP_port** is a number that identifies the port, which conforms to the User Datagram Protocol.

This is the BBMD that the station attempts to register with as a foreign device.



For example, for a BBMD at 10.77.17.143 using the standard UDP port 0xBAC0 (decimal 47808), either entry format is accepted: 10.77.17.143:47808 or 10.77.17.143:0xBAC0

- Step 4 In the **Registration Lifetime** property, either accept the default 15 minutes, or enter another value.
The station automatically re-registers within this period.
- Step 5 To enable this configuration, click **Save**.
This station attempts to register with the remote BBMD.
- Step 6 From the **Bacnet Device Manager**, perform a device discover.
If the foreign device configuration was successful, you should see devices from other subnets respond.

New BACnet IP network BBMD configuration

If configuring the host station while installing BACnet/IP devices for a new BACnet/IP network, there are probably no existing BBMDs. In this case, follow the instructions for configuring a station as a BBMD.

However, you would skip the step to specify a remote BBMD, unless one exists. This allows BACnet/IP devices on remote subnets to register with the station as foreign devices. You would not see other BBMDs in the **Broadcast Distribution Table**, but you could see them if other devices have registered as foreign devices in the **Foreign Device Table**.

Chapter 8 BACnet/SC (Secure Connect)

Topics covered in this chapter

- ◆ BACnet/SC and BACnet/IP
- ◆ Message handling
- ◆ Node certificates
- ◆ Setting up a secure hub
- ◆ Creating a direct node connection
- ◆ Setting up a secure port
- ◆ FAQ

BACnet/SC, added in Niagara 4.11, provides a link layer for securing the messages communicated among the Secure Connect nodes of a BACnet network. BACnet Secure Connect (BACnet/SC) uses TLS (Transport Layer Security) version 1.3 to provide strong data encryption and device authentication. With the increased likelihood of cyber attacks on critical infrastructure, including buildings, it is important to secure automation networks. BACnet/SC secures communication on TCP/IP networks using standard IT technologies that are used in critical applications, such as online banking.

TLS is the widely-available protocol that uses PKI (Public Key Infrastructure) keys to encrypt and decrypt data transmissions, and certificates to ensure that only authorized devices can communicate with each other.

With TLS protecting a BACnet network, companies can use a service, such as Qualys, which is a web application and cloud-based service used to detect data breaches and critical misconfigurations across a company's global hybrid environment. E-banking and other critical applications use such services to identify vulnerabilities including cross-site scripting (XSS) and SQL injection.

BACnet/SC and BACnet/IP

You can integrate BACnet/SC into a Niagara BACnet network to transmit standard BACnet messages. BACnet/SC can run over the same cables and routers used by BACnet/IP. BACnet/SC nodes that do not host a hub may make outbound connections to the public Internet through firewalls without additional software or hardware. A hub hosted in a Niagara station accepts inbound traffic and should not be connected to the public Internet.

In addition to providing the enhanced security required by building automation systems, BACnet/SC can eliminate much of the concern and work that non-standard systems require of IT departments.

Significant differences between BACnet/IP and BACnet/SC include these:

| Feature | BACnet/IP | BACnet/SC |
|-------------------------|---------------|-----------------------------------------------------------------------------------|
| Transport Protocol | UDP | TCP |
| Device authentication | none | TLS (version 1.3) |
| IP addresses | mostly static | static or dynamic |
| BACnet/IP/BBMD | used mostly | none |
| BACnet/SC hubs | none | a minimum of one per network |
| Encrypted communication | none | TLS (version 1.3 with option for 128-bit and 256-bit elliptic curve cryptography) |
| DNS | not used | supported |
| NAT tolerant | none | yes |

You can integrate BACnet/SC into a station and configure it, which includes having the appropriate certificates in place. For devices that only support BACnet/IP, a firmware upgrade is needed. You can send messages between a BACnet/SC network and a non-secure network such as BACnet/IP and MS/TP provided that you have SC-IP and/or SC-MS/TP routers in your network. Once the BACnet/SC network is configured, it is easy to update a device in a BACnet/SC, IP or MS/TP environment.

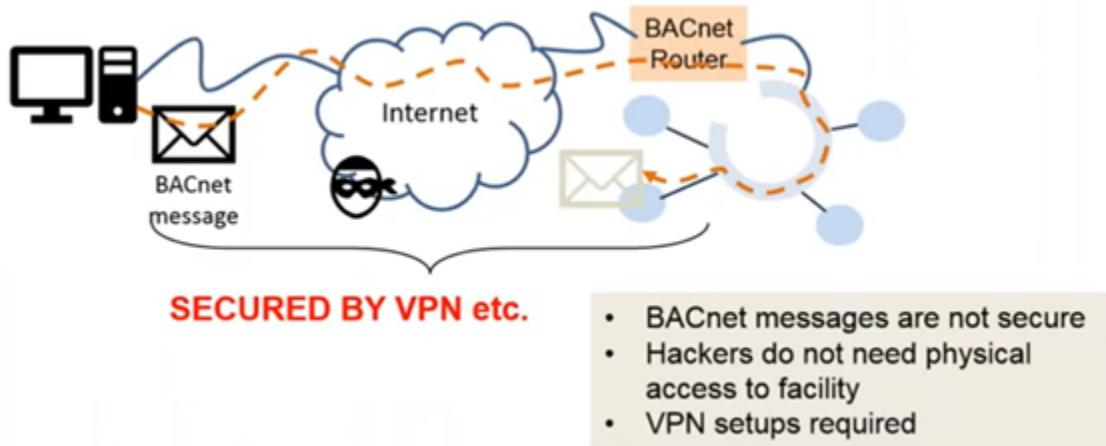
Message handling

BACnet/SC is not bound by IP network boundaries and subnet structures. Not having to depend on an IP subnet X or Y frees up network design across the IP infrastructure.

Traditional BACnet over the Internet

Even a network not connected to the Internet is vulnerable to a malicious attacker who has physical access to the facility.

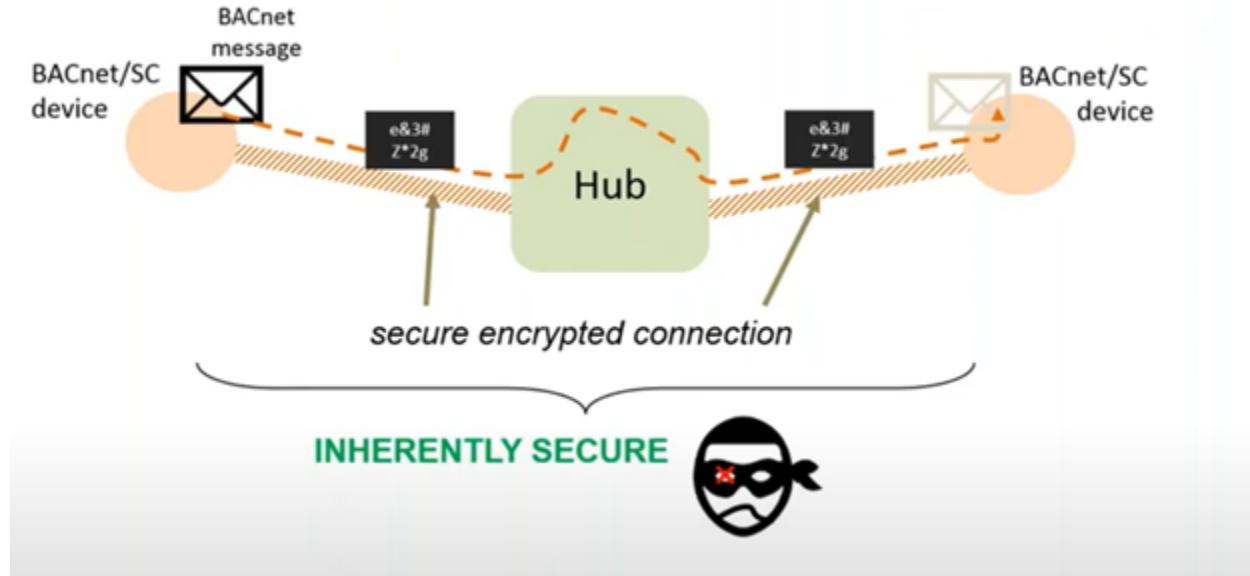
Figure 17 BACnet/IP message flow



Without BACnet/SC, a BACnet network requires a VPN (Virtual Private Network) to protect its traffic from hackers.

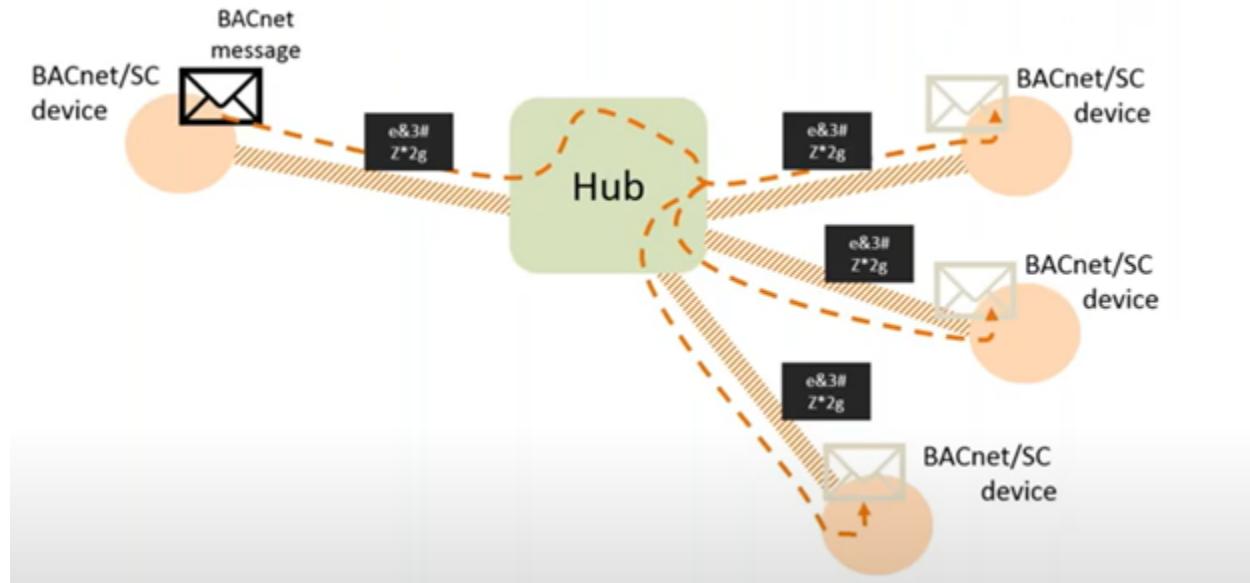
BACnet Secure Connect

Figure 18 BACnet/SC single message flow



The BACnet/SC link layer is compatible with other link layer types and supports all BACnet application and network layer messages. BACnet/SC does not change message content. Messages sent over a BACnet/SC network are encrypted and travel in a secure channel using a hub. This contrasts with messages sent over BACnet MS/TP (Multidrop Serial Bus/Token Passing) or BACnet/IP using UDP (User Datagram Protocol), which are not encrypted and could be sniffed and tampered with.

Figure 19 BACnet/SC broadcast messages



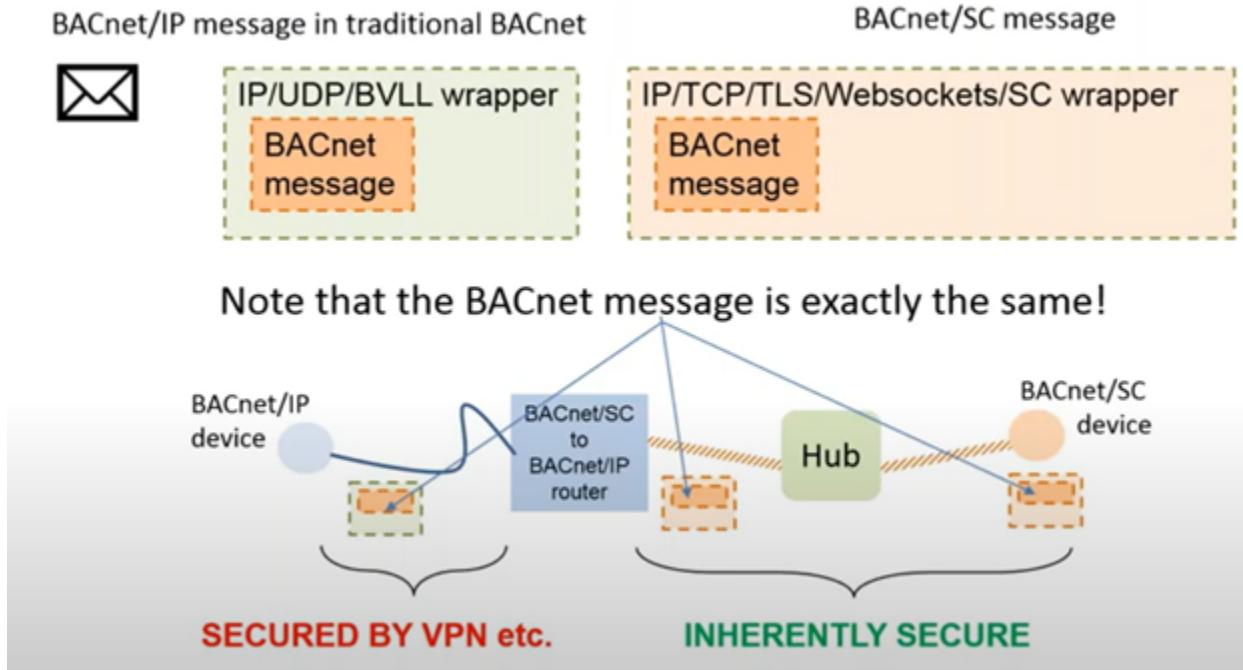
Messages that have multiple recipients are very common in building automation. With BACnet/SC, all devices join the network by connecting to the hub. The hub then broadcasts the 'who-is' and 'who-has' requests to all participants. This eliminates IP-level broadcast or multicast transmits through IP subnet boundaries. These broadcast messages do not require special firewall configuration. Since messages pass through without special configuration and a BACnet network maintains its communications pattern at the link level, the

network uses whatever IP infrastructure is available. The IP infrastructure itself is not important. Only HTTPS needs to be open.

Backward message compatibility

BACnet/SC provides a different kind of wrapper around the same message (orange boxes) than other link layers such as BACnet/IP.

Figure 20 BACnet message wrappers



BACnet/SC supports both traditional IP and secure connections (blue box).

NOTE:

The hub must accept inbound connections and if the hub is a Niagara station it should not be on the public Internet. Best practice is for all devices to be on the same VPN.

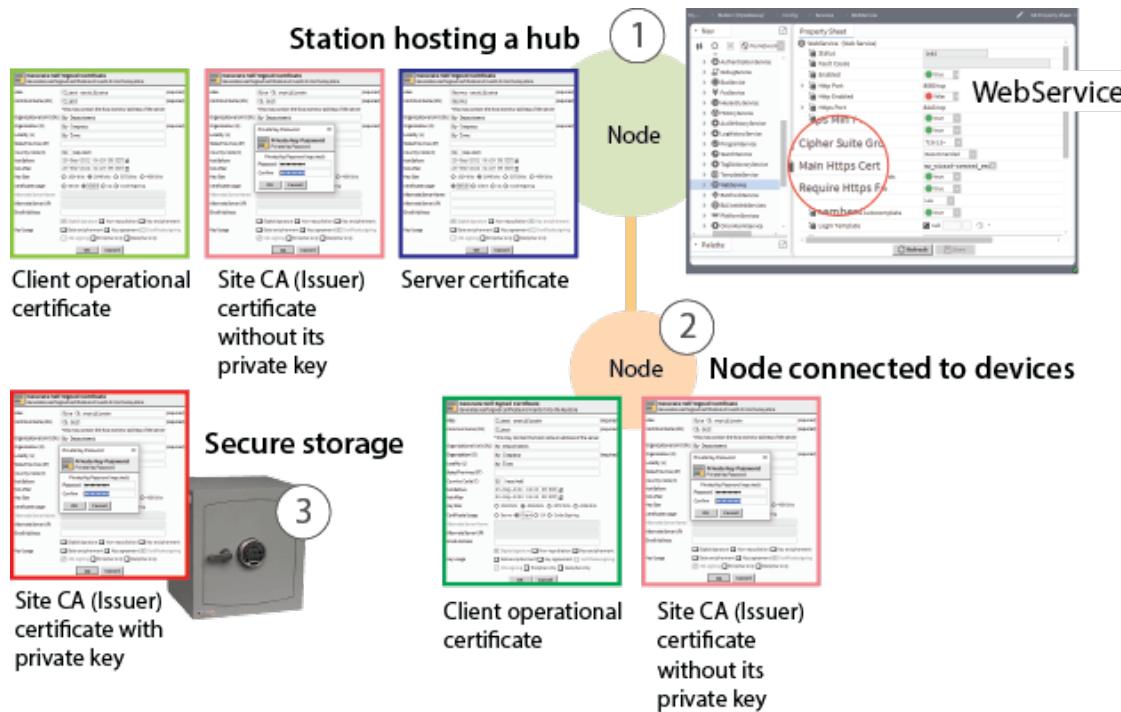
Node certificates

Certificates ensure authenticity among networked devices and are used for communication. A node can be a Supervisor station, remote controller station, or a device, such as a thermostat, HVAC, light, camera, etc.

To manage secure communication in a BACnet/SC network, you need three types of certificates:

- A single site (root or intermediate) CA certificate, also called an issuer certificate. The private key of this certificate signs the network's server and client certificates.
- A client certificate for each node, also called an operational certificate.
- A server certificate for the nodes that host a hub function or nodes that accept direct connections. In some cases, the server certificate can also be used as a client certificate.

This drawing identifies the certificates required to secure the nodes of the BACnet network. A Supervisor station has the same requirements as any other node. Devices at the edge also require certificates.



| | Function and comments | Required certificates | Where to configure | Property names |
|----------|----------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
| 1 | Initiate a local hub connection to the node's hub function or direct connections to other nodes | client (identified in green) | Expand BacnetNetwork → Bacnet Comm → Network → ScPort → Link and double-click Credentials . | Operational Certificate |
| | Verify the server certificate of another node when initiating a hub or direct connection. | site CA (issuer) without its private key (identified in light red) You may email this certificate. | | Issuer Certificate1 Issuer Certificate2 |
| | Verify the client certificate submitted by another node when accepting a hub or direct connection. | server | | Main Https Cert |
| 2 | Initiate a hub connection to a hub or direct connections to other nodes | client (identified in green) Each node has its own unique client certificate. | Expand BacnetNetwork → Bacnet Comm → Network → ScPort → Link and double-click Credentials . | Operational Certificate |
| | Verify the server certificate of another node when initiating a hub or direct connection. | site CA (issuer) without its private key (identified in light red). You may email this certificate. | | Issuer Certificate1 Issuer Certificate2 |
| 3 | Used to sign all client and server certificates | site CA (issuer) with its private key (identified in red) This certificate's private key is password protected. Once you finish signing your server and client certificates you must store this certificate securely. Do not email it to anyone. | Click Tools → Certificate Management . | N/A |

Creating a site CA certificate

This certificate, also called an issuer certificate, is a CA certificate (root or intermediate) used to sign all of the BACnet/SC operational certificates. This certificate, exported without its private key, verifies the authenticity of the server and client certificates.

Prerequisites: You have the required authority to create certificates. You are working in Workbench on a computer that is dedicated to certificate management, is not on the Internet or the company's LAN and is physically secure in a vault or other secure location.

- Step 1 Access the Workbench **Certificate Management** view by clicking **Tools→Certificate Management**.

The **Certificate Management** view opens to the **User Key Store**.

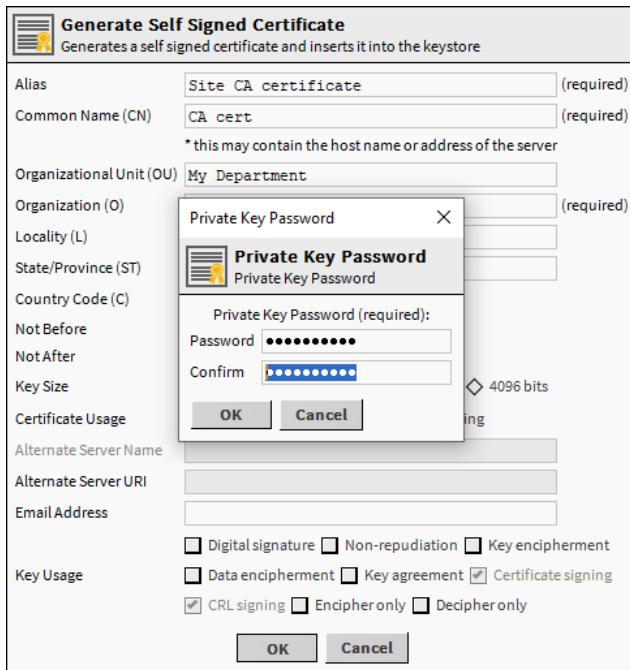
| Alias | Subject | Not After | Key Algorithm | Key Size | Valid |
|---------|----------|------------------------------|---------------|----------|-------|
| default | Niagara4 | Fri Dec 22 10:38:44 EST 2023 | RSA | 2048 | true |

This key store contains an auto-generated, self-signed default certificate that cannot be deleted and should be used for recovery purposes.

- Step 2 Confirm that you opened the Workbench **User Key Store** and click the **New** button at the bottom of the view.

NOTE: If you opened the platform/station **Certificate Management** view by mistake, you can still create a site CA certificate, but it will not be available to sign the other certificates.

The **Generate Self Signed Certificate** window opens.



All certificates begin as self-signed certificates. The Secure Connect site CA certificate may be a root certificate and remain self-signed. It can be signed by another CA certificate and change into an intermediate CA certificate.

Step 3 Fill in the form and click **OK**.

- Use **Alias** to identify this as a site certificate.
- The **Common Name (CN)** becomes the **Subject** (also known as the Distinguished Name). For a site certificate, the **Common Name (CN)** may be the same as the **Alias**.
- **Organization** should be the name of the company.
- Although **Locality** and **State/Province** are not required and are arbitrary, leaving them blank generates a warning message.
- The two-character **Country Code** is required and must be a known value, such as: US, IN, CA, FR, DE, ES, etc. (refer to the ISO CODE column at countrycode.org).
- Based on the **Not Before** and **Not After** dates, certificate validity defaults to a year.
- **Key Size** defaults to 2048. A larger key improves security and does not significantly affect communication time. The only impact it has is to lengthen the time it takes to create the certificate initially.
- For **Certificate Usage**, select **CA**.

The **Private Key Password** window opens.

Step 4 Enter and confirm a strong password, and click **OK**.

The system informs you that the certificate has been submitted. Soon the certificate appears behind the **Info** message in the **User Key Store** table.

Step 5 To continue, click **OK**.

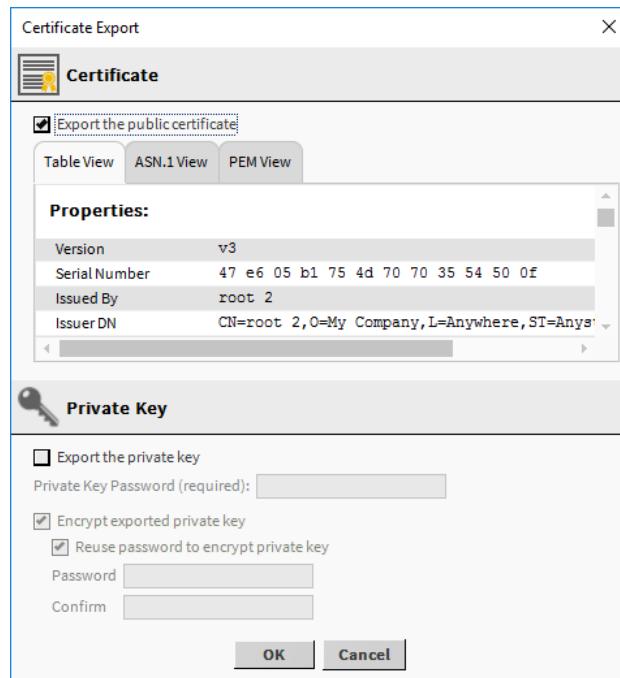
The root CA certificate now exists with both its keys in the Workbench **User Key Store**. From this location you can use it to sign other certificates (intermediate, server, and client).

NOTE: Since this certificate is not signed by any higher certificate authority, it is always identified with an exclamation icon (!). This is normal for a root CA and does not need to be corrected. Because the CA can be used to sign server and client certificates, and establish trust between BACnet devices, you must protect the computer (and thumb drive) on which it resides by keeping the computer off the Internet, corporate LAN, and most securely, in a locked physical location.

For this certificate to authenticate the certificates it signs, you now need to export it with only its public key and load it into the **Issuer Certificate 1** or **Issuer Certificate 2** properties.

- Step 6 Select the new site certificate and click **Export**.

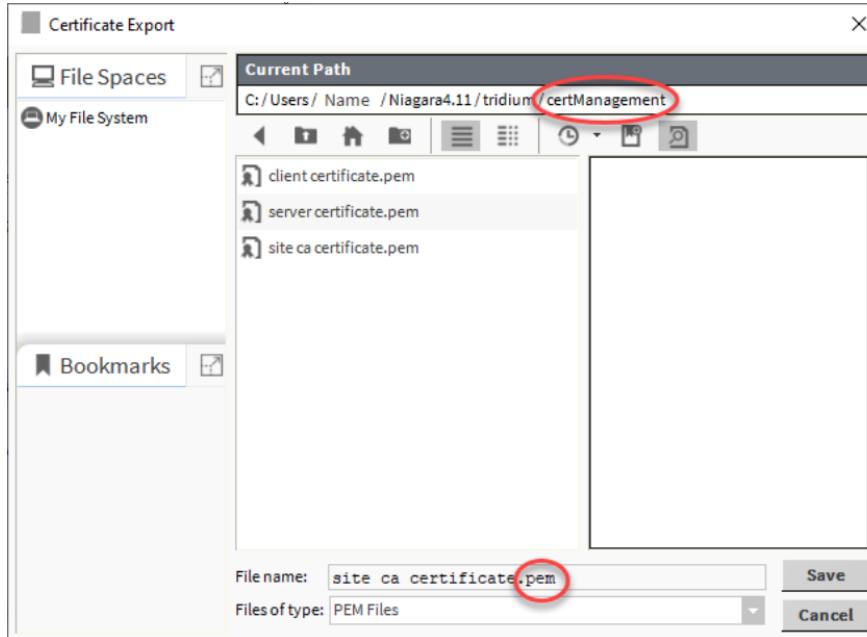
The **Certificate Export** window opens.



CAUTION: Do not click the check box to **Export the private key**. The only time you click this check box is when you are backing up the certificate to another location for safe keeping.

- Step 7 To create the site certificate that will be loaded into one of the **Issuer Certificate** properties, click **OK**.

The **Certificate Export** window opens with the file ready to export as a .pem file.



Notice the Current Path. This is where the system stores the exported certificate.

Step 8 Navigate to a certificate folder or location on a thumb drive, and click **Save**.

The system reports that it exported the certificate successfully.

Step 9 To complete the export, click **OK**.

Creating operational certificates

You create an operational certificate, which can be a client or a server certificate within a secure connect network.

Prerequisites: You have the required authority to create certificates. You are working on the device for which you want to create operational certificates.

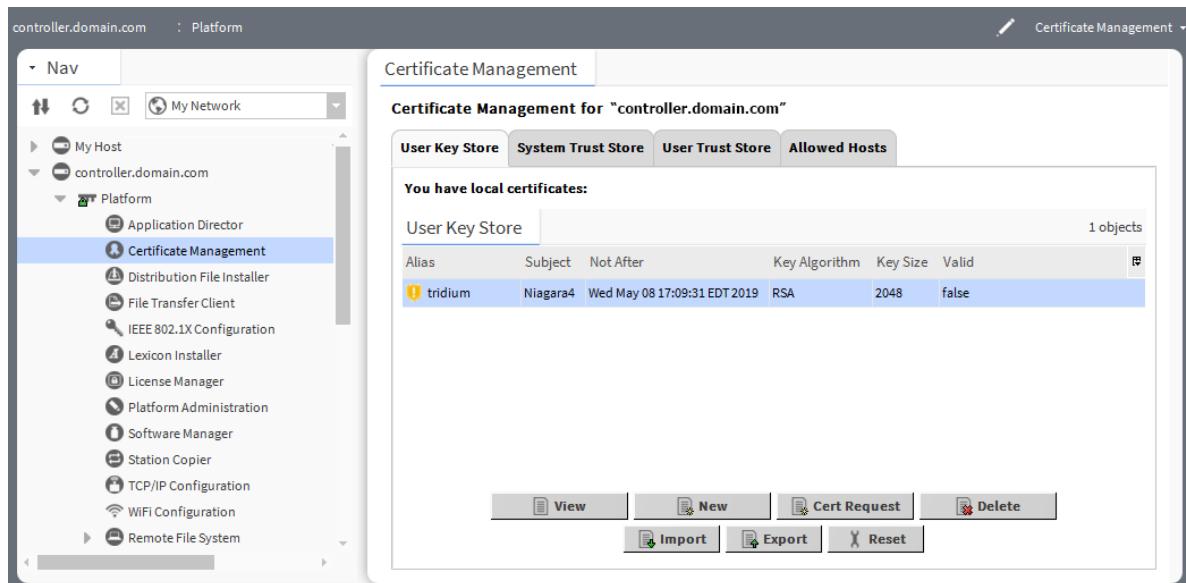
Client certificates and server certificates differ in their roles. Client certificates are used for nodes (ports) that do not have a hub function and can only initiate, not accept a connection to a hub function or direct connections to the node switch. You cannot use a client certificate in the web service for accepting connections. Server certificates created by Niagara's certificate tools can be used for nodes (hub ports) that can also host a hub, which means that they can initiate and accept a local connection to themselves and accept hub or direct connections from other nodes.

This procedure uses Niagara's tools running in a PC or remote controlled station.

Step 1 Do one of the following:

- Connect to your platform and double-click **Certificate Management** in the **Nav Container View**.
- In the **Platform** Nav tree, double-click **Certificate Management**.
- Connect to your station, expand **Config→Services→PlatformServices** and double-click **CertManagerService**.

The **Certificate Management** view for the device opens to the **User Key Store** for the platform stores.



These certificate management stores are for the PC or controller platform and station. They are different from the stores for Workbench.

Step 2 To create an operational certificate, click **New**.

The **Generate Self Signed Certificate** window opens.

| | | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|------------|
| Alias | node_3 | (required) |
| Common Name (CN) | node_3 | (required) |
| * this may contain the host name or address of the server | | |
| Organizational Unit (OU) | | |
| Organization (O) | Tridium | (required) |
| Locality (L) | | |
| State/Province (ST) | | |
| Country Code (C) | US | (required) |
| Not Before | 04-Nov-2021 11:57 AM EDT | |
| Not After | 04-Nov-2022 11:57 AM EDT | |
| Key Size | <input type="radio"/> 1024 bits <input type="radio"/> 2048 bits <input type="radio"/> 3072 bits <input type="radio"/> 4096 bits | |
| Certificate Usage | <input type="radio"/> Server <input checked="" type="radio"/> Client <input type="radio"/> CA <input type="radio"/> Code Signing | |
| Alternate Server Name | | |
| Alternate Server URI | | |
| Email Address | | |
| <input checked="" type="checkbox"/> Digital signature <input type="checkbox"/> Non-repudiation <input type="checkbox"/> Key encipherment <input type="checkbox"/> Data encipherment <input type="checkbox"/> Key agreement <input type="checkbox"/> Certificate signing <input type="checkbox"/> CRL signing <input type="checkbox"/> Encipher only <input type="checkbox"/> Decipher only | | |
| <input type="button" value="OK"/> <input type="button" value="Cancel"/> | | |

Step 3 Fill in all required information and click **OK**:

- Use **Alias** to identify this as an operational certificate (client or server certificate).
- The **Common Name (CN)** becomes the **Subject**, also known as the Distinguished Name. For an operational certificate, the **Common Name (CN)** may be the same as the Alias. If you use host-name validation, it's recommended that the **Common Name** is identical to the hostname or IP address of the device on which this certificate is generated.

- **Organization** is the name of the company.
- Although **Locality** and **State/Province** are not required, leaving them blank generates a warning message.
- The two-character **Country Code** must be a value such as: US, FR, DE. Refer to the ISO CODE column at countrycode.org.
- For certificate Usage, select **Client or Server**.

Your newly created operational certificate is now added to the **User Key Store** from where you can generate a certificate signing request (csr) for this certificate.

Creating a CSR for an operational certificate

A CSR (Certificate Signing Request) creates a .csr file for each operational certificate. This file can be signed by a site certificate.

Prerequisites: You have added operational certificates to the platform/station **User Key Store**.

Step 1 To open the platform/station certificate stores, do one of the following:

- Connect to your platform and double-click **Certificate Management** in the **Nav Container View**.
- In the **Platform** Nav tree, double-click **Certificate Management**.
- Connect to your station, expand **Config→Services→PlatformServices** and double-click **CertManagerService**.

Step 2 Select the operational certificate and click **Cert Request**.

The **Certificate Request Info** window opens.

Step 3 Confirm that the certificate properties are correct and click **OK**.

The system displays the **certManagement** folder for you to choose the location to store the CSR.

The **Alias** for the certificate is used as the file name of the CSR.

Step 4 Use the default folder, or select a different folder in which to store the CSR and click **Save**.

The system displays, CSR generation complete.

Step 5 To finish, click **OK**.

NOTE: Once you create a CSR, do not delete the original certificate from which you created the CSR. Later in the process you will import a signed certificate back into the **User Key Store** where its public key must match the private key of the original certificate. Creating a new certificate with the same name does not generate the same key pair and results in errors when you try to import the signed certificate. If it is absolutely necessary (for example, if the computer on which the certificate is stored is vulnerable), you may export the original certificate with its private key and import it into the **User Key Store** when you receive the signed certificate. But, ideally, you should leave the original certificate in the **User Key Store** of the original secure host.

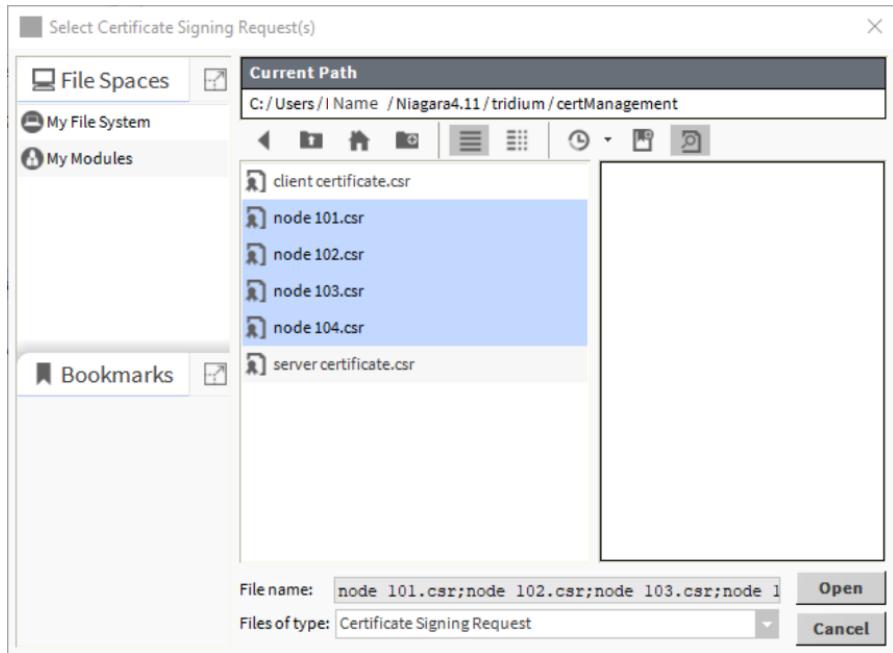
Signing multiple certificates

If you have to sign certificates for many controllers, remote stations, or Supervisor PC, signing each one individually can take a long time. This procedure helps you sign all your certificates at once.

Prerequisites: Your CA certificate performing the signing is in the Workbench **User Key Store** and you exported your certificates as CSR files (signing requests) to a folder on your hard drive.

Step 1 Navigate to **Tools→Certificate Signer Multiple Selection Tool**.

The **Select Certificate Signing Request(s)** window opens.



Step 2 Locate your certificate signing requests on your file system.

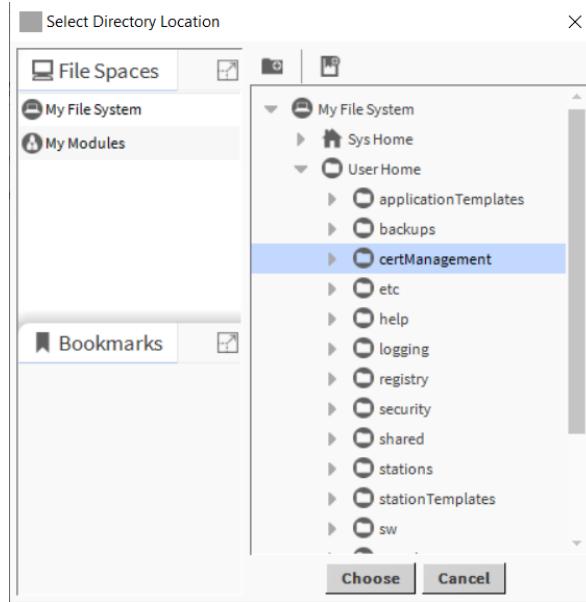
Step 3 Select the certificates to be signed and click **Open**.

The **Certificate Signer Multiple Selection Tool** view opens.

| Filename | Subject | Subject DN | Key Algorithm |
|--------------|----------|---------------------------------------------------------------------------|---------------|
| node 101.pem | Node 101 | C=US,ST=My State,L=My Town,O=My Company,OU=My Department,CN=Node 101 | RSA |
| node 102.pem | node 102 | C=US,ST=My State,L=My Town,O=My Organization,OU=My Department,CN=node 102 | RSA |
| node 103.pem | node 103 | C=US,ST=My State,L=My Town,O=My Organization,OU=My Department,CN=node 103 | RSA |
| node 104.pem | node 104 | C=US,ST=My State,L=My Town,O=My Company,OU=My Department,CN=node 104 | RSA |

Step 4 To sign the certificates, select the issuer using the **CA Alias** drop-down list, enter the **CA Password**, and click **Generate**.

The issuer's private key signs the certificates and opens the **Select Directory Location** window.



Step 5 Expand **User Home**, select **certManagement** and click **Choose**.

This is the default folder. You may choose a different folder.

The system confirms the number of certificates saved.

Step 6 To confirm completion, click **OK** twice.

Importing the signed certificates

Signing a certificate creates a .pem file, which is only intended for importing back into the **User Key Store** that contains the original certificate with the matching private key. For operational certificates, this is the platform/station **User Key Store** that originally created the certificate and CSR.

Prerequisites: You have the signed .pem file(s). The platform/station **User Key Store** is open.

Step 1 Click **Import**.

Step 2 Using the **Certificate Import** browser, locate and select the signed certificate's .pem file and click **Open**.

The **Certificate Import** window opens.

Step 3 Confirm that you are importing the correct certificate and click **OK**.

If the **Alias** of the certificate you are importing is not the same as the **Alias** of the certificate you are replacing, the system prompts you for the **Alias** of the certificate to replace.

Step 4 If needed, enter the **Alias** and click **OK**.

The green shield icon (✓) replaces the yellow shield icon (⚠) next to the certificate **Alias** in the **User Key Store** tab.

Step 5 Using the operating system, delete the .pem file(s) from the secure computer.

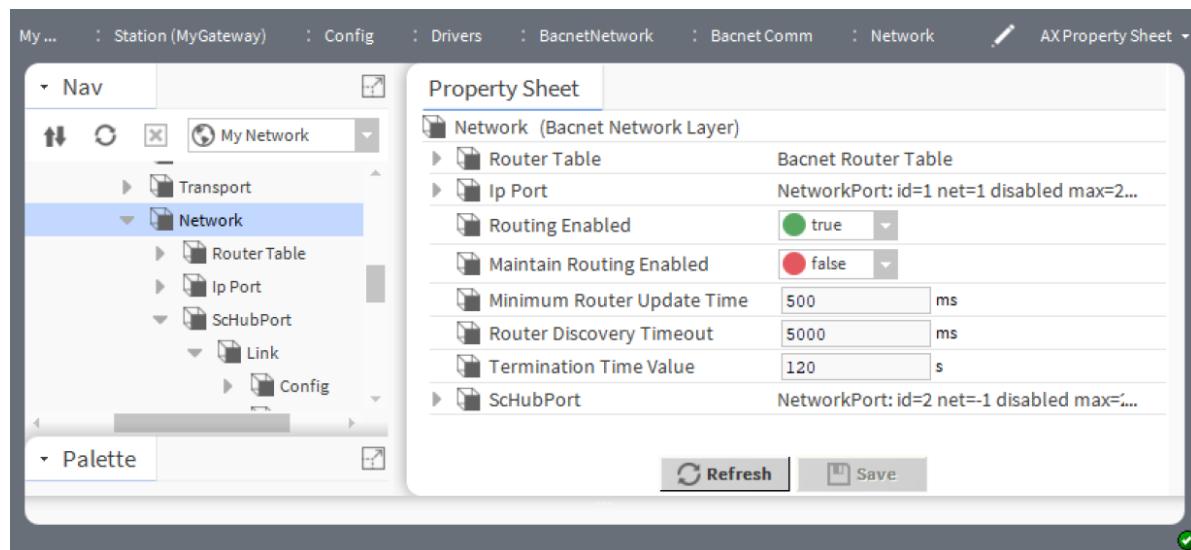
Setting up a secure hub

A secure primary or failover hub accepts connections from remote devices, usually remote controllers. One or two certificates in the hub authenticate server connections and encrypt these messages.

Prerequisites: You are working in Workbench on a PC or laptop computer connected to a remote controller station on a BACnet network. The **bacnet** palette is open. A client certificate for this node exists and you know where to find the issuer certificate.

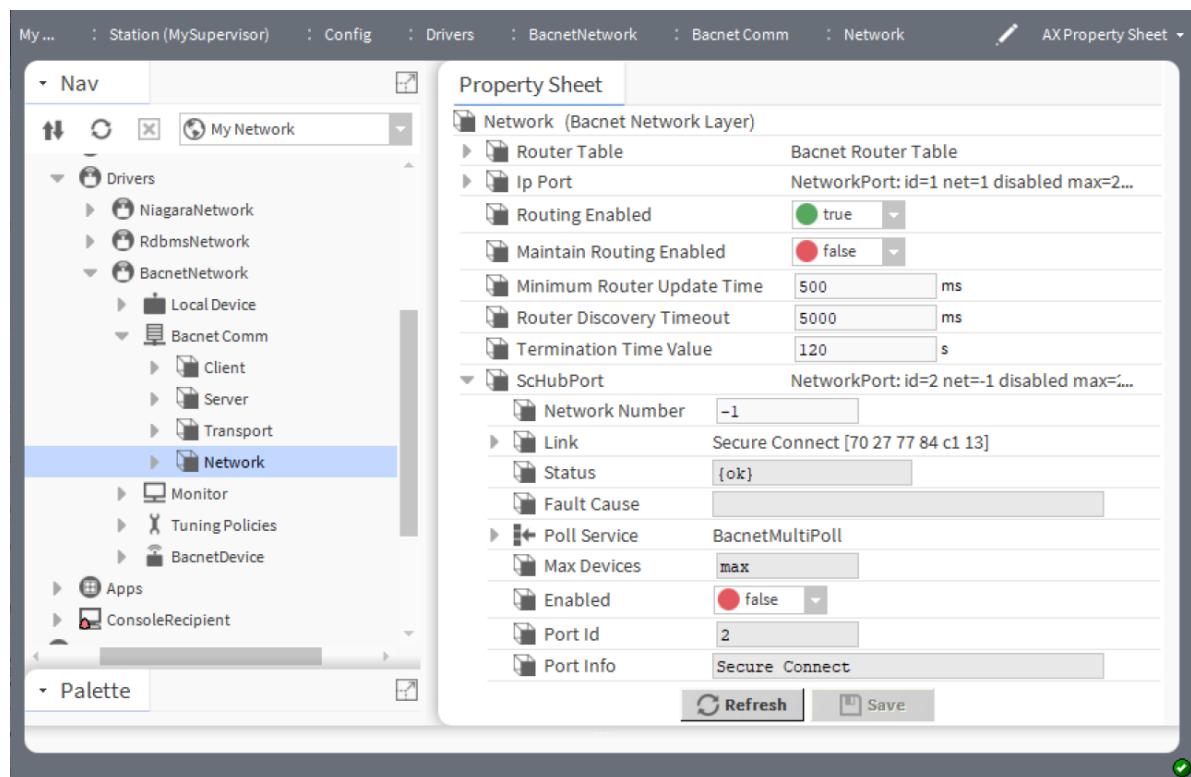
Step 1 To add a secure hub, expand **Config→Drivers→BacnetNetwork→BacnetComm** and double-click **Network**.

The **Network (Bacnet Network Layer) Property Sheet** opens.



Step 2 In the **bacnet** palette, expand **NetworkPorts**, select **ScHubPort**, add it to this **Property Sheet**, enter a name for this **ScHubPort** in the **Name** window that opens, and click **OK**.

The system adds the **ScHubPort** component to the **Property Sheet**.



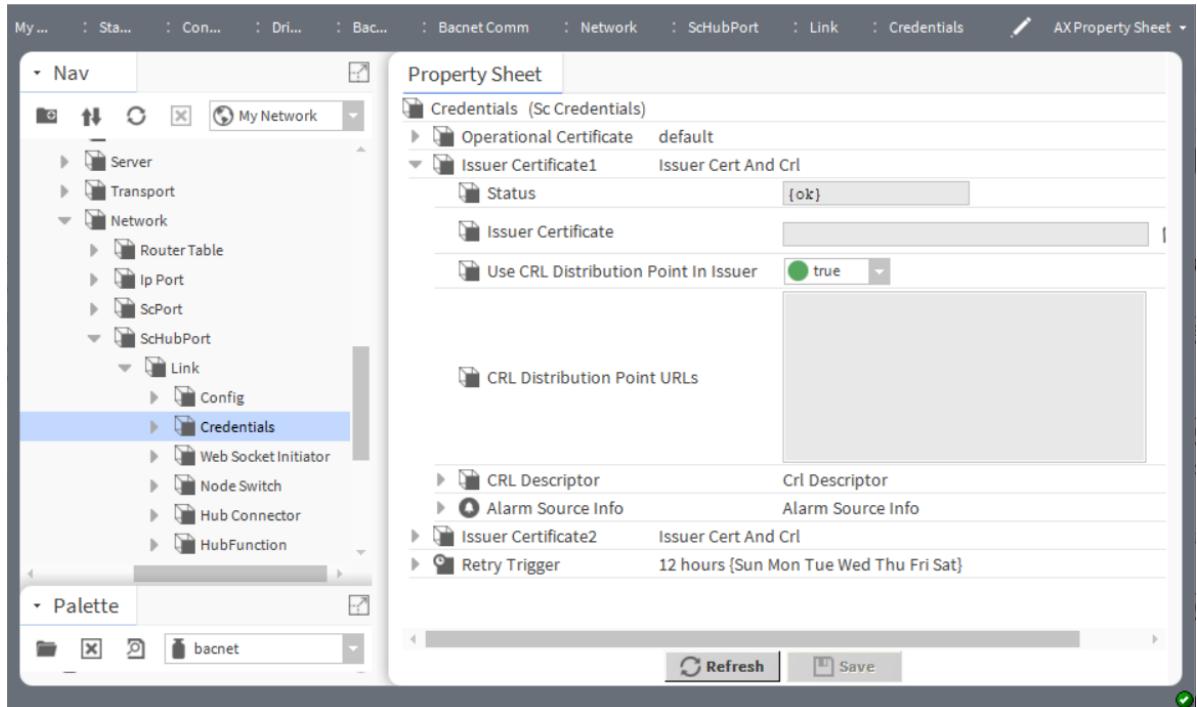
Step 3 Expand **ScHubPort** and enter the **Network Number**.

The driver automatically generates the **Vmac Address** (visible when you expand **Link**).

Step 4 To assign the operational (client) certificate, click **Link**, click **Credentials**, expand **Credentials** and select the certificate's alias from the **Alias** drop-down list.

The hub's operational certificate is used to make a local connection from a node that hosts the hub function.

The **Credentials** properties open.

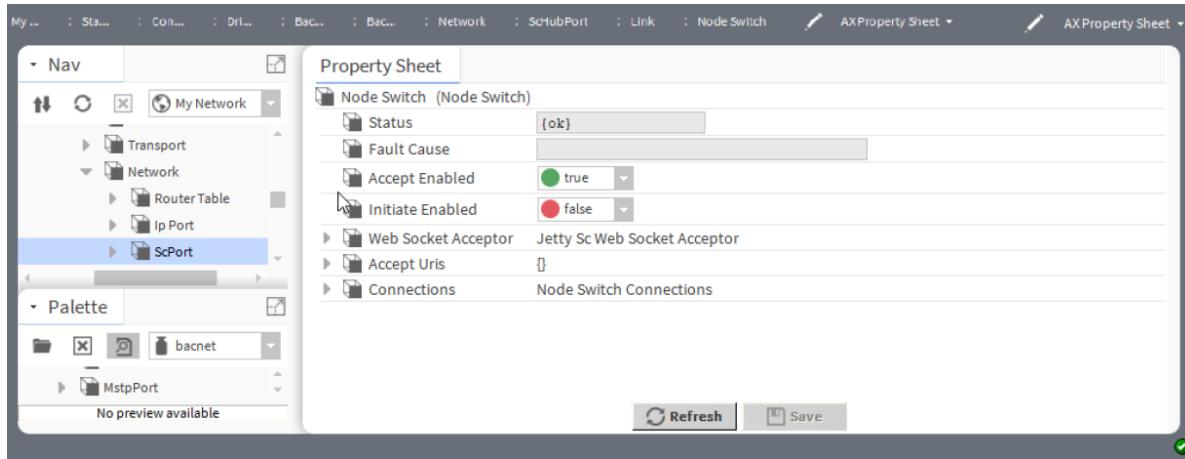


Step 5 To assign an issuer certificate (CA), expand **Issuer Certificate1**, click the folder icon to the right of **Issuer Certificate**, use the **File Chooser** to select the exported public certificate for the Secure Connect site CA certificate and click **Open**.

An issuer certificate is an exported Secure Connect site CA certificate without its private key. This certificate verifies a remote device's authenticity when it makes a connection to the hub.

Step 6 To specify the behavior of the node that hosts the hub towards incoming and outgoing direct connections, double-click **Link** in the Nav tree and expand **Node Switch**.

The **Node Switch** properties expand.



Step 7 Set Accept Enabled and the Initiate Enabled to true as needed and click Save.

This configures the node that hosts the hub to accept and/or initiate direct connections.

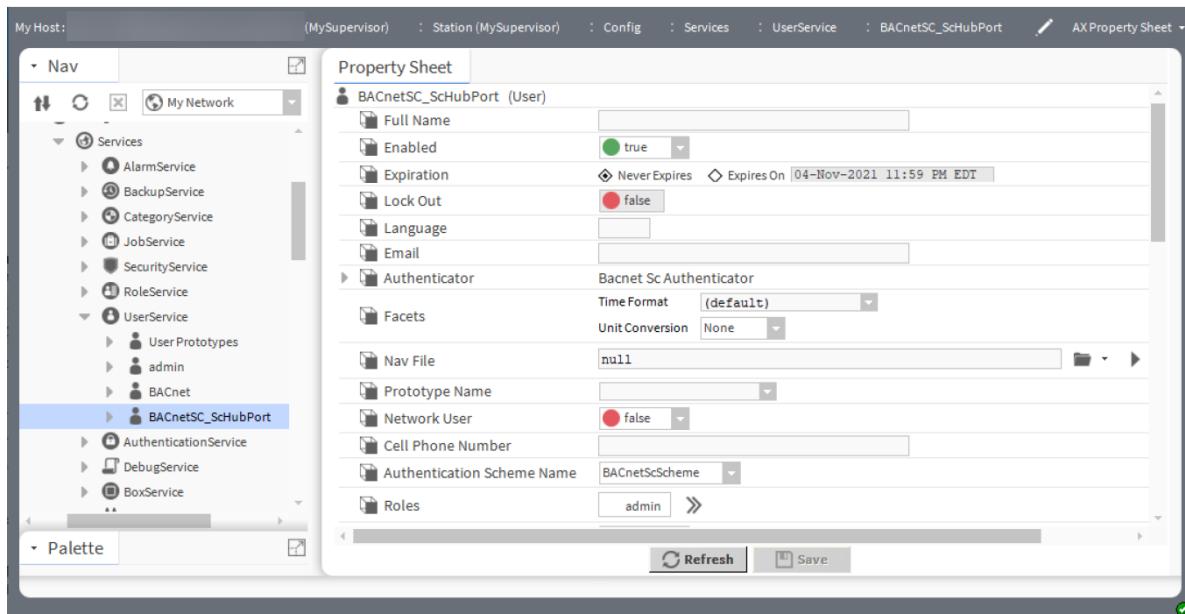
Notice that **Fault Cause** displays a message indicating that no BACnet/SC user is associated with this SC port. You will fix this condition shortly.

Step 8 To enable the hub function, double-click HubFunction in the Nav tree click or expand HubFunction, set Enabled to true and click Save.

Again, **Fault Cause** indicates that no BACnet/SC user is associated with this SC port.

Step 9 To associate a user, right-click HubFunction and select Actions→Add Sc User.

This action simultaneously adds a **BACnetSC_ScHubPort** user to the **UserService** and configures this user's **Authenticator** property with the ORD that points to the secure hub port you just configured.



To access the Property Sheet shown, expand **Config→Services→UserService** and double-click **BACnetSC_ScHubPort**.

This is a machine user required by BACnet Secure Connect. This user's sole purpose is to associate an incoming request with a secure connect port. Do not configure this user as you would an administrator or other human user. It must not have any permissions in the station.

Adding a hub function to a Secure Connect port

The **ScHubPort** is pre-configured with its **Hub Connector's Primary Connection** pointing to itself, its **Hub Address** set to `localhost` and its **Hub Path** set to `hub`. This procedure shows you how to add a **Hub Function** to a standard **ScPort**.

Prerequisites: The Primary Connection is configured. The **bacnet** palette is open.

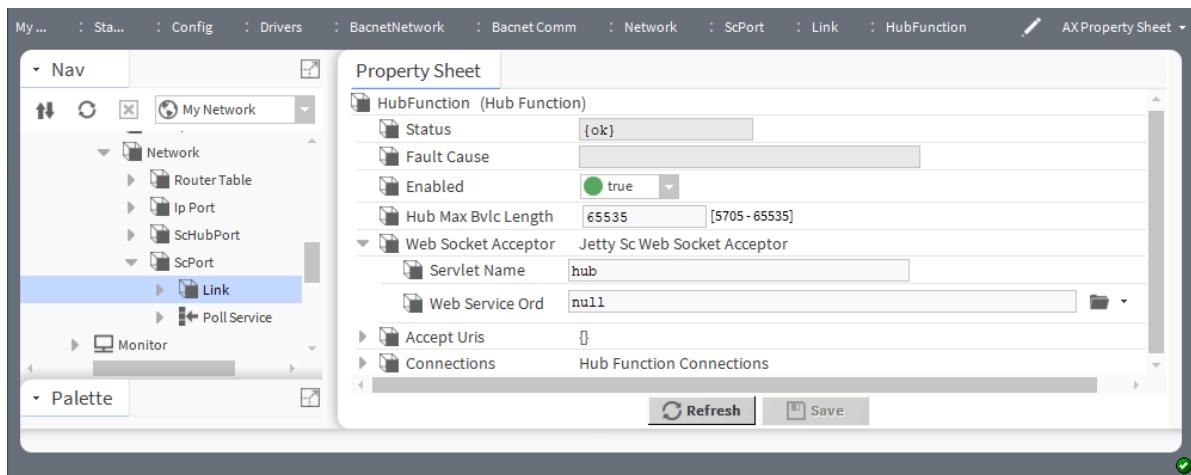
Step 1 Add a plain **ScPort** to the **Network** layer in the station (**BacnetNetwork→Bacnet Comm→Network**).

The difference between an **ScPort** and **ScHubPort** is that the **ScPort** does not have **Hub Function**.

Step 2 Start by configuring the port: double-click **ScPort**, expand **Link**, configure **Network Number**, expand **Credentials**, set the **Operational Certificate** and **Issuer Certificate(s)** and click **Save**.

Step 3 To add the **Hub Function**, expand the **SecureConnect** node in the **bacnet** palette, drag a **Hub Function** component to the **Link** layer and click the **HubFunction** you just added.

The **HubFunction Property Sheet** opens.

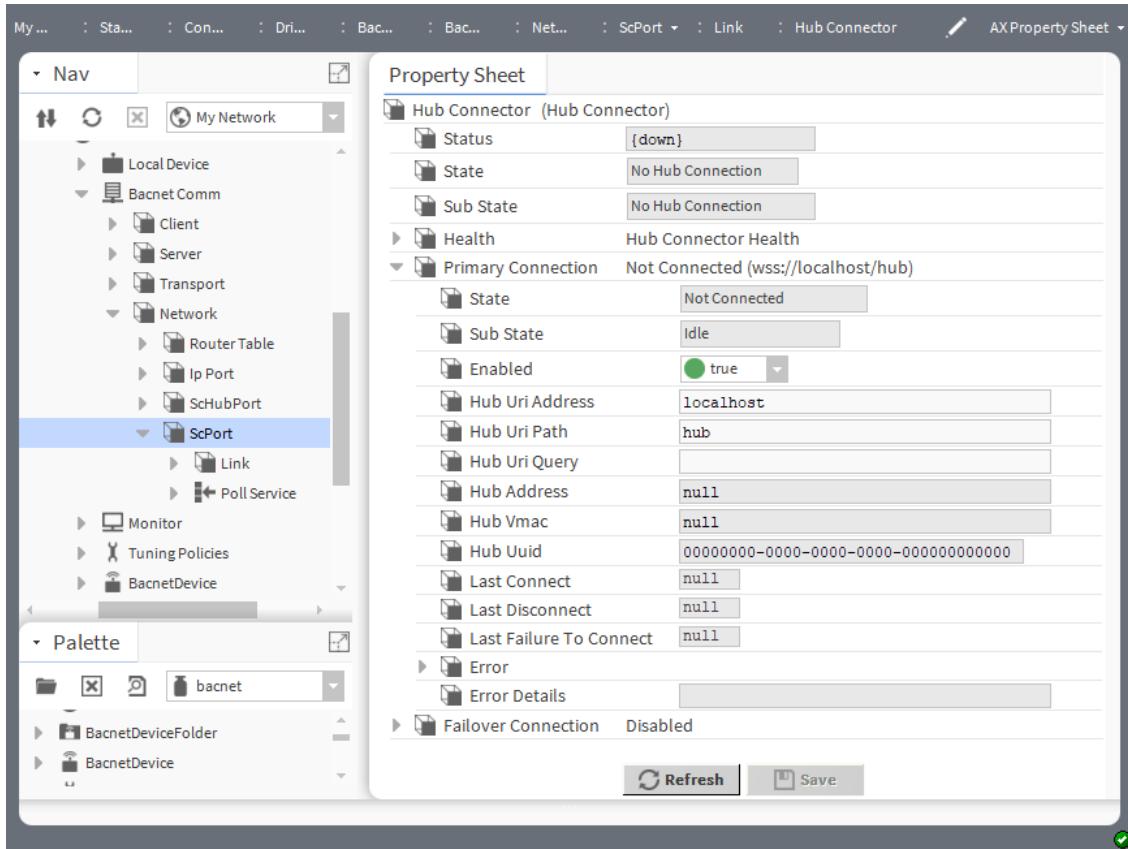


Step 4 Set **Enabled** to `true`, expand **Web Socket Acceptor**, set **Servlet Name** to `hub` and click **Save**.

Step 5 To connect the **Primary Connection** to the local hub, do one of the following:

- Double-click **ScPort** in the Nav tree, click the **Hub Connector** and expand **Primary Connection**.

The **Hub Connector Primary Connection** properties open.



Change **Hub Uri Address** to `localhost` and **Hub Uri Path** to `hub` and click **Save**.

- Right-click **Hub Function**, click **Actions→Configure Local Hub Connector**.
The **Configure Local Hub Connector** window opens.
Select the Primary or Failover connection from the drop-down list and click **OK**.

Configuring the WebService

This procedure configures the hub station's default **WebService** with a server.

Prerequisites: The station's **User Key Store** contains a certificate signed by your site Secure Connect CA certificate with which to authenticate incoming hub or direct connections.

Configuring the SC server certificate as an additional certificate is not recommended because BACnet devices may not include an SNI header. You can configure additional server certificates in the **WebService** to be presented to browser connections using the **AdditionalHttpsCerts** component from the **web** palette. The certificates presented to browsers are signed by a CA certificate in the browser's **System Trust Store**.

Step 1 Expand **Station→Config→Services** and double-click **WebService**.

The **WebService Property Sheet** opens.

Step 2 Select the hub's server certificate from the **Main Https Cert** drop-down list and click **Save**.

This configures the **WebService**.

The server certificate assigned to the **WebService** must be signed by the secure connect site CA certificate. In the **WebService**, you can configure additional certificates that are signed by a CA certificate in the system and/or a Browser Trust Store. The **Subject Alternate Names** property in these additional certificates is configured to present the certificates to browser connections based on an SNI header in the browser's connection request.

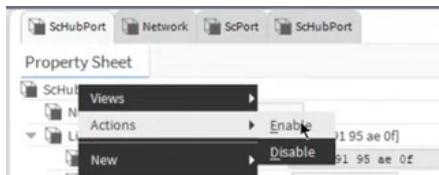
Enabling a secure hub connection

The final step when setting up a secure hub is to enable a secure connection.

Prerequisites: You set up the credentials for **ScHubPort** and under **WebService**, you configured the server certificate for connections to be accepted.

Step 1 To enable the hub connection, navigate to the **ScHubPort** node in the Nav tree.

Step 2 Right-click **ScHubPort** and select **Actions→Enable**.



The driver enables the connection.

Step 3 To confirm that the state of the connection, expand **Link** and double-click **Hub Connector**.

The **Hub Connector State** displays that the hub is connected or provides an error message if the connection was unsuccessful.



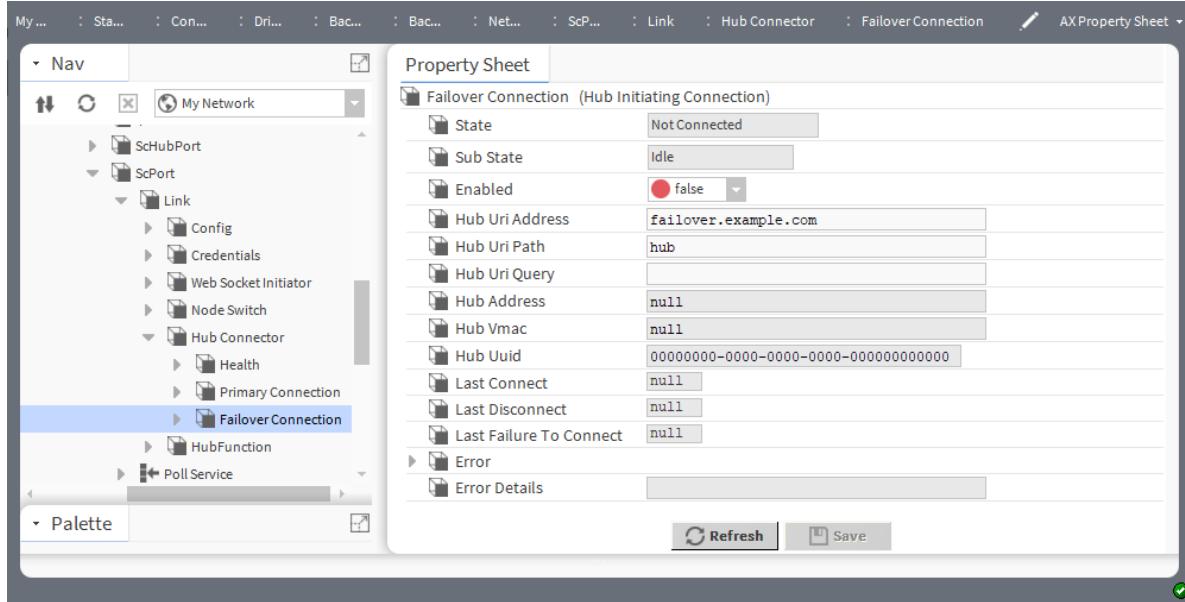
Setting up a failover hub

A failover hub backs up the functionality of a primary hub. A node (device), that cannot connect to the primary hub due to hub failure or maintenance, connects instead to the failover hub. When the primary hub comes online again, the node automatically switches back to the primary hub.

Prerequisites: You are working in Workbench connected to hub station.

Step 1 To locate the failover **Property Sheet**, in the Nav tree expand **Config→Drivers→BacnetNet-work→BacnetComm→Network→ScHubPort→Hub Connector** and double-click **Failover Connection**.

The **Failover Connection Property Sheet** opens.



Step 2 Set the **Enabled** property to **true**.

Step 3 Enter a value for **Hub Uri Address** and click **Save**.

This activates the **Failover Connection**, which takes over automatically if the primary connection fails.

Creating a direct node connection

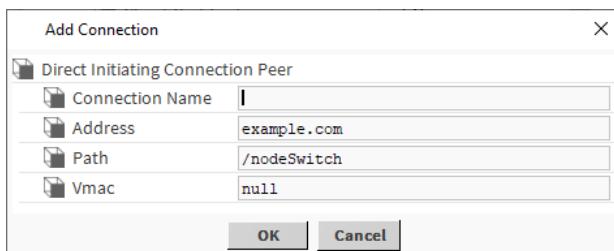
The node switch can accept and initiate direct connections with other BACnet/SC nodes. This procedure demonstrates how to connect two nodes, as an example, Node 1 and Node 2.

Prerequisites: The nodes have been enabled and configured. The **bacnet** palette is open.

Secure Connect supports only secure web sockets.

Step 1 To create a connection from Node 2 to Node 1, expand Node 2, expand **Link→Node Switch** and do one of the following:

- Right-click **Connections** and choose **Actions→Add Connection**. an **Add Connection** window opens.



- In the palette, expand **SecureConnect**, drag an **InitiatingConnection1** component from the palette to the **Connections** folder under **Node Switch** and double-click **InitiatingConnector1** in the Nav tree. The **Direct Initiating Connection Property Sheet** opens.

Step 2 Configure the following connection properties and click **OK** or **Save**.

- Connection Name** provides a name for the connection.
- Address or Peer Uri Address** is the station address (Universal Resource Identifier), for example, Localhost.

- **Path or Peer Uri Path** identifies the **Node Switch** path.

This adds a New Initiating Connection.

- Step 3** To complete the connection, right-click it and select **Action→Connect**.

This establishes communication from Node 2 to Node 1.

- Step 4** To go back to Node 1 and verify the connection, open the **ScHubPort Property Sheet** and confirm that under **Node Switch** Node 1 has accepted the connection.

- Step 5** To disconnect any connection, right-click the connection and select **Action→Disconnect**.

This disconnects the communication.

- Step 6** To clean up Idle Accepted connections from the system after a specified time, expand **Connections** and set **Enable Idle Cleanup** to true.

The default clean-up time is 1 hour for connections in a hub function and 24 hours for connections in the node switch. You can change these default values as needed.

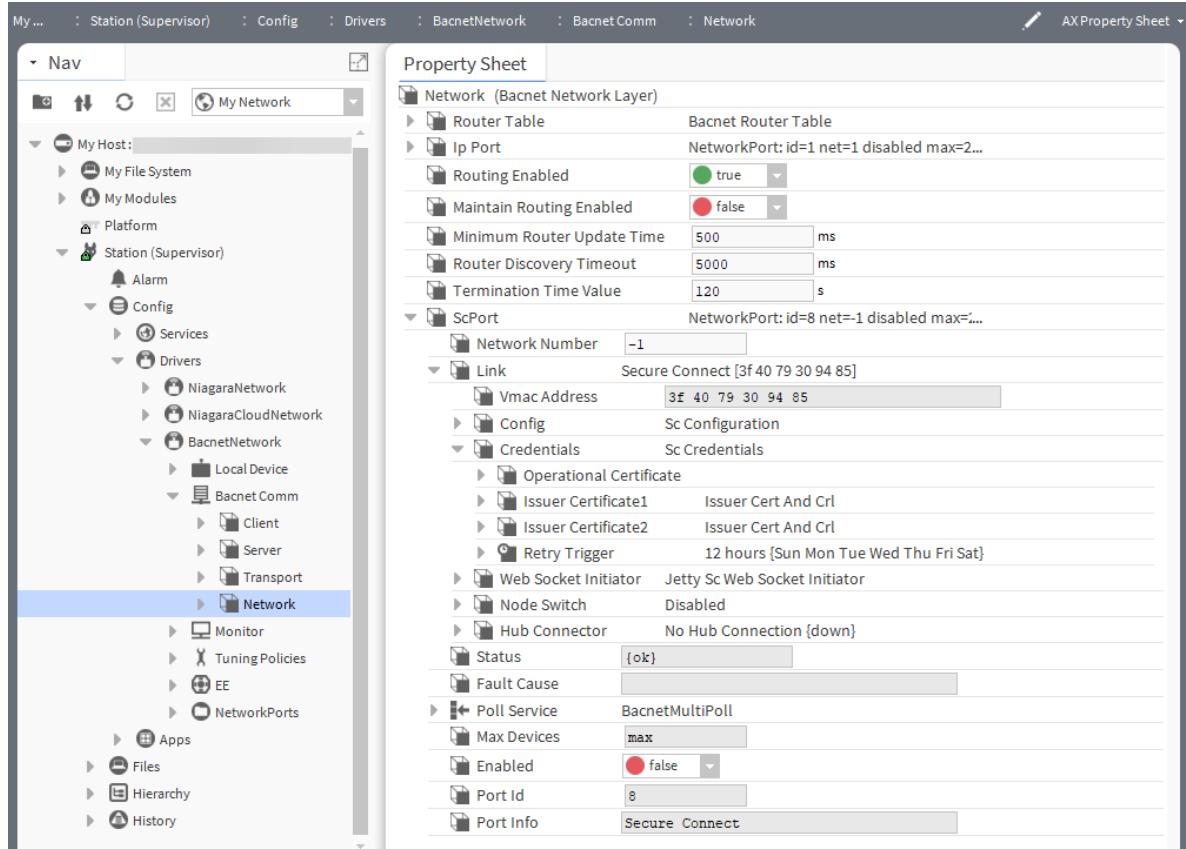
Setting up a secure port

The BACnet **ScPort** component uses a client certificate (operational certificate) to initiate hub connections with a primary or failover hub or to initiate direct connections with other BACnet/SC nodes. This procedure sets up the security in a single remote controller (node). Each individual device on your BACnet network also requires secure communication. Refer to the device manufacturer's documentation to configure devices that do not run Niagara 4.

Prerequisites: You configured the local device for this remote controller. You are working in Workbench running on a PC or laptop computer and are connected to the remote controller station.

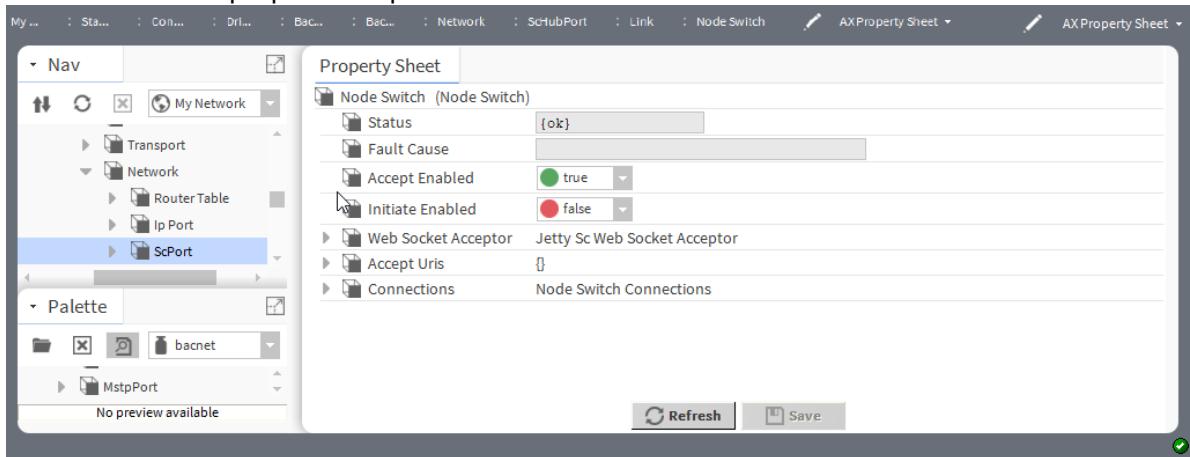
- Step 1** To add the secure port, open the **bacnet** palette, expand the **NetworkPorts** folder in the palette, add an **ScPort** to the **Network Property Sheet**, enter a name for this **ScPort** in the **Name** window that opens, and click **OK**.

The system adds the **ScPort** component to the **Property Sheet**.



- Step 2** Double-click the port you just added, configure the **Network Number** and click **Save**.
- Step 3** To assign the operational (client) certificate, expand **Link**→**Credentials**→**Operational Certificate**, select the certificate's alias from the **Alias** drop-down list and provide the certificate's associated password, which is either the unique password or the global certificate password.
- Step 4** To assign an issuer certificate (CA), expand **Issuer Certificate1**, click the folder icon to the right of **Issuer Certificate**, use the **File Chooser** to select the exported public certificate for the Secure Connect site CA certificate and click **Open**.
An issuer certificate is an exported Secure Connect site CA certificate without its private key. This certificate verifies a remote device's authenticity when it makes a connection to the hub.
- Step 5** To specify the behavior of the node that hosts the hub towards incoming and outgoing direct connections, expand **Node Switch**.

The **Node Switch** properties expand.



Step 6 Set **Accept Enabled** and the **InitiateEnabled** as needed and click **Save**.

Setting **Accept Enabled** to **true** triggers the **Fault Cause** to report No BACnet/SC user associated with this link layer.

Step 7 To associate a user, right-click **HubFunction** and select **Actions→Add Sc User**

FAQ

BACnet/SC is designed to meet the requirements of IT departments for ease of installation, standard security best practices and minimal ongoing support.

What is a BACnet node?

A BACnet/SC node can be a simple device, such as a thermostat, or a more sophisticated device that routes to an existing BACnet system, or it could be the main workstation for the entire facility. A node connects to the primary hub from which it receives broadcast and unicast messages. Or it can directly connect to one or more nodes to receive unicast messages from that node only.

What happens when a certificate expires?

A device with an expired certificate cannot connect to the hub. Device certificates need to be updated in a timely manner. This can be an administrative challenge especially if protecting your installation is extremely important.

Does a hub re-route messages or decode them?

The TLS connection from the device terminates at the hub. The hub decodes the message, then encrypts it again to send it on to the next connection. Decoding lets the hub read the destination address and properly forward the message to the correct recipient. A hub receives messages from only trusted devices.

What about the BBMD (BACnet/IP Broadcast Management Device)?

BACnet/SC requires no BBMD configuration with static IP addresses. The hub takes part of this functionality in that it distributes the broadcast. The hub is the heart of the BACnet/SC network. It does not require a static IP address to pass data through a secure web socket using a URI (Universal Resource Identifier). The host can use a regular URI DNS name and even a resource path to locate the hub and make a web socket connection.

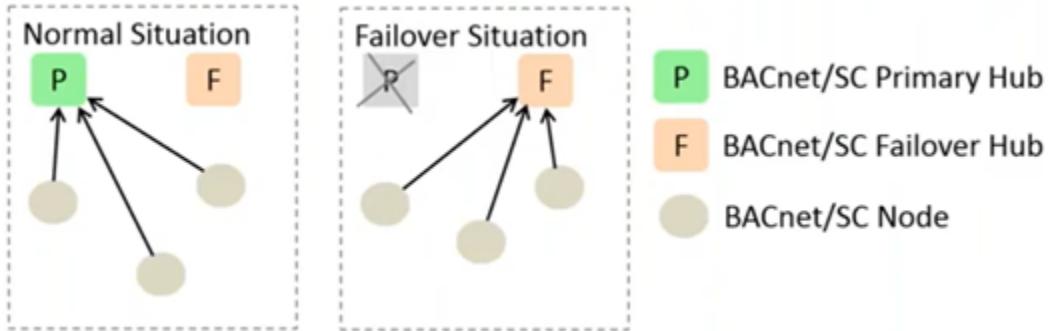
Can a network have different types of hubs?

A hub may be a controller, a BACnet router, a Supervisor PC or a cloud service. With cloud deployment, a hub may be a virtual machine in a data center.

What happens if a hub fails?

If the primary hub fails, a redundant, failover hub automatically takes over.

Figure 21 Redundant hub



The failover hub takes over when the device node cannot reach the primary hub. A node is configured to identify both the primary and failover hubs. The use of a failover hub is optional, but recommended.

When the primary hub is available again, the network automatically reconnects to it.

Should we get rid of your BACnet/IP with VPN?

Using a VPN is more secure than standard BACnet/IP. You should upgrade your networks one step at a time. This is possible because messages that are sent over a BACnet/SC link layer can be routed to and from BACnet/IP link layers.

What about scalability?

The number of devices a BACnet/SC supports depends on the hub. A large installation may have multiple hubs. Certificate management is designed to support large numbers of hubs and devices.

Chapter 9 Troubleshooting

Topics covered in this chapter

- ◆ Manually adding a skipUploadslot
- ◆ Rebooting a host

This topic provides suggestions for some issues you may encounter.

I configured Stale Time in my tuning policies for 10 minutes, but the points in my graphic go stale well before 10 minutes are up.

The length of time a point value remains fresh depends on the stale time and on the poll cycle time. For each point, the system measures the stale time from the last time the point was updated. A 10-minute stale time, and an 8-minute poll cycle time can result in point values that are nearly eight minutes old. If you stop polling, those points begin going stale roughly two minutes, not 10 minutes later.

This can be confusing when viewers expect that the points in a graphic should stay up for the 10 full minutes (or the configured Stale Time) before going stale. However, the actual time depends on how long ago they were last polled.

After discovering devices, some appear highlighted in orange.

The orange highlight indicates that the device is in fault. Often this is because of a duplicate Device ID (same Object_ID in the Device object of multiple devices). This is illegal in BACnet; each device must have a unique Object_ID for its device object.

You can still add such devices to the station, however, all but one will have a Device ID of -1 (unassigned), and be unreachable. Devices must be locally re-configured to have unique Device IDs. Then, in the station, each **BacnetDevice**'s corresponding (**Config**) Device Object must have its **Object Id** property set to match accordingly. Or, delete the devices, re-discover, and add them again.

My station takes a long time to start up.

Upon station startup, the BACnet driver automatically uploads all Device object properties for each represented BACnet device, after which the station's **BacnetNetwork** is fully initialized. Typically, such property uploads are inconsequential if there are mostly BACnet/IP or BACnet Ethernet devices.

However, if a **BacnetNetwork** includes many MS/TP devices, a long delay at station startup may result. During all these Device object uploads, more critical point data reads and writes cannot occur.

Any **BacnetDevice** with a Boolean slot named **Skip Upload** with a value of `true`, skips this upload at station startup. At any time you can still manually invoke the **Upload** action on any **BacnetDevice** to refresh these values—typically, most device object properties have static values.

One or more of my proxy points reports a status, such as disabled, but the driver reports the opposite status: enabled.

BACnet proxy points are unique from proxy points in most other field bus drivers because the BACnet protocol reports its own native abnormal status for data objects. The Framework learns about a BACnet point's abnormal status from the **Status_Flags** property of the BACnet object.

Possible abnormal BACnet Status_Flags include the following:

- IN_ALARM — appears as `{alarm}` in the Framework
- FAULT — appears as `{fault}` in the Framework
- OVERRIDDEN — appears as `{overridden}` in the Framework
- OUT_OF_SERVICE — appears as `{alarm, overridden}` `outOfService` in the Framework

To get this native status, you do not have to create a proxy point expressly for the status flags of a BACnet object. By default only the selected property is polled and reflected in the **Read Value** of the **ProxyExt**. However, you can include the polling of other data (including statusFlags), by adding to the facets of the point.

When you add statusFlags to facets, the driver merges the native BACnet point status with the Framework's status. Without adding the statusFlags key to the point's facets, the driver returns only the default Framework status for the proxy point. For example, if the BACnet object is in alarm, the Framework may report it as `{ok}`.

The default behavior of BACnet proxy points using COV (Change of Value) instead of polling, matches polled-point behavior for status.

I exported data from a Supervisor station to a remote schedule object and immediately the object went into fault.

By default, the Framework exports all schedule data to the BACnet schedule object. Depending on the target BACnet device's implementation by vendor, some properties of its schedule objects may be read-only or may not exist. For example, a BACnet schedule object may allow writes to its weekly schedule events, but not to its exception schedule (special events, if they are read-only or perhaps do not even exist). The object may not even have a weekly schedule or an exception schedule (a Schedule must only have one or the other; it may have both). Finally, the object's **Priority_For_Writing** property may be read-only. A fault occurs if a target object cannot handle the data exported to it.

Manually adding a skipUploadslot

To speed station startup in networks with multiple MS/TP devices, adding a skipUploadslot and setting its value to `true` can speed processing. This procedure documents how to add a slot using the slot sheet. Alternatively, you can use the **Batch Editor** of the station's **ProgramService** to do the same on multiple selected **BacnetDevices**.

- Step 1 Go to the slot sheet of the **BacnetDevice** (typically, one for an MS/TP device).
- Step 2 Add a slot with **Name**: `skipUpload` and choose **Type**: `baja, Boolean`.
- Step 3 Go to the property sheet of the **BacnetDevice** and set the value of this new property to: `true`.
- Step 4 **Save** these changes.
- Step 5 Change the **uploadOnStartup** property on the **BacnetNetwork** property sheet from `false` (default) to `true`.

This automatically creates a hidden skipUploadslot, with a value of `true` on child **BacnetDevices** (from this point on).

When engineering a **BacnetNetwork** with mostly MS/TP devices in, a possible tactic is to set the network-level **uploadOnStart** property to `false` before adding **BacnetDevices**. Then you could go to the slot sheet of **BacnetDevices** for Bacnet/IP and/or Bacnet/Ethernet devices and either delete (or clear to change from the property sheet) the hidden skipUploadslot.

Rebooting a host

In some cases, job circumstances require that a remote BACnet client has the ability to reboot the host that is running the station. To do this, you must change the **Reinitialize Allowed** property of the **Bacnet Comm→Server** component from `false` to `true`. This property is hidden, by default.

Prerequisites: You have admin privileges.

- Step 1 Clear the Hidden config flag on the **Reinitialize Allowed** property on the Server's **Slot Sheet**.
- Step 2 Change the **Reinitialize Allowed** property from `false` to `true`.
- Step 3 Restart the station.

In most cases, this results in a host reboot.

- Step 4 Change **Reinitialize Allowed** back to **false**.
- Step 5 Enable the **Hidden** config flag on the **Reinitialize Allowed** property on the Server's **Slot Sheet**.

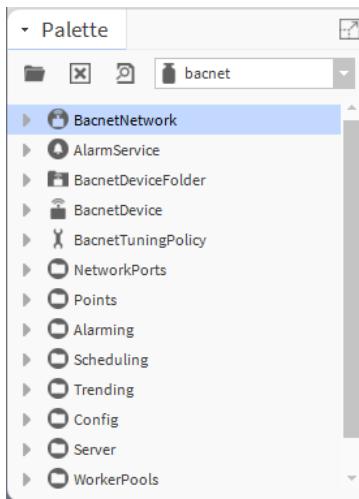
Chapter 10 Components in the bacnet module

Topics covered in this chapter

- ◆ BACnet EDE service (bacnetEDE-WbEdeService)
- ◆ BacnetNetwork (bacnet-BacnetNetwork)
- ◆ Tuning Policies (bacnet-BacnetTuningPolicyMap)
- ◆ BacnetDeviceFolder
- ◆ BacnetDevice (bacnet-BacnetDevice)
- ◆ Points
- ◆ Schedule
- ◆ Trending
- ◆ config
- ◆ Server
- ◆ Worker Pool

This module supports all BACnet features that run in a remote controller.

Figure 22 The bacnet palette

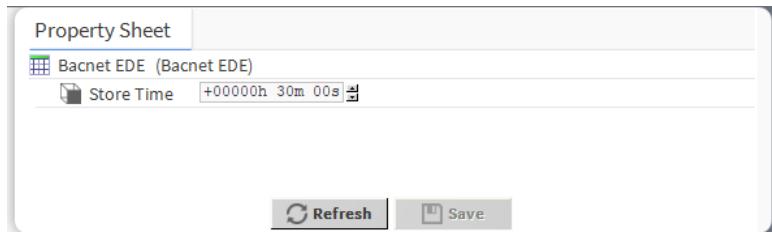


The **BacnetDeviceFolder** is a single component with no child components.

The BacnetDevice folder contains seven child components, which are to be dragged to the station.

BACnet EDE service (bacnetEDE-WbEdeService)

EDE (Engineering Data Exchange) is a CSV- or XLSX-formatted file for exchanging object data. The BIG-EU (BACnet Interest Group Europe) specification defines the formats. This component supports EDE files.

Figure 23 BACnet EDE Property Sheet

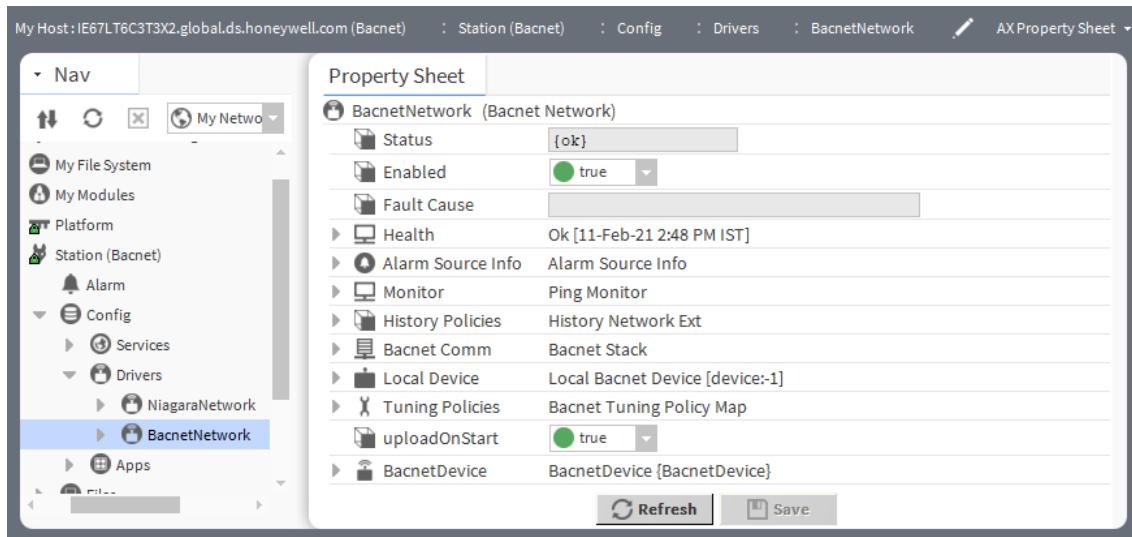
You access this property by expanding the **My Host→My Tools** node in the Nav tree and double-clicking on **Bacnet EDE**. If the Bacnet EDE service is not available under **My Tools**, install it by clicking **Tools→Bacnet EDE**.

| Property | Value | Description |
|------------|------------------------------------------------|-------------|
| Store Time | hours minutes seconds (defaults to 30 minutes) | |

BacnetNetwork (bacnet-BacnetNetwork)

This component is the base container for all BACnet components (devices and proxy points). If added to the station, it resides under the station's **Drivers** container. Its default view is the **Bacnet Device Manager**. In addition to devices and proxy points, this component contains the station's BACnet communications protocol stack (Bacnet Comm), and a Local Bacnet Device, which configures the station's representation as a BACnet device.

The driver supports only one BACnet network component in a station regardless of how many different BACnet link-layer protocols are being used.

Figure 24 BacnetNetwork properties

To access these properties, expand **Config→Drivers**, right-click **BacnetNetwork** and click **Views→AX Property Sheet**.

In addition to the common properties (Status, Enabled, Health and Fault Cause), this component has these properties.

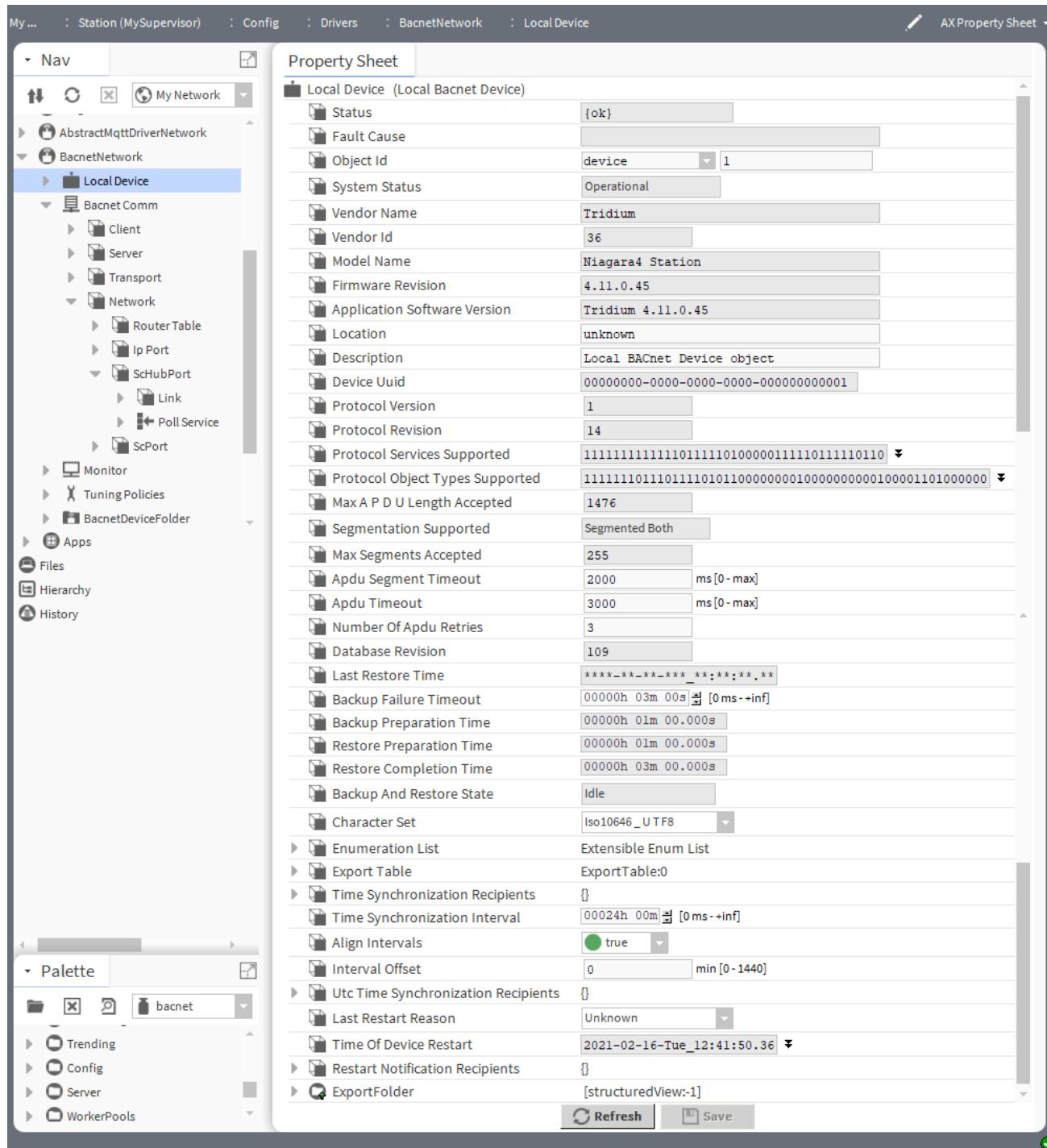
| Property | Value | Description |
|-------------------|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Alarm Source Info | additional properties | Contains a set of properties for configuring and routing alarms when this component is the alarm source. For property descriptions, refer to the <i>Niagara Alarms Guide</i> |
| Monitor | ping monitor | Configures a network's ping mechanism, which verifies network health. This includes verifying the health of all connected objects (typically, devices) by pinging each device at a repeated interval. The <i>Niagara Drivers Guide</i> documents these properties. |
| History Policies | additional properties | Configures the history network extension, which is documented in a separate topic. |
| Bacnet Comm | additional properties | Contains BACnet communication properties, which are documented in a separate topic. |
| Local Device | additional properties | Contains device properties, which are documented in a separate topic. |
| Tuning Policies | additional properties | Configures network rules for evaluating both write requests to writable proxy points as well as the acceptable freshness of read requests. For more information, refer to the <i>Niagara Drivers Guide</i> . |

Local Device (bacnet-LocalBacnetDevice)

This frozen container under the BacnetNetwork represents a BACnet station on the BACnet internetwork. Its child BacnetExportTable container provides server-side functions to export station objects as BACnet objects and service BACnet client requests. In general, this means that it provides BACnet server responses to remote client requests from other BACnet devices on a connected network.

With few exceptions, every component outside the Bacnet Local Device represents a BACnet client operation. Client operations involve BacnetDevice objects and child proxy points.

Figure 25 Local Device properties



You access local device properties by expanding **Config**→**Drivers**→**BacnetNetwork** in the Nav tree and double-clicking the **Local Device** node.

In addition to the common properties (Status and Fault Cause), this component has these properties.

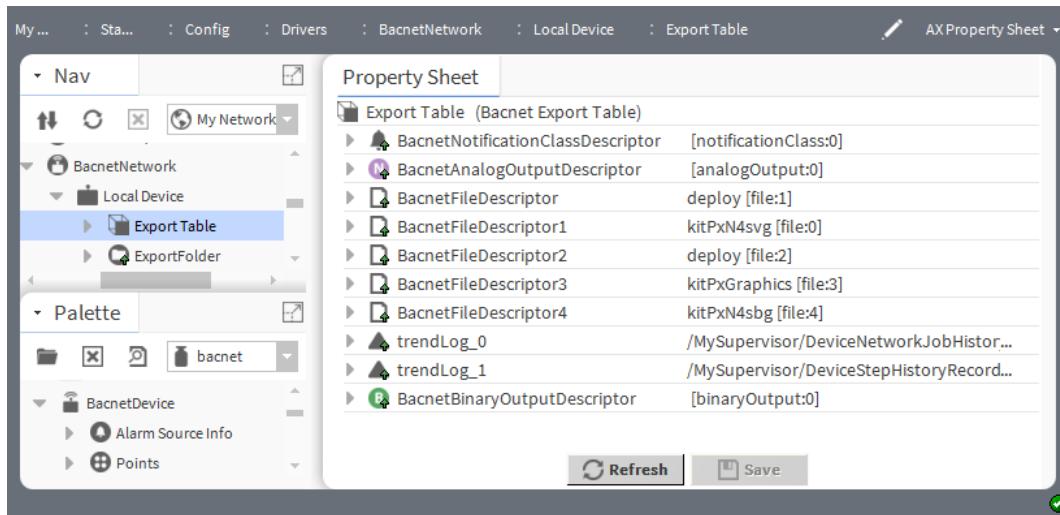
| Property | Value | Description |
|---------------------------------------------|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| System Status | read-only | Reports the current condition of the network. |
| Vendor Name | read-only | Reports the vendor number. |
| Vendor Id | read-only | Reports an identifier for the vendor's device. |
| Model Name | read-only | Reports the manufacturer's model name for the device. |
| Firmware Revision | read-only | Reports the version of the device firmware. |
| Application Software Version | read-only | Reports the version of the manufacturer's software running on the device. |
| Location | text (defaults to unknown) | Describes the device location. |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. |
| Device Uuid (Universally Unique Identifier) | read-only | Uniquely identifies the object in the system. When you upgrade a BACnet driver or drop in a BACnet network from the palette, the driver automatically sets this 128-bit universally unique identifier (UUID) to a random value. You should not need to change this property. The driver uses it behind the scenes. |
| Protocol Version | read-only | Displays the version number of the BACnet protocol. |
| Protocol Revision | read-only | Displays the current BACnet protocol level. |
| Protocol Services Supported | check box list | Expands to identify all the available BACnet device services supported by this version of Niagara. |
| Protocol Object Types Supported | check box list | Expands to identify all the available BACnet object types supported by this version of Niagara. |
| Max A P D U Length Accepted | read-only | Reports the number of points per poll or APDU (Application Protocol Data Units). |
| Segmentation Supported | read-only | Indicates the type of message segmentation the device supports. If a message exceeds the Max Segment Accepted length, the device must be able to break it into segments or part of the message is lost. |
| Max Segments Accepted | read-only | For a single segment, defines the number of characters the device supports. |
| Apdu Segment Timeout | milliseconds (defaults to 2000 ms) | Defines the time the system waits before retransmitting an APDU segment. This property is only relevant if devices are doing segmentation. |

| Property | Value | Description |
|---------------------------------|------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | Works for both client and server operations. For a small IP-based network, an optimal setting may be: 5000 ms. |
| Apdu Timeout | milliseconds (defaults to 3000 ms) | Defines how long the system waits before retransmitting an APDU that requires acknowledgment, for which no acknowledgement has been received. Works for both client and server operations. Large, routed networks may require higher timeouts. |
| Number of Apdu Retries | number (defaults to 3) | Defines the maximum number APDU retransmissions. Works for both client and server operations. For a small IP-based network, an optimal setting may be: 1 retry. |
| Database Revision | read-only | Reports the version of the database. |
| Last Restore Time | read-only | Reports when a backup .dist file was last restored by a remote BACnet Workstation client. If never, all timestamp properties show asterisks (*). |
| Backup Failure Timeout | time range (defaults to 3 minutes) | Specifies the time the server device must wait before deciding that the client has given up, and can leave backup/restore mode. The client writes this to the server prior to beginning the backup/restore procedure. |
| Backup Preparation Time | read-only | Reports the time taken, after receiving a restore request, to gracefully shut down the station and close files before beginning the backup. |
| Restore Preparation Time | read-only | Reports the time taken, after receiving a restore request, to gracefully shut down the station and close files before beginning the dist file installation. |
| Restore Completion Time | timestamp (defaults to 3 minutes) | Reports the time taken, after completing a dist file install from a restore, for the station to restart and become responsive. |
| Backup and Restore State | read-only | Shows the current status of any backup or restore operation. The driver displays <code>Idle</code> when no operation is active. This changes to <code>Preparing</code> for backup and so forth. |
| Character Set | drop-down list (defaults to <code>Ansi C3_4</code>) | Defines the character set supported, with other selections: IBM/Microsoft DBCS, JIS C 6226, ISO 10646 (UCS-4), ISO 10646 (UCS-2), ISO 8859-1, and Unknown. |
| Enumeration List | sets of enum facets | Lists the BACnet properties that can be exported to configure data exchange. This Extensible Enum List provides access to each property's configuration facets. |
| Export Table | Bacnet Export Manager | Provides the special manager views to associate devices and points (local objects) with BACnet objects (exported objects). You discover objects or add them using the New button. The <i>Bacnet Export Manager</i> (plugin) topic documents this table. |
| Time Synchronization Recipients | Bacnet List Of | Define the devices to receive time synchronization messages from the driver. This list of recipients is for periodic time synchronization messages. You add to and remove entries from this list using the right-click addElement and removeElement |

| Property | Value | Description |
|-------------------------------------|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | actions. Entries in this list are BacnetRecipients . They can be either a device object identifier, or a BACnet address. |
| Time Synchronization Interval | timestamp (defaults to 24 hours) | Configures how frequently the software sends the time synchronization messages. |
| Align Intervals | true (defaults) or false | Turns on and off the configuration of standard synchronization intervals for controller stations. true sets up the same time synchronization interval in each host controller station. false allows a different time synchronization interval for each host controller station. |
| Interval Offset | minutes | Specifies an offset from the beginning of the interval at which to send the periodic time synchronization messages. If Align Interval is false, the driver ignores this property. |
| Utc Time Synchronization Recipients | Bacnet List Of | Lists the recipients for periodic Time Synchronization messages. You can add and remove entries to this list using the right-click addElement and removeElement actions. Entries in this list are BacnetRecipients. They can be either a device object identifier, or a BACnet address. |
| Last Restart Reason | drop-down list | Reports the reason for the last restart. |
| Time of Device Restart | date and time | Specifies the time of device restart based on the time of day (Time), a sequence number (SeqNum) or a date and time (DateTime). |
| Restart Notification Recipients | Bacnet List Of | Defines the alarm recipients to receive notification that the station restarted. |
| ExportFolder | Bacnet Array | Sets up a folder array. |

Export Table (bacnet-BacnetExportTable)

This frozen slot under **BacnetNetwork→Local Device** contains all the server (export) descriptors BACnet export folders used to export station objects as BACnet objects. You use different export manager views of this table to export station objects. It has no configuration properties but three important views.

Figure 26 Example of Export Table properties

You access this Property Sheet by expanding **Config**→**Drivers**→**BacnetNetwork**→**Local Device**, right-clicking **Export Table** and clicking **Views**→**AX Property Sheet**.

Server descriptors

BACnet server descriptors are the components responsible for exporting station objects as BACnet objects. Each descriptor resides under the **Export Table** in the Nav tree, and corresponds to a particular component, file, or history in the station.

You add, edit, and manage server descriptors using the manager views of the **Export Table**. Each manager view simplifies selection and enforces object ID instance rules. The driver automatically gives server descriptors names (upon creation) using an `<ObjectType>_<InstanceNumber>` convention. For example: `analogValue_1`, `trendLog_0`, `schedule_2`, and so on.

NOTE: Generally, it is recommended that you do not change the driver-assigned name of the export descriptor. If you wish to change the name by which the object is known to BACnet clients, do this in the `objectName` property, which is initialized with the name of the exposed component.

There are three categories of BACnet server descriptors:

- Export descriptors (for components, reflecting the station object space)
- File descriptors (for files)
- Log descriptors (for histories)

In addition, as needed, you can add BACnet export folders to organize BACnet server descriptors.

Export descriptors

BACnet export descriptors include 15 different types to export components, including Boolean, Enum, and Numeric points (read-only and writable), and various Schedule types. You add them using the **Bacnet Export Manager** view.

Externally, to another BACnet device, the exported BACnet object opens with properties that source from two different areas of the station. For example, each exported component (point or schedule) exports as a BACnet object with properties from the component itself, plus additional properties in its export descriptor.

NOTE: You do not see export (component) descriptors in the **Bacnet File Export Manager** and **Bacnet Niagara Log Export Manager** views. These views show other non-component server descriptors.

File descriptors

A BACnet file descriptor exports a file under the station directory as a BACnet file object. You add file descriptors using the **Bacnet File Export Manager** view of the **Export Table**.

NOTE: You do not see file descriptors in the **Bacnet Export Manager** and **Bacnet Niagara Log Export Manager** views. These views show other server descriptors.

Log descriptors

A BACnet log descriptor exports a history as a BACnet trend log object. There are two log descriptors:

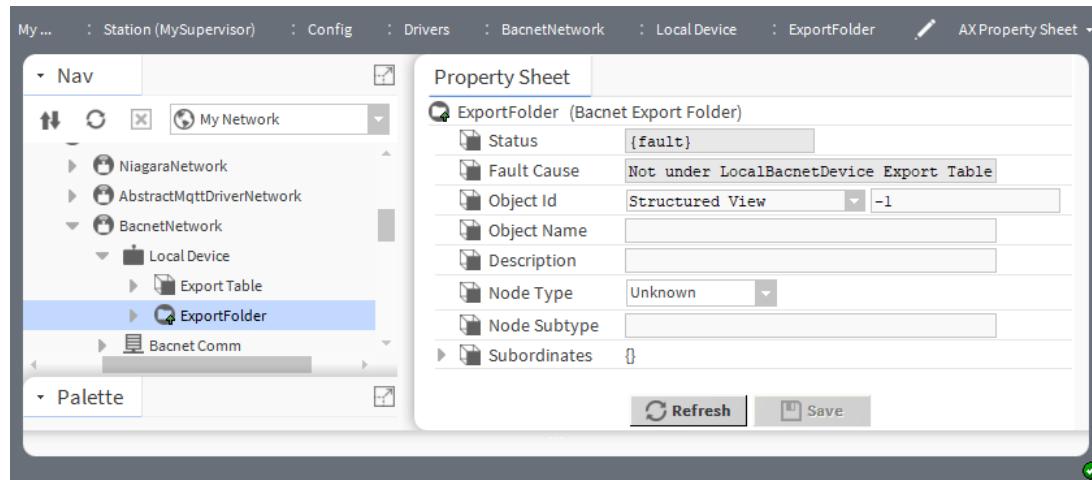
- The system creates a **BacnetNiagaraHistoryDescriptor** when you select and add a standard history using the **Bacnet Niagara Log Export Manager** view of the **Export Table**. The resulting exported trend log object is compliant only with the original broken specification for BACnet trend logs, which was superseded and fixed in Addendum B to the 2001 version of the spec.
- The system creates a **BacnetTrendLogDescriptor** when you add a specialized **BacnetTrendLogExt** extension to a point. The resulting exported trend log object is fully-BACnet compliant with the 2004 version of the BACnet spec.

ExportFolder (bacnet-BacnetExportFolder)

This component organizes export descriptors. It resides in two places in a station: under **Local Device** where it organizes the export descriptors found in the local station database, and elsewhere under the **Bacnet Comm→Network** component where it organizes server export descriptors.

NOTE: After adding an export folder, you can rename it (click to select it, then **Ctrl + R**). For clarity, rename it to match its **Object Name**.

Figure 27 ExportFolder properties



To add an export folder to a station, open the **bacnet** palette, expand the **Server** folder and copy the **ExportFolder** component to a location in the station.

To access these properties, right-click the **ExportFolder** and click **Views→AX Property Sheet**.

In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|-------------|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name. | Identifies a BACnet object. This name should be unique within this specific device. |

| Property | Value | Description |
|--------------|--------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | name.name (default includes the entire component path in the station using period ("/) delimiters between parent.child levels) | <p>For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device.</p> <p>For a proxy point, this is the default name in the station.</p> <p>You can shorten or edit Object Name, however, each Object Name must be unique among all server descriptors under the Export Table.</p> |
| Description | text | <p>Adds information to more fully explain an object's purpose, function or location.</p> <p>NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the Add and Edit window.</p> |
| Node Type | drop-down list, defaults to unknown | <p>Classifies a BACnet object (node) as:</p> <p>Unknown System Network Device Organizational Area Equipment Point Collection Property Functional Other</p> |
| Node Subtype | text, defaults to blank | Works with Node Type to more specifically classify a text descriptor. |
| Subordinates | additional properties | Refers to the structured view objects, which are available to other structured view objects. |

List Of Group Members (bacnet-BacnetListOf)

This component represents a BACnet list containing a non-indexed sequence of objects of a particular BACnet data type.

Multiple lists are supported including:

- Time Synchronization Recipients
- Utc Time Synchronization Recipients
- Restart Notification Recipients
- Device Address Binding

Bacnet Comm (bacnet-BacnetStack)

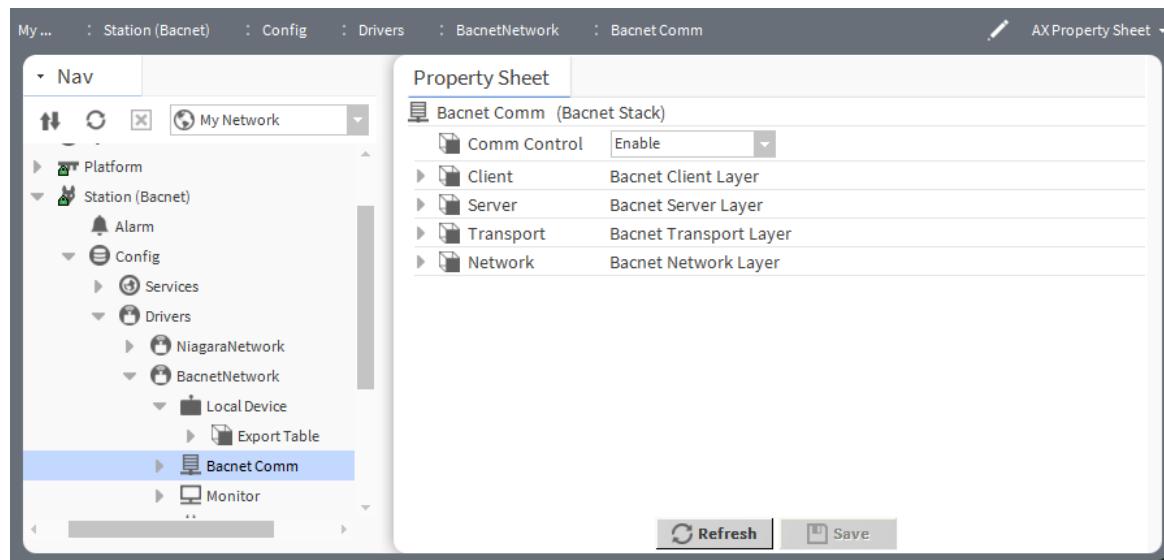
This frozen child container slot/component provides the protocol stack for BACnet communications. In turn, it has child frozen container slots **Client** (BacnetClientLayer), **Server** (BacnetServerLayer), **Transport** (BacnetTransportLayer), and **Network** (BacnetNetworkLayer).

The **Bacnet Comm** properties configure the protocol stack for all BACnet communications. **Bacnet Comm** has a **Comm Control** property that you must enable for any BACnet communications.

Relative to other **Bacnet Comm** child containers, the properties under the **Client**, **Server**, and **Transport** containers require little configuration (from default values).

Possible exceptions apply to the**Server** component. These exceptions are not typical, and can be skipped whenever the station's BACnet driver operates only in client mode, which means that it is not licensed and configured to operate as a BACnet server.

Figure 28 Bacnet Comm properties



You access these properties by clicking **Tools**→**Bacnet Service**, and expanding **Bacnet Comm**; or by expanding the **BackupNetwork** in the Nav tree and double-clicking the **Bacnet Comm** container.

| Property | Value | Description |
|--------------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Comm Control | drop-down list | Configures communication control: Enable turns communication control on. Disable turns communication control off. Disable Initiation |
| Client | read-only | Represents the client side of the application layer of the Bacnet communications stack. It has no available properties. |
| Server | additional properties | Contains server-related properties. Refer to a separate topic in this document. |
| Transport | additional properties | Contains transport-related properties. Refer to the properties by name in this dictionary. |
| Network | additional properties | Contains network-related properties. Refer to separate topics in this document. |

Client (bacnet-BacnetClientLayer)

This component represents the client side of the application layer of the BACnet communications stack.

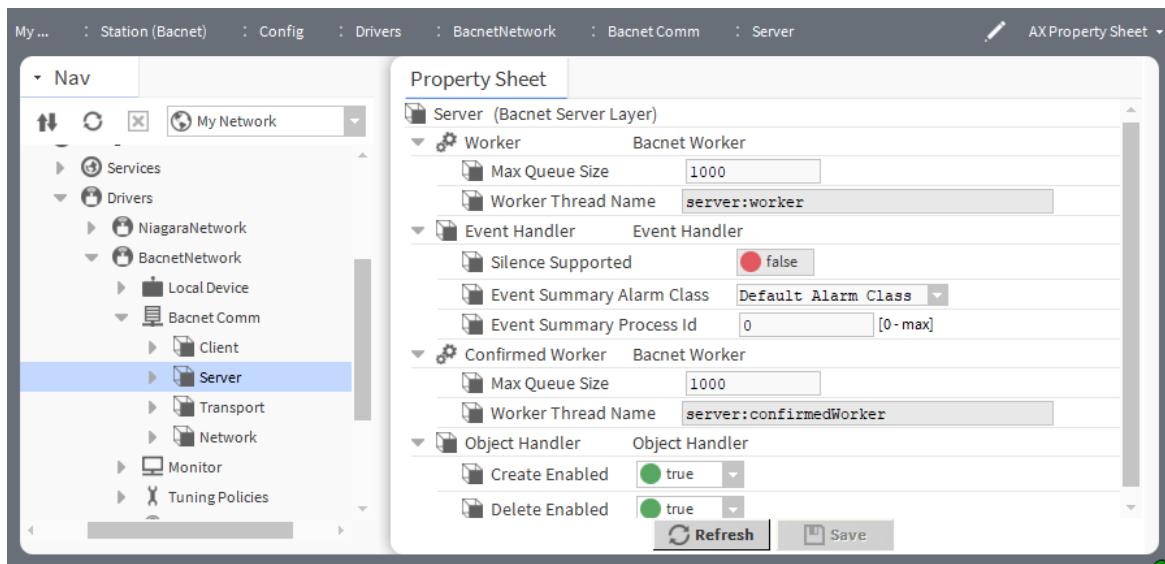
To access this node in the Nav tree, expand the BACnet network and **Bacnet Comm**. This layer of the driver has no properties.

Server (bacnet-BacnetServerLayer)

This component represents the server side of the application layer of the BACnet communications stack.

These properties configure how the driver manages messages, such as event (alarm) notifications. Worker components queue and process threads (reads and writes to BACnet) in a running station. At least four frozen Worker slots under a BacnetNetwork manage thread processing. Often, the default configuration of Workers provides good results. However, in some scenarios, configuration changes may improve performance.

Figure 29 Server properties



You access these property containers under **BacnetNetwork→Bacnet Comm: Worker, Event Handler, Confirmed Worker, and Object Handler**.

| Property | Value | Description |
|------------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Worker | additional properties | Contains worker-related server properties. Refer to the properties by name in this dictionary. |
| Event Handler | additional properties | Reports event notification messages and acknowledgments. Messages include: ConfirmedEventNotification, UnconfirmedEventNotification and AcknowledgmentAlarm. |
| Confirmed Worker | additional properties | Contains server-related properties. Refer to separate topics in this document. |
| Object Handler | additional properties | Contains server-related properties. Refer to separate topics in this document. |

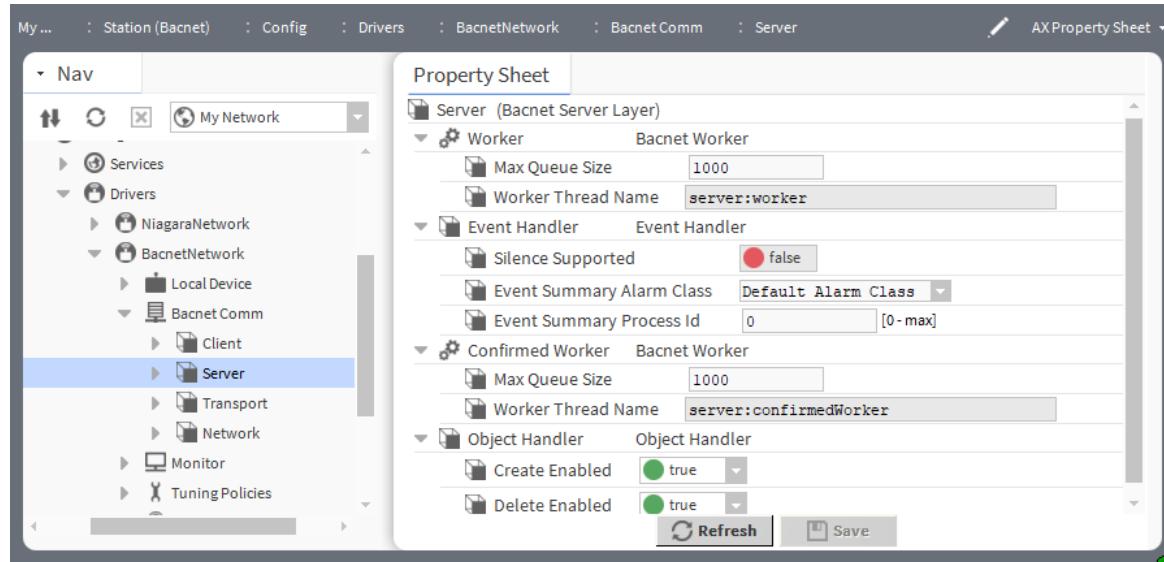
Server, Worker and Confirmed Worker (bacnet-BacnetWorker)

These components manage the queue and thread processing for reads and writes to BACnet devices. A BacnetNetwork has at least three frozen Worker slots. The driver provides at least one visible slot under the **Bacnet Comm→Server**, component. Other slots may be hidden.

Sometimes, default configuration of Workers provides good BACnet driver performance. However, in some scenarios configuration adjustments may be needed.

These properties configure the queuing and processing of server messages (responses to requests from BACnet clients). Workers manage the queue and thread processing for reads and writes to BACnet. Sometimes the default configuration of worker components provides good BACnet driver performance. However, if a station operates as a BACnet server with a large number of BACnet clients, configuration changes to one or both workers may be beneficial.

Figure 30 Worker properties



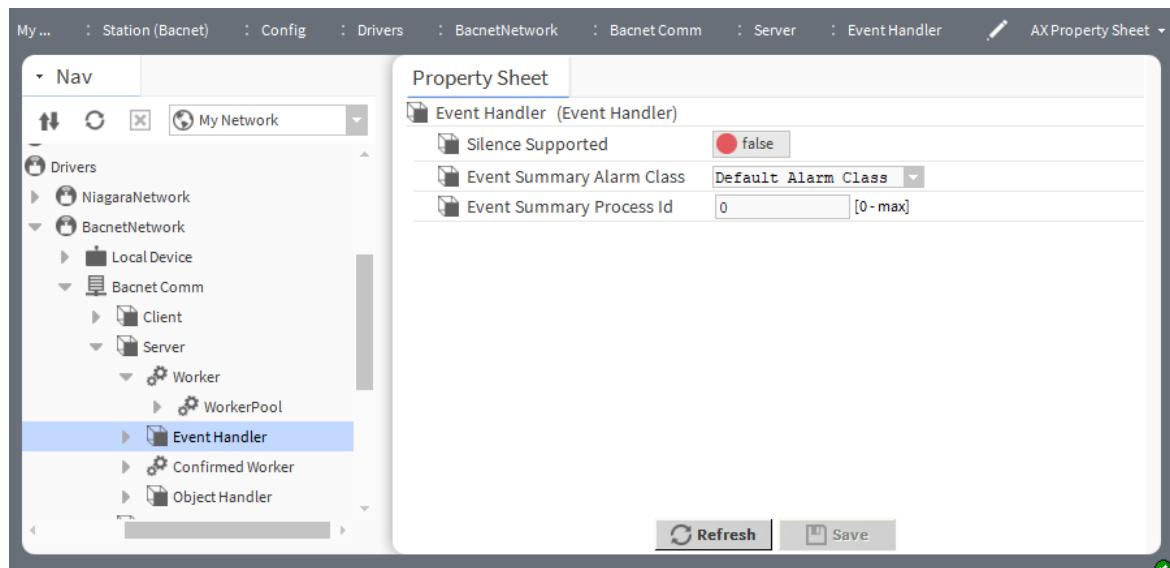
To access these properties, expand **Config→Drivers→BacnetNetwork→Bacnet Comm→Server** and double-click **Worker**.

To access a hidden worker slot, you must first clear the hidden flag from the network's slot sheet. Separate workers help to prevent certain confirmed event notifications from blocking the processing of all incoming requests to the BACnet server layer.

| Property | Value | Description |
|--------------------|-------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Max Queue Size | number (defaults to 1000) | <p>Specifies the maximum number of items that can be queued for worker processing. In a few cases, particularly in a large system, increasing this size may be beneficial.</p> <p>Queue size does not allocate a fixed memory size. Rather, the amount of memory used is dynamic, changing as message items are added and removed from the queue.</p> |
| Worker Thread Name | read-only in the format: server:worker or server:confirmed-worker | <p>Reports the identifiers for threads managed by the worker.</p> <p>Processes unconfirmed server messaging, including event (alarm) notifications, confirmed COV, and so on. For example Who-Is and Who-Has responses as well as unconfirmed COV server messaging.</p> |

Server, Event Handler (bacnet-EventHandler)

This component is a frozen container slot of the **BacnetNetwork→Bacnet Comm→Server**. It processes event notification messages and the acknowledgment of those messages, including **ConfirmedEventNotification**, **UnconfirmedEventNotification**, and **AcknowledgmentAlarm**.

Figure 31 Server, Event Handler properties

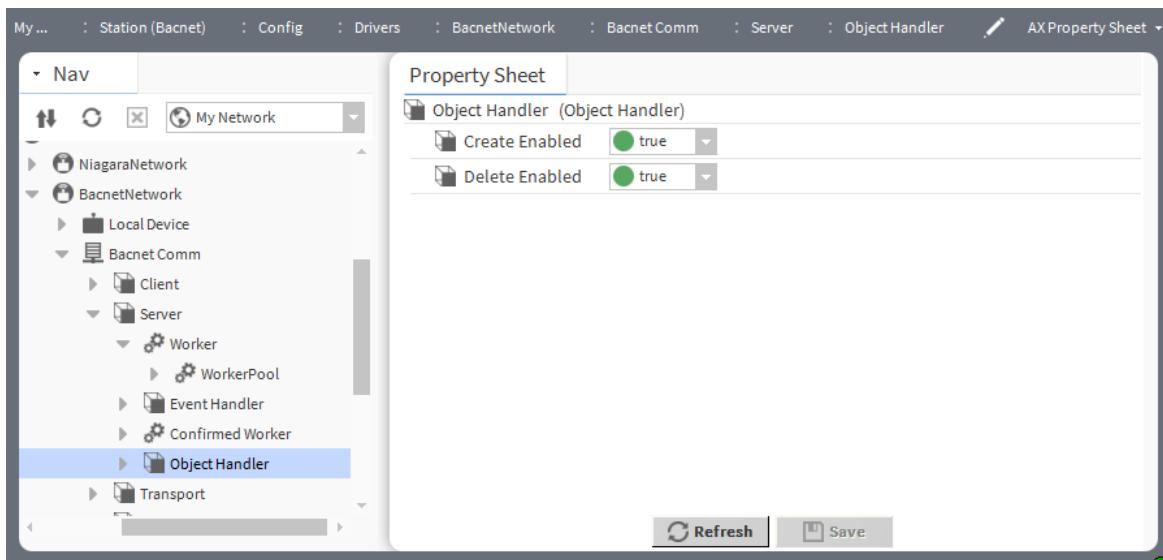
To access these properties, expand **Config**→**Drivers**→**BacnetNetwork**→**Bacnet Comm**→**Server** and double-click **Event Handler**.

| Property | Value | Description |
|---------------------------|--------------------------------------------|-----------------------------------------------------------------|
| Silence Supported | true or false (default) | |
| Event Summary Alarm Class | drop-down list | Selects the alarm class for a server event. |
| Event Summary Process Id | number from 0 to a maximum (defaults to 0) | Specifies the number of a text string that describes the alarm. |

Server, Object Handler (bacnet-ObjectHandler)

This component is a frozen container slot of the **BacnetNetwork**→**Bacnet Comm**→**Server**.

You can create and delete objects dynamically by setting the **Create Enabled** and **Delete Enabled** properties to **true**. You can create and delete objects like Event Enrollment, Trend Log, Schedule, Notification Class, and Calender.

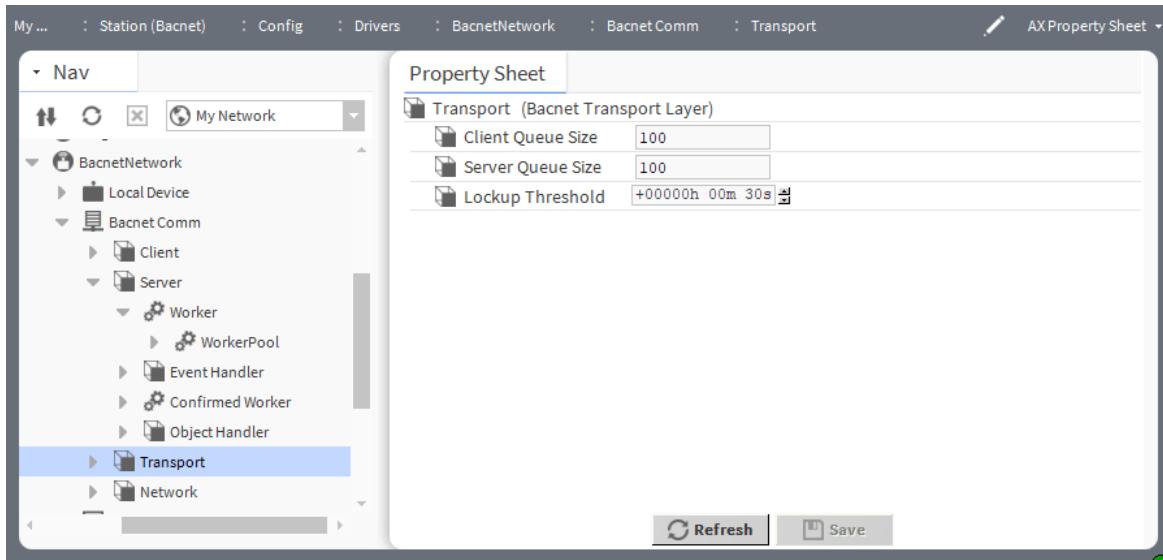
Figure 32 Object Handler properties

To access these properties, expand **Config→Drivers→BacnetNetwork→Bacnet Comm→Server** and double-click **Object Handler**.

| Property | Value | Description |
|----------------|----------------------------|------------------------------------------|
| Create Enabled | true or false (default) | Enables the dynamic creation of objects. |
| Delete Enabled | true or false (default) | Enables the dynamic deletion of objects. |

Transport (bacnet-BacnetTransportLayer)

This component implements the BACnet transport layer. It resides under the **BacnetNetwork** in the **Bacnet Comm** container, and contains properties that typically do not require adjustment.

Figure 33 Transport properties

To access, click **Config→Drivers** followed by the BACnet network, then expand **Bacnet Comm** and double-click **Transport**.

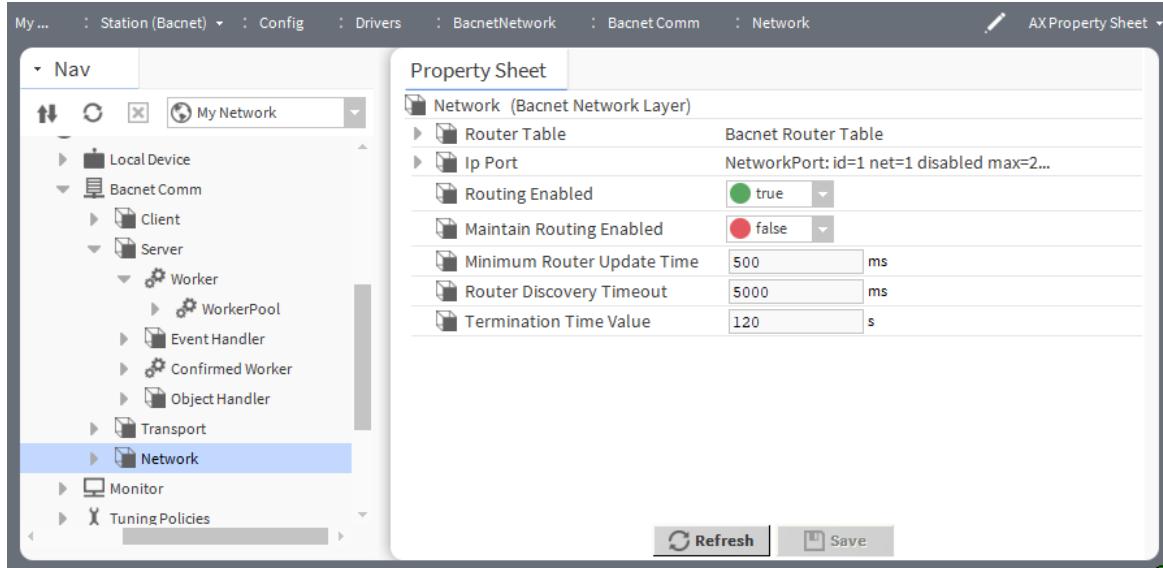
| Property | Value | Description |
|-------------------|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Client Queue Size | number (defaults to 100) | Defines the number of records supported in the client-side queue. |
| Server Queue Size | number (defaults to 100) | Defines the number of records supported in the server-side queue. |
| Lockup Threshold | hours, minutes and seconds | Defines the time, which system waits for either a transaction time-out or a response from a remote device. After this time, system will detect a lockup of the transport state machine and restart it, abandoning the transaction. |

Network (*bacnet-BacnetNetworkLayer*)

This component represents the generic superclass for all network layer implementations. In practical terms, it is the container for all network ports, plus the BACnet router table. **Network** resides under **Bacnet Comm** of a **BacnetNetwork** or **BacnetWsNetwork**.

The screen capture shows an **Ip Port**. The same properties define an **Ethernet Port** and an **Mstp Port** (available only when using the **bacnet** module).

Figure 34 Bacnet Comm Network properties



You access these properties by expanding **Config→Drivers→BacnetNetwork→Bacnet Comm** and double-clicking the **Network** node.

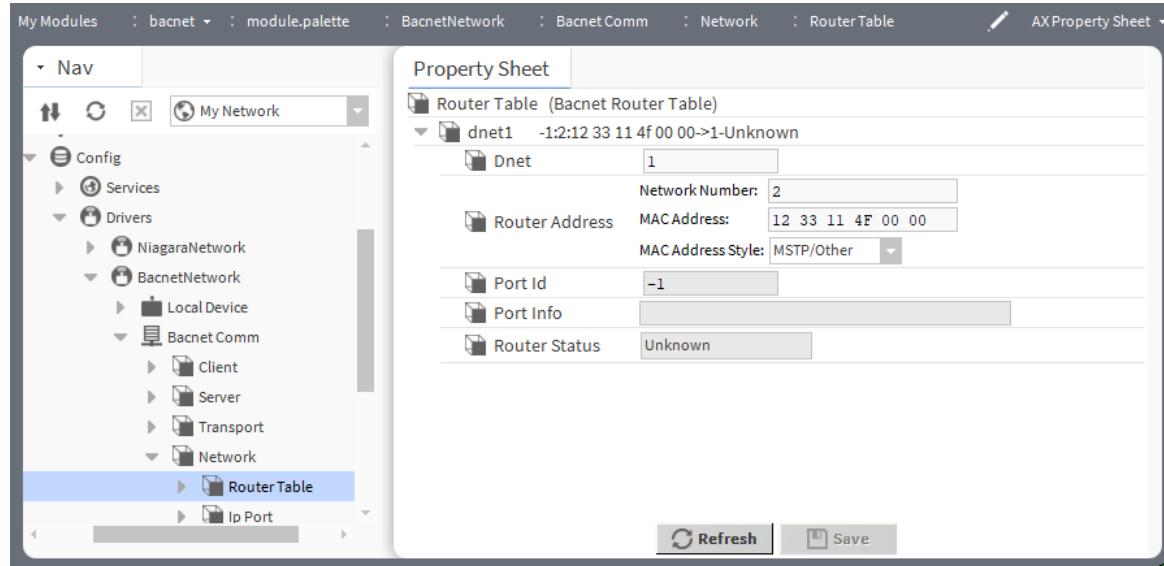
| Property | Value | Description |
|-----------------|-------------------------|--------------------------------------------------------------------------------|
| Router Table | additional properties | Displays router properties when a network includes more than one network port. |
| IP Port | additional properties | Configures IP port properties. |
| Routing Enabled | true (default) or false | Turns network routing on and off. |

| Property | Value | Description |
|----------------------------|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | true enables multiple network ports. false disables multiple network ports. If the driver determines that the network is configured with multiple routes to the same network segment, it sets this property to false. Configuring multiple routes to the same network segment results in message flood issues and is considered a mis-configuration. |
| Maintain Routing Enabled | true or false (default) | Controls the routing of messages. true overrides a false setting of Routing Enabled . false allows the driver to disable routing when it detects a router loop. If Routing Enabled is set to false because of a router loop or because multiple routes lead to the same segment, this property allows you to override the automatic setting. |
| Minimum Router Update Time | milliseconds (defaults to 500 ms) | Specifies the minimum amount of time allowed to update the router. |
| Router Discovery Timeout | milliseconds (defaults to 5000 ms) | Sets how long the system waits to discover the router. When this time passes, the system reports an error. |
| Termination Time Value | seconds (defaults to 120 s) | Defines a time, which system database will request with the PTP connection open, in the absence of any packet traffic. |

Network, Router Table (bacnet-BacnetRouterTable)

This component stores the table known routers to BACnet networks. It appears under **BacnetNetwork**→**Bacnet Comm**→**Network** and lists router entries by dnet (destination network).

Figure 35 Router Table properties



To access, click **Config**→**Drivers** followed by the BACnet network, then expand **Bacnet Comm**→**Network** and double-click **Router Table**.

After device discovery using the **Bacnet Device Manager**, where the **All Networks** option was chosen, additional router table entries may exist. These reflect other discovered remote BACnet routers. Such entries may populate following an **All Networks** discovery even if **Routing Enabled** is **false**. Entries like this may also result from unsolicited messages received from remote networks.

After 24 hours of no traffic directed to a network, the driver clears the router table entries to prevent the table from being cluttered with old entries. As needed, it discovers router table entries again.

If the station detects a BACnet router loop, the driver may automatically set the **Network's Routing Enabled** property to **false**. When this happens, the station stops acting as a BACnet router for its enabled network ports and generates corresponding entries in the stations's LogHistory, which note detected mis-configuration and disabled BACnet router functionality.

If needed, you can override this behavior by setting the **Maintain Routing Enabled** property to **true**, and resetting **Routing Enabled** back to **true**.

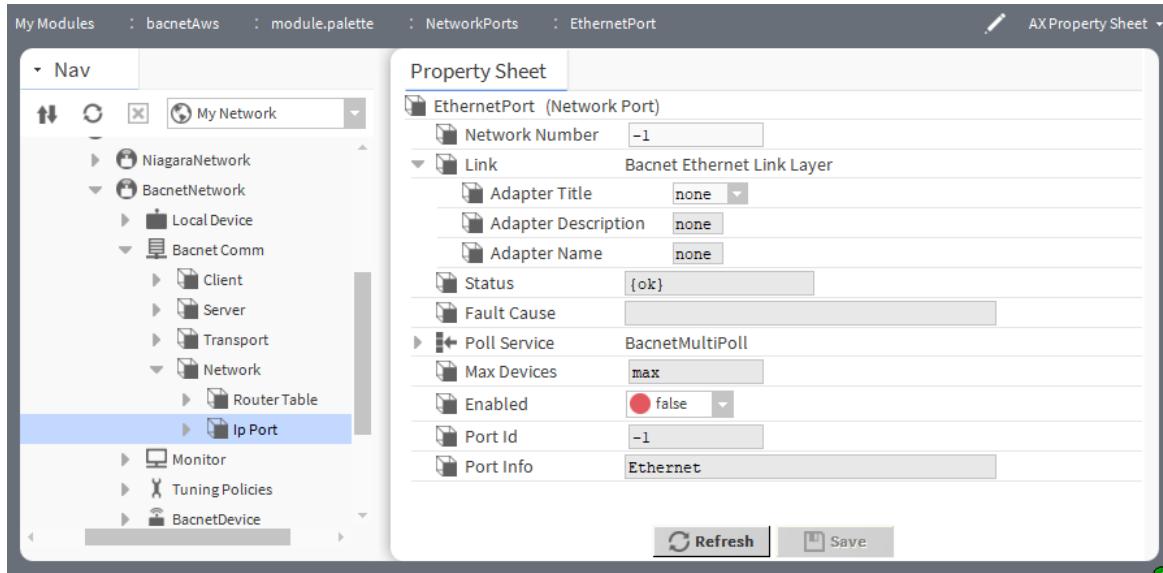
| Property | Value | Description |
|----------------|---------------------|-----------------------------------------------------------------------------------------------------------------------|
| Dnet | number | Identifies the BACnet device network (Dnet). |
| Router Address | multiple properties | Defines the address of the network router including Network Number , MAC Address and MAC address style. |
| Port Id | read-only | Reports the number of the Ethernet port you are configuring. |
| Port Info | read-only | Reports the type of port (Ethernet, MS/TP, etc.). |
| Router Status | read-only | Indicates the status of the router. |

Network ports (bacnet-NetworkPort)

These components send and receive messages to and from the BACnet link layer. All BACnet networks support these ports: **IpPort** and **EthernetPort**. The **MstpPort** works only in a remote controller and is available only in the **bacnet** module/palette. In addition, network ports include the **ScPort** and **SCHubPort**.

Each port supports the same port-level properties, including **Poll Service**, **Port Id**, **Port Info**. Each provides a unique set of **Link** properties.

Figure 36 Example: Ethernet Port properties, Link container expanded



You access these properties by expanding **Config→Drivers→BacnetNetwork→Bacnet Comm→Network** followed by double-clicking the port node.

If needed, add a port from the **bacnet**, **bacnetAws** or **bacnetOws** palettes. Each port container communicates based on the type of port (Ip, Ethernet, MS/TP, etc). The **bacnet** module supports up to 4 MS/TP ports, depending on the host platform, as well as licensing considerations.

In addition to the common properties (Status, Fault Cause and Enabled), this component has these properties.

| Property | Value | Description |
|----------------|-------------------------------------|-----------------------------------------------------------------------------------|
| Network Number | number from 1-65535 (defaults to 0) | Defines a unique number for the network. |
| Link | additional properties | Configures the Ethernet connection. A separate topic documents these properties. |
| Poll Service | additional properties | Contains network polling properties. A separate topic documents these properties. |
| Max Devices | read-only | Indicates maximum devices available in the network. |
| Port Id | read-only | Reports the number of the Ethernet port you are configuring. |
| Port Info | read-only | Reports the type of port (Ethernet, MS/TP, etc.). |

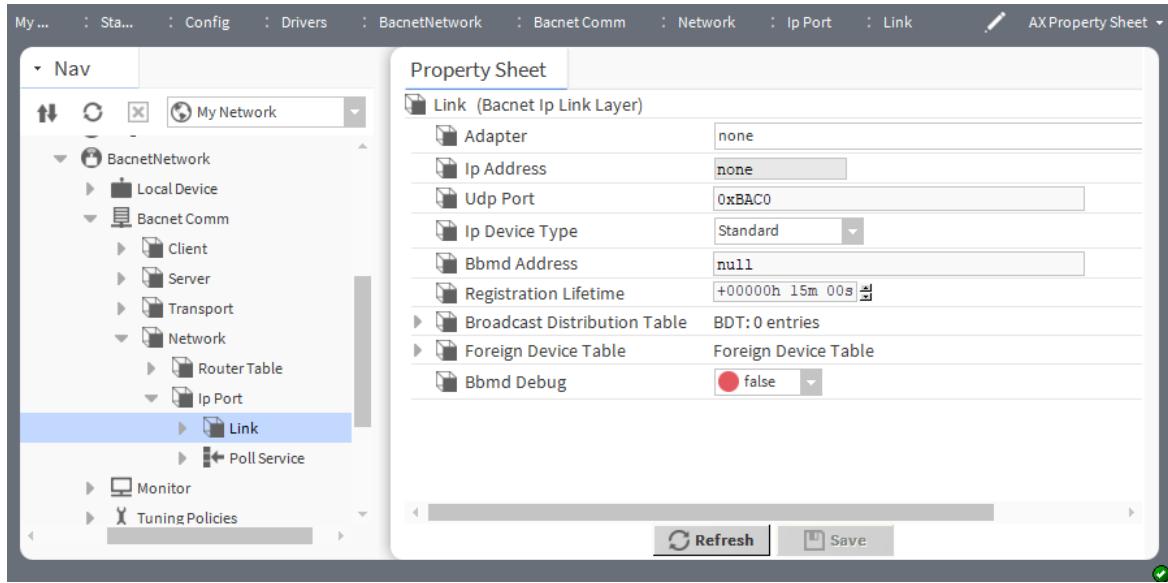
Actions

- **Enable** activates the port.
- **Disable** deactivates the port.

Link (Ip) (bacnet-BacnetIpLinkLayer)

This component is the BACnet/IP virtual link layer implementation. Link binding is automatic to the selected Ip adapter of the host platform.

The **Ip Port** container provides for BACnet Annex J communications as a BACnet/IP device. This **Link** node is also the parent container for the **Broadcast Distribution Table** and **Foreign Device Table**.

Figure 37 Link Ip properties

You access these properties by expanding **Config**→**Drivers**→**BacnetNetwork**→**Bacnet Comm**→**Network**→**Ip Port** and double-clicking **Link**.

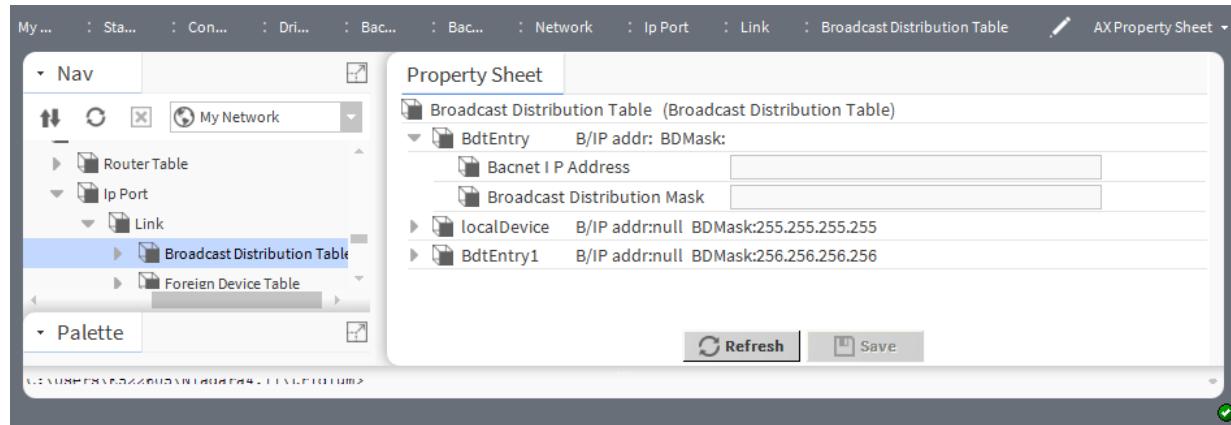
| Property | Value | Description |
|------------------------------|--------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Adapter | text (defaults to none) | Defines the Supervisor PC's network adapter or VPN connection. |
| Ip Address | IP address | Defines the IP address of the source or destination device. |
| Udp Port | text (defaults to 0xBAC0 (decimal 47808)) | Defines the number (in Hexadecimal) of the UDP (User Datagram Protocol) network port. |
| Ip Device Type | drop-down list | Defines the Ip device type (Standard, Foreign Device or Bbmd). |
| Bbmd Address | text (defaults to null) | Identifies a known BBMD (BACnet Broadcast Management Device) on a remote subnet, using its full BACnet/IP MAC address, that is IP_Address:UDP_port. |
| Registration Lifetime | hours, minutes, seconds (defaults to 15 minutes) | Defines a period of time during which the driver may register the station again. |
| Broadcast Distribution Table | additional properties | Provides a list of all other participating BBMDs, including their IP address and broadcast distribution masks when the station operates as a BBMD. |
| Foreign Device Table | additional properties | Provides a list of either BACnet device or foreign device that have registered with framework, including their IP address, time to live, and purge time when the station operates as a BBMD. |
| Bbmd Debug | true or false (default) | Enables (true) and disables (false) the debug feature. |

Link (Ip), Broadcast Distribution Table (bacnet-BroadcastDistributionTable)

This component is a frozen child slot under of the BacnetLinkLayer (**Link**) node under a network port, which itself is a child of **BacnetNetwork→Bacnet Comm**. If the station is operating as a BBMD (BACnet Broadcast Management Device), this BDT table contains the IP addresses and broadcast distribution masks of all other known participating BBMDs.

This table's default **Bdt Manager** view allows you to manually enter known BBMDs, if needed. The system automatically propagates all changes to the BDT to the other BBMDs in the table.

Figure 38 Broadcast Distribution Table properties



You access these properties by expanding **Config→Drivers→BacnetNetwork→Bacnet Comm→Network→Ip Port→Link**, right-click **Broadcast Distribution Table** and click **Views→AX Property Sheet**.

| Property | Value | Description |
|---------------------------------------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BdtEntry, Bacnet IP Address | IP address | Configures the IP address used to broadcast to the BACnet network. |
| BdtEntry, Broadcast Distribution Mask | number | Sets up either a subnet mask or all ones (1s). This mask indicates if a BBMD is to send a directed broadcast (retransmitted by appropriately configured IP routers) or a unicast message to the indicated BBMD, which then retransmits the forwarded broadcast message. |

Link (Ip), Foreign Device Table (bacnet-ForeignDeviceTable)

This component is a frozen child slot of the BACnet Ip Port Link layer node under a **Network** port, which itself is a child of **BacnetNetwork→Bacnet Comm**. If the station is operating as a BACnet foreign device, this FDT table contains information about other BACnet foreign devices registered with the station. Typically, this table is automatically populated, and devices are purged after the requested lifetime expires.

The **Fdt Manager** is the default view. Using it you can manually enter foreign devices if they are incapable of self-registration. Manually entered foreign devices are not purged from the FDT table.

In a case of an IP subnet, where only a few (or perhaps just one) BACnet/IP device exists, a local BBMD may be considered excessive for BACnet broadcast message support. An alternative for that subnet is for each BACnet/IP device to register as a foreign device with a particular BBMD on a remote subnet. Once registered, the driver adds the device to that BBMD's FDT (Foreign Device Table). It then becomes that BBMD's responsibility to deliver global BACnet broadcast messages to the remote device.

Because this scheme is sometimes used for BACnet/IP devices that are not permanent, it was designed with a mandatory registration lifetime feature. When any BACnet/IP device registers as a foreign device with a BBMD, it must specify its **Time To Live** value, in seconds. It is then expected to re-register within this period, otherwise the BBMD removes (purges) it from its FDT. This prevents unnecessary broadcast delivery attempts to part-time participants.

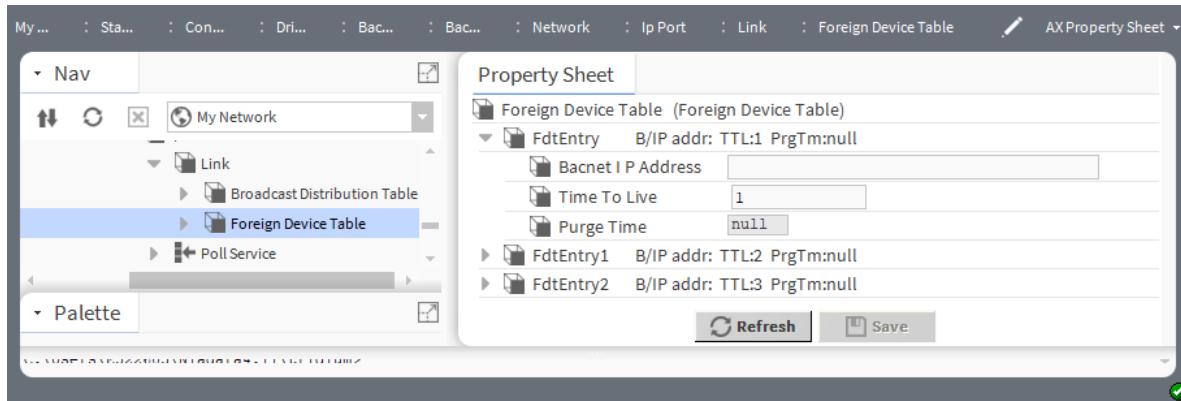
The FDT in any BBMD reflects the current list of its registered foreign devices, along with each device's **Time To Live** value and calculated purge time. If the host station is configured as a BBMD, you can see all entries in its FDT.

NOTE:

The term "foreign device" implies no stigma—it is purely BBMD-centric. The most expedient configuration for a station may well be as a foreign device, providing its local IP subnet has no BBMD and each additional BACnet/IP device (if any) is currently registered as a foreign device with a remote BBMD.

Unlike a BBMD's broadcast distribution table, which is identical in each BBMD, the foreign device table in each BBMD is unique to that BBMD.

Figure 39 Foreign Device Table properties



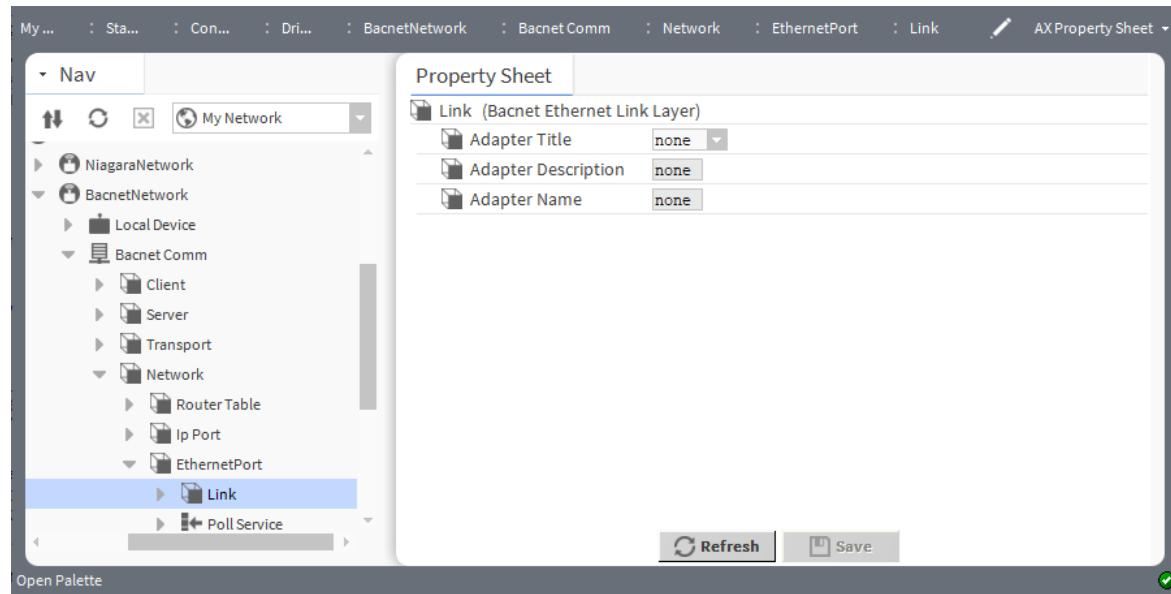
You access these properties by expanding **Config→Drivers→BacnetNetwork→Bacnet Comm→Network→Ip Port→Link**, right-click **Foreign Device Table** and click **Views→AX Property Sheet**.

| Property | Value | Description |
|-----------------------------|------------|-----------------------------------------------------------------------------------------------------------|
| FdtEntry, Bacnet IP Address | IP address | Configures the IP address used by the FDT (Foreign Device Table). |
| FdtEntry, Time to Live | seconds | Defines the period of time within which a foreign device must re-register or be purged from the database. |
| FdtEntry, Purge Time | read-only | Reports when the device is scheduled to be removed from the database. |

Link (bacnet-BacnetEthernetLinkLayer)

This component is a virtual link layer implementation. It is available under the **Bacnet Comm, Network-Ports** folder of the **bacnet** palette.

Link binding is automatic to the selected Ethernet adapter of the host platform.

Figure 40 Link (Ethernet) properties

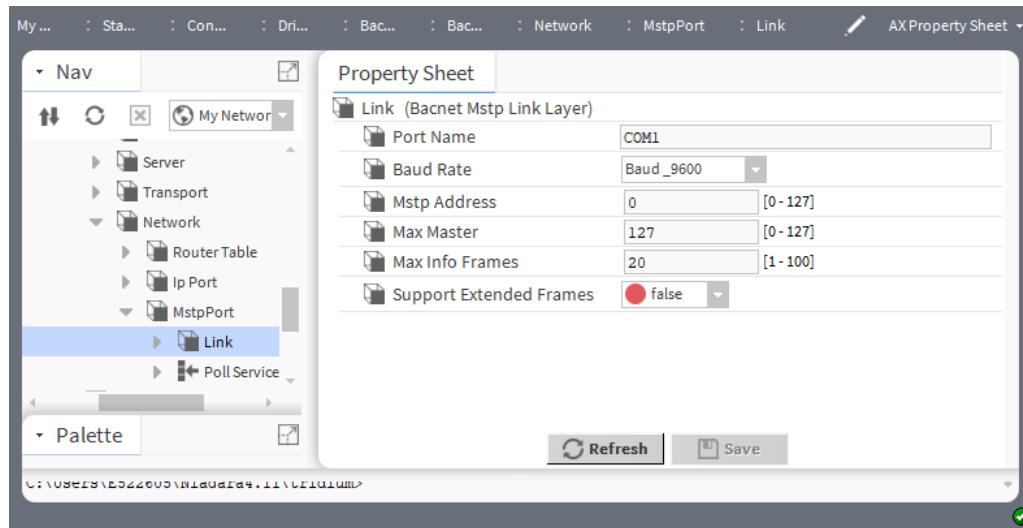
To access these properties, drag or copy an **EthernetPort** from the **bacnet** palette (**NetworkPorts** node) to **BacnetNetwork**→**Bacnet Comm**→**Network**.

To update these properties, expand **Config**→**Drivers**→**BacnetNetwork**→**Bacnet Comm**→**Network**→**EthernetPort** and double-clicking **Link**.

| Property | Value | Description |
|---------------------|-----------------------------------|-------------------------------------------------------------|
| Adapter Title | drop-down list (defaults to none) | Selects the Ethernet adapter to use. |
| Adapter Description | text (defaults to none) | Provides additional information about the Ethernet adapter. |
| Adapter Name | text (defaults to none) | Provides a name for the Ethernet adapter. |

Link (bacnet-BacnetMstpLinkLayer)

This component is the Bacnet MS/TP virtual link layer implementation.

Figure 41 Link (Mstp Port) properties

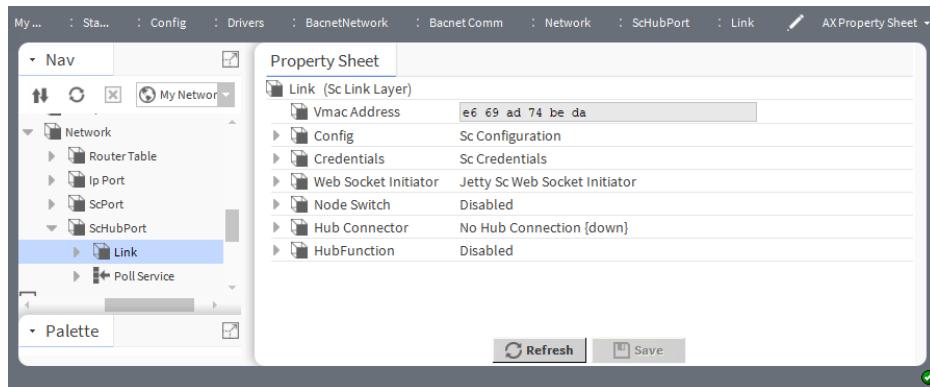
To access these properties, drag or copy an **MstpPort** from the **bacnet** palette (**NetworkPorts** node) to **BacnetNetwork**→**Bacnet Comm**→**Network**.

To update these properties, expand **Config**→**Drivers**→**BacnetNetwork**→**Bacnet Comm**→**Network**→**MstpPort** and double-clicking **Link**.

| Property | Value | Description |
|-------------------------|---------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| Port Name | text | Defines the communication port to use: none, COM2 or COM3. |
| Baud Rate | drop-down list (defaults to Baud9600) | Defines communication speed in bits per second. |
| Mstp Address | number in the range from 0 to 127 (defaults to 0) | Sets a unique, 8-bit number, BACnet MAC (Media Access Control) address on an MSTP trunk in decimal. |
| Max Master | number | Defines the highest known master device on the network, with possible room for expansion, if needed. |
| Max Info Frames | number | Controls how many messages are sent before passing the token, and may be increased up to 50 to increase performance in some cases. |
| Support Extended Frames | true or false (default) | Enables and disables the use of larger frames, which, if supported by the device, may improve performance. |

Link (bacnet-ScLinkLayer)

This component is an instance of a BACnet secure communication virtual link layer.

Figure 42 Link (SCPort) properties

To access these properties, drag or copy an **ScPort** from the **bacnet** palette (under the **NetworkPorts** node) to **BacnetNetwork**→**Bacnet Comm**→**Network**.

To update these properties, expand **Config**→**Drivers**→**BacnetNetwork**→**Bacnet Comm**→**Network**→**ScPort** and double-click **Link**.

For property descriptions, refer to the properties dictionary chapter in this document.

| Property | Value | Description |
|--------------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Vmac Address | six bytes (six octets) | Configures a unique Virtual MAC (VMAC) address for the device. The driver initially sets this value to a random set of six bytes (six octets). You should not need to change this number. |

Actions

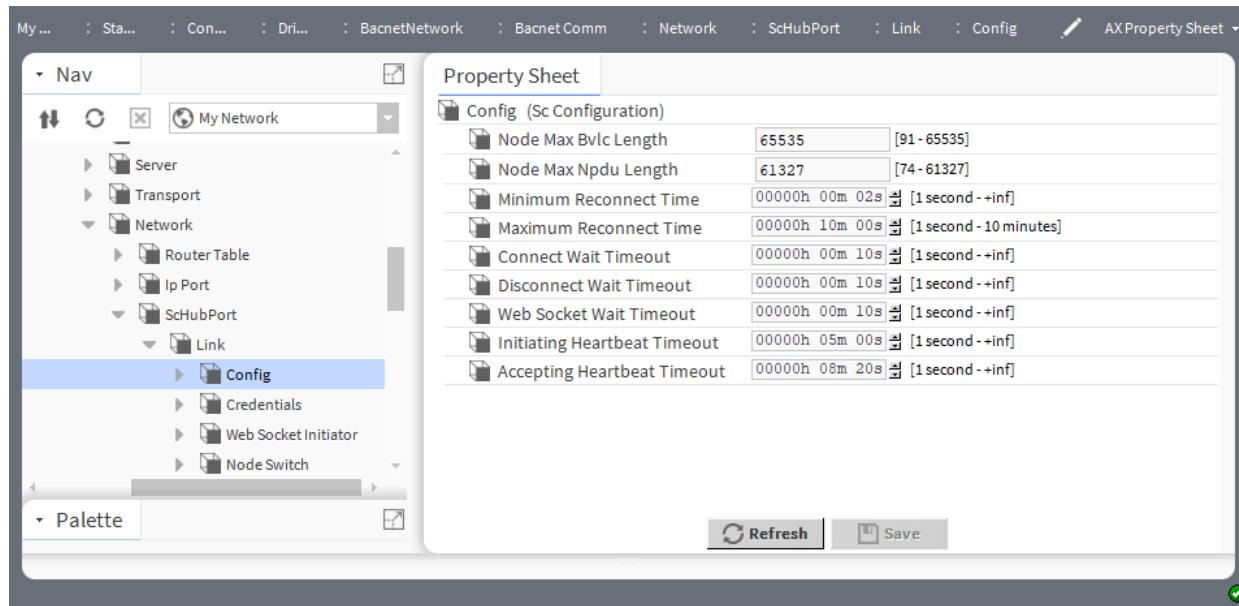
There are two actions on a **Link** component.

- **Change Vmac Address** opens a window for changing the Vmac address.
- **Regenerate Vmac Address** generates a new Vmac address with a random value.

Config (bacnet-ScConfiguration)

This component configures secure connect timings.

Figure 43 Config properties



To access these properties, expand **BacnetNetwork**→**Bacnet Comm**→**Network**, then expand either **ScHubPort** or **ScPort**, expand **Link** and double-click **Config**.

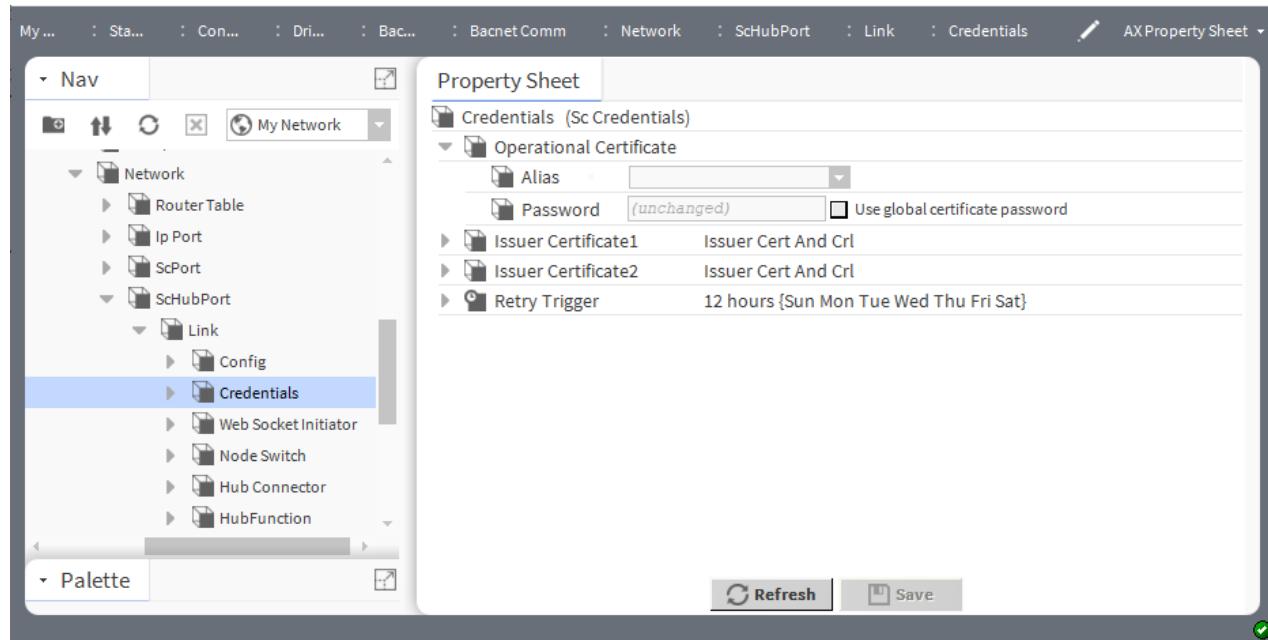
| Property | Value | Description |
|-------------------------|-------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Node Max Bvlc Length | number (defaults to 65535) | Specifies the maximum BVLC (BACnet Virtual Link Control) message size that can be received and processed by the node in bytes. |
| Node Max Npdu Length | number (defaults to 61327) | Specifies the maximum NPDU (Network Protocol Data Unit) message size that can be handled by the node's network entity in bytes. This value must be less than the Node Max Bvlc Length . The difference determines the maximum size of BVLC destination and data options. |
| Minimum Reconnect Time | hours minutes seconds (defaults to two seconds) | <p>Defines the shortest amount of time to wait following a failure before attempting the connection again.</p> <p>A second consecutive primary connection failure waits an amount of time that is double the minimum. A third failure waits four times the minimum, a fourth eight times the minimum, etc. up to the Maximum Reconnect Time.</p> <p>Once it reaches the maximum, which defaults to 10 minutes, the connection attempts to reconnect every maximum amount of time forever.</p> |
| Maximum Reconnect Time | hours minutes seconds (defaults to two minutes) | Defines the longest amount of time to wait following a connection failure before attempting to connect again. |
| Connect Wait Timeout | hours minutes seconds (defaults to ten seconds) | Defines how long a connection waits for an initial response message before concluding that the attempt failed. |
| Disconnect Wait Timeout | hours minutes seconds (defaults to ten seconds) | Defines how long a node waits for a response message while disconnecting. |

| Property | Value | Description |
|------------------------------|---------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Web Socket Wait Timeout | hours minutes seconds (defaults to ten seconds) | Defines how long a connection waits without success to establish a web socket connection before concluding that the attempt failed. |
| Initiating Heartbeat Timeout | hours minutes seconds (defaults to five minutes) | Defines how long an initiating hub or direct connection waits without receiving any messages before sending a heartbeat request message. If no messages are received after another timeout period, the connection will be disconnected. |
| Accepting Heartbeat Timeout | hours minutes seconds (defaults to eight minutes) | Similar to Initiating Heartbeat Timeout except that it applies to accepting hub and direct connections. The value of this timeout is typically longer than the Initiating Heartbeat Timeout so that the initiating connections usually send heartbeat request messages. |

Credentials (bacnet-ScCredentials)

This component configures Secure Connect credentials for the BACnet secure ports (**ScPort** and **ScHubPort**).

Figure 44 Credentials properties



To access these properties, expand **BacnetNetwork**→**Bacnet Comm**→**Network**, then expand either **ScHubPort** or **ScPort**, expand **Link** and double-click **Credentials**.

| Property | Value | Description |
|-------------------------|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Operational Certificate | text (read-only) | Identifies the client certificate used by a node to initiate hub and direct connections. |
| Alias | drop-down list | Selects the operational certificate to be used. |
| Password | text and check box | As of Niagara 4.13, the certificate is password-protected by a unique password or the global certificate password. Prompts the user to provide the user-defined password or the global certificate password associated with the certificate. |

| Property | Value | Description |
|---------------------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Issuer Certificate1 | additional properties | Identifies the site CA (issuer) certificate for the port. For additional properties, refer to "Issuer Certificate1 and 2 (bacnet-IssuerCertAndCrl)." |
| Issuer Certificate2 | additional properties | Identifies the port's alternate site CA (issuer) certificate when updating the operational or issuer certificates across the network. For additional properties, refer to "Issuer Certificate1 and 2 (bacnet-IssuerCertAndCrl)." |

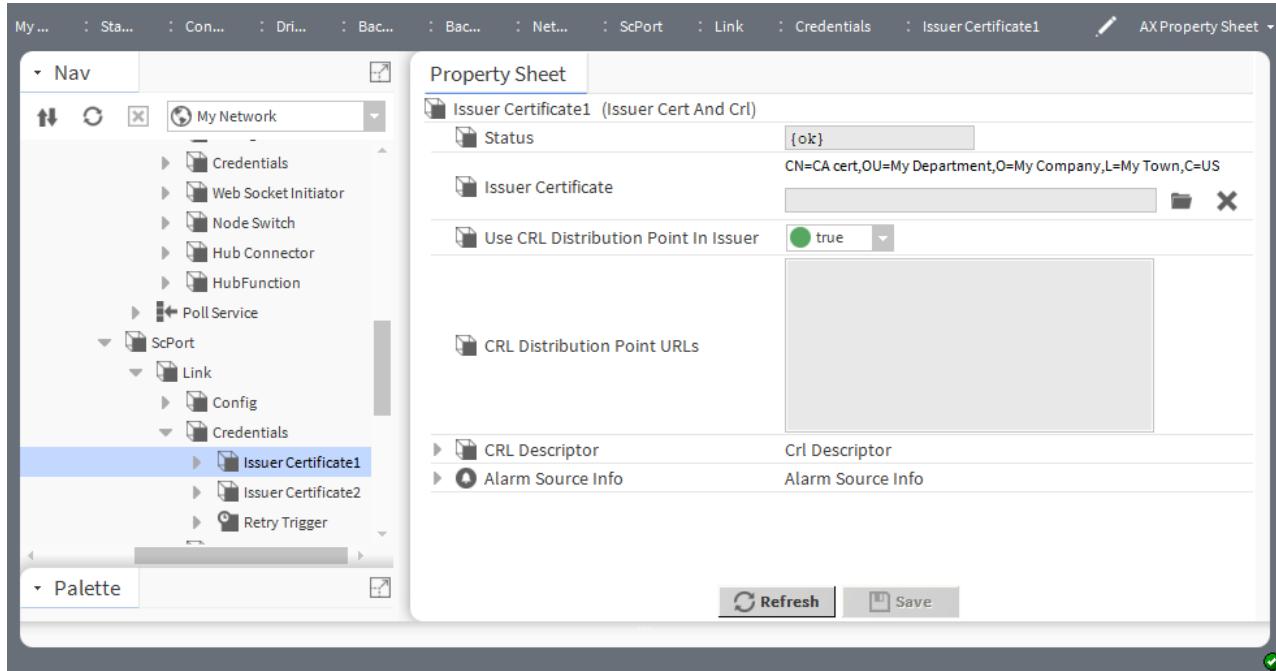
Actions

Retry Failed Crl Retrievals attempts to retrieve CRLs (Certificate Revocation List) that failed to be retrieved.

Issuer Certificate1 and 2 (bacnet-IssuerCertAndCrl)

This component configures the port's issuer certificate. It is the same for both **Issuer Certificate1** and **Issuer Certificate2**.

Figure 45 Issuer Certificate properties



To access these properties, expand **BacnetNetwork**→**Bacnet Comm**→**Network**, then expand either **ScHubPort** or **ScPort**, expand **Link**→**Credentials** and double-click **Issuer Certificate1**.

In addition to the standard property, (Status and Alarm Source Info), these properties support the issuer certificate.

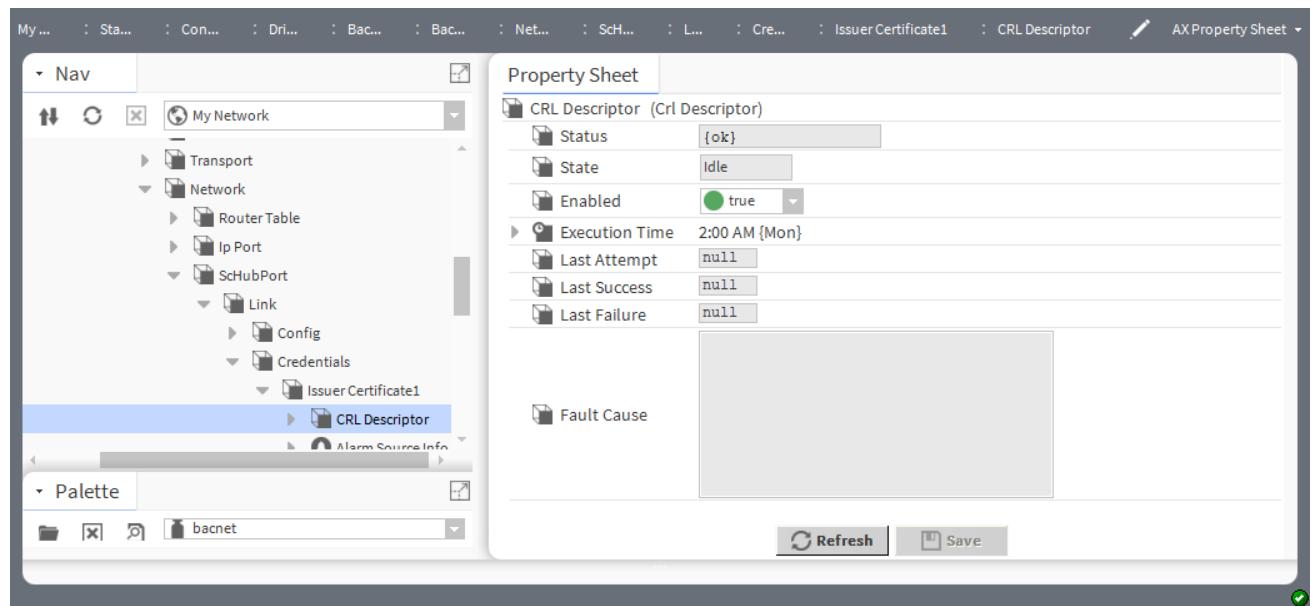
| Property | Value | Description |
|--------------------------------------|-------------------------|---------------------------------------------------------------------|
| Issuer Certificate | File Chooser | Defines the issuer certificate for the port. |
| Use CRL Distribution Point In Issuer | true (default) or false | Uses the URL defined in the issuer certificate for retrieving CRLs. |

| Property | Value | Description |
|------------------------------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CRL Distribution Point URLs. | Text value | Specifies URLs used to retrieve CRLs. This property is read-only if you set Use CRL Distribution Point In Issuer to true and it displays the CRL URLs defined in the issuer certificate. Otherwise, you can enter the URLs manually. |
| CRL Descriptor | additional properties | Attempts to download a CRL from the CRL distribution points configured in CRL Distribution Point URLs . |

CRL Descriptor (bacnet-CrlDescriptor)

This component manages and reports on the CRL descriptor.

Figure 46 CRL Descriptor properties



In addition to the standard properties (Status, Enabled and Fault Cause), these properties support the CRL Descriptor.

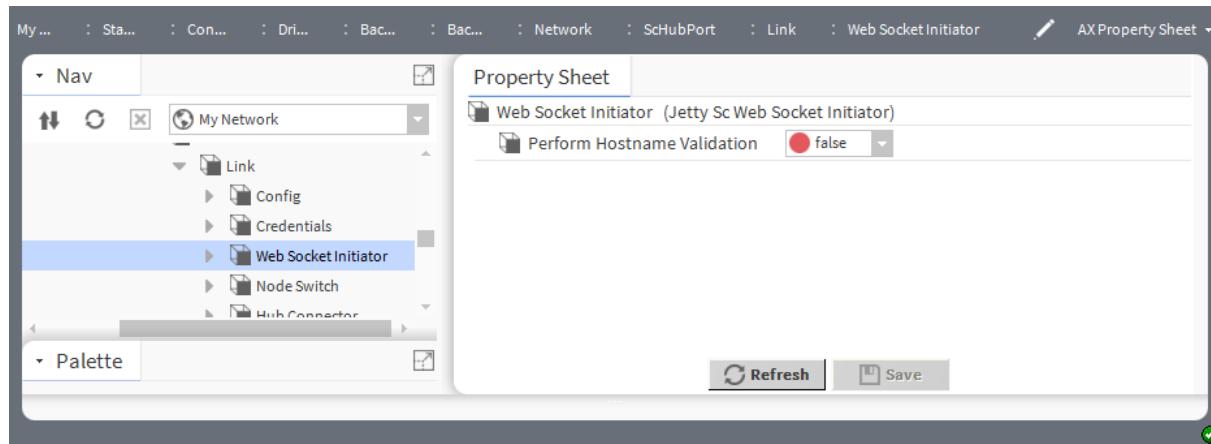
| Property | Value | Description |
|----------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| State | read-only | Reports the current state of the component. |
| Execution Time | additional properties | Configures the frequency with which data are imported from a remote trend log (or trend log multiple) into a history (or histories). Data import may occur at regular intervals (Interval), Daily or when manually requested (Manual). Time Of Day specifies a specific daily time. Randomization configures an import based on time. Days Of Week specifies on which days to apply the import time. |
| Last Attempt | read-only | Reports the date and time of the last attempted execution. |

| Property | Value | Description |
|--------------|-----------|------------------------------------------------------------------------------------------------------------|
| Last Success | read-only | Reports the last time the station successfully performed this function. |
| Last Failure | read-only | Reports the last time the system failed to perform this function. Refer to Fault Cause for details. |

Web Socket Initiator (bacnet-JettyScWebSocketInitiator)

This component supports the port link function.

Figure 47 Web Socket Initiator properties



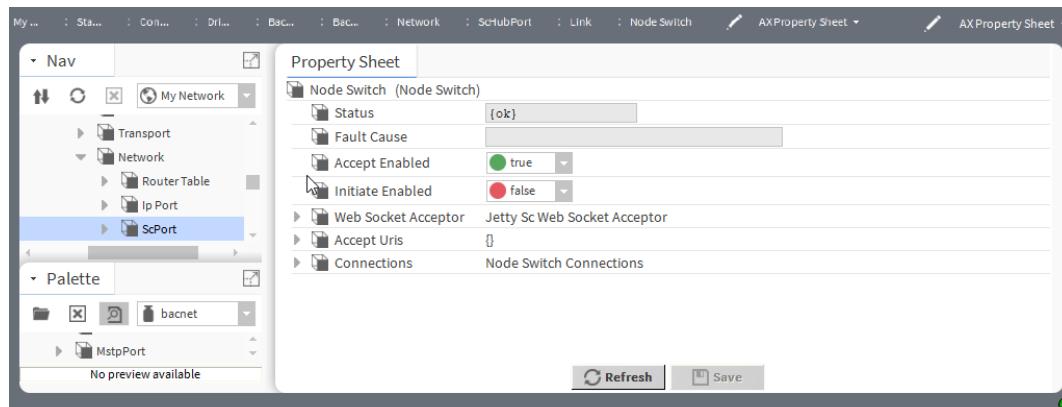
To access these properties, expand **Config**→**Drivers**→**BacnetNetwork**→**Bacnet Comm**→**Network**, then expand either **ScHubPort** or **ScPort**, expand Link and double-click **Web Socket Initiator**.

| Property | Value | Description |
|-----------------------------|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Perform Hostname Validation | true or false (default) | Enables verification that the DNS name in the subjectAltName field of the server certificate matches the host portion of the URL used to make the request. |

Node Switch (bacnet-NodeSwitch)

This component manages the initiated and accepted direct connections made with other SC nodes.

Figure 48 Node Switch properties



To access these properties, expand **Config**→**Drivers**→**BacnetNetwork**→**Bacnet Comm**→**Network**→**ScPort** and double-click **Node Switch**.

In addition to the standard properties (Status and Fault Cause), these sets of properties configure a node switch.

| Property | Value | Description |
|---------------------|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Accept Enabled | true or false (default) | Determines if this port can accept direct connections from other nodes (true) or not (false). To support accepting connections, a BACnet/SC user must be associated with this port. |
| Initiate Enabled | true or false (default) | Determines whether (true) or not (false) this port is enabled to initiate direct connections to other nodes. |
| Web Socket Acceptor | additional properties | Configure the web socket servlet used to accept direct connections from other SC nodes. For property descriptions, refer to "Node Switch/Hub Function, Web Socket Acceptor (bacnet-JettyScWebSocketAcceptor)." |
| Accept Uris | array | Identifies the possible URIs that an initiating SC node can use to make a direct connection to this node. The values in this property are sent in the address resolution ack message. |
| Connections | additional properties | Configure node switch connections. For these properties, refer to "Connections (bacnet-NodeSwitchConnections)." |

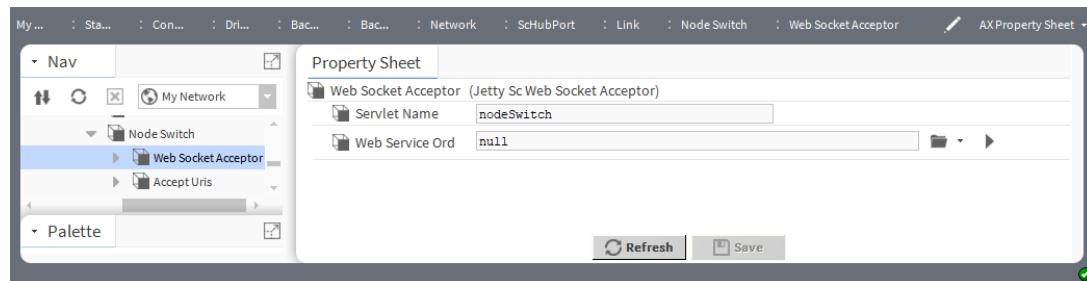
Actions

- **Add Sc User** adds a secure connect user to the **UserService**. If a user is already associated with BACnet/SC, you receive an error message. No message confirms the addition.
- **Remove Sc User** removes a secure connect user from the **UserService**. No message confirms the deletion.

Web Socket Acceptor (bacnet-JettyScWebSocketAcceptor)

This component supports both the Node Switch and Hub Function.

Figure 49 Web Socket Acceptor properties



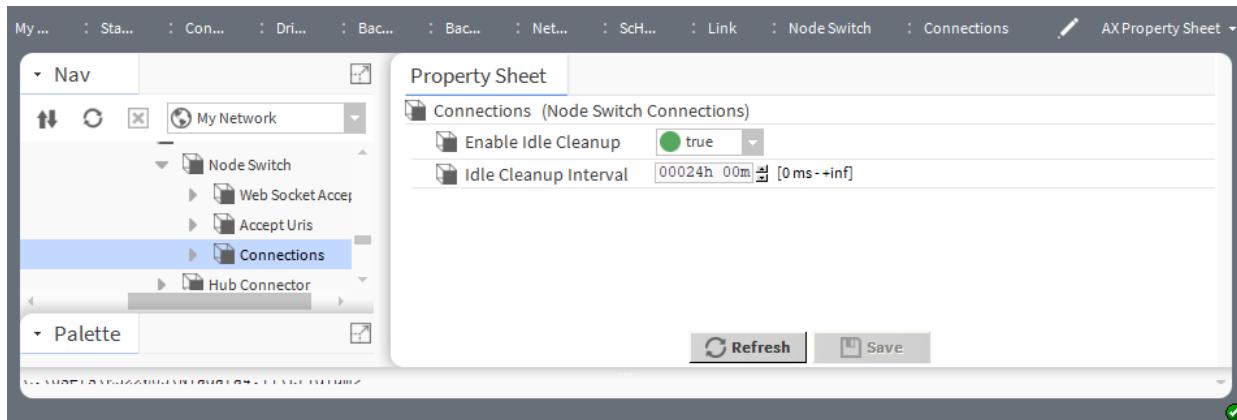
To access these properties, expand **Config**→**Drivers**→**BacnetNetwork**→**Bacnet Comm**→**Network**→**Sc-Port**→**Node Switch** and double-click **Web Socket Acceptor**.

| Property | Value | Description |
|-----------------|-------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Servlet Name | text (defaults to nodeSwitch or hub depending on where the component is used) | Defines the path to the web socket servlet that accepts either direct (node switch) or hub (hub function) connections. |
| Web Service Ord | ORD | Configures the web service Ord of type BOrd. When you set it to <code>BOrd.NULL</code> , the web socket acceptor registers on the station's default web service and listens on that service's port. You can also configure the Web Service Ord to point to an alternate web service, which enables the web socket servlets to listen on a different port than the default web service. |

Connections (bacnet-NodeSwitchConnections)

This component serves in multiple locations to display and clean up connections.

Figure 50 Connections properties



The screen capture shows the **Connections** component under the **Node Switch**. It also appears under the **Hub Function**.

To access these properties, expand **Config→Drivers→BacnetNetwork→Bacnet Comm→Network→Sc-Port→Node Switch** and double-click **Connections**.

| Property | Value | Description |
|-----------------------|------------------------------------------|-----------------------------------------------------|
| Enable Idle Cleanup | true (default) or false | Turns connection cleanup on (true) and off (false). |
| Idle Cleanup Interval | hours and minutes (defaults to 24 hours) | Configures when to clean up connections. |

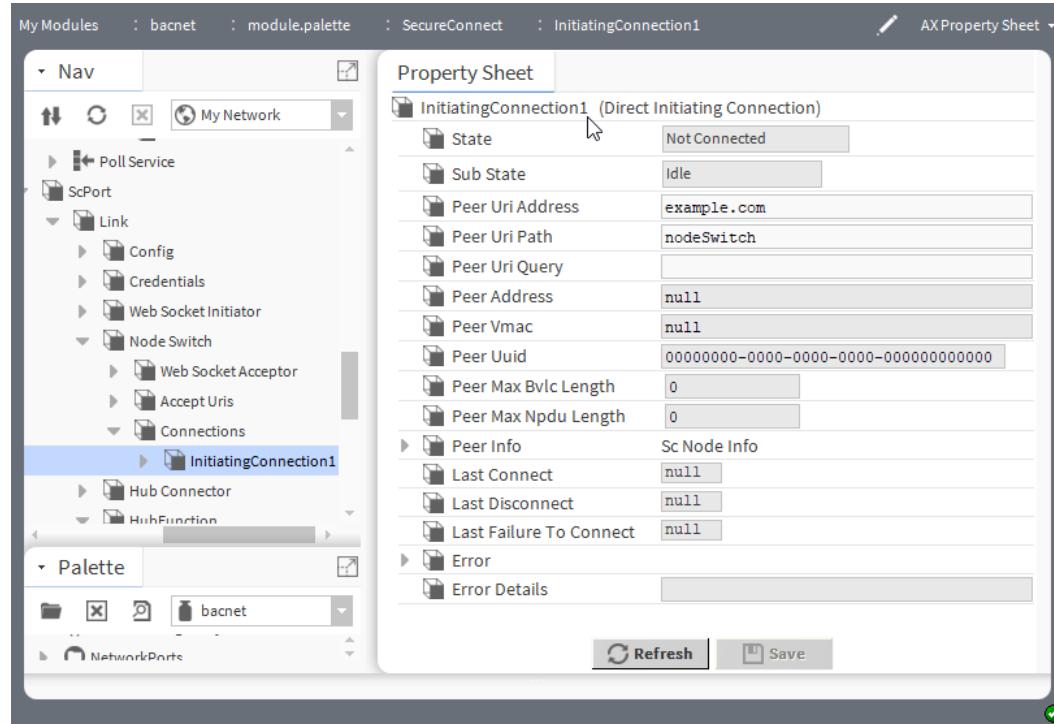
Actions

- **Add Connection (Node Switch only)** allows optional configuration of a new `DirectInitiatingConnection` that is added to the **Connections** component.
- **Disconnect All** disconnects all connected hub-accepting connections on the **Hub Function** and direct-initiating and -accepting connections on the **Node Switch**.
- **Remove All Idle Accepted** deletes all hub accepting connections on the hub function. On the node switch, it deletes all direct accepting connections that are disconnected in the idle state.

Direct Initiating Connection (bacnet-DirectInitiatingConnection)

This component configures a direct initiating connection between two BACnet/SC nodes.

Figure 51 Direct Initiating Connection properties



To access these properties, expand a secure connect port, expand **Link→Node Switch→Connections** and double-click the **InitiatingConnection1** component.

| Property | Value | Description |
|----------------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| State | read-only | Indicates the current high-level state of the connection. |
| Sub State | read-only | Indicates the current detailed state of the connection. |
| Peer Uri Address | text | Constitutes the authority component (host and port) of the connection's web socket.. |
| Peer Uri Path | text | Specifies the path component of the connection's web socket URI. |
| Peer Uri Query | text | Indicates the query component of the connection's web socket URI. |
| Peer Address | Internet address | Specifies the Internet address of the peer node to which this connection is linked. |
| Peer Vmac | number | Specifies the VMAC address of the peer node to which this connection is linked. |
| Peer Uuid | number | Constitutes the device UUID of the peer node to which this connection is linked. |
| Peer Max Bvlc Length | read-only | Reports the maximum length of the BVLC (BACnet Virtual Link Control) message that can be received and processed by the target node in bytes as reported in the Connect-Accept message.. |

| Property | Value | Description |
|-------------------------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Peer Max Npdu Length | read-only | Reports the maximum length of the NPDU (Network Protocol Data Unit) message that can be handled by the target node's network entity in bytes as reported in the Connect-Accept message. |
| Peer Info | additional properties | For property descriptions, refer to Peer Info, page 156 . |
| Last Connect | read-only | Indicates the last time this connection successfully connected. |
| Last Disconnect | read-only | Indicates the last time this connection was disconnected. |
| Last Failure To Connect | read-only | Indicates the last time this connection failed to connect. |
| Error | additional properties | Specifies the error class and code of the most recent error for the connection. |
| Error Details | read-only | Specifies additional details about the most recent error for the connection. |

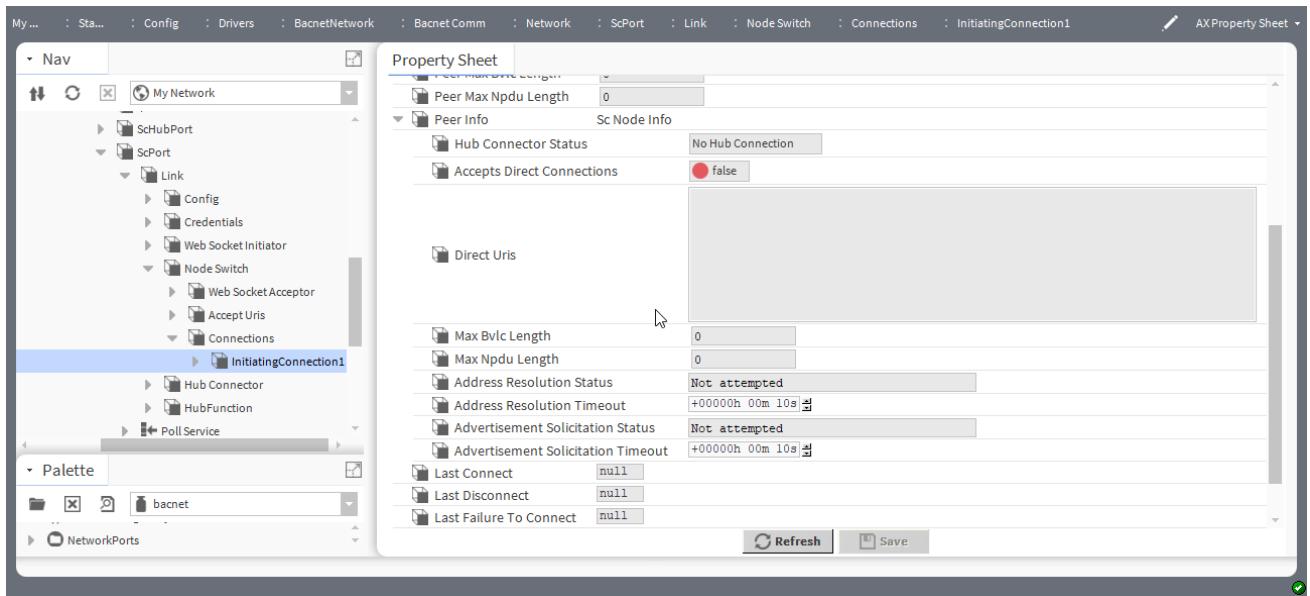
Actions

- **Connect** attempts to make a connection to the peer node.
- **Disconnect** stops the connection with the peer node.
- **Change Peer Vmac** sets the **Peer Vmac** property. This is required for sending Address-Resolution and Advertisement-Solicitation messages if the connection has not yet been established.
- **Send Address Resolution** sends an Address-Resolution message to the peer node based on the **Peer Vmac** property value.
- **Send Advertisement Solicitation** sends an Advertisement-Solicitation message to the peer node based on the **Peer Vmac** property value.

Peer Info

These properties provide information about the peer-to-peer connection.

Figure 52 Peer Info properties



To access these properties, expand **BacnetNetwork**→**Bacnet Comm**→**Network**, expand a secure connect port or hub port, expand **Link**→**Node Switch**→**Connections**, double-click an initiating connection and expand **Peer Info**.

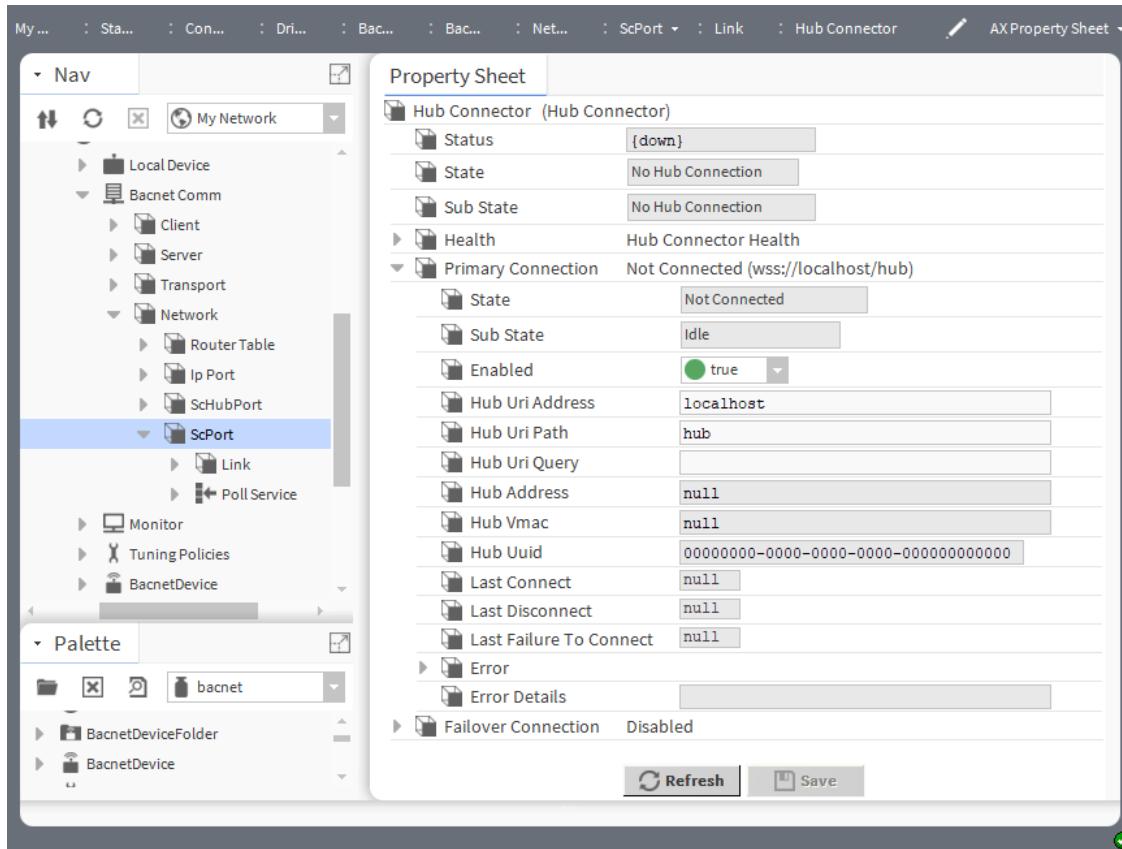
| Property | Value | Description |
|----------------------------|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Hub Connector Status | read-only | Reports the condition of the target node's connection. |
| Accepts Direct Connections | read-only | Indicates if the target node is capable of accepting direct connections. |
| Direct UrIs | read-only | Identifies the possible URIs that the initiating connection can use to make a direct connection to the target node. The values in this property are received in the Address-Resolution-Ack message. |
| Max Bvlc Length | read-only | Reports the maximum length of the BVLC (BACnet Virtual Link Control) message that can be received and processed by the target node in bytes as reported by in the Advertisement message. |
| Max Npdu Length | read-only | Reports the maximum length of the NPDU (Network Protocol Data Unit) message that can be handled by the node's network entity of the target node in bytes as reported in the Advertisement message. |
| Address Resolution Status | read-only | Reports the status of the last Address-Resolution message sent to the target node. |
| Address Resolution Timeout | hours minutes seconds (defaults to 10 seconds) | Configures how long to wait for an Address-Resolution-ACK message after sending an Address-Resolution message to the target node before timing out. |

| Property | Value | Description |
|------------------------------------|------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| Advertisement Solicitation Status | read-only | Reports the status of the last Advertisement-Solicitation message sent to the target node. |
| Advertisement Solicitation Timeout | hours minutes seconds (defaults to 10 seconds) | Configures how long to wait for an Advertisement message after sending an Advertisement-Solicitation message to the target node before timing out. |

Link, Hub Connector (bacnet-HubConnector)

This component configures a primary hub and, optionally, a failover hub. The same component is present in an **ScPort** and **ScHubPort**.

Figure 53 Hub Connector properties



To access these properties, expand **BacnetNetwork**→**Bacnet Comm**→**Network**, then expand either **ScPort** or **ScHubPort**, expand **Link** and double-click **Hub Connector**.

In addition to the standard properties (Status and Health), these properties configure the hub connector.

| Property | Value | Description |
|-----------|-----------|-----------------------------------------------------------------|
| State | read-only | Indicates the current high-level state of the hub connector. |
| Sub State | read-only | Indicates the current more detailed state of the hub connector. |

| Property | Value | Description |
|---------------------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Primary Connection | additional properties | Configures the connection to the primary hub. For a description of these properties, refer to "Hub Connector connection (bacnet-HubInitiatingConnection)." |
| Failover Connection | additional properties | Configures the connection to the optional failover hub. For a description of these properties, refer to "Hub Connector connection (bacnet-HubInitiatingConnection)." |

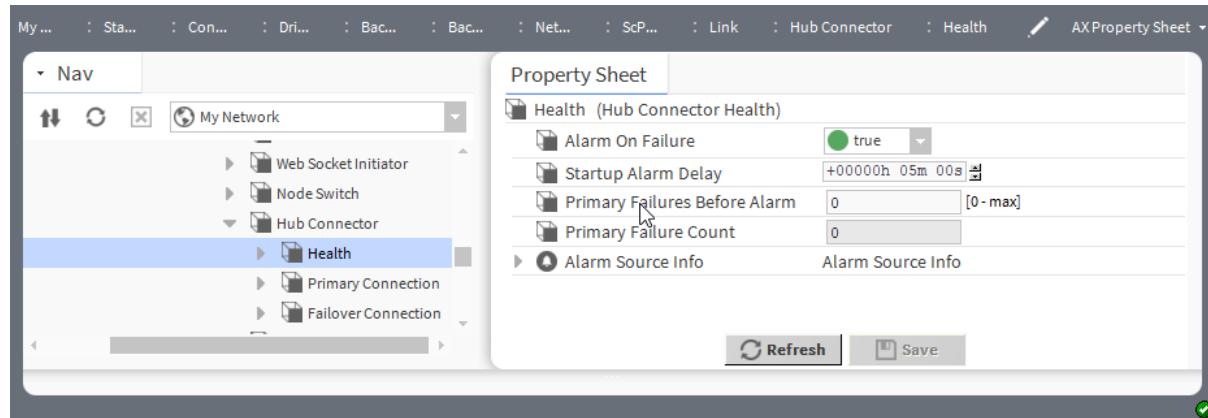
Actions

- **Force Connect** causes an immediate re-connection attempt to the primary hub when the hub connector is not connected to any hub or is connected to the failover hub. An error will be displayed if the hub connector is not in one of these sub states.
- **Disconnect** causes the active hub connection to be disconnected. This may result in a re-connection attempt to the primary hub.

Health (bacnet-HubConnectorHealth)

This component configures the hub connector health. The same component is present in an **ScPort** and **ScHubPort**.

Figure 54 Health properties



To access these properties, expand **BacnetNetwork**→**Bacnet Comm**→**Network**, then expand either **ScPort** or **ScHubPort**, expand **Link**, double-click **Hub Connector** and double-click **Health**.

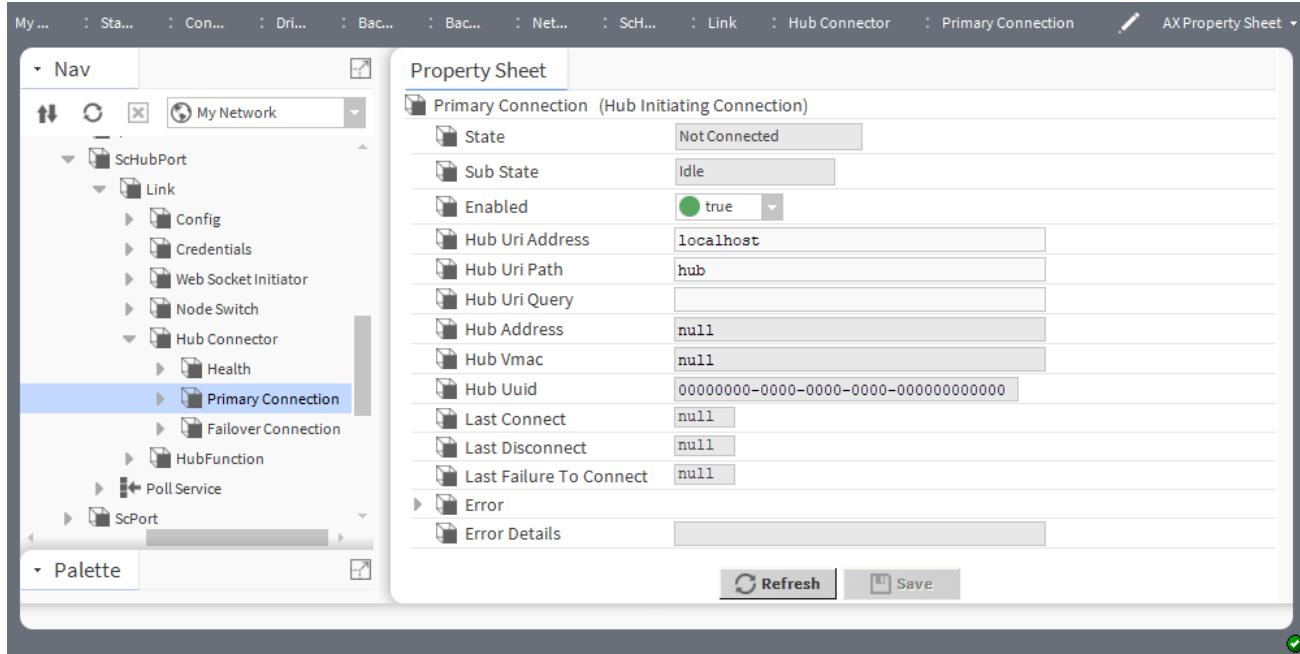
| Property | Value | Description |
|-------------------------------|-----------------------------------------------|---------------------------------------------------------------------------------|
| Alarm On Failure | true (default) or false | Activates or deactivates the alarm upon failure function. |
| Startup Alarm Delay | hours minutes seconds (defaults to 5 minutes) | Specifies the duration of the alarm delay upon startup. |
| Primary Failures Before Alarm | numeric value | Indicates the number of primary connection failures before the alarm is raised. |
| Primary Failure Count | numeric value | Indicates the number of primary connection failures. |

Hub Connector connection (bacnet-HubInitiatingConnection)

Two components configure the hub connector: Primary and Failover. The primary connection is configured to connect to the primary hub. The driver attempts this connection before attempting the failover

connection to the failover hub. When connected to the failover hub, the driver re-attempts the primary connection again and, if re-established, it drops the failover connection.

Figure 55 Primary Connection properties



To access these properties, drag or copy an **ScHubPort** from the **bacnet** palette (**NetworkPorts** node) to **BacnetNetwork**→**Bacnet Comm**→**Network**.

To update these properties, expand **Config**→**Drivers**→**BacnetNetwork**→**Bacnet Comm**→**Network**→**ScHubPort**→**Link**→**Hub Connector** and double-click **Primary Connection** or **Failover Connection**.

| Property | Value | Description |
|------------------|-------------------------|---------------------------------------------------------------------------------------------------------|
| Connection State | read-only | Indicates the current high-level state of the connection. |
| Sub State | read-only | Indicates the current detailed state of the connection. |
| Enabled | true (default) or false | Activates (true) and deactivates (false) use of the object. |
| Hub URI Address | text | Constitutes the authority component (host and port) of the connection's web socket URI. |
| Hub URI Path | filepath | Specifies the path component of the connection's web socket URI. |
| Hub URI Query | text value | Indicates the query component of the connection's web socket URI. |
| Hub Vmac | read-only | Specifies the Vmac address of the node that hosts the hub function to which this connection is linked. |
| Hub Uuid | read-only | Constitutes the device UUID of the node that hosts the hub function to which this connection is linked. |
| Hub Address | read-only | Specifies the Internet address of the hub function to which this connection is linked. |
| Last Connect | read-only | Indicates the last time this connection successfully connected. |

| Property | Value | Description |
|--------------------|--------------------------------|------------------------------------------------------------------------|
| Last Disconnect | read-only | Indicates the last time this connection was disconnected. |
| Last Failure | read-only timestamp | Indicates the last time this connection failed to connect. |
| Error, Error Class | read-only, numeric class error | Specifies the error class of the most recent error for the connection. |
| Error, Error Code | read-only | Indicates the error code for the most recent error for the connection. |
| Error Details | read-only | Specifies the nature of the error. |

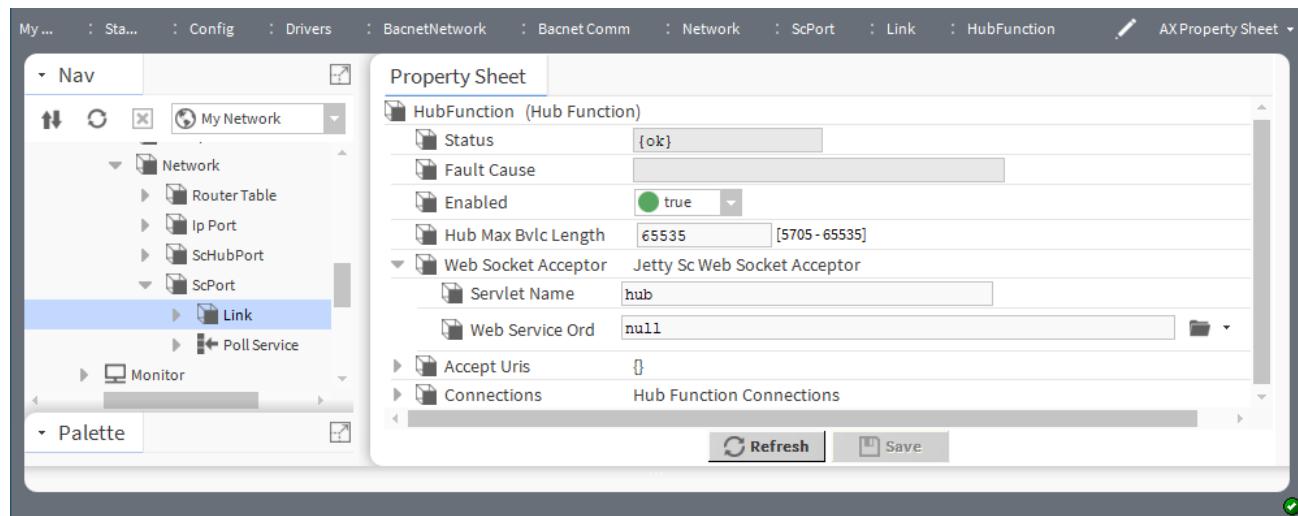
Actions

- **Disconnect** disconnects the connection to a hub. This may cause the hub connector to try to reconnect.

Hub Function (bacnet-HubFunction)

This component configures the hub function.

Figure 56 Hub Function properties



To access these properties, drag or copy a **Hub Function** component from the **bacnet** palette (**Secure-Connect** node) to **BacnetNetwork**→**Bacnet Comm**→**Network**→**ScHubPort**→**Link** folder.

To update these properties, expand **Config**→**Drivers**→**BacnetNetwork**→**Bacnet Comm**→**Network**→**ScHubPort**→**Link** and double-click **Hub Function**.

In addition to the standard property (Enabled), these properties support this component.

| Property | Value | Description |
|---------------------|---------------|-------------------------------------------------------------------------------------|
| Hub Max Bvlc Length | numeric value | Specifies the maximum Bvlc message size that the hub function can process in bytes. |

Actions

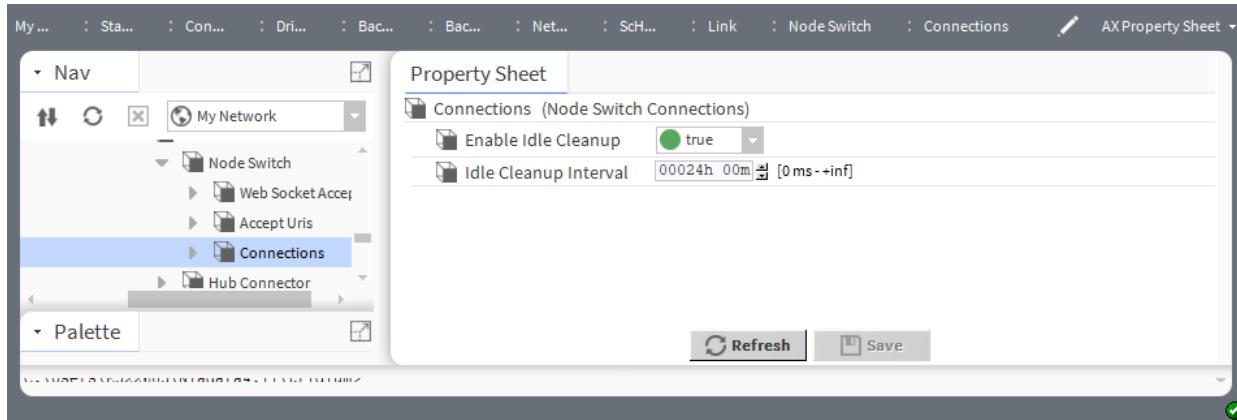
- **Add Sc User** adds a secure connect user to the **UserService**. If a user is already associated with BACnet/SC, you receive an error message. No message confirms the addition.
- **Remove Sc User** removes a secure connect user from the **UserService**. No message confirms the deletion.

- **Configure Local Hub Connector** configures the settings for the primary or the failover connections.

Connections (bacnet-HubFunctionConnections)

This component configures the hub function connections. It supports the **ScHubPort**.

Figure 57



To access these properties, expand **Config→Drivers→BacnetNetwork→Bacnet Comm→Network→ScHubPort→Link→Hub Function**, and double-click **Connections**.

| Property | Value | Description |
|-----------------------|-------------------------------------------|------------------------------------------------------------------------|
| Enable Idle Cleanup | true (default) or false | Turns connections cleanup on (true) and off (false). |
| Idle Cleanup Interval | hours minutes seconds (defaults to 1hour) | Configures when to clean up connections. |

Actions

- **Disconnect All** disconnects all connected hub-accepting connections on the **Hub Function** and direct-initiating and -accepting connections on the **Node Switch**.
- **Remove All Idle Accepted** deletes all hub accepting connections on the hub function or direct accepting connections on the node switch that are disconnected (in the idle state).

Poll Service (bacnet-BacnetMultiPoll)

This component configures and manages a group of BACnet proxy points to be polled. Each BACnet network port (Ip Port, EthernetPort, MstpPort, ScPort and ScHubPort) provides its own **PollService** with three polling rates (**Fast**, **Normal**, and **Slow**). This varies from other network drivers that use a single **PollService**.

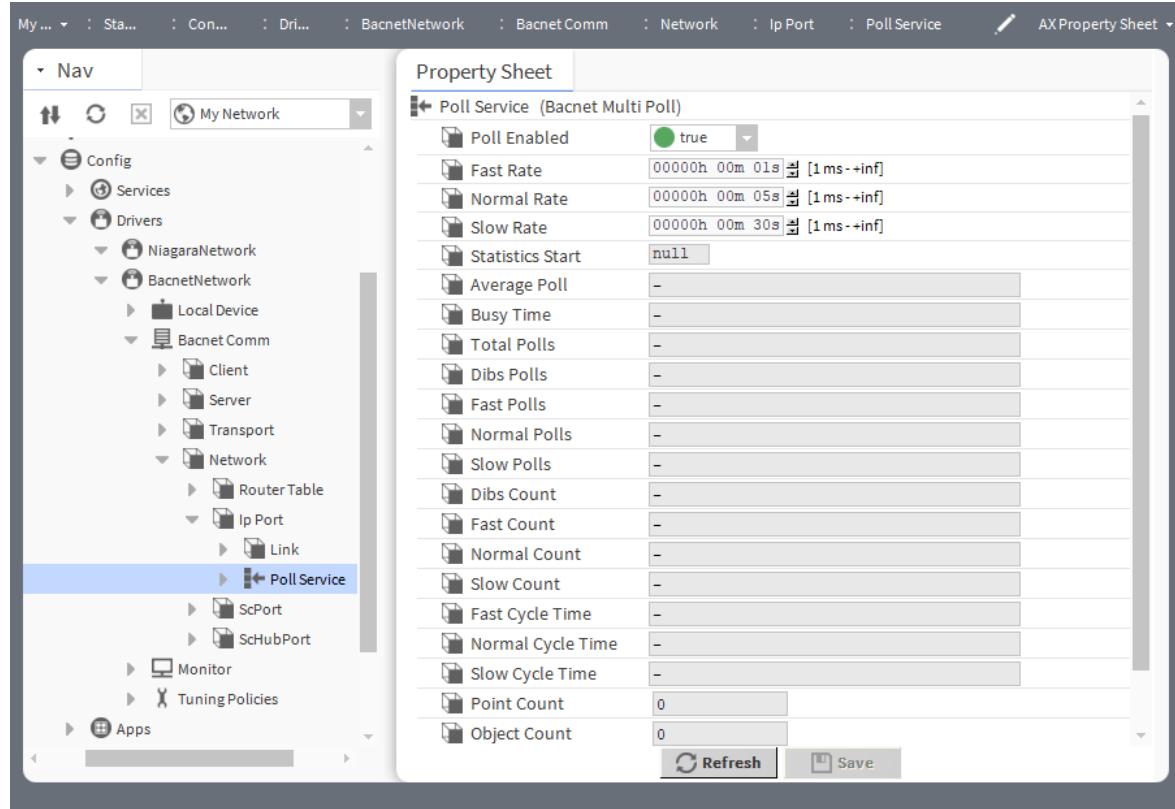
In addition to the three rates of speed (fast, normal, and slow), a fourth speed refers to the dibs stack. When a point is subscribed, it immediately gets first dibs and goes to the top of the dibs stack. The poll scheduler always polls the dibs before doing anything else using last-in, first-out (LIFO). This means that as long as entries are in the dibs stack, the driver polls them as fast as possible with no artificial delays.

When the dibs stack is empty, the scheduler attempts to poll the components using an algorithm designed to create uniform network traffic. For example, if the **Sets the target polling interval for devices that can be polled (they are pollable)** and assigned to the **Fast Rate group**. is configured to 5000 ms and five components are currently subscribed at this speed, the scheduler attempts to poll each component with a second of delay.

Every ten seconds the poll scheduler rechecks the speeds for configuration changes. This means that if you change a point's configuration from slow to fast, it takes at most ten seconds for the change to take effect.

The driver updates statistics every ten seconds. You may manually reset statistics using the reset statistics action.

Figure 58 Poll Service properties



You access these properties by expanding **Config**→**Drivers**, the BACnet network **Bacnet Comm**→**Network** followed by expanding a port (**Ip Port**, **EthernetPort** or **MstpPort**) and double-clicking the **Poll Service** node.

| Property | Value | Description |
|--------------|------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Enabled | true (default) or false | Determines user control over polling. true enables a system user to use the Live Updates (play) button in History views to poll for live data for the associated imported history(ies). false renders this button unavailable in history views for the associated imported history(ies). |
| Fast Rate | hours minutes seconds (defaults to 1 second) | Sets the target polling interval for devices that can be polled (they are pollable) and assigned to the Fast Rate group. |
| Normal Rate | hours minutes seconds (defaults to 5 seconds) | Sets the target polling interval for devices that can be polled (they are pollable) and assigned to the Fast Rate group. |
| Slow Rate | hours minutes seconds (defaults to 30 seconds) | Sets the target polling interval for devices that can be polled (they are pollable) and assigned to the Fast Rate group. |

| Property | Value | Description |
|-------------------|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Statistics Start | read-only timestamp | Reports either the last manual reset of poll statistics or, if statistics have not been reset, the first steady state time immediately following the last station restart. |
| Average Poll | read-only | Reports the average time spent during each poll event. This does not relate to the total time required to complete a poll cycle for any of the three rates. It is the time spent polling a given group of objects before pausing and switching to another group either using the same or a different poll rate. |
| Busy Time | read-only | <p>Displays a percentage of time spent by the poll thread actually polling points using all poll rates. Includes (in parentheses) the ratio of time spent polling/total time since statistics were restarted.</p> <p>Given a small amount of time is spent transitioning among poll rates, and with the thread sleeping to evenly space out polling messages, it is unlikely to ever see Busy Time reach exactly 100%. However, any percentage above 95% indicates that the poll thread is basically spending all of its time actually polling.</p> <p>NOTE: In the case of the Poll Service for a BACnet network port, because two threads are used for polling, it is possible to see a Busy Time approaching 200%. In this case, divide Busy Time in half to get an average busy time for each thread.</p> |
| Total Polls | read only | Reports the total number of polls conducted and the time spent waiting for polls to execute. This time is the same time indicated in the ratio of the Busy Time property. Typically, the total poll count indicates the number of times the PollService polled any object. It is not a running total of the actual poll cycles. |
| Dibs Polls | read-only | Reports the percentage and ratio of the number of DIBS polls versus total polls. |
| Fast Polls | read-only | Reports the number of polls made processing the fast queue. |
| Normal Polls | read-only | Reports the number of polls made processing the normal queue. |
| Slow Polls | read-only | Reports the number of polls made processing the slow queue. |
| Dibs Count | read-only | Reports the current and average number of components in the DIBS stack. (DIBS stands for Distributed Internet Backup System). |
| Fast Count | read-only | Reports the current and average number of components in the fast queue. |
| Normal Count | read-only | Reports the current and average number of components in the normal queue. |
| Slow Count | read-only | Reports the current and average number of components in the slow queue. |
| Fast Cycle Time | read-only | Reports the average cycle time for the fast queue. |
| Normal Cycle Time | read-only | Reports the average cycle time for the normal queue. |

| Property | Value | Description |
|-------------------|-----------|-----------------------------------------------------------------|
| Slow Cycle Time | read-only | Reports the average cycle time for the slow queue. |
| Point Count | read-only | Reports the current and average number of points being polled. |
| Object Count | read-only | Reports the current and average number of objects being polled. |
| Virtual Count | read-only | Reports the virtual count of points and objects being polled. |
| Number of Threads | read-only | Reports the total number of polling threads. |

Actions

- **Enable** starts polling.
- **Disable** suspends polling.
- **Reset Statistics** manually retrieves fresh statistics.
- **Rebuild Poll Lists**

Tuning Policies (bacnet-BacnetTuningPolicyMap)

This component is a container for one or more BACnet tuning policy.

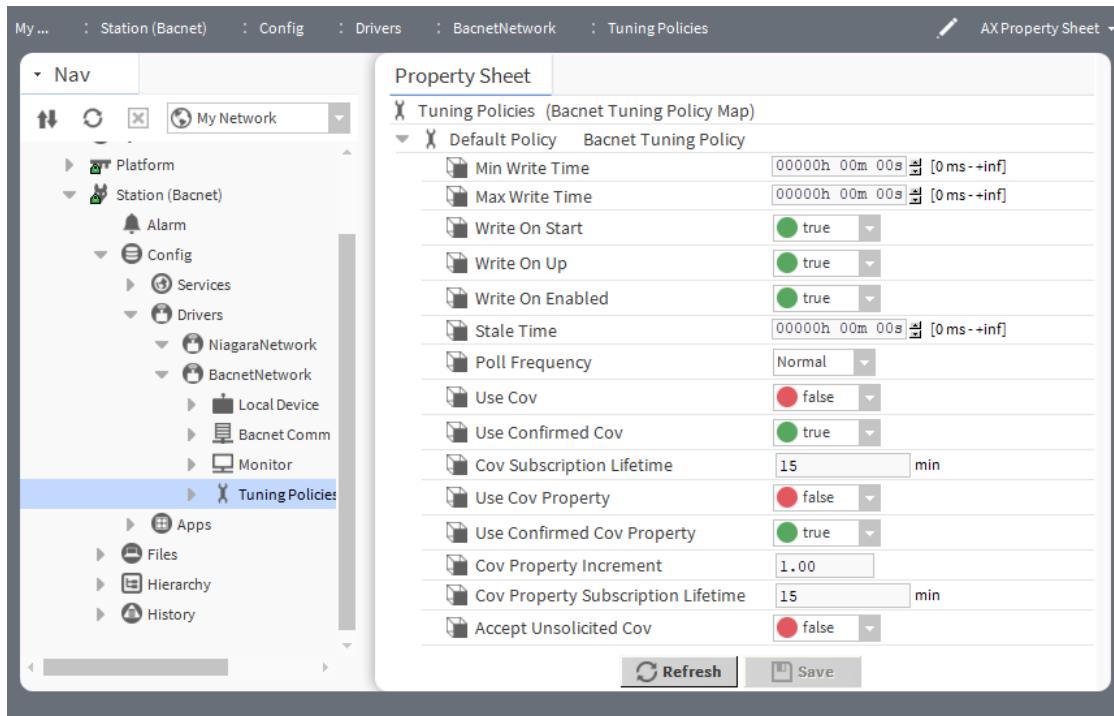
When using the BACnet driver you can create multiple tuning policies and assign specific tuning policies to BACnet proxy points as needed based on the **Bacnet Comm→Network** port used (Ip Port, Ethernet Port, Mstp Port). The BACnet driver also supports virtual components that do not exist in the station. It is possible to assign the BACnet virtual components to the different tuning policies, where it requires encoding of the tuning policy as a parameter at the end of the ord in the widget binding (in the Px view). For example, `ord;policy=tuningPolicyName`. This is an actual ord in a widget binding: `Station:|slot:/Drivers/BacnetNetwork/AHU_24/virtual|virtual:/analogValue_184/presentValue;policy=myVirtualPolicy`.

Each BACnet network port has its own **PollService**, including three poll rates (**Fast**, **Normal**, and **Slow**). This varies from other driver networks that use a single **PollService**.

For an explanation of driver tuning policies, refer to the *Niagara Drivers Guide*.

Bacnet Tuning Policy (Default Policy) (bacnet-BacnetTuningPolicy)

This component defines a tuning policy for the BacnetNetwork. It includes standard tuning policy properties and additional properties related to client-side usage of the BACnet Subscribe_COV service. For an explanation of driver tuning policies, refer to the *Niagara Drivers Guide*.

Figure 59 Default Policy properties

One way to access these properties by expanding **BacnetNetwork→Tuning Policies** and double-clicking **Default Policy**.

| Property | Value | Description |
|----------------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Min Write Time | time (defaults to 0) | Specifies the minimum amount of time allowed between writes to writable proxy points, especially ones that have one or more linked inputs. This provides a way to throttle rapidly changing values so that only the last value is written. The default value (0) disables this rule causing all value changes to attempt to write. |
| Max Write Time | time (defaults to 0) | Specifies the maximum amount of time to wait before rewriting the value, in case nothing else has triggered a write, to writable proxy points. Any write action resets this timer. The default (0) disables this rule resulting in no timed rewrites. |
| Write on Start | true (default) or false | Determines a writable proxy point's behavior when the station starts. true initiates a write when the station first reaches a steady state. false prevents a write when the station first reaches a steady state. NOTE: Consider setting to false except for critical proxy points, otherwise large networks may experience write-queue-overflow exceptions. |
| Write on Up | true (default) or false | Determines a writable proxy point's behavior when the point and its parent device transition from down to up. |

| Property | Value | Description |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | <p>true initiates a write when a transition from down to up occurs.</p> <p>false prevents a write when a transition from down to up occurs.</p> |
| Write on Enabled | true (default) or false | <p>Determines a writable proxy point's behavior when the point's status transitions from disabled to normal (enabled).</p> <p>true initiates a write when the transition occurs.</p> <p>false prevents a write when the transition occurs.</p> |
| Stale Time | hours minutes seconds (defaults to zero) | <p>Defines the period of time without a successful read (indicated by a read status of {ok}) after which a point's value is considered to be too old to be meaningful (stale).</p> <p>A non-zero value causes the point to become stale (status stale) if the configured time elapses without a successful read, indicated by Read Status {ok}.</p> <p>The default value (zero) disables the stale timer causing points to become stale immediately when unsubscribed.</p> |
| Poll Frequency | drop-down list (defaults to Normal) | <p>References the On Demand Poll Scheduler rates under the NiagaraNetwork's History Policies container slot.</p> <p>Configures how often this type of poll occurs.</p> |
| Use Cov | true or false If the device was discovered, and the station database determined that the device indicates support for server-side COV, this property defaults to true. Otherwise, it defaults to false indicating that no proxy points under the device use COV. | <p>Enables (true) and disables (false) a device's support for COV (change of value) as a way to monitor proxy point values.</p> <p>Assuming the device supports subscription to the COV service, true triggers the driver to attempt the necessary updates (proxy subscriptions) to the value of each point using the BACnet COV subscription to the device. If the subscription attempt succeeds, the Read Status property of the point's BacnetProxyExt displays COV. If the subscription attempt fails, the driver uses normal polling and the Read Status property shows Polled.</p> <p>When true, individual proxy points under the device may use COV subscriptions, depending on their assigned tuning policy.</p> <p>When false, the driver ignores any proxy subscription updates.</p> |
| Use Confirmed Cov | true (default) or false | <p>Controls device updates.</p> <p>If enabled (true), and the assigned proxy points are under a BacnetDevice that supports confirmed COV notifications, the driver attempts any necessary updates (proxy subscriptions) using BACnet confirmed COV subscriptions to the device.</p> <p>If disabled (false), the driver requests unconfirmed COV notifications.</p> |
| Cov Subscription Lifetime | time range (defaults to 15 minutes) | Indicates the lifetime, in minutes, for which the software subscribes for COV notifications, then (if necessary) periodically subscribes again. A value of zero (0) configures an indefinite period of time, although this is not guaranteed to persist across resets of the server device. |

| Property | Value | Description |
|------------------------------------|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Cov Property | true or false (default) | Indicates if the system uses COV Property notification services (true) to receive data about points in this device for which COV Property is supported. |
| Use Confirmed Cov Property | true (default) or false | Indicates if the software requests confirmed (true) or unconfirmed (false) COV notifications on a COV property request. |
| Cov Property Increment | number to two decimal places | Defines an increment for a COV property to apply to the sending of COV notifications. This property applies to numeric COV values only. |
| Cov Property Subscription Lifetime | number of minutes (defaults to 15) | Configures the lifetime, in minutes, for which the software subscribes to the COV property for COV notifications. Zero (0) configures an indefinite period of time, although it is not guaranteed to persist across resets of the server device. |
| Accept Unsolicited Cov | true or false (default) | Allows (true) or prohibits (false) COV notifications to update a point that is being polled. false prevents the unsolicited updates. |

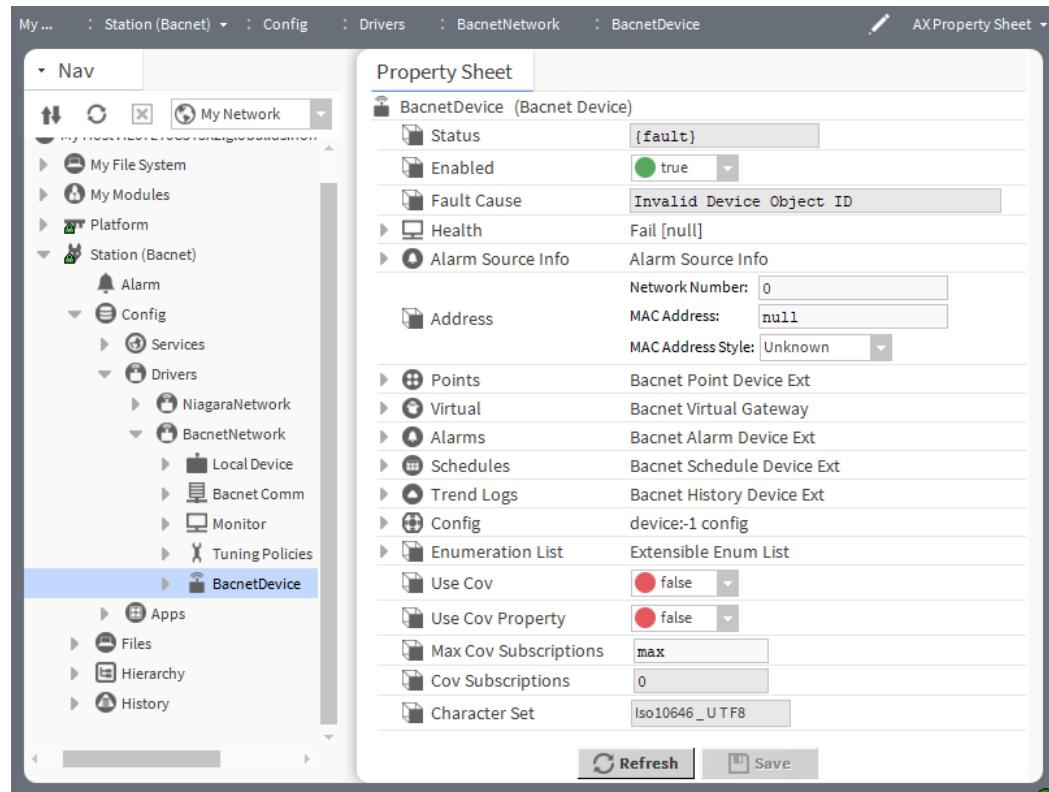
BacnetDeviceFolder

This component implements a folder under a BacnetNetwork.

Typically, you add such folders using the **New Folder** button in the **Bacnet Device Manager** view of the BacnetNetwork. Each BacnetDeviceFolder has its own **Bacnet Point Manager** view. The BacnetDeviceFolder is also available in the **bacnet** palette.

BacnetDevice (bacnet-BacnetDevice)

This component is a Framework representation of a remote BACnet device. Each BacnetDevice resides under the station's BacnetNetwork. Each BacnetDevice contains a full complement of device extensions (containers), including Points, Schedules, and Trend Logs (histories), for modeling data (from that device) in the station.

Figure 60 BacnetDevice properties

One way to access these properties is to expand **BacnetNetwork** and double-click **BacnetDevice**.

In addition to the common properties (Status, Enabled, Health and Fault Cause), this component has these properties.

| Property | Value | Description |
|-------------------|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Alarm Source Info | additional properties | Contains a set of properties for configuring and routing alarms when this component is the alarm source. For property descriptions, refer to the <i>Niagara Alarms Guide</i> . <i>The Niagara Alarms Guide</i> documents these properties. |
| Address | additional properties | Serves as a collective property name for Network Number , MAC Address , and MAC address style. |
| Points | additional properties | Opens the Bacnet Point Manager . |
| Virtual | additional properties | Provides access to the virtual components in the station's virtual component space, which is specific to the device. This is in addition to the standard collection of slots for device-level components. |
| Alarms | additional properties | Specifies how alarms from that device are mapped into the station's own alarm subsystem, plus provide status properties related to alarm sharing. Provides access to three alarm-related properties: Alarm Class , Last Received Time , and Niagara Process Id . |

| Property | Value | Description |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Schedules | additional properties | Opens the BACnet Schedule Import Manager . The Property Sheet provides access to the Retry Trigger component, Subscribe Window , and Skip Write On Error properties. |
| Trend Logs | additional properties | Opens the BACnet History Manager . You use this manager to import BACnet trend log objects from the device to the station as histories. The property sheet accesses to the Retry Trigger properties. |
| Enumeration List | list of facets | Configures a long list of BACnet facets. |
| Use Cov | true or false (If the device was discovered, and the station database determined that the device indicates support for server-side COV, this property defaults to true. Otherwise, it defaults to false indicating that no proxy points under the device use COV.) | Enables (true) and disables (false) a device's support for COV (change of value) as a way to monitor proxy point values. Assuming the device supports subscription to the COV service, true triggers the driver to attempt the necessary updates (proxy subscriptions) to the value of each point using the BACnet COV subscription to the device. If the subscription attempt succeeds, the Read Status property of the point's BacnetProxyExt displays COV . If the subscription attempt fails, the driver uses normal polling and the Read Status property shows Polled . When true, individual proxy points under the device may use COV subscriptions, depending on their assigned tuning policy. When false, the driver ignores any proxy subscription updates. |
| Use Cov Property | true or false (default) | Indicates if the system uses COV Property notification services (true) to receive data about points in this device for which COV Property is supported. |
| Max Cov Subscriptions | text (defaults to max) | Specifies the maximum number of COV (change of value) subscriptions that the database attempts to use with this device. This restriction prevents the station from consuming all of the available subscription space in the device. Applies only if Use Cov is true. |
| Cov Subscriptions | read-only | Reports the number of active COV client subscriptions to the device. |
| Character Set | drop-down list (defaults to Ansi C3_4) | Defines the character set supported, with other selections: IBM/Microsoft DBCS, JIS C 6226, ISO 10646 (UCS-4), ISO 10646 (UCS-2), ISO 8859-1, and Unknown. |

BacnetPointFolder (bacnet-BacnetPointFolder)

This component is the BACnet implementation of a folder under a **BacnetDevice→Points** container of the **BacnetPointDeviceExt**.

To add a point folder, use the **New Folder** button in the **Bacnet Point Manager** view of the **Points** component/container. Each **BacnetPointFolder** has its own **Bacnet Point Manager** view. The **BacnetPointFolder** is also available in the **bacnet** palette.

This component is also available in the palette under the **Points** folder.

Virtual (bacnet-BacnetVirtualGateway)

This component is the BACnet driver's implementation of the Baja Virtual Gateway. A virtual gateway resides under the station's component space. Other object spaces are **Files** and **History**. For a general explanation about Baja virtual components, refer to the *Niagara Drivers Guide*.

Each **BacnetDevice** and **BacnetWsDevice** has its own virtual gateway, at the same level as its device extensions (**Points**, **Schedules**, and so on).

Accessing components under this gateway dynamically adds them as virtual points while they are subscribed, but they exist only in memory (are not persisted in the station database like proxy points). When virtual points become unsubscribed, they driver automatically removes them from the station database.

Expanding a **BacnetVirtualGateway** results in a list of the device's BACnet objects. Each is a **BacnetVirtualObject** that contains a number of **BACnetVirtualProperty** components to represent its properties. Thus, virtual points are at the object, property level.

BacnetVirtualObject (bacnet-BacnetVirtualObject)

This component represents the driver implementation of a Baja virtual point, where each **BacnetVirtualObject** corresponds to a BACnet object. **BacnetVirtualObjects** can provide basic monitor access in Px views and be used for one-off read/write access to configuration properties.

Virtual objects reside in the station's virtual component space, and are not persisted in the station's database in its component space (Config), like regular components. You find them by expanding the **Property Sheet** of the **Virtual** node (**BacnetVirtualGateway**) under each **BacnetDevice** and **BacnetWsDevice**. The **Property Sheet** of each **BacnetVirtualObject** provides dynamic property values for the object.

For a general explanation about Baja virtual components, refer to the *Niagara Drivers Guide*.

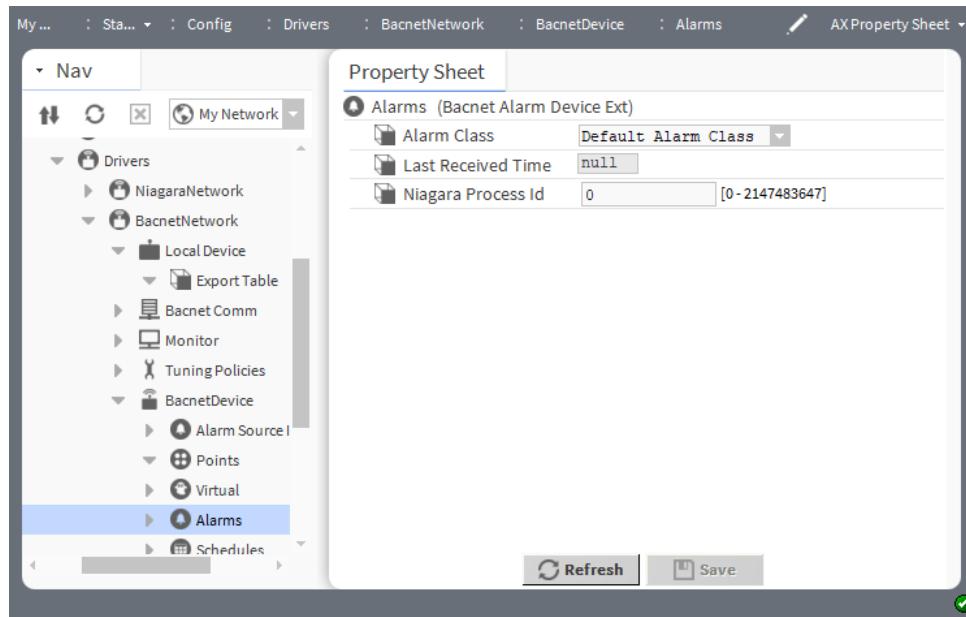
BacnetVirtualProperty (bacnet-BacnetVirtualProperty)

This component represents a property under a **BacnetVirtualObject**. Essentially, each property represents an available virtual point under that **BacnetDevice**, via its parent **BacnetVirtualObject** and its parent **BacnetVirtualGateway**.

Usage of BACnet virtual properties is anticipated for monitor access in Px views and for one-off read/write access to configuration properties.

Alarms (bacnet-BacnetAlarmDeviceExt)

Each **BacnetDevice** has an **Alarms** device extension. This extension applies to BACnet event notifications (alarms) sent to the station from a device.

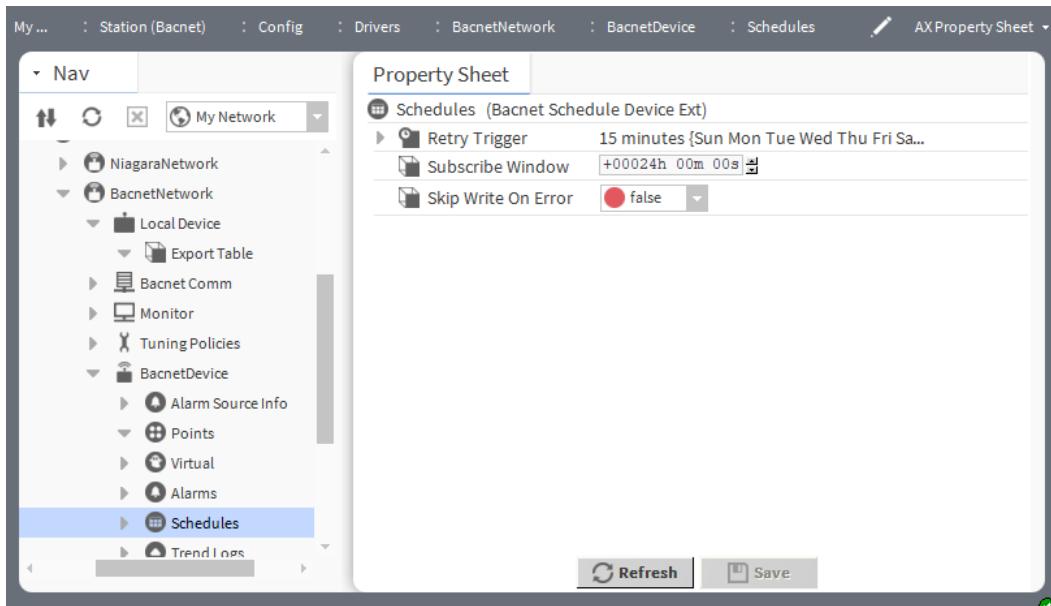
Figure 61 BacnetAlarmDeviceExt (Alarms) properties

One way to view these properties is to expand **BacnetNetwork**→**BacnetDevice** followed by double-clicking the **Alarms** node in the Nav tree.

| Property | Value | Description |
|--------------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Alarm Class | drop-down list | Defines alarm routing options and priorities. Typical alarm classes include High, Medium and Low. An alarm class of Low might send an email message, while an alarm class of High might trigger a text message to the department manager. |
| Last Received Time | read-only | Reports the date-timestamp of the last BACnet event notification received from the device. |
| Niagara Process Id | integer | Specifies a process ID, which needs to match the BACnet process identifier used to send notifications to this client station. The default value is 0. |

Schedules (bacnet-BacnetScheduleDeviceExt)

This device extension configures schedule properties.

Figure 62 Schedules properties

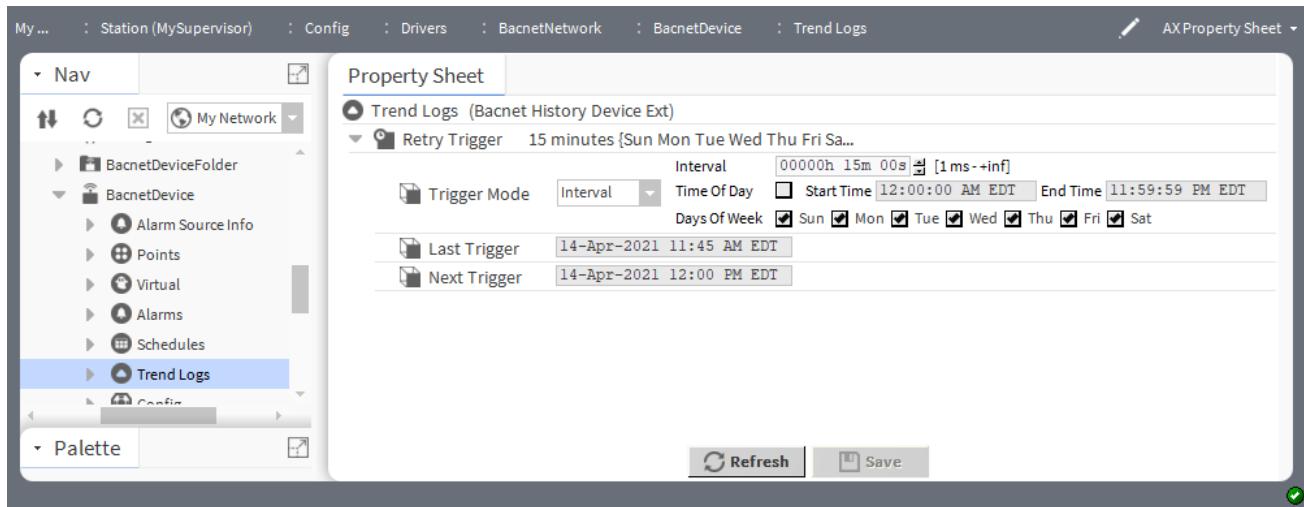
To access these properties, expand **Config**→**Drivers**→**BacnetNetwork**→**BacnetDevice**, right-click **Schedules** and click **Views**→**AX Property Sheet**.

| Property | Value | Description |
|---------------------|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Retry Trigger | additional properties | Defines how frequently to attempt a failed operation again. This continues until successful execution occurs. Appears in the Nav tree but not in any manager view and is unique in that it requires no linking of its output for operation. <i>Getting Started with Niagara</i> documents Retry Trigger properties. |
| Subscribe Window | hours minutes seconds | Configures a period of time. |
| Skip Write On Error | true or false (default) | Configures what happens when an error occurs. true skips a write when an error occurs. false always writes the record. |

Trend Logs (bacnet-BacnetHistoryDeviceExt)

This component is a frozen device extension under every **BacnetDevice**, and the container for **BacnetHistoryImport** components.

The default view is the **Bacnet History Import Manager**, used to import data from trend log objects in the device, into the station as histories.

Figure 63 Bacnet History Device Ext

This folder under the **BacnetNetwork**→**BacnetDevice** node in the Nav tree serves as the container for BACnet history objects, which you import from a device into a station.

A BACnet device must contain trend log objects (and/or trend log multiple objects) to make use of histories. A Discover command determines this. If it finds no such objects, the trend logs extension has no practical application.

By default, this container also has a **Retry Trigger**, for automatic usage as needed.

For general information on this device extension and the retry trigger, refer to the *Niagara Drivers Guide*.

Config container (bacnet-BacnetConfigDeviceExt)

This frozen device extension under every **BacnetDevice** functions as the container for **Config**-type objects, which represent individual BACnet objects in the device. By default, it contains a **BacnetDeviceObject** for the device's sole **BacnetDevice** object.

A separate topic documents the **BacnetDeviceObject**.

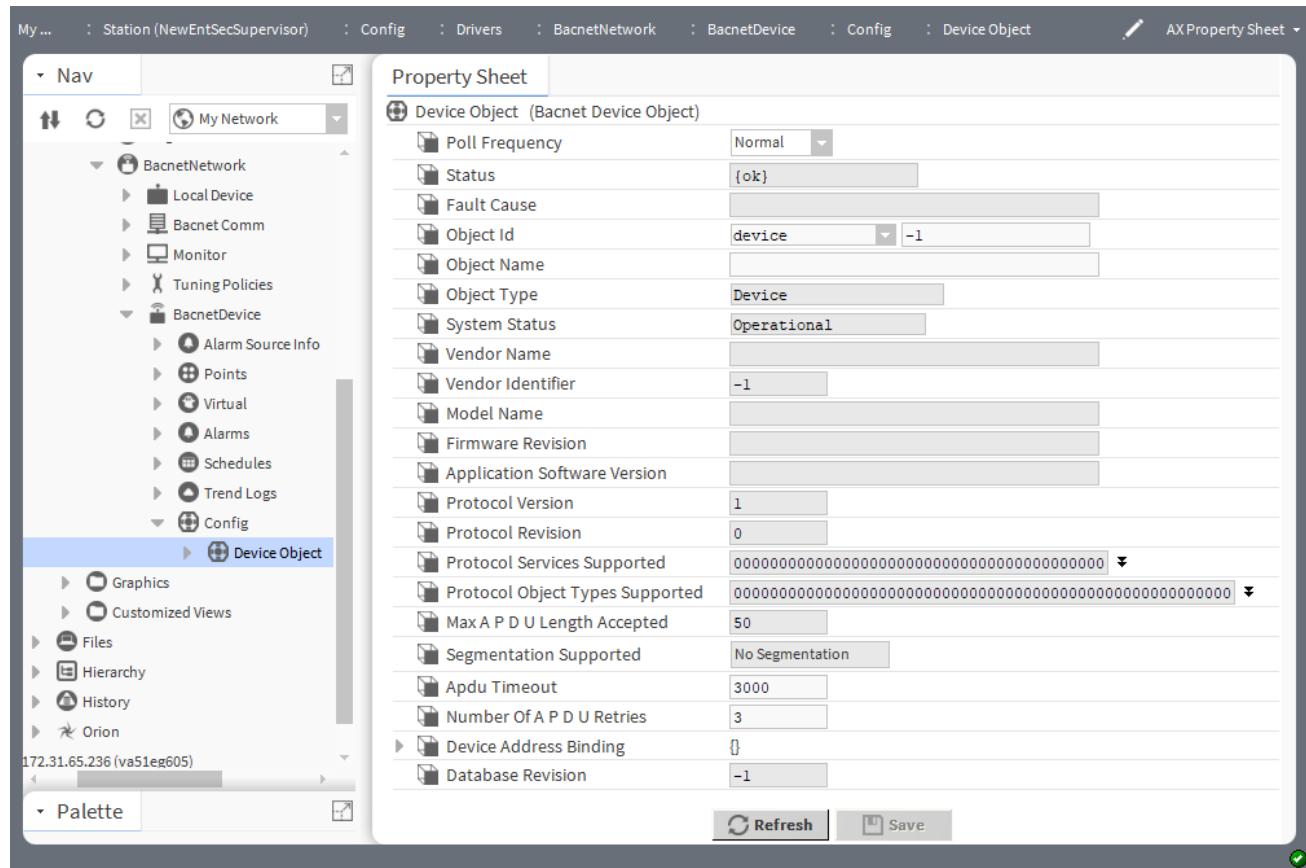
Device Object (bacnet-BacnetDeviceObject)

This component provides a view of a BACnet object in its native format—where all object properties are presented as a whole, as defined by the BACnet specification.

Unlike BACnet proxy points, which you create using the **Bacnet Point Manager**, device objects, when subscribed, poll all properties, however, not as efficiently as the selective proxy point model polls. Slots of BACnet device objects use primitive or special data types, and are not compatible with normal linking logic.

Config device objects are expected to be useful for one-time commissioning or for proprietary objects for which the proxy point interaction may be insufficient.

Figure 64 Device Object properties



To access an added or discovered device, expand the **Config**→**Drivers**→**BACnetNetwork**→**BacnetDevice**→**Config** and double-click **Device Object**.

In addition to the standard properties (Poll Frequency, Status and Fault Cause), these properties configure BACnet devices.

| Property | Value | Description |
|-------------|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Object Id | drop-down list and number (from 0 to 4194302) | <p>Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number.</p> <p>For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device.</p> |
| Object Name | Object Id and Instance Number | <p>Identifies a BACnet object. This name should be unique within this specific device.</p> <p>For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device.</p> <p>For a proxy point, this is the default name in the station.</p> <p>You can shorten or edit Object Name, however, each Object Name must be unique among all server descriptors under the Export Table.</p> |

| Property | Value | Description |
|---------------------------------|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Object Type | read-only | Reports the type of object. |
| System Status | read-only | Reports the current condition of the network. |
| Vendor Name | read-only | Reports the vendor number. |
| Vendor Identifier | read-only | Reports an identifier for the vendor's device. |
| Model Name | read-only | Reports the manufacturer's model name for the device. |
| Firmware Revision | read-only | Reports the version of the device firmware. |
| Application Software Version | read-only | Reports the version of the manufacturer's software running on the device. |
| Protocol Version | read-only | Displays the version number of the BACnet protocol. |
| Protocol Revision | read-only | Displays the current BACnet protocol level. |
| Protocol Services Supported | read-only | Expands to identify all the available BACnet device services supported by this version of Niagara. |
| Protocol Object Types Supported | read-only | Expands to identify all the available BACnet object types supported by this version of Niagara. |
| Max A P D U Length Accepted | read-only | Reports the number of points per poll or APDU (Application Protocol Data Units). |
| Segmentation Supported | read-only | Indicates the type of message segmentation the device supports. If a message exceeds the Max Segment Accepted length, the device must be able to break it into segments or part of the message is lost. |
| Apdu Timeout | milliseconds (defaults to 3000 ms) | Defines the time the system waits before retransmitting an APDU segment. This property is only relevant if devices are doing segmentation. Works for both client and server operations. For a small IP-based network, an optimal setting may be: 5000 ms. |
| Number of A P D U Retries | number (defaults to 3) | Defines the maximum number APDU retransmissions. Works for both client and server operations. For a small IP-based network, an optimal setting may be: 1 retry. |
| Device Address Binding | read-only (may be hidden) | Reflects the client-side operation entries, essentially a list of Device objects (by Object Id) and BACnet device address. These identify actual device parameters used when making client BACnet service requests to the devices. This list (bacnet-BacnetListOf) is documented in a separate topic. |
| Database Revision | read-only | Reports the version of the database. |

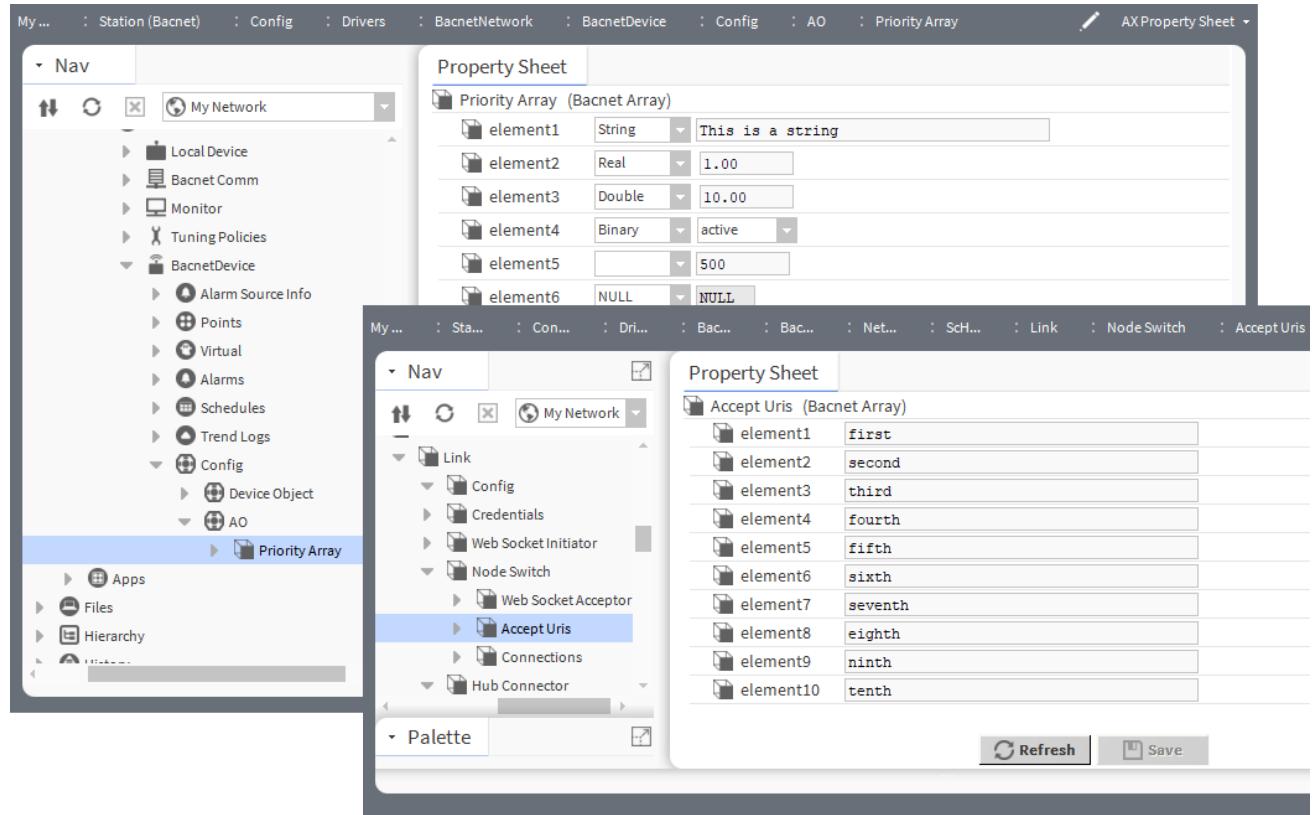
BACnet array (bacnet-BacnetArray)

This component represents an indexed sequence of objects of a particular BACnet data type. For example, **Accept Uris**, which appears under **BacnetNetwork→Bacnet Comm→Network→ScHubPort→Link→Node Switch** is a BACnet array.

This component appears as a **Priority Array** under an **AO** (Analog Output), **BO** (Binary Output) and **MSO** (MultiState Output) under the **Config** node of a **BacnetDevice**. Its elements are named element1 through elementN by default.

Under **Hub Function**, this component appears as **Accept Uri**.

Figure 65 Bacnet Array properties



To access this array under the **BacnetDevice**, expand **Config**→**Drivers**→**BacnetNetwork**→**BacnetDevice**→**Config**, expand **AO**, **BO** or **MSO** and double-click **Priority Array**.

To access this array under a **Hub Function**, expand **Config**→**Drivers**→**BacnetNetwork**→**Bacnet Comm**→**Network**, then expand a secure port (**ScPort** or **ScHubPort**), expand **Link**→**Node Switch** and double-click **Accept Uri**.

| Property | Value | Description |
|-----------------------------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| element1-n (Priority Array) | drop-down list | Identifies the type of value: Real, Double, Binary, Integer or String. |
| value | depends on the type | Enters a discrete value. |
| element1-n (Accept Uri) | text | Identifies the possible URIs that an initiating SC node can use to make a direct connection to this node. The values in this property are sent in the address resolution ack message. |

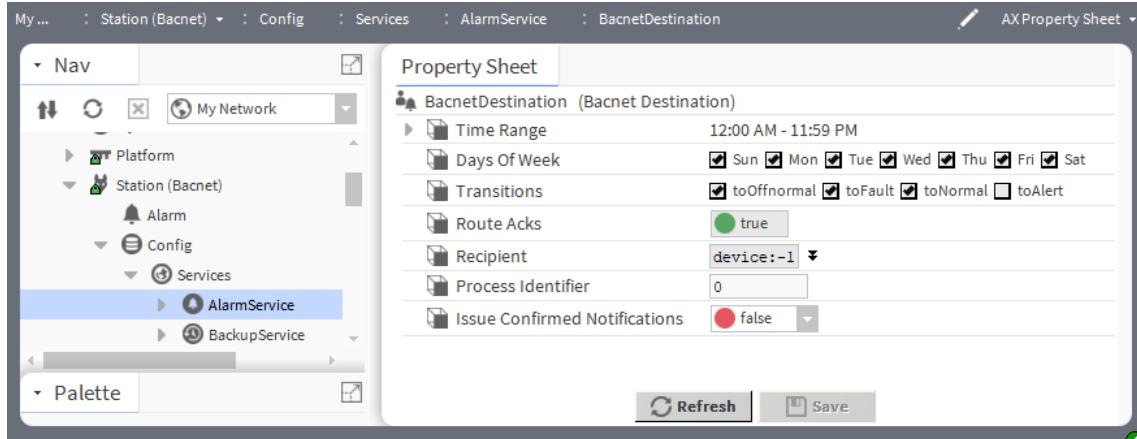
Actions for secure connect port Accept Uri array

- **addArrayElement** opens a window
- **removeArrayElement** deletes a selected element.

Bacnet Destination (bacnet-BacnetDestination)

This component represents a BACnet device that receives alarms from the Framework as a type of alarm recipient. It resides with the standard alarm recipients and alarm classes under the station's AlarmService.

Figure 66 BacnetDestination properties



One way to access these properties is to expand **Station→Config→Services→AlarmService** and double-click the **BacnetDestination** component. If your AlarmService does not have this component, open the **bacnet** palette and drag it from the **Alarming** folder to the **AlarmService** in the station.

Stations operating (and licensed) as a BACnet server use this alarm class recipient to route alarms. When linked as a recipient to an AlarmClass that is exported to BACnet (as a Notification Class object), each **BacnetDestination** specifies an entry under the **Recipient_List** for that Notification Class.

| Property | Value | Description |
|------------------------|--------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Time Range, Start Time | hour and time zone | Configures when to start the function. |
| Time Range, End Time | hour and time zone | Configures when to end the function. |
| Days of Week | check marks (all default to enabled) | Specifies the days of the week to include. |
| Transitions | check marks | Specifies which status transitions are sent, where the default is all but <code>toAlert</code> . |
| Route Acks | true (default) or false | Enables (true) and disables (false) the routing of alarm acknowledgements to the recipient. The framework does not route trap (event notification) acknowledgements if you select false. |
| Recipient | additional properties (defaults to device-1 which indicates that the destination is not defined) | Specifies one of two alarm event destinations. Device identifies a unique object Instance Number . Address provides an alternate address with separate properties. <i>Niagara Alarms Guide</i> |

| Property | Value | Description |
|------------------------------|-----------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Process Identifier | integer (defaults to 0) | Specifies a number in the Recipient List for the station. This integer from the BACnet Notification Class object of the sending device, must match this ID in the Alarms device extension of the receiving device. |
| Issue Confirmed Notification | true or false (defaults to false, which means that the system sends only unconfirmed event notifications) | Specifies if the station sends confirmed (true) or unconfirmed (false) event notifications. |

Points

This view lists the proxy points in the point device extension under the **NiagaraNetwork**. The Points folder is the top parent container for real-time data values originating from a device.

Points (bacnet-BacnetPointDeviceExt)

This container is the BACnet implementation of the **PointDeviceExt**, a frozen container under every **BacnetDevice**.

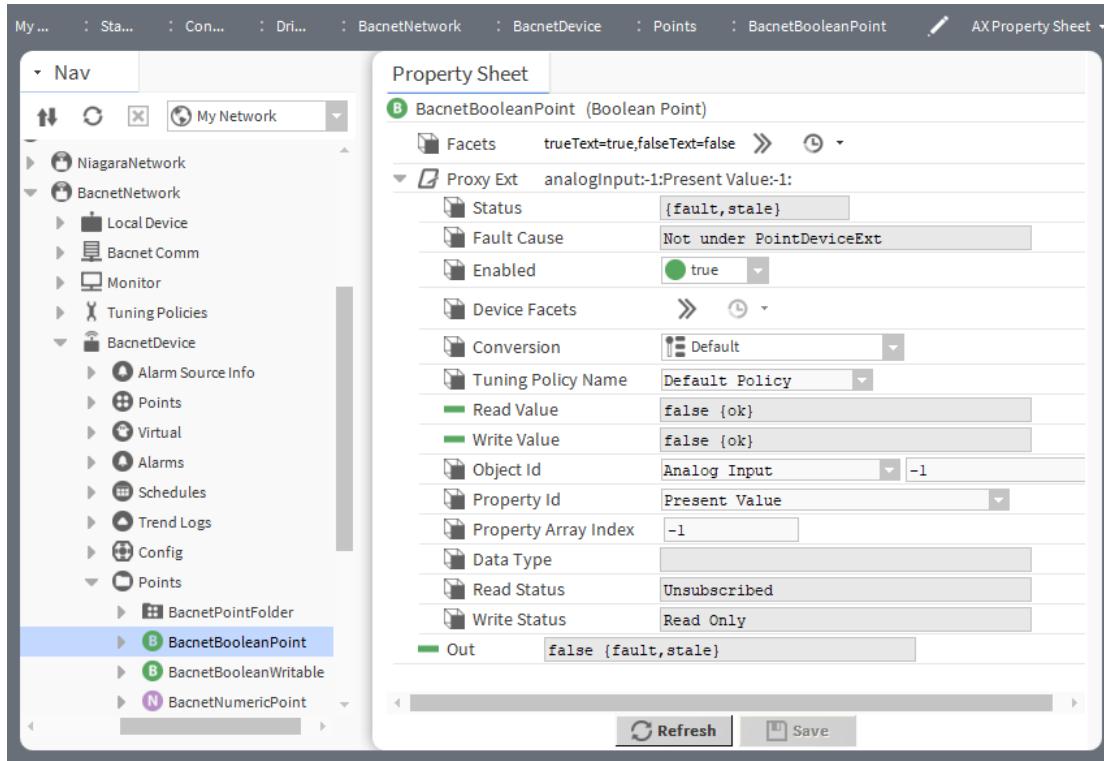
This component's primary view is the **Bacnet Point Manager**. For information about this view, refer to **Bacnet Point Manager** topic in this guide.

To access this container, expand **Config**→**Drivers**→**BacnetNetwork**→**BacnetDevice**, and double-click **Points**.

BACnet proxy points are similar to other driver proxy points. Refer to the *Niagara Drivers Guide* for general information.

Proxy Ext (bacnet-BacnetBooleanProxyExt)

This proxy extension maps a BACnet Boolean value to a BooleanPoint or a BooleanWritable.

Figure 67 Bacnet Boolean proxy point properties

This component is under the **Points** folder in the palette. To add it to a station, drag a **BacnetBooleanPoint** to the **BacnetNetwork**→**BacnetDevice**→**Points** folder. To view the properties, expand the **BacnetBooleanPoint** or double-click the **Proxy Ext** node in the Nav tree.

In addition to the common properties (Status, Enabled and Fault Cause), this component has these properties.

| Property | Value | Description |
|---------------|-----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Facets | Opens the Config Facets window | |
| Proxy Ext | Out, In, and Fall-back values null indicates that the point is an empty placeholder. | Indicate from where the point's value originates, including details specific to the parentage of the point's network and communications (driver). |
| Device Facets | additional properties | Determine how values are formatted for display depending on the context and the type of data. Examples include engineering units and decimal precision for numeric types, and descriptive value (state) text for boolean and enum types. With the exception of proxy points (with possible defined device facets), point facets do not affect how the framework processes the point's value. Besides control points, various other components have facets too. For example, many kitControl1 and schedule components have facets. Details about point facets apply to these components too, unless especially noted. |

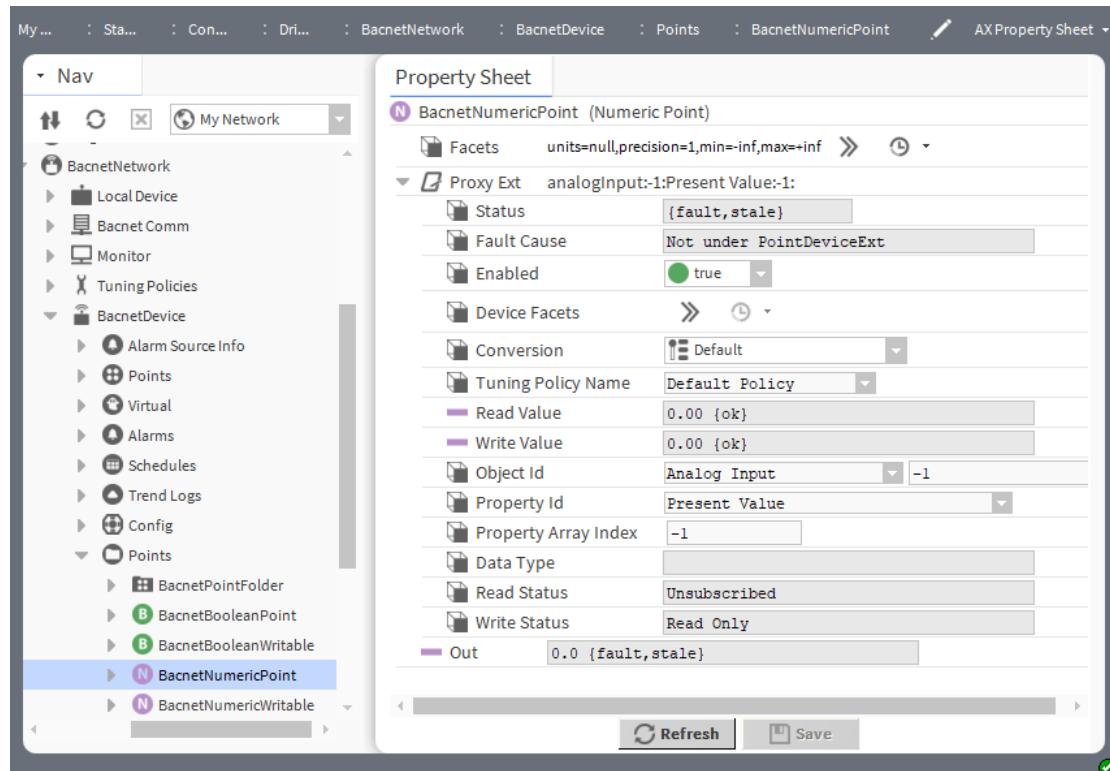
| Property | Value | Description |
|----------------------|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | You access facets by clicking an Edit button or a chevron >>. Both open an Edit Facets window. |
| Conversion | drop-down list (defaults to Default) | Configures the facets to use when converting values from device facets to point facets. |
| Tuning Policy Name | drop-down list (defaults to Default Policy) | Selects a network tuning policy by name. This policy defines stale time and minimum and maximum update times. During polling, the system uses the tuning policy to evaluate both write requests and the acceptability (freshness) of read requests. |
| Read Value | read-only | Displays the last value read from the device, expressed in device facets. |
| Write Value | read-only (applies to writable types only) | Displays the last value written using device facets. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Property Id | number (defaults to 85) | Defines the specific data item being proxied. Often, this is Present Value or Event State . This unique numeric value correlates to and identifies the data item. For example, the number 85 correlates to a present-value property. NOTE: Changing this ID on an existing BACnet proxy point can produce undesirable results, especially if the change assigns a different control point type. Instead, make a new proxy point. |
| Property Array Index | number | Identifies a particular element in an array of stored properties. For example the ObjectList is stored as an array of object identifiers. This value applies only to an arrayed property, like Priority Array. For example, if the point proxies priority level 7 of a priority array input for a binary output point, this value is 7. The Present Value property is not an array. For all properties that are not arrays, use an array index of -1. |
| Data Type | read-only | Reports the ASN (Abstract Syntax Notation) data type for a device property. The driver automatically converts values to this appropriate type. For example, an analog input present value is an ASN REAL that can be interpreted by a string point as a character string. ENUMERATED is another example. |
| Read Status | read-only | Indicates if the read status of the proxy extension is: Polled, COV, or Unsubscribed. This status depends on a number of factors. If a read error occurs, other descriptive text may appear instead. |

| Property | Value | Description |
|--------------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Write Status | read-only | <p>Indicates if the proxy extension is read-only or can be written to.</p> <p>read only indicates the proxy extension cannot be written to.</p> <p>writable or ok indicates that the proxy extension can be written to.</p> <p>For writable proxy extensions, this property indicates either that the last write occurred within the effective tuning policy period or, if a write operation failed, it provides descriptive text.</p> <p>For example, if you create a writable point for a read-only object (say, a NumericWritable for presentValue of an Analog_Input object) and attempt to write from the system, the write status may show: Property:Write Access Denied. If the error is actually a BACnet error, the colon-separated format contains the error class and error code returned by the device.</p> |
| Out | read-only | <p>Displays the current value of the proxy point including facets and status.</p> <p>The value depends on the type of control point.</p> <p>Facets define how the value displays, including the value's number of decimal places, engineering units, or text descriptors for Boolean/enum states. You can edit point facets to poll for additional properties, such as the native statusFlags and/or priorityArray level.</p> <p>Status reports the current health and validity of the value. Status is specified by a combination of status flags, such as fault, overridden, alarm, and so on. If no status flag is set, status is considered normal and reports {ok}.</p> |

Proxy Ext (bacnet-BacnetNumericProxyExt)

This proxy extension maps a BACnet numeric value to a NumericPoint or NumericWritable.

Figure 68 Bacnet numeric proxy point properties



This component is under the **Points** folder in the palette. To add it to a station, drag a **BacnetNumericPoint** to the **BacnetNetwork**→**BacnetDevice**→**Points** folder. To view the properties, expand the **BacnetNumericPoint** or double-click the **Proxy Ext** node in the Nav tree.

In addition to the common properties (Status, Enabled and Fault Cause), this component has these properties.

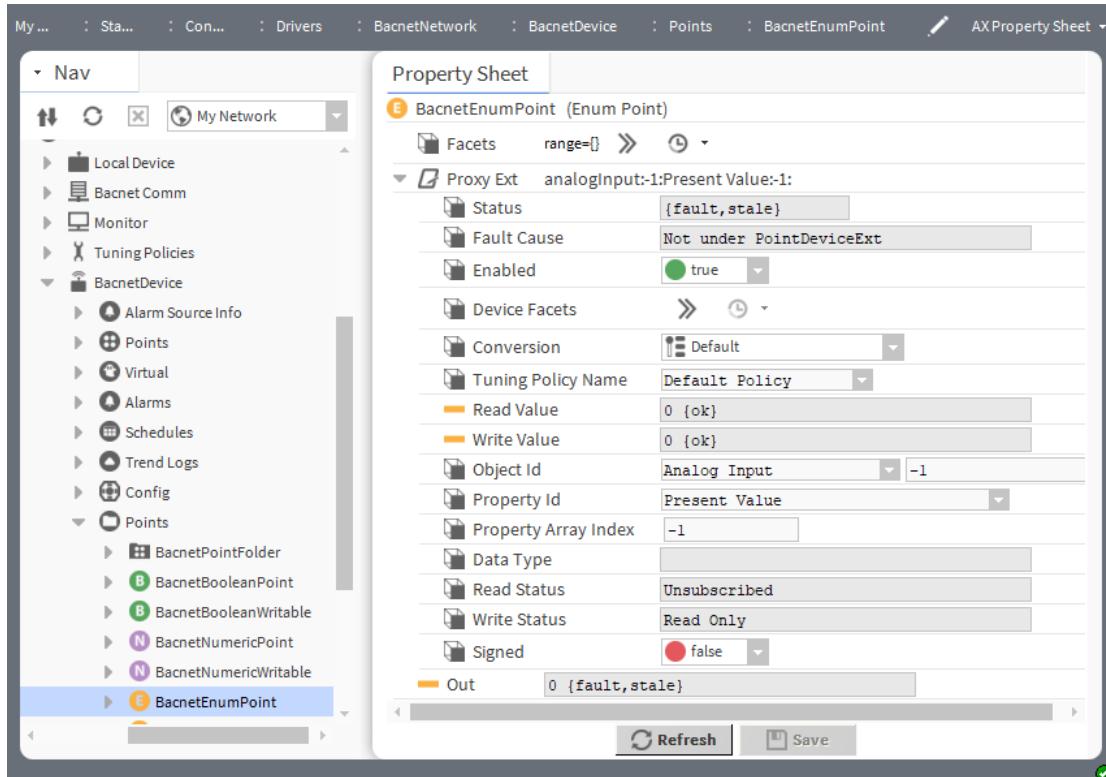
| Property | Value | Description |
|-----------|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Facets | Opens the Config Facets window | <p>Determine how values are formatted for display depending on the context and the type of data. Examples include engineering units and decimal precision for numeric types, and descriptive value (state) text for boolean and enum types.</p> <p>With the exception of proxy points (with possible defined device facets), point facets do not affect how the framework processes the point's value.</p> <p>Besides control points, various other components have facets too. For example, many kitControl and schedule components have facets. Details about point facets apply to these components too, unless especially noted.</p> <p>You access facets by clicking an Edit button or a chevron >>. Both open an Edit Facets window.</p> |
| Proxy Ext | Out, In, and Fall-back values | <p>Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number.</p> <p>For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For</p> |

| Property | Value | Description |
|----------------------|-----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Device Facets | additional properties | <p>Determine how values are formatted for display depending on the context and the type of data. Examples include engineering units and decimal precision for numeric types, and descriptive value (state) text for boolean and enum types.</p> <p>With the exception of proxy points (with possible defined device facets), point facets do not affect how the framework processes the point's value.</p> <p>Besides control points, various other components have facets too. For example, many <code>kitControl</code> and schedule components have facets. Details about point facets apply to these components too, unless especially noted.</p> <p>You access facets by clicking an Edit button or a chevron >>. Both open an Edit Facets window.</p> |
| Conversion | drop-down list (defaults to Default) | Configures the facets to use when converting values from device facets to point facets. |
| Tuning Policy Name | drop-down list (defaults to Default Policy) | <p>Selects a network tuning policy by name. This policy defines stale time and minimum and maximum update times.</p> <p>During polling, the system uses the tuning policy to evaluate both write requests and the acceptability (freshness) of read requests.</p> |
| Read Value | read-only | Displays the last value read from the device, expressed in device facets. |
| Write Value | read-only (applies to writable types only) | Displays the last value written using device facets. |
| Object Id | drop-down list and number (from 0 to 4194302) | <p>Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number.</p> <p>For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device.</p> |
| Property Id | number (defaults to 85) | <p>Defines the specific data item being proxied. Often, this is Present Value or Event State. This unique numeric value correlates to and identifies the data item. For example, the number 85 correlates to a present-value property.</p> <p>NOTE: Changing this ID on an existing BACnet proxy point can produce undesirable results, especially if the change assigns a different control point type. Instead, make a new proxy point.</p> |
| Property Array Index | number | Identifies a particular element in an array of stored properties. For example the <code>ObjectList</code> is stored as an array of object identifiers. |

| Property | Value | Description |
|--------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | <p>This value applies only to an arrayed property, like Priority Array. For example, if the point proxies priority level 7 of a priority array input for a binary output point, this value is 7.</p> <p>The Present Value property is not an array. For all properties that are not arrays, use an array index of -1.</p> |
| Data Type | read-only | Reports the ASN (Abstract Syntax Notation) data type for a device property. The driver automatically converts values to this appropriate type. For example, an analog input present value is an ASN REAL that can be interpreted by a string point as a character string. ENUMERATED is another example. |
| Read Status | read-only | Indicates if the read status of the proxy extension is: Polled, COV, or Unsubscribed. This status depends on a number of factors. If a read error occurs, other descriptive text may appear instead. |
| Write Status | read-only | <p>Indicates if the proxy extension is read-only or can be written to.</p> <p><code>read only</code> indicates the proxy extension cannot be written to.</p> <p><code>writable</code> or <code>ok</code> indicates that the proxy extension can be written to.</p> <p>For writable proxy extensions, this property indicates either that the last write occurred within the effective tuning policy period or, if a write operation failed, it provides descriptive text.</p> <p>For example, if you create a writable point for a read-only object (say, a NumericWritable for <code>presentValue</code> of an Analog_Input object) and attempt to write from the system, the write status may show: <code>Property:Write Access Denied</code>. If the error is actually a BACnet error, the colon-separated format contains the error class and error code returned by the device.</p> |
| Out | read-only | <p>Displays the current value of the proxy point including facets and status.</p> <p>The value depends on the type of control point.</p> <p>Facets define how the value displays, including the value's number of decimal places, engineering units, or text descriptors for Boolean/enum states. You can edit point facets to poll for additional properties, such as the native <code>statusFlags</code> and/or <code>priorityArray</code> level.</p> <p>Status reports the current health and validity of the value. Status is specified by a combination of status flags, such as <code>fault</code>, <code>overridden</code>, <code>alarm</code>, and so on. If no status flag is set, status is considered normal and reports <code>{ok}</code>.</p> |

Proxy Ext (bacnet-BacnetEnumProxyExt)

This component handles multistate values in a BACnet device, and maps enumerated values in the device to the parent `EnumPoint` or `EnumWritable`.

Figure 69 Bacnet enum proxy point properties

This component is under the **Points** folder in the palette. To add it to a station, drag a **BacnetEnumPoint** to the **BacnetNetwork**→**BacnetDevice**→**Points** folder. To view the properties, expand the **BacnetEnumPoint** or double-click the **Proxy Ext** node in the Nav tree.

In addition to the common properties (Status, Enabled and Fault Cause), this component has these properties.

| Property | Value | Description |
|-----------|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Facets | Opens the Config Facets window | <p>Determine how values are formatted for display depending on the context and the type of data. Examples include engineering units and decimal precision for numeric types, and descriptive value (state) text for boolean and enum types.</p> <p>With the exception of proxy points (with possible defined device facets), point facets do not affect how the framework processes the point's value.</p> <p>Besides control points, various other components have facets too. For example, many kitControl and schedule components have facets. Details about point facets apply to these components too, unless especially noted.</p> <p>You access facets by clicking an Edit button or a chevron >>. Both open an Edit Facets window.</p> |
| Proxy Ext | Out, In, and Fall-back values | <p>Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number.</p> <p>For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For</p> |

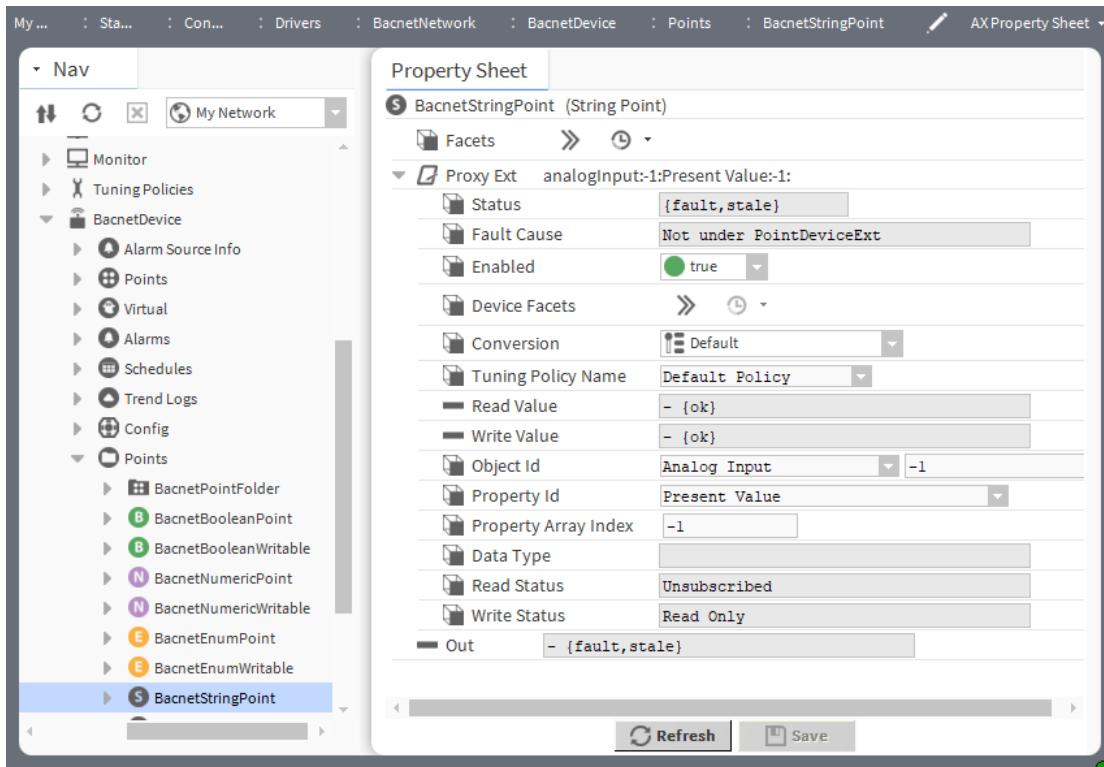
| Property | Value | Description |
|----------------------|-----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Device Facets | additional properties | <p>Determine how values are formatted for display depending on the context and the type of data. Examples include engineering units and decimal precision for numeric types, and descriptive value (state) text for boolean and enum types.</p> <p>With the exception of proxy points (with possible defined device facets), point facets do not affect how the framework processes the point's value.</p> <p>Besides control points, various other components have facets too. For example, many <code>kitControl</code> and schedule components have facets. Details about point facets apply to these components too, unless especially noted.</p> <p>You access facets by clicking an Edit button or a chevron >>. Both open an Edit Facets window.</p> |
| Conversion | drop-down list (defaults to Default) | Configures the facets to use when converting values from device facets to point facets. |
| Tuning Policy Name | drop-down list (defaults to Default Policy) | <p>Selects a network tuning policy by name. This policy defines stale time and minimum and maximum update times.</p> <p>During polling, the system uses the tuning policy to evaluate both write requests and the acceptability (freshness) of read requests.</p> |
| Read Value | read-only | Displays the last value read from the device, expressed in device facets. |
| Write Value | read-only (applies to writable types only) | Displays the last value written using device facets. |
| Object Id | drop-down list and number (from 0 to 4194302) | <p>Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number.</p> <p>For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device.</p> |
| Property Id | number (defaults to 85) | <p>Defines the specific data item being proxied. Often, this is Present Value or Event State. This unique numeric value correlates to and identifies the data item. For example, the number 85 correlates to a present-value property.</p> <p>NOTE: Changing this ID on an existing BACnet proxy point can produce undesirable results, especially if the change assigns a different control point type. Instead, make a new proxy point.</p> |
| Property Array Index | number | Identifies a particular element in an array of stored properties. For example the <code>ObjectList</code> is stored as an array of object identifiers. |

| Property | Value | Description |
|--------------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | <p>This value applies only to an arrayed property, like Priority Array. For example, if the point proxies priority level 7 of a priority array input for a binary output point, this value is 7.</p> <p>The Present Value property is not an array. For all properties that are not arrays, use an array index of -1.</p> |
| Data Type | read-only | Reports the ASN (Abstract Syntax Notation) data type for a device property. The driver automatically converts values to this appropriate type. For example, an analog input present value is an ASN REAL that can be interpreted by a string point as a character string. ENUMERATED is another example. |
| Read Status | read-only | Indicates if the read status of the proxy extension is: Polled, COV, or Unsubscribed. This status depends on a number of factors. If a read error occurs, other descriptive text may appear instead. |
| Write Status | read-only | <p>Indicates if the proxy extension is read-only or can be written to.</p> <p>read only indicates the proxy extension cannot be written to.</p> <p>writable or ok indicates that the proxy extension can be written to.</p> <p>For writable proxy extensions, this property indicates either that the last write occurred within the effective tuning policy period or, if a write operation failed, it provides descriptive text.</p> <p>For example, if you create a writable point for a read-only object (say, a NumericWritable for presentValue of an Analog_Input object) and attempt to write from the system, the write status may show: Property: Write Access Denied. If the error is actually a BACnet error, the colon-separated format contains the error class and error code returned by the device.</p> |
| Signed | true or false (default) | |
| Out | read-only | <p>Displays the current value of the proxy point including facets and status.</p> <p>The value depends on the type of control point.</p> <p>Facets define how the value displays, including the value's number of decimal places, engineering units, or text descriptors for Boolean/enum states. You can edit point facets to poll for additional properties, such as the native statusFlags and/or priorityArray level.</p> <p>Status reports the current health and validity of the value. Status is specified by a combination of status flags, such as fault, overridden, alarm, and so on. If no status flag is set, status is considered normal and reports {ok}.</p> |

Proxy Ext (bacnet-BacnetStringProxyExt)

This component handles the point configuration of a point of generic type in a BACnet device. It is the default for types NULL, OCTET_STRING, CHARACTER_STRING, BIT_STRING, DATE, TIME, and OBJECT_IDENTIFIER.

Figure 70 Bacnet string proxy point properties



This component is under the **Points** folder in the palette. To add it to a station, drag a **BacnetStringPoint** to the **BacnetNetwork→BacnetDevice→Points** folder. To view the properties, expand the **BacnetStringPoint** or double-click the **Proxy Ext** node in the Nav tree.

In addition to the common properties (Status, Enabled and Fault Cause), this component has these properties.

| Property | Value | Description |
|--------------------|---------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Facets | Opens the Config Facets window | <p>Determine how values are formatted for display depending on the context and the type of data. Examples include engineering units and decimal precision for numeric types, and descriptive value (state) text for boolean and enum types.</p> <p>With the exception of proxy points (with possible defined device facets), point facets do not affect how the framework processes the point's value.</p> <p>Besides control points, various other components have facets too. For example, many <code>kitControl</code> and schedule components have facets. Details about point facets apply to these components too, unless especially noted.</p> <p>You access facets by clicking an Edit button or a chevron >>. Both open an Edit Facets window.</p> |
| Proxy Ext | Out, In, and Fall-back values | <p>Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number.</p> <p>For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device.</p> |
| Device Facets | additional properties | <p>Determine how values are formatted for display depending on the context and the type of data. Examples include engineering units and decimal precision for numeric types, and descriptive value (state) text for boolean and enum types.</p> <p>With the exception of proxy points (with possible defined device facets), point facets do not affect how the framework processes the point's value.</p> <p>Besides control points, various other components have facets too. For example, many <code>kitControl</code> and schedule components have facets. Details about point facets apply to these components too, unless especially noted.</p> <p>You access facets by clicking an Edit button or a chevron >>. Both open an Edit Facets window.</p> |
| Conversion | drop-down list (defaults to Default) | Configures the facets to use when converting values from device facets to point facets. |
| Tuning Policy Name | drop-down list (defaults to Default Policy) | <p>Selects a network tuning policy by name. This policy defines stale time and minimum and maximum update times.</p> <p>During polling, the system uses the tuning policy to evaluate both write requests and the acceptability (freshness) of read requests.</p> |
| Read Value | read-only | Displays the last value read from the device, expressed in device facets. |
| Write Value | read-only (applies to writable types only) | Displays the last value written using device facets. |

| Property | Value | Description |
|----------------------|-----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Object Id | drop-down list and number (from 0 to 4194302) | <p>Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number. For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device.</p> |
| Property Id | number (defaults to 85) | <p>Specifies a number in the Recipient List for the station. This integer from the BACnet Notification Class object of the sending device, must match this ID in the Alarms device extension of the receiving device.</p> |
| Property Array Index | number | <p>Identifies a particular element in an array of stored properties. For example the ObjectList is stored as an array of object identifiers.</p> <p>This value applies only to an arrayed property, like Priority Array. For example, if the point proxies priority level 7 of a priority array input for a binary output point, this value is 7.</p> <p>The Present Value property is not an array. For all properties that are not arrays, use an array index of -1.</p> |
| Data Type | read-only | <p>Reports the ASN (Abstract Syntax Notation) data type for a device property. The driver automatically converts values to this appropriate type. For example, an analog input present value is an ASN REAL that can be interpreted by a string point as a character string. ENUMERATED is another example.</p> |
| Read Status | read-only | <p>Indicates if the read status of the proxy extension is: Polled, COV, or Unsubscribed. This status depends on a number of factors. If a read error occurs, other descriptive text may appear instead.</p> |

| Property | Value | Description |
|--------------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Write Status | read-only | <p>Indicates if the proxy extension is read-only or can be written to.</p> <p>read only indicates the proxy extension cannot be written to.</p> <p>writable or ok indicates that the proxy extension can be written to.</p> <p>For writable proxy extensions, this property indicates either that the last write occurred within the effective tuning policy period or, if a write operation failed, it provides descriptive text.</p> <p>For example, if you create a writable point for a read-only object (say, a NumericWritable for presentValue of an Analog_Input object) and attempt to write from the system, the write status may show: Property:Write Access Denied. If the error is actually a BACnet error, the colon-separated format contains the error class and error code returned by the device.</p> |
| Out | read-only | <p>Displays the current value of the proxy point including facets and status.</p> <p>The value depends on the type of control point.</p> <p>Facets define how the value displays, including the value's number of decimal places, engineering units, or text descriptors for Boolean/enum states. You can edit point facets to poll for additional properties, such as the native statusFlags and/or priorityArray level.</p> <p>Status reports the current health and validity of the value. Status is specified by a combination of status flags, such as fault, overridden, alarm, and so on. If no status flag is set, status is considered normal and reports {ok}.</p> |

Out Of Service Extension (bacnet-OutOfServiceExt)

The system automatically adds this component to each object exported to BACnet.

Figure 71 out OfService Ext properties



You access this extension on a control point that has been exported to BACnet. This component has two properties.

| Property | Value | Description |
|----------------|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Out of Service | true (default) or false | Affects only the BACnet exposure and access to the source component. If the source (exported) component is a proxy point, the equivalent Enabled property in its ProxyExt is not affected. true writes the point's current value to the Present Value property. Often, a remote BACnet client sets (or clears) this Out of Service state. You can set this property to false if needed. |
| Present Value | various | While Out Of Service is set to true , this property reflects the last written Present Value . |

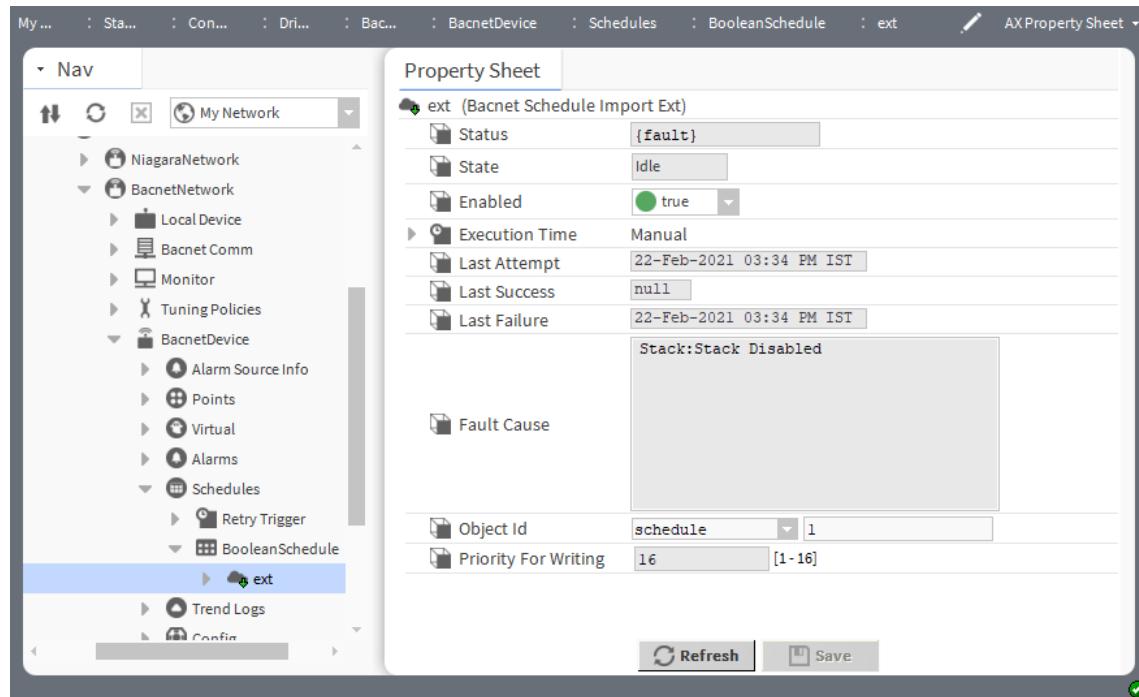
Schedule

Schedule include services, folders and other model building blocks associated with a module. You drag them to a property or wire sheet from a palette.

Bacnet Schedule Import Ext (bacnet-BacnetScheduleImportExt)

This component is a child extension of a schedule that is being imported from a BACnet device. The schedule in the remote BACnet device is the master. The driver periodically synchronizes its local copy by reading the appropriate values. You add BacnetScheduleImportExts using the **Bacnet Schedule Import Manager** view.

Figure 72 Bacnet Schedule Import Ext properties



This extension imports one or more remote Schedule and/or Calendar objects into a station as a read-only schedule component. You use these BACnet schedules or calendars in the station.

The BACnet property **Priority for Writing** is included, however, it means little in the station usage of this imported schedule.

In addition to the common properties (Status, Enabled and Fault Cause), this component has these properties.

| Property | Value | Description |
|----------------------|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| State | read-only | Indicates the condition of the BACnet schedule. |
| Last Attempt | read-only | Reports the date and time of the last attempted execution. |
| Last Success | read-only | Reports the last time the station successfully performed this function. |
| Last Failure | read-only timestamp | Reports the last time the system failed to perform this function. Refer to Fault Cause for details. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Priority for Writing | numeric text values 1 through 16 | Reports schedule priority, which defaults to 16. |

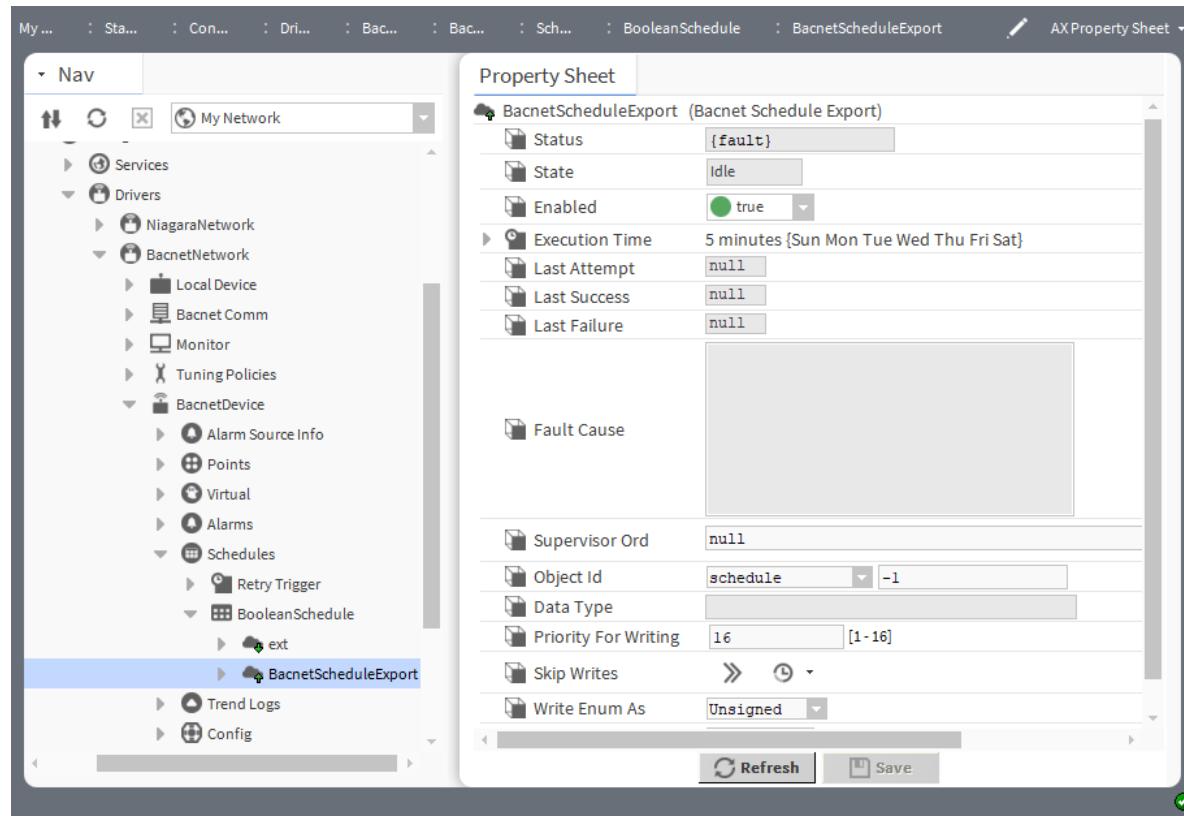
Execution Time

This extension has four control-time trigger properties: **Execution Time** (Interval, Manual, Daily), **Trigger Mode**, **Last Trigger**, and **Next Trigger**. These properties are documented in the *Niagara Drivers Guide*.

Bacnet Schedule Export (bacnet-BacnetScheduleExport)

This component exports the configuration of a schedule or calendar into a specific Schedule or Calendar object in a BACnet device. The schedule in the remote BACnet device functions as the subordinate schedule. The driver periodically synchronizes the values by writing its local values to the remote device using Write-Property service requests. You add BacnetScheduleExports using the **Bacnet Schedule Export Manager** view.

Figure 73 Bacnet Schedule Export properties



You can export any schedule component (residing anywhere in the station), into a device's existing BACnet Schedule or Calendar object. In this case, the schedule has supervisory control over the remote BACnet Schedule object, essentially swapping in its configuration.

In addition to the common properties (Status, Enabled and Fault Cause), this component has these properties.

| Property | Value | Description |
|----------------|-----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| State | read-only | Indicates the condition of the BACnet schedule. |
| Last Attempt | read-only | Reports the date and time of the last attempted execution. |
| Last Success | read-only | Reports the last time the station successfully performed this function. |
| Last Failure | read-only timestamp | Reports the last time the system failed to perform this function. Refer to Fault Cause for details. |
| Supervisor Ord | additional properties | Identifies the local station ORD for the schedule. |
| Object Id | drop-down list and number (from 0 to 4194302) | <p>Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number.</p> <p>For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device.</p> |

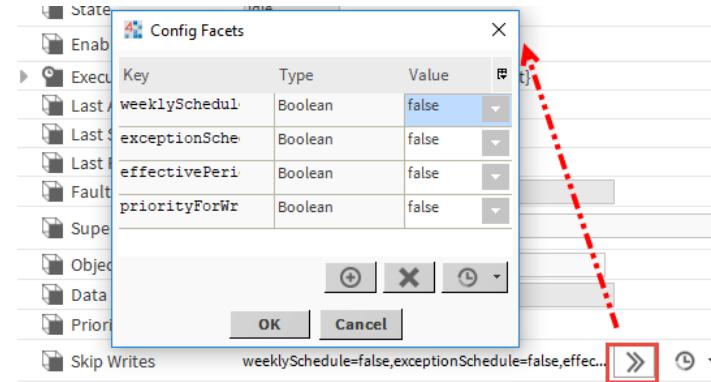
| Property | Value | Description |
|----------------------|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Data Type | read-only | Reports the ASN (Abstract Syntax Notation) data type for a device property. The driver automatically converts values to this appropriate type. For example, an analog input present value is an ASN REAL that can be interpreted by a string point as a character string. ENUMERATED is another example. |
| Priority for Writing | numeric text values 1 through 16 | Reports schedule priority, which defaults to 16. |
| Skip Writes | read-only | Adjusts which property types in a target BACnet schedule object the system writes to upon an export from a weekly schedule component. |
| Write Enum As | drop-down list | Defines how to write Enum data as Unsigned: ENUMERATED or as an INTEGER. |

About schedule export Skip Writes

Depending on the BACnet device's implementation by vendor, some properties of its schedule objects may be read-only. For example, a BACnet schedule object may allow writes to its weekly schedule events, but not to its exception schedule (special events), if they are read-only (or perhaps do not even exist). Or, the object's **Priority For Writing** property may be read-only, or the object may not even have a weekly schedule or exception schedule (a schedule must only have one or the other, it may have both).

To allow for this, a **BacnetScheduleExport** component provides a **Skip Writes** property in which you can specify the properties to be written upon an export from the software. From the property sheet, click the far-right side control for a popup **Facets Editor**.

Figure 74 Skip Writes property of BacnetScheduleExport component



As shown above, the default is to not skip writing any properties upon export (`false` for all), meaning that all property areas of the source schedule component are written to the target BACnet Schedule object. As needed, set any of these in the **Facets Editor** to `true`, such that the software does not attempt to write to them. This can allow an export without a fault.

NOTE: Skip Writes does not apply if exporting to a BACnet Calendar (from a **CalendarSchedule**).

The Skip Write property areas (facets) are:

- **weeklySchedule** — Corresponds to the regular day-of-week events of the source schedule, as defined in its **Weekly Scheduler** view.
- **exceptionSchedule** — Corresponds to all special events of the source schedule.
- **effectivePeriod** — As defined by the **Effective Period** in the source schedule.

- **priorityForWriting** — As defined by the Priority For Writing property in the **BacnetScheduleExport** component itself (instead of the source schedule).

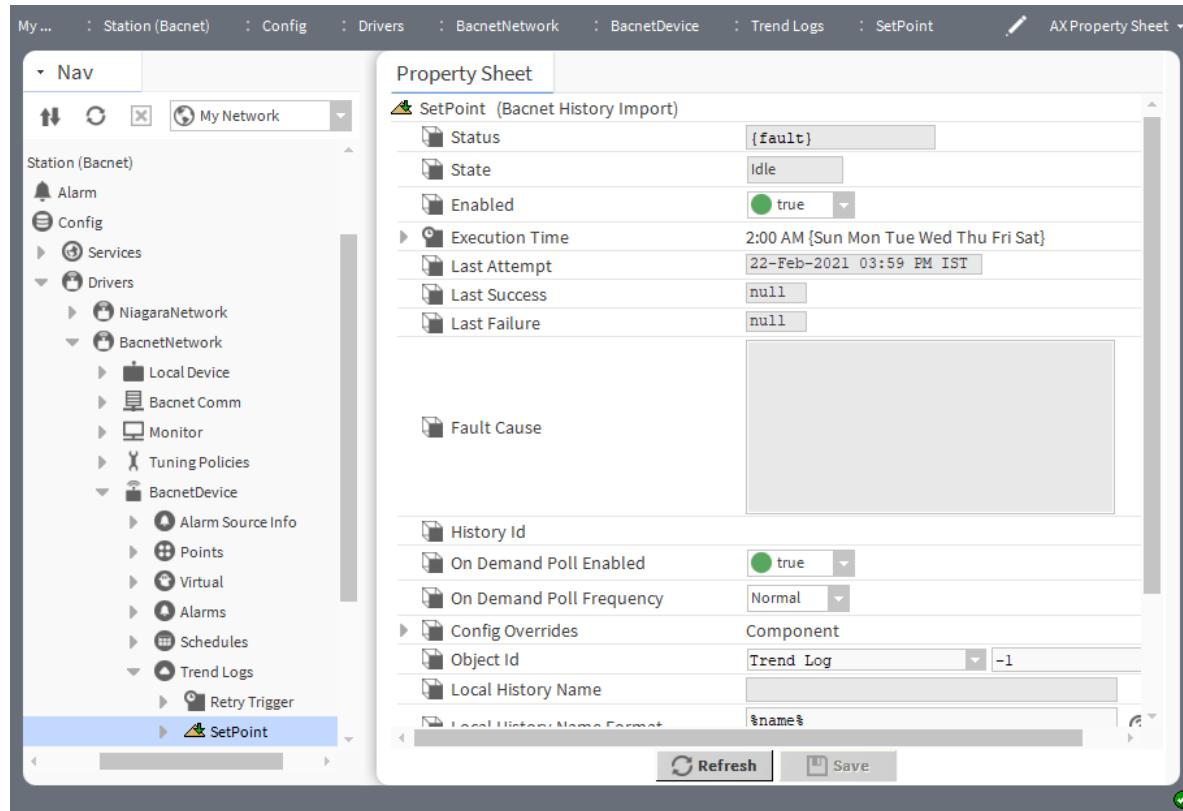
Trending

This extension creates a history for the StatusNumeric out value of a component using COV.

Bacnet History Import (bacnet-BacnetHistoryImport)

This component configures the retrieval of data from the station database for the purpose of archiving data in a trend log (history).

Figure 75 Bacnet History Import properties



To access this property sheet, double-click the **BacnetHistoryImport** component in the Nav side bar.

You add **BacnetHistoryImports** (import trend logs) using the **Bacnet History Import Manager** view of the parent **BacnetHistoryDeviceExt**.

In addition to the common properties (Status, Enabled and Fault Cause), this component has these properties.

| Property | Value | Description |
|----------------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| State | read-only | Indicates the condition of the BACnet schedule. |
| Execution time | multiple properties | <p>Configures the frequency with which data are imported from a remote trend log (or trend log multiple) into a history (or histories). Data import may occur at regular intervals (Interval), Daily or when manually requested (Manual).</p> <p>Time Of Day specifies a specific daily time.</p> |

| Property | Value | Description |
|------------------------------|-------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | Randomization configures an import based on time. Days Of Week specifies on which days to apply the import time. |
| Last Attempt | read-only | Reports the date and time of the last attempted execution. |
| Last Success | read-only | Reports the last time the station successfully performed this function. |
| Last Failure | read-only timestamp | Reports the last time the system failed to perform this function. Refer to Fault Cause for details. |
| History Id | read-only | Displays a combination of the station name and history name. |
| On Demand Poll Enabled | true (default) or false | Determines user control over polling. true enables a system user to use the Live Updates (play) button in History views to poll for live data for the associated imported history(ies). false renders this button unavailable in history views for the associated imported history(ies). |
| On Demand Poll Frequency | drop-down list | References the On Demand Poll Scheduler rates under the NiagaraNetwork's History Policies container slot. Configures how often this type of poll occurs. |
| Config Overrides, capacity | drop-down list (defaults to Unlimited) | Defines the maximum number of history records allowed in the associated table. Unlimited enforces no limitation on the number of records. Record Count opens an additional property for defining the table limit. |
| Config Overrides, fullPolicy | drop-down list (defaults to Roll) | Defines what happens if Capacity is set to Record Count and the specified record count is reached. Roll overwrites the oldest records with the newest ones. This ensures that the latest data are recorded. Stop terminates recording when the number of stored records reaches the specified capacity. Full policy has no effect if Capacity is Unlimited . |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Local History Name | read-only | Reports the names of local trend logs. |
| Local History Name Format | text, BFormat (Baja Format) syntax (defaults to %name%, which recreates | Defines the format string script for returning a local history. This is the name of the history initially created and populated by the trend log data and includes, typically, the name of the BACnet device component and the BACnet trend log object. |

| Property | Value | Description |
|----------------------------------|--------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | the BACnet name of the trend log object) or a combination of text and BFormat. | <p>For example: %parent.parent.displayName% %name% could define the name of a VAV fan: J7Bnet BldgA_VAV1_Fan.</p> <p>You can modify the default BFormat with other BFormat scripting. You could enter static text for the history name—or, some combination of static text and valid BFormat scripting text.</p> <p>In the case of trend log multiples (BacnetTrendLogMultipleImport), the default %name% value results in histories named using the BACnet name of the trend log multiple object, with underscore delimiters between the BACnet names of each _object_property record.</p> |
| Reference Time | hours, minutes, seconds | Defines the timestamp the system uses to include records in the trend log. |
| Max Records Per Request | 0 (default) to max | Defines the maximum number of records that can be archived in this trend log. The Framework asks for 10 records per request. |
| Always Request By Reference Time | true or false (default) | Configures if the driver should always use the Reference Time to request records. |
| Last Sequence Number Processed | number | Defines a maximum record number to process. |

Bacnet Trend Log Multiple Import (bacnet-BacnetTrendLogMultipleImport)

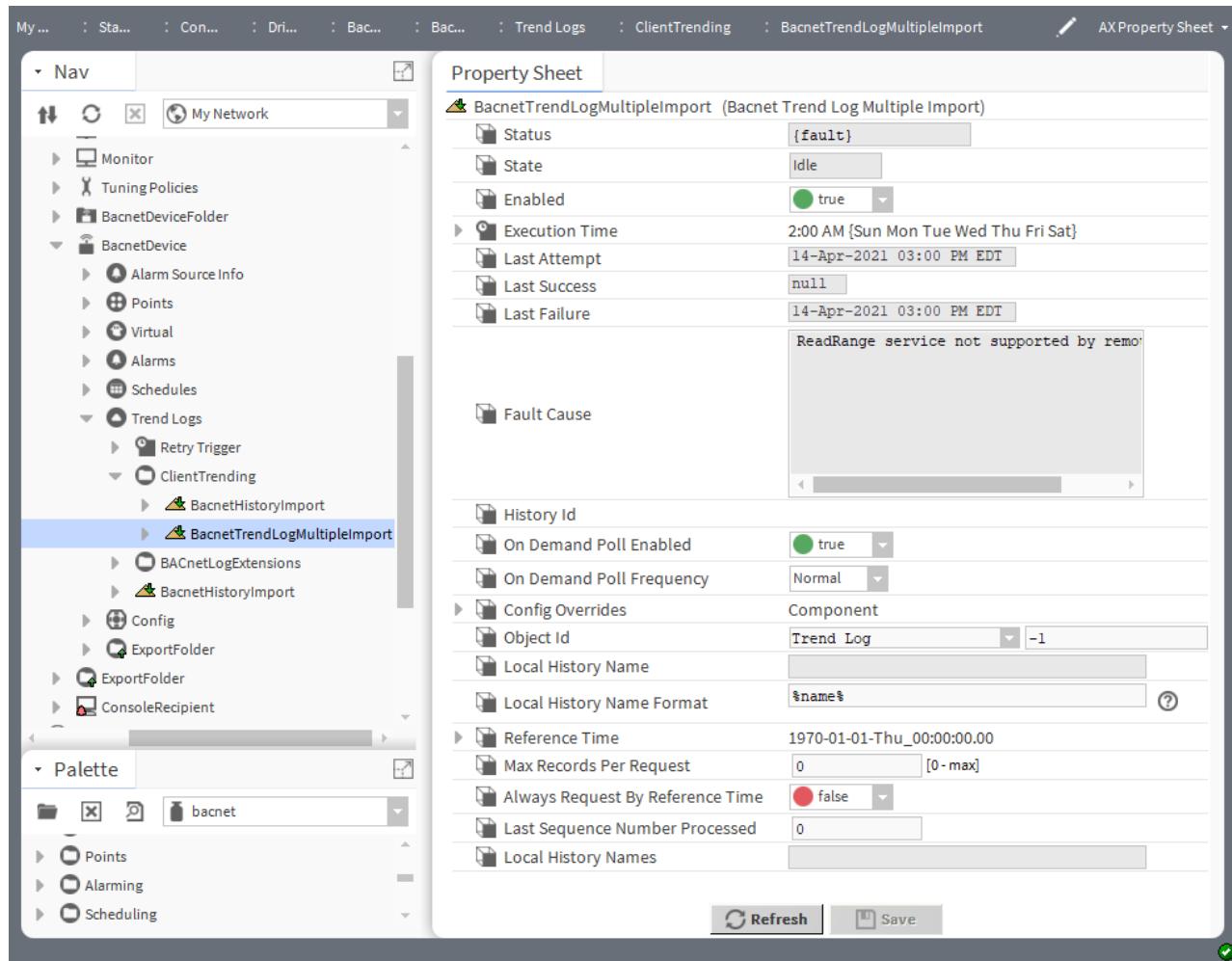
This component defines the archive action to retrieve data from a BACnet trend log multiple object into a history created by the addition of this component.

This component monitors and records values for one or more properties of one or more referenced objects, either in the same device or in an external device. A key feature is that the extension records all values using the same timestamp. One import descriptor typically results in multiple histories in a station—one history per object/property value in the record.

You add **BacnetTrendLogMultipleImports** (import Trend Log Multiples) using the **Bacnet History Import Manager** view of the device's parent trend logs extension. This import descriptor also has its own special **Bacnet Trend Multiple View**.

You can also copy a **ClientTrending** component from the **bacnet** palette. This component contains a **BacnetTrendLogMultipleImport** component.

A BACnet device must contain trend log objects (and/or trend log multiple objects) to make use of histories. A discover command determines this. If it finds no such objects, the **Trend Logs** extension has no practical application.

Figure 76 Trend Log Multiple Import properties

To access these properties, expand **Config**→**Drivers**→**BacnetNetwork**→**BacnetDevice**→**Trend Log-s**→**ClientTrending** and double-click a trend log descriptor.

In addition to the common properties (Status, Enabled and Fault Cause), this component has these properties.

| Property | Value | Description |
|----------------|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| State | read-only | Indicates the condition of the BACnet schedule. |
| Execution time | multiple properties | <p>Configures the frequency with which data are imported from a remote trend log (or trend log multiple) into a history (or histories). Data import may occur at regular intervals (Interval), Daily or when manually requested (Manual).</p> <p>Time Of Day specifies a specific daily time.</p> <p>Randomization configures an import based on time.</p> <p>Days Of Week specifies on which days to apply the import time.</p> |
| Last Attempt | read-only | Reports the date and time of the last attempted execution. |

| Property | Value | Description |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Last Success | read-only | Reports the last time the station successfully performed this function. |
| Last Failure | read-only timestamp | Reports the last time the system failed to perform this function. Refer to Fault Cause for details. |
| History Id | read-only | Displays a combination of the station name and history name. |
| On Demand Poll Enabled | true (default) or false | Determines user control over polling. true enables a system user to use the Live Updates (play) button in History views to poll for live data for the associated imported history(ies). false renders this button unavailable in history views for the associated imported history(ies). |
| On Demand Poll Frequency | drop-down list | References the On Demand Poll Scheduler rates under the NiagaraNetwork's History Policies container slot. Configures how often this type of poll occurs. |
| Config Overrides | additional properties | Configures the facets to use when converting values from device facets to point facets. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Local History Name | read-only | Reports the names of local trend logs. |
| Local History Name Format | text, BFormat (Baja Format) syntax (defaults to %name%, which recreates the BACnet name of the trend log object) or a combination of text and BFormat. | Defines the format string script for returning a local history. This is the name of the history initially created and populated by the trend log data and includes, typically, the name of the BACnet device component and the BACnet trend log object. For example: %parent.parent.displayName% %name% could define the name of a VAV fan: J7Bnet BldgA_VAV1_Fan. You can modify the default BFormat with other BFormat scripting. You could enter static text for the history name—or, some combination of static text and valid BFormat scripting text. In the case of trend log multiples (BacnetTrendLogMultipleImport), the default %name% value results in histories named using the BACnet name of the trend log multiple object, with underscore delimiters between the BACnet names of each _object_property record. |
| Reference Time | hours, minutes, seconds | Defines the timestamp the system uses to include records in the trend log. |
| Max Records Per Request | 0 (default) to max | Defines the maximum number of records that can be archived in this trend log. The Framework asks for 10 records per request. |

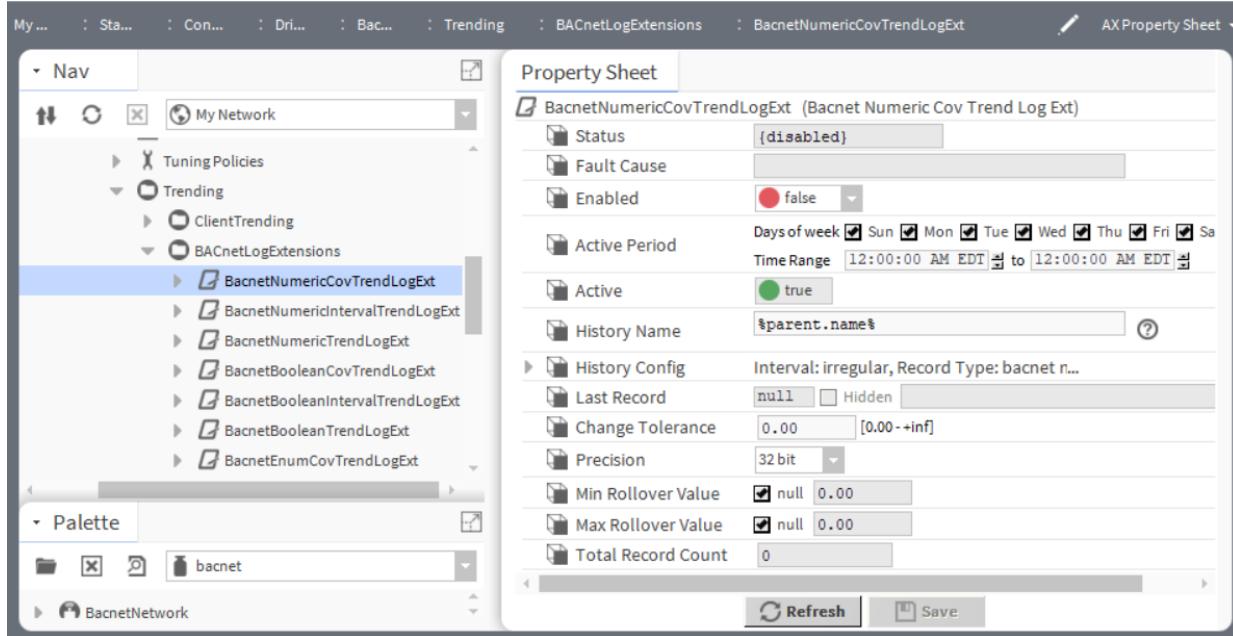
| Property | Value | Description |
|----------------------------------|-------------------------|------------------------------------------------------------------------------------------|
| Always Request By Reference Time | true or false (default) | Configures if the driver should always use the Reference Time to request records. |
| Last Sequence Number Processed | number | Defines a maximum record number to process. |
| Local History Names | read-only | Reports the names of local trend logs. |

Trending, Bacnet Numeric Cov Trend Log Ext (bacnet-BacnetNumericCovTrendLogExt)

This extension creates a history for the StatusNumeric out value of a component using COV.

Unlike the equivalent standard history extension (**NumericCovHistoryExt**), you can export the history created by this extension as a fully-compliant BACnet trend log object.

Figure 77 Bacnet Numeric Cov Trend Log Ext properties



To access these properties, expand **Config→Drivers→BacnetNetwork→BacnetDevice→Trend Logs→BacnetLogExtensions** and double-click the **BacnetNumericTrendLogExt**.

In addition to the common properties (Status, Enabled and Fault Cause), this component has these properties.

| Property | Value | Description |
|---------------------------------|-----------------------------------------------|----------------------------------------------------------------------------------|
| Active Period, Days of the Week | check boxes (all default to enabled) | Specifies the days of the week to include. |
| Active Period, Time Range | from and to times (both default to 12 AM EDT) | Configures when the time period starts and ends. |
| Active | read-only | Indicates the current state of the trend log based on the Active Period . |

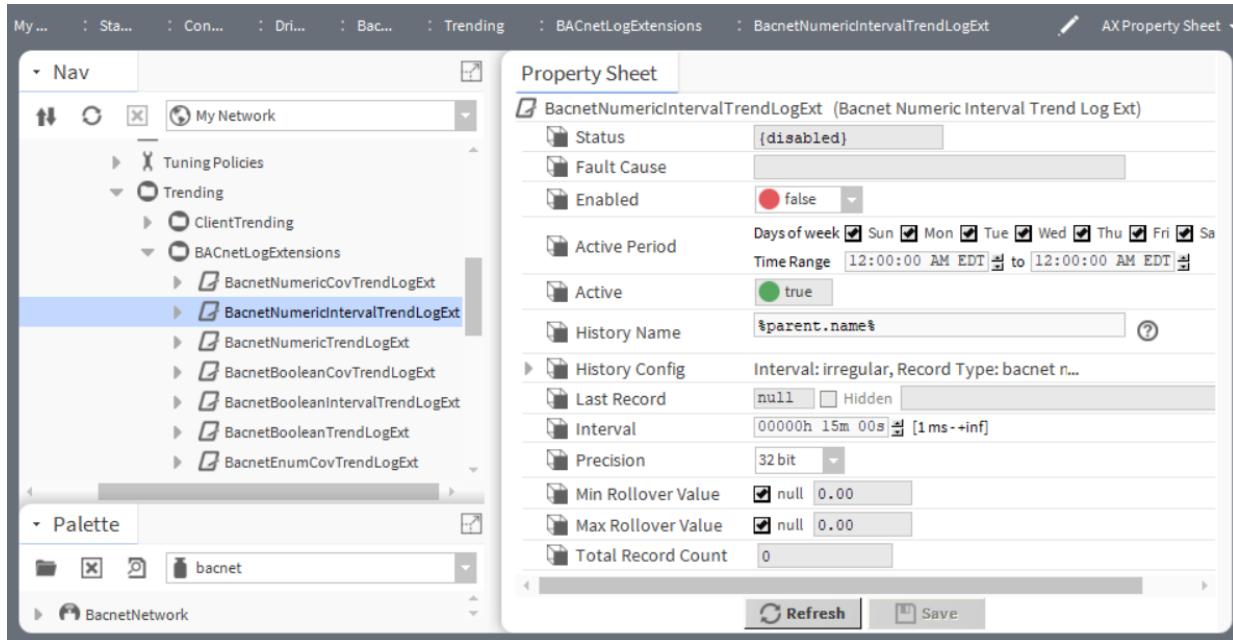
| Property | Value | Description |
|--------------------|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | true indicates the log is active. false indicates the log is no longer active because the current time is outside the Active Period . |
| History Name | BFormat string (defaults to %parent.name%) | Defines a pattern for deriving the name of the history created by this extension. It can be static text or a simple pattern that allows the actual history name to be derived from the context. The default value (%parent.name%) sets the history name in the Id to the name of the parent of this extension. Changes to this property cause the format to be reapplied and the history Id in the history config is to be updated. |
| History Config | additional properties | Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties. |
| Last record | read-only | Stores a copy of the most recent record successfully appended to the history for this extension. |
| Interval | hours, minutes, seconds | Controls data collection behavior for this Trend Log component. If the Interval value is set to 0, the driver collects trends when a change of value (COV) on the parent point occurs. The subscription lifetime is equal to twice (2x) the value of the COV Resubscription Interval property. If the Interval value is set to any number greater than 0, the driver collects trend data at the specified time interval. |
| Precision | 32 bit (default), 64 bit | Selects 32 bit or 64 bit options for the history data logging. The 64 bit option allows for higher level of precision but consumes more memory. |
| Min Rollover Value | number (defaults to null) | Defines the starting point for calculations for cumulative logging after a rollover. Rollover occurs after a running total maximum value is reached. Select the null option if a Min Rollover Value is unknown. |
| Max Rollover Value | number (defaults to null) | Defines a maximum value for calculations when a rollover is detected by the history logging process. Using this parameter and the Min Rollover Value parameter helps you avoid getting negative numbers when you are logging total data, such as energy usage. |
| Total Record Count | read-only | Indicates the total number of records collected by the Trend Log object since creation. When the value of Total Record Count reaches its maximum possible value, the next value it takes is zero. |

Bacnet Numeric Interval Trend Log Ext (bacnet-BacnetNumericIntervalTrendLogExt)

This extension collects a history for the **StatusNumeric** out value of a component using a defined interval.

Unlike the equivalent standard history extension (**NumericIntervalHistoryExt**), you can export this extension as a fully-compliant BACnet trend log object.

Figure 78



To access these properties, expand **Config→Drivers→BacnetNetwork→BacnetDevice→Trend Logs→BacnetLogExtensions** and double-click the **BacnetNumericIntervalTrendLogExt**.

In addition to the common properties (Status, Enabled and Fault Cause), this component has these properties.

| Property | Value | Description |
|---------------------------------|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Active Period, Days of the Week | check boxes (all default to enabled) | Specifies the days of the week to include. |
| Active Period, Time Range | from and to times (both default to 12 AM EDT) | Configures when the time period starts and ends. |
| Active | read-only | Indicates the current state of the trend log based on the Active Period . true indicates the log is active. false indicates the log is no longer active because the current time is outside the Active Period . |
| History Name | BFormat string (defaults to %parent.name%) | Defines a pattern for deriving the name of the history created by this extension. It can be static text or a simple pattern that allows the actual history name to be derived from the context. The default value (%parent.name%) sets the history name in the Id to the name of the parent of this extension. Changes to this property cause the format to be reapplied and the history Id in the history config is to be updated. |
| History Config | additional properties | Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties. |
| Last record | read-only | Stores a copy of the most recent record successfully appended to the history for this extension. |

| Property | Value | Description |
|--------------------|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Interval | hours, minutes, seconds | <p>Controls data collection behavior for this Trend Log component.</p> <p>If the Interval value is set to 0, the driver collects trends when a change of value (COV) on the parent point occurs. The subscription lifetime is equal to twice (2x) the value of the COV Resubscription Interval property.</p> <p>If the Interval value is set to any number greater than 0, the driver collects trend data at the specified time interval.</p> |
| Precision | 32 bit (default), 64 bit | Selects 32 bit or 64 bit options for the history data logging. The 64 bit option allows for higher level of precision but consumes more memory. |
| Min Rollover Value | number (defaults to null) | Defines the starting point for calculations for cumulative logging after a rollover. Rollover occurs after a running total maximum value is reached. Select the null option if a Min Rollover Value is unknown. |
| Max Rollover Value | number (defaults to null) | Defines a maximum value for calculations when a rollover is detected by the history logging process. Using this parameter and the Min Rollover Value parameter helps you avoid getting negative numbers when you are logging total data, such as energy usage. |
| Total Record Count | read-only | Indicates the total number of records collected by the Trend Log object since creation. When the value of Total Record Count reaches its maximum possible value, the next value it takes is zero. |

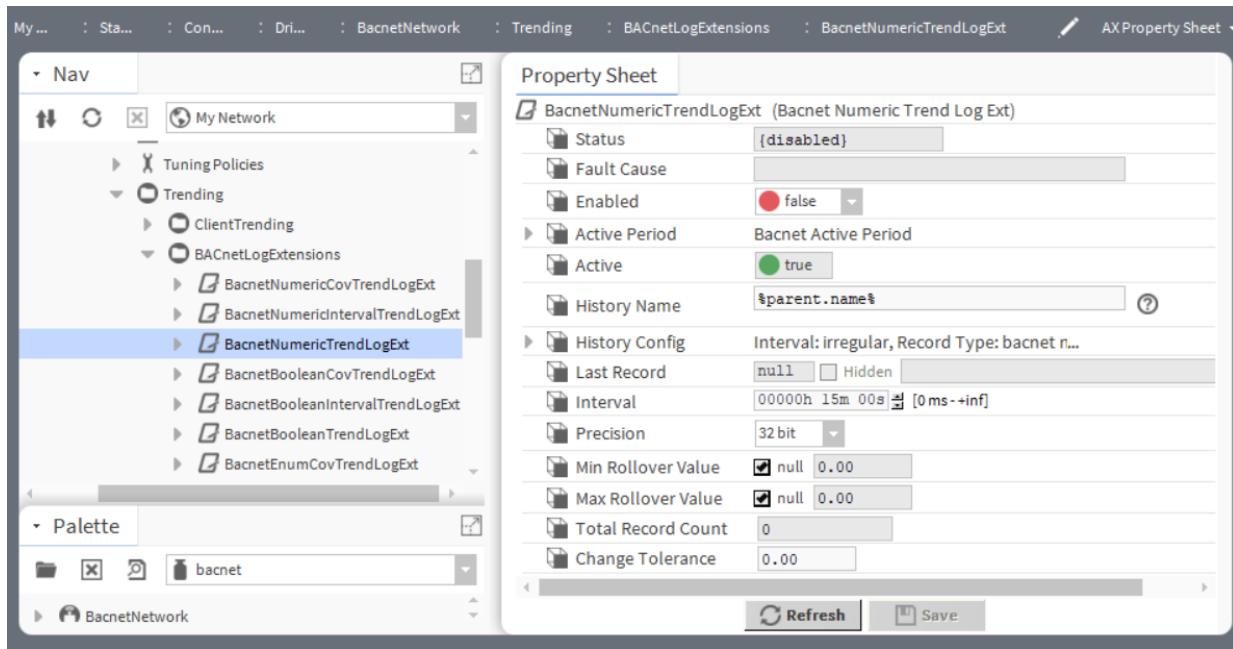
Bacnet Numeric Trend Log Ext (bacnet-BacnetNumericTrendLogExt)

This extension creates a history for the StatusNumeric out value of a component using either a COV or a defined interval. It is available in the **bacnet** palette.

This extension resides on the a BACnet device. You can export the history created by this extension as a fully-compliant BACnet trend log object.

NOTE: Only the following read-only properties qualify for initial value setting during object creation. Requests that include non-qualified properties from the client cause a request to fail.

- Device Status Facets
- Discovery History Type

Figure 79 Bacnet Numeric Trend Log Ext properties

To access these properties, expand **Config→Drivers→BacnetNetwork→BacnetDevice→Trend Logs→BacnetLogExtensions** and double-click the **BacnetNumericTrendLogExt**.

In addition to the common properties (Status, Enabled and Fault Cause), this component has these properties.

| Property | Value | Description |
|---------------------------|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Active Period, Start Time | Date and Time selectors | Configures when to start the log. |
| Active Period, Stop Time | Date and Time selectors | Configures when to stop the log. |
| Active | read-only | Indicates the current state of the trend log based on the Active Period . true indicates the log is active. false indicates the log is no longer active because the current time is outside the Active Period . |
| History Name | BFormat string (defaults to %parent.name%) | Defines a pattern for deriving the name of the history created by this extension. It can be static text or a simple pattern that allows the actual history name to be derived from the context. The default value (%parent.name%) sets the history name in the Id to the name of the parent of this extension. Changes to this property cause the format to be reapplied and the history Id in the history config is to be updated. |
| History Config | additional properties | Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties. |
| Last record | read-only | Stores a copy of the most recent record successfully appended to the history for this extension. |

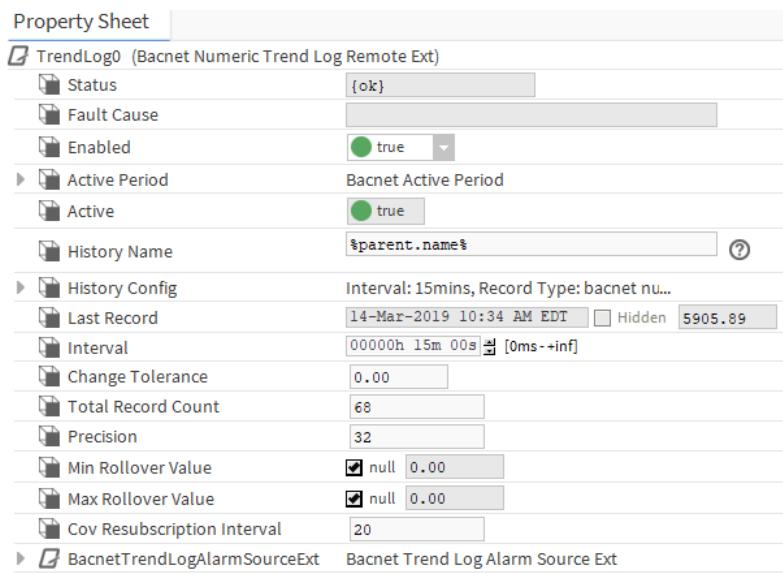
| Property | Value | Description |
|--------------------|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Interval | hours, minutes, seconds | Controls data collection behavior for this Trend Log component. If the Interval value is set to 0, the driver collects trends when a change of value (COV) on the parent point occurs. The subscription lifetime is equal to twice (2x) the value of the COV Resubscription Interval property. If the Interval value is set to any number greater than 0, the driver collects trend data at the specified time interval. |
| Precision | 32 bit (default), 64 bit | Selects 32 bit or 64 bit options for the history data logging. The 64 bit option allows for higher level of precision but consumes more memory. |
| Min Rollover Value | number (defaults to null) | Defines the starting point for calculations for cumulative logging after a rollover. Rollover occurs after a running total maximum value is reached. Select the null option if a Min Rollover Value is unknown. |
| Max Rollover Value | number (defaults to null) | Defines a maximum value for calculations when a rollover is detected by the history logging process. Using this parameter and the Min Rollover Value parameter helps you avoid getting negative numbers when you are logging total data, such as energy usage. |
| Total Record Count | read-only | Indicates the total number of records collected by the Trend Log object since creation. When the value of Total Record Count reaches its maximum possible value, the next value it takes is zero. |
| Change Tolerance | text | Specifies the minimum change in value that must occur before the system writes a record to the database. |

Bacnet Numeric Trend Log Remote Ext (bacnet-BacnetNumericTrendLogRemoteExt)

This extension is created dynamically for a remote BACnet device. It creates a history for the StatusNumeric out value of a component using COV or a defined interval.

You can export the history created by this extension as a fully-compliant BACnet trend log object.

NOTE: This single extension applies to both COV and Interval data collection and may be used in place of the deprecated COV and Interval Trend Log Extensions for this data type.

Figure 80 Bacnet Numeric Trend Log Remote Ext properties

In addition to the common properties (Status, Enabled and Fault Cause), this component has these properties.

| Property | Value | Description |
|---------------------------------|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Active Period, Days of the Week | check boxes (all default to enabled) | Specifies the days of the week to include. |
| Active Period, Time Range | from and to times (both default to 12 AM EDT) | Configures when the time period starts and ends. |
| Active | read-only | Indicates the current state of the trend log based on the Active Period . true indicates the log is active. false indicates the log is no longer active because the current time is outside the Active Period . |
| History Name | BFormat string (defaults to %parent.name%) | Defines a pattern for deriving the name of the history created by this extension. It can be static text or a simple pattern that allows the actual history name to be derived from the context. The default value (%parent.name%) sets the history name in the Id to the name of the parent of this extension. Changes to this property cause the format to be reapplied and the history Id in the history config is to be updated. |
| History Config | additional properties | Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties. |
| Last record | read-only | Stores a copy of the most recent record successfully appended to the history for this extension. |
| Interval | hours, minutes, seconds | Controls data collection behavior for this Trend Log component. |

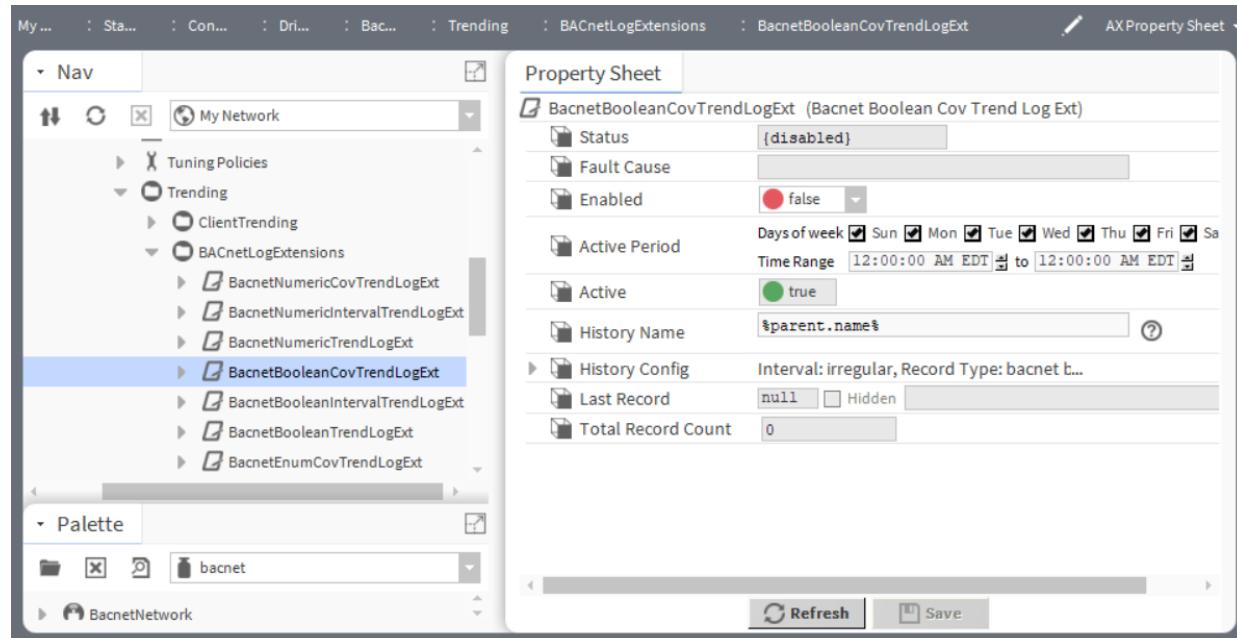
| Property | Value | Description |
|-----------------------------|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | If the Interval value is set to 0, the driver collects trends when a change of value (COV) on the parent point occurs. The subscription lifetime is equal to twice (2x) the value of the COV Resubscription Interval property. If the Interval value is set to any number greater than 0, the driver collects trend data at the specified time interval. |
| Precision | 32 bit (default), 64 bit | Selects 32 bit or 64 bit options for the history data logging. The 64 bit option allows for higher level of precision but consumes more memory. |
| Min Rollover Value | number (defaults to null) | Defines the starting point for calculations for cumulative logging after a rollover. Rollover occurs after a running total maximum value is reached. Select the null option if a Min Rollover Value is unknown. |
| Max Rollover Value | number (defaults to null) | Defines a maximum value for calculations when a rollover is detected by the history logging process. Using this parameter and the Min Rollover Value parameter helps you avoid getting negative numbers when you are logging total data, such as energy usage. |
| Cov Resubscription Interval | time range (defaults to 5 minutes) | Specifies the length of time that the extension waits before initiating a re-subscription request. |

Bacnet Boolean Cov Trend Log Ext (bacnet-BacnetBooleanCovTrendLogExt)

This extension collects a history for the StatusBoolean out value of a component using COV (Change of Value).

Unlike the standard history extension (BooleanCOVHistoryExt), you can export the history created by this extension as a fully-compliant BACnet trend log object.

Figure 81 Bacnet Boolean Cov Trend Log Ext properties



To access these properties, expand **Config→Drivers→BacnetNetwork→BacnetDevice→Trend Logs→BacnetLogExtensions** and double-click the **BacnetBooleanCovTrendLogExt**.

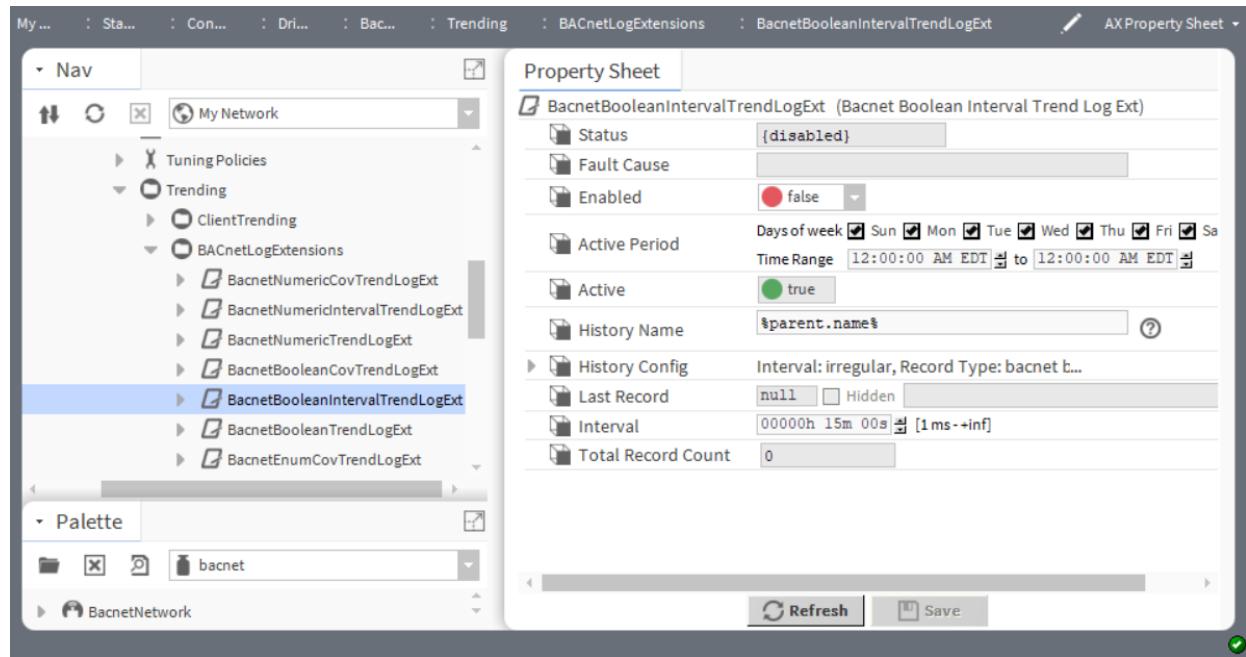
In addition to the common properties (Status, Fault Cause and Enabled), this component has these properties.

| Property | Value | Description |
|---------------------------------|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Active Period, Days of the Week | check boxes (all default to enabled) | Specifies the days of the week to include. |
| Active Period, Time Range | from and to times (both default to 12 AM EDT) | Configures when the time period starts and ends. |
| Active | read-only | Indicates the current state of the trend log based on the Active Period . true indicates the log is active. false indicates the log is no longer active because the current time is outside the Active Period . |
| History Name | BFormat string (defaults to %parent.name%) | Defines a pattern for deriving the name of the history created by this extension. It can be static text or a simple pattern that allows the actual history name to be derived from the context. The default value (%parent.name%) sets the history name in the Id to the name of the parent of this extension. Changes to this property cause the format to be reapplied and the history Id in the history config is to be updated. |
| History Config | additional properties | Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties. |
| Last record | read-only | Stores a copy of the most recent record successfully appended to the history for this extension. |
| Total Record Count | read-only | Indicates the total number of records collected by the Trend Log object since creation. When the value of Total Record Count reaches its maximum possible value, the next value it takes is zero. |

Bacnet Boolean Interval Trend Log Ext (bacnet-BacnetBooleanIntervalTrendLogExt)

This extension collects a history for the Status Boolean out value of a component using a defined interval.

Unlike the equivalent standard history extension (**BooleanIntervalHistoryExt**), you can export the history created by this extension as a fully-compliant BACnet trend log object.

Figure 82 Bacnet Boolean Interval Trend Log Ext properties

To access these properties, expand **Config**→**Drivers**→**BacnetNetwork**→**BacnetDevice**→**Trend Logs**→**BacnetLogExtensions** and double-click the **BacnetBooleanIntervalTrendLogExt**.

In addition to the common properties (Status, Fault Cause and Enabled), this component has these properties.

| Property | Value | Description |
|---------------------------------|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Active Period, Days of the Week | check boxes (all default to enabled) | Specifies the days of the week to include. |
| Active Period, Time Range | from and to times (both default to 12 AM EDT) | Configures when the time period starts and ends. |
| Active | read-only | Indicates the current state of the trend log based on the Active Period . true indicates the log is active. false indicates the log is no longer active because the current time is outside the Active Period . |
| History Name | BFormat string (defaults to %parent.name%) | Defines a pattern for deriving the name of the history created by this extension. It can be static text or a simple pattern that allows the actual history name to be derived from the context. The default value (%parent.name%) sets the history name in the Id to the name of the parent of this extension. Changes to this property cause the format to be reapplied and the history Id in the history config is to be updated. |
| History Config | additional properties | Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties. |

| Property | Value | Description |
|--------------------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Last record | read-only | Stores a copy of the most recent record successfully appended to the history for this extension. |
| Interval | hours, minutes, seconds | Controls data collection behavior for this Trend Log component. If the Interval value is set to 0, the driver collects trends when a change of value (COV) on the parent point occurs. The subscription lifetime is equal to twice (2x) the value of the COV Resubscription Interval property. If the Interval value is set to any number greater than 0, the driver collects trend data at the specified time interval. |
| Total Record Count | read-only | Indicates the total number of records collected by the Trend Log object since creation. When the value of Total Record Count reaches its maximum possible value, the next value it takes is zero. |

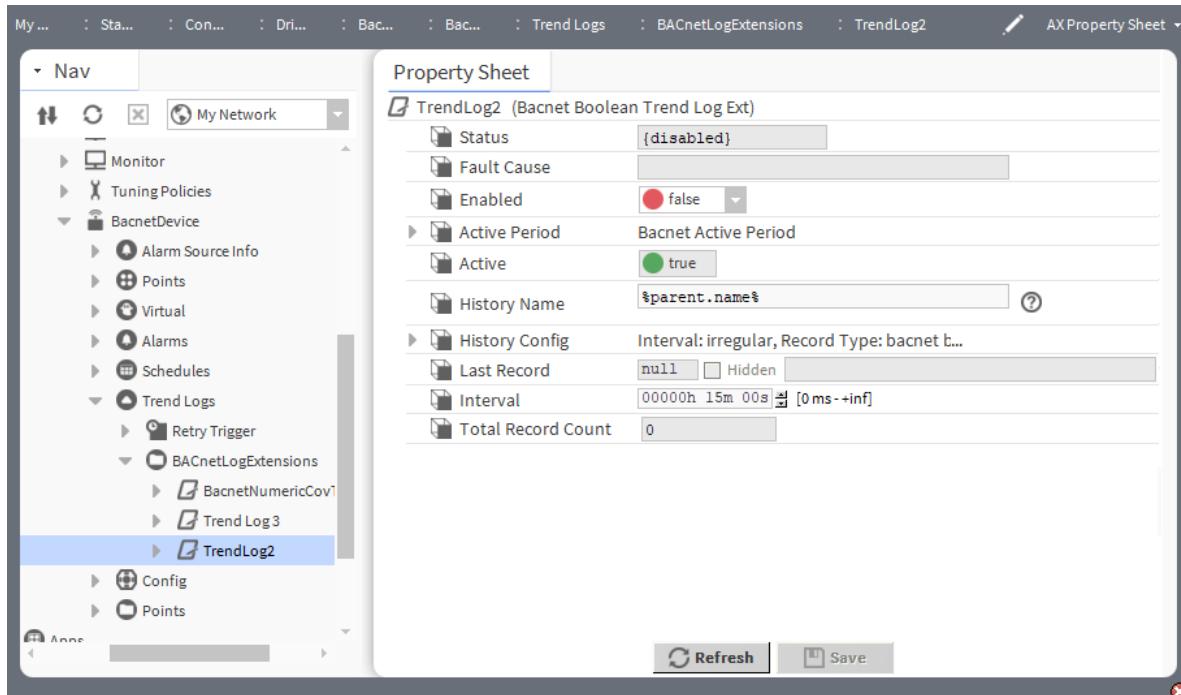
Bacnet Boolean Trend Log Ext (bacnet-BacnetBooleanTrendLogExt)

This extension creates a history for the StatusBoolean out value of a component using COV or a defined interval.

You can export the history created by this extension as a fully-compliant BACnet trend log object.

NOTE: This single extension applies to both COV and Interval data collection and may be used in place of the deprecated COV and Interval Trend Log Extensions for this data type.

Figure 83 Bacnet Boolean Trend Log Ext properties



To access these properties, expand **Config**→**Drivers**→**BacnetNetwork**→**BacnetDevice**→**Trend Logs**→**BacnetLogExtensions** and double-click the **BacnetBooleanTrendLogExt**.

In addition to the common properties (Status, Fault Cause and Enabled), this component has these properties.

| Property | Value | Description |
|---------------------------|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Active Period, Start Time | Date and Time selectors | Configures when to start the log. |
| Active Period, Stop Time | Date and Time selectors | Configures when to stop the log. |
| Active | read-only | Indicates the current state of the trend log based on the Active Period . true indicates the log is active. false indicates the log is no longer active because the current time is outside the Active Period . |
| History Name | BFormat string (defaults to %parent.name%) | Defines a pattern for deriving the name of the history created by this extension. It can be static text or a simple pattern that allows the actual history name to be derived from the context. The default value (%parent.name%) sets the history name in the Id to the name of the parent of this extension. Changes to this property cause the format to be reapplied and the history Id in the history config is to be updated. |
| History Config | additional properties | Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties. |
| Last record | read-only | Stores a copy of the most recent record successfully appended to the history for this extension. |
| Interval | hours, minutes, seconds | Controls data collection behavior for this Trend Log component. If the Interval value is set to 0, the driver collects trends when a change of value (COV) on the parent point occurs. The subscription lifetime is equal to twice (2x) the value of the COV Resubscription Interval property. If the Interval value is set to any number greater than 0, the driver collects trend data at the specified time interval. |
| Total Record Count | read-only | Indicates the total number of records collected by the Trend Log object since creation. When the value of Total Record Count reaches its maximum possible value, the next value it takes is zero. |

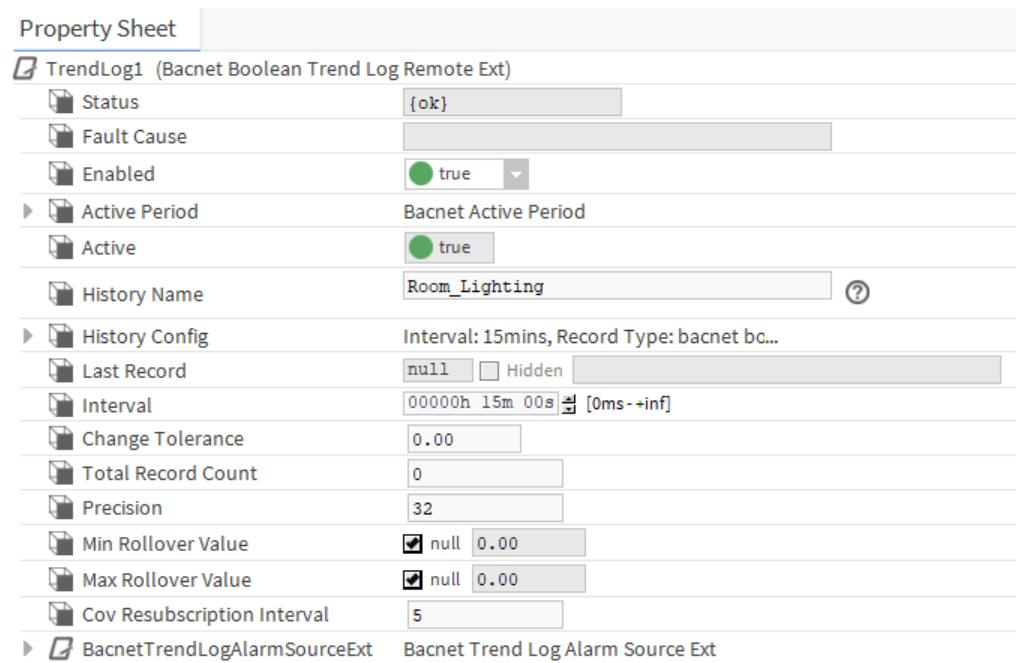
Bacnet Boolean Trend Log Ext (bacnet-BacnetBooleanTrendRemoteLogExt)

This extension is created dynamically for a remote BACnet device. It creates a history for the StatusBoolean out value of a component using COV or a defined interval.

You can export the history created by this extension as a fully-compliant BACnet trend log object.

NOTE: This single extension applies to both COV and Interval data collection and may be used in place of the deprecated COV and Interval Trend Log Extensions for this data type.

Figure 84 Bacnet Boolean Trend Log Remote Ext properties



In addition to the common properties (Status, Enabled and Fault Cause), this component has these properties.

| Property | Value | Description |
|---------------------------|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Active Period, Start Time | Date and Time selectors | Configures when to start the log. |
| Active Period, Stop Time | Date and Time selectors | Configures when to stop the log. |
| Active | read-only | Indicates the current state of the trend log based on the Active Period . <code>true</code> indicates the log is active. <code>false</code> indicates the log is no longer active because the current time is outside the Active Period . |
| History Name | BFormat string (defaults to %parent.name%) | Defines a pattern for deriving the name of the history created by this extension. It can be static text or a simple pattern that allows the actual history name to be derived from the context. The default value (%parent.name%) sets the history name in the Id to the name of the parent of this extension. Changes to this property cause the format to be reapplied and the history Id in the history config is to be updated. |
| History Config | additional properties | Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties. |
| Last record | read-only | Stores a copy of the most recent record successfully appended to the history for this extension. |
| Interval | hours, minutes, seconds | Controls data collection behavior for this Trend Log component. |

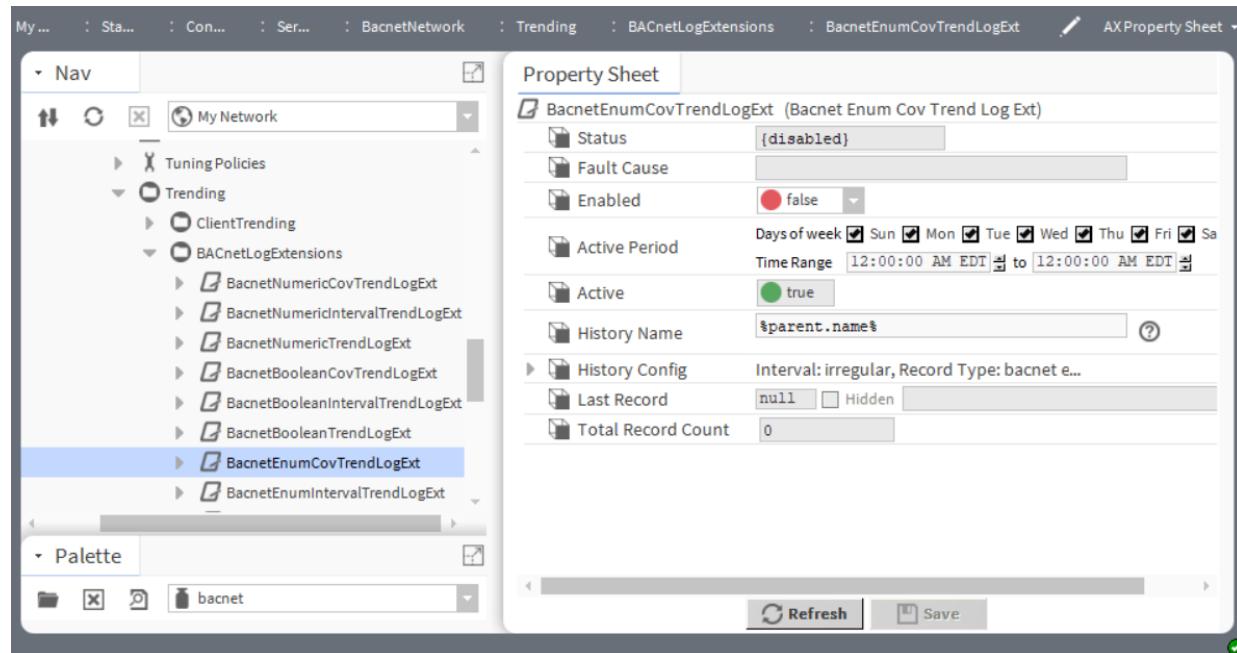
| Property | Value | Description |
|-----------------------------|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | If the Interval value is set to 0, the driver collects trends when a change of value (COV) on the parent point occurs. The subscription lifetime is equal to twice (2x) the value of the COV Resubscription Interval property. If the Interval value is set to any number greater than 0, the driver collects trend data at the specified time interval. |
| Precision | 32 bit (default), 64 bit | Selects 32 bit or 64 bit options for the history data logging. The 64 bit option allows for higher level of precision but consumes more memory. |
| Min Rollover Value | number (defaults to null) | Defines the starting point for calculations for cumulative logging after a rollover. Rollover occurs after a running total maximum value is reached. Select the null option if a Min Rollover Value is unknown. |
| Max Rollover Value | number (defaults to null) | Defines a maximum value for calculations when a rollover is detected by the history logging process. Using this parameter and the Min Rollover Value parameter helps you avoid getting negative numbers when you are logging total data, such as energy usage. |
| Cov Resubscription Interval | time range (defaults to 5 minutes) | Specifies the length of time that the extension waits before initiating a re-subscription request. |

Bacnet Enum Cov Trend Log Ext (bacnet-BacnetEnumCovTrendLogExt)

This extension collects a history for the StatusEnum out value of a component using COV.

Unlike the equivalent standard history extension (EnumCovHistoryExt), you can export the history created by this extension as a fully-compliant BACnet trend log object.

Figure 85 Bacnet Enum Cov Trend Log Ext properties



To access these properties, expand **Config→Drivers→BacnetNetwork→BacnetDevice→Trend Logs→BacnetLogExtensions** and double-click the **BacnetEnumCovTrendLogExt**.

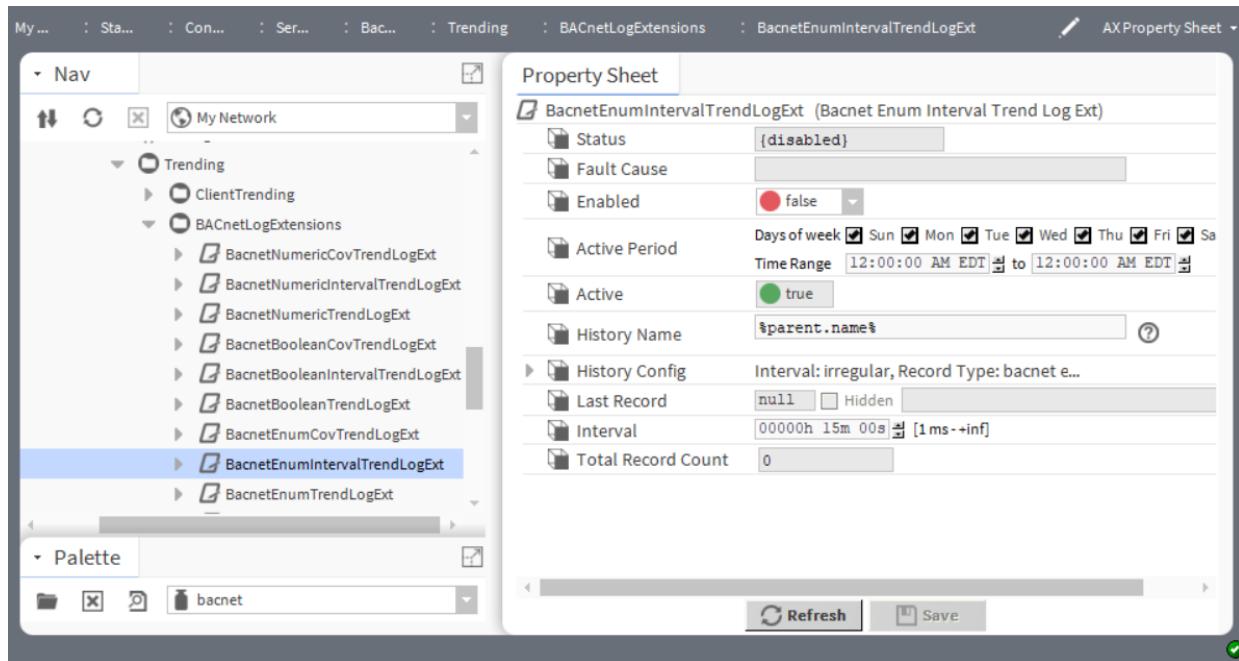
In addition to the common properties (Status, Fault Cause and Enabled), this component has these properties.

| Property | Value | Description |
|---------------------------------|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Active Period, Days of the Week | check boxes (all default to enabled) | Specifies the days of the week to include. |
| Active Period, Time Range | from and to times (both default to 12 AM EDT) | Configures when the time period starts and ends. |
| Active | read-only | Indicates the current state of the trend log based on the Active Period . true indicates the log is active. false indicates the log is no longer active because the current time is outside the Active Period . |
| History Name | BFormat string (defaults to %parent.name%) | Defines a pattern for deriving the name of the history created by this extension. It can be static text or a simple pattern that allows the actual history name to be derived from the context. The default value (%parent.name%) sets the history name in the Id to the name of the parent of this extension. Changes to this property cause the format to be reapplied and the history Id in the history config is to be updated. |
| History Config | additional properties | Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties. |
| Last record | read-only | Stores a copy of the most recent record successfully appended to the history for this extension. |
| Total Record Count | read-only | Indicates the total number of records collected by the Trend Log object since creation. When the value of Total Record Count reaches its maximum possible value, the next value it takes is zero. |

Bacnet Enum Interval Trend Log Ext (bacnet-BacnetEnumIntervalTrendLogExt)

This extension collects a history for the StatusEnum out value of a component using a defined interval.

Unlike the equivalent standard history extension (EnumIntervalHistoryExt), you can export the history created by this extension as a fully-compliant BACnet trend log object.

Figure 86 Bacnet Enum Interval Trend Log Ext properties

To access these properties, expand **Config→Drivers→BacnetNetwork→BacnetDevice→Trend Logs→BacnetLogExtensions** and double-click the **BacnetEnumIntervalTrendLogExt**.

In addition to the common properties (Status, Fault Cause and Enabled), this component has these properties.

| Property | Value | Description |
|---------------------------------|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Active Period, Days of the Week | check boxes (all default to enabled) | Specifies the days of the week to include. |
| Active Period, Time Range | from and to times (both default to 12 AM EDT) | Configures when the time period starts and ends. |
| Active | read-only | Indicates the current state of the trend log based on the Active Period . true indicates the log is active. false indicates the log is no longer active because the current time is outside the Active Period . |
| History Name | BFormat string (defaults to %parent.name%) | Defines a pattern for deriving the name of the history created by this extension. It can be static text or a simple pattern that allows the actual history name to be derived from the context. The default value (%parent.name%) sets the history name in the Id to the name of the parent of this extension. Changes to this property cause the format to be reapplied and the history Id in the history config is to be updated. |
| History Config | additional properties | Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties. |

| Property | Value | Description |
|--------------------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Last record | read-only | Stores a copy of the most recent record successfully appended to the history for this extension. |
| Interval | hours, minutes, seconds | Controls data collection behavior for this Trend Log component. If the Interval value is set to 0, the driver collects trends when a change of value (COV) on the parent point occurs. The subscription lifetime is equal to twice (2x) the value of the COV Resubscription Interval property. If the Interval value is set to any number greater than 0, the driver collects trend data at the specified time interval. |
| Total Record Count | read-only | Indicates the total number of records collected by the Trend Log object since creation. When the value of Total Record Count reaches its maximum possible value, the next value it takes is zero. |

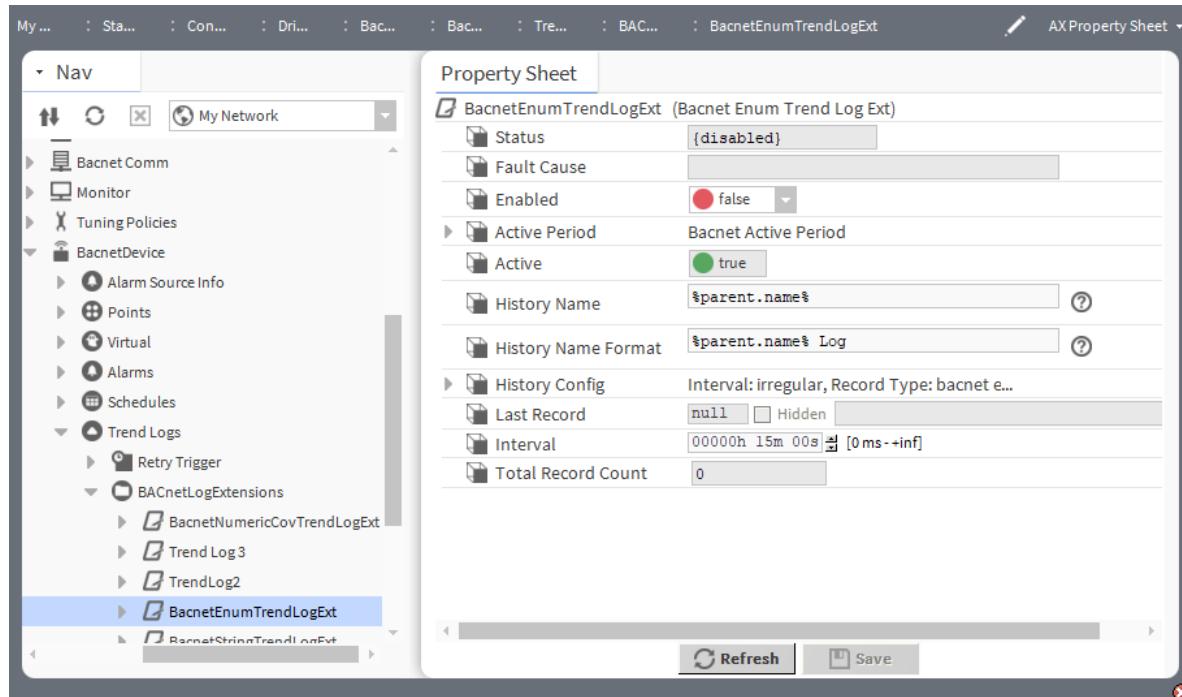
Bacnet Enum Trend Log Ext (bacnet-BacnetEnumTrendLogExt)

This extension creates a history for the StatusEnum out value of a component using COV or a defined interval.

You can export the history created by this extension as a fully-compliant BACnet trend log object.

NOTE: This single extension applies to both COV and Interval data collection and may be used in place of the deprecated COV and Interval Trend Log Extensions for this data type.

Figure 87 Bacnet Enum Trend Log Ext properties



To access these properties, expand **Config**→**Drivers**→**BacnetNetwork**→**BacnetDevice**→**Trend Logs**→**BacnetLogExtensions** and double-click the **BacnetEnumTrendLogExt**.

In addition to the common properties (Status, Fault Cause and Enabled), this component has these properties.

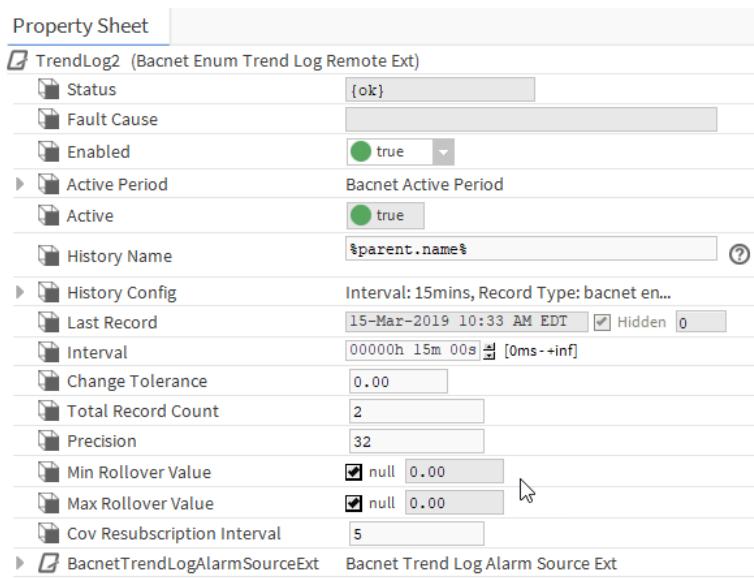
| Property | Value | Description |
|---------------------------|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Active Period, Start Time | Date and Time selectors | Configures when to start the log. |
| Active Period, Stop Time | Date and Time selectors | Configures when to stop the log. |
| Active | read-only | Indicates the current state of the trend log based on the Active Period . true indicates the log is active. false indicates the log is no longer active because the current time is outside the Active Period . |
| History Name | BFormat string (defaults to %parent.name%) | Defines a pattern for deriving the name of the history created by this extension. It can be static text or a simple pattern that allows the actual history name to be derived from the context. The default value (%parent.name%) sets the history name in the Id to the name of the parent of this extension. Changes to this property cause the format to be reapplied and the history Id in the history config is to be updated. |
| History Config | additional properties | Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties. |
| Last record | read-only | Stores a copy of the most recent record successfully appended to the history for this extension. |
| Interval | hours, minutes, seconds | Controls data collection behavior for this Trend Log component. If the Interval value is set to 0, the driver collects trends when a change of value (COV) on the parent point occurs. The subscription lifetime is equal to twice (2x) the value of the COV Resubscription Interval property. If the Interval value is set to any number greater than 0, the driver collects trend data at the specified time interval. |
| Total Record Count | read-only | Indicates the total number of records collected by the Trend Log object since creation. When the value of Total Record Count reaches its maximum possible value, the next value it takes is zero. |

Bacnet Enum Trend Log Remote Ext (bacnet-BacnetEnumTrendLogRemoteExt)

This extension creates a history for the StatusEnum out value of a component using COV or a defined interval.

You can export the history created by this extension as a fully-compliant BACnet trend log object.

NOTE: This single extension applies to both COV and Interval data collection and may be used in place of the deprecated COV and Interval Trend Log Extensions for this data type.

Figure 88 Bacnet Enum Trend Log Remote Ext properties

In addition to the common properties (Status, Fault Cause and Enabled), this component has these properties.

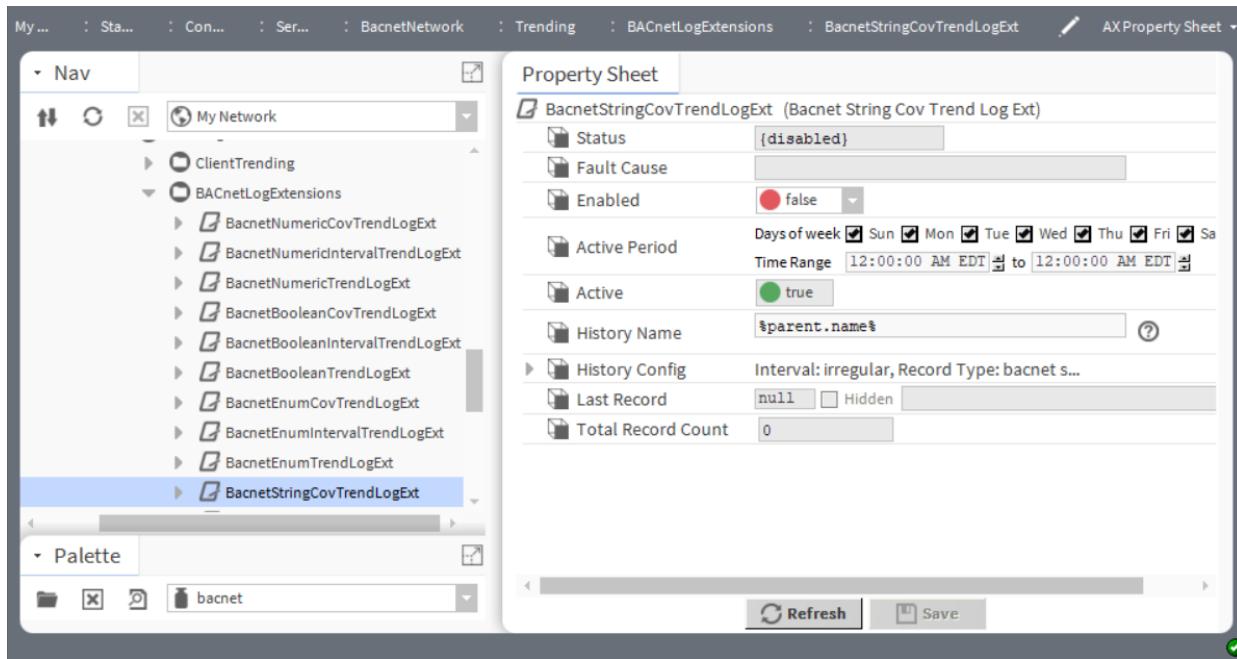
| Property | Value | Description |
|---------------------------|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Active Period, Start Time | Date and Time selectors | Configures when to start the log. |
| Active Period, Stop Time | Date and Time selectors | Configures when to stop the log. |
| Active | read-only | Indicates the current state of the trend log based on the Active Period . true indicates the log is active. false indicates the log is no longer active because the current time is outside the Active Period . |
| History Name | BFormat string (defaults to %parent.name%) | Defines a pattern for deriving the name of the history created by this extension. It can be static text or a simple pattern that allows the actual history name to be derived from the context. The default value (%parent.name%) sets the history name in the Id to the name of the parent of this extension. Changes to this property cause the format to be reapplied and the history Id in the history config is to be updated. |
| History Config | additional properties | Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties. |
| Last record | read-only | Stores a copy of the most recent record successfully appended to the history for this extension. |
| Interval | hours, minutes, seconds | Controls data collection behavior for this Trend Log component. If the Interval value is set to 0, the driver collects trends when a change of value (COV) on the parent point occurs. The |

| Property | Value | Description |
|-----------------------------|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | subscription lifetime is equal to twice (2x) the value of the COV Resubscription Interval property. If the Interval value is set to any number greater than 0, the driver collects trend data at the specified time interval. |
| Change Tolerance | text | Specifies the minimum change in value that must occur before the system writes a record to the database. |
| Total Record Count | read-only | Indicates the total number of records collected by the Trend Log object since creation. When the value of Total Record Count reaches its maximum possible value, the next value it takes is zero. |
| Precision | 32 bit (default), 64 bit | Selects 32 bit or 64 bit options for the history data logging. The 64 bit option allows for higher level of precision but consumes more memory. |
| Min Rollover Value | number (defaults to null) | Defines the starting point for calculations for cumulative logging after a rollover. Rollover occurs after a running total maximum value is reached. Select the null option if a Min Rollover Value is unknown. |
| Max Rollover Value | number (defaults to null) | Defines a maximum value for calculations when a rollover is detected by the history logging process. Using this parameter and the Min Rollover Value parameter helps you avoid getting negative numbers when you are logging total data, such as energy usage. |
| Cov Resubscription Interval | time range (defaults to 5 minutes) | Specifies the length of time that the extension waits before initiating a re-subscription request. |

Bacnet String Cov Trend Log Ext (bacnet-BacnetStringCovTrendLogExt)

This extension collects a history for the StatusString out value of a component using COV.

Unlike the equivalent standard history extension (StringIntervalHistoryExt), you can export the history created by this extension as a fully-compliant BACnet trend log object.

Figure 89 Bacnet String Cov Trend Log Ext properties

To access these properties, expand **Config→Drivers→BacnetNetwork→BacnetDevice→Trend Logs→BacnetLogExtensions** and double-click the **BacnetStringCovTrendLogExt**.

In addition to the common properties (Status, Fault Cause and Enabled), this component has these properties.

| Property | Value | Description |
|---------------------------------|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Active Period, Days of the Week | check boxes (all default to enabled) | Specifies the days of the week to include. |
| Active Period, Time Range | from and to times (both default to 12 AM EDT) | Configures when the time period starts and ends. |
| Active | read-only | Indicates the current state of the trend log based on the Active Period . true indicates the log is active. false indicates the log is no longer active because the current time is outside the Active Period . |
| History Name | BFormat string (defaults to %parent.name%) | Defines a pattern for deriving the name of the history created by this extension. It can be static text or a simple pattern that allows the actual history name to be derived from the context. The default value (%parent.name%) sets the history name in the Id to the name of the parent of this extension. Changes to this property cause the format to be reapplied and the history Id in the history config is to be updated. |
| History Config | additional properties | Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties. |

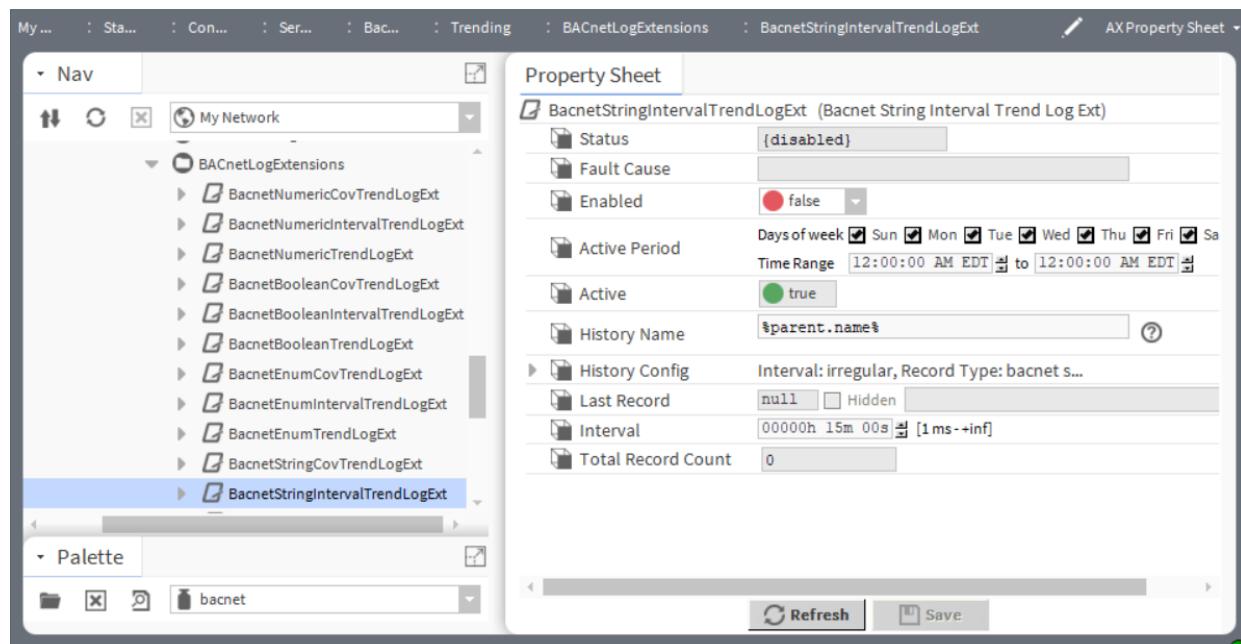
| Property | Value | Description |
|--------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Last record | read-only | Stores a copy of the most recent record successfully appended to the history for this extension. |
| Total Record Count | read-only | Indicates the total number of records collected by the Trend Log object since creation. When the value of Total Record Count reaches its maximum possible value, the next value it takes is zero. |

Bacnet String Interval Trend Log Ext (bacnet-BacnetStringIntervalTrendLogExt)

This extension collects a history for the StatusString out value of a component using a defined interval.

Unlike the equivalent standard history extension (StringIntervalHistoryExt), you can export the history created by this extension as a fully-compliant BACnet trend log object.

Figure 90 Bacnet String Interval Trend Log Ext properties



To access these properties, expand **Config→Drivers→BacnetNetwork→BacnetDevice→Trend Logs→BacnetLogExtensions** and double-click the **BacnetStringIntervalTrendLogExt**.

In addition to the common properties (Status, Fault Cause and Enabled), this component has these properties.

| Property | Value | Description |
|---------------------------------|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| Active Period, Days of the Week | check boxes (all default to enabled) | Specifies the days of the week to include. |
| Active Period, Time Range | from and to times (both default to 12 AM EDT) | Configures when the time period starts and ends. |
| Active | read-only | Indicates the current state of the trend log based on the Active Period . true indicates the log is active. |

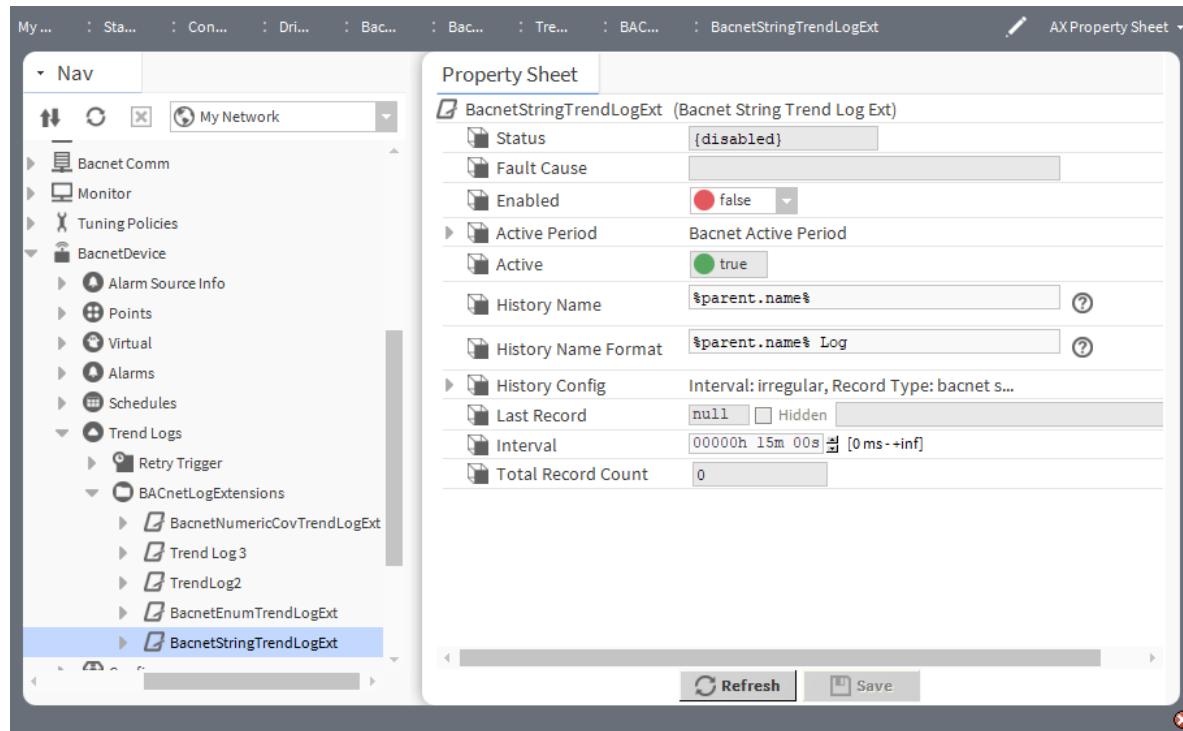
| Property | Value | Description |
|--------------------|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | false indicates the log is no longer active because the current time is outside the Active Period . |
| History Name | BFormat string (defaults to %parent.name%) | Defines a pattern for deriving the name of the history created by this extension. It can be static text or a simple pattern that allows the actual history name to be derived from the context. The default value (%parent.name%) sets the history name in the Id to the name of the parent of this extension. Changes to this property cause the format to be reapplied and the history Id in the history config is to be updated. |
| History Config | additional properties | Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties. |
| Last record | read-only | Stores a copy of the most recent record successfully appended to the history for this extension. |
| Interval | hours, minutes, seconds | Controls data collection behavior for this Trend Log component. If the Interval value is set to 0, the driver collects trends when a change of value (COV) on the parent point occurs. The subscription lifetime is equal to twice (2x) the value of the COV Resubscription Interval property. If the Interval value is set to any number greater than 0, the driver collects trend data at the specified time interval. |
| Total Record Count | read-only | Indicates the total number of records collected by the Trend Log object since creation. When the value of Total Record Count reaches its maximum possible value, the next value it takes is zero. |

Bacnet String Trend Log Ext (bacnet-BacnetStringTrendLogExt)

This extension creates a history for the StatusString out value of a component using COV or a defined interval.

You can export the history created by this extension as a fully-compliant BACnet trend log object.

NOTE: This single extension applies to both COV and Interval data collection and may be used in place of the deprecated COV and Interval Trend Log Extensions for this data type.

Figure 91 Bacnet String Trend Log Ext properties

To access these properties, expand **Config→Drivers→BacnetNetwork→BacnetDevice→Trend Logs→BacnetLogExtensions** and double-click the **BacnetStringTrendLogExt**.

In addition to the common properties (Status, Fault Cause and Enabled), this component has these properties.

| Property | Value | Description |
|---------------------------|----------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Active Period, Start Time | Date and Time selectors | Configures when to start the log. |
| Active Period, Stop Time | Date and Time selectors | Configures when to stop the log. |
| Active | read-only | Indicates the current state of the trend log based on the Active Period . <code>true</code> indicates the log is active. <code>false</code> indicates the log is no longer active because the current time is outside the Active Period . |
| History Name | BFormat string (defaults to <code>%parent.name%</code>) | Defines a pattern for deriving the name of the history created by this extension. It can be static text or a simple pattern that allows the actual history name to be derived from the context. The default value (<code>%parent.name%</code>) sets the history name in the Id to the name of the parent of this extension. Changes to this property cause the format to be reapplied and the history Id in the history config is to be updated. |
| History Config | additional properties | Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties. |

| Property | Value | Description |
|--------------------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Last record | read-only | Stores a copy of the most recent record successfully appended to the history for this extension. |
| Interval | hours, minutes, seconds | Controls data collection behavior for this Trend Log component. If the Interval value is set to 0, the driver collects trends when a change of value (COV) on the parent point occurs. The subscription lifetime is equal to twice (2x) the value of the COV Resubscription Interval property. If the Interval value is set to any number greater than 0, the driver collects trend data at the specified time interval. |
| Total Record Count | read-only | Indicates the total number of records collected by the Trend Log object since creation. When the value of Total Record Count reaches its maximum possible value, the next value it takes is zero. |

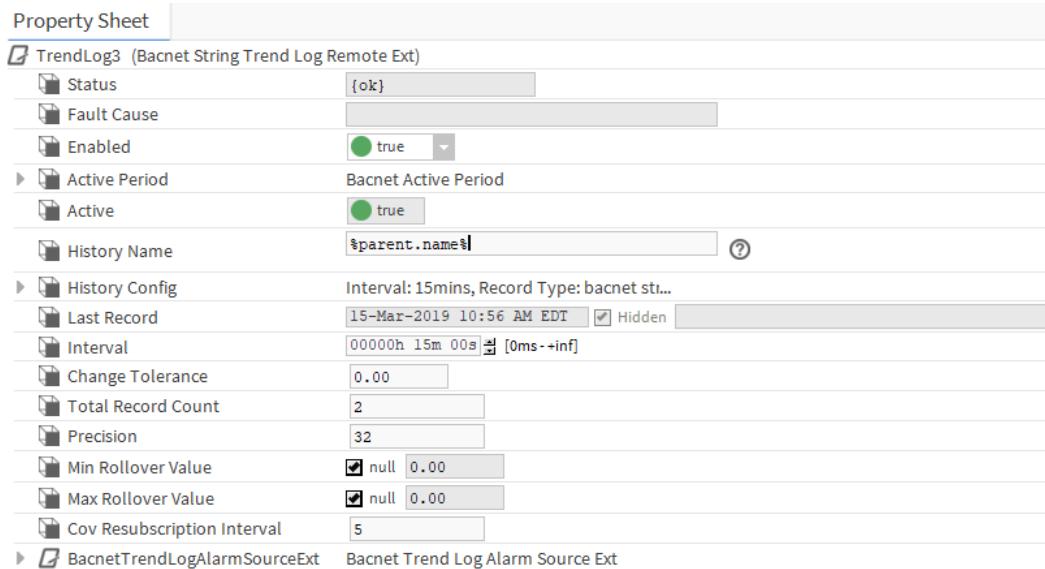
Bacnet String Trend Log Remote Ext (bacnet-BacnetStringTrendLogRemoteExt)

This extension is created dynamically for a remote BACnet device. It creates a history for the StatusString out value of a component using COV or a defined interval.

You can export the history created by this extension as a fully-compliant BACnet trend log object.

NOTE: This single extension applies to both COV and Interval data collection and may be used in place of the deprecated COV and Interval Trend Log Extensions for this data type.

Figure 92 Bacnet String Trend Log Remote Ext properties



In addition to the common properties (Status, Enabled and Fault Cause), this component has these properties.

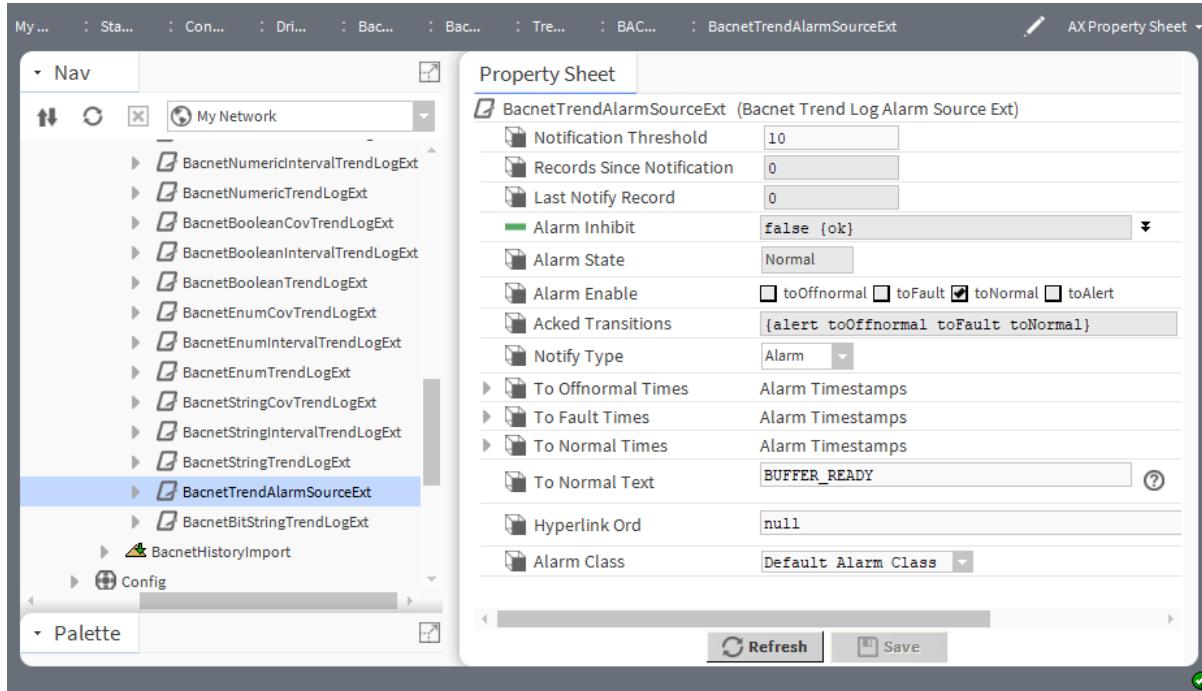
| Property | Value | Description |
|---------------------------|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Active Period, Start Time | Date and Time selectors | Configures when to start the log. |
| Active Period, Stop Time | Date and Time selectors | Configures when to stop the log. |
| Active | read-only | Indicates the current state of the trend log based on the Active Period . <code>true</code> indicates the log is active. <code>false</code> indicates the log is no longer active because the current time is outside the Active Period . |
| History Name | BFormat string (defaults to %parent.name%) | Defines a pattern for deriving the name of the history created by this extension. It can be static text or a simple pattern that allows the actual history name to be derived from the context. The default value (%parent.name%) sets the history name in the Id to the name of the parent of this extension. Changes to this property cause the format to be reapplied and the history Id in the history config is to be updated. |
| History Config | additional properties | Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties. |
| Last record | read-only | Stores a copy of the most recent record successfully appended to the history for this extension. |
| Interval | hours, minutes, seconds | Controls data collection behavior for this Trend Log component. If the Interval value is set to 0, the driver collects trends when a change of value (COV) on the parent point occurs. The subscription lifetime is equal to twice (2x) the value of the COV Resubscription Interval property. If the Interval value is set to any number greater than 0, the driver collects trend data at the specified time interval. |
| Change Tolerance | text | Specifies the minimum change in value that must occur before the system writes a record to the database. |
| Total Record Count | read-only | Indicates the total number of records collected by the Trend Log object since creation. When the value of Total Record Count reaches its maximum possible value, the next value it takes is zero. |
| Precision | 32 bit (default), 64 bit | Selects 32 bit or 64 bit options for the history data logging. The 64 bit option allows for higher level of precision but consumes more memory. |
| Min Rollover Value | number (defaults to null) | Defines the starting point for calculations for cumulative logging after a rollover. Rollover occurs after a running total maximum value is reached. Select the null option if a Min Rollover Value is unknown. |

| Property | Value | Description |
|-----------------------------|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Max Rollover Value | number (defaults to null) | Defines a maximum value for calculations when a rollover is detected by the history logging process. Using this parameter and the Min Rollover Value parameter helps you avoid getting negative numbers when you are logging total data, such as energy usage. |
| Cov Resubscription Interval | time range (defaults to 5 minutes) | Specifies the length of time that the extension waits before initiating a re-subscription request. |

Bacnet Trend Log Alarm Source Ext (bacnet-BacnetTrendLogAlarmSourceExt)

This component defines the intrinsic alarming/notification for a server-side Trend Log object exposed to BACnet. If needed, you paste it as a child of any of the BACnet trend log extensions under the source component.

Figure 93 Bacnet Trend Alarm Sourc Ext properties



This component is in the **bacnet** palette, under the **Trending→BacnetLogExtensions** folder.

| Property | Value | Description |
|----------------------------|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Notification Threshold | number (defaults to 10) | Indicates the value that once exceeded by the number of records since the last buffer-ready event was raised, a new buffer-ready event is raised. |
| Records Since Notification | read-only | Reports the number of records generated since notification. |
| Last Notify Record | read-only | Reports the last notification record generated. |
| Alarm Inhibit | check-mark (defaults to the | When null is enabled, reports the incoming value from the device. You cannot change this value. |

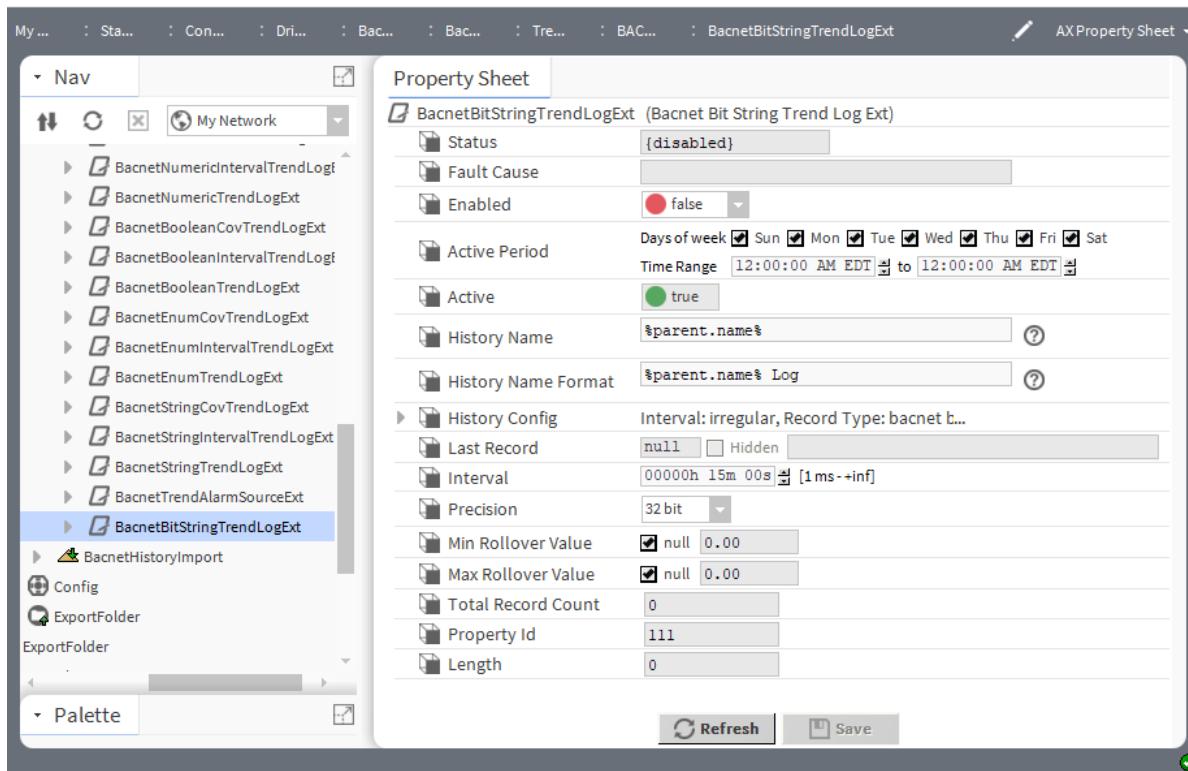
| Property | Value | Description |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | incoming value from the device) | To change this value, click the double-arrow to the right and remove the null check mark. |
| Alarm State | read-only | Displays the current state of the alarm: Normal, Low Limit, High Limit or Fault. |
| Alarm Enable | check boxes (defaults to toNormal) | Selects the type of occurrence that generates an alarm. <code>toOffnormal</code> generates an alarm when an entity transitions from normal to offnormal. <code>toFault</code> generates an alarm when an entity transitions from normal to fault. |
| Acked Transitions | read-only | Reports information about the occurrence. |
| Notify Type | drop-down list (defaults to Alarm) | Identifies what happens to trigger a record in the trend log. |
| To Offnormal Times | container | When a point transitions to an offnormal state, reports four pieces of information: Alarm Time displays when the alarm condition occurred. Ack Time displays the time that the alarm was acknowledged. Normal Time displays the time that the to-normal event occurred. Count displays the total number of offnormal events. |
| To Fault Times | container for four values Alarm Time defaults to null, which means that the event has not occurred Ack Time defaults to null Normal Time defaults to null Count defaults to zero (0) | When a point transitions to a fault state, reports four pieces of information: Alarm Time displays the time that the to-fault event occurred. Ack Time displays the time that the alarm was acknowledged. Normal Time displays the time that the To Normal event occurred. Count displays the total number of Offnormal events. |
| To Normal Times | container | Reports four pieces of information related to when a point transitioned to a normal state: Alarm Time : when the event generated the alarm. Ack Time : when the alarm was acknowledged. Normal Time : when the state of the point returned to normal. Count : the number of such alarms. |
| To Normal Text | text | Configures what displays when the component transitions to a normal status. When applicable, text entered for Fault Algorithm , High Limit Text and/or Low Limit Text may override this text. |

| Property | Value | Description |
|---------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Hyperlink Ord | chooser | Selects a chooser for locating an alarm source. |
| Alarm Class | drop-down list | Defines alarm routing options and priorities. Typical alarm classes include High, Medium and Low. An alarm class of Low might send an email message, while an alarm class of High might trigger a text message to the department manager. |

Bacnet Bit String Trend Log Ext (bacnet-BacnetBitStringTrendLogExt)

This component provides a bit string trend log extension.

Figure 94 Bacnet Bit String Trend Log Ext properties



To access these properties, expand **Config→Drivers→BacnetNetwork→BacnetDevice→Trend Logs→BacnetLogExtensions** and double-click the **BacnetBitStringTrendLogExt**.

In addition to the common properties (Status, Fault Cause and Enabled), this component has these properties.

| Property | Value | Description |
|---------------------------------|-----------------------------------------------|--------------------------------------------------------------------------|
| Active Period, Days of the Week | check boxes (all default to enabled) | Specifies the days of the week to include. |
| Active Period, Time Range | from and to times (both default to 12 AM EDT) | Configures when the time period starts and ends. |
| Active | read-only | Indicates the current state of the trend log based on the Active Period. |

| Property | Value | Description |
|---------------------|----------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | true indicates the log is active. false indicates the log is no longer active because the current time is outside the Active Period . |
| History Name | BFormat string (defaults to %parent.name%) | Defines a pattern for deriving the name of the history created by this extension. It can be static text or a simple pattern that allows the actual history name to be derived from the context. The default value (%parent.name%) sets the history name in the Id to the name of the parent of this extension. Changes to this property cause the format to be reapplied and the history Id in the history config is to be updated. |
| History Name Format | BFormat string plus text (defaults to %parent.name% Log) | Specifies the name of the trend log using a BFormat string. |
| History Config | additional properties | Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties. |
| Last record | read-only | Stores a copy of the most recent record successfully appended to the history for this extension. |
| Interval | hours, minutes, seconds | Controls data collection behavior for this Trend Log component. If the Interval value is set to 0, the driver collects trends when a change of value (COV) on the parent point occurs. The subscription lifetime is equal to twice (2x) the value of the COV Resubscription Interval property. If the Interval value is set to any number greater than 0, the driver collects trend data at the specified time interval. |
| Precision | 32 bit (default), 64 bit | Selects 32 bit or 64 bit options for the history data logging. The 64 bit option allows for higher level of precision but consumes more memory. |
| Min Rollover Value | number (defaults to null) | Defines the starting point for calculations for cumulative logging after a rollover. Rollover occurs after a running total maximum value is reached. Select the null option if a Min Rollover Value is unknown. |
| Max Rollover Value | number (defaults to null) | Defines a maximum value for calculations when a rollover is detected by the history logging process. Using this parameter and the Min Rollover Value parameter helps you avoid getting negative numbers when you are logging total data, such as energy usage. |
| Total Record Count | read-only | Indicates the total number of records collected by the Trend Log object since creation. When the value of Total Record Count reaches its maximum possible value, the next value it takes is zero. |

| Property | Value | Description |
|-------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Property Id | read-only | Specifies a number in the Recipient List for the station. This integer from the BACnet Notification Class object of the sending device, must match this ID in the Alarms device extension of the receiving device. |
| Length | read-only | |

config

This component functions as the container for any config-type objects, which represent individual BACnet objects in the device.

ConfigFolder

This component implements a folder under a BacnetConfigDeviceExt (Config).

Typically, you add such folders using the **New Folder** button in the **Bacnet Config Manager** view of the Config container. Each BacnetConfigFolder has its own **Bacnet Config Manager** view.

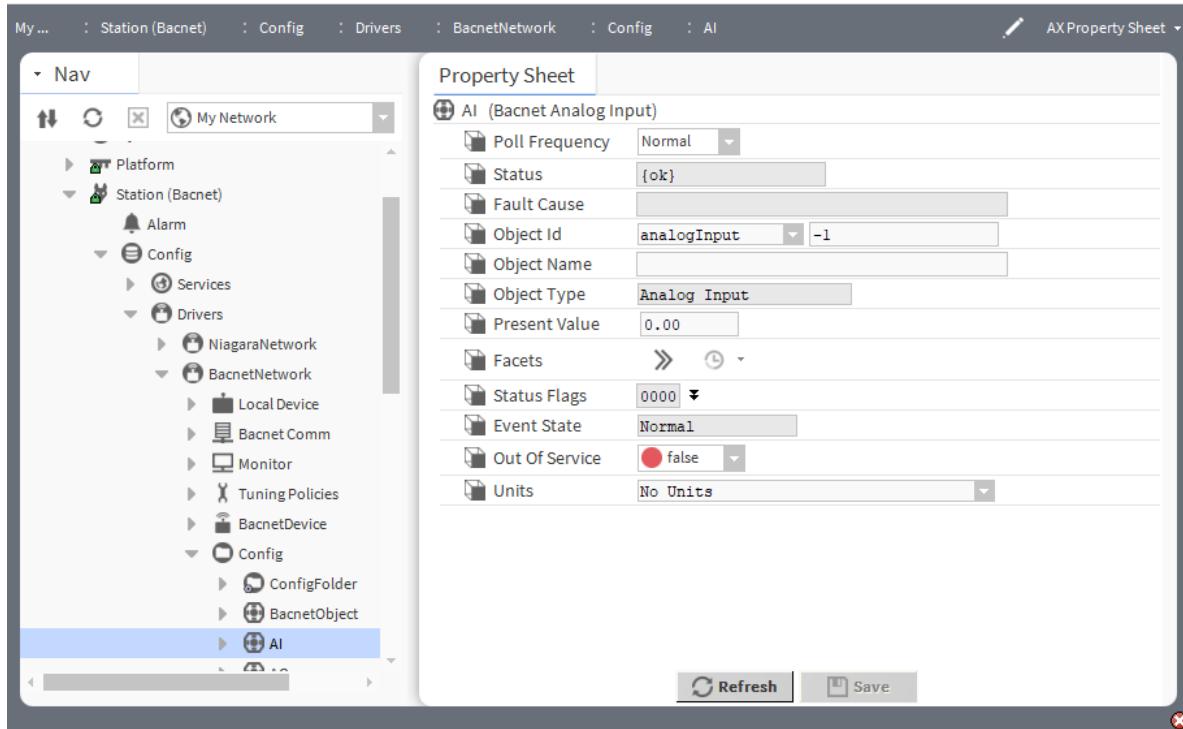
BacnetObject

This general purpose config object provides properties of a BACnet object type that is not included in the config folder of the **bacnet** palette, or if a BACnet OWS Supervisor or BACnet AWS Supervisor, not included in the config folders of the **bacnetOws** and **bacnetAws** palette.

Config, AI (Bacnet Analog Input)

This component is a Config object that represents a BACnet Analog Input object in its entirety.

Figure 95 Analog Input properties



To access these properties, copy an **AI** component from the **bacnet** palette (**Config** folder) to the **Bacnet-Device→Config** folder and double-click the component.

In addition to the common properties (Status and Fault Cause), this component has these properties.

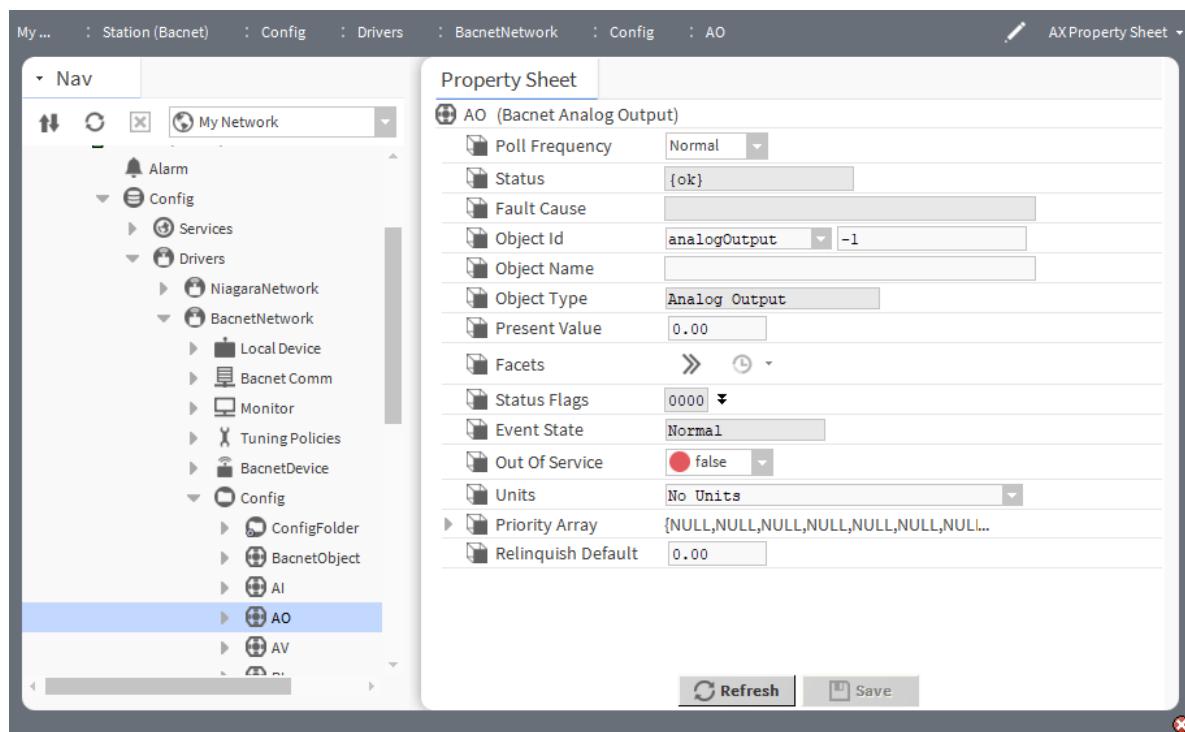
| Property | Value | Description |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork 's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name . name . name . name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |
| Present Value (config group) | number to two decimal places | Represents a BACnet list containing a non-indexed sequence of objects of a particular BACnet data type. |
| Facets | Config window or read-only property | Determine how values are formatted for display depending on the context and the type of data. Examples include engineering units and decimal precision for numeric types, and descriptive value (state) text for boolean and enum types. With the exception of proxy points (with possible defined device facets), point facets do not affect how the framework processes the point's value. Besides control points, various other components have facets too. For example, many kitControl and schedule components have facets. Details about point facets apply to these components too, unless especially noted. You access facets by clicking an Edit button or a chevron >>. Both open an Edit Facets window. |
| Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Event State | read-only | Reports the status of the event. |

| Property | Value | Description |
|----------------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Out of Service | true (default) or false | Affects only the BACnet exposure and access to the source component. If the source (exported) component is a proxy point, the equivalent Enabled property in its ProxyExt is not affected. true writes the point's current value to the Present Value property. Often, a remote BACnet client sets (or clears) this Out of Service state. You can set this property to false if needed. |
| Units | read-only | Reports the unit of measure for the entity. |

Config, AO (Bacnet Analog Output)

This component is a Config object that represents a BACnet Analog Output object in its entirety.

Figure 96 Analog Output properties



To access these properties, copy an **AO** component from the **bacnet** palette (**Config** folder) to the **Bacnet-Device→Config** folder and double-click the component.

In addition to the common properties (Status and Fault Cause), this component has these properties.

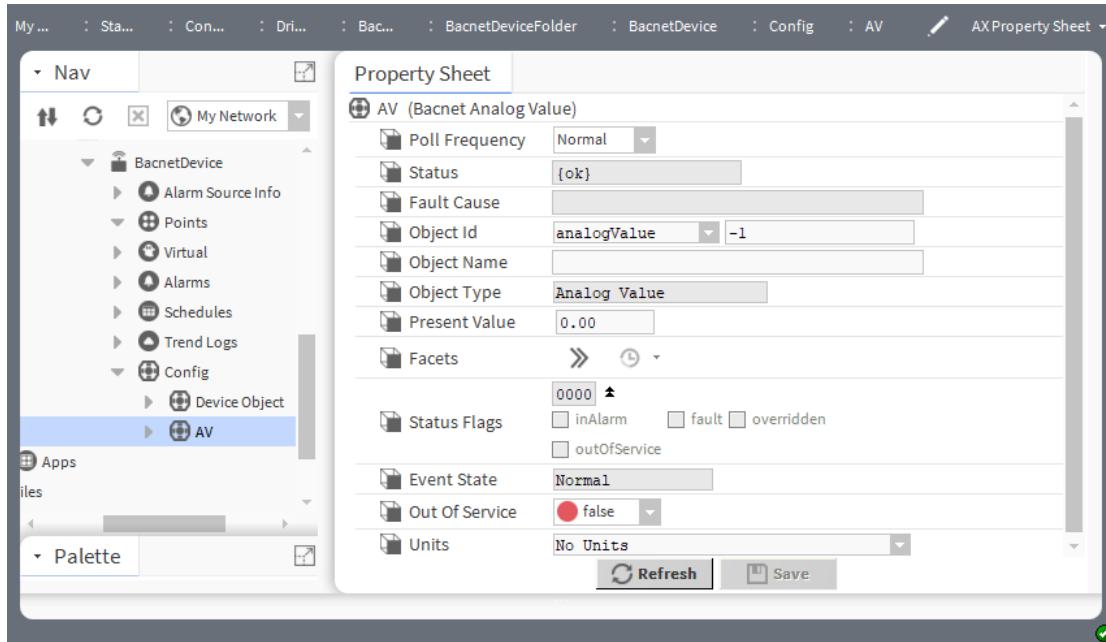
| Property | Value | Description |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |
| Present Value (config group) | | Represents a BACnet list containing a non-indexed sequence of objects of a particular BACnet data type. |
| Facets | Config window or read-only property | Determine how values are formatted for display depending on the context and the type of data. Examples include engineering units and decimal precision for numeric types, and descriptive value (state) text for boolean and enum types. With the exception of proxy points (with possible defined device facets), point facets do not affect how the framework processes the point's value. Besides control points, various other components have facets too. For example, many kitControl and schedule components have facets. Details about point facets apply to these components too, unless especially noted. You access facets by clicking an Edit button or a chevron >>. Both open an Edit Facets window. |
| Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Event State | read-only | Reports the status of the event. |
| Out of Service | true (default) or false | Affects only the BACnet exposure and access to the source component. If the source (exported) component is a proxy point, the equivalent Enabled property in its ProxyExt is not affected. |

| Property | Value | Description |
|--------------------|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | true writes the point's current value to the Present Value property. Often, a remote BACnet client sets (or clears) this Out of Service state. You can set this property to <code>false</code> if needed. |
| Units | read-only | Reports the unit of measure for the entity. |
| Priority Array | list of values | Sets up an array of object identifiers that identify priority values. |
| Relinquish Default | number to two decimal places | Sets up a number to serve as the Present Value default. |

Config, AV (Bacnet Analog Value)

This component is a Config object that represents a BACnet Analog Value object in its entirety.

Figure 97 AV properties



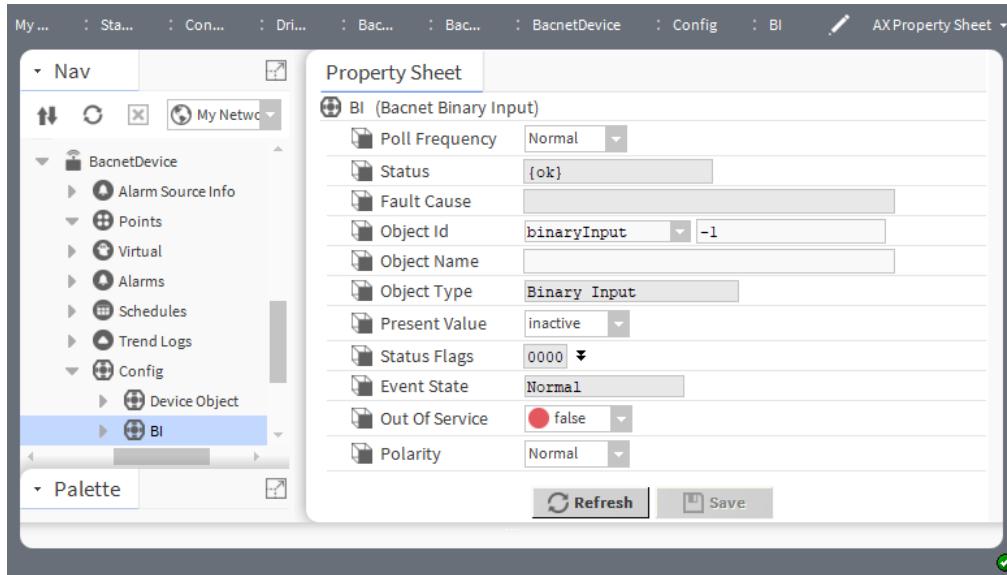
To access these properties, copy an **AV** component from the **bacnet** palette (**Config** folder) to the **Bacnet-Device→Config** folder and double-click the component.

| Property | Value | Description |
|----------------|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork 's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For |

| Property | Value | Description |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | <p>Identifies a BACnet object. This name should be unique within this specific device.</p> <p>For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device.</p> <p>For a proxy point, this is the default name in the station.</p> <p>You can shorten or edit Object Name, however, each Object Name must be unique among all server descriptors under the Export Table.</p> |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |
| Present Value (config group) | list | Represents a BACnet list containing a non-indexed sequence of objects of a particular BACnet data type. |
| Facets | Config window or read-only property | <p>Determine how values are formatted for display depending on the context and the type of data. Examples include engineering units and decimal precision for numeric types, and descriptive value (state) text for boolean and enum types.</p> <p>With the exception of proxy points (with possible defined device facets), point facets do not affect how the framework processes the point's value.</p> <p>Besides control points, various other components have facets too. For example, many kitControl and schedule components have facets. Details about point facets apply to these components too, unless especially noted.</p> <p>You access facets by clicking an Edit button or a chevron >>. Both open an Edit Facets window.</p> |
| Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Event State | read-only | Reports the status of the event. |
| Out of Service | true (default) or false | <p>Affects only the BACnet exposure and access to the source component. If the source (exported) component is a proxy point, the equivalent Enabled property in its ProxyExt is not affected.</p> <p>true writes the point's current value to the Present Value property. Often, a remote BACnet client sets (or clears) this Out of Service state.</p> <p>You can set this property to false if needed.</p> |
| Units | read-only | Reports the unit of measure for the entity. |

Config, BI (Bacnet Binary Input)

This component is a Config object that represents a BACnet Binary Input object in its entirety.

Figure 98 BI properties

To access these properties, copy an **BI** component from the **bacnet** palette (**Config** folder) to the **Bacnet-Device→Config** folder and double-click the component.

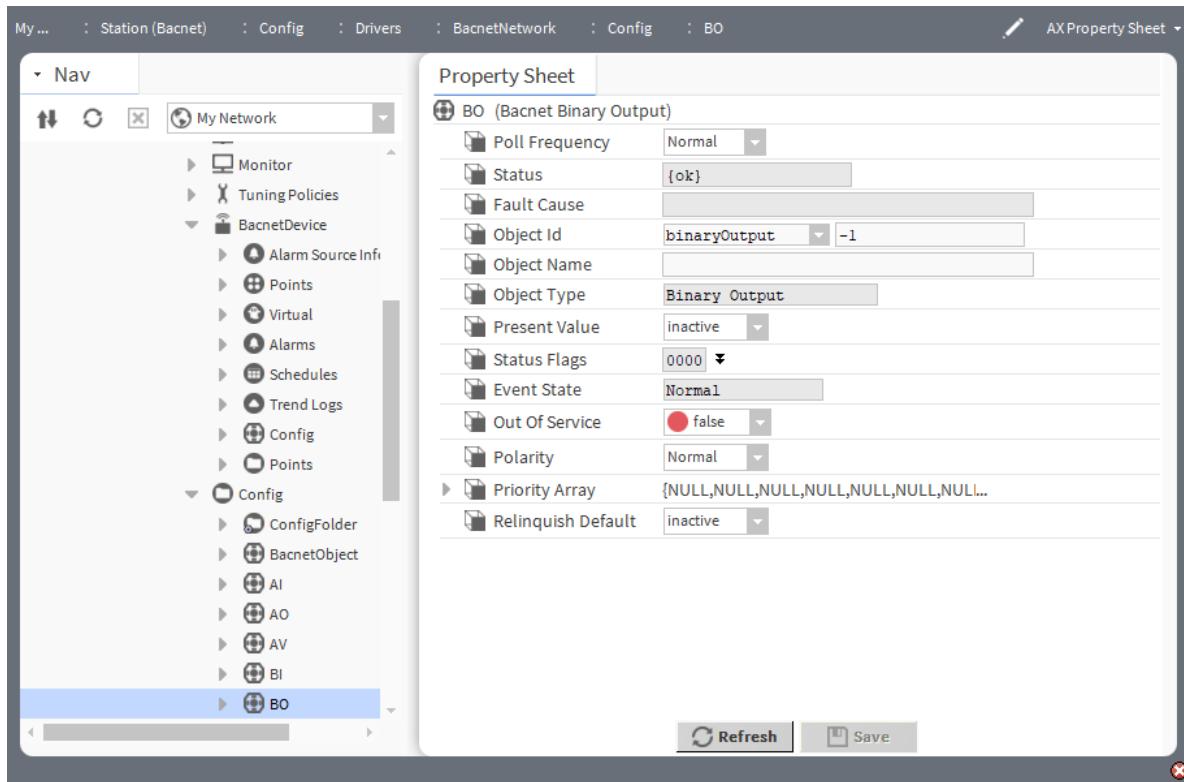
In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |
| Present Value (config group) | list | Represents a BACnet list containing a non-indexed sequence of objects of a particular BACnet data type. |

| Property | Value | Description |
|----------------|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Facets | Config window or read-only property | <p>Determine how values are formatted for display depending on the context and the type of data. Examples include engineering units and decimal precision for numeric types, and descriptive value (state) text for boolean and enum types.</p> <p>With the exception of proxy points (with possible defined device facets), point facets do not affect how the framework processes the point's value.</p> <p>Besides control points, various other components have facets too. For example, many <code>kitControl</code> and schedule components have facets. Details about point facets apply to these components too, unless especially noted.</p> <p>You access facets by clicking an Edit button or a chevron >>. Both open an Edit Facets window.</p> |
| Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Event State | read-only | Reports the status of the event. |
| Out of Service | true (default) or false | <p>Affects only the BACnet exposure and access to the source component. If the source (exported) component is a proxy point, the equivalent <code>Enabled</code> property in its <code>ProxyExt</code> is not affected.</p> <p>true writes the point's current value to the <code>Present Value</code> property. Often, a remote BACnet client sets (or clears) this Out of Service state.</p> <p>You can set this property to <code>false</code> if needed.</p> |
| Polarity | drop-down list | Identifies the polarity related to an input. Values are: Normal and Reverse. |

Config, BO (Bacnet Binary Output)

This component is a Config object that represents a BACnet Binary Output object in its entirety.

Figure 99 Binary Output properties

To access these properties, copy an **BO** component from the **bacnet** palette (**Config** folder) to the **Bacnet-Device→Config** folder and double-click the component.

In addition to the common properties (Status and Fault Cause), this component has these properties.

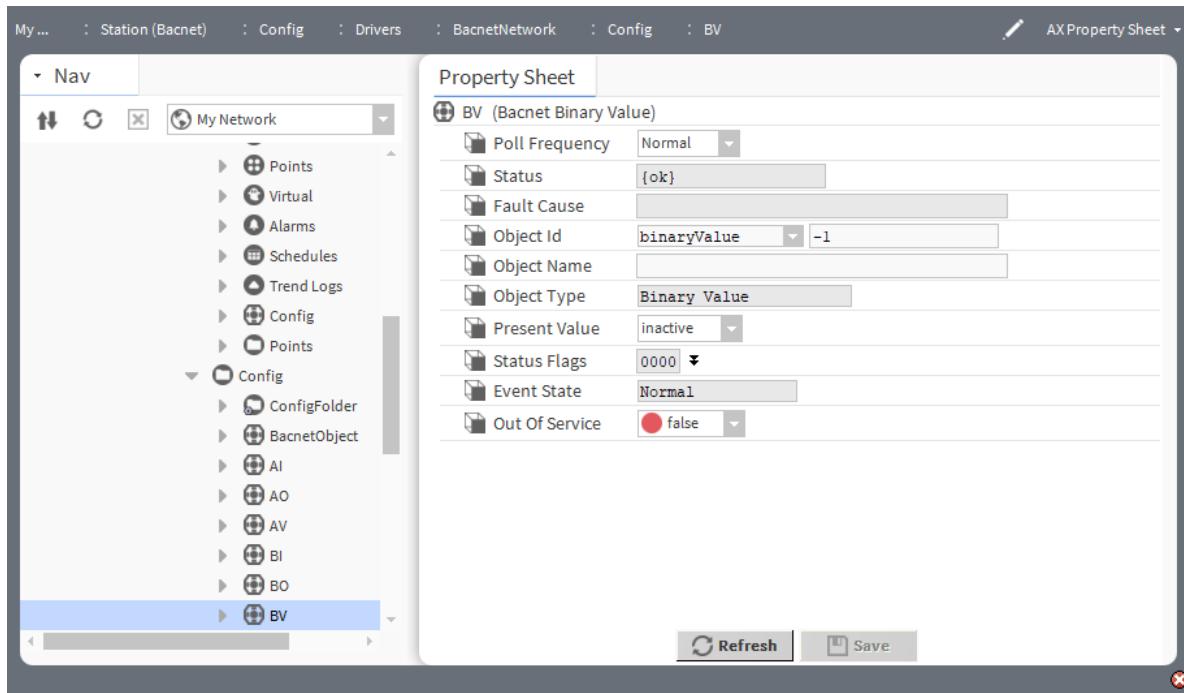
| Property | Value | Description |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |

| Property | Value | Description |
|------------------------------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Object Type | drop-down list of BACnet objects | Reports the type of object. |
| Present Value (config group) | list | Represents a BACnet list containing a non-indexed sequence of objects of a particular BACnet data type. |
| Facets | Config window or read-only property | <p>Determine how values are formatted for display depending on the context and the type of data. Examples include engineering units and decimal precision for numeric types, and descriptive value (state) text for boolean and enum types.</p> <p>With the exception of proxy points (with possible defined device facets), point facets do not affect how the framework processes the point's value.</p> <p>Besides control points, various other components have facets too. For example, many <code>kitControl</code> and schedule components have facets. Details about point facets apply to these components too, unless especially noted.</p> <p>You access facets by clicking an Edit button or a chevron >>. Both open an Edit Facets window.</p> |
| Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Event State | read-only | Reports the status of the event. |
| Out of Service | true (default) or false | <p>Affects only the BACnet exposure and access to the source component. If the source (exported) component is a proxy point, the equivalent <code>Enabled</code> property in its <code>ProxyExt</code> is not affected.</p> <p>true writes the point's current value to the <code>Present Value</code> property. Often, a remote BACnet client sets (or clears) this Out of Service state.</p> <p>You can set this property to false if needed.</p> |
| Polarity | | Identifies the polarity related to an input. Values are: Normal and Reverse. |
| Priority Array | number | Sets up an array of object identifiers that identify priority values. |
| Relinquish Default | number to two decimal places | Sets up a number to serve as the <code>Present Value</code> default. |

Config, BV (Bacnet Binary Value)

This component is a Config object that represents a BACnet Binary Value object in its entirety.

Figure 100 Binary Value properties



To access these properties, copy an **BV** component from the **bacnet** palette (**Config** folder) to the **Bacnet-Device→Config** folder and double-click the component.

In addition to the common properties (Status and Fault Cause), this component has these properties.

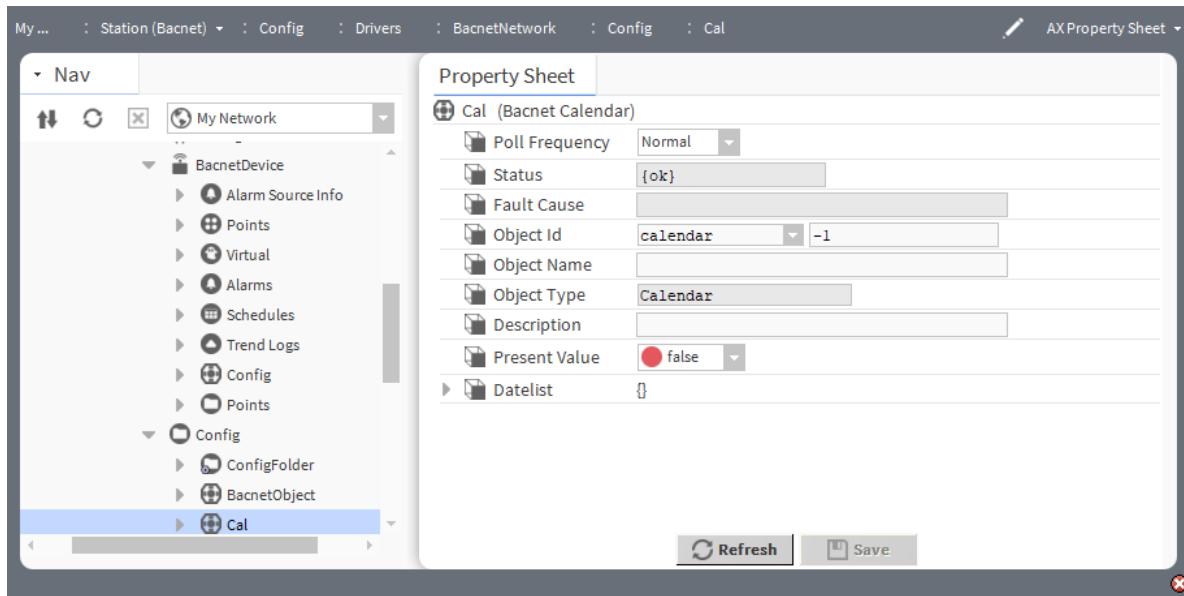
| Property | Value | Description |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |

| Property | Value | Description |
|------------------------------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Present Value (config group) | list | Represents a BACnet list containing a non-indexed sequence of objects of a particular BACnet data type. |
| Facets | Config window or read-only property | <p>Determine how values are formatted for display depending on the context and the type of data. Examples include engineering units and decimal precision for numeric types, and descriptive value (state) text for boolean and enum types.</p> <p>With the exception of proxy points (with possible defined device facets), point facets do not affect how the framework processes the point's value.</p> <p>Besides control points, various other components have facets too. For example, many <code>kitControl</code> and schedule components have facets. Details about point facets apply to these components too, unless especially noted.</p> <p>You access facets by clicking an Edit button or a chevron >>. Both open an Edit Facets window.</p> |
| Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Event State | read-only | Reports the status of the event. |
| Out of Service | true (default) or false | <p>Affects only the BACnet exposure and access to the source component. If the source (exported) component is a proxy point, the equivalent <code>Enabled</code> property in its <code>ProxyExt</code> is not affected.</p> <p>true writes the point's current value to the <code>Present Value</code> property. Often, a remote BACnet client sets (or clears) this Out of Service state.</p> <p>You can set this property to false if needed.</p> |

Config, Cal (Bacnet Calendar)

This component is a Config object that represents a BACnet Calendar object. Included among its properties is a Datalist property that contains calendar entries.

Figure 101 Cal properties



To access these properties, copy a **Cal** component from the **bacnet** palette (**Config** folder) to the **BacnetDevice**→**Config** folder and double-click the component.

In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork 's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. |

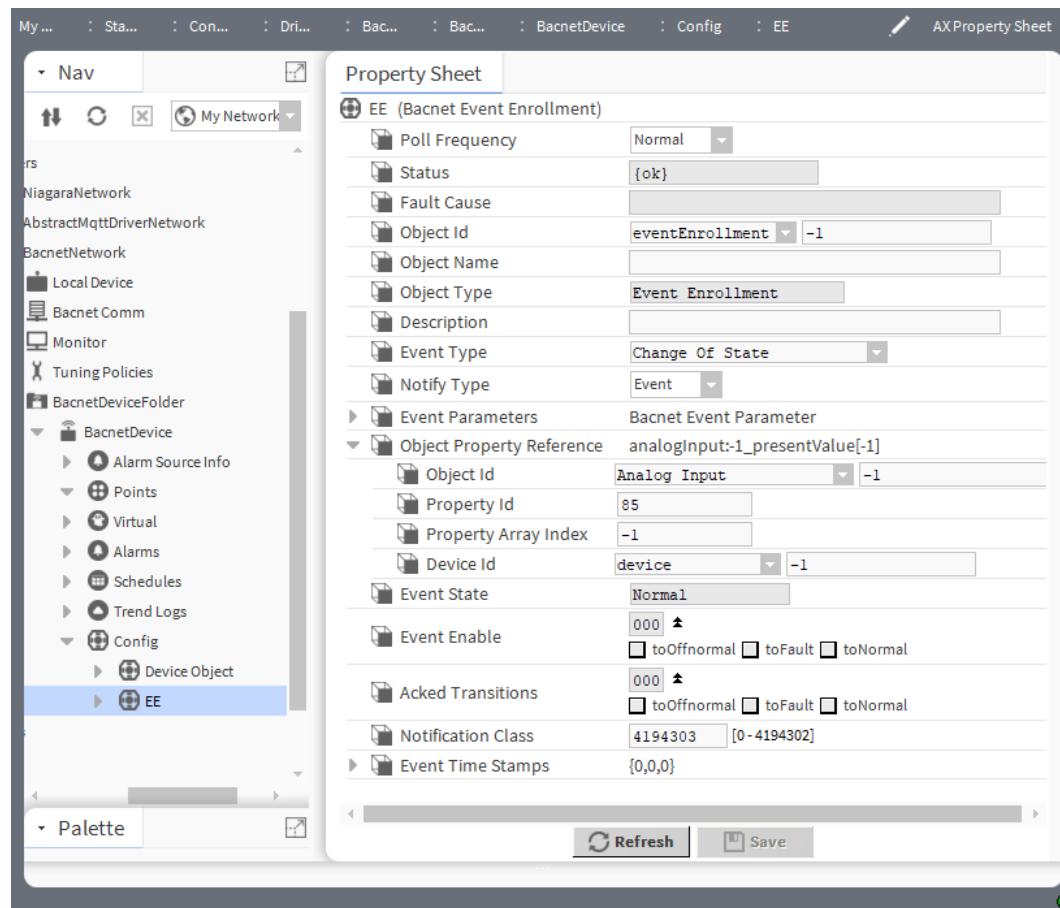
| Property | Value | Description |
|---------------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the Add and Edit window. |
| Present Value | true (default) or false | Indicates the current value of the calendar: true if the current date is in the Datalist and false if it is not. |
| DateList | additional properties | Provides a list of dates for the Bacnet Calendar. |

Config, EE (Bacnet Event Enrollment)

This component is a Config object that represents a BACnet Event Enrollment object.

The properties of this object describe an event that may be an error or alarm condition. They reference source object and property data, and specify notification class and device recipients. The component is located in the **bacnet** palette under the **Config** folder.

Figure 102 EE properties



To access these properties, copy an **EE** component from the **bacnet** palette (**Config** folder) to the **Bacnet-Device→Config** folder and double-click the component.

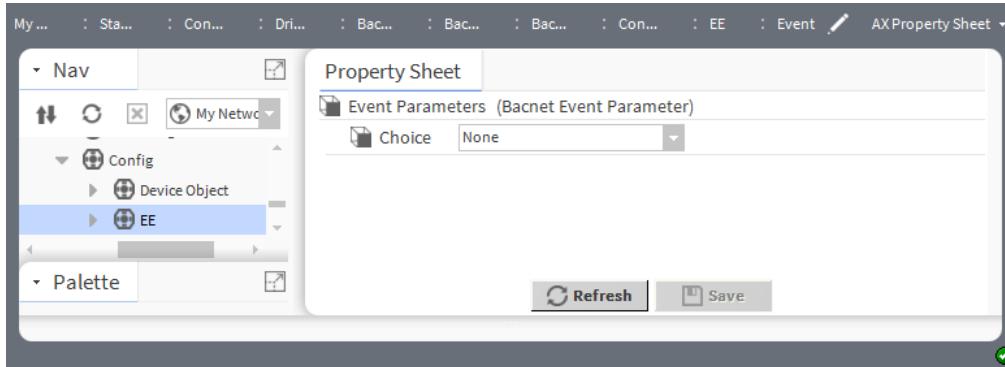
In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the Add and Edit window. |
| Event Type | read-only | Reports the type of event. |
| Notify Type | (optional) Alarm (default) or Event | Indicates if the notification is from an alarm or an event. This property applies if the source component has an alarm extension. |
| Event Parameters | additional properties | Displays the list of parameters which can be set for the event. |
| Object Property Reference | additional properties | Contains the properties related to the referenced object. |
| Event State | read-only | Reports the status of the event. |
| Event Enable | check boxes | Selects an event status: toOffnormal, toFault, toNormal. |
| Acked Transitions | check boxes | Selects an acknowledgment transition: toOffnormal, toFault, toNormal. |
| Notification Class | number (defaults to -1) | Specifies the notification class used for routing when handling and generating event notifications for this object. |
| Event Time Stamps | additional properties | Defines a BACnet array of elements. |

EE (Event Parameter)

This component provides a single property to configure for a calendar event: **Choice**.

Figure 103 Event Parameter property



To access these properties, copy an **EE** component from the **bacnet** palette (**Config** folder) to the **Bacnet-Device→Config** folder, expand the component and double-click **Event Parameters**.

| Property | Value | Description |
|----------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Choice | drop-down list | Selects among a list of event-related options: Buffer Ready Deprecated Change of Life Safety Extended Buffer Ready Unsigned Range Reserved Access Event Double Out Of Range Signed Out OF Range Unsigned Out Of Range Change Of Characterstring Change of Status Flags Change of Reliability |

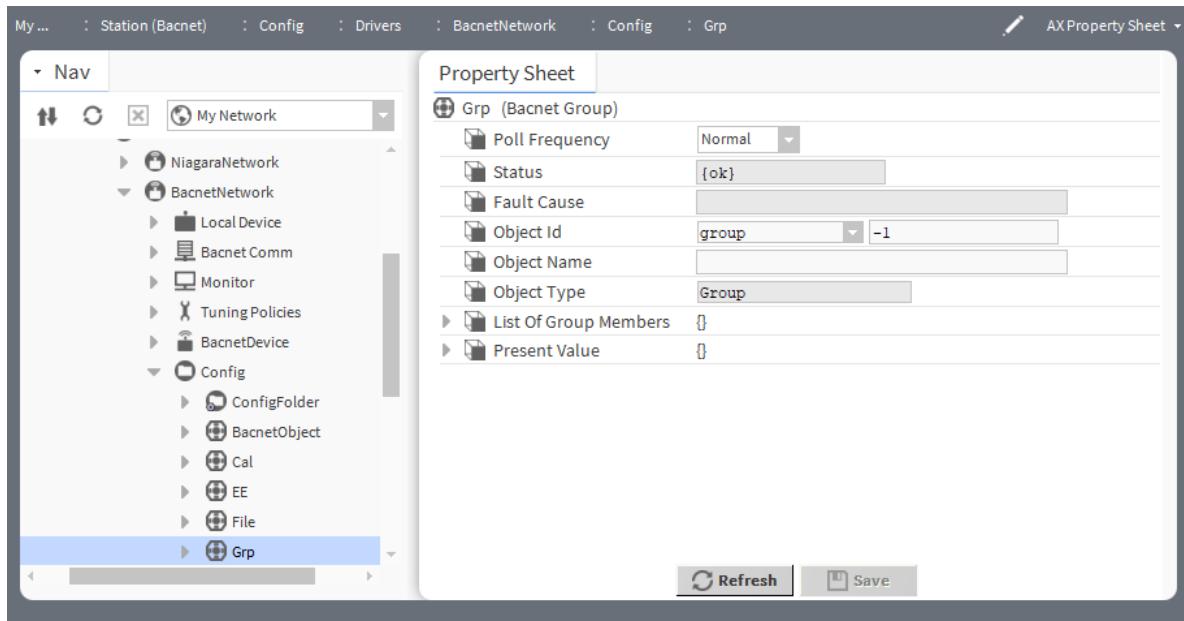
Config, File (Bacnet File)

This component is a Config object that represents a BACnet File object.

Config, Grp (Bacnet Group)

This component is a Config object that represents a BACnet Group object. A Group object provides access to multiple properties of multiple objects in a read single operation.

Figure 104 Bacnet Group properties



To access this property expand, station **Config**→**Drivers**→**BacnetNetwork**→**Config**→**Grp**.

In addition to the common properties (Status and Fault Cause), this component has these properties.

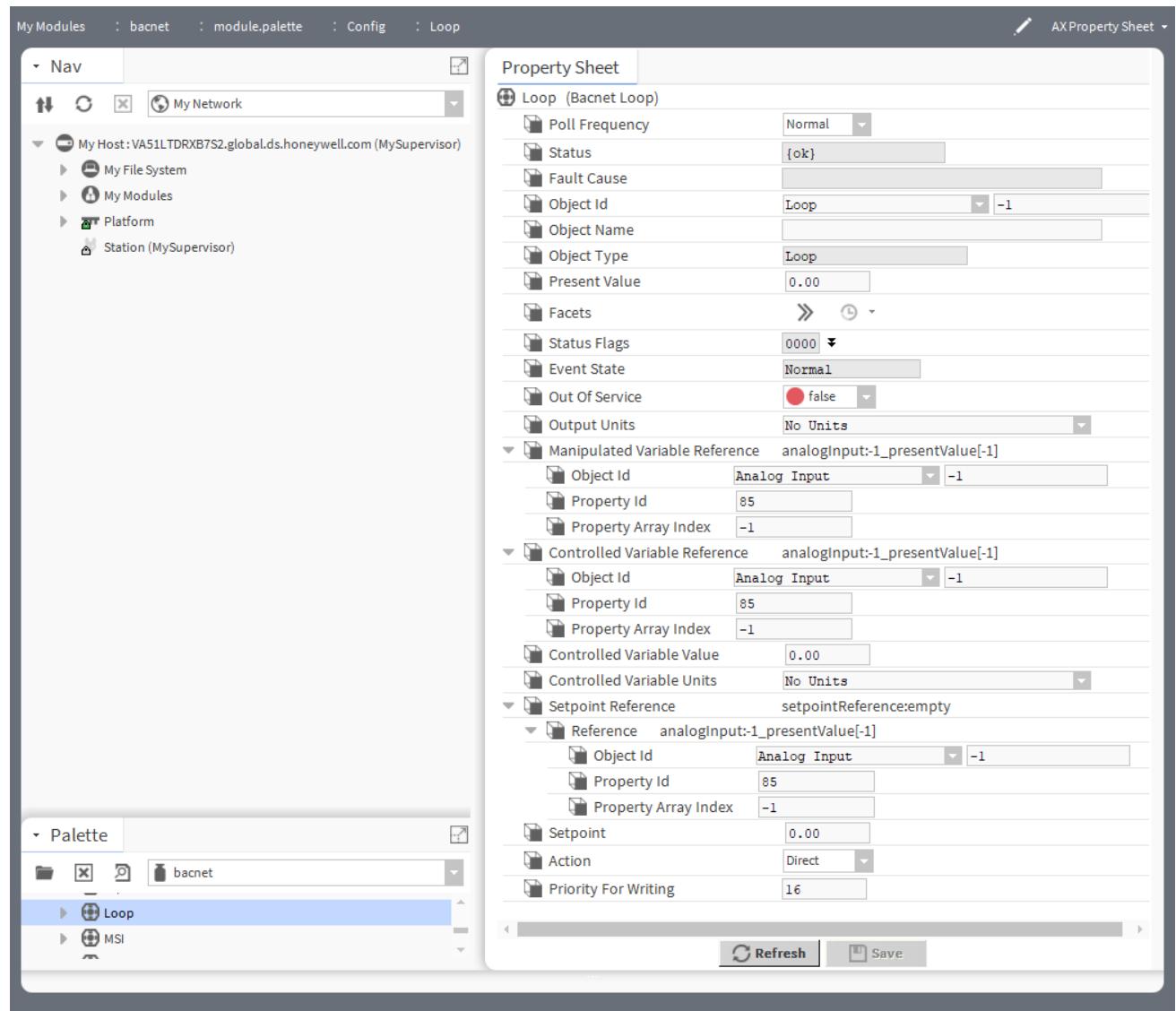
| Property | Value | Description |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork 's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |

| Property | Value | Description |
|------------------------------|-------|---------------------------------------------------------------------------------------------------------|
| List of Group Members | list | Represents a BACnet list containing a non-indexed sequence of objects of a particular BACnet data type. |
| Present Value (config group) | list | Represents a BACnet list containing a non-indexed sequence of objects of a particular BACnet data type. |

Config, Loop (Bacnet Loop)

This component is a Config object that represents a BACnet Loop object. A Loop object provides a standardized control loop implementation.

Figure 105 Loop properties



To access these properties, expand **BacnetDevice→Config** and double-click **Loop**.

In addition to the common properties (Status and Fault Cause), this component has these properties.

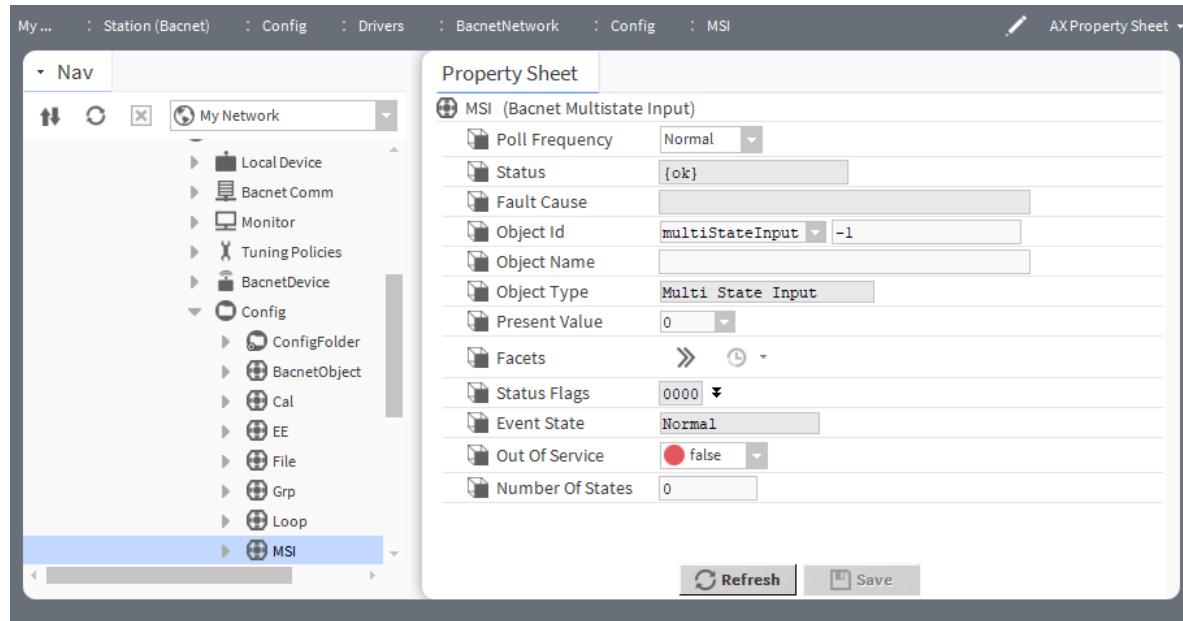
| Property | Value | Description |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |
| Present Value (config value) | list | Represents a BACnet list containing a non-indexed sequence of objects of a particular BACnet data type. |
| Facets | Config window or read-only property | Determine how values are formatted for display depending on the context and the type of data. Examples include engineering units and decimal precision for numeric types, and descriptive value (state) text for boolean and enum types. With the exception of proxy points (with possible defined device facets), point facets do not affect how the framework processes the point's value. Besides control points, various other components have facets too. For example, many kitControl and schedule components have facets. Details about point facets apply to these components too, unless especially noted. You access facets by clicking an Edit button or a chevron >>. Both open an Edit Facets window. |
| Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Event State | read-only | Reports the status of the event. |
| Out of Service | true (default) or false | Affects only the BACnet exposure and access to the source component. If the source (exported) component is a proxy point, the equivalent Enabled property in its ProxyExt is not affected. |

| Property | Value | Description |
|--------------------------------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | true writes the point's current value to the Present Value property. Often, a remote BACnet client sets (or clears) this Out of Service state. You can set this property to <code>false</code> if needed. |
| Output Units | drop-down list | Selects unit of measure for output. |
| Manipulated Variable Reference | additional properties | Configures variables. |
| Controlled Variable Reference | additional properties | Configures variables. |
| Setpoint Reference | additional properties | Configures setpoint references. Refer to Reference in this dictionary. |
| Setpoint | number to two decimal places | Configures the point's setpoint value. |
| Action | drop-down list (defaults to Direct) | Selects the type of action: Direct or Reverse. |
| Priority for Writing | numeric text values 1 through 16 | Reports schedule priority, which defaults to 16. |

Config, MSI (Bacnet Multistate Input)

This component is a Config object that represents a BACnet Multi-state Input object.

Figure 106 MSI properties



To access this property expand, **Station→Config→Drivers→BacnetNetwork→Config**, double-click **MSI**. In addition to the common properties (Status and Fault Cause), this component has these properties.

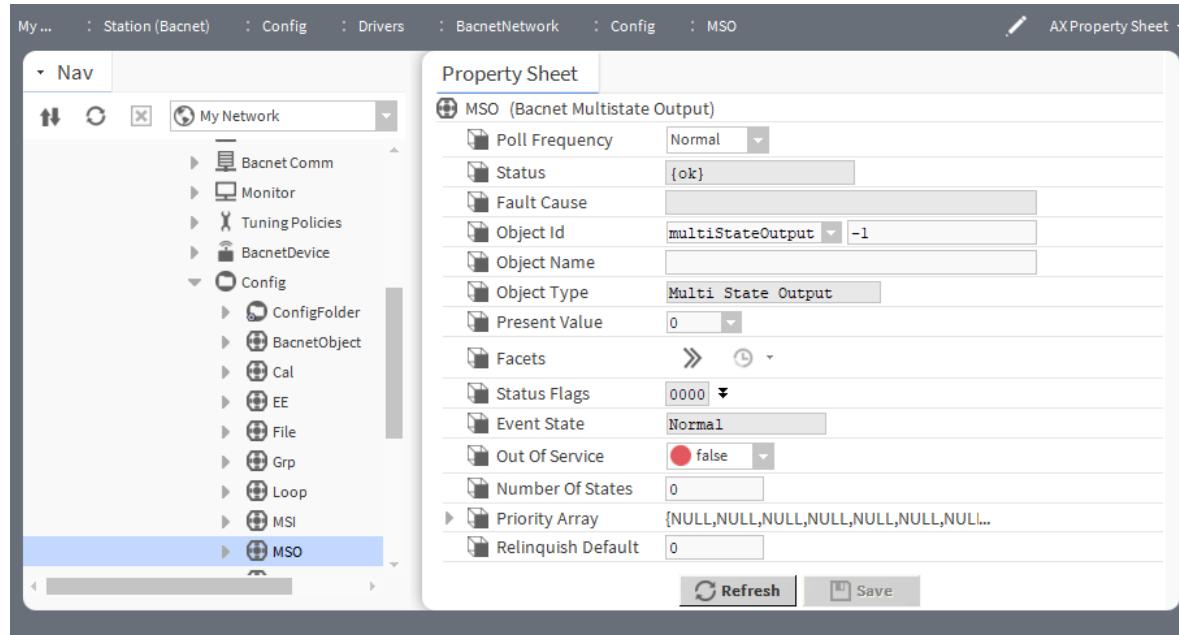
| Property | Value | Description |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |
| Present Value (config group) | list | Represents a BACnet list containing a non-indexed sequence of objects of a particular BACnet data type. |
| Facets | Config window or read-only property | Determine how values are formatted for display depending on the context and the type of data. Examples include engineering units and decimal precision for numeric types, and descriptive value (state) text for boolean and enum types. With the exception of proxy points (with possible defined device facets), point facets do not affect how the framework processes the point's value. Besides control points, various other components have facets too. For example, many kitControl and schedule components have facets. Details about point facets apply to these components too, unless especially noted. You access facets by clicking an Edit button or a chevron >>. Both open an Edit Facets window. |
| Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Event State | read-only | Reports the status of the event. |

| Property | Value | Description |
|------------------|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Out of Service | true (default) or false | Affects only the BACnet exposure and access to the source component. If the source (exported) component is a proxy point, the equivalent Enabled property in its ProxyExt is not affected. true writes the point's current value to the Present Value property. Often, a remote BACnet client sets (or clears) this Out of Service state. You can set this property to false if needed. |
| Number of States | number | Defines the number of states required by the Bacnet Multi-state Input component. |

Config, MSO (Bacnet Multistate Output)

This component is a Config object that represents a BACnet Multi-state Output object.

Figure 107 MSO properties



To access this property expand, **Station→Config→Drivers→BacnetNetwork→Config**, double-click **MSO**

In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|----------------|-----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork 's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For |

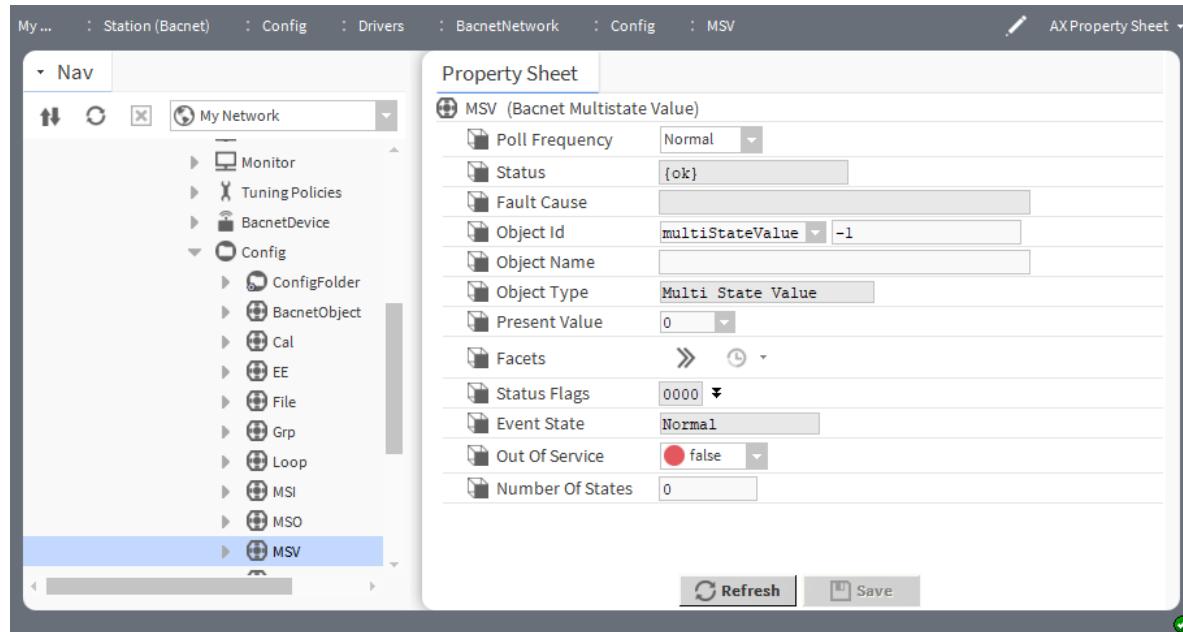
| Property | Value | Description |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | <p>Identifies a BACnet object. This name should be unique within this specific device.</p> <p>For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device.</p> <p>For a proxy point, this is the default name in the station.</p> <p>You can shorten or edit Object Name, however, each Object Name must be unique among all server descriptors under the Export Table.</p> |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |
| Present Value (config group) | list | Represents a BACnet list containing a non-indexed sequence of objects of a particular BACnet data type. |
| Facets | Config window or read-only property | <p>Determine how values are formatted for display depending on the context and the type of data. Examples include engineering units and decimal precision for numeric types, and descriptive value (state) text for boolean and enum types.</p> <p>With the exception of proxy points (with possible defined device facets), point facets do not affect how the framework processes the point's value.</p> <p>Besides control points, various other components have facets too. For example, many kitControl and schedule components have facets. Details about point facets apply to these components too, unless especially noted.</p> <p>You access facets by clicking an Edit button or a chevron >>. Both open an Edit Facets window.</p> |
| Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Event State | read-only | Reports the status of the event. |
| Out of Service | true (default) or false | <p>Affects only the BACnet exposure and access to the source component. If the source (exported) component is a proxy point, the equivalent Enabled property in its ProxyExt is not affected.</p> <p>true writes the point's current value to the Present Value property. Often, a remote BACnet client sets (or clears) this Out of Service state.</p> <p>You can set this property to false if needed.</p> |
| Number of States | number | Defines the number of states required by the Bacnet Multi-state Input component. |

| Property | Value | Description |
|--------------------|--------|-----------------------------------------------------------------------|
| Priority Array | number | Sets up an array of object identifiers that identify priority values. |
| Relinquish Default | number | Sets up a number to serve as the Present Value default. |

Config, MSV (Bacnet Multistate Value)

This component is a Config object that represents a BACnet Multi-state Value object.

Figure 108 MSV properties



To access this property expand, **Station→Config→Drivers→BacnetNetwork→Config**, double-click **MSV**

In addition to the common properties (Status and Fault Cause), this component has these properties.

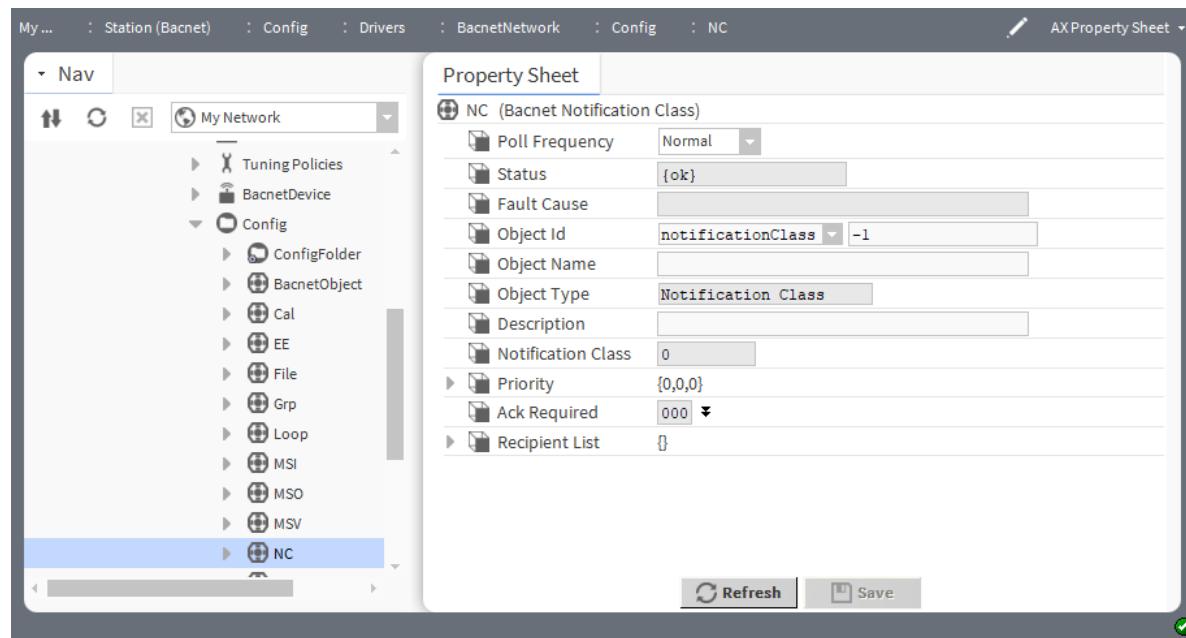
| Property | Value | Description |
|----------------|----------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name.name (default includes the entire component path in the station) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. |

| Property | Value | Description |
|------------------------------|------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | using period ("/") delimiters between parent.child levels) | For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |
| Present Value (config group) | list | Represents a BACnet list containing a non-indexed sequence of objects of a particular BACnet data type. |
| Facets | Config window or read-only property | <p>Determine how values are formatted for display depending on the context and the type of data. Examples include engineering units and decimal precision for numeric types, and descriptive value (state) text for boolean and enum types.</p> <p>With the exception of proxy points (with possible defined device facets), point facets do not affect how the framework processes the point's value.</p> <p>Besides control points, various other components have facets too. For example, many kitControl and schedule components have facets. Details about point facets apply to these components too, unless especially noted.</p> <p>You access facets by clicking an Edit button or a chevron >>. Both open an Edit Facets window.</p> |
| Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Event State | read-only | Reports the status of the event. |
| Out of Service | true (default) or false | <p>Affects only the BACnet exposure and access to the source component. If the source (exported) component is a proxy point, the equivalent Enabled property in its ProxyExt is not affected.</p> <p>true writes the point's current value to the Present Value property. Often, a remote BACnet client sets (or clears) this Out of Service state.</p> <p>You can set this property to false if needed.</p> |
| Number of States | number | Defines the number of states required by the Bacnet Multi-state Input component. |

Config, NC (Bacnet Notification Class)

This component is a Config object that represents a BACnet Notification Class object.

Figure 109 NC (bacnet Notification Class)



To access this property expand, **Station→Config→Drivers→BacnetNetwork→Config**, double-click NC

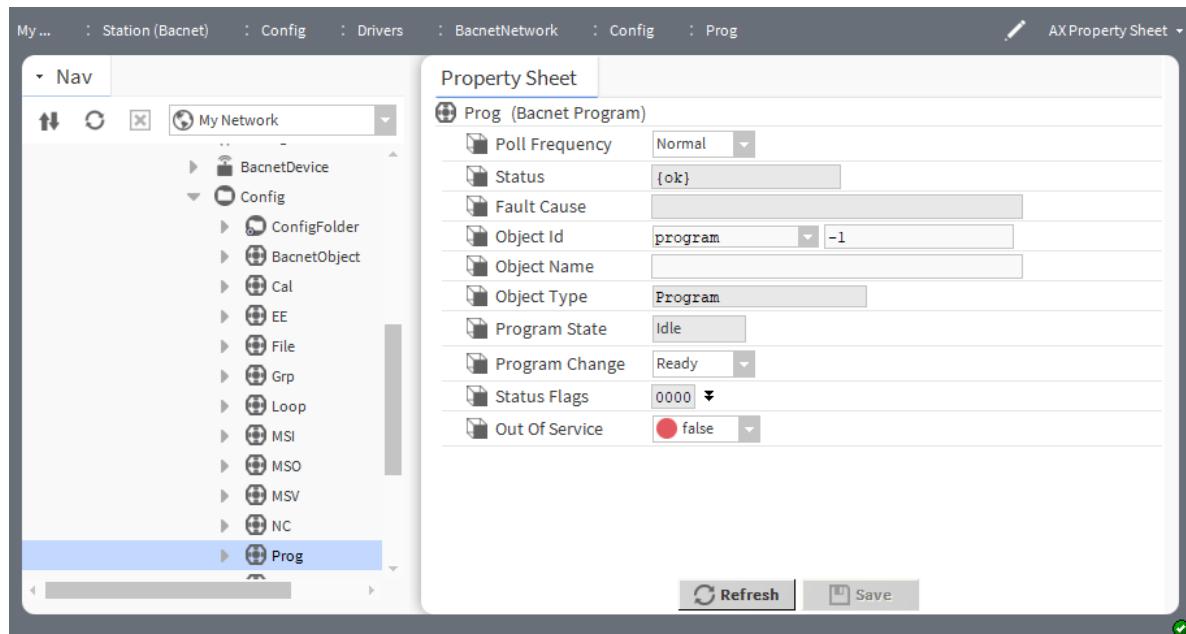
| Property | Value | Description |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. |

| Property | Value | Description |
|--------------------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the Add and Edit window. |
| Notification Class | number (defaults to -1) | Specifies the notification class used for routing when handling and generating event notifications for this object. |
| Priority | array | Sets up an array of object identifiers that identify priority values. |
| Ack Required | check boxes | Selects an acknowledgment option: toOffnormal, toFault, toNormal. |
| Recipient List | list | Lists recipients. |

Config, Prog (Bacnet Program)

This component is a Config object that represents a BACnet Program object. A Program object lets a program running in the device to be controlled, and reports the program's present status.

Figure 110 Prog (Bacnet Program) properties



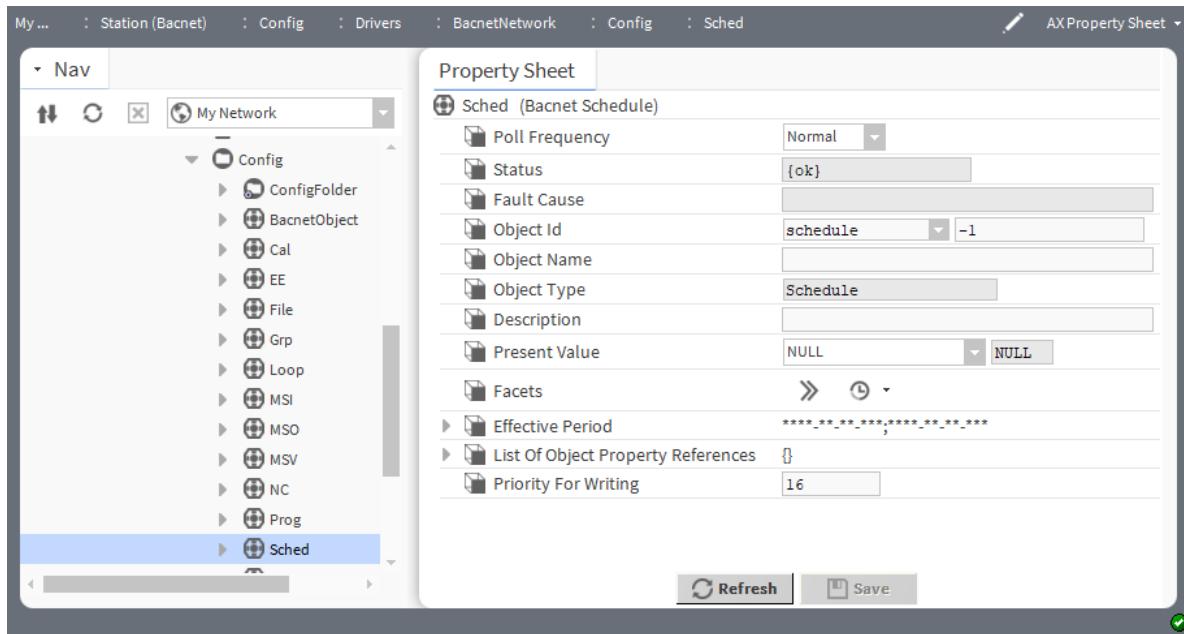
To access this property expand, **Station→Config→Drivers→BacnetNetwork→Config**, double-click **Prog**. In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |
| Program State | read-only | Reports the current state of the program. |
| Program Change | drop-down list | Selects a program transition. |
| Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Out of Service | true (default) or false | Affects only the BACnet exposure and access to the source component. If the source (exported) component is a proxy point, the equivalent Enabled property in its ProxyExt is not affected. true writes the point's current value to the Present Value property. Often, a remote BACnet client sets (or clears) this Out of Service state. You can set this property to false if needed. |

Config, Sched (Bacnet Schedule)

This component is a Config object that represents a BACnet Schedule object.

Figure 111 Sched properties



To access this property expand, **Config→Drivers→BacnetNetwork→Config**, double-click **Sched**

In addition to the common properties (Status and Fault Cause), this component has these properties.

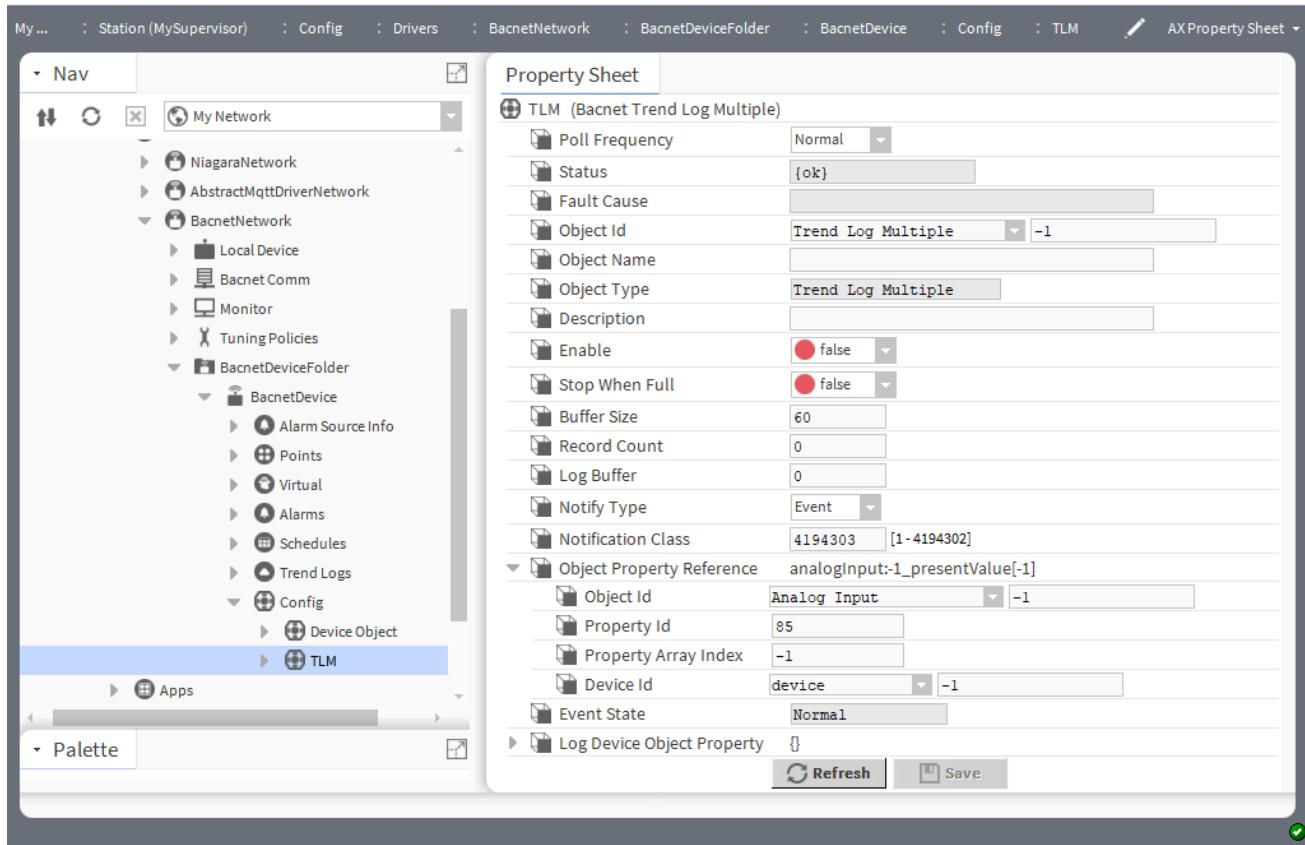
| Property | Value | Description |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number. For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name, however, each Object Name must be unique among all server descriptors under the Export Table. |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. |

| Property | Value | Description |
|-------------------------------------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the Add and Edit window. |
| Present Value (config group) | list | Represents a BACnet list containing a non-indexed sequence of objects of a particular BACnet data type. |
| Facets | Config window or read-only property | <p>Determine how values are formatted for display depending on the context and the type of data. Examples include engineering units and decimal precision for numeric types, and descriptive value (state) text for boolean and enum types.</p> <p>With the exception of proxy points (with possible defined device facets), point facets do not affect how the framework processes the point's value.</p> <p>Besides control points, various other components have facets too. For example, many <code>kitControl</code> and schedule components have facets. Details about point facets apply to these components too, unless especially noted.</p> <p>You access facets by clicking an Edit button or a chevron >>. Both open an Edit Facets window.</p> |
| Effective Period | additional properties | Sets up the beginning and end dates that identify when the schedule is in force. |
| List of Objects Property References | BACnet array | Sets up an array of object property references. |
| Priority for Writing | number | Specifies a number in the Recipient List for the station. This integer from the BACnet Notification Class object of the sending device, must match this ID in the Alarms device extension of the receiving device. |

Config, TLog (Bacnet Trend Log)

This component is a Config object that provides the configuration of a Trend Log object in a BACnet device (to access its log data, you must import that Trend Log under the **Trend Logs** extension of that BacnetDevice).

Figure 112 TLog properties



To access this property expand, **Station→Config→Drivers→BacnetNetwork→Config**, double-click **TLog**

In addition to the common properties (Status, Enable, and Fault Cause), this component has these properties.

| Property | Value | Description |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. |

| Property | Value | Description |
|----------------------------|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the Add and Edit window. |
| Stop When Full | true or false (default) | Controls what happens when the trend log exceeds its capacity. |
| Buffer Size | number | Configures the size of the trend log buffer. |
| Record Count | number | Configures the number of records to store in the trend log. |
| Log Buffer | number | Configures the size of the log buffer. |
| Notify Type | (optional) Alarm (default) or Event | Indicates if the notification is from an alarm or an event. This property applies if the source component has an alarm extension. |
| Notification Class | number (defaults to -1) | Specifies the notification class used for routing when handling and generating event notifications for this object. |
| Object Property Reference | additional properties | Contains the properties related to the referenced object. |
| Event State | read-only | Reports the status of the event. |
| Log Device Object Property | array | Sets up a BACnet array for trend log device objects. |

Config, TLM (Bacnet Trend Log Multiple)

This component is a Config object that represents a BACnet trend log multiple object that monitors and records values of one or more properties for one or more referenced objects, either in the same device as the trend log multiple object or in an external device.

Server

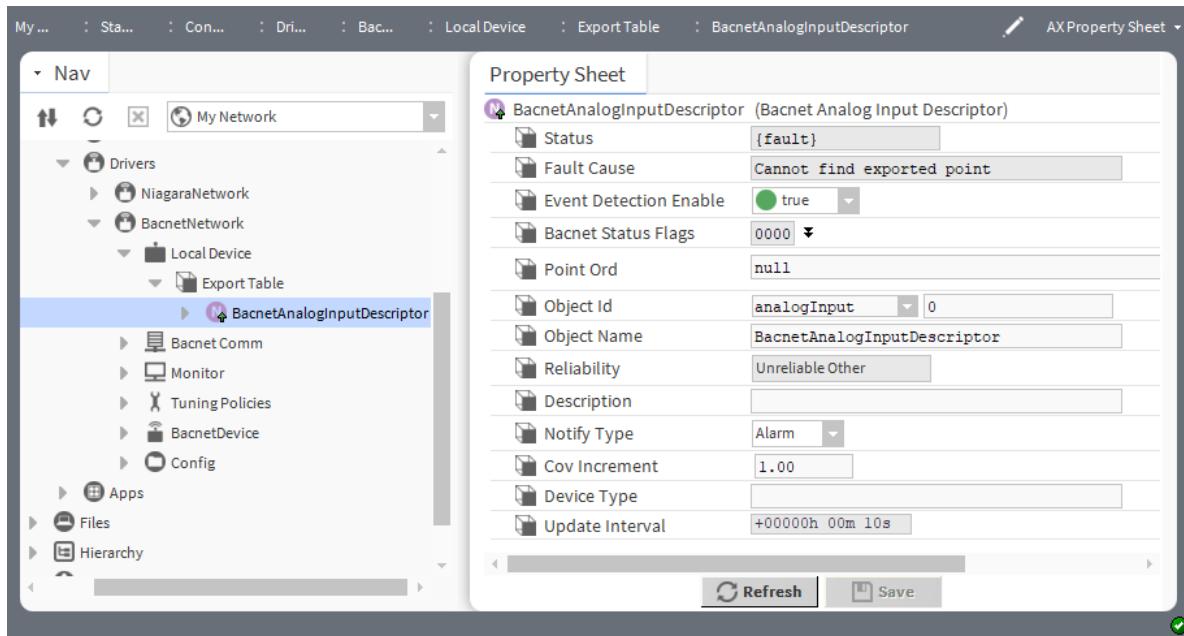
Server, **ServerAIDesc** in the palette exposes a numeric component as an analog input object.

Bacnet Analog Input Descriptor (ServerAIDesc)

This component (**Server**, **ServerAIDesc** in the palette) exposes a numeric component as an analog input object.

You view these properties when you associate an input descriptor with a BACnet object. To define this distributor, expand the BACnet network, **Local Device**→**Export Table** and discover objects or create a new object, then click the view drop-down list and select AX Property Sheet.

Figure 113 Bacnet Analog Input Descriptor properties



In addition to the common properties (Status and Fault Cause), this component has these properties.

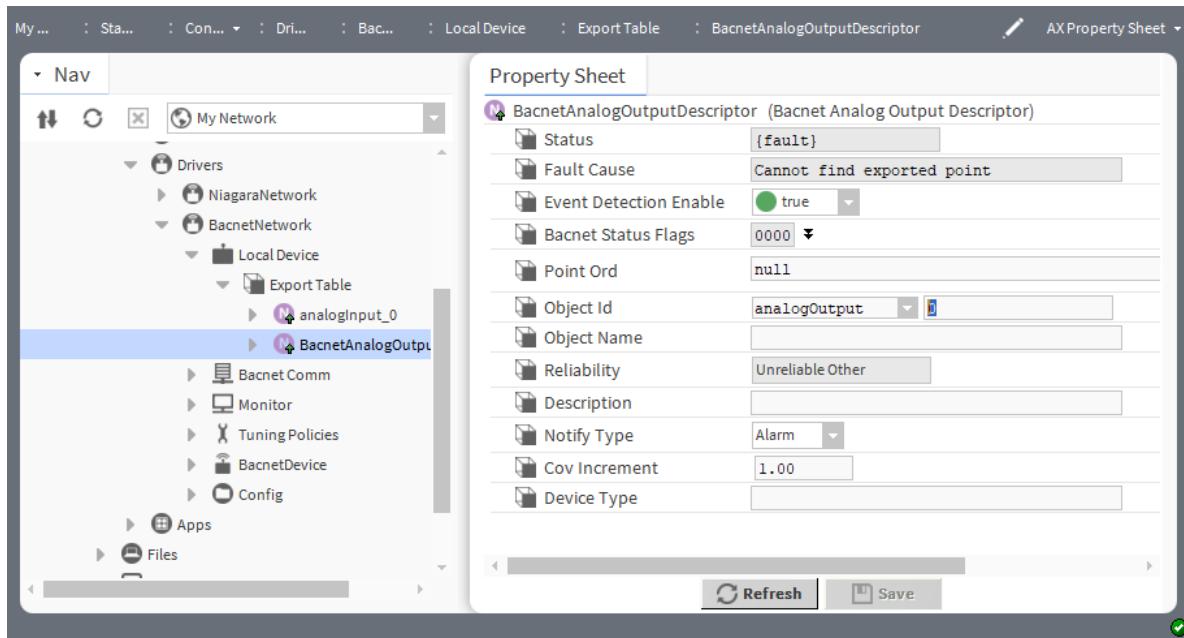
| Property | Value | Description |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Event Detection Enable | true (default) or false | Turns detection of an input event on (true) and off (false) for analog, binary and multi-state descriptors. |
| Bacnet Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Point Ord | ord (defaults to null) | Defines the ORD (address) for the point. |
| Object ID | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Reliability | read-only | Provides information when an error interrupts communication with a device. |

| Property | Value | Description |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | No Fault Detected indicates that things are OK. Unreliable Other indicates that the exported BACnet object appears in fault. This might happen, for example, if the source (exported) component is a proxy point under some other driver, and device communications are down or if the point has been disabled. In this case, the BACnet export descriptor retains a status of ok. |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the New and Edit windows. |
| Notify Type | (optional) Alarm (default) or Event | Indicates if the notification is from an alarm or an event. This property applies if the source component has an alarm extension. |
| Cov Increment | number only for analog object types (Analog Input, Analog Output, Analog Value, Analog Value Priority, Loop, defaults to 1.00) | Specifies the minimum COV required before a COVNotification is issued to subscriber BACnet COV-clients. |
| Device Type | (optional) text only for some object types (Analog Input, Analog Output, Binary Input, Binary Output, Multi-state Input, Multi-state Output) | Defines the type of physical input or output. This might apply mostly to exported Ndio points where each point corresponds to a physical device. |
| Update Interval | read-only | Reports the frequency of updates. |

Bacnet Analog Output Descriptor (ServerAODesc)

This component (**Server**, **ServerAODesc**) exposes a NumericWritable as a BACnet analog output object. You use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components. The properties are the same as those documented for the Bacnet Analog Input Descriptor.

Figure 114 Bacnet Analog Output Descriptor properties



In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Event Detection Enable | true (default) or false | Turns detection of an input event on (true) and off (false) for analog, binary and multi-state descriptors. |
| Bacnet Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Point Ord | ord (defaults to null) | Defines the ORD (address) for the point. |
| Object ID | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Reliability | read-only | Provides information when an error interrupts communication with a device. |

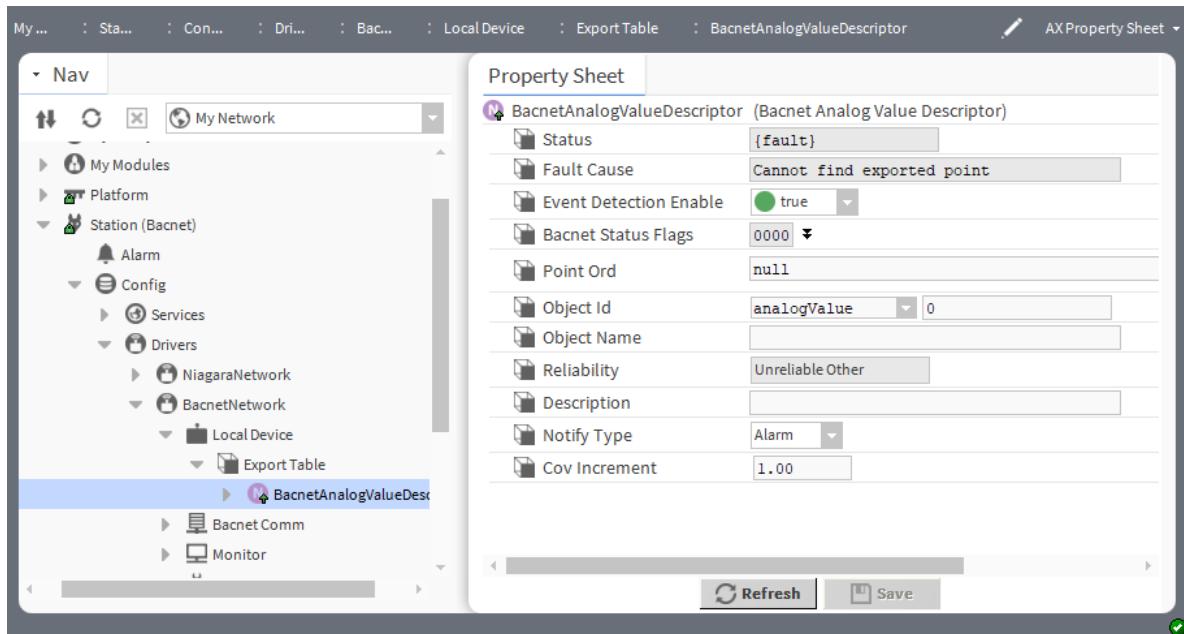
| Property | Value | Description |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | No Fault Detected indicates that things are OK. Unreliable Other indicates that the exported BACnet object appears in fault. This might happen, for example, if the source (exported) component is a proxy point under some other driver, and device communications are down or if the point has been disabled. In this case, the BACnet export descriptor retains a status of ok. |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the New and Edit windows. |
| Notify Type | (optional) Alarm (default) or Event | Indicates if the notification is from an alarm or an event. This property applies if the source component has an alarm extension. |
| Cov Increment | number only for analog object types (Analog Input, Analog Output, Analog Value, Analog Value Priority, Loop, defaults to 1.00) | Specifies the minimum COV required before a COVNotification is issued to subscriber BACnet COV-clients. |
| Device Type | (optional) text only for some object types (Analog Input, Analog Output, Binary Input, Binary Output, Multi-state Input, Multi-state Output) | Defines the type of physical input or output. This might apply mostly to exported Ndio points where each point corresponds to a physical device. |

Bacnet Analog Value Descriptor (ServerAVDesc)

This component (**Server**, **ServerAVDesc** in the palette) exposes a numeric-type component as a BACnet Analog Value object (non-commandable).

You use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

Figure 115 Bacnet Analog Value Descriptor properties



In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Event Detection Enable | true (default) or false | Turns detection of an input event on (true) and off (false) for analog, binary and multi-state descriptors. |
| Bacnet Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Point Ord | ord (defaults to null) | Defines the ORD (address) for the point. |
| Object ID | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Reliability | read-only | Provides information when an error interrupts communication with a device. |

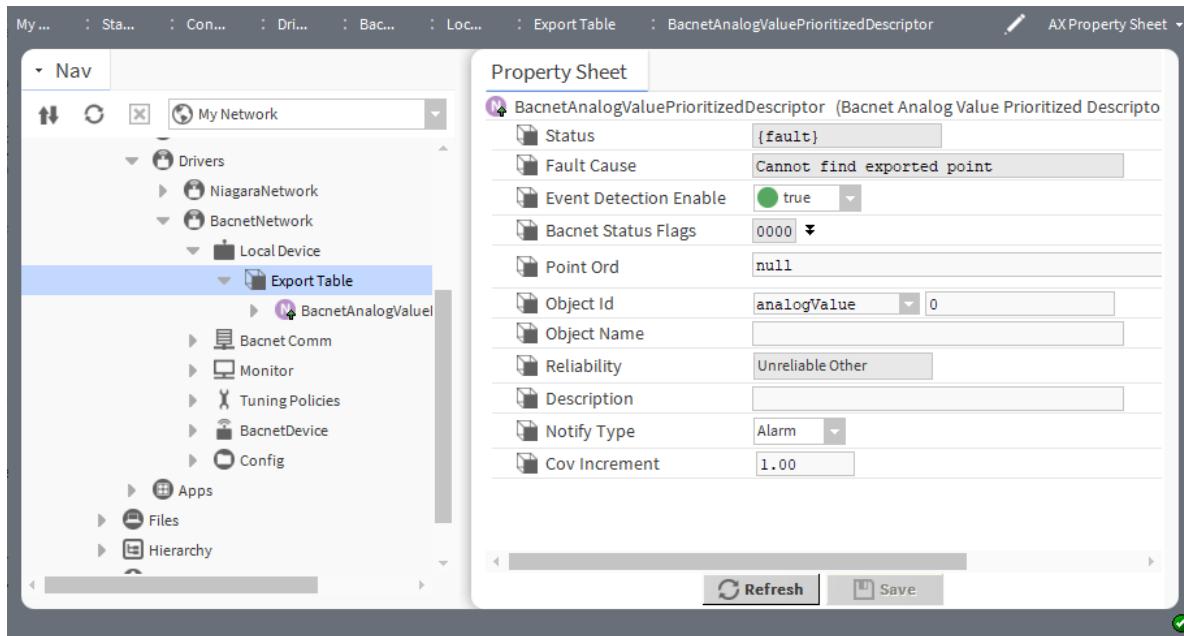
| Property | Value | Description |
|---------------|--------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | No Fault Detected indicates that things are OK. Unreliable Other indicates that the exported BACnet object appears in fault. This might happen, for example, if the source (exported) component is a proxy point under some other driver, and device communications are down or if the point has been disabled. In this case, the BACnet export descriptor retains a status of ok. |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the New and Edit windows. |
| Notify Type | (optional) Alarm (default) or Event | Indicates if the notification is from an alarm or an event. This property applies if the source component has an alarm extension. |
| Cov Increment | number only for analog object types (Analog Input, Analog Output, Analog Value, Analog Value Priority, Loop, defaults to 1.00) | Specifies the minimum COV required before a COVNotification is issued to subscriber BACnet COV-clients. |

Bacnet Analog Value Prioritized Descriptor (**ServerAVPDesc**)

This component (**Server**, **ServerAVDesc** in the palette) exposes a NumericWritable as a writable (commandable) Bacnet Analog Value object.

You use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

Figure 116 Bacnet Analog Value Prioritized Descriptor properties



In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Event Detection Enable | true (default) or false | Turns detection of an input event on (true) and off (false) for analog, binary and multi-state descriptors. |
| Bacnet Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Point Ord | ord (defaults to null) | Defines the ORD (address) for the point. |
| Object ID | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name... (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Reliability | read-only | Provides information when an error interrupts communication with a device. |

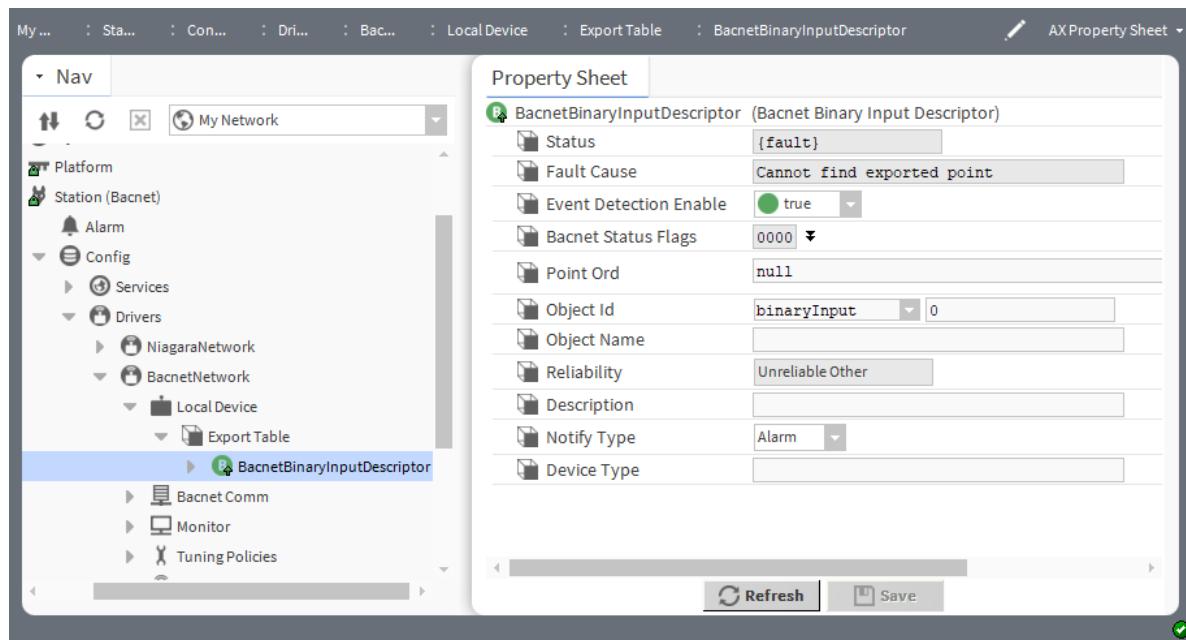
| Property | Value | Description |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | No Fault Detected indicates that things are OK. Unreliable Other indicates that the exported BACnet object appears in fault. This might happen, for example, if the source (exported) component is a proxy point under some other driver, and device communications are down or if the point has been disabled. In this case, the BACnet export descriptor retains a status of ok. |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the New and Edit windows. |
| Notify Type | (optional) Alarm (default) or Event | Indicates if the notification is from an alarm or an event. This property applies if the source component has an alarm extension. |
| Cov Increment | number only for analog object types (Analog Input, Analog Output, Analog Value, Analog Value Priority, Loop, defaults to 1.00) | Specifies the minimum COV required before a COVNotification is issued to subscriber BACnet COV-clients. |
| Device Type | (optional) text only for some object types (Analog Input, Analog Output, Binary Input, Binary Output, Multi-state Input, Multi-state Output) | Defines the type of physical input or output. This might apply mostly to exported Ndio points where each point corresponds to a physical device. |

Bacnet Binary Input Descriptor (ServerBIDesc)

This component exposes a boolean-type component as a BACnet Binary Input object. You use the **Bacnet Export Manager** view of the Bacnet **Export Table** to add, edit, delete, and access exported components.

You use the **Bacnet Export Manager** view of the Bacnet **Export Table** to add, edit, delete, and access exported components.

Figure 117 Binary Input Descriptor properties



In addition to the common properties (Status and Fault Cause), this component has these properties.

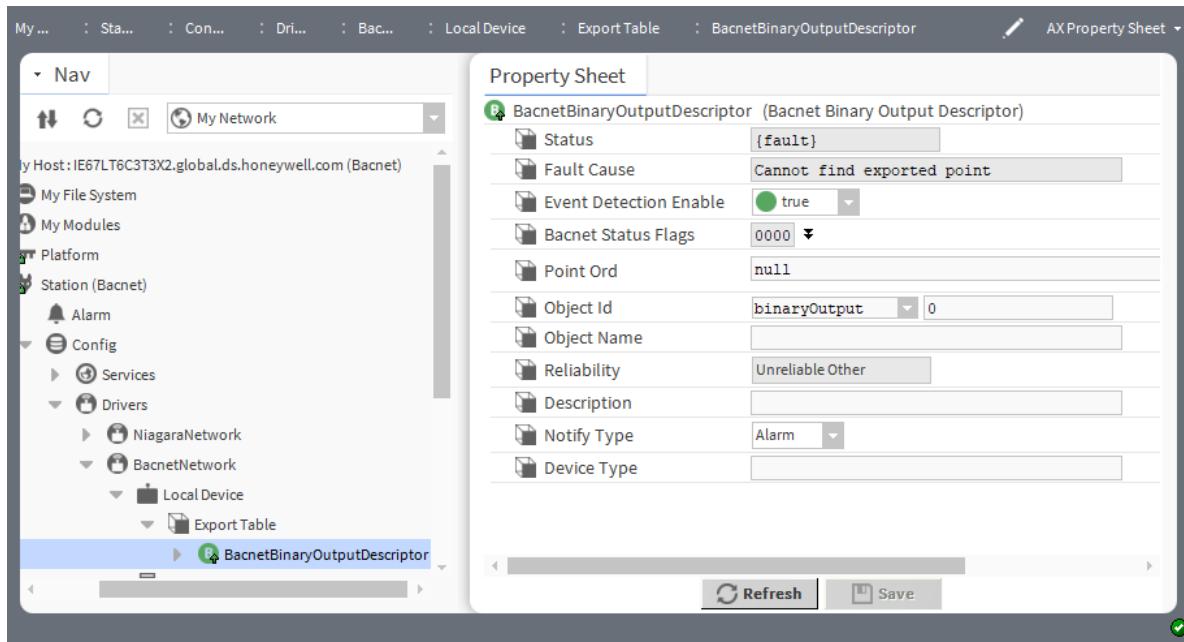
| Property | Value | Description |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Event Detection Enable | true (default) or false | Turns detection of an input event on (true) and off (false) for analog, binary and multi-state descriptors. |
| Bacnet Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Point Ord | ord (defaults to null) | Defines the ORD (address) for the point. |
| Object ID | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Reliability | read-only | Provides information when an error interrupts communication with a device. |

| Property | Value | Description |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | No Fault Detected indicates that things are OK. Unreliable Other indicates that the exported BACnet object appears in fault. This might happen, for example, if the source (exported) component is a proxy point under some other driver, and device communications are down or if the point has been disabled. In this case, the BACnet export descriptor retains a status of ok. |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the New and Edit windows. |
| Notify Type | (optional) Alarm (default) or Event | Indicates if the notification is from an alarm or an event. This property applies if the source component has an alarm extension. |
| Device Type | (optional) text only for some object types (Analog Input, Analog Output, Binary Input, Binary Output, Multi-state Input, Multi-state Output) | Defines the type of physical input or output. This might apply mostly to exported Ndio points where each point corresponds to a physical device. |

Bacnet Binary Output Descriptor (ServerBODesc)

This component exposes a BooleanWritable as a BACnet Binary Output object. You use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

Figure 118 Binary Output Descriptor properties



You use the **Bacnet Export Manager** view of the BACnet **Export Table** to add, edit, delete, and access exported components.

In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Event Detection Enable | true (default) or false | Turns detection of an input event on (true) and off (false) for analog, binary and multi-state descriptors. |
| Bacnet Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Point Ord | ord (defaults to null) | Defines the ORD (address) for the point. |
| Object ID | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |

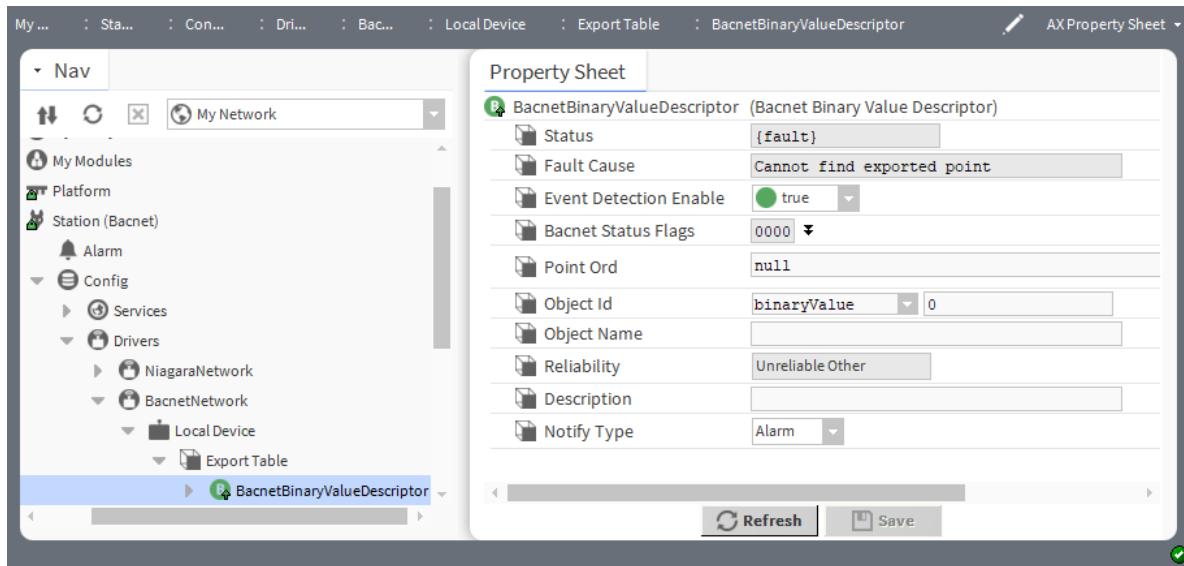
| Property | Value | Description |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Reliability | read-only | <p>Provides information when an error interrupts communication with a device.</p> <p>No Fault Detected indicates that things are OK.</p> <p>Unreliable Other indicates that the exported BACnet object appears in fault. This might happen, for example, if the source (exported) component is a proxy point under some other driver, and device communications are down or if the point has been disabled.</p> <p>In this case, the BACnet export descriptor retains a status of ok.</p> |
| Description | text | <p>Provides additional information, which could include the camera's geographical location or other unique information.</p> <p>NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the Add and Edit window.</p> |
| Notify Type | (optional) Alarm (default) or Event | Indicates if the notification is from an alarm or an event. This property applies if the source component has an alarm extension. |
| Device Type | (optional) text only for some object types (Analog Input, Analog Output, Binary Input, Binary Output, Multi-state Input, Multi-state Output) | Defines the type of physical input or output. This might apply mostly to exported Ndio points where each point corresponds to a physical device. |

Bacnet Binary Value Descriptor (ServerBVDesc)

This component exposes boolean-type component as a BACnet Binary Value object (non-commandable). You use the **Bacnet Export Manager** view of the **Bacnet Export Table** to add, edit, delete, and access exported components.

You use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

Figure 119 Binary Value Descriptor properties



In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Event Detection Enable | true (default) or false | Turns detection of an input event on (true) and off (false) for analog, binary and multi-state descriptors. |
| Bacnet Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Point Ord | ord (defaults to null) | Defines the ORD (address) for the point. |
| Object ID | drop-down list and number (from 0 to 4194302) | <p>Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number.</p> <p>For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device.</p> |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | <p>Identifies a BACnet object. This name should be unique within this specific device.</p> <p>For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device.</p> <p>For a proxy point, this is the default name in the station.</p> <p>You can shorten or edit Object Name, however, each Object Name must be unique among all server descriptors under the Export Table.</p> |
| Reliability | read-only | Provides information when an error interrupts communication with a device. No Fault Detected indicates that things are OK. |

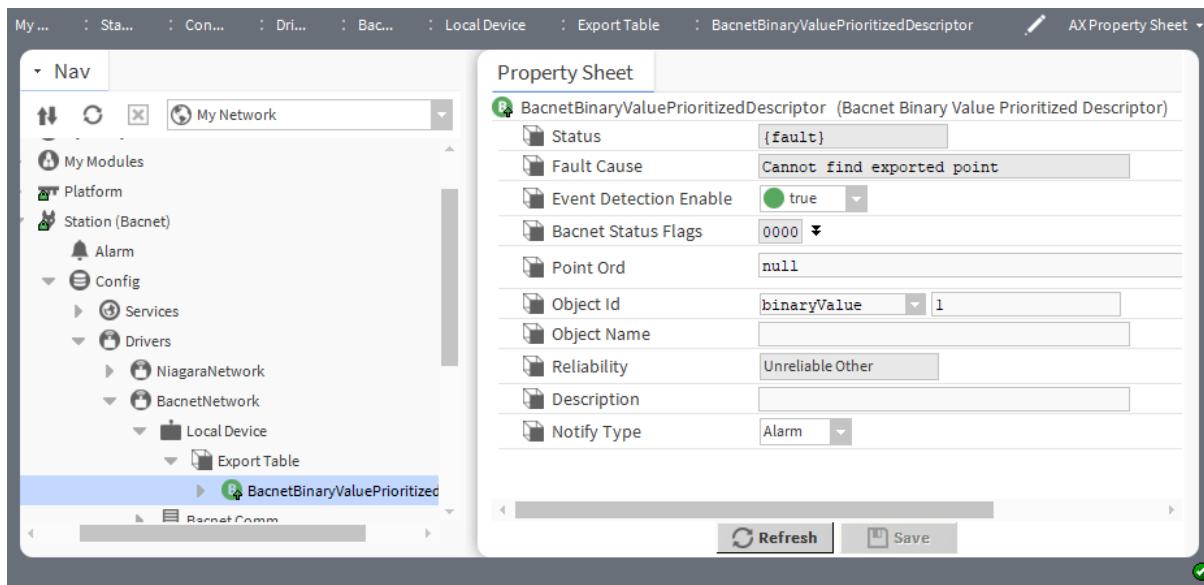
| Property | Value | Description |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | <p>Unreliable Other indicates that the exported BACnet object appears in fault. This might happen, for example, if the source (exported) component is a proxy point under some other driver, and device communications are down or if the point has been disabled.</p> <p>In this case, the BACnet export descriptor retains a status of ok.</p> |
| Description | text | <p>Provides additional information, which could include the camera's geographical location or other unique information.</p> <p>NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the New and Edit windows.</p> |
| Notify Type | (optional) Alarm (default) or Event | Indicates if the notification is from an alarm or an event. This property applies if the source component has an alarm extension. |
| Device Type | (optional) text only for some object types (Analog Input, Analog Output, Binary Input, Binary Output, Multi-state Input, Multi-state Output) | Defines the type of physical input or output. This might apply mostly to exported Ndio points where each point corresponds to a physical device. |

Bacnet Binary Value Prioritized Descriptor (ServerBVPDesc)

This component exposes a BooleanWritable as a writable (commandable) BACnet Binary Value object. You use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

You use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

Figure 120 Binary Value Prioritized Descriptor properties



In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Event Detection Enable | true (default) or false | Turns detection of an input event on (true) and off (false) for analog, binary and multi-state descriptors. |
| Bacnet Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Point Ord | ord (defaults to null) | Defines the ORD (address) for the point. |
| Object ID | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number. For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name, however, each Object Name must be unique among all server descriptors under the Export Table. |
| Reliability | read-only | Provides information when an error interrupts communication with a device. No Fault Detected indicates that things are OK. |

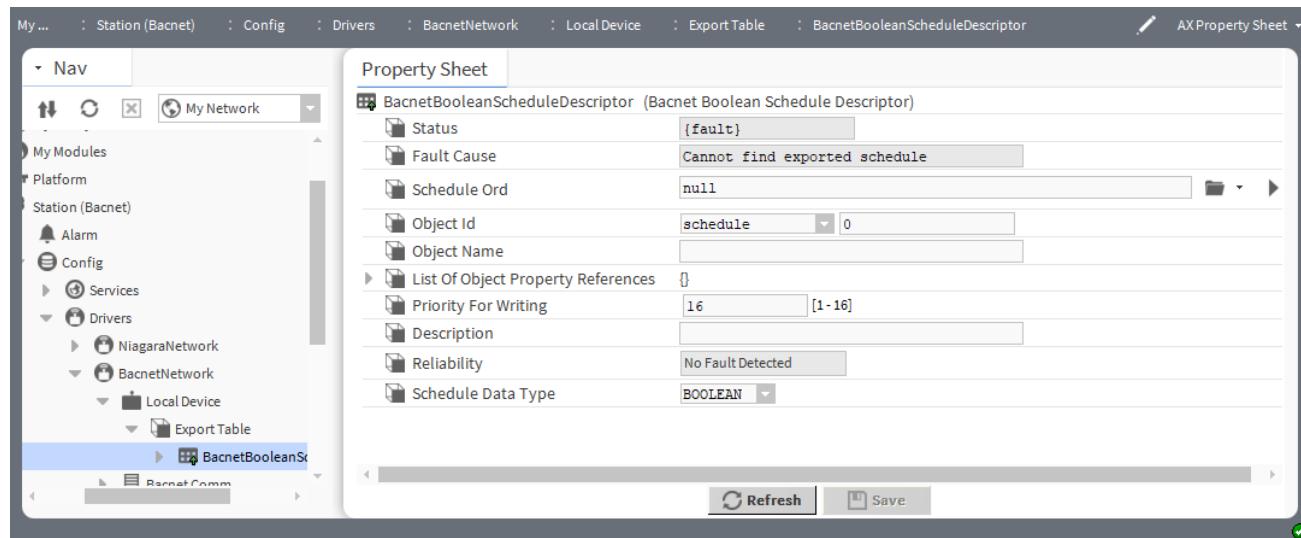
| Property | Value | Description |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | <p>Unreliable Other indicates that the exported BACnet object appears in fault. This might happen, for example, if the source (exported) component is a proxy point under some other driver, and device communications are down or if the point has been disabled.</p> <p>In this case, the BACnet export descriptor retains a status of ok.</p> |
| Description | text | <p>Provides additional information, which could include the camera's geographical location or other unique information.</p> <p>NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the New and Edit windows.</p> |
| Notify Type | (optional) Alarm (default) or Event | Indicates if the notification is from an alarm or an event. This property applies if the source component has an alarm extension. |
| Device Type | (optional) text only for some object types (Analog Input, Analog Output, Binary Input, Binary Output, Multi-state Input, Multi-state Output) | Defines the type of physical input or output. This might apply mostly to exported Ndio points where each point corresponds to a physical device. |

Bacnet Boolean Schedule Descriptor (ServerBoolSchedDesc)

This component exposes a BooleanSchedule as a BACnet Schedule object. You use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

You use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

Figure 121 Bacnet Boolean Schedule Descriptor properties



In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Object ID | drop-down list and number (from 0 to 4194302) | <p>Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number.</p> <p>For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device.</p> |
| Object Name | text in the format name . name . name . name (default includes the entire component path in the station using period (".") delimiters between parent.child levels) | <p>Identifies a BACnet object. This name should be unique within this specific device.</p> <p>For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device.</p> <p>For a proxy point, this is the default name in the station.</p> <p>You can shorten or edit Object Name, however, each Object Name must be unique among all server descriptors under the Export Table.</p> |
| List of Object Property References | read-only | Provides a standard Bacnet ListOf sequence, which contains a non-indexed sequence of references related to the schedule. |
| Priority for Writing | numeric text values 1 through 16 | Reports schedule priority, which defaults to 16. |
| Description | text | <p>Provides additional information, which could include the camera's geographical location or other unique information.</p> <p>NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the Add and Edit window.</p> |
| Reliability | read-only | <p>Provides information when an error interrupts communication with a device.</p> <p>No Fault Detected indicates that things are OK.</p> <p>Unreliable Other indicates that the exported BACnet object appears in fault. This might happen, for example, if the source (exported) component is a proxy point under some other driver, and device communications are down or if the point has been disabled.</p> <p>In this case, the BACnet export descriptor retains a status of ok.</p> |
| Scheduled Data Type | drop-down list | Configures the type of data used for the schedule. |

Bacnet Calendar Descriptor (ServerCalendarDesc)

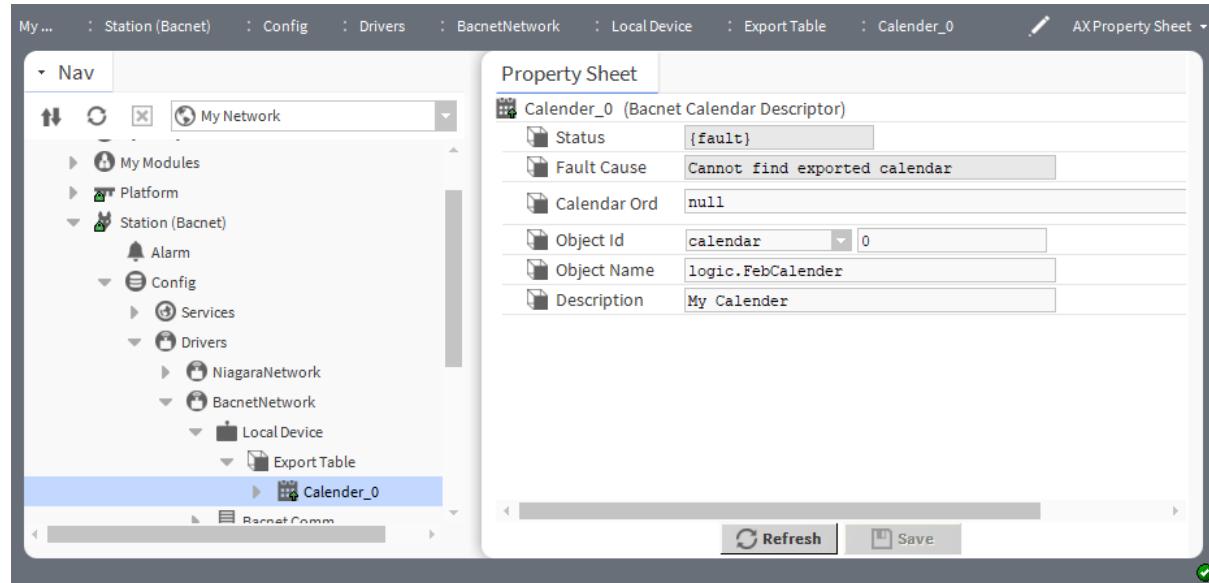
This descriptor component exposes a CalendarSchedule to BACnet as a Calendar object. You use the **Bacnet Export Manager** view of the Bacnet **Export Table** to add, edit, delete, and access exported components.

You use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

NOTE: Only the following read-only properties qualify for initial value setting during object creation. Requests that include non-qualified properties from the client cause a request to fail.

- Device Status Facets
- Discovery History Type
- Object Name
- Description

Figure 122 Bacnet Calendar Descriptor properties



In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|--------------|-----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Calender Ord | | |
| Object ID | drop-down list and number (from 0 to 4194302) | <p>Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number.</p> <p>For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device.</p> |

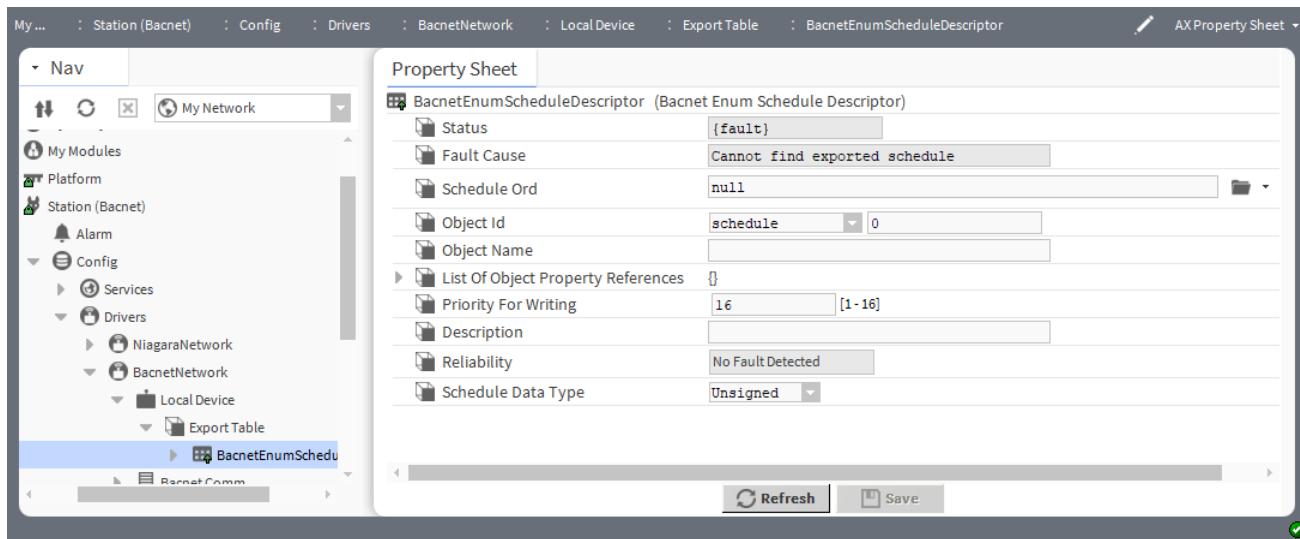
| Property | Value | Description |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | <p>Identifies a BACnet object. This name should be unique within this specific device.</p> <p>For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device.</p> <p>For a proxy point, this is the default name in the station.</p> <p>You can shorten or edit Object Name, however, each Object Name must be unique among all server descriptors under the Export Table.</p> |
| Description | text | <p>Provides additional information, which could include the camera's geographical location or other unique information.</p> <p>NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the Add and Edit window.</p> |

Bacnet Enum Schedule Descriptor (**ServerEnumSchedDesc**)

This component exposes a **EnumSchedule** to BACnet as a schedule object. You use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

You use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

Figure 123 Bacnet Enum Schedule Descriptor properties



For property descriptions, refer to the properties dictionary chapter in this document.

Bacnet Event Enrollment Descriptor (**bacnet-BacnetEventEnrollmentDescriptor**)

This dynamically created export descriptor exposes an **Event Enrollment** component. The Event Enrollment descriptor cannot be added manually using the **Bacnet Export Manager** view of the **BACnet Export Table**. You can only create an Event Enrollment object dynamically using a remote **Create-Object** request.

The Event Enrollment object is used for working with information related to the management of events in BACnet systems. For example, this object can be used to create: alarms, external alarms, and algorithmic alarms. Or it might be used to describe an event that might be an error condition ("Input out of range") or an alarm that other devices need to know about. It can directly tell one device about the event or use a Notification Class object to tell multiple devices.

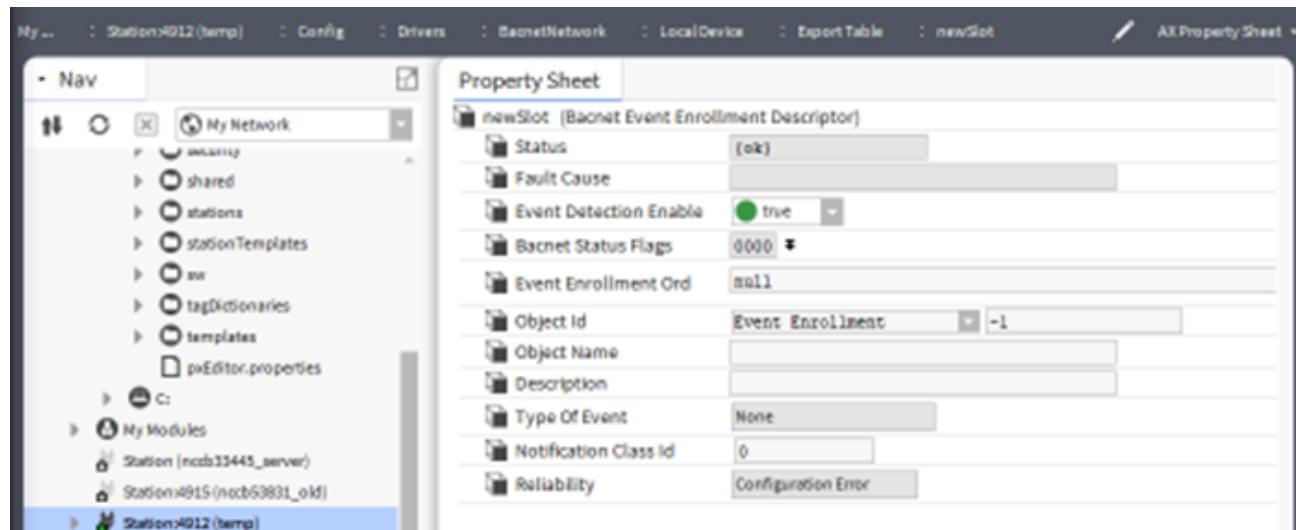
A BACnet client can send a request to a BACnet controller to dynamically create this component by using either an Object Id or Object Type. Optionally, the request may simultaneously specify initial values for one or more of a specified set of component properties.

NOTE: Only the following read-only properties qualify for initial value setting during object creation. Requests that include non-qualified properties from the client cause a request to fail.

- Device Status Facets
- Discovery History Type
- Object Name
- Description
- Notify Type
- Event Parameters
- Object Property Reference
- Event Enable
- Notification Class

You use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

Figure 124 Bacnet Event Enrollment Descriptor properties



In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|------------------------|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| Event Detection Enable | true (default) or false | Turns detection of an input event on (true) and off (false) for analog, binary and multi-state descriptors. |
| Bacnet Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |

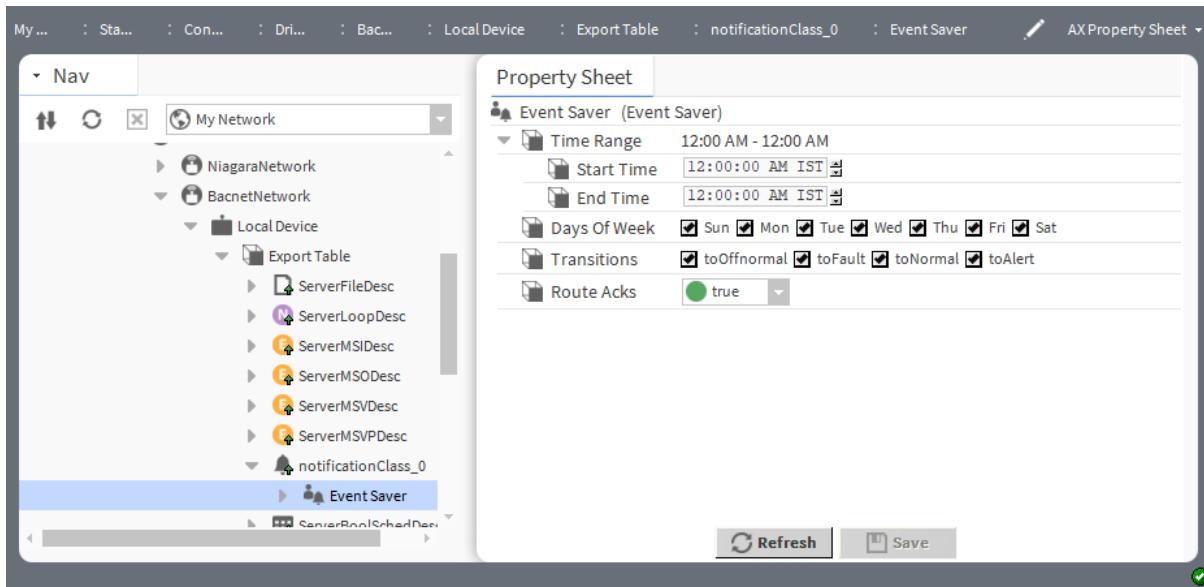
| Property | Value | Description |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Event Enrollment Ord | Slot or Handle | Defines the path to the source Event Enrollment object. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name . name . name . name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the Add and Edit window. |
| Type of Event | read-only | Indicates the referenced object event type, such as an error or alarm condition. |
| Notification Class Id | text | Specifies the notification class used for routing when handling and generating event notifications for this object. |
| Reliability | read-only | Provides information when an error interrupts communication with a device. No Fault Detected indicates that things are OK. Unreliable Other indicates that the exported BACnet object appears in fault. This might happen, for example, if the source (exported) component is a proxy point under some other driver, and device communications are down or if the point has been disabled. In this case, the BACnet export descriptor retains a status of ok. |

Event Saver (Event Saver)

The Event Saver component is a slot on the Bacnet Notification Class Descriptor. This component provides properties for setting time range and event routing information.

You use the **Bacnet Export Manager** view of the BACnet **Export Table** to add, edit, delete, and access exported components.

Figure 125 Event Saver properties



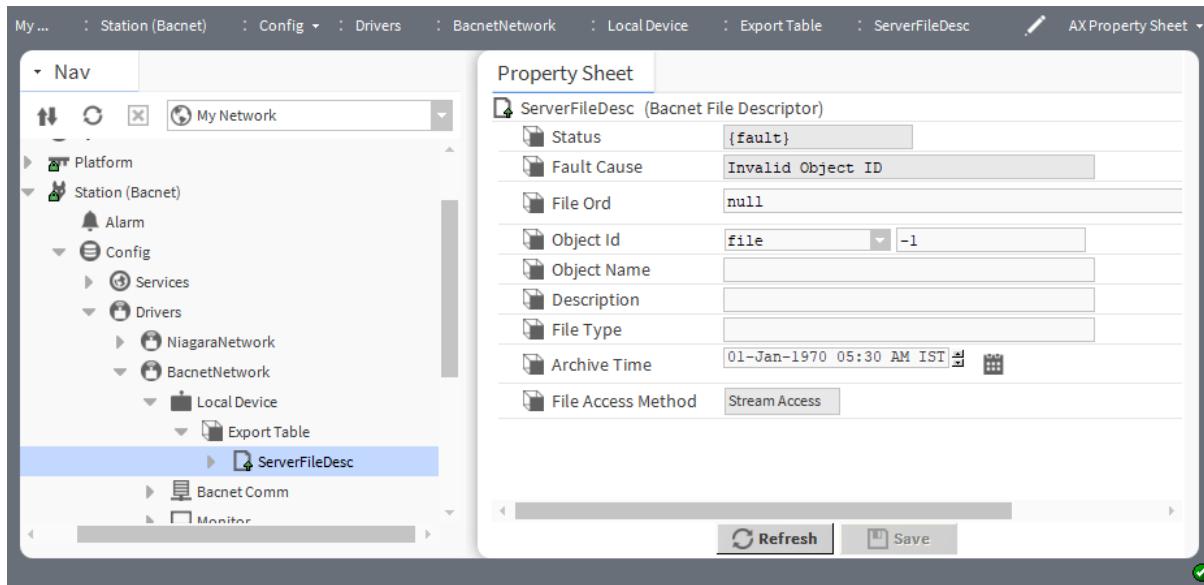
| Property | Value | Description |
|--------------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Time Range | hour and time zone | Sets the time of day to begin and stop the function (for example, trigger schedule, alarm event) |
| Days Of Week | check boxes | Selects to which days the times apply. |
| Transitions | check marks | Indicates the total number of records collected by the Trend Log object since creation. When the value of Total Record Count reaches its maximum possible value, the next value it takes is zero. |
| Route Acks | true (default) or false | Enables (true) and disables (false) the routing of alarm acknowledgements to the recipient. The framework does not route trap (event notification) acknowledgements if you select false. |

Bacnet File Descriptor (ServerFileDesc)

This component exposes a file under the station's folder as a BACnet file object. You use the **Bacnet File Export Manager** view of the Bacnet Export Table to add, edit, delete, and access exported files.

All files are accessed using the STREAM_ACCESS method.

Figure 126 Bacnet File Descriptor properties



One way to view these properties after dragging this descriptor to the **BacnetNetwork**→**Local Device**→**Export Table**, is to double-click the **ServerFileDesc** node in the Nav tree.

In addition to the common properties (Status and Fault Cause), this component has these properties.

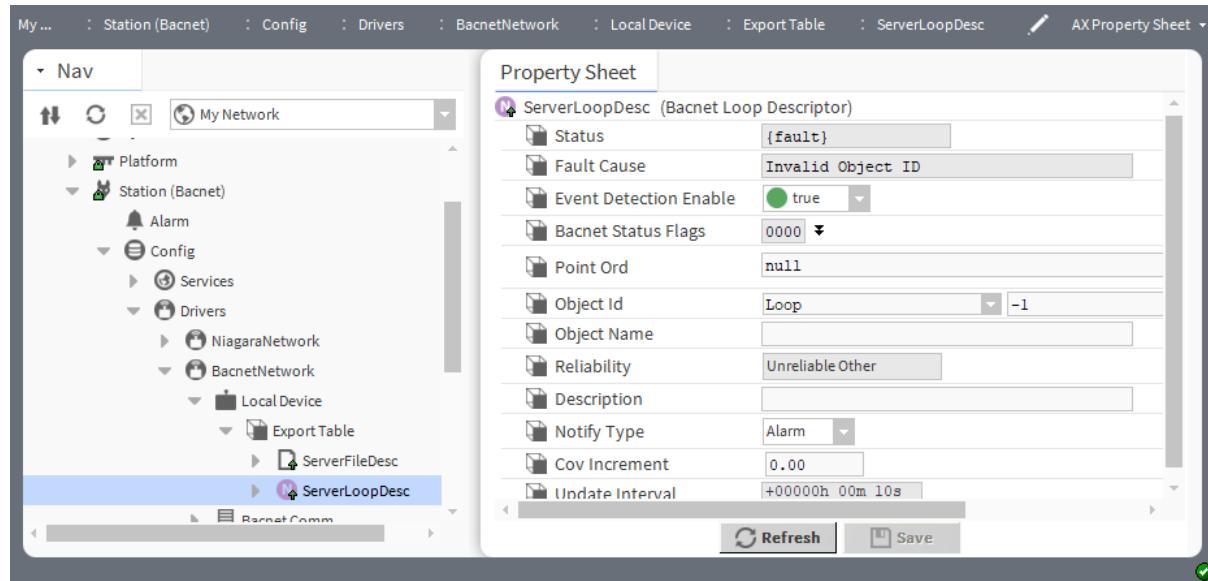
| Property | Value | Description |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| File Ord | text | Configures the file location. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the Add and Edit window. |

| Property | Value | Description |
|--------------------|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| File Type | text string (defaults to blank) | Specifies the BACnet property, as a text string intended to identify the use of this file. While the default value is blank, the actual exported BACnet File object contains some default values, such as <code>text/plain</code> or <code>application/zip</code> . |
| Archive Time | timestamp | Compares Timestamp against the actual file timestamp in order to set the BACnet property Archive (True if <code>Archive Time < file timestamp</code> , or False if <code>Archive Time > file timestamp</code>). |
| File Access Method | read-only | <p>Reports the BACnet property that indicates the type of file access supported: Stream Access only.</p> <p>NOTE: Additional properties in the exported File object exist, even though they are not visible in the Bacnet File Descriptor property sheet. These properties include:</p> <ul style="list-style-type: none"> • File Size (integer, in bytes) • Modification Date (actual file timestamp) • Archive (True or False, see <code>Archive Time</code> property above) • Read Only (True or False, as determined by the local Java file system) |

Bacnet Loop Descriptor (ServerLoopDesc)

This component exposes a kitControl LoopPoint to BACnet as a Loop object. You use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

Figure 127 Bacnet Loop Descriptor properties



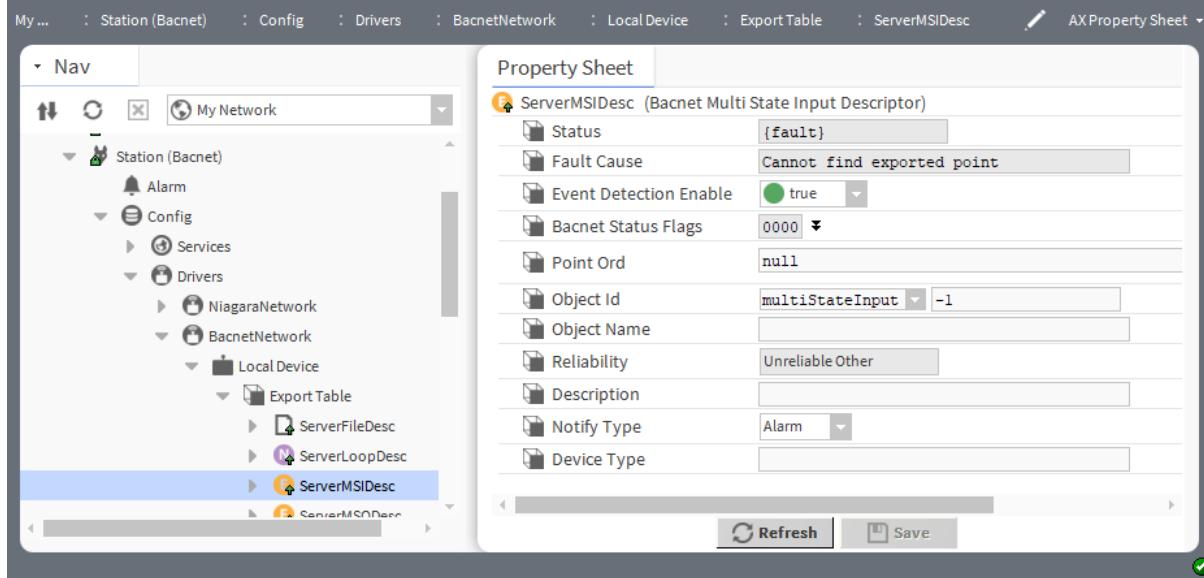
One way to view these properties after dragging this descriptor to the **BacnetNetwork**→**Local Device**→**Export Table**, is to double-click the **ServerLoopDesc** node in the Nav tree.

For property descriptions, refer to the properties dictionary chapter in this document.

Bacnet Multi State Input Descriptor (ServerMSIDesc)

This component exposes an EnumPoint as a BACnet Multi-state Input object. You use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

Figure 128 Bacnet Multi State Input Descriptor properties



You use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

In addition to the common properties (Status and Fault Cause), this component has these properties.

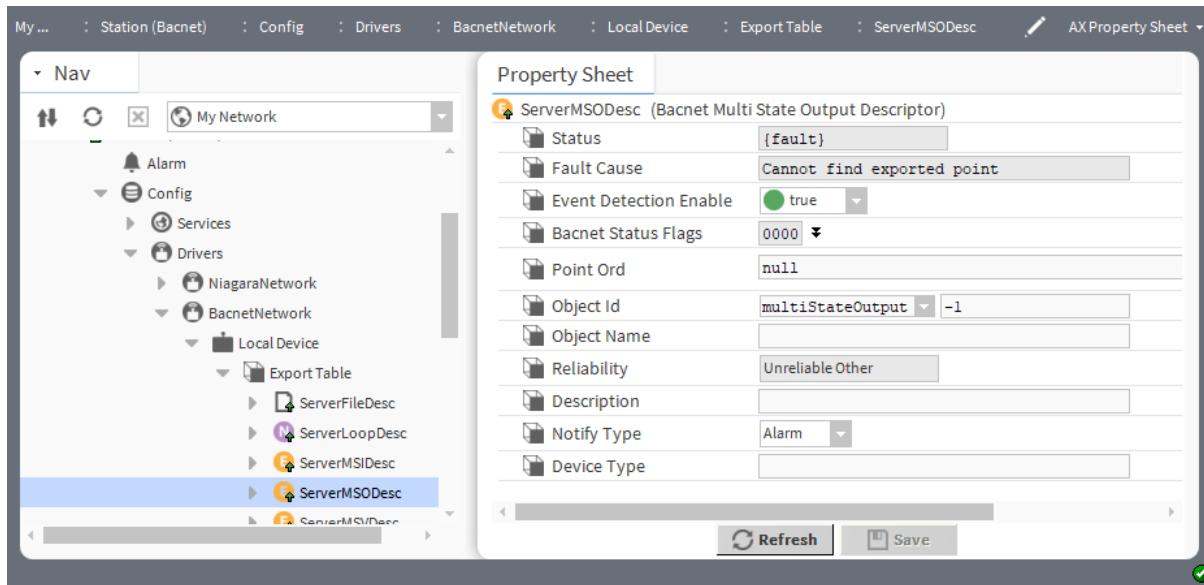
| Property | Value | Description |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Event Detection Enable | true (default) or false | Turns detection of an input event on (true) and off (false) for analog, binary and multi-state descriptors. |
| Bacnet Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Point Ord | ord (defaults to null) | Defines the ORD (address) for the point. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name . name . name . name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. |

| Property | Value | Description |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Reliability | read-only | <p>Provides information when an error interrupts communication with a device.</p> <p>No Fault Detected indicates that things are OK.</p> <p>Unreliable Other indicates that the exported BACnet object appears in fault. This might happen, for example, if the source (exported) component is a proxy point under some other driver, and device communications are down or if the point has been disabled.</p> <p>In this case, the BACnet export descriptor retains a status of ok.</p> |
| Description | text | <p>Provides additional information, which could include the camera's geographical location or other unique information.</p> <p>NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the New and Edit windows.</p> |
| Notify Type | (optional) Alarm (default) or Event | Indicates if the notification is from an alarm or an event. This property applies if the source component has an alarm extension. |
| Device Type | (optional) text only for some object types (Analog Input, Analog Output, Binary Input, Binary Output, Multi-state Input, Multi-state Output) | Defines the type of physical input or output. This might apply mostly to exported Ndio points where each point corresponds to a physical device. |

Bacnet Multi State Output Descriptor (**ServerMSODesc**)

This component exposes an **EnumWritable** as BACnet Multi-state Output object. You use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

Figure 129 Bacnet Multi State Output Descriptor properties



You use the **Bacnet Export Manager** view of the **Bacnet Export Table** to add, edit, delete, and access exported components.

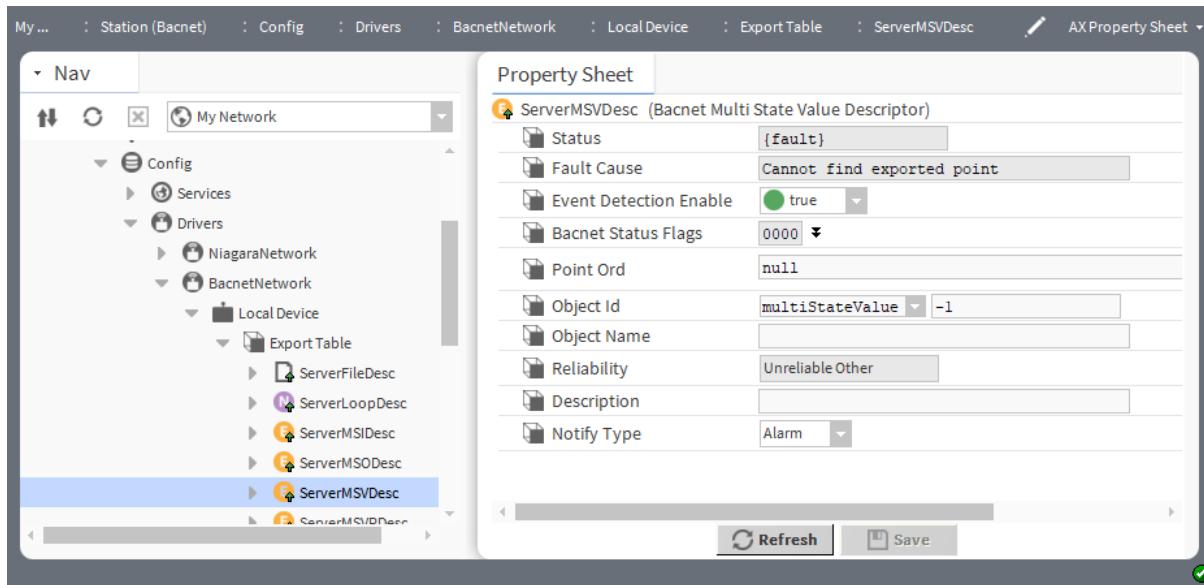
In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Event Detection Enable | true (default) or false | Turns detection of an input event on (true) and off (false) for analog, binary and multi-state descriptors. |
| Bacnet Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Point Ord | ord (defaults to null) | Defines the ORD (address) for the point. |
| Object ID | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Reliability | read-only | Provides information when an error interrupts communication with a device. |

| Property | Value | Description |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | No Fault Detected indicates that things are OK. Unreliable Other indicates that the exported BACnet object appears in fault. This might happen, for example, if the source (exported) component is a proxy point under some other driver, and device communications are down or if the point has been disabled. In this case, the BACnet export descriptor retains a status of ok. |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the New and Edit windows. |
| Notify Type | (optional) Alarm (default) or Event | Indicates if the notification is from an alarm or an event. This property applies if the source component has an alarm extension. |
| Device Type | (optional) text only for some object types (Analog Input, Analog Output, Binary Input, Binary Output, Multi-state Input, Multi-state Output) | Defines the type of physical input or output. This might apply mostly to exported Ndio points where each point corresponds to a physical device. |

Bacnet Multi State Value Descriptor (ServerMSVDesc)

This component exposes an EnumPoint as BACnet Multi-state Value object (non-commandable). You use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

Figure 130 Bacnet Multi State Value Descriptor properties

You use the **Bacnet Export Manager** view of the **Bacnet Export Table** to add, edit, delete, and access exported components.

In addition to the common properties (Status and Fault Cause), this component has these properties.

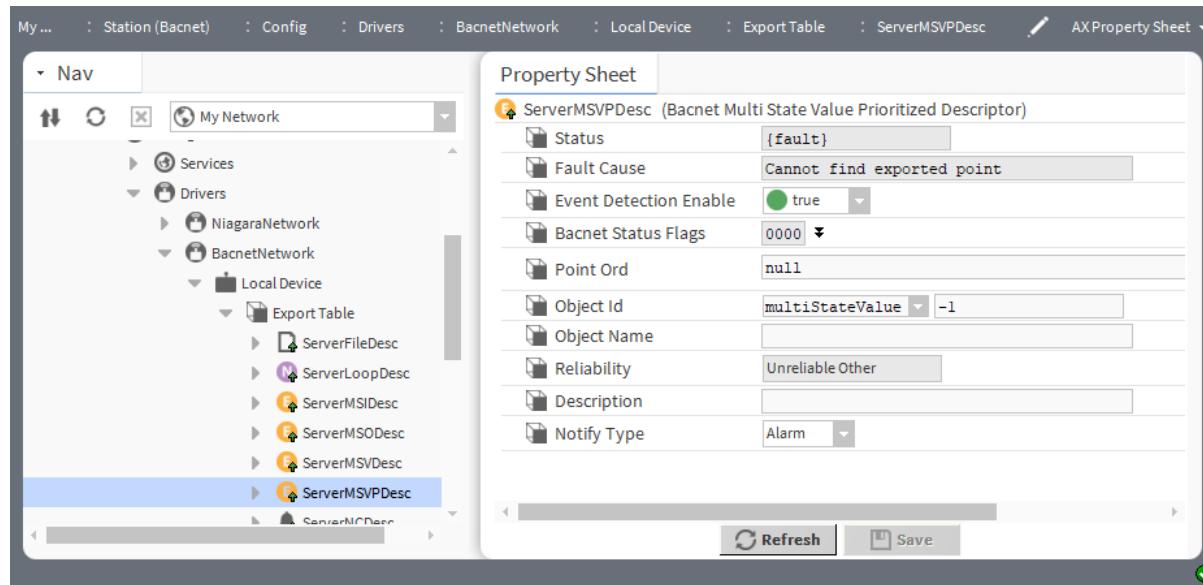
| Property | Value | Description |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Event Detection Enable | true (default) or false | Turns detection of an input event on (true) and off (false) for analog, binary and multi-state descriptors. |
| Bacnet Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Point Ord | ord (defaults to null) | Defines the ORD (address) for the point. |
| Object ID | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Reliability | read-only | Provides information when an error interrupts communication with a device. |

| Property | Value | Description |
|-------------|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | No Fault Detected indicates that things are OK. Unreliable Other indicates that the exported BACnet object appears in fault. This might happen, for example, if the source (exported) component is a proxy point under some other driver, and device communications are down or if the point has been disabled. In this case, the BACnet export descriptor retains a status of ok. |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the New and Edit windows. |
| Notify Type | (optional) Alarm (default) or Event | Indicates if the notification is from an alarm or an event. This property applies if the source component has an alarm extension. |

Bacnet Multi State Value Prioritized Descriptor (ServerMSVPDesc)

This component exposes an `EnumWritable` as a writable (commandable) BACnet Multi-state Value object. You use the **Bacnet Export Manager** view of the Bacnet Export Table to add, edit, delete, and access exported components.

Figure 131 Bacnet Multi State Value Prioritized Descriptor properties



You use the **Bacnet Export Manager** view of the Bacnet Export Table to add, edit, delete, and access exported components.

In addition to the common properties (Status and Fault Cause), this component has these properties.

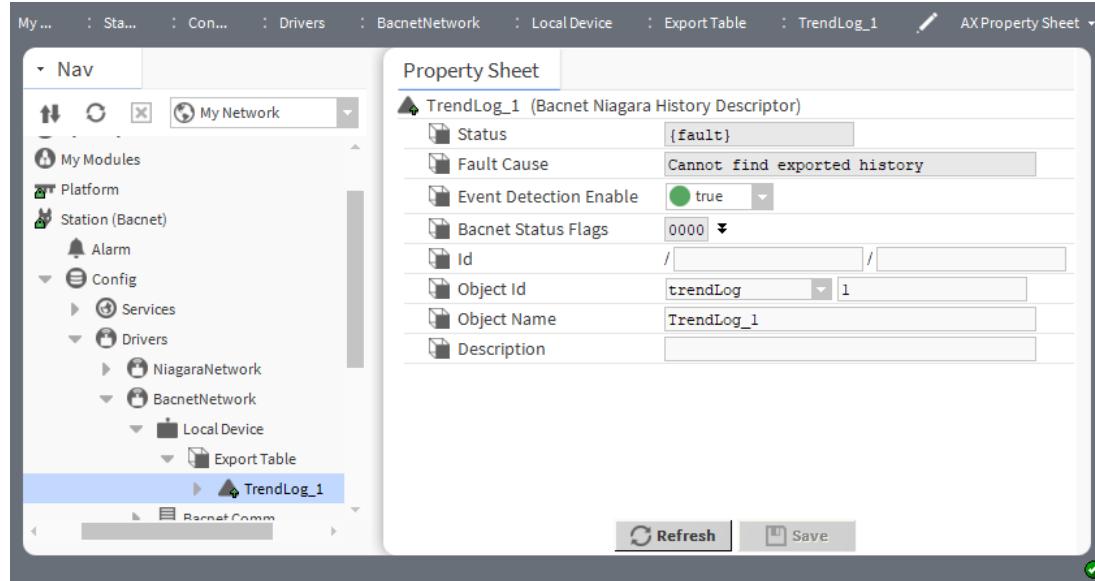
| Property | Value | Description |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Event Detection Enable | true (default) or false | Turns detection of an input event on (true) and off (false) for analog, binary and multi-state descriptors. |
| Bacnet Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Point Ord | ord (defaults to null) | Defines the ORD (address) for the point. |
| Object ID | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (".") delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Reliability | read-only | Provides information when an error interrupts communication with a device. No Fault Detected indicates that things are OK. Unreliable Other indicates that the exported BACnet object appears in fault. This might happen, for example, if the source (exported) component is a proxy point under some other driver, and device communications are down or if the point has been disabled. In this case, the BACnet export descriptor retains a status of ok . |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the New and Edit windows. |
| Notify Type | (optional) Alarm (default) or Event | Indicates if the notification is from an alarm or an event. This property applies if the source component has an alarm extension. |

Bacnet Niagara History Descriptor (ServerNiagaraHistoryDesc)

This component is the server log descriptor that exposes a standard history to BACnet as a trend log object. It supports exported histories based on time and index request for the trend log data. You use the **Bacnet**

Niagara Log Export Manager view of the **BACnet Export Table** to add, edit, delete, and access these descriptors. Alternatively, you can add one of several **BacnetTrendLogExt** (specialized history extension) too a point to create another type of log descriptor.

Figure 132 Bacnet history descriptor properties



You use the **Bacnet Export Manager** view of the **Bacnet Export Table** to add, edit, delete, and access exported components.

In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|------------------------|-----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Event Detection Enable | true (default) or false | Turns detection of an input event on and off for analog, binary and multi-state descriptors. How to find: expand the BACnet network, Local Device → Export Table and discover objects or create a new descriptor, then click the view drop-down list, select AX Property Sheet , and expand the descriptor |
| Bacnet Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. How to find: expand the BACnet network, Local Device → Export Table and discover objects or create a new descriptor, then click the view drop-down list, select AX Property Sheet , and expand the descriptor or expand the BACnet network, Local Device → Export Table and double-click an input |
| Id | read-only | Appears only for a BacnetNiagaraHistoryDescriptor , and reflects the same history Export Ord in the station. How to find: expand the BACnet network, Local Device → Export Table and discover objects or create a new descriptor, then click the view drop-down list, select AX Property Sheet , and expand the descriptor |
| Object ID | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . |

| Property | Value | Description |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name . name . name . name (default includes the entire component path in the station using period ("/) delimiters between parent.child levels) | <p>Identifies a BACnet object. This name should be unique within this specific device. When you select an object to add as a proxy point, this is the default name in the station for the BACnet proxy point.</p> <p>Including the path enforces (externally) the BACnet requirement for unique names for all objects in a device.</p> <p>How to find: Appears on all device and component Property Sheets and on the Search for objects by name or id window.</p> |
| Description | text | <p>Provides additional information, which could include the camera's geographical location or other unique information.</p> <p>NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the Add and Edit window.</p> <p>How to find: This property appears on multiple Property Sheets and in multiple windows.</p> |

Bacnet Notification Class Descriptor (ServerNCDesc)

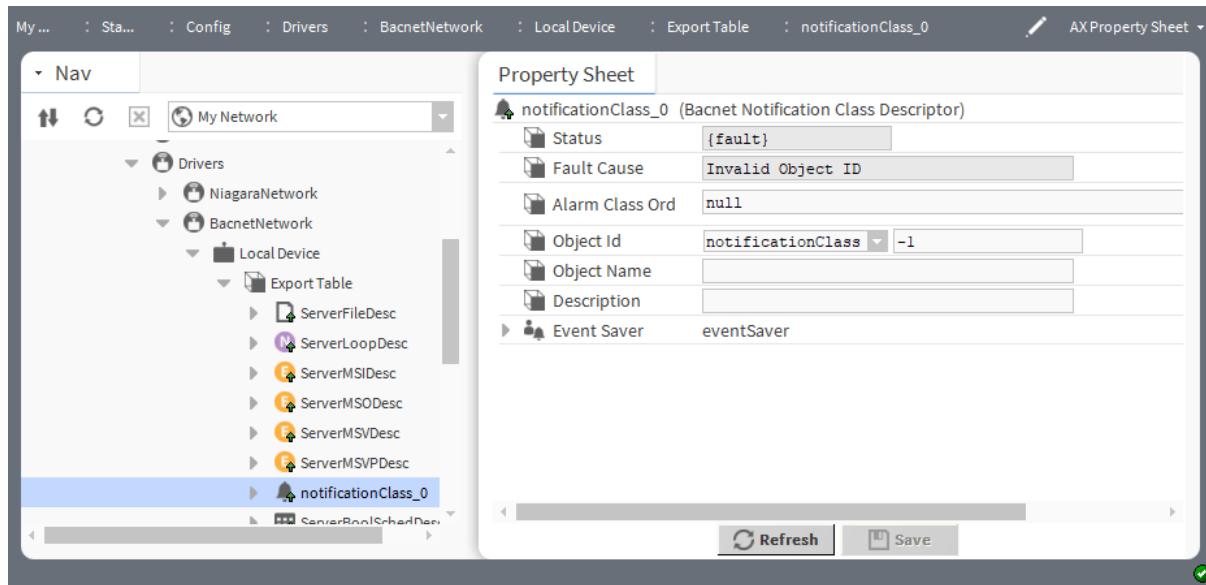
This export descriptor exposes a station's AlarmClass to BACnet as a Notification Class object.

You use the Bacnet Export Manager view of the BacnetExportTable to add, edit, delete, and access exported components.

NOTE: Only the following read-only properties qualify for initial value setting during object creation. Requests that include non-qualified properties from the client cause a request to fail.

- Device Status Facets
- Discovery History Type
- Object Name
- Description
- Priority
- Ack Required
- Recipient List

Figure 133 Bacnet Notification Class Descriptor properties



To add this descriptor to the Export Table, expand **Config**→**Drivers**→**BacnetNetwork**→**Local Device**, double-click **Export Table**, click **New**, select **Bacnet Notification Class Descriptor** and click **OK**.

Once the descriptor is available you access its properties by right-clicking **Export Table**, clicking **Views**→**AX Property Sheet** and expanding **BacnetNotificationClassDescriptor**.

In addition to the common properties (Status and Fault Cause), this component has these properties.

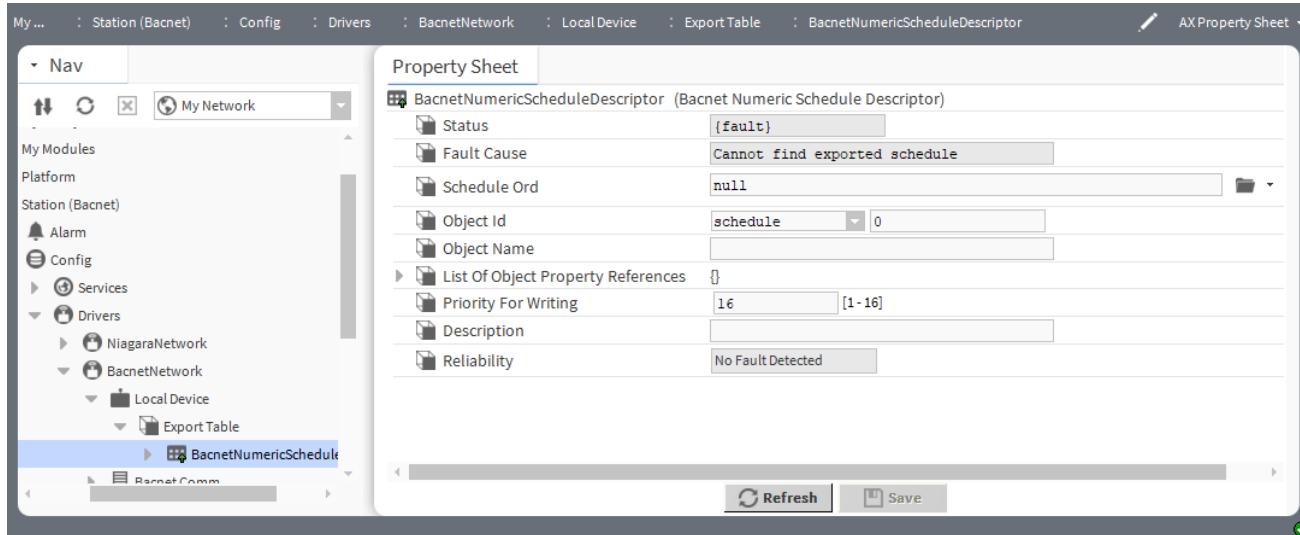
| Property | Value | Description |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Alarm Class Ord | text (ord) | Indicates the path to the connected alarm class. |
| Object ID | drop-down list and number (from 0 to 4194302) | <p>Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number. For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device.</p> |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | <p>Identifies a BACnet object. This name should be unique within this specific device.</p> <p>For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device.</p> <p>For a proxy point, this is the default name in the station.</p> <p>You can shorten or edit Object Name, however, each Object Name must be unique among all server descriptors under the Export Table.</p> |

| Property | Value | Description |
|-------------|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Description | text | <p>Provides additional information, which could include the camera's geographical location or other unique information.</p> <p>NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the Add and Edit window.</p> |
| Event Saver | additional properties | A separate topic documents these properties. |

Bacnet Numeric Schedule Descriptor (ServerNumSchedDesc)

This component exposes a NumericSchedule to BACnet as a schedule object. In the **bacnet** palette, it appears under the **Server** folder as **ServerNumSchedDesc**. However, you typically use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

Figure 134 Bacnet Numeric Schedule Descriptor properties



You use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|---------------|--------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Scheduled Ord | ord | Configures the location of the schedule. |
| Object Id | drop-down list and number (from 0 to 4194302) | <p>Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number.</p> <p>For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device.</p> |
| Object Name | text in the format name.name.name.name (default includes the | Identifies a BACnet object. This name should be unique within this specific device. |

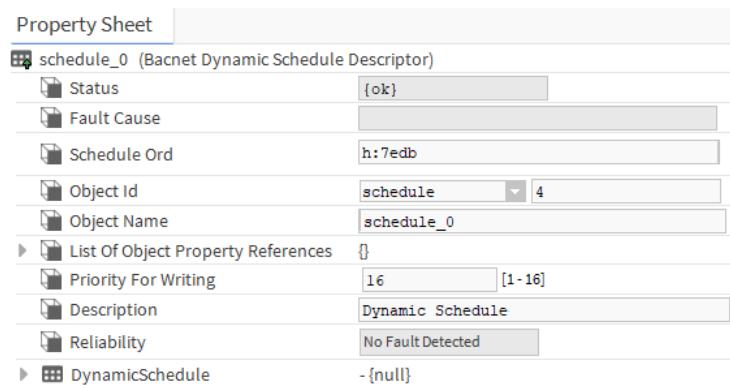
| Property | Value | Description |
|------------------------------------|-----------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | entire component path in the station using period (/) delimiters between parent.child levels) | For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| List of Object Property References | read-only | Provides a standard BacnetListOf sequence, which contains a non-indexed sequence of references related to the schedule. |
| Priority for Writing | numeric text values 1 through 16 | Reports schedule priority, which defaults to 16. |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the Add and Edit window. |
| Reliability | read-only | Provides information when an error interrupts communication with a device. No Fault Detected indicates that things are OK. Unreliable Other indicates that the exported BACnet object appears in fault. This might happen, for example, if the source (exported) component is a proxy point under some other driver, and device communications are down or if the point has been disabled. In this case, the BACnet export descriptor retains a status of ok. |

Bacnet Dynamic Schedule Descriptor (bacnet-BacnetDynamicScheduleDescriptor)

This component exposes a Bacnet Dynamic Schedule Descriptor component. You use the **Bacnet Export Manager** view of the BACnet **Export Table** to add, edit, delete, and access exported components.

NOTE: Only the following read-only properties qualify for initial value setting during object creation. Requests that include non-qualified properties from the client cause a request to fail.

- Device Status Facets
- Discovery History Type
- Object Name
- Description
- Schedule Default
- List of Object Property References
- Priority for Writing

Figure 135 Bacnet Dynamic Schedule Descriptor properties

You use the **Bacnet Export Manager** view of the BACnet **Export Table** to add, edit, delete, and access exported components.

In addition to the common properties (Status and Fault Cause), this component has these properties.

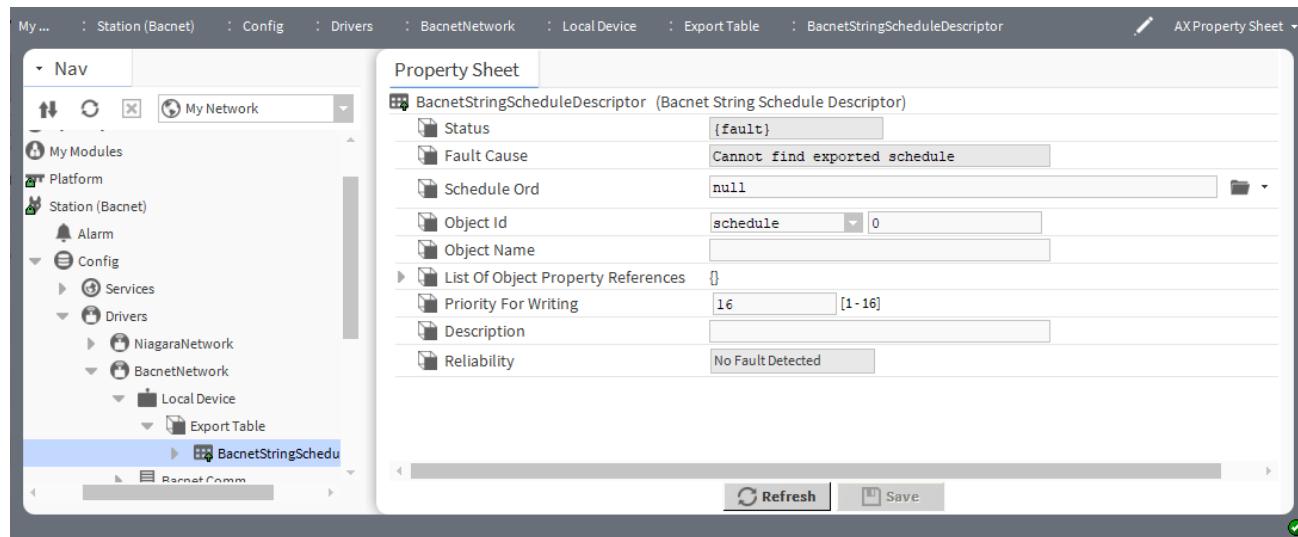
| Property | Value | Description |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Schedule Ord | Slot of Handle | This is the path to the source Schedule object. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name . name . name . name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| List Of Object Property References | read-only | Provides a standard Bacnet ListOf sequence, which contains a non-indexed sequence of references related to the schedule. |
| Priority for Writing | numeric text values 1 through 16 | Reports schedule priority, which defaults to 16. |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the Add and Edit window. |

| Property | Value | Description |
|------------------|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Reliability | read-only | <p>Provides information when an error interrupts communication with a device.</p> <p>No Fault Detected indicates that things are OK.</p> <p>Unreliable Other indicates that the exported BACnet object appears in fault. This might happen, for example, if the source (exported) component is a proxy point under some other driver, and device communications are down or if the point has been disabled.</p> <p>In this case, the BACnet export descriptor retains a status of ok.</p> |
| Dynamic Schedule | additional properties | Contains Dynamic Schedule properties. |

Bacnet String Schedule Descriptor (ServerStringSchedDesc)

This component exposes a StringSchedule to BACnet as a Schedule object. You use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

Figure 136 Bacnet String Schedule Descriptor properties



You use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|---------------|-----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Scheduled Ord | | |
| Object Id | drop-down list and number (from 0 to 4194302) | <p>Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number.</p> <p>For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device.</p> |

| Property | Value | Description |
|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | <p>Identifies a BACnet object. This name should be unique within this specific device.</p> <p>For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device.</p> <p>For a proxy point, this is the default name in the station.</p> <p>You can shorten or edit Object Name, however, each Object Name must be unique among all server descriptors under the Export Table.</p> |
| List of Object Property References | read-only | Provides a standard BacnetListOf sequence, which contains a non-indexed sequence of references related to the schedule. |
| Priority for Writing | numeric text values 1 through 16 | Reports schedule priority, which defaults to 16. |
| Description | text | <p>Provides additional information, which could include the camera's geographical location or other unique information.</p> <p>NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the Add and Edit window.</p> |
| Reliability | read-only | <p>Provides information when an error interrupts communication with a device.</p> <p>No Fault Detected indicates that things are OK.</p> <p>Unreliable Other indicates that the exported BACnet object appears in fault. This might happen, for example, if the source (exported) component is a proxy point under some other driver, and device communications are down or if the point has been disabled.</p> <p>In this case, the BACnet export descriptor retains a status of ok.</p> |

Bacnet Trend Log Descriptor (bacnet-BacnetTrendLogDescriptor)

This component is the server descriptor that exposes a history created with a BACnet trend log extension as a BACnet trend log object. You can use the **Bacnet Export Manager** view of the **BACnet Export Table** to add, edit, delete, and access exported components.

Unlike a history exported with a Bacnet Niagara History Descriptor (created by a standard history extension), a history exported by a Bacnet Trend Log Descriptor appears as a fully BACnet-compliant trend log object. This means that it supports, by sequence number, requests in addition to by time requests from external BACnet devices. The driver creates these descriptors in the root of the **BACnet Export Table** when you copy one of the various trend log extensions into a component.

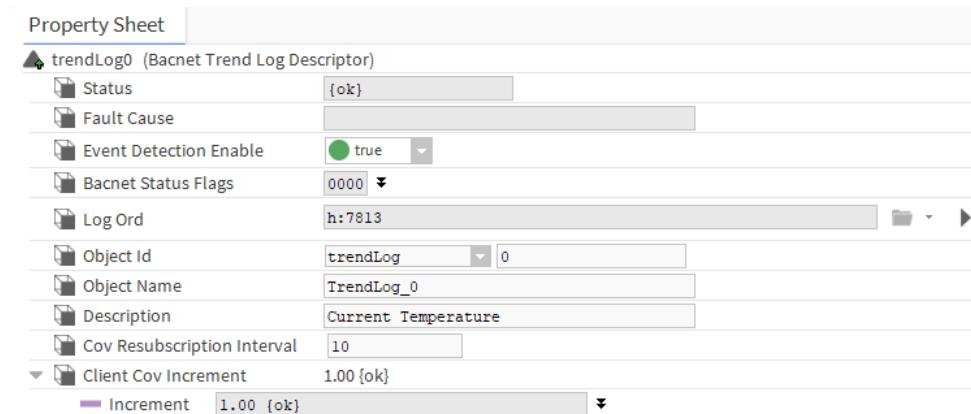
NOTE: Only the following read-only properties qualify for initial value setting during object creation. Requests that include non-qualified properties from the client cause a request to fail.

- Device Status Facets
- Discovery History Type
- Object Name

- Description
- Log Device Object Property
- Notification Class
- Notify Type

NOTE: Niagara does not support setting the **Log Device Object Property Reference** value directly from Workbench. It is recommended that you set this property only through a **Write Property** BACnet call over the network.

Figure 137 Properties of BacnetTrendLogDescriptor



You use the **Bacnet Export Manager** view of the **Bacnet Export Table** to add, edit, delete, and access exported components.

In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Event Detection Enable | true (default) or false | Turns detection of an input event on (true) and off (false) for analog, binary and multi-state descriptors. |
| Bacnet Status Flag | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Log Ord | read-only | Indicates the path of the selected trend log. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. |

| Property | Value | Description |
|-----------------------------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the Add and Edit window. |
| COV Resubscription Interval | number | Specifies the length of time that the extension waits before initiating a re-subscription request. |
| Client COV Increment | number to two decimal places | Specifies the minimum COV required before a COVNotification is issued to subscriber BACnet COV-clients. |

Worker Pool

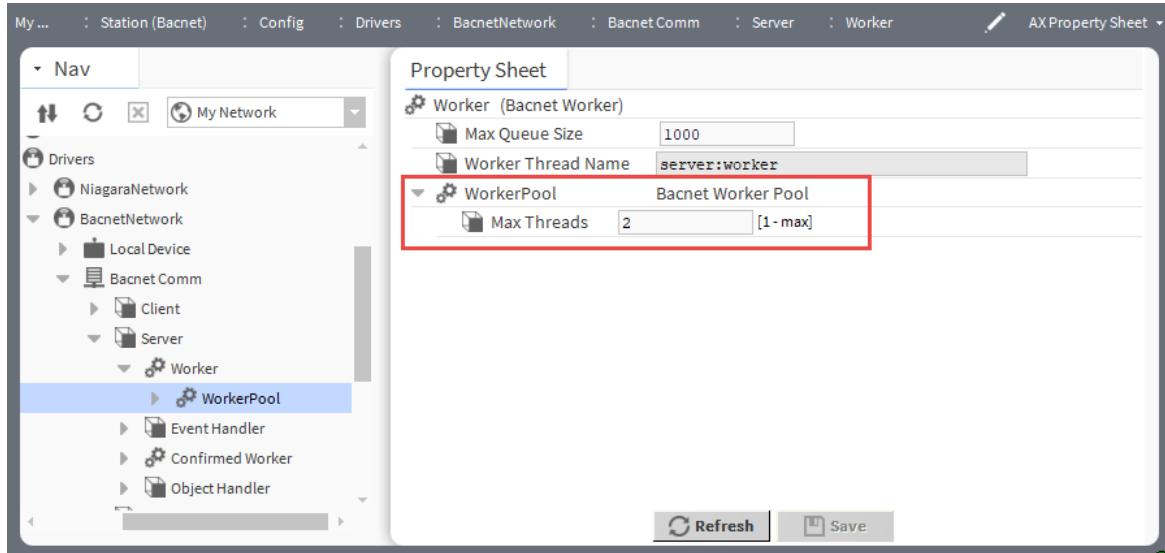
This component in the bacnet palette may improve messaging performance in cases where BACnet-queue-full messages have occurred.

Worker Pool (bacnet-BacnetWorkerPool)

This component in the **bacnet** palette may improve messaging performance in cases where BACnet-queue-full messages have occurred. Between the two **Bacnet Comm→Server** workers, typically the **Confirmed Worker** benefits most from adding a **Worker Pool**.

You can paste a **Worker Pool** component from the **bacnet** palette as a child under a **Worker**.

Figure 138 Worker Pool property



To access this property (**Max Threads**) for a **WorkerPool**, drag or copy a **WorkerPool** from the **bacnet** palette (**WorkerPools** node) to **BacnetNetwork→Bacnet Comm→Server→Worker**.

NOTE: BacnetWorker COV Worker does not support a **Worker Pool** child. An error results if you attempt to add one under it.

| Property | Value | Description |
|-------------|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Max Threads | text (defaults to 2) | Specifies the maximum number of threads in the worker pool. Often this is sufficient, but may be increased if necessary. However, large numbers may prove to be detrimental, especially if a controller platform with limited resources. |

Chapter 11 Components in the bacnetAws module

Topics covered in this chapter

- ◆ BacnetAwsNetwork
- ◆ Local Device
- ◆ dynamicObjects folder (bacnetAws-BacnetDynamicObjects)
- ◆ BacnetAwsDeviceFolder
- ◆ BacnetAwsDevice
- ◆ BacnetEventLogImport
- ◆ Elog
- ◆ Door
- ◆ ACC
- ◆ CMD
- ◆ LCO
- ◆ PC
- ◆ SVO

Components include services, folders and other model building blocks associated with a module. You may drag them to a property or wire sheet from a palette. The components of the **bacnetAWS** module support BACnet features running in a Supervisor PC.

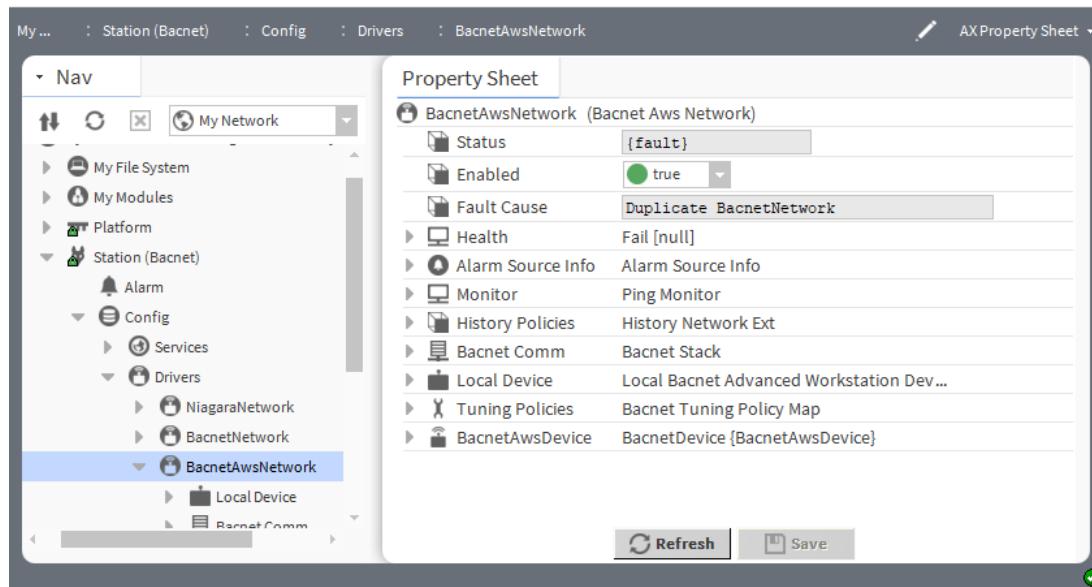
Descriptions included in the following topics appear as context-sensitive help topics when accessed by:

- Right-clicking on the object and selecting **Views→Guide Help**
- Clicking **Help→Guide On Target**

BacnetAwsNetwork

This component is the base container for all BACnet components (devices and proxy points). It resides under the station's **Drivers** container. Its default view is the **Bacnet Aws Device Manager**. In addition to devices and proxy points, this component contains the station's BACnet communications protocol stack (Bacnet Comm), and a Local Bacnet Aws Device, which configures the station's representation as a BACnet device.

The driver supports only one BACnet network component in a station regardless of how many different BACnet link-layer protocols are being used.

Figure 139 BacnetAwsNetwork Property Sheet

To access these properties, expand **Drivers** double-click the **BacnetAwsNetwork** and click, drop-down view menu and select **AX Property Sheet**

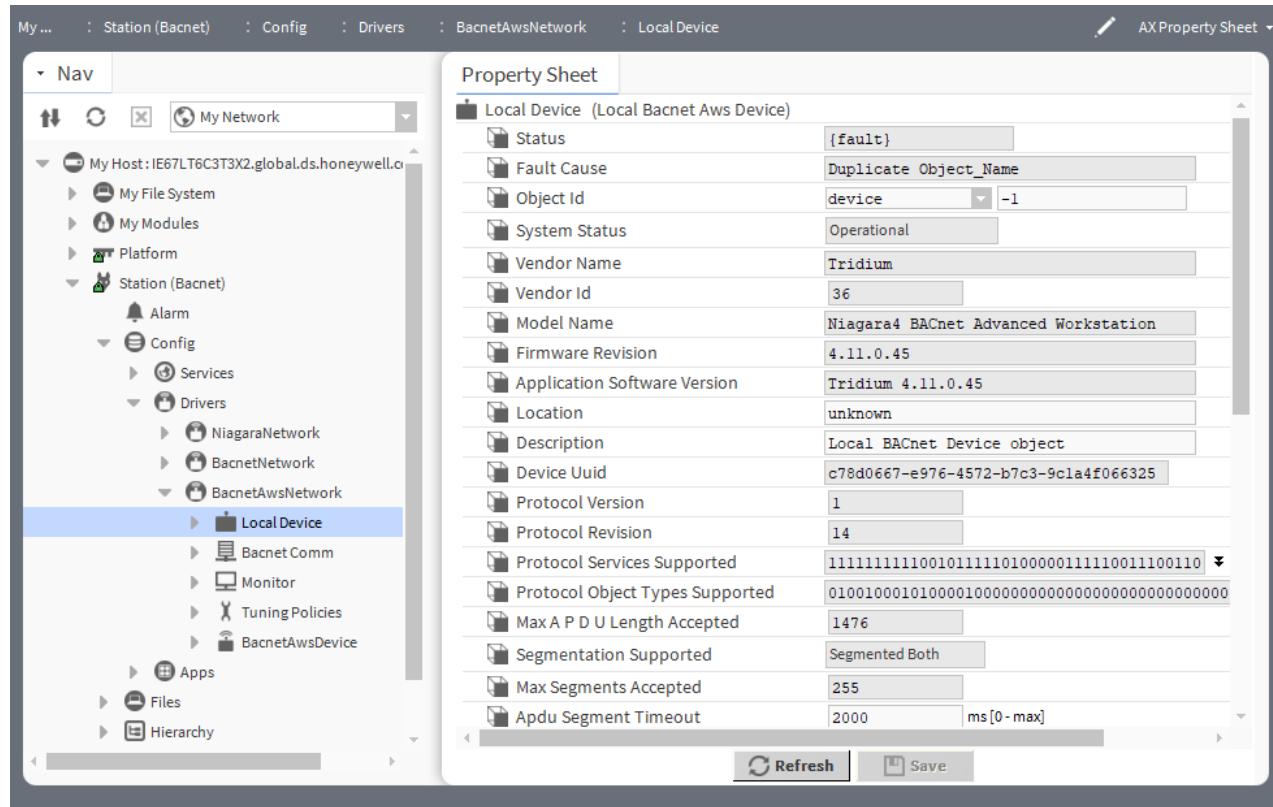
In addition to the common properties (Status, Enabled, Health and Fault Cause), this component has these properties.

| Property | Value | Description |
|-------------------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Alarm Source Info | additional properties | Contains a set of properties for configuring and routing alarms when this component is the alarm source. For property descriptions, refer to the <i>Niagara Alarms Guide</i> |
| Monitor | ping monitor | Configures a network's ping mechanism, which verifies network health. This includes verifying the health of all connected objects (typically, devices) by pinging each device at a repeated interval. The <i>Niagara Drivers Guide</i> documents these properties. |
| History Policies | additional properties | Configures the history network extension, which is documented in a separate topic. |
| Bacnet Comm | additional properties | Contains BACnet communication properties, which are documented in a separate topic. |
| Local Device | additional properties | Contains device properties, which are documented in a separate topic. |
| Tuning Policies | additional properties | Configures network rules for evaluating both write requests to writable proxy points as well as the acceptable freshness of read requests. For more information, refer to the <i>Niagara Drivers Guide</i> . |

Local Device

This frozen container under a BACnet network represents a BACnet station on the BACnet internetwork. Its child **Bacnet Export Table** container provides server-side functions to export station objects as BACnet objects and service BACnet client requests. In general, this means that it provides BACnet server responses to remote client requests from other BACnet devices on a connected network.

Figure 140 Local Device properties



To access these properties, expand **Drivers** and the BACnet network, then double-click **Local Device**.

The Local Bacnet Aws Device has all the same properties and components of a regular BACnet Local Device.

In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|---------------|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| System Status | read-only | Reports the current condition of the network. |
| Vendor Name | read-only | Reports the vendor number. |
| Vendor Id | read-only | Reports an identifier for the vendor's device. |
| Model Name | read-only | Reports the manufacturer's model name for the device. |

| Property | Value | Description |
|---------------------------------------------|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Firmware Revision | read-only | Reports the version of the device firmware. |
| Application Software Version | read-only | Reports the version of the manufacturer's software running on the device. |
| Location | text (defaults to unknown) | Describes the device location. |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. |
| Device Uuid (Universally Unique IDentifier) | 128-bit number (default is auto-generated) | <p>Uniquely identifies the object in the system.</p> <p>When you upgrade a BACnet driver or drop in a BACnet network from the palette, the driver automatically sets this 128-bit universally unique identifier (UUID) to a random value.</p> <p>You should not need to change this property. The driver uses it behind the scenes.</p> |
| Protocol Version | read-only | Displays the version number of the BACnet protocol. |
| Protocol Revision | read-only | Displays the current BACnet protocol level. |
| Protocol Services Supported | check box list | Expands to identify all the available BACnet device services supported by this version of Niagara. |
| Protocol Object Types Supported | check box list | Expands to identify all the available BACnet object types supported by this version of Niagara. |
| Max A P D U Length Accepted | read-only | Reports the number of points per poll or APDU (Application Protocol Data Units). |
| Segmentation Supported | read-only | Indicates the type of message segmentation the device supports. If a message exceeds the Max Segment Accepted length, the device must be able to break it into segments or part of the message is lost. |
| Max Segments Accepted | read-only | For a single segment, defines the number of characters the device supports. |
| Apdu Segment Timeout | milliseconds (defaults to 2000 ms) | <p>Defines the time the system waits before retransmitting an APDU segment. This property is only relevant if devices are doing segmentation.</p> <p>Works for both client and server operations. For a small IP-based network, an optimal setting may be: 5000 ms.</p> |
| Apdu Timeout | milliseconds (defaults to 3000 ms) | <p>Defines how long the system waits before retransmitting an APDU that requires acknowledgment, for which no acknowledgement has been received.</p> <p>Works for both client and server operations. Large, routed networks may require higher timeouts.</p> |
| Number of Apdu Retries | number (defaults to 3) | <p>Defines the maximum number APDU retransmissions.</p> <p>Works for both client and server operations. For a small IP-based network, an optimal setting may be: 1 retry.</p> |
| Database Revision | read-only | Reports the version of the database. |

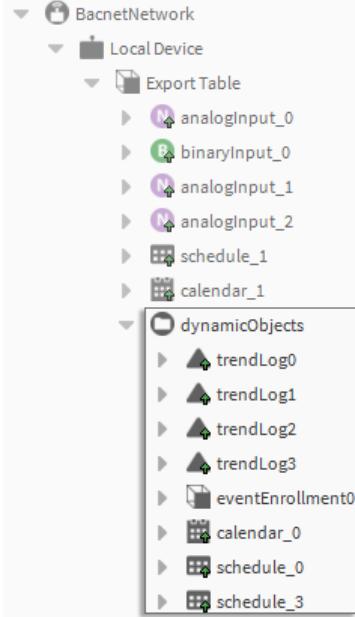
| Property | Value | Description |
|---------------------------------|------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Last Restore Time | read-only | Reports when a backup .dist file was last restored by a remote BACnet Workstation client. If never, all timestamp properties show asterisks (*). |
| Backup Failure Timeout | hours minutes seconds (defaults to 3 minutes) | Specifies the time the server device must wait before deciding that the client has given up, and can leave backup/restore mode. The client writes this to the server prior to beginning the backup/restore procedure. |
| Backup Preparation Time | read-only | Reports the time taken, after receiving a restore request, to gracefully shut down the station and close files before beginning the backup. |
| Restore Preparation Time | read-only | Reports the time taken, after receiving a restore request, to gracefully shut down the station and close files before beginning the dist file installation. |
| Restore Completion Time | read-only | Reports the time taken, after completing a dist file install from a restore, for the station to restart and become responsive. |
| Backup and Restore State | read-only | Shows the current status of any backup or restore operation. The driver displays <code>Idle</code> when no operation is active. This changes to <code>Preparing for backup</code> and so forth. |
| Character Set | drop-down list (defaults to <code>Ansi C3_4</code>) | Defines the character set supported, with other selections: IBM/Microsoft DBCS, JIS C 6226, ISO 10646 (UCS-4), ISO 10646 (UCS-2), ISO 8859-1, and Unknown. |
| Enumeration List | sets of enum facets | Lists the BACnet properties that can be exported to configure data exchange. This Extensible Enum List provides access to each property's configuration facets. |
| Export Table | table properties | Provides the special manager views to associate devices and points (local objects) with BACnet objects (exported objects). You discover objects or add them using the New button. The <i>Bacnet Export Manager</i> (plugin) topic documents this table. |
| Time Synchronization Recipients | Bacnet List Of | Define the devices to receive time synchronization messages from the driver. This list of recipients is for periodic time synchronization messages. You add to and remove entries from this list using the right-click <code>addElement</code> and <code>removeElement</code> actions. Entries in this list are BacnetRecipients . They can be either a device object identifier, or a BACnet address. |
| Time Synchronization Interval | timestamp (defaults to 24 hours) | Configures how frequently the software sends the time synchronization messages. |
| Align Intervals | true (defaults) or false | Turns on and off the configuration of standard synchronization intervals for controller stations. <code>true</code> sets up the same time synchronization interval in each host controller station. <code>false</code> allows a different time synchronization interval for each host controller station. |

| Property | Value | Description |
|-------------------------------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Interval Offset | minutes | Specifies an offset from the beginning of the interval at which to send the periodic time synchronization messages. If Align Interval is <code>false</code> , the driver ignores this property. |
| Utc Time Synchronization Recipients | Bacnet List Of | Lists the recipients for periodic Time Synchronization messages. You can add and remove entries to this list using the right-click <code>addElement</code> and <code>removeElement</code> actions. Entries in this list are BacnetRecipients. They can be either a device object identifier, or a BACnet address. |
| Last Restart Reason | drop-down list | Reports the reason for the last restart. |
| Time Of Device Restart | date and time | Specifies the time of device restart based on the time of day (Time), a sequence number (SeqNum) or a date and time (DateTime). |
| Restart Notification Recipients | Bacnet List Of | Defines the alarm recipients to receive notification that the station restarted. |
| ExportFolder | Bacnet array | Sets up a folder array. |

dynamicObjects folder (bacnetAws-BacnetDynamicObjects)

When a client sends a request to the controller for the creation of an object, the BACnet controller automatically creates a **dynamicObjects** folder under the **BacnetNetwork→Local Device→Export Table**.

Figure 141 dynamicObjects Folder



All dynamically created objects are located in this folder automatically as they are created. For more details about each descriptor, see the individual descriptor topics in this document.

BacnetAwsDeviceFolder

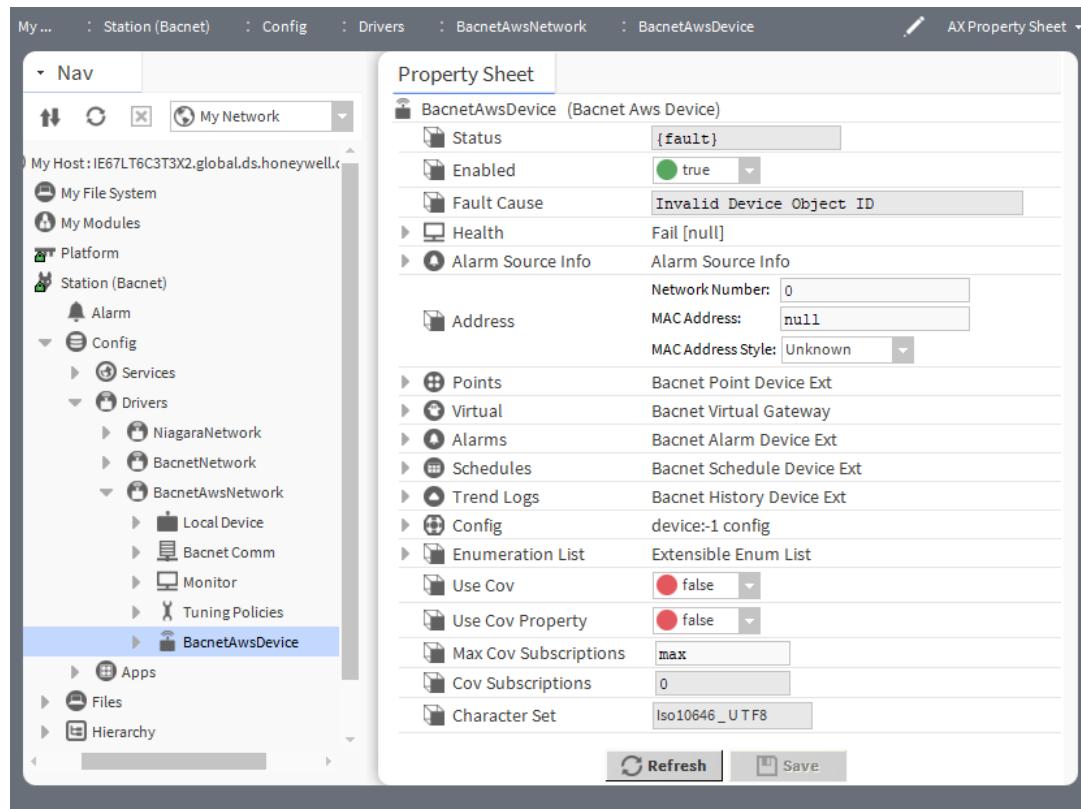
This component implements a folder under a BacnetAwsNetwork.

You add such folders using the **New Folder** button in the **Bacnet Aws Device Manager** view. The **BacnetDeviceFolder** is also available in the **bacnetAws** palette.

BacnetAwsDevice

This component is a Framework representation of a remote BACnet device. Each BacnetAwsDevice resides under the station's BacnetAwsNetwork. Each BacnetAwsDevice contains a full complement of device extensions (containers), including Points, Schedules, and Trend Logs (histories), for modeling data (from that device) in the station.

Figure 142 BacnetAwsDevice properties



To access these properties, expand **Drivers** and the **BacnetAwsNetwork**, then double-click the BACnet device.

In addition to the common properties (Status, Enabled, Health and Fault Cause), this component has these properties.

| Property | Value | Description |
|-------------------|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Alarm Source info | additional properties | Contains a set of properties for configuring and routing alarms when this component is the alarm source. For property descriptions, refer to the <i>Niagara Alarms Guide</i> |
| Address | IP address | Serves as a collective property name for Network Number , MAC Address , and MAC address style. |
| Points | container | Defines the ORD (address) for the point. |
| Virtual | Virtual component | Provides access to the virtual components in the station's virtual component space, which is specific to the device. This is in |

| Property | Value | Description |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | addition to the standard collection of slots for device-level components. |
| Alarms | additional properties | Specifies how alarms from that device are mapped into the station's own alarm subsystem, plus provide status properties related to alarm sharing. Provides access to three alarm-related properties: Alarm Class , Last Received Time , and Niagara Process Id , which are documented in a separate topic. |
| Schedules | container | Opens the Bacnet Schedule Import Manager of this schedule extention. The Property Sheet provides access to the Retry Trigger component, Subscribe Window , and Skip Write On Error properties, which are documented in a separate topic. |
| Trend Logs | Device extension | Opens the Bacnet History Manager . You use this manager to import BACnet trend log objects from the device to the station as histories. The property sheet accesses to the Retry Trigger properties, which is documented in the <i>Niagara Drivers Guide</i> . |
| Config | additional properties | Serves as a container for device configuration properties, which are documented in a separate topic. |
| Enumeration List | additional properties | Lists the BACnet properties that can be exported to configure data exchange. This Extensible Enum List provides access to each property's configuration facets. |
| Use Cov | true or false If the device was discovered, and station database determined that the device indicates support for server-side COV, this property defaults to true. Otherwise, it defaults to false indicating that no proxy points under the device use COV. | Enables and disables a device's support for COV (change of value) as a way to monitor proxy point values. Assuming the device supports subscription to the COV service, true triggers the driver to attempt the necessary updates (proxy subscriptions) to the value of each point using the BACnet COV subscription to the device. If the subscription attempt succeeds, the Read Status property of the point's BacnetProxyExt displays COV. If the subscription attempt fails, the driver uses normal polling and the Read Status property shows Polled. When true, individual proxy points under the device may use COV subscriptions, depending on their assigned tuning policy. When false, the driver ignores any proxy subscription updates. |
| Use Cov Property | true or false | Enables true and disables false use of the Use Cov property. |
| Max Cov Subscriptions | text (defaults to max) | Specifies the maximum number of COV (change of value) subscriptions that the database attempts to use with this device. This restriction prevents the station from consuming all of the available subscription space in the device. Applies only if Use Cov is true. |

| Property | Value | Description |
|------------------|----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cov Subscription | read-only | Reports number of active COV client subscriptions to the device. |
| Character Set | drop-down list (defaults to Ansi C3_4) | Defines the character set supported, with other selections: IBM/Microsoft DBCS, JIS C 6226, ISO 10646 (UCS-4), ISO 10646 (UCS-2), ISO 8859-1, and Unknown. |

Config

BacnetAwsConfigDeviceExt is a frozen device extension under every **BacnetAwsDevice**.

This component functions as the container for any config-type objects, which represent individual BACnet objects in the device.

AwsConfigFolder

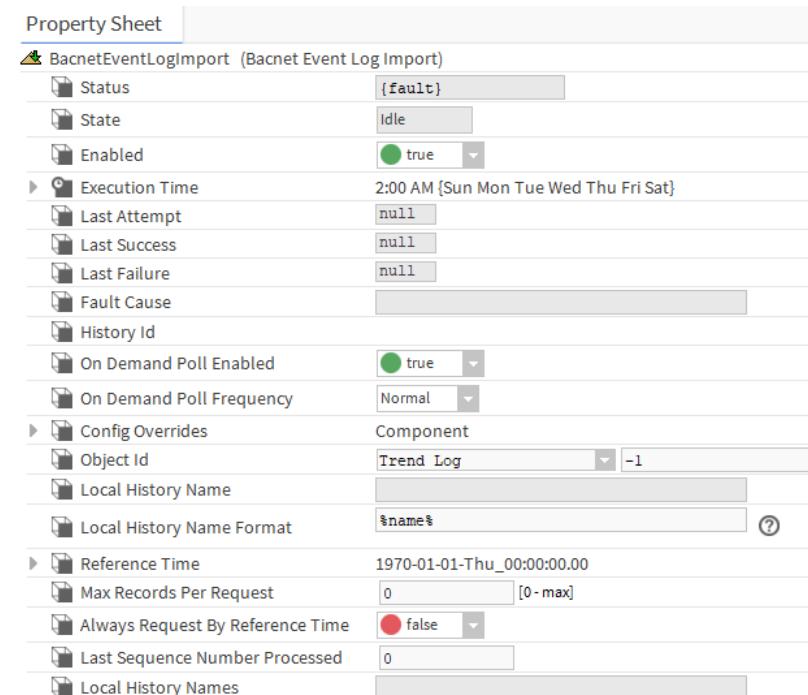
This component implements a folder under a BacnetAwsNetwork.

Typically, you add `AwsConfigFolder` using the **bacnetAWS** palette and dragging it into the **Bacnet Aws Config Manager** view of the **BacnetAwsNetwork**. Each `AwsConfigFolder` has its own **Bacnet Aws Config Manager** view.

BacnetEventLogImport

This component defines the archive action to retrieve data from a BACnet Event Log object into a history created by the addition of this component. It functions in a Supervisor station and is available only from the **bacnetAWS** palette.

Figure 143 Bacnet Event Log Import



To access these properties in a Supervisor station, expand the BACnet network and device, double-click the **Trend Logs** folder (📁) and expand the **BacnetEventLogImport** component. If no import component is

present, open a **bacnetAWS** palette, expand **Trending→Client Trending**, and drag a **BacnetEventLogImport** component to the **Trend Logs** folder.

In addition to the common properties (Status, Enabled and Fault Cause), this component has these properties.

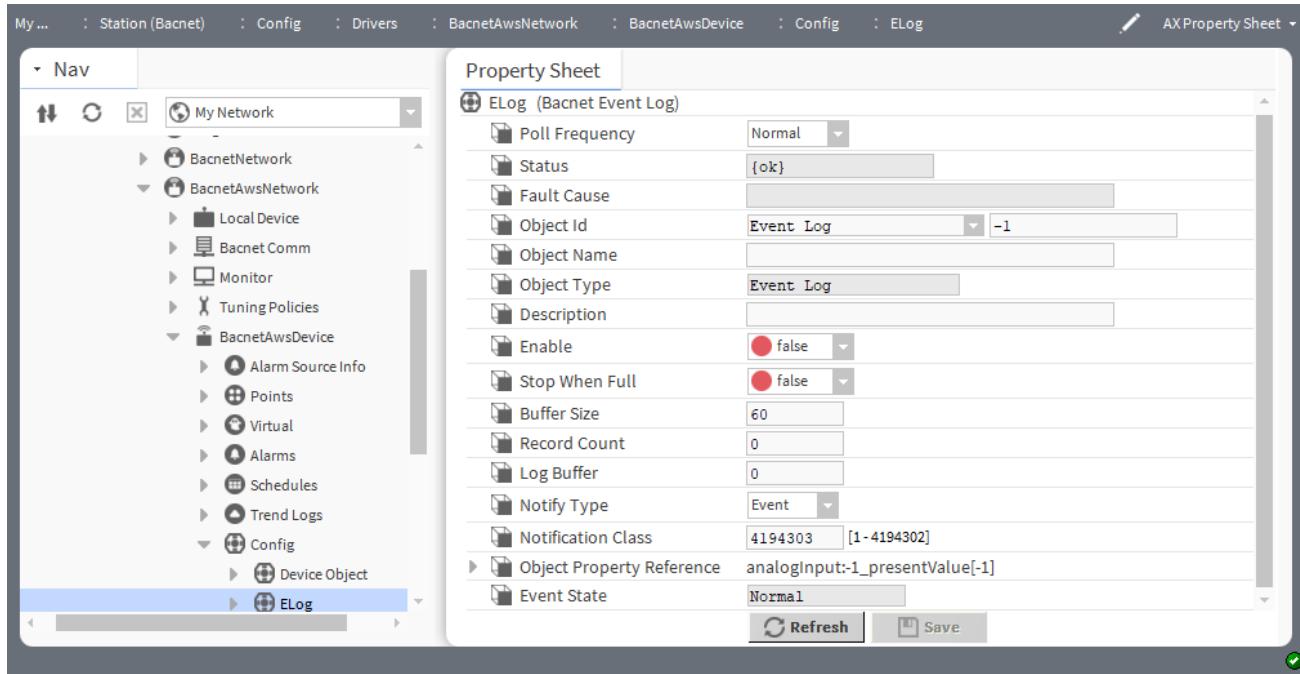
| Property | Value | Description |
|------------------------------|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Execution Time | multiple properties | <p>Configures the frequency with which data are imported from a remote trend log (or trend log multiple) into a history (or histories). Data import may occur at regular intervals (Interval), Daily or when manually requested (Manual).</p> <p>Time Of Day specifies a specific daily time.</p> <p>Randomization configures an import based on time.</p> <p>Days Of Week specifies on which days to apply the import time.</p> |
| Last Attempt | read-only | Reports the date and time of the last attempted execution. |
| Last Success | read-only | Reports the last time the system failed to perform this function. Refer to Fault Cause for details. |
| Last Failure | read-only timestamp | Reports the last time the station successfully performed this function. |
| History Id | read-only | Displays a combination of the station name and history name. |
| On Demand Poll Enabled | true (default) or false | Enables and disables polling of both local history sources and history imports for live data. The reported data appear on history charts and in history table views. |
| On Demand Poll Frequency | drop-down list | Configures how often this type of poll occurs. |
| Config Overrides, capacity | drop-down list (defaults to Unlimited) | <p>Defines the maximum number of history records allowed in the associated table.</p> <p>Unlimited enforces no limitation on the number of records.</p> <p>Record Count opens an additional property for defining the table limit.</p> |
| Config Overrides, fullPolicy | drop-down list (defaults to Roll) | <p>Defines what happens if Capacity is set to Record Count and the specified record count is reached.</p> <p>Roll overwrites the oldest records with the newest ones. This ensures that the latest data are recorded.</p> <p>Stop terminates recording when the number of stored records reaches the specified capacity.</p> <p>Full policy has no effect if Capacity is Unlimited.</p> |
| Object Id | drop-down list and number (from 0 to 4194302) | <p>Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number.</p> <p>For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device.</p> |
| Local History Name | read-only | Reports the name of the local history component. |

| Property | Value | Description |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Local History Name Format | text, BFormat (Baja Format) syntax (defaults to %name%, which recreates the BACnet name of the trend log object) or a combination of text and BFormat. | <p>Defines the format string script for returning a local history. This is the name of the history initially created and populated by the trend log data and includes, typically, the name of the BACnet device component and the BACnet trend log object.</p> <p>For example: %parent.parent.displayName% %name% could define the name of a VAV fan: J7Bnet BldgA_VAV1_Fan.</p> <p>You can modify the default BFormat with other BFormat scripting. You could enter static text for the history name—or, some combination of static text and valid BFormat scripting text.</p> <p>In the case of trend log multiples (BacnetTrendLogMultipleImport), the default %name% value results in histories named using the BACnet name of the trend log multiple object, with underscore delimiters between the BACnet names of each _object_property record.</p> |
| Reference Time | hours, minutes, seconds | Defines the timestamp the system uses to include records in the trend log. |
| Max Records Per Request | 0 (default) to max | Defines the maximum number of records that can be archived in this trend log. The Framework asks for 10 records per request. |
| Always Request By Reference Time | true or false (default) | Configures if the driver should always use the Reference Time to request records. |
| Last Sequence Number Processed | number | Defines a maximum record number to process. |
| Local History Names | read-only | Reports the name of the local history component. |

Elog

This component is a Config object that represents a Bacnet Event Log object in its entirety. An Event Log object records BACnet event notifications with timestamps and other pertinent data for subsequent retrieval.

Figure 144 Elog properties



To access these properties, expand Drivers→BacnetAwsNetwork→BacnetAwsDevice→Config and double-click **ELog**.

In addition to the common properties (Status and Fault Cause), this component has these properties.

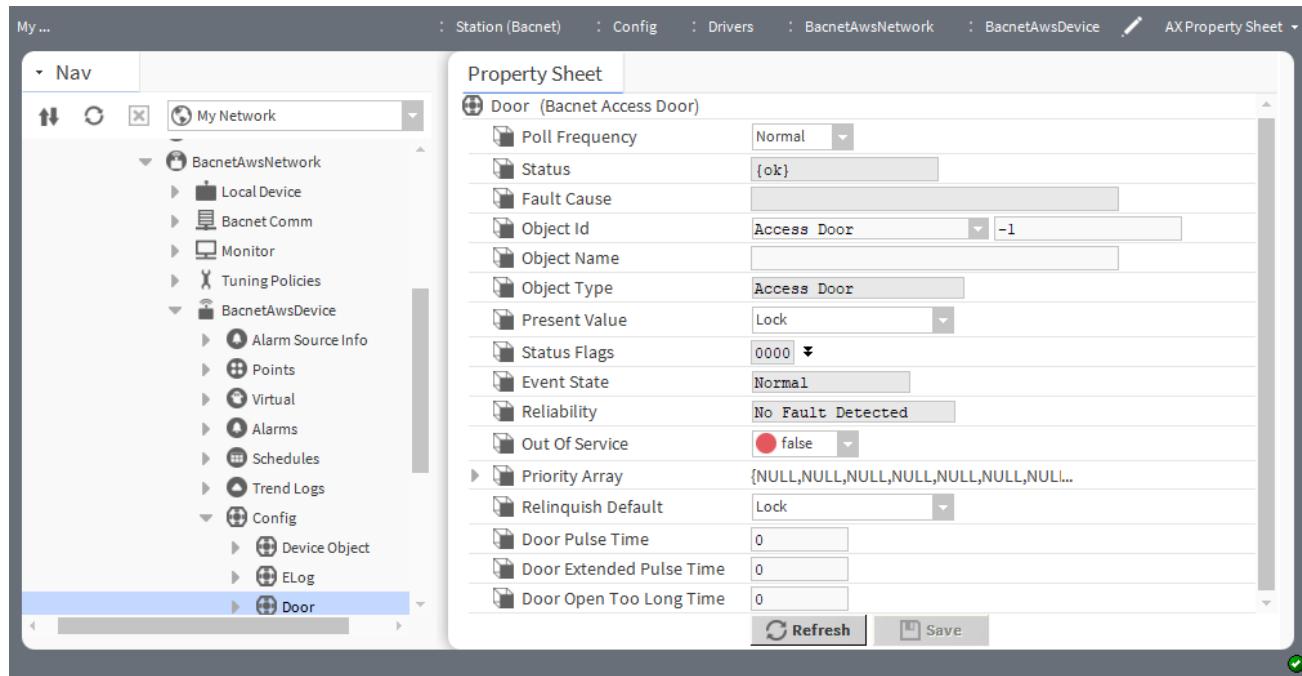
| Property | Value | Description |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |

| Property | Value | Description |
|---------------------------|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. NOTE: After creation (adding) BACnet file descriptors, note that each descriptor also has additional properties accessible in its property sheet, along with those seen in the Add and Edit window. |
| Enabled | true (default) or false | Activates (true) and deactivates (false) use of the object (network, device, point, component, table, schedule, descriptor, etc.). |
| Stop When Full | true or false (default) | Controls what happens when the trend log exceeds its capacity. |
| Buffer Size | number | Configures the size of a event log buffer. |
| Record Count | number | Configures the number of records to store in the trend log. |
| Log Buffer | number | Configures the number of records to store in the trend log. |
| Notify Type | (optional) Alarm (default) or Event | Indicates if the notification is from an alarm or an event. This property applies if the source component has an alarm extension. |
| Notification Class | number (defaults to -1) | Specifies the notification class used for routing when handling and generating event notifications for this object. |
| Object Property Reference | additional properties | Contains the properties related to the referenced object. |
| Event State | read-only | Reports the status of the event. |

Door

This component is a config object that represents a Bacnet Access Door object in its entirety.

Figure 145 Door properties



To access these properties, expand Drivers→BacnetAwsNetwork→BacnetAwsDevice→Config and double-click **Door**.

In addition to the common properties (Status and Fault Cause), this component has these properties.

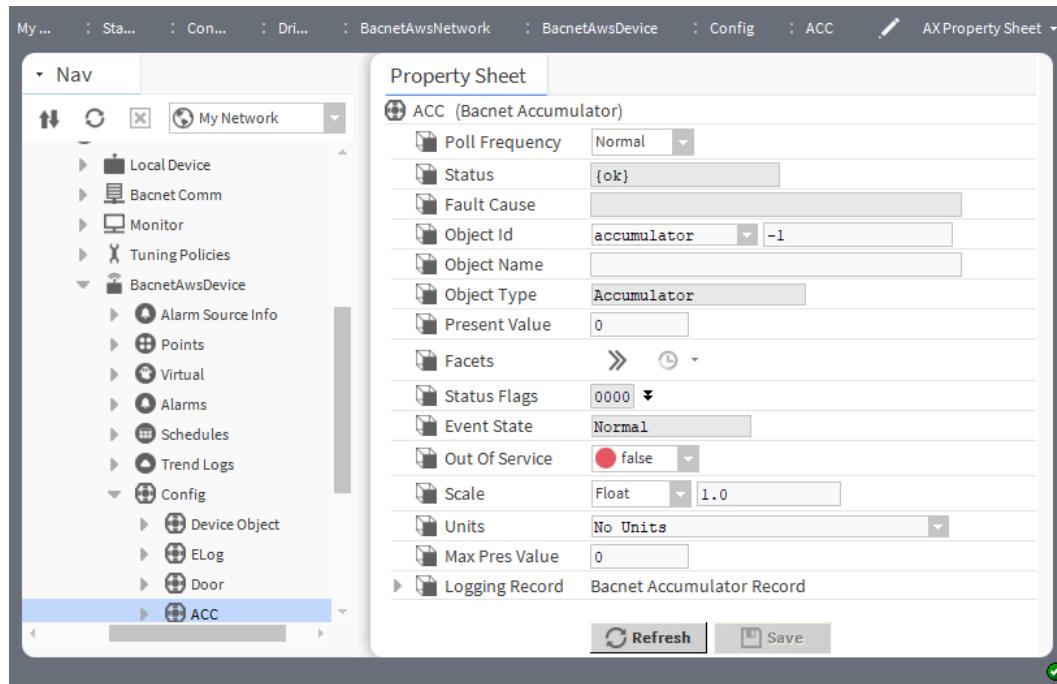
| Property | Value | Description |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork 's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |

| Property | Value | Description |
|--------------------------|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Status Flag | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Event State | read-only | Reports the status of the event. |
| Reliability | read-only | <p>Provides information when an error interrupts communication with a device.</p> <p>No Fault Detected indicates that things are OK.</p> <p>Unreliable Other indicates that the exported BACnet object appears in fault. This might happen, for example, if the source (exported) component is a proxy point under some other driver, and device communications are down or if the point has been disabled.</p> <p>In this case, the BACnet export descriptor retains a status of ok.</p> |
| Out of Service | true (default) or false | <p>Affects only the BACnet exposure and access to the source component. If the source (exported) component is a proxy point, the equivalent Enabled property in its ProxyExt is not affected.</p> <p>true writes the point's current value to the Present Value property. Often, a remote BACnet client sets (or clears) this Out of Service state.</p> <p>You can set this property to false if needed.</p> |
| Priority Array | array | Sets up an array of object identifiers that identify priority values. This array is for Notification Class priorities. |
| Door Pulse Time | number | |
| Door Extended Pulse Time | number | |
| Door Open Too Long Time | number | |

ACC

This component is a Config object that represents a Bacnet Accumulator object. The Accumulator object represents the characteristics of a meter that indicate measurements by counting pulses.

Figure 146 ACC properties



To access these properties, expand Drivers→BacnetAwsNetwork→BacnetAwsDevice→Config and double-click ACC.

In addition to the common properties (Status and Fault Cause), this component has these properties.

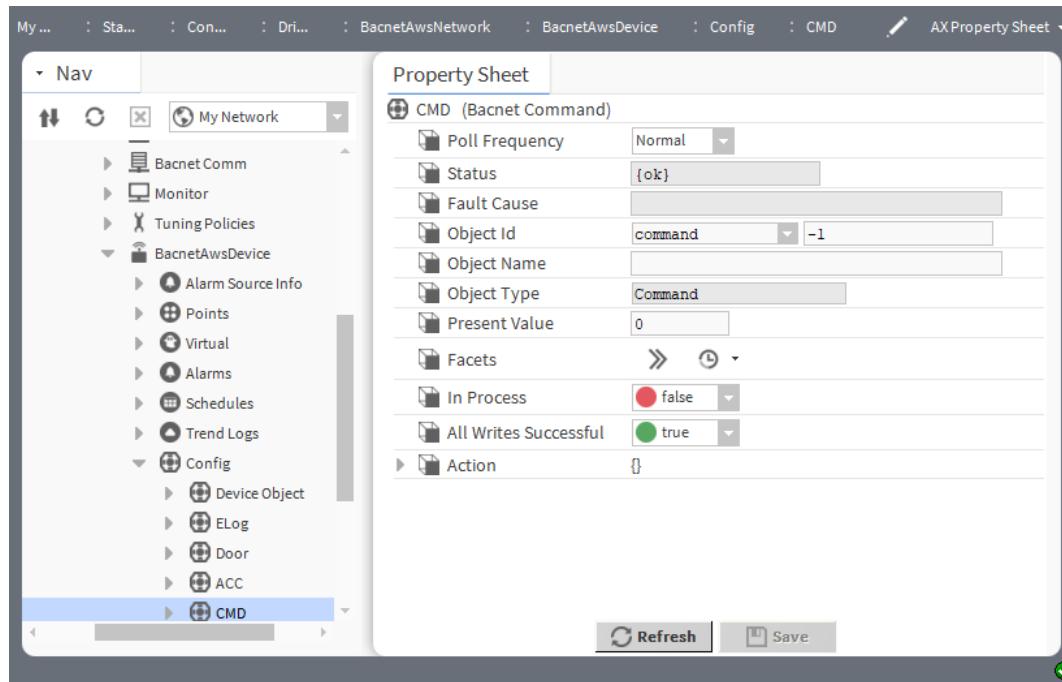
| Property | Value | Description |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number. For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name, however, each Object Name must be unique among all server descriptors under the Export Table. |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |

| Property | Value | Description |
|----------------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Facets | Config window or read-only property | Identifies the units used for the current component or point value. These could be parent or child point facets. |
| Enable | true (default) or false | Activates (true) and deactivates (false) use of the object (network, device, point, component, table, schedule, descriptor, etc.). |
| Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Event State | read-only | Reports the status of the event. |
| Out Of Service | true (default) or false | Affects only the BACnet exposure and access to the source component. If the source (exported) component is a proxy point, the equivalent Enabled property in its ProxyExt is not affected. true writes the point's current value to the Present Value property. Often, a remote BACnet client sets (or clears) this Out of Service state. You can set this property to false if needed. |
| Scale | drop-down list | |
| Max Pres Value | defaults to +inf | |
| Logging Record | additional properties | |

CMD

This component is a Config object that represents a Bacnet Command object in its entirety.

Figure 147 CMD properties



To access these properties, expand Drivers→BacnetAwsNetwork→BacnetAwsDevice→Config and double-click ELog.

The Bacnet Command object writes multiple values to multiple objects in multiple devices to complete a specific purpose.

In addition to the common properties (Status and Fault Cause), this component has these properties.

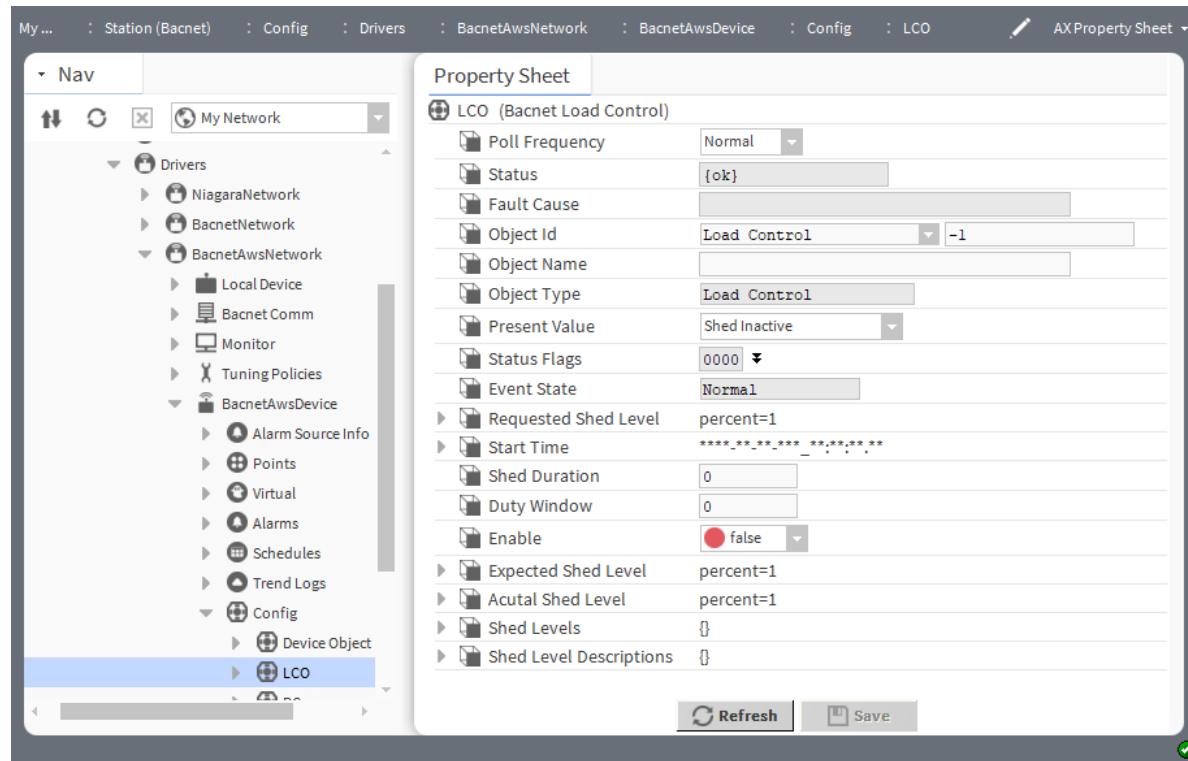
| Property | Value | Description |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number. For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name, however, each Object Name must be unique among all server descriptors under the Export Table. |

| Property | Value | Description |
|-----------------------|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| Object Type | drop-down list of BACnet objects | Reports the type of object. |
| Present Value | number | While Out Of Service is set to <code>true</code> , this property reflects the last written Present Value . |
| Facets | Config window or read-only property | Configures the units used for the current component or point value. These could be parent or child point facets. |
| In Process | <code>true</code> or <code>false</code> (default) | |
| All Writes Successful | <code>true</code> (default) or <code>false</code> | |
| Action | additional properties | |

LCO

This component is a Config object that represents a Bacnet Load Control object in its entirety.

Figure 148 LCO properties



To access these properties, expand **Drivers**→**BacnetAwsNetwork**→**BacnetAwsDevice**→**Config** and double-click **LCO**.

In addition to the common properties (Status and Fault Cause), this component has these properties.

The Bacnet Load Control object allows external control over the shedding of a load that it controls.

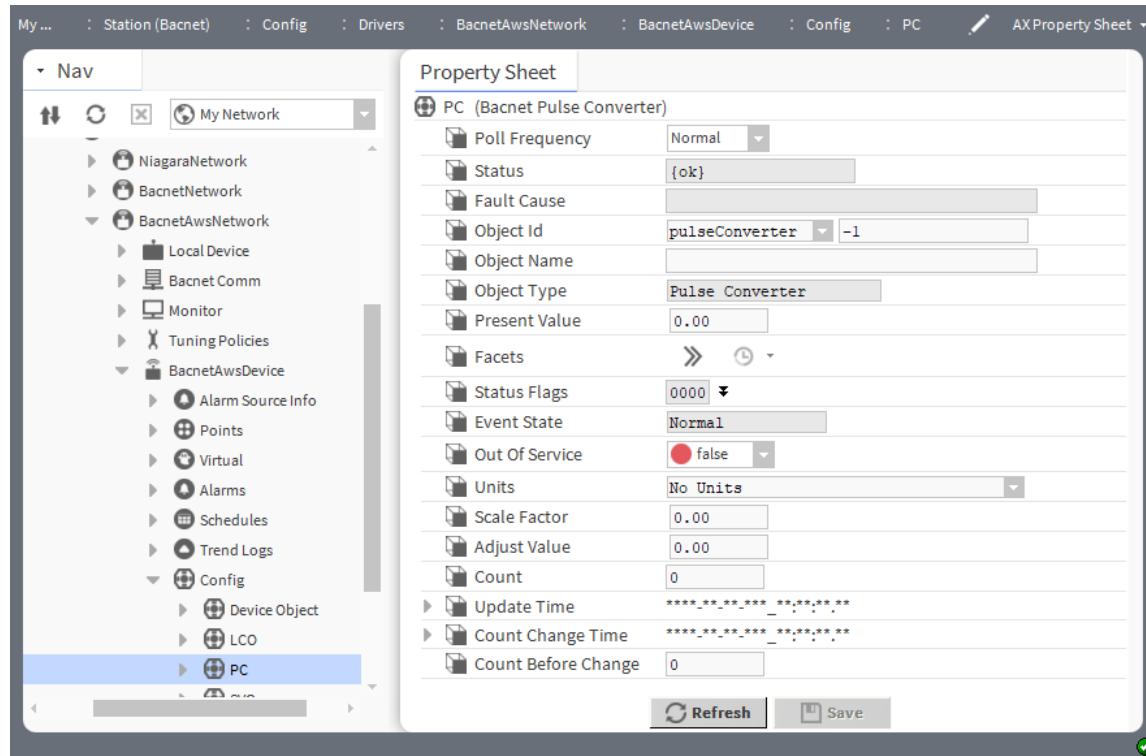
| Property | Value | Description |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | Identifies a BACnet object. This name should be unique within this specific device. For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device. For a proxy point, this is the default name in the station. You can shorten or edit Object Name , however, each Object Name must be unique among all server descriptors under the Export Table . |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |
| Present Value | true (default) or false | While Out Of Service is set to true, this property reflects the last written Present Value . |
| Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Event State | read-only | Reports the status of the event. |
| Requested Shed Level | additional properties | |
| Start Time | additional properties | Configures when to begin. |
| Shed Duration | number (defaults to zero (0)) | Configures shed duration. |
| Duty Window | number (defaults to zero (0)) | |
| Enable | true (default) or false | Activates (true) and deactivates (false) use of the object (network, device, point, component, table, schedule, descriptor, etc.). |
| Expected Shed Level | additional properties | |
| Actual Shed Level | additional properties | |

| Property | Value | Description |
|-------------------------|-------|-------------|
| Shed Levels | | |
| Shed Level Descriptions | | |

PC

This component is a Config object that represents a Bacnet Pulse Converter object in its entirety.

Figure 149 PC properties



To access these properties, expand **Drivers**→**BacnetAwsNetwork**→**BacnetAwsDevice**→**Config** and double-click **LCO**.

The Pulse Converter object monitors counts or pulses that typically represent a metered process, such as water or energy usage.

In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|----------------|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork 's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For |

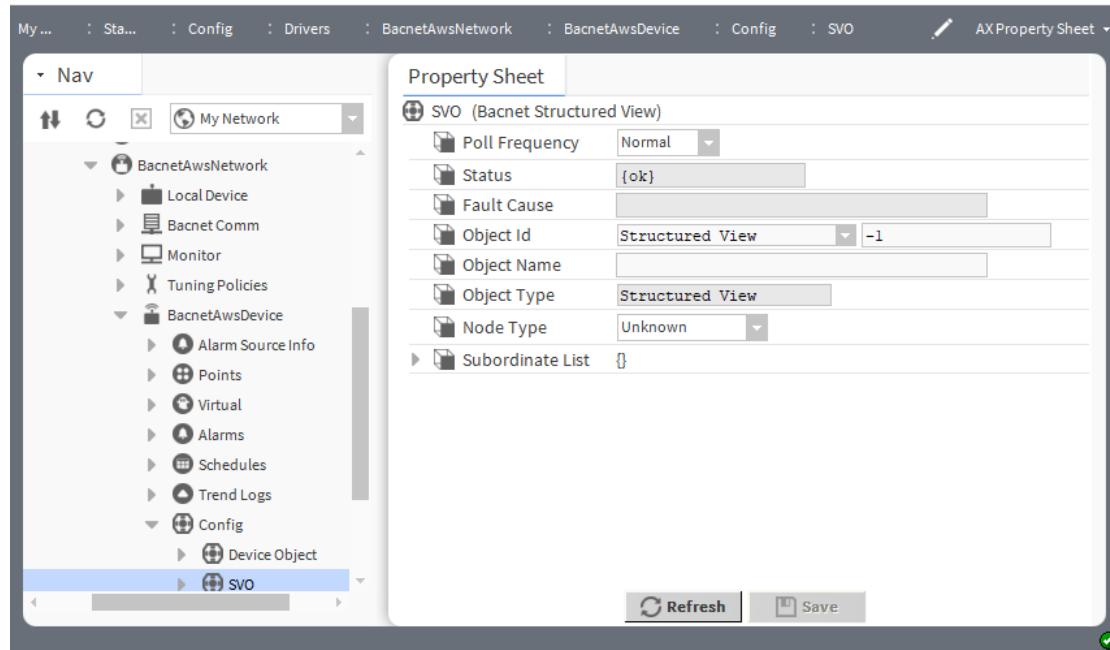
| Property | Value | Description |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | <p>Identifies a BACnet object. This name should be unique within this specific device.</p> <p>For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device.</p> <p>For a proxy point, this is the default name in the station.</p> <p>You can shorten or edit Object Name, however, each Object Name must be unique among all server descriptors under the Export Table.</p> |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |
| Facets | Config window or read-only property | <p>Determine how values are formatted for display depending on the context and the type of data. Examples include engineering units and decimal precision for numeric types, and descriptive value (state) text for boolean and enum types.</p> <p>With the exception of proxy points (with possible defined device facets), point facets do not affect how the framework processes the point's value.</p> <p>Besides control points, various other components have facets too. For example, many <code>kitControl</code> and schedule components have facets. Details about point facets apply to these components too, unless especially noted.</p> <p>You access facets by clicking an Edit button or a chevron >. Both open an Edit Facets window.</p> |
| Present value | number | While Out Of Service is set to <code>true</code> , this property reflects the last written Present Value . |
| Status Flags | read-only indicator and check boxes | Configures four status flags for BACnet. These options are available to enable detection of specific status conditions. |
| Event State | read-only | Reports the status of the event. |
| Out of Service | <code>true</code> (default) or <code>false</code> | <p>Affects only the BACnet exposure and access to the source component. If the source (exported) component is a proxy point, the equivalent Enabled property in its <code>ProxyExt</code> is not affected.</p> <p><code>true</code> writes the point's current value to the Present Value property. Often, a remote BACnet client sets (or clears) this Out of Service state.</p> <p>You can set this property to <code>false</code> if needed.</p> |
| Units | read-only | Reports the unit of measure for the entity. |
| Scale Factor | | |
| Adjust Value | | |

| Property | Value | Description |
|---------------------|-----------------------|-------------|
| Count | | |
| Update Time | additional properties | |
| Count Change Time | additional properties | |
| Count Before Change | | |

SVO

This component is a Config object that represents a BACnet structured view object. This type of object acts as a container for references to other BACnet objects, possibly including other structured view objects. It provides an organizational (and potentially hierarchical) interface to BACnet objects in a system.

Figure 150 SVO properties



To access these properties, expand **Drivers**→**BacnetAwsNetwork**→**BacnetAwsDevice**→**Config** and double-click **LCO**.

In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|----------------|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poll Frequency | drop-down list (defaults to Normal) | References the On Demand Poll Scheduler rates under the NiagaraNetwork 's History Policies container slot. Configures how often this type of poll occurs. |
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For |

| Property | Value | Description |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Object Name | text in the format name.name.name (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | <p>Identifies a BACnet object. This name should be unique within this specific device.</p> <p>For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device.</p> <p>For a proxy point, this is the default name in the station.</p> <p>You can shorten or edit Object Name, however, each Object Name must be unique among all server descriptors under the Export Table.</p> |
| Object Type | drop-down list of BACnet objects | Reports the type of object. |
| Node Type | drop-down list, defaults to unknown | <p>Classifies a BACnet object (node) as:</p> <p>Unknown System Network Device Organizational Area Equipment Point Collection Property Functional Other</p> |
| Subordinate List | additional properties | Refers to the structured view objects, which are available to other structured view objects. |

Chapter 12 Components in the bacnetOws module

Topics covered in this chapter

- ◆ BacnetOwsNetwork
- ◆ BacnetOwsDeviceFolder

Components include services, folders and other model building blocks associated with a module. You may drag them to a property or wire sheet from a palette. The components of the **bacnetOws** module support BACnet features running in a Supervisor PC.

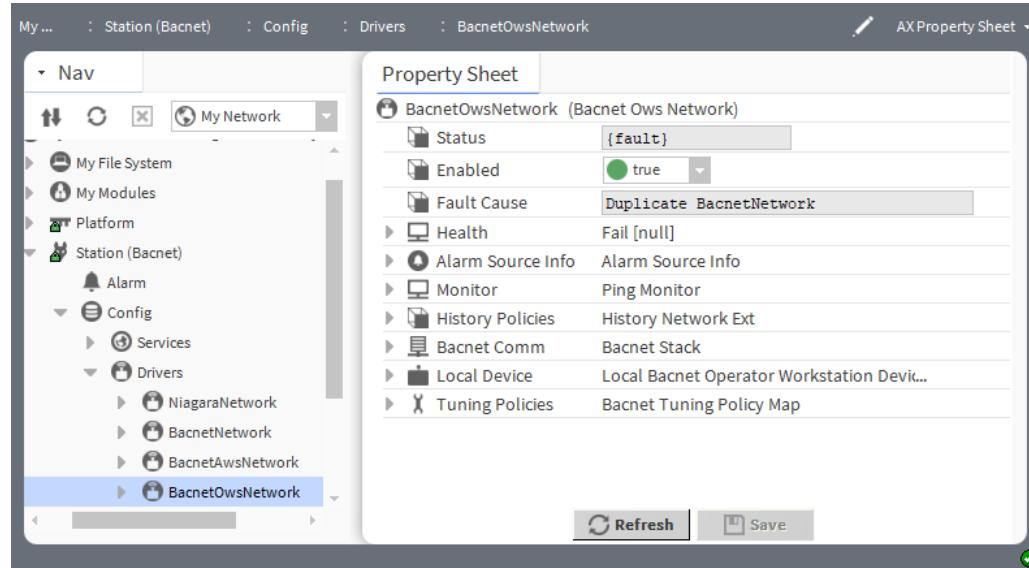
Descriptions included in the following topics appear as context-sensitive help topics when accessed by:

- Right-clicking on the object and selecting **Views→Guide Help**
- Clicking **Help→Guide On Target**

BacnetOwsNetwork

This component is the base container for all BACnet components (devices and proxy points). It resides under the station's **Drivers** container. Its default view is the **Bacnet Ows Device Manager**. In addition to devices and proxy points, this component contains the station's BACnet communications protocol stack (Bacnet Comm), and a Local Bacnet Device, which configures the station's representation as a BACnet device.

Figure 151 BacnetOwsNetwork properties



You access these properties by clicking **Tools→Bacnet Service**, by expanding the **BacnetOwsNetwork** in the Nav bar and double-clicking **Bacnet Comm**.

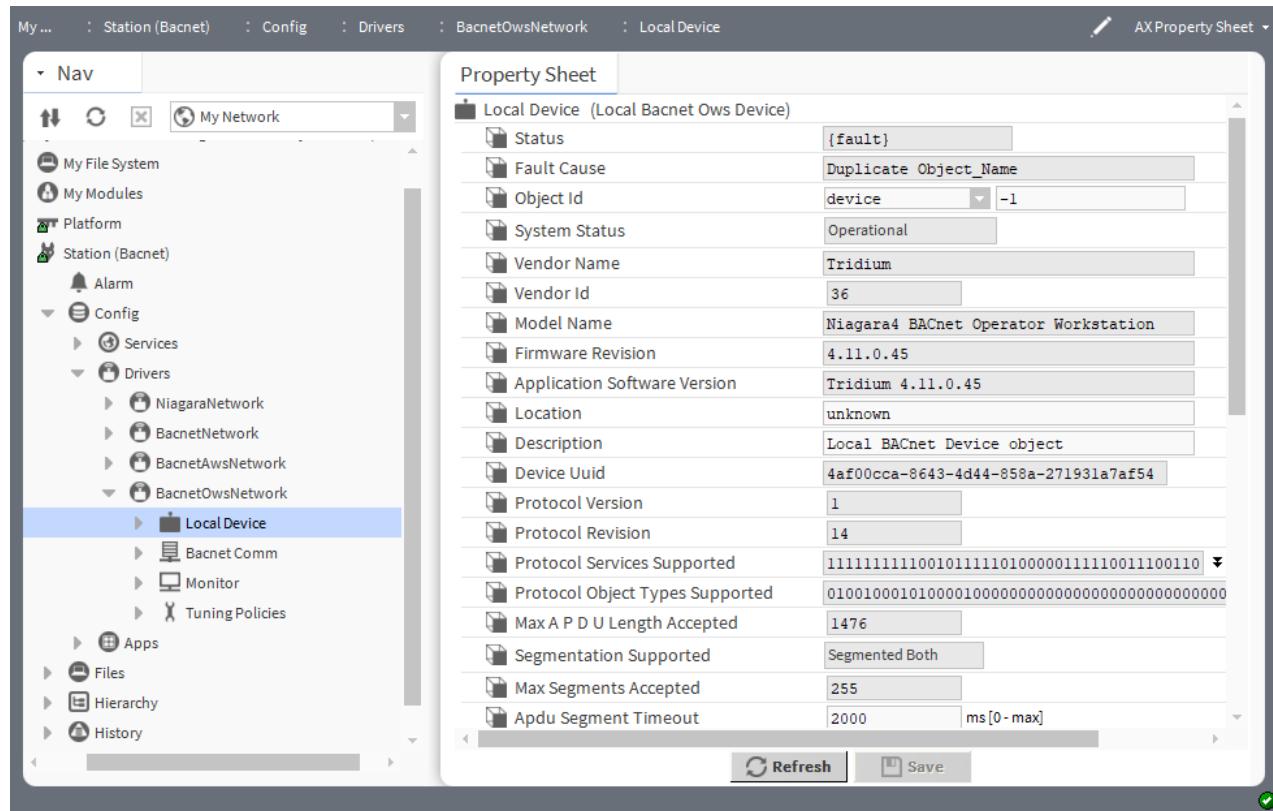
In addition to the common properties (Status, Enabled, Health and Fault Cause), this component has these properties.

| Property | Value | Description |
|-------------------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Alarm Source Info | additional properties | Contains a set of properties for configuring and routing alarms when this component is the alarm source. For property descriptions, refer to the <i>Niagara Alarms Guide</i> . |
| Monitor | ping monitor | Configures a network's ping mechanism, which verifies network health. This includes verifying the health of all connected objects (typically, devices) by pinging each device at a repeated interval. The <i>Niagara Drivers Guide</i> documents these properties. |
| History Policies | additional properties | Configures the history network extension, which is documented in a separate topic. |
| Bacnet Comm | additional properties | Contains BACnet communication properties, which are documented in a separate topic. |
| Local Device | additional properties | Contains device properties, which are documented in a separate topic. |
| Tuning Policies | additional properties | Configures network rules for evaluating both write requests to writable proxy points as well as the acceptable freshness of read requests. For more information, refer to the <i>Niagara Drivers Guide</i> . |

Local Device

This frozen container under the BacnetOwsNetwork represents a BACnet station on the BACnet internet-work. Its child BacnetExportTable container provides server-side functions to export station objects as BACnet objects and service BACnet client requests. In general, this means that it provides BACnet server responses to remote client requests from other BACnet devices on a connected network.

Figure 152 Local Bacnet Ows Device properties



You access local device properties by expanding **Drivers**→**BacnetOwsNetwork** in the Nav tree and double-clicking the **Local Device** node.

The Local Bacnet Aws Device has all the same properties and components of a regular BACnet Local Device. In addition to the common properties (Status and Fault Cause), this component has these properties.

| Property | Value | Description |
|------------------------------|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Object Id | drop-down list and number (from 0 to 4194302) | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| System Status | read-only | Reports the current condition of the network. |
| Vendor Name | read-only | Reports the vendor number. |
| Vendor Id | read-only | Reports an identifier for the vendor's device. |
| Model Name | read-only | Reports the manufacturer's model name for the device. |
| Firmware Revision | read-only | Reports the version of the device firmware. |
| Application Software Version | read-only | Reports the version of the manufacturer's software running on the device. |

| Property | Value | Description |
|---------------------------------------------|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Location | text (defaults to unknown) | Describes the device location. |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. |
| Device Uuid (Universally Unique Identifier) | read-only | Uniquely identifies the object in the system. When you upgrade a BACnet driver or drop in a BACnet network from the palette, the driver automatically sets this 128-bit universally unique identifier (UUID) to a random value. You should not need to change this property. The driver uses it behind the scenes. |
| Protocol Version | read-only | Displays the version number of the BACnet protocol. |
| Protocol Revision | read-only | Displays the current BACnet protocol level. |
| Protocol Services Supported | check box list | Expands to identify all the available BACnet device services supported by this version of Niagara. |
| Protocol Object Types Supported | check box list | Expands to identify all the available BACnet object types supported by this version of Niagara. |
| Max A P D U Length Accepted | read-only | Reports the number of points per poll or APDU (Application Protocol Data Units). |
| Segmentation Supported | read-only | Indicates the type of message segmentation the device supports. If a message exceeds the Max Segment Accepted length, the device must be able to break it into segments or part of the message is lost. |
| Max Segments Accepted | read-only | For a single segment, defines the number of characters the device supports. |
| Apdu Segment Timeout | milliseconds (defaults to 2000 ms) | Defines the time the system waits before retransmitting an APDU segment. This property is only relevant if devices are doing segmentation. Works for both client and server operations. For a small IP-based network, an optimal setting may be: 5000 ms. |
| Apdu Timeout | milliseconds (defaults to 3000 ms) | Defines how long the system waits before retransmitting an APDU that requires acknowledgment, for which no acknowledgement has been received. Works for both client and server operations. Large, routed networks may require higher timeouts. |
| Number of Apdu Retries | number (defaults to 3) | Defines the maximum number APDU retransmissions. Works for both client and server operations. For a small IP-based network, an optimal setting may be: 1 retry. |
| Database Revision | read-only | Reports the version of the database. |
| Last Restore Time | read-only | Reports when a backup .dist file was last restored by a remote BACnet Workstation client. If never, all timestamp properties show asterisks (*). |

| Property | Value | Description |
|-------------------------------------|----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Backup Failure Timeout | time range (defaults to 3 minutes) | Specifies the time the server device must wait before deciding that the client has given up, and can leave backup/restore mode. The client writes this to the server prior to beginning the backup/restore procedure. |
| Backup Preparation Time | time range (defaults to 1 minute) | Reports the time taken, after receiving a restore request, to gracefully shut down the station and close files before beginning the backup. |
| Restore Preparation Time | | Reports the time taken, after receiving a restore request, to gracefully shut down the station and close files before beginning the dist file installation. |
| Restore Completion Time | timestamp (defaults to 3 minutes) | Reports the time taken, after completing a dist file install from a restore, for the station to restart and become responsive. |
| Backup and Restore State | read-only | Shows the current status of any backup or restore operation. The driver displays <code>Idle</code> when no operation is active. This changes to <code>Preparing for backup</code> and so forth. |
| Character Set | drop-down list (defaults to Ansi C3_4) | Defines the character set supported, with other selections: IBM/Microsoft DBCS, JIS C 6226, ISO 10646 (UCS-4), ISO 10646 (UCS-2), ISO 8859-1, and Unknown. |
| Enumeration List | sets of enum facets | Lists the BACnet properties that can be exported to configure data exchange. This Extensible Enum List provides access to each property's configuration facets. |
| Export Table | Bacnet Export Manager | Provides the special manager views to associate devices and points (local objects) with BACnet objects (exported objects). You discover objects or add them using the New button. The <i>Bacnet Export Manager</i> (plugin) topic documents this table. |
| Time Synchronization Recipients | Bacnet List Of | Define the devices to receive time synchronization messages from the driver. This list of recipients is for periodic time synchronization messages. You add to and remove entries from this list using the right-click <code>addElement</code> and <code>removeElement</code> actions. Entries in this list are BacnetRecipients . They can be either a device object identifier, or a BACnet address. |
| Time Synchronization Interval | timestamp (defaults to 24 hours) | Configures how frequently the software sends the time synchronization messages. |
| Align Intervals | true (defaults) or false | Turns on and off the configuration of standard synchronization intervals for controller stations. <code>true</code> sets up the same time synchronization interval in each host controller station. <code>false</code> allows a different time synchronization interval for each host controller station. |
| Interval Offset | minutes | Specifies an offset from the beginning of the interval at which to send the periodic time synchronization messages. If Align Interval is <code>false</code> , the driver ignores this property. |
| Utc Time Synchronization Recipients | Bacnet List Of | Lists the recipients for periodic Time Synchronization messages. You can add and remove entries to this list using the |

| Property | Value | Description |
|---------------------------------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | right-click addElement and removeElement actions. Entries in this list are BacnetRecipients. They can be either a device object identifier, or a BACnet address. |
| Last Restart Reason | drop-down list | Reports the reason for the last restart. |
| Time of Device Restart | date and time | Specifies the time of device restart based on the time of day (Time), a sequence number (SeqNum) or a date and time (DateTime). |
| Restart Notification Recipients | Bacnet List Of | Defines the alarm recipients to receive notification that the station restarted. |

BacnetOwsDeviceFolder

This component implements a folder under a BacnetOwsNetwork.

Typically, you add such folders using the **New Folder** button in the **Bacnet Ows Device Manager** view of the BacnetOwsNetwork. Each BacnetOwsDeviceFolder has its own **Bacnet Ows Device Manager** view. This folder is also available in the **bacnetOws** palette.

Chapter 13 Components in the histories module

Topics covered in this chapter

- ◆ history-HistoryConfig

Components include services, folders and other model building blocks associated with a module.

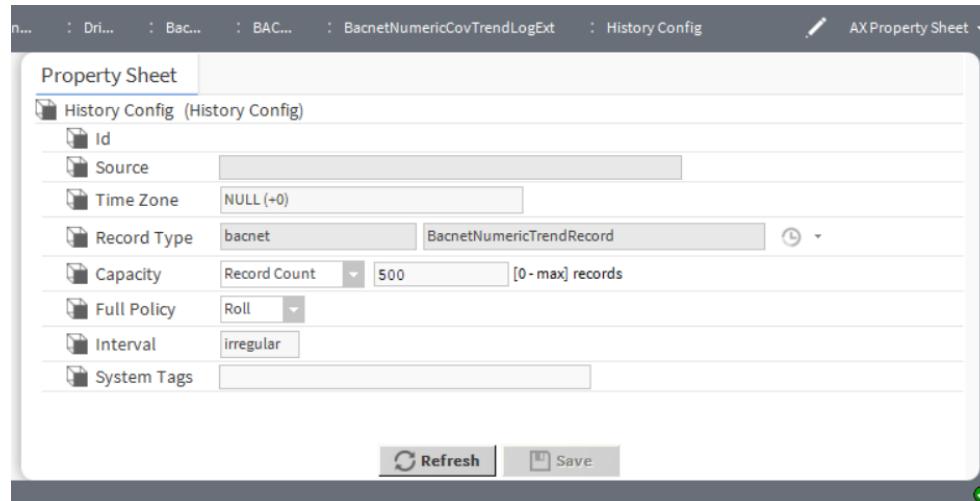
Descriptions included in the following topics appear as context-sensitive help topics when accessed by:

- Right-clicking on the object and selecting **Views→Guide Help**
- Clicking **Help→Guide On Target**

history-HistoryConfig

This component configures a history in the History database.

Figure 153 History Config properties



This following table summarizes the **History Config** properties.

| Property | Value | Description |
|----------------|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| History Config | additional properties | Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties. |
| Id | read-only | Displays string results from a value configured in history extension's History Name property. An error string here indicates the History Name property is incorrectly configured. |
| Source | read-only | Displays the ORD of the active history extension. |
| Time zone | Text String | Displays the time zone of the active history extension. |

| Property | Value | Description |
|-------------|--------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Record type | Text | Displays the data that the record holds in terms of: extension type (history) and data type (BooleanTrendRecord, NumericTrendRecord, and so on). |
| Capacity | Record Count: nnn (500 default), Unlimited | Specifies local storage capacity for histories. In general, 500 (default record count) or less is adequate for a controller station because those records are usually archived (exported) to a Supervisor station. For this reason, a very large number, such as 250,000 is acceptable for Supervisor stations. Unlimited is not the wisest choice even for a Supervisor station. |
| Full Policy | Roll (default), Stop | Applies only if Capacity is set to Record Count. When capacity reaches record count, the newest records overwrite the oldest records. |
| Interval | Text string | Defines, for Interval-based data collection, the cycle time, or how often the history properties are checked. Changing this property creates a new history (or splits-off) from the original history because histories with different intervals are not compatible. |
| System tags | text | Assigns additional metadata (the System Tag) to a history extension. This identifier is then available for selective import or export of histories using the Niagara System History Import or Niagara System History Export option (using the System Tag Patterns). Each System Tag is separated by a semicolon. For example: NorthAmerica;Region1;Cities. |

Chapter 14 Views

Topics covered in this chapter

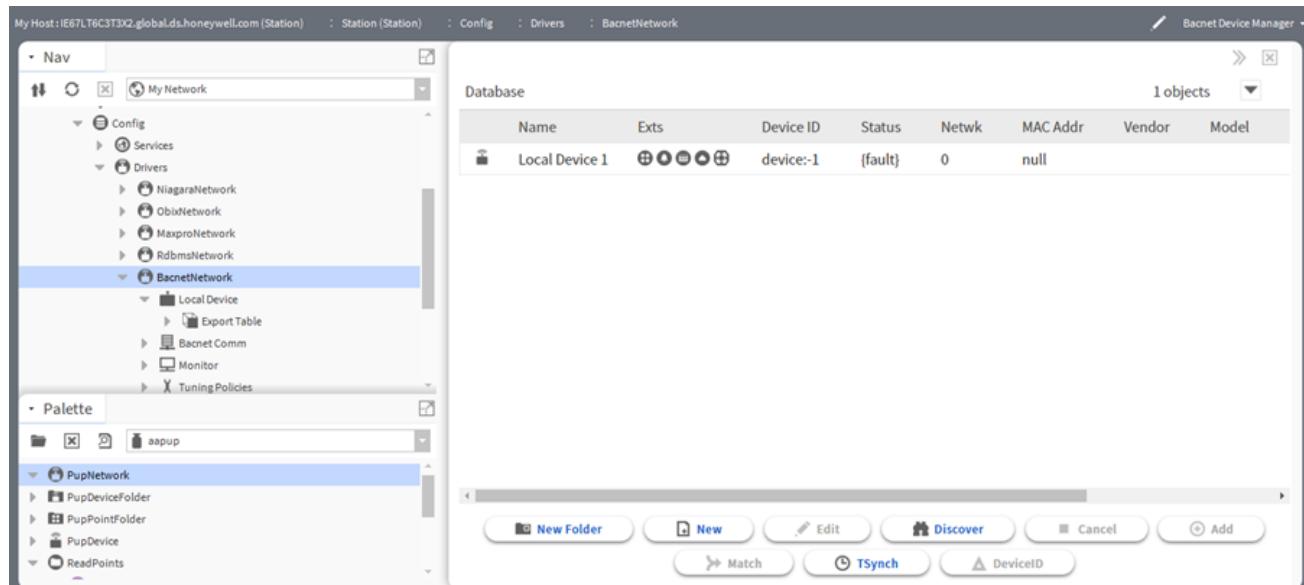
- ◆ Bacnet Device Manager
- ◆ EDE Bacnet Device Manager
- ◆ Bacnet Point Manager
- ◆ EDE Bacnet Point Manager
- ◆ Bacnet Export Manager (bacnet-BacnetExportManager)
- ◆ Svo Subordinate Manager view
- ◆ Bacnet File Export Manager
- ◆ Bacnet Niagara Log Export Manager
- ◆ Bacnet Schedule Import Manager
- ◆ Bacnet Schedule Export Manager
- ◆ Bacnet History Import Manager
- ◆ Device Histories view
- ◆ Bacnet Trend Multiple View
- ◆ Bacnet Config Manager
- ◆ Bacnet Aws Device Manager
- ◆ Bacnet Aws Config Manager
- ◆ Bdt Manager
- ◆ Fdt Manager
- ◆ Bacnet Ows Device Manager
- ◆ HTML-5 Bacnet Ows Device Manager
- ◆ HTML 5 Bacnet Device UxManager View

Plugins provide views of components and can be accessed in many ways. For example, double-click a component in the Nav tree to see its default view. In addition, you can right-click on a component and select from its **Views** menu.

For summary documentation on any view, select **Help→On View (F1)** from the menu or press **F1** while the view is open.

Bacnet Device Manager

This is the default view for the BacnetNetwork, and works in a similar manner to other device managers that support online device discovery.

Figure 154 Bacnet Device Manager

To open this view, expand **Config**→**Drivers** and double-click **BacnetNetwork** or right-click **BacnetNetwork** and click **Views**→**Bacnet Device Manager**.

Discovered pane

As in other manager views, when you click **Discover** this manager goes into Learn mode, opens two panes, and executes a discover job. Discover returns multiple objects, each of which occupies one row in the table.

| Column | Description |
|--------------|--------------------------------------------------|
| Device Name | Displays the device name. |
| Device ID | Displays the device identifier. |
| Network | Displays the network number. |
| MAC Addr | Displays the Media Access Control (MAC) address. |
| Vendor | Displays the name of device manufacturer. |
| Model | Displays the device model name. |
| Firmware Rev | Displays the firmware version in use. |

Database pane

This pane displays the newly added devices either from Discovered pane or by using **New** button.

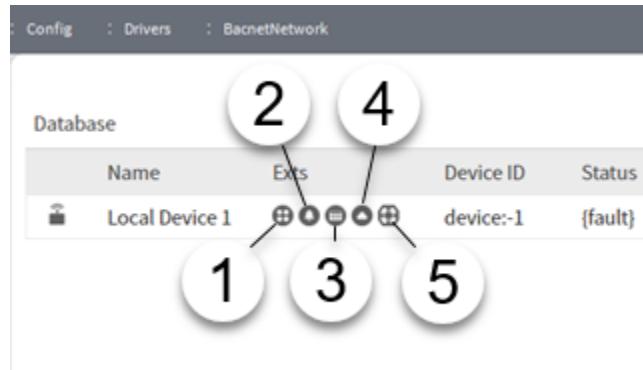
| Column | Description |
|-----------|----------------------------------------------------------------------------------------------------------|
| Name | Displays the name of the BACnet device. |
| Type | Displays the type of BACnet device. |
| Exts | Displays the device extension's hyperlinks, including: Points, Alarms, Schedules, Trend Logs and Config. |
| Device ID | Displays the assigned ID to the device. |
| Status | Display the current status of the device. |

| Column | Description |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------|
| Netwk | Displays the network number. |
| MAC Addr | Displays the Media Access Control (MAC) address. |
| MAC APDU | Displays the Media Access Control (MAC) Application Protocol Data Units (APDU) for the device. |
| Segmentation | Indicates if message segmentation was required. If a BACnet message does not fit in a single packet, the driver uses segmentation. |
| Vendor | Displays the name of the device manufacturer. |
| Model | Displays the device model name. |
| Firmware Rev | Displays the version of firmware in use. |
| App SW Version | Indicates the application software version. |
| Encoding | Displays the name of the universal standard used for coding multilingual text. |
| Health | Displays the status of the device. |
| Enabled | Indicates if the device is online (enabled) or not (disabled). |
| Use Cov | Indicates if change-of-value was used (true) to monitor proxy point values. |
| Use Cov Property | Indicates if the tuning policy processed COV notification (true). |
| Max Cov Subscriptions | Displays the maximum number of COV subscriptions that the database attempts to use with this device. |
| Cov Subscriptions | Displays the number of COV subscriptions the database used with this device. |

Exts—Extension hyperlinks

Double-clicking an extension hyperlink opens a **Property Sheet** or other view.

Figure 155 Extension hyperlinks



1. Points extension hyperlink: opens the **Bacnet Point Manager** view.
2. Alarms extension hyperlink opens the **Alarms Property Sheet**.
3. Schedule extension hyperlink opens the **Bacnet Schedule Import Manager**.
4. History extension hyperlink opens the **Bacnet History Import Manager**.
5. Config extension hyperlink opens the **Bacnet Config Manager**.

Buttons

- **New Folder** creates a new folder for devices. Each such folder provides its own set of manager views.

- **New** creates a new device record in the database.
- **Edit** opens the device's database record for updating.
- **Discover** runs a discover job to locate installed devices, which appear in the **Discovered** pane. This view has a standard appearance that is similar to all **Device Manager** views.
- **Cancel** ends the current discovery job.
- **Add** inserts into the database a record for the discovered and selected object.
- **Match** associates a discovered device with a record that is already in the database.
- **TSync** synchronizes the time.
- **DeviceID** changes the device ID in the device.

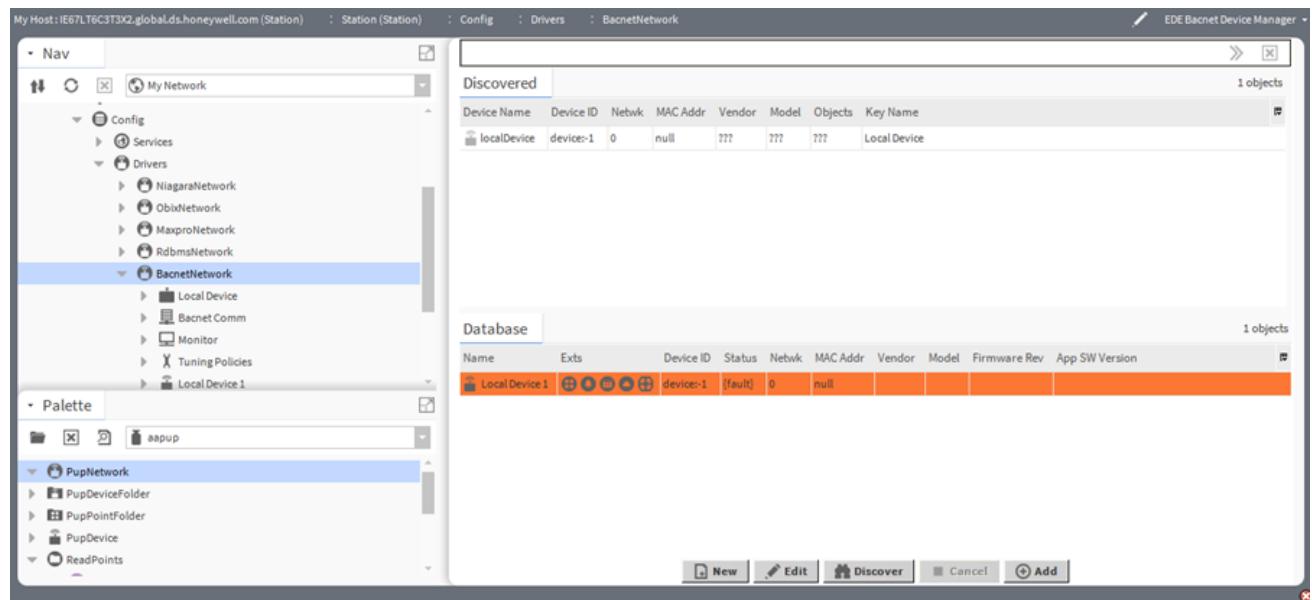
Manager menu

These actions are available on this view's Manager menu.

| Menu item | Description |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| New Folder | Creates a new folder (same function as the New Folder button). |
| All Descendents | Expands the selected folder to reveal all subordinate objects. |
| New | Creates a new device record in the database. |
| Edit | Opens the Edit window for updating the selected record. |
| Learn Mode | Controls the discovery of devices and adding them to the database. |
| Discover | Initiates a discovery job (same as clicking the Discover button). |
| Cancel | Stops the current job. |
| Add | Opens an Add window with properties to configure before adding the selected record(s) in the Discovery pane to the database. |
| Quick Add | Adds the selected record(s) in the Discovery pane to the database. |
| Match | Opens a window with properties to configure before matching the discovered record with a record in the database. |
| Quick Match | Matches the selected record with its sister record that is already in the database. |
| DevTemplate Mode | Opens a Templates pane. The <i>Niagara Templates Guide</i> documents templates. |
| Who Has | Opens a window for searching for objects by name. |
| TSynch | Aligns the current time in the device with the time maintained by the network. |
| Tag Mode | Opens a Tag Dictionary pane for managing metadata tags. The <i>Niagara Tagging Guide</i> documents tag. |
| TagIt | Associates an item of metadata, such as location or unique configuration, with the object. |
| Template Config | Accesses the station template that defines configuration options. |

EDE Bacnet Device Manager

This EDE (Engineering Data Exchange) view works in a similar manner to other device managers that support online device discovery. The **bacnet** module provides this manager.

Figure 156 EDE Bacnet Device Manager

To open this view, expand **Config→Drivers**, right-click **BacnetNetwork** and click **Views→EDE Bacnet Device Manager**.

Discovered pane

As in other manager views, when you click **Discover** this manager goes to Learn mode, splits into two panes, and executes a discover job. Discover returns multiple objects, each of which occupies one row in the table.

| Column name | Description |
|--------------|--------------------------------------------------|
| Device name | Displays the device name. |
| Device ID | Displays the device ID. |
| Network | Displays the network number. |
| MAC Addr | Displays the Media Access Control (MAC) address. |
| Vendor | Displays the name of device manufacturer. |
| Model | Displays the device model name. |
| Objects | Displays BACnet objects. |
| Firmware Rev | Displays the firmware version in use. |

Database pane

This section displays the newly added devices either from Discovered pane or by using **New** button.

| Column name | Description |
|-------------|----------------------------------------------------------------------------------------------------------|
| Name | Displays the name of the BACnet device. |
| Type | Displays the device type. |
| Ext | Displays the device extension's hyperlinks, including: Points, Alarms, Schedules, Trend Logs and Config. |
| Device Id | Displays the assigned ID to the device. |

| Column name | Description |
|-------------------|--------------------------------------------------|
| Status | Display the current status of the device. |
| Network | Displays the network number. |
| MAC Address | Displays the Media Access Control (MAC) address. |
| Vendor | Displays the name of the device manufacturer. |
| Firmware revision | Displays the version of firmware in use. |
| Model | Displays the device model name. |
| Health | Displays the current status of the device. |
| Enabled | Displays the enabled status. |
| Use COV | Displays the status of COV property. |

Buttons

- **New** creates a new device record in the database.
- **Edit** opens the device's database record for updating.
- **Discover** runs a discover job to locate installed devices, which appear in the **Discovered** pane. This view has a standard appearance that is similar to all **Device Manager** views.
- **Cancel** ends the current discovery job.
- **Add** inserts into the database a record for the discovered and selected object.

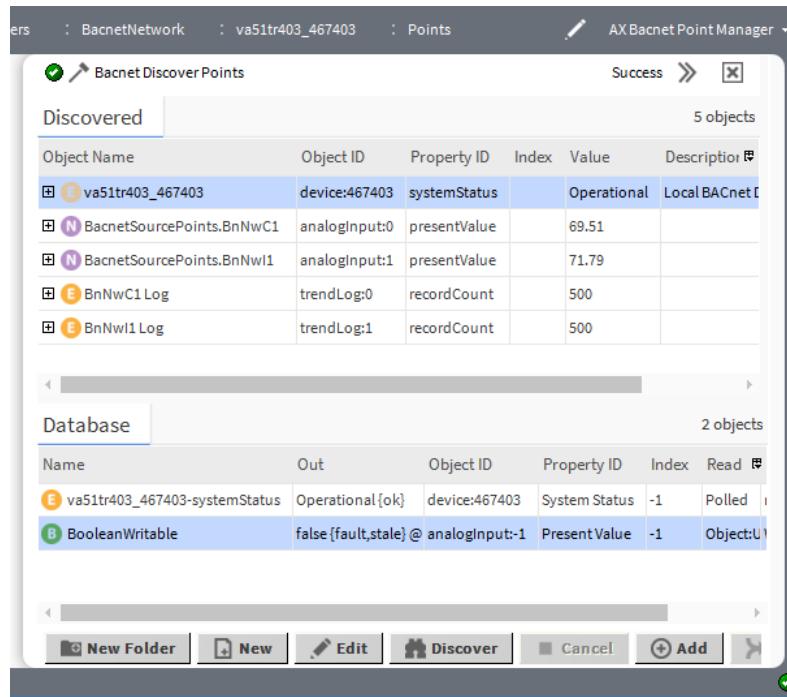
Bacnet Point Manager

This is the default view for the **Points** extension (or **Points** folder) under any **BacnetDevice**, and works similarly to the way other point managers work, which support online point discovery. Refer to the *Niagara Drivers Guide* for general details.

Discovered pane

By default, the **Discovered** table for BACnet objects lists the single BACnet device object at top, and other objects underneath, with each object initially occupying a single row.

Figure 157 Discovered BACnet points showing present values



To open this view, expand **Config→Drivers→BacnetNetwork→Bacnet Device** and double-click on the **Points Folder**.

To sort objects click on any column header. To group like type objects together, sort by **Object ID** (BACnet object type).

To select other properties (apart from **Present_Value**) as a candidate to proxy, click the plus (+) icon to expand the object.

Discovered pane

| Column | Description |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Object Name | Displays the name of the discovered BACnet object. This name is unique within this specific device. |
| Object ID | Displays the identifier for the discovered BACnet object. This number is a combination of BACnet object type and instance number (unique within the object type). |
| Property ID | Reports the Property ID for the discovered BACnet object. By default, all objects that have a Present_Value property are listed with this value on top. BACnet objects without a Present_Value property (for example, a device object or trend log object) show another property as the top Property ID . Typically, present value is the most useful piece of data from any BACnet object. However, you can expand any discovered object to see all its properties as children in the Discovered pane, where each one is a separate proxy point candidate. Additionally, properties that are arrays are further expandable. |
| Index | Reports a number that points to a value in an array, if selected, otherwise this column cell is blank. For example, if you expand a priority-type object, such as object type binaryOutput , and expand again on its property array, each child row displays with a unique index number (1-16). NOTE: Use this to proxy a point to write to only one specific level of the Priority_Array property of the target BACnet object (instead of accessing all levels, by proxy of only the default presentValue property). This may be useful in your control scheme, if you have a number of possible sources in the station you wish to evaluate (on a priority basis), to write to one BACnet priority level (only) in the target BACnet object. |

| Column | Description |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Value | Reports the static value of the associated property, captured when the system retrieved the object list (or, for any non-default properties, when you expanded the top property). Discovered values display showing any descriptors associated with the BACnet object's related properties (Units, Active_Text, Inactive_Text, and so on). |
| Description | Reports any available character string Description for the BACnet object. In many cases, this value may be blank. |

Database pane

The **Database** table contains existing BACnet proxy points, where each appears as a row in the table. Each proxy point represents one data item from a specific BACnet object in that device.

Figure 158 Database shows Bacnet proxy points in the station

| Name | Out | Object ID | Property ID | Index | Read | Write |
|-----------------------------------------------------------|-------------------|---------------|---------------|-------|--------------|-------|
| B Control_Logic.Heating_System.BoilerControl.Boiler_1 | off{down,stale} | binaryInput:2 | Present Value | -1 | Unsubscribed | R |
| B Control_Logic.Heating_System.PumpControl.PumpStatusOR | false{down,stale} | binaryInput:1 | Present Value | -1 | Unsubscribed | R |
| N Control_Logic.Heating_System.BoilerControl.HotWaterTemp | 0.0°C{down,stale} | analogInput:2 | Present Value | -1 | Unsubscribed | R |
| N Control_Logic.Heating_System.BoilerControl.LoopPoint | 0.0{down,stale} | analogInput:4 | Present Value | -1 | Unsubscribed | R |

To sort points, click on any column header. Often, this is useful to sort by **Object ID** (BACnet object type) or, perhaps, by name.

If you created **BacnetPointFolders** under a device's **Points** container, use the **All Descendents** tool to view all proxy points in the device from the main (root) **Device Points Manager**. For details, refer to the **Drivers Guide**.

| Column | Description |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Path | Reports the station path to the proxy point. |
| Name | Reports the name for the BACnet proxy point. If you added the point from a discover (selecting the default presentValue property), and did not edit Object Name , this name is identical to the BACnet object's name. As needed, you can edit this value. |
| Type | Reports the type of point that the proxy is based upon (BooleanWritable, NumericPoint, etc.). |
| Out | Reports the current out value, including any point facets. This defaults to the single (configured) property value along with status for the proxy point. Since BACnet supports its own native status in addition to the Framework's status, you can edit point facets to poll for additional properties, such as the native status Flags and/or priority array levels. |
| Enabled | Reports if the proxy point is currently enabled for communication. |
| Object ID | Reports the Object Id property of the proxied BACnet object, which is a combination of BACnet object type and instance number (unique within that type). In this column, these two properties appear separated by a colon (:), using descriptors for type. For example: |
| | <ul style="list-style-type: none"> • analogInput:3 • multiStateValue:3 • binaryOutput:3 |
| Property ID | Reports the BACnet property name proxied by the point. For example Present Value or Event State , depending on the selection. |

| Column | Description |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | NOTE: Do not change the Property ID on an existing BACnet proxy point. Changing it may produce undesirable results, especially if a different control point type would apply. Instead, make a new proxy point. |
| Index | Provides a numeric index into the property array. Applies only to an arrayed property, such as Priority Array, otherwise it defaults -1 for no index. |
| Tuning Policy Name | Displays the name of the network's tuning policy component assigned to the point. |
| Data Type | Displays the ASN (Abstract Syntax Notation) data type for the property (ENUMERATED, REAL, and so on). |
| Read | Identifies the read-only read status of the proxy extension, which is typically either Polled, COV, or unsubscribed, depending on a number of factors. If a read error occurs, other descriptive text may appear instead. |
| Write | Identifies the read-only write status of the proxy extension, which is typically read only if the proxy point is not writable or, if it is a writable point type, either writable or ok (the last write occurred within the effective tuning policy period). Or, if a write operation failed, the write status provides some descriptive text. For example, If you create a writable point for a read-only object (say, a NumericWritable for presentValue of an Analog_Input object), and attempt to write to it, the write status may show: Property: Write Access Denied. If the error is actually a BACnet Error, then this colon-separated format will contain the Error Class and Error Code returned by the device. |
| Device Facets | Reports the learned facets from the source BACnet object. |
| Facets | Reports the facets in use by the parent proxy point. |
| Conversion | Identifies the conversion used between device facets and point facets (typically Default). |
| Read Value | Reports the last value read from device, and expressed in device facets. |
| Write Value | Reports the last value written using device facets. (Applies to writable types only.) |
| Fault Cause | Displays the reason why the object is unable to connect the device. |

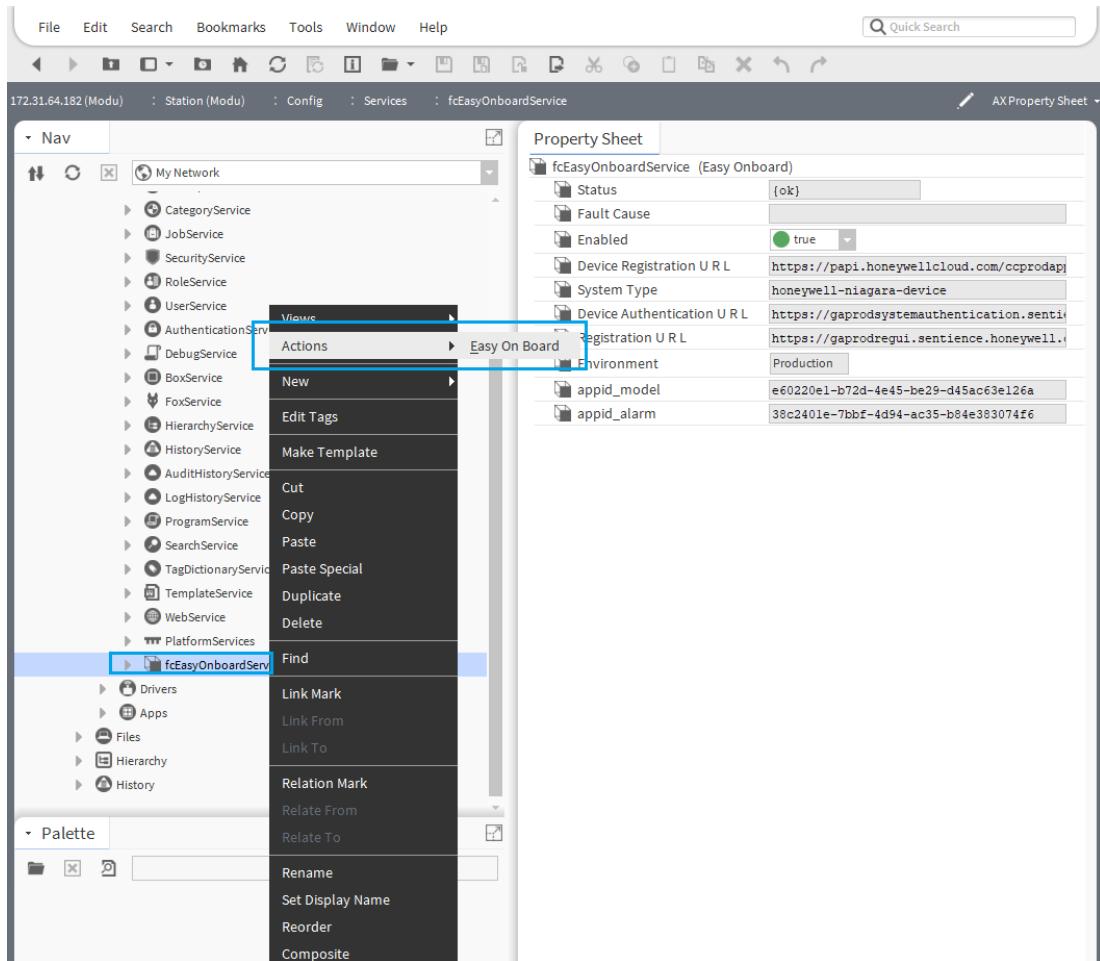
Buttons

- **New Folder** creates a new folder for devices. Each such folder provides its own set of manager views.
- **New** creates a new device record in the database.
- **Edit** opens the device's database record for updating.
- **Discover** runs a discover job to locate installed devices, which appear in the **Discovered** pane. This view has a standard appearance that is similar to all **Device Manager** views.
- **Cancel** ends the current discovery job.
- **Add** inserts into the database a record for the discovered and selected object.
- **Match** associates a discovered device with a record that is already in the database.
- **TagIt** associates metadata, such as location or unique configuration with the object.

Bacnet Point Manager actions

Two actions are available in this view. These actions are on the proxy extension itself, and not the parent control point.

Figure 159 Actions for any Bacnet ProxyExt



These actions are briefly described as follows:

- Force Read results in an immediate poll of the source BACnet object's property.
- Force Write attempts to write from the Framework to the BACnet object's property. If the property is read-only (or otherwise protected by the BACnet device), the system displays an error in the Write Status property.

Manager menu

These actions are available on the **BacnetDeviceFolder** Manager menu.

| Menu item | Description |
|-----------------|--------------------------------------------------------------------------|
| New Folder | Creates a new folder (same function as the New Folder button). |
| All Descendents | Expands the selected folder to reveal all subordinate objects. |
| New | Creates a new device record in the database. |
| Edit | Opens the Edit window for updating the selected record. |
| Learn Mode | Controls the discovery of devices and adding them to the database. |
| Discover | Initiates a discovery job (same as clicking the Discover button). |
| Cancel | Stops the current job. |

| Menu item | Description |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Add | Opens an Add window with properties to configure before adding the selected record(s) in the Discovery pane to the database. |
| Quick Add | Adds the selected record(s) in the Discovery pane to the database. |
| Match | Opens a window with properties to configure before matching the discovered record with a record in the database. |
| Quick Match | Matches the selected record with its sister record that is already in the database. |
| Tag Mode | Opens a Tag Dictionary pane for managing metadata tags. The <i>Niagara Tagging Guide</i> documents tag. |
| TagIt | Associates an item of metadata, such as location or unique configuration, with the object. |

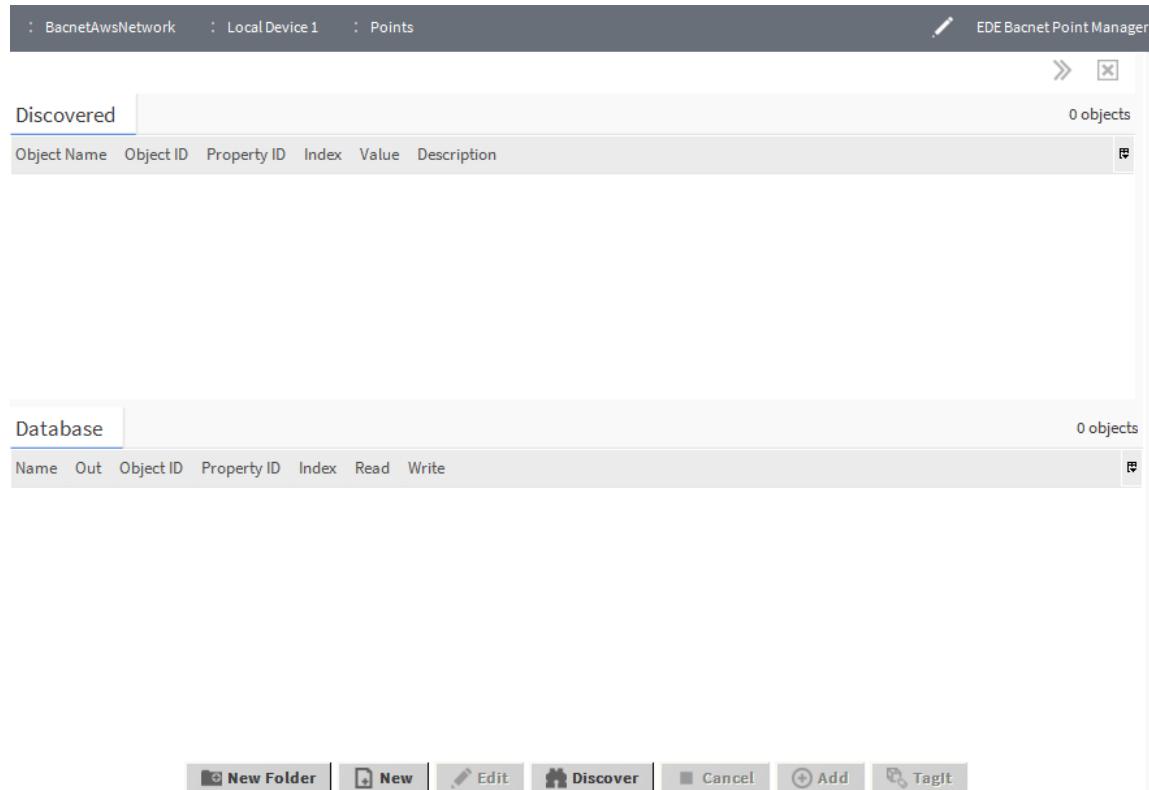
EDE Bacnet Point Manager

This is the default view for the **Points** extension (or **Points** folder) under any **BacnetDevice**, and works similarly to the way other point managers work, which support online point discovery. Refer to the *Niagara Drivers Guide* for general details.

Discovered Pane

By default, the **Discovered** table for BACnet objects lists the single BACnet device object at top, and other objects underneath, with each object initially occupying a single row.

Figure 160 EDE Point Manager View



To open this view, expand **Config**→**Drivers**→**BacnetNetwork**→**Bacnet Device** and double-click on the **Points Folder**.

Discovered pane

| Column | Description |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Object Name | Displays the unique name of the discovered BACnet object. |
| Object ID | Identifies the discovered BACnet object. This number is a combination of BACnet object type and instance number (unique within that type). In this column, these two properties appear separated by a colon (:), using descriptors for type. |
| Property Id | Reports the Property ID for the discovered BACnet object. By default, all objects that have a Present_Value property are listed with this value on top. BACnet objects without a Present_Value property (for example, a device object or trend log object) list showing another property as the top Property ID . Typically, present value is the most useful piece of data from any BACnet object. However, you can expand any discovered object to see all its properties as children in the Discovered pane, where each one is a separate proxy point candidate. Additionally, properties that are arrays are further expandable. |
| Index | Reports a number into an arrayed property, if selected, otherwise this column cell is blank. |
| Value | Reports the static value of the associated property, captured when the system retrieved the object list (or, for any non-default properties, when you expanded the top property). Discovered values display showing any descriptors associated with the BACnet object's related properties (Units, Active_Text, Inactive_Text, and so on). |
| Description | Reports any available character string Description for the BACnet object. In many cases, this value may be blank. |

Database pane

The **Database** table contains existing BACnet proxy points, where each appears as a row in the table. Each proxy point represents one data item from a specific BACnet object in that device.

| Column | Description |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Path | Reports the station path to the proxy point. |
| Name | Reports the name for the BACnet proxy point. If you added the point from a discover (selecting the default presentValue property), and did not edit Object Name , this name is identical to the BACnet object's name. As needed, you can edit this value. |
| Type | Reports the type of point that the proxy is based upon (BooleanWritable, NumericPoint, etc.). |
| Out | Reports the current out value, including any point facets. This defaults to the single (configured) property value along with status for the proxy point. Since BACnet supports its own native status in addition to the Framework's status, you can edit point facets to poll for additional properties, such as the native status Flags and/or priority Array levels. |
| Enabled | Reports if the proxy point is currently enabled for communication. |
| Object ID | Reports the Object Id property of the proxied BACnet object, which is a combination of BACnet object type and instance number (unique within that type). In this column, these two properties appear separated by a colon (:), using descriptors for type. |
| Property ID | Reports the BACnet property name proxied by the point. For example Present Value or Event State , depending on the selection. |
| Index | Applies only if an arrayed property, like Priority Array (otherwise it is -1 for no index). Provides a numeric index into the property array. |
| Tuning Policy Name | Identifies the name of the network's tuning policy component assigned to the point. |
| Data Type | Identifies the ASN data type for the property (ENUMERATED, REAL, and so on). |
| Read | Identifies the read-only read status of the proxy extension, which is typically either Polled, COV, or unsubscribed, depending on a number of factors. If a read error occurs, other descriptive text may appear instead. |
| Write | Identifies the read-only write status of the proxy extension, which is typically read only if the proxy point is not writable or, if it is a writable point type, either writable or ok (the last write occurred within the effective tuning policy period). Or, if a write operation fails, the write status provides some descriptive text. |

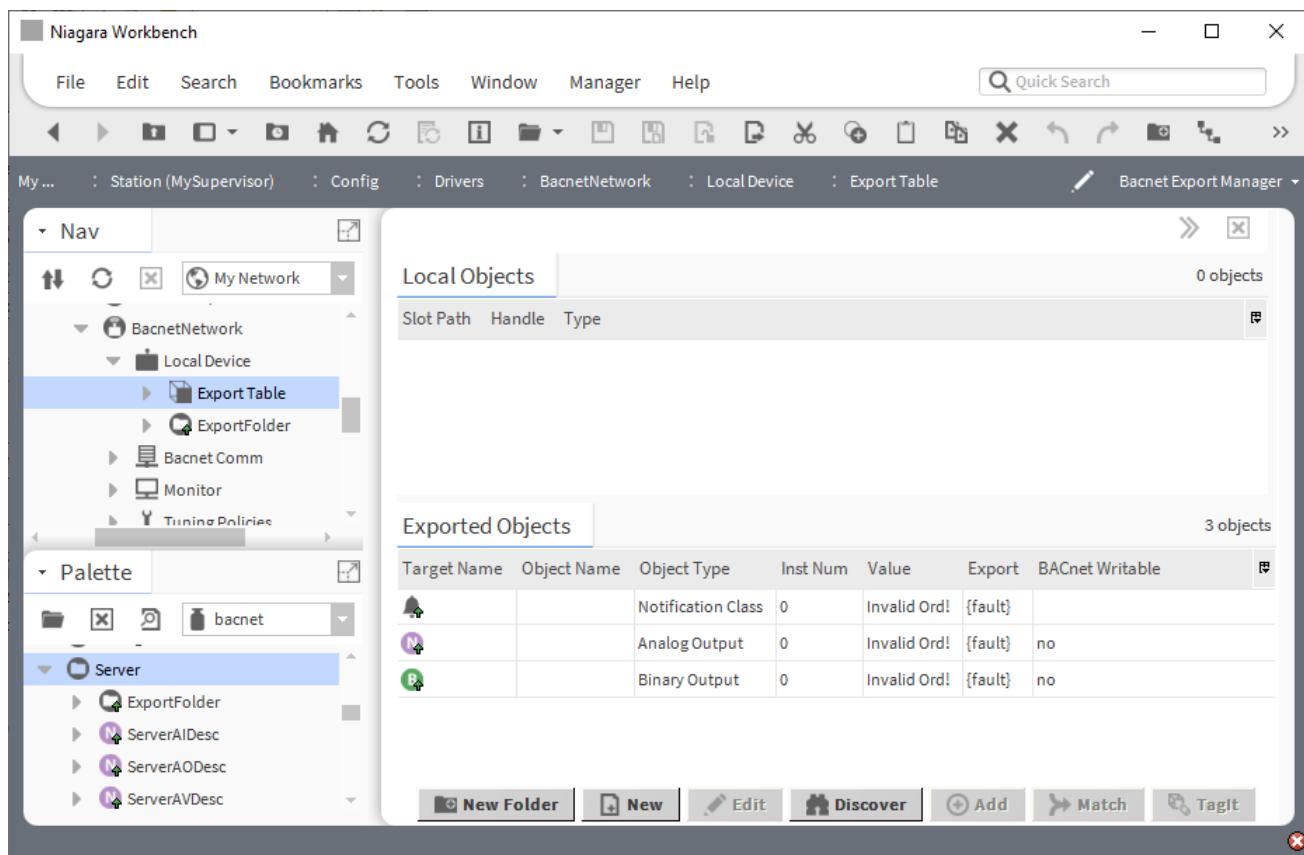
| Column | Description |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | For example, If you create a writable point for a read-only object (say, a NumericWritable for presentValue of an Analog_Input object), and attempt to write to it, the write status may show: <code>Property:Write Access Denied</code> . If the error is actually a BACnet Error, then this colon-separated format will contain the Error Class and Error Code returned by the device. |
| Device Facets | Reports the learned facets from the source BACnet object. |
| Facets | Reports the facets in use by the parent proxy point. |
| Conversion | Identifies the conversion used between device facets and point facets (typically <code>Default</code>). |
| Read Value | Reports the last value read from device, and expressed in device facets. |
| Write Value | Reports the last value written using device facets. (Applies to writable types only.) |
| Fault cause | Displays the reason why the object is unable to connect the device. |

Buttons

- **New Folder** creates a new folder for devices. Each such folder provides its own set of manager views.
- **New** creates a new device record in the database.
- **Edit** opens the device's database record for updating.
- **Discover** runs a discover job to locate installed devices, which appear in the **Discovered** pane. This view has a standard appearance that is similar to all **Device Manager** views.
- **Cancel** ends the current discovery job.
- **Add** inserts into the database a record for the discovered and selected object.
- **Match** associates a discovered device with a record that is already in the database.
- **TagIt** associates metadata, such as location or unique configuration with the object.

Bacnet Export Manager (bacnet-BacnetExportManager)

This manager lists BACnet export descriptors and any BACnet export folders. It is the default view for the **Export Table**, as well as any child **Bacnet Export Folder**.

Figure 161 Bacnet Export Manager

To open this manager, expand **Config→Drivers→BacnetNetwork→Local Device** and double-click **Export Table**.

Local Objects pane

| Column name | Description |
|-------------|----------------------------------|
| Slot Path | Identifies the export slot path. |
| Handle | Displays a number. |
| Type | Displays the type of descriptor. |

Exported Objects pane

| Column name | Description |
|-------------|------------------------------------------------------------|
| Name | Provides additional information |
| Target Name | Identifies the target descriptor name. |
| Object Name | Displays the unique name for the exported data descriptor. |
| Type | Displays the type of descriptor. |
| Object Type | Displays the type of exported data descriptor. |
| Inst Number | Displays an instance number for the data descriptor. |
| Value | Displays the current value of the data descriptor. |

| Column name | Description |
|-----------------|------------------------------------------------------------|
| Export Ord | |
| Export | Displays the descriptor's export status. |
| Bacnet Writable | Displays whether the descriptor is BACnet writable or not. |
| Fault Cause | Displays the reason why the export failed. |
| Description | Displays additional information for the descriptor. |
| BACnet Writable | |

Buttons

- **New Folder** creates a new folder for devices. Each such folder provides its own set of manager views.
- **New** creates a new device record in the database.
- **Edit** opens the device's database record for updating.
- **Discover** runs a discover job to locate installed devices, which appear in the **Discovered** pane. This view has a standard appearance that is similar to all **Device Manager** views.
- **Add** inserts into the database a record for the discovered and selected object.
- **Match** associates a discovered device with a record that is already in the database.
- **TagIt** associates metadata, such as location or unique configuration with the object.

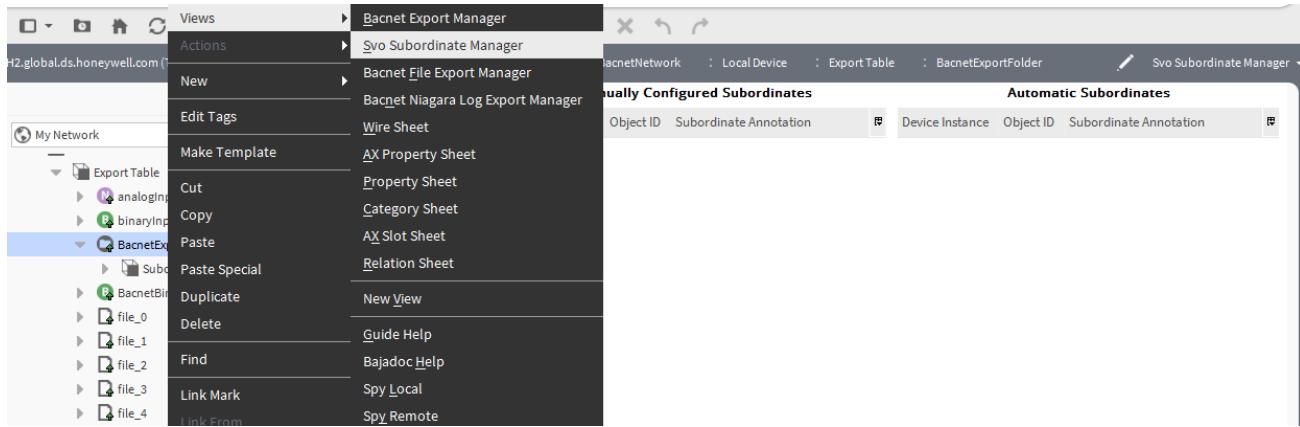
Manager menu

These actions are available on the **Export Table Manager** menu.

| Menu item | Description |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| New Folder | Creates a new folder (same function as the New Folder button). |
| All Descendents | Expands the selected folder to reveal all subordinate objects. |
| New | Creates a new device record in the database. |
| Edit | Opens the Edit window for updating the selected record. |
| Learn Mode | Controls the discovery of devices and adding them to the database. |
| Discover | Initiates a discovery job (same as clicking the Discover button). |
| Add | Opens an Add window with properties to configure before adding the selected record(s) in the Discovery pane to the database. |
| Quick Add | Adds the selected record(s) in the Discovery pane to the database. |
| Match | Opens a window with properties to configure before matching the discovered record with a record in the database. |
| Quick Match | Matches the selected record with its sister record that is already in the database. |
| Tag Mode | Opens a Tag Dictionary pane for managing metadata tags. The <i>Niagara Tagging Guide</i> documents tag. |
| TagIt | Associates an item of metadata, such as location or unique configuration, with the object. |

Svo Subordinate Manager view

This manager lists related BACnet export descriptors and references to subordinate structured view objects. It functions in a remote controller station, not in a Supervisor station.

Figure 162 Svo Subordinate Manager view on the BacnetExportFolder

To access this view, log in to the remote controller station, expand **Config→Drivers→BacnetNetwork** right-click **ExportFolder** and click **Views→Svo Subordinate Manager**

In the exposed structured view object (from a BACnet export folder), subordinates appear in two property lists:

- **Subordinate_List:** List of object identifiers for all BACnet objects the SVO organizes.
- **Subordinate_Annotations:** List of descriptions of the same objects—the same size as the Subordinate_List, where entries in one correspond to entries in the other.

Most entries are automatic subordinates, shown on the right side. These result from either child BACnet server descriptors or export folders.

The buttons below the left side of the view allow you to **Edit** or **Delete** manually-added entries for subordinate objects.

NOTE: The same BACnet object can be a subordinate in more than one Structured View Object. For example, an Analog Input object that represents a temperature can be a subordinate in both a Temperatures SVO tree as well as a Admin Building SVO tree.

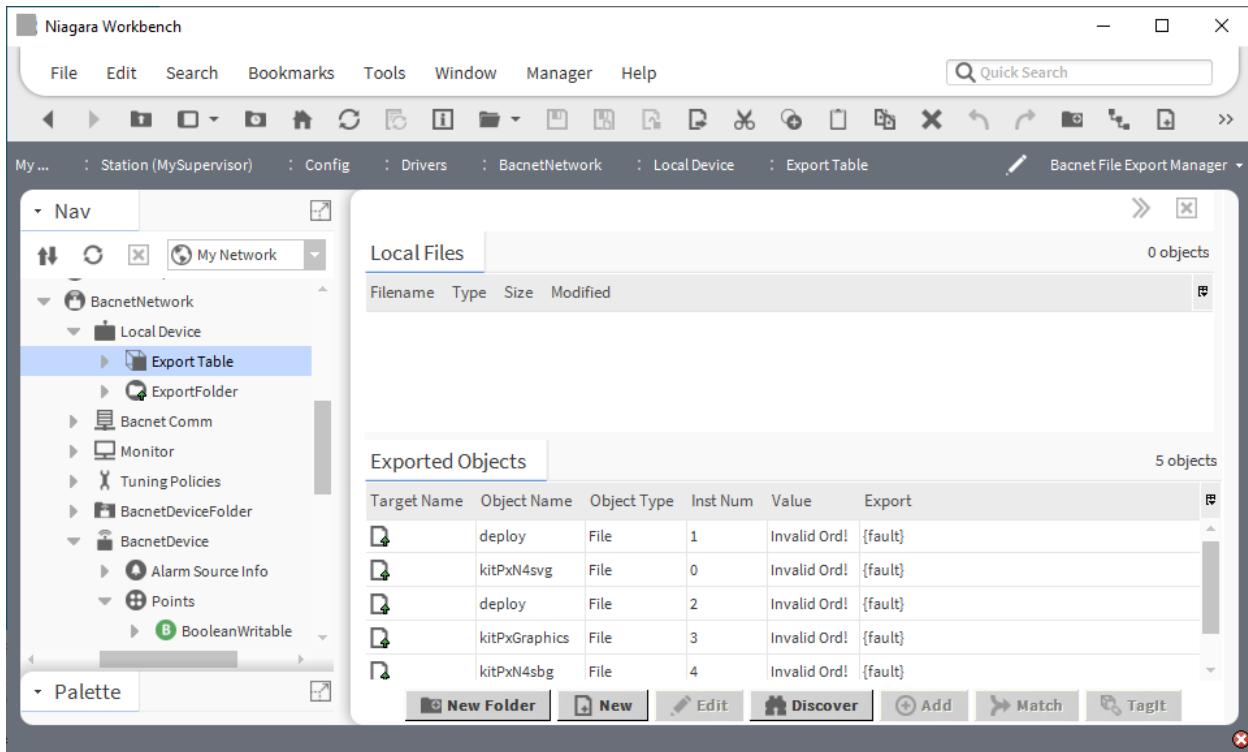
Buttons

- **New** creates a new entry.
- **Edit** opens an existing entry for editing.
- **Delete** removes the selected entry.
- **Exit Auto** disables automatic subordinates.

Bacnet File Export Manager

This view exports files under the station's folder as BACnet File objects. It shows only BACnet file descriptors and any BACnet export folders.

You can access these files as read-only or as writables, depending on the local file system. Currently, the BACnet File_Access_Method is Stream Access only.

Figure 163 Bacnet File Export Manager

To access this view, expand **Config→Drivers→BacnetNetwork→Local Device**, right-click **Export Table** and click **Views→Bacnet File Export Manager**.

Local Files pane

As in other manager views, when you click **Discover** this manager goes to Learn mode, opens to two panes, and executes a discover job. Discover returns multiple objects, each of which occupies one row in the table.

| Column name | Description |
|-------------|--------------------------------------------|
| Filename | Displays the name of the export file. |
| Type | Displays the file type. |
| Size | Displays the file size. |
| Modified | Displays the details of file modification. |

Exported Objects pane

Following are the columns and buttons in database pane:

| Column name | Description |
|-------------|-----------------------------------------------------------|
| Name | Displays more information. |
| Target Name | Displays the file name. |
| Object Name | Displays the file's unique name. |
| Object Type | Displays the object type, which, in this case, is a file. |
| Inst Name | Displays file Id number. |

| Column name | Description |
|-------------|-------------------------------------------------|
| Value | Displays the file path. |
| Export | Displays the status of export. |
| Fault Cause | Displays the reason why the export failed. |
| Description | Displays additional information about the file. |

Buttons

- **New Folder** creates a new folder for devices. Each such folder provides its own set of manager views.
- **New** creates a new device record in the database.
- **Edit** opens the device's database record for updating.
- **Discover** runs a discover job to locate installed devices, which appear in the **Discovered** pane. This view has a standard appearance that is similar to all **Device Manager** views.
- **Add** inserts into the database a record for the discovered and selected object.
- **Match** associates a discovered device with a record that is already in the database.
- **TagIt** associates metadata, such as location or unique configuration with the object.

Manager menu

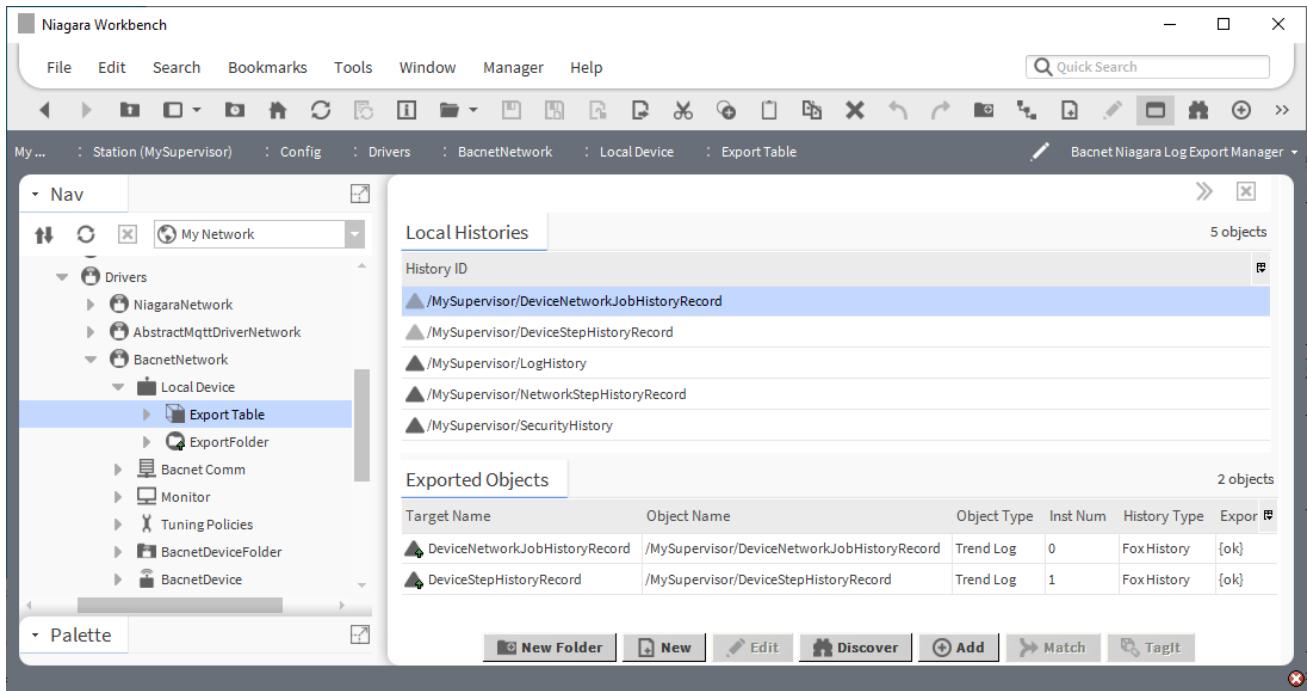
These actions are available on the **BacnetDeviceFolder** Manager menu.

| Menu item | Description |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| New Folder | Creates a new folder (same function as the New Folder button). |
| All Descendents | Expands the selected folder to reveal all subordinate objects. |
| New | Creates a new device record in the database. |
| Edit | Opens the Edit window for updating the selected record. |
| Learn Mode | Controls the discovery of devices and adding them to the database. |
| Discover | Initiates a discovery job (same as clicking the Discover button). |
| Add | Opens an Add window with properties to configure before adding the selected record(s) in the Discovery pane to the database. |
| Quick Add | Adds the selected record(s) in the Discovery pane to the database. |
| Match | Opens a window with properties to configure before matching the discovered record with a record in the database. |
| Quick Match | Matches the selected record with its sister record that is already in the database. |
| Tag Mode | Opens a Tag Dictionary pane for managing metadata tags. The <i>Niagara Tagging Guide</i> documents tag. |
| TagIt | Associates an item of metadata, such as location or unique configuration, with the object. |

Bacnet Niagara Log Export Manager

This view exports standard station histories as BACnet trend log objects, as well as manages BacnetTrendLogDescriptors.

Figure 164 Bacnet Niagara Log Export Manager



To open this view, expand **Config**→**Drivers** followed by expanding the BACnet network node in the Nav tree, right-clicking the **Export Table**, and clicking **Views**→**Bacnet Niagara Log Export Manager**.

NOTE: If fully BACnet-compliant trend log objects are required, you must configure the source points with one of the **BacnetTrendLogExt** components from the **bacnet** palette.

Local Histories pane

As in other manager views, when you click **Discover** this manager enters Learn mode, splits into two panes, and executes a discovery job. The job returns local history objects, each of which occupies one row in the exported objects table.

| Column name | Description |
|-------------|------------------------------------------|
| History ID | Displays the object ID for this history. |

Exported Objects pane

| Column name | Description |
|--------------|----------------------------------------------------------|
| Name | Provides additional information |
| Target Name | Displays the name of the history. |
| Object Name | Displays the path and unique name of the object. |
| Type | Displays the type of the history (Audit or Log History). |
| Object Type | Displays the object type. |
| Inst Num | Displays the object instance number. |
| History Type | Displays the history type. |
| Export Ord | Displays the ord of the exported history. |
| Export | Displays the status of the export. |

| Column name | Description |
|-------------|--------------------------------------------------|
| Fault cause | Displays the problem in exporting history. |
| Description | Displays additional information for the history. |

Buttons

- **New Folder** creates a new folder for devices. Each such folder provides its own set of manager views.
- **New** creates a new device record in the database.
- **Edit** opens the device's database record for updating.
- **Discover** runs a discover job to locate installed devices, which appear in the **Discovered** pane. This view has a standard appearance that is similar to all **Device Manager** views.
- **Add** inserts into the database a record for the discovered and selected object.
- **Match** associates a discovered device with a record that is already in the database.

Manager menu

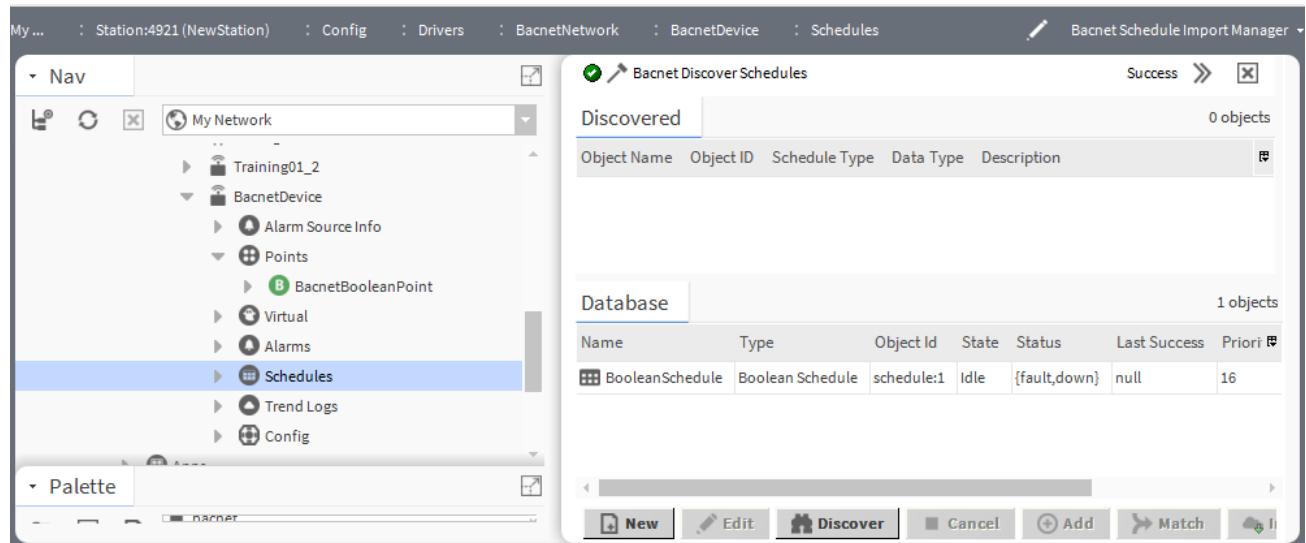
These actions are available on the **BacnetDeviceFolder** Manager menu.

| Menu item | Description |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| New Folder | Creates a new folder (same function as the New Folder button). |
| All Descendents | Expands the selected folder to reveal all subordinate objects. |
| New | Creates a new device record in the database. |
| Edit | Opens the Edit window for updating the selected record. |
| Learn Mode | Controls the discovery of devices and adding them to the database. |
| Discover | Initiates a discovery job (same as clicking the Discover button). |
| Add | Opens an Add window with properties to configure before adding the selected record(s) in the Discovery pane to the database. |
| Quick Add | Adds the selected record(s) in the Discovery pane to the database. |
| Match | Opens a window with properties to configure before matching the discovered record with a record in the database. |
| Quick Match | Matches the selected record with its sister record that is already in the database. |
| Change History Device Name | Opens a window to change the name. |
| Tag Mode | Opens a Tag Dictionary pane for managing metadata tags. The <i>Niagara Tagging Guide</i> documents tag. |
| TagIt | Associates an item of metadata, such as location or unique configuration, with the object. |

Bacnet Schedule Import Manager

This component is the container for BACnet schedules under a BacnetDevice. Its default view is the **Bacnet Schedule Import Manager**.

Figure 165 Bacnet Schedule Import Manager



To access this view, expand **Config→Drivers→BacnetNetwork→BacnetDevice** and double-click **Schedules**.

Discovered pane

As in other manager views, when you click **Discover** this manager goes to Learn mode, splits into two panes, and executes a discover job. Discover also returns Schedule objects, each of which occupies one row in this view. Following are the columns displayed in Discovered pane:

| Column name | Description |
|---------------|-------------------------------------------------------------|
| Object name | Displays the object's unique name. |
| Object ID | Displays the identifier assigned to the object. |
| Schedule Type | Displays the schedule type. |
| Data Type | Displays the type of data. |
| Description | Displays the additional information of the schedule object. |

Database pane

Following are the columns and buttons in the database pane:

| Column name | Description |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name | Indicates the name of the BACnet schedule. |
| Type | Indicates the type of schedule. |
| Object ID | Indicates the ID of schedule. |
| State | Reports the particular schedule state. |
| Status | Indicates the condition at the last check. {ok} indicates that the component is licensed and polling successfully. {down} indicates that the last check was unsuccessful, perhaps because of an incorrect property, or possibly loss of network connection. {disabled} indicates that the Enable property is set to false. {fault} indicates another problem. Refer to Fault Cause for more information. |

| Column name | Description |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Enabled | Reports if the schedule is enabled for communication. |
| Last Attempt | Displays when the driver attempted the last import. |
| Last Success | Displays when the import was successful. |
| Last Failure | Displays where the last import failed. |
| Fault Cause | Displays the reason why the last import failed. |
| Priority For Writing | Reports schedule priority, which defaults to 16. |
| Execution Time | Reports the frequency with which data are imported from a remote trend log (or trend log multiple) into a history (or histories). Data import occurs at intervals, daily or when requested. |

Buttons

- **New** creates a new device record in the database.
- **Edit** opens the device's database record for updating.
- **Discover** runs a discover job to locate installed devices, which appear in the **Discovered** pane. This view has a standard appearance that is similar to all **Device Manager** views.
- **Cancel** ends the current discovery job.
- **Add** inserts into the database a record for the discovered and selected object.
- **Match** associates a discovered device with a record that is already in the database.
- **Import** imports the selected schedule objects.
- **TagIt** associates metadata, such as location or unique configuration with the object.

Manager menu

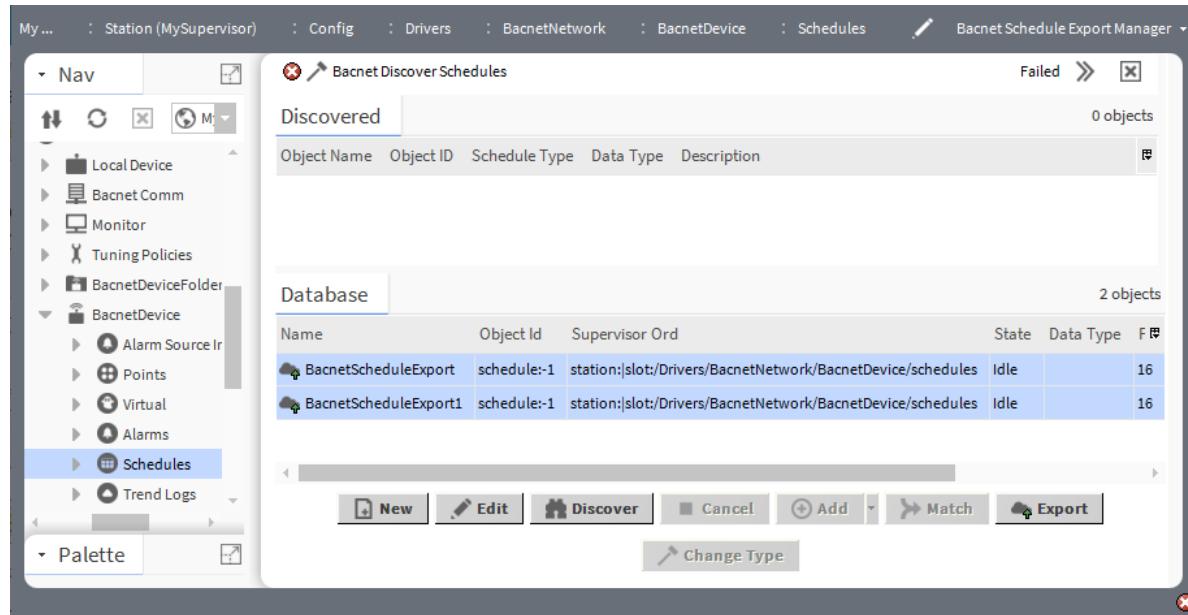
These actions are available on the **BacnetDeviceFolder** Manager menu.

| Menu item | Description |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| New | Creates a new device record in the database. |
| Edit | Opens the Edit window for updating the selected record. |
| Learn Mode | Controls the discovery of devices and adding them to the database. |
| Discover | Initiates a discovery job (same as clicking the Discover button). |
| Cancel | Stops the current job. |
| Add | Opens an Add window with properties to configure before adding the selected record(s) in the Discovery pane to the database. |
| Quick Add | Adds the selected record(s) in the Discovery pane to the database. |
| Match | Opens a window with properties to configure before matching the discovered record with a record in the database. |
| Quick Match | Matches the selected record with its sister record that is already in the database. |
| Import | Runs the import job. |
| Tag Mode | Opens a Tag Dictionary pane for managing metadata tags. The <i>Niagara Tagging Guide</i> documents tag. |
| TagIt | Associates an item of metadata, such as location or unique configuration, with the object. |

Bacnet Schedule Export Manager

This component corresponds to schedules or calendars that are pushed to specific BACnet schedule or calendar objects in the remote device. Each exported schedule points to a component in the station. The Framework writes events in an exported schedule (or calendar) as a client-side operation to the identified BACnet schedule object in that BACnet device.

Figure 166 Bacnet Schedule Export Manager



To access this view, expand **Config**→**Drivers**→**BacnetNetwork**→**BacnetDevice**, right-click **Schedules** click **Views**→**Bacnet Schedule Export Manager**.

Discovered Pane

As in other manager views, when you click **Discover** this manager goes to Learn mode, splits into two panes, and executes a discover job. Discover also returns multiple schedule objects, each of which occupies one row in this view.

| Column | Description |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Object name | Displays the object's unique name. |
| Object ID | Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number . For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device. |
| Schedule type | Displays the schedule type. |
| Data type | Reports the ASN (Abstract Syntax Notation) data type for a device property. The driver automatically converts values to this appropriate type. For example, an analog input present value is an ASN REAL that can be interpreted by a string point as a character string. ENUMERATED is another example. |
| Description | Displays the additional information about object. |

Database pane

| Column | Description |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name | Indicates the name of the BACnet schedule. |
| Type | Indicates the type of the schedule. |
| Object Id | <p>Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number.</p> <p>For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet inter-network. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device.</p> |
| Supervisor Ord | Identifies the local station ORD for the schedule. |
| State | Indicates the condition of the BACnet schedule. |
| Data Type | Reports the ASN (Abstract Syntax Notation) data type for a device property. The driver automatically converts values to this appropriate type. For example, an analog input present value is an ASN REAL that can be interpreted by a string point as a character string. ENUMERATED is another example. |
| Enabled | Reports if the schedule is enabled for communication. |
| Priority For Writing | Reports the priority for saving a schedule. |
| Execution Time | Reports the frequency with which data are imported from a remote trend log (or trend log multiple) into a history (or histories). Data import occurs at intervals, daily or when requested. |
| Last Attempt | Displays when the driver attempted the last export. |
| Last Success | Displays when the export was successful. |
| Last Failure | Displays when the last export failed. |
| Fault Cause | Displays the reason why the last export failed. |
| Skip Writes | Adjusts which property types in a target BACnet schedule object the system writes to upon an export from a weekly schedule component. |
| Write Enum As | Defines how to write Enum data as Unsigned: ENUMERATED or as an INTEGER. |
| Out Of Service | <p>Affects only the BACnet exposure and access to the source component. If the source (exported) component is a proxy point, the equivalent Enabled property in its ProxyExt is not affected.</p> <p><code>true</code> writes the point's current value to the Present Value property. Often, a remote BACnet client sets (or clears) this Out of Service state.</p> <p>You can set this property to <code>false</code> if needed.</p> |

Buttons

- **New** creates a new device record in the database.
- **Edit** opens the device's database record for updating.
- **Discover** runs a discover job to locate installed devices, which appear in the **Discovered** pane. This view has a standard appearance that is similar to all **Device Manager** views.
- **Cancel** ends the current discovery job.
- **Add** inserts into the database a record for the discovered and selected object.
- **Match** associates a discovered device with a record that is already in the database.
- **Export** exports the selected schedule object(s).
- **Change Type** changes the schedule type.

Manager menu

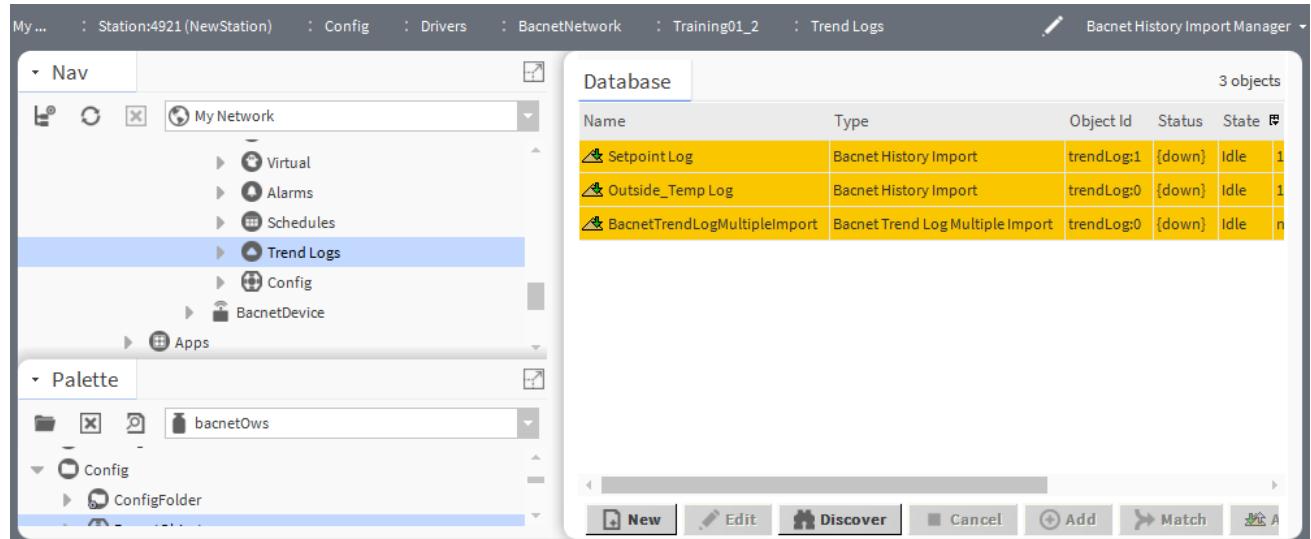
These actions are available on the **BacnetDeviceFolder Manager** menu.

| Menu item | Description |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| New | Creates a new device record in the database. |
| Edit | Opens the Edit window for updating the selected record. |
| Learn Mode | Controls the discovery of devices and adding them to the database. |
| Discover | Initiates a discovery job (same as clicking the Discover button). |
| Cancel | Stops the current job. |
| Add | Opens an Add window with properties to configure before adding the selected record(s) in the Discovery pane to the database. |
| Quick Add | Adds the selected record(s) in the Discovery pane to the database. |
| Match | Opens a window with properties to configure before matching the discovered record with a record in the database. |
| Quick Match | Matches the selected record with its sister record that is already in the database. |
| Export | Runs the export job. |
| Change Type | Opens a window from which to change the schedule type. |
| Tag Mode | Opens a Tag Dictionary pane for managing metadata tags. The <i>Niagara Tagging Guide</i> documents tag. |
| TagIt | Associates an item of metadata, such as location or unique configuration, with the object. |

Bacnet History Import Manager

This manager displays active import descriptors. It is the default view of the trend logs.

Figure 167 Bacnet History Import Manager



To access this view, double-click **Trend Logs** folder under any **BacnetDevice** component.

Discovered pane

As in other manager views, when you click **Discover** this manager goes to Learn mode, splits into two panes, and executes a discover job. Discover also returns multiple trend log objects, each of which occupies one row in this view. Following are the columns in the Discovered pane:

| Column name | Description |
|--------------|------------------------------------------------------|
| Object Name | Displays the unique name of the history trend log. |
| Object ID | Displays an identifier for the history trend log. |
| History Type | Displays the type of trend log. |
| Description | Displays additional information about the trend log. |

Database pane

| Column name | Description |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name | Displays the history trend log name. |
| Type | Displays the type of trend log. |
| Object ID | Displays the trend log identifier. |
| Execution Time | Reports the frequency with which data are imported from a remote trend log (or trend log multiple) into a history (or histories). Data import occurs at intervals, daily or when requested. |
| Enabled | Reports if the history is currently enabled for communication. |
| Status | Displays the status of the trend log. |
| State | Displays the current state of the trend log. |
| Last Attempt | Displays when the driver attempted the last import. |
| Last Success | Displays when the import was successful. |
| Last Failure | Displays when the last import failed. |
| Fault Cause | Displays the reason why the last import failed. |
| Capacity | Displays the available capacity for trend logs. |
| Full Policy | Indicates what happens when capacity storage is reached. |
| On Demand Poll Enabled | Indicates if polling on demand is enabled. |
| On Demand Poll Frequency | Displays the frequency for on-demand polling. |
| Time zone | Displays the current time zone. |
| Local History Name Format | Reports the format used for local histories. |
| Local History Name | Reports the name of the local history. |
| Discovery History Type | Displays the type of history. |
| Always Request By Reference Time | Reports if reference time is required for a history request. |

Buttons

- **New** creates a new device record in the database.
- **Edit** opens the device's database record for updating.

- **Discover** runs a discover job to locate installed devices, which appear in the **Discovered** pane. This view has a standard appearance that is similar to all **Device Manager** views.
- **Cancel** ends the current discovery job.
- **Add** inserts into the database a record for the discovered and selected object.
- **Match** associates a discovered device with a record that is already in the database.
- **Archive** archives the selected trend logs.
- **TagIt** associates metadata, such as location or unique configuration with the object.

Right-click actions

The following right-click actions are available:

- **Execute** imports the trend logs.
- **Clear Records in Device** deletes trend log histories from the

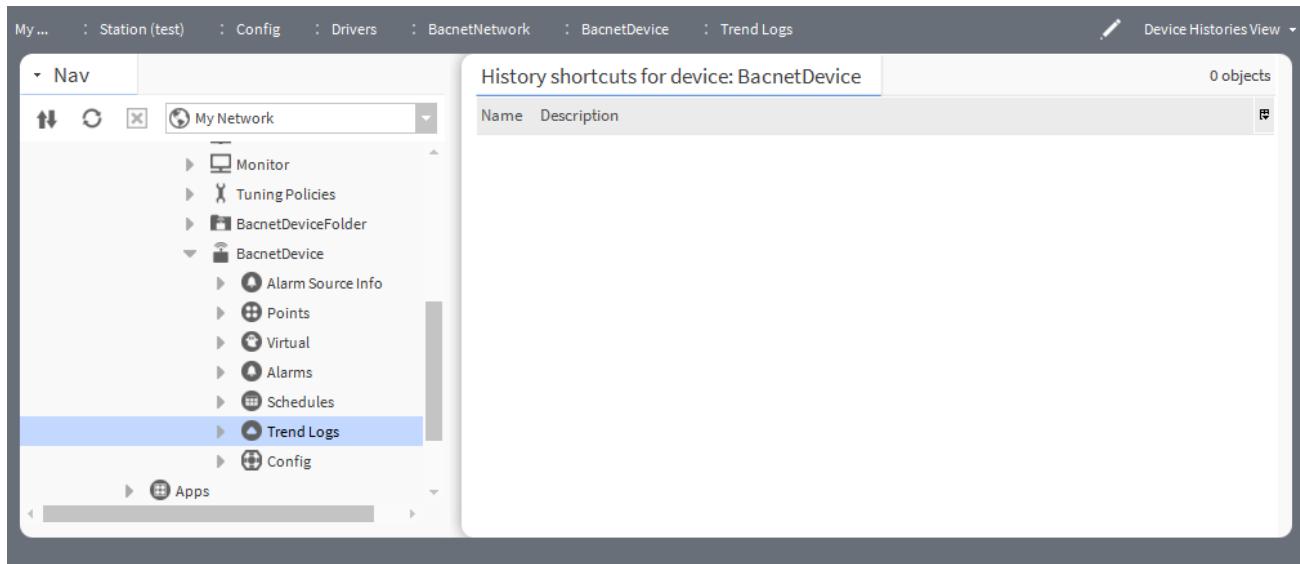
Manager menu

These actions are available on the **BacnetDeviceFolder** Manager menu.

| Menu item | Description |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| New | Creates a new device record in the database. |
| Edit | Opens the Edit window for updating the selected record. |
| Learn Mode | Controls the discovery of devices and adding them to the database. |
| Discover | Initiates a discovery job (same as clicking the Discover button). |
| Cancel | Stops the current job. |
| Add | Opens an Add window with properties to configure before adding the selected record(s) in the Discovery pane to the database. |
| Quick Add | Adds the selected record(s) in the Discovery pane to the database. |
| Match | Opens a window with properties to configure before matching the discovered record with a record in the database. |
| Quick Match | Matches the selected record with its sister record that is already in the database. |
| Archive | Opens a window to create a backup. |
| Tag Mode | Opens a Tag Dictionary pane for managing metadata tags. The <i>Niagara Tagging Guide</i> documents tag. |
| TagIt | Associates an item of metadata, such as location or unique configuration, with the object. |

Device Histories view

This view lists history shortcuts for Supervisor devices.

Figure 168 Device History view

To access this view, right-click the Trend Logs node in the Nav tree and click **Views→Device Histories View**

| Column name | Description |
|-------------|------------------------------------------------------|
| Name | Displays the name of the history type. |
| Description | Displays additional information of the history type. |

Bacnet Trend Multiple View

This view is the default view for the **BacnetTrendLogMultipleImport** component.

To access this view, double-click the import descriptor in the Nav tree, or right-click the **BacnetTrendLogMultipleImport** component and click **Views→Bacnet Trend Multiple View**.

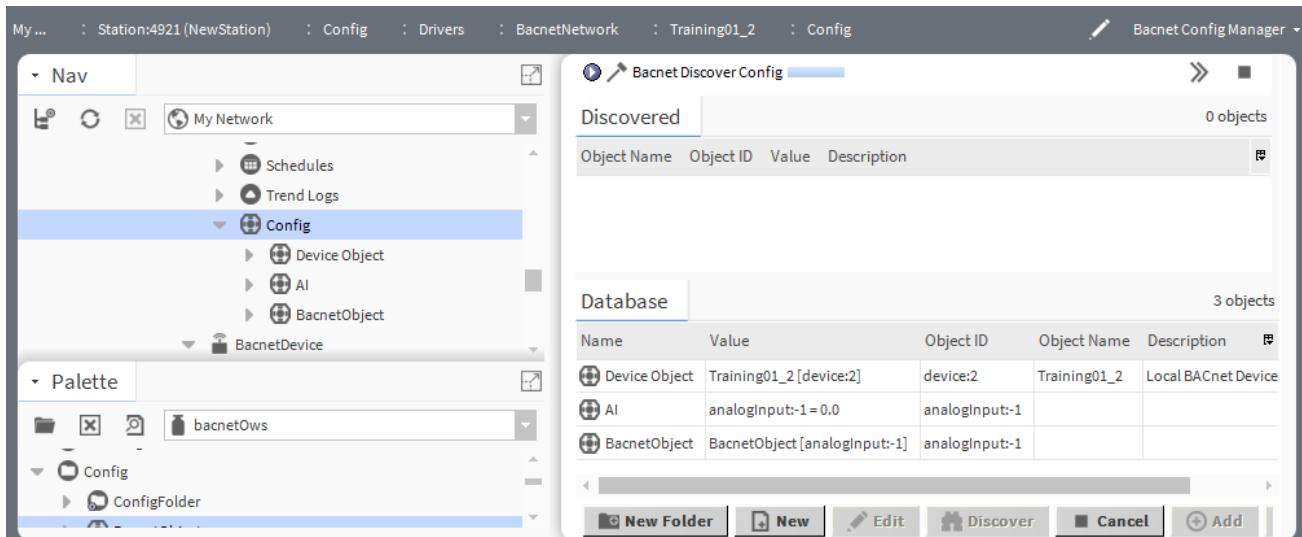
Bacnet Config Manager

This table view manages configuration records for a wide group of devices. It is the default view of the **BacnetConfigDeviceExt**, which is available from the **bacnet** module and palette.

From this view, you can learn other config-type objects, each representing a BACnet object in that device. You might do this to evaluate a BACnet device, rather than for permanent station engineering. In this case, remember to delete the config-type objects afterwards, as they consume station resources.

NOTE: Another way to evaluate the objects in a device is via its Virtual Gateway. This may actually be easier, as the resulting virtual components are automatically removed after browsing.

Figure 169 Bacnet Config Manager



To open this view, expand **Config**→**Drivers**→**BacnetNetwork**→**BacnetDevice** and double-click the **Config** node in the Nav tree.

Discovered pane

As in other manager views, when you click **Discover** this manager goes to Learn mode, splits into two panes, and executes a discover job. Discover also returns multiple objects, each of which occupies one row in this view. By default, the **Discovered** table for bacnet objects lists the single Bacnet Device object at top and others underneath, with each object occupying a single row.

| Column name | Description |
|-------------|---------------------------------------------------|
| Object Name | Displays the unique name of the device. |
| Object ID | Displays the identifier assigned to the device. |
| Value | Displays the current value for the device. |
| Description | Provides additional information about the device. |

Database pane

| Column name | Description |
|----------------|---------------------------------------------------|
| Path | Displays the path to the device in the station. |
| Name | Displays the device name. |
| Type | Displays the device type. |
| Value | Displays the current value for the device. |
| Object ID | Displays the identifier assigned to the object. |
| Object Name | Displays the object's unique name. |
| Description | Displays additional information about the object. |
| Poll Frequency | Displays how often the driver polls the device. |

Buttons

- **New Folder** creates a new folder for devices. Each such folder provides its own set of manager views.

- **New** creates a new device record in the database.
- **Edit** opens the device's database record for updating.
- **Discover** runs a discover job to locate installed devices, which appear in the **Discovered** pane. This view has a standard appearance that is similar to all **Device Manager** views.
- **Cancel** ends the current discovery job.
- **Add** inserts into the database a record for the discovered and selected object.
- **Match** associates a discovered device with a record that is already in the database.
- **TagIt** associates metadata, such as location or unique configuration with the object.

Manager menu

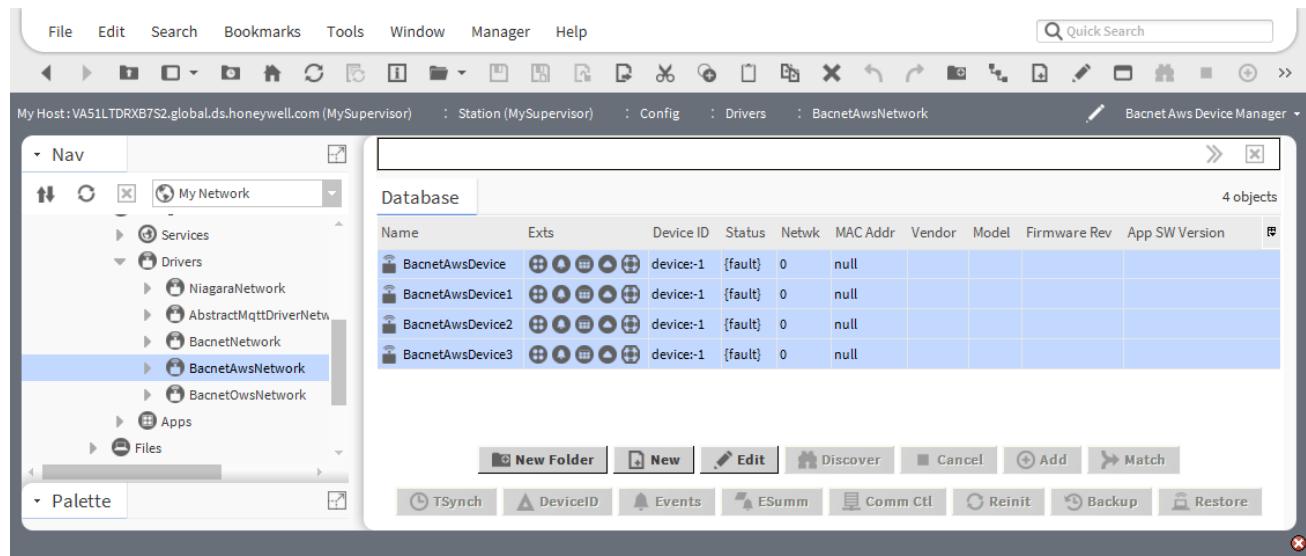
These actions are available on the **BacnetDeviceFolder** Manager menu.

| Menu item | Description |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| New Folder | Creates a new folder (same function as the New Folder button). |
| All Descendents | Expands the selected folder to reveal all subordinate objects. |
| New | Creates a new device record in the database. |
| Edit | Opens the Edit window for updating the selected record. |
| Learn Mode | Controls the discovery of devices and adding them to the database. |
| Discover | Initiates a discovery job (same as clicking the Discover button). |
| Cancel | Stops the current job. |
| Add | Opens an Add window with properties to configure before adding the selected record(s) in the Discovery pane to the database. |
| Quick Add | Adds the selected record(s) in the Discovery pane to the database. |
| Match | Opens a window with properties to configure before matching the discovered record with a record in the database. |
| Quick Match | Matches the selected record with its sister record that is already in the database. |
| Tag Mode | Opens a Tag Dictionary pane for managing metadata tags. The <i>Niagara Tagging Guide</i> documents tag. |
| TagIt | Associates an item of metadata, such as location or unique configuration, with the object. |

Bacnet Aws Device Manager

The **Bacnet Aws Device Manager** is the default view for the BacnetAwsNetwork (and any BacnetAwsDeviceFolder).

All BacnetAwsDevices resides under the BacnetAwsNetwork. A BacnetAwsDevice differs from other BACnet device by its device extensions, Trend Logs, Config, and the default views on those extensions.

Figure 170 Bacnet Aws Device Manager

To open this view, expand **Config→Drivers** and double-click **BacnetAwsNetwork**.

Discovered pane

As in other manager views, when you click **Discover** this manager goes to Learn mode, splits into two panes, and executes a discover job. Discover also returns multiple aws device objects, each of which occupies one row in this view. Following are the columns in the Discovered pane:

Table 3 Columns

| Column name | Description |
|-------------|--------------------------------------------------|
| Device Name | Displays the name of the device. |
| Device ID | Displays the device ID. |
| Netwrk | Displays the device's network number. |
| Mac Addr | Displays the Media Access Control (MAC) address. |
| Vendor | Displays the name of the device manufacturer. |
| Model | Displays the device model name. |
| Objects | Lists BACnet objects. |

Database Pane

Following are the columns and buttons in the database pane:

Table 4 Columns

| Column name | Description |
|-------------|----------------------------------------------------------------------------------------------------------|
| Name | Displays the name of the device. |
| Ext | Displays the device extension's hyperlinks, including: Points, Alarms, Schedules, Trend Logs and Config. |
| Device ID | Displays the device ID. |
| Status | Displays the current status of the device. |

| Column name | Description |
|-------------------|------------------------------------------------------|
| Network | Displays the device's network number. |
| Mac Addr | Displays the Media Access Control (MAC) address. |
| Vendor | Displays the name of the device manufacturer. |
| Model | Displays the device model name. |
| Firmware Revision | Displays the current firmware version of the device. |

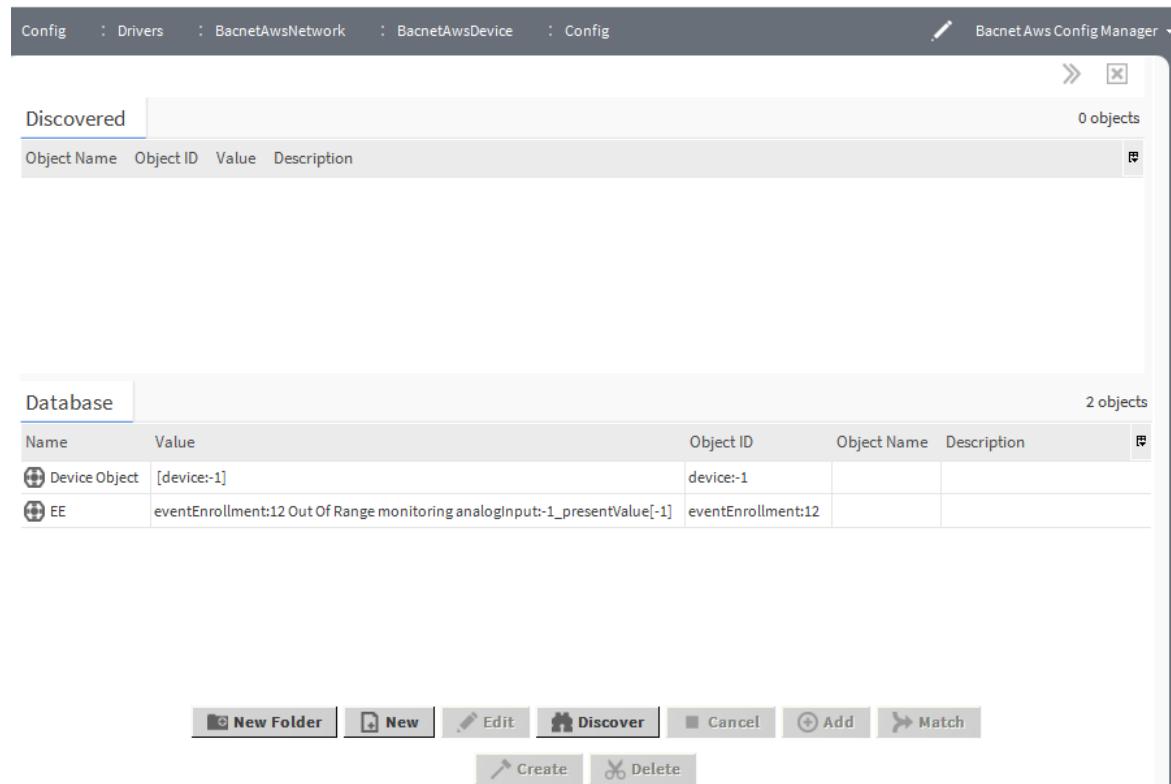
Buttons

- **New Folder** creates a new folder for devices. Each such folder provides its own set of manager views.
- **New** creates a new device record in the database.
- **Edit** opens the device's database record for updating.
- **Discover** runs a discover job to locate installed devices, which appear in the **Discovered** pane. This view has a standard appearance that is similar to all **Device Manager** views.
- **Cancel** ends the current discovery job.
- **Add** inserts into the database a record for the discovered and selected object.
- **Match** associates a discovered device with a record that is already in the database.
- **TSync** synchronizes the time.
- **DeviceID** changes the device ID in the device.
- **Events** retrieves the current event information and sends it to the alarm console.
- **ESumm** opens the **Get Enrollment Summary** window. This window
- **Comm ctl** opens the **Device Communication Control** window.
- **Reinit** opens the **ReinitializeDevice** window.
- **Backup** backs up the selected device record.
- **Restore** restores a backed-up device record.

Bacnet Aws Config Manager

The default view of the Bacnet Aws Config Device Ext is the **Bacnet Aws Config Manager**.

Figure 171 Bacnet AWS config manager



The **Bacnet Aws Config Manager** is also the default view for any `BacnetAwsConfigFolder` under the `Config` container. To view, right-click `BacnetAwsConfigDeviceExt` or `BacnetAwsConfigFolder` and select **Views→Bacnet Aws Config Manager**.

The **Bacnet Aws Config Manager** provides all the same functions as the regular **Bacnet Config Manager** for a BacnetDevice in a BacnetNetwork, plus allows the creation or deletion of BACnet objects (if supported by the target BACnet device).

Discovered pane

As in other manager views, when you click **Discover** this manager goes to Learn mode, splits into two panes, and executes a discover job. Discover also returns multiple aws device objects, each of which occupies one row in this view. Following are the columns in the Discovered pane:

Table 5 Columns

| Column name | Description |
|-------------|-------------------------------------------------|
| Object name | Displays the object's unique name. |
| Object ID | Displays the identifier assigned to the object. |
| Value | Displays the value. |
| Description | Displays additional information about device. |

Database Pane

Following are the columns and buttons in the database pane:

Table 6 Columns

| Column name | Description |
|----------------|-------------------------------------------------|
| Name | Display the name of the object. |
| Type | Displays the object type. |
| Value | Displays the current value. |
| Object ID | Displays the identifier assigned to the object. |
| Object name | Displays the object's unique name. |
| Description | Displays additional information about device. |
| Poll Frequency | Displays how often the driver polls the device. |

Buttons

- **New Folder** creates a new folder for devices. Each such folder provides its own set of manager views.
- **New** creates a new device record in the database.
- **Edit** opens the device's database record for updating.
- **Discover** runs a discover job to locate installed devices, which appear in the **Discovered** pane. This view has a standard appearance that is similar to all **Device Manager** views.
- **Cancel** ends the current discovery job.
- **Add** inserts into the database a record for the discovered and selected object.
- **Match** associates a discovered device with a record that is already in the database.
- **Create** adds one or more new BACnet objects to the remote BACnet device. A popup **New** window selects the BACnet object type and numbers to add.
- **Delete** sends a delete-object request to the remote BACnet device, which removes the BACnet object from the remote BACnet device.

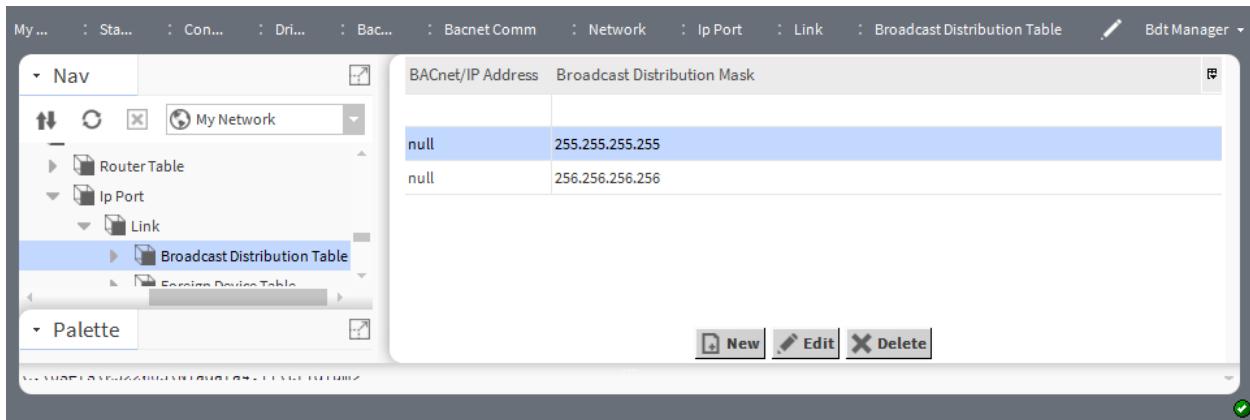
NOTE: The **Delete** key on the keyboard performs a local delete only. The delete keyboard action removes the selected BACnet object from the AWS device **Config** folder but it does not send a delete-object request to the remote device.

Bdt Manager

This is the default view of the **Broadcast Distribution Table** (BDT) under the **Link** container of an **IP Port**.

When the station operates as a BBMD (BACnet broadcast Management Device), it maintains this table listing all participating BBMDs, including the IP address and broadcast distribution mask for each. If needed, you can manually create and edit individual BBMDs.

Figure 172 Bdt Manager View



To open this view, expand **Config→Drivers→BacnetNetwork→Bacnet Comm→Network→Ip Port→Link**, and double-click **Broadcast Distribution Table**.

Columns

| Column name | Description |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bacnet/IP Address | Displays the IP address of the BBMD. |
| Broadcast Distribution Mask | Displays the subnet mask or all ones (1s). This mask indicates if a BBMD is to send a directed broadcast (retransmitted by appropriately configured IP routers) or a unicast message to the indicated BBMD, which then retransmits the forwarded broadcast message. |

Buttons

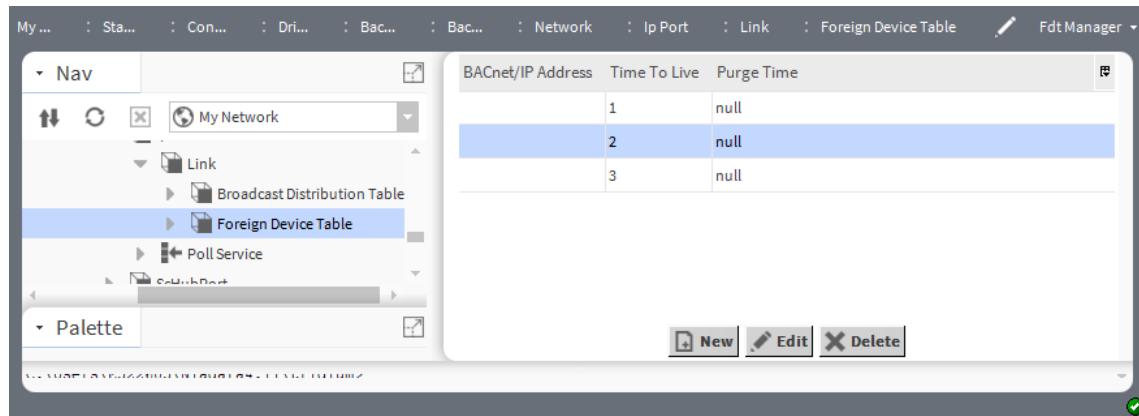
- New** creates a new device record in the database.
- Edit** opens the device's database record for updating.
- Delete** sends a delete-object request to the remote BACnet device, which removes the BACnet object from the remote BACnet device.

NOTE: The **Delete** key on the keyboard performs a local delete only. The delete keyboard action removes the selected BACnet object from the AWS device **Config** folder but it does not send a delete-object request to the remote device.

Fdt Manager

This is the default view of the **Foreign Device Table** under the **Link** container of **Ip port**.

When the station operates as a BBMD (BACnet broadcast Management Device), this table lists other BACnet foreign devices that have registered with the system database, including their IP address, time to live, and purge time. This view allows you to manually edit the FDT, if necessary, to support devices that cannot register themselves.

Figure 173 Fdt Manager view

To open this view, expand **Config→Drivers→BacnetNetwork→Bacnet Comm→Network→Ip Port→Link**, and double-click **Foreign Device Table**.

Columns

| Column name | Description |
|-------------------|------------------------------------------------------------------------------------------------------------|
| Bacnet/IP address | Displays the IP address of the foreign device. |
| Time To Live | Displays the period of time within which a foreign device must re-register or be purged from the database. |
| Purge Time | Reports when the device is scheduled to be removed from the database. |

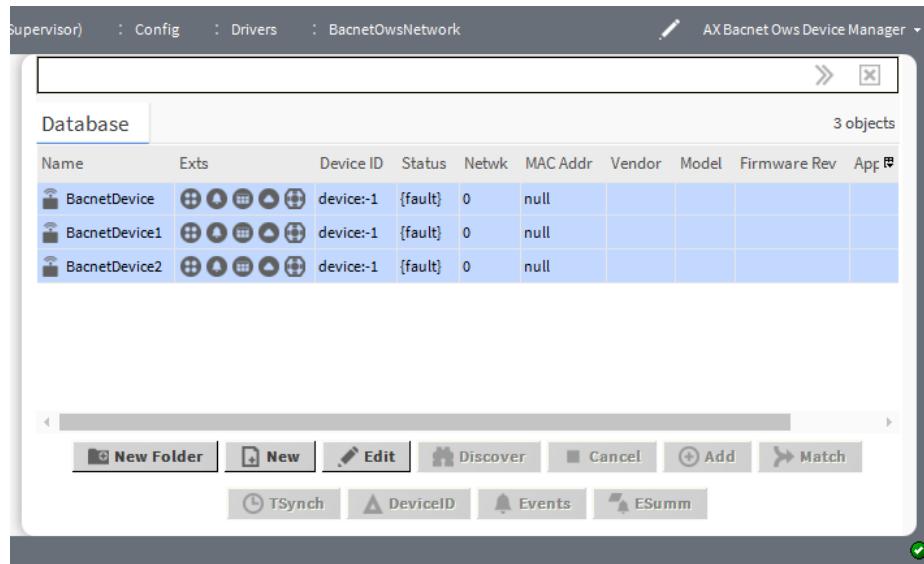
Buttons

- **New** creates a new device record in the database.
- **Edit** opens the device's database record for updating.
-

NOTE: The **Delete** key on the keyboard performs a local delete only. The delete keyboard action removes the selected BACnet object from the AWS device **Config** folder but it does not send a delete-object request to the remote device.

Bacnet Ows Device Manager

The **Bacnet Ows Device Manager** is the default view for the BacnetOwsNetwork (and any BacnetOwsDeviceFolder).

Figure 174 Bacnet OWS Device Manager View

To view, double-click the BacnetOwsNetwork, or right-click the network and select **Views→Bacnet Ows Device Manager**.

Discovered pane

As in other manager views, when you click **Discover** this manager goes to Learn mode, splits into two panes, and executes a discover job. Discover also returns multiple Ows device objects, each of which occupies one row in this view. Following are the columns in the Discovered pane:

Table 7 Columns

| Column name | Description |
|-------------|--------------------------------------------------|
| Device Name | Displays the name of the device. |
| Device Id | Displays the device ID. |
| Network | Displays the network number. |
| Mac Addr | Displays the Media Access Control (MAC) address. |
| Vendor | Displays the name of the device manufacturer. |
| Model | Displays the device model name. |
| Objects | Lists BACnet objects. |

Database pane

Table 8 Columns

| Column name | Description |
|-------------|----------------------------------------------------------------------------------------------------------|
| Name | Displays the name of the device. |
| Exts | Displays the device extension's hyperlinks, including: Points, Alarms, Schedules, Trend Logs and Config. |
| Device ID | Displays the device ID. |
| Status | Displays the current status of the device. |

| Column name | Description |
|-------------------|--------------------------------------------------|
| Network | Displays the network number. |
| Mac Addr | Displays the Media Access Control (MAC) address. |
| Vendor | Displays the name of the device manufacturer. |
| Model | Displays the device model name. |
| Firmware revision | Displays the revision of firmware in use. |

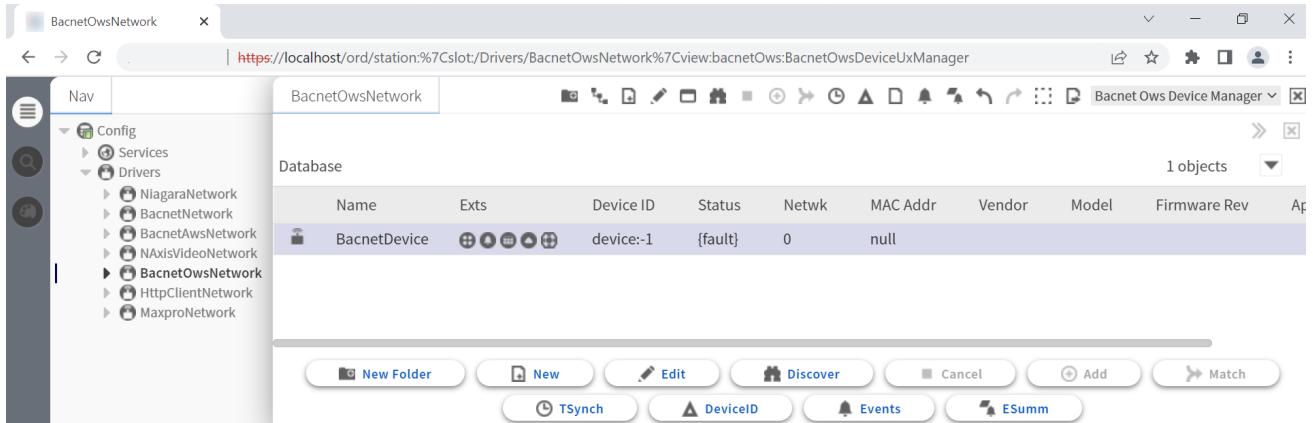
Buttons

- **New Folder** creates a new folder for devices. Each such folder provides its own set of manager views.
- **New** creates a new device record in the database.
- **Edit** opens the device's database record for updating.
- **Discover** runs a discover job to locate installed devices, which appear in the **Discovered** pane. This view has a standard appearance that is similar to all **Device Manager** views.
- **Cancel** ends the current discovery job.
- **Add** inserts into the database a record for the discovered and selected object.
- **Match** associates a discovered device with a record that is already in the database.
- **TSync** synchronizes the time.
- **DeviceID** changes the device ID in the device.
- **Events** retrieves the current event information and sends it to the alarm console.
- **Esumm** opens the **Get Enrollment Summary** window. This window

HTML-5 Bacnet Ows Device Manager

This view is a web-browser-based implementation of a **BacnetOwsNetwork**. You can access it from a browser where it provides the same functions as the equivalent Workbench view.

Figure 175 BacnetOwsDeviceUxManager view



To access this view, expand **Config**→**Drivers** and double-click **BacnetOwsNetwork** or right-click the **BacnetOwsNetwork** and click **Views**→**BacnetOwsDeviceManager**.

Figure 176 Toolbar of Bacnet Ows Device Manager



| Tools | Description |
|--------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| New Folder () | It creates a new folder in the database pane. |
| Trace Descendants () | Displays all descendants or immediate children of the selected parent proxy point. |
| Create new objects () | It creates new object in the database pane. |
| Edit objects () | Opens the device's database record for updating |
| Expand and collapse discovery pane () | Expands and collapses the objects in the database. |
| Start the discovery process () | Runs a discover job to locate installed devices, which appear in the Discovered pane. This view has a standard appearance that is similar to all Device Manager views. |
| Cancel the discovery process () | Ends the current discovery job. |
| Add the discovered object () | Inserts into the database a record for the discovered and selected object. |
| Match the discovered object () | Associates a discovered device with a record that is already in the database. |
| Time Synchronization () | It sends time synchronization to the network. |
| Object_Identifier () | It changes the Device Object ID for the remote devices. |
| Search for object ID () | It searches the objects by name or Id. |
| Get device event information () | It provides the device event information and retrieves the current event information to the alarm console. |
| Get enrollment summary () | It provides the event summary for the event initiating objects. |
| Undo () | Reverses the previous command. |
| Redo () | Restores a command-action after the Undo command has removed it. |
| Multi-selection Mode () | Enables you to individually select multiple points without holding down the ctrl key. |
| Export () | Exports the current view or object. |

Buttons

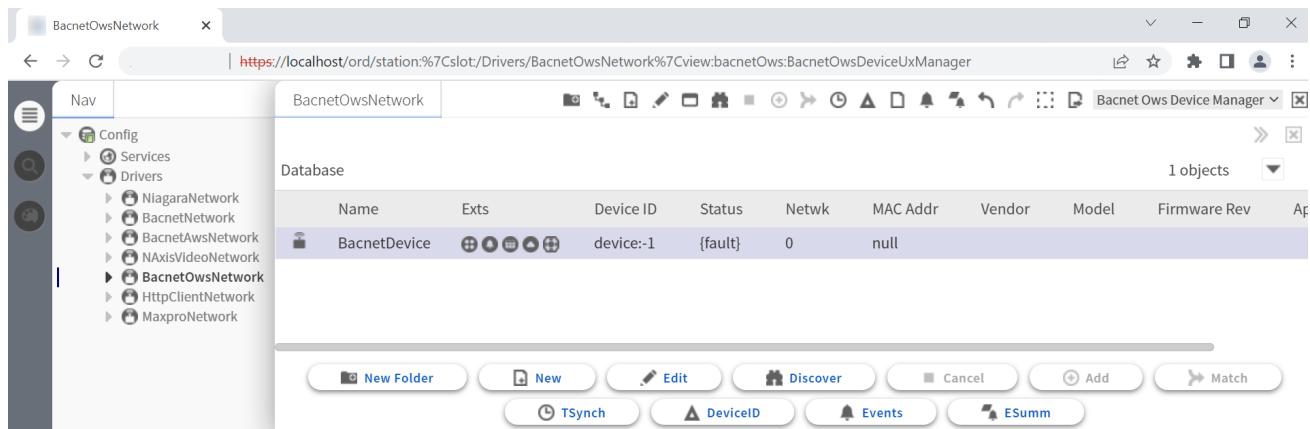
- **New Folder** creates a new folder for devices. Each such folder provides its own set of manager views.
- **New** creates a new device record in the database.
- **Edit** opens the device's database record for updating.
- **Discover** runs a discover job to locate installed devices, which appear in the **Discovered** pane. This view has a standard appearance that is similar to all **Device Manager** views.

- **Cancel** ends the current discovery job.
- **Add** inserts into the database a record for the discovered and selected object.
- **Match** associates a discovered device with a record that is already in the database.
- **TSync** synchronizes the time.
- **DeviceID** changes the device ID in the device.
- **Events** retrieves the current event information to the alarm console.
- **ESumm** provides the event summary for the event initiating objects.

HTML 5 Bacnet Device UxManager View

In Niagara 4.13 and later, there is added browser support for the Niagara Network Device Manager View. The HTML 5 version of this view is a web-browser-based implementation and it provides the same functions as the Workbench view.

Figure 177 Bacnet Device UxManager View



You can access it from a browser of your desktop or mobile device. To access this view from the browser, expand **Config**→**Drivers** and double-click **BacnetNetwork** or right-click **BacnetNetwork** and click **Views**→**BacnetDeviceManager**.

Figure 178 HTML 5 Toolbar



This toolbar offers familiar Workbench. The following tools, however, are unique to the HTML5 view.

| | |
|---|----------------------------------------------------------------------------------------------------------------------|
| 1 | Trace Descendants Displays all descendants or immediate children of the selected parent proxy point. |
| 2 | Multi-selection Mode Enables you to individually select multiple points without holding down the ctrl key. |

Chapter 15 Windows

Topics covered in this chapter

- ◆ Configure Device Discovery window
- ◆ Device Communication Control window
- ◆ EDE Export Window
- ◆ EDE Configuration Window
- ◆ Get Enrollment Summary window
- ◆ New (and Edit) Bacnet Descriptors
- ◆ New (and Edit) BBMD record
- ◆ New (and Edit) device window
- ◆ New (and Edit) device history window
- ◆ New (and Edit) export folder window
- ◆ Add (and Edit) export history window
- ◆ New (and Edit) export objects window
- ◆ New (and Edit) file export window
- ◆ New (and Edit) foreign device record
- ◆ New (and Edit) point window
- ◆ New (and Edit) window for schedule import
- ◆ New (Edit) window for schedule export
- ◆ ReinitializeDevice window
- ◆ Search for objects by name or id (who-has config) window
- ◆ Synchronize Time window

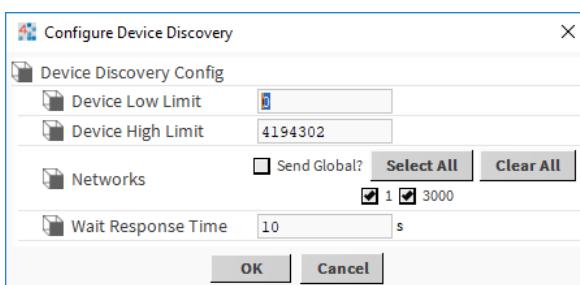
Windows create and edit database records or collect information when accessing a component. You access them by dragging a component from a palette into a station or by clicking a button.

Windows do not support **On View (F1)** and **Guide on Target** help. To learn about the information each contains, search the help system for key words.

Configure Device Discovery window

This window filters the discovery process, allowing you to limit the range of device IDs or to enable discovery to find all networks (global search).

Figure 179 Bacnet Configure Device Discovery



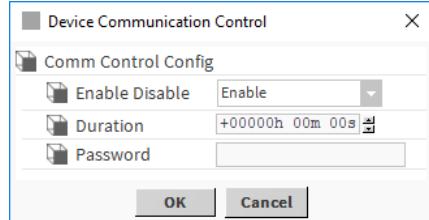
This window opens when you click **Discover** in the **Bacnet Device Manager**.

| Property | Value | Description |
|---------------------------------|----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Device Low Limit | number (defaults to zero (0)) | Defines the device Instance Number below which discovery ignores devices. |
| Device High Limit | number (defaults to 4194302) (the maximum valid BACnet device Instance Number) | Defines the device Instance Number above which discovery ignores devices. |
| Networks, Send Global? | check box | Configures where to send messages. Send Global? (All Networks) - Database searches on all possible BACnet networks segments, including all locally connected networks, as well as segments accessed over BACnet routers. |
| Networks, Select All, Clear All | buttons and check boxes (default to selected) | Selects <Known networks (by network number) > - Choices reflect the number of network ports available under Bacnet Comm, Network component, plus any additional remote networks discovered by receiving "I-am-router" messages. Click to include/exclude as needed. By default, all known networks are individually pre-selected. |
| Wait Response Time | seconds (defaults to 10) | Determines how many seconds the framework waits before determining that all BACnet devices that exist have responded to the discovery request. |

Device Communication Control window

This window controls the configuration of communication properties.

Figure 180 Device Communication Control properties



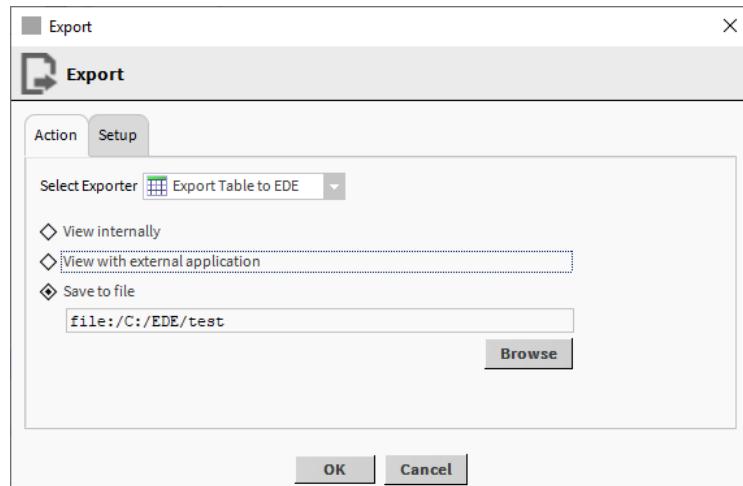
To open this window, in the Supervisor station, expand **Config→Drivers**, double-click **BacnetAwsNetwork** and click **Comm Ctl**.

| Property | Value | Description |
|----------------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Enable Disable | drop-down list (defaults to Enable) | <p>Permits and prohibits the configuration of communication properties.</p> <p>Enable permits configuration of these properties.</p> <p>Disable prohibits communication property configuration.</p> <p>Disable Initiation permits configuration for the amount of time defined by Duration, after which the properties can no longer be configured.</p> |
| Duration | hours minutes seconds | Defines a period of time during which device communication control may be configured. |
| Password | text | Defines a password to require when configuring communication control. |

EDE Export Window

This window exports configured devices and points from a BACnet network.

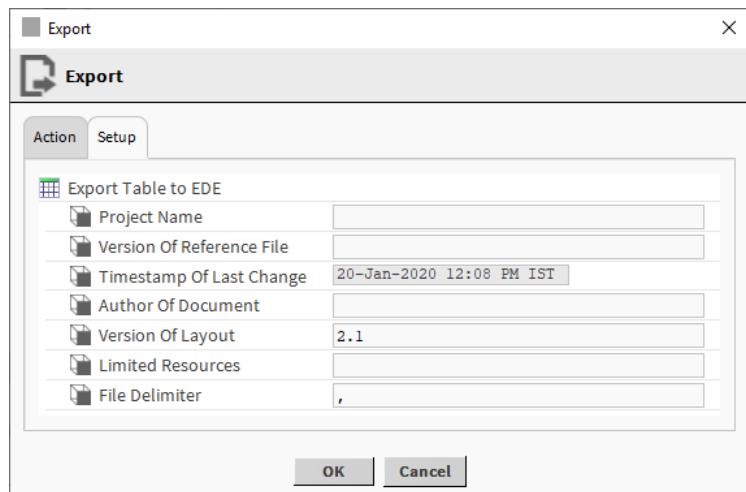
Figure 181 Export window Action tab



This window opens when you right-click the Export Table folder under a BACnet network Local Device and click **Export**.

This window has following properties:

- **Select Exporter** selects the Export Table to EDE from the drop-down list.
- **Save to File** saves the data to the location identified by clicking the **Browse** button.

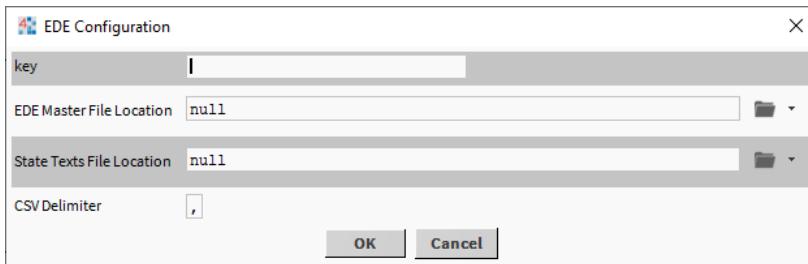
Figure 182 Export window Setup tab

This window opens when you right-click the **Export Table** folder under **Local Device** in the BACnet network, click **Export** and click the **Setup** tab.

| Property | Value | Description |
|---------------------------|---------------------------------------|-----------------------------------------------------------------|
| Project Name | text | Specifies the project name for the EDE configuration. |
| Version Of Reference File | text | Specifies the version of the file. |
| Timestamp of Last change | read only | Displays the time stamp when the file is being exported. |
| Author of the Document | text | Specifies the user who created the document. |
| Version of Layout | read only | Displays the version of the layout as configured in the driver. |
| Limited Resources | text | Mentions extra information about the file. |
| File Delimiter | punctuation (defaults to a comma (,)) | Specifies the delimiter used for the file. |

EDE Configuration Window

This window assigns a key to a pair of EDE configuration files to make discovery easier.

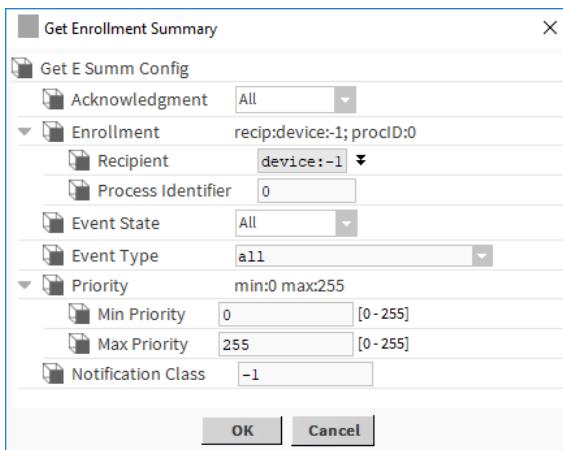
Figure 183 EDE Configuration Window

To access this window double-click the **Bacnet Network** node in the Nav tree, click **Discover**, select a discovered EDE device and click **Add**.

| Property | Value | Description |
|--------------------------|------------------------------------------|--------------------------------------------------------------------------------------------------------|
| Key | text (defaults to blank) | Sets up a short text string to identify the EDE master configuration file. |
| EDE Master file Location | text (defaults to null) | Selects the path to the EDE master configuration file. |
| State text file | text (defaults to null) | Selects to the state text file. |
| CSV Delimiter | single-character (defaults to comma (,)) | Configures the symbol that separates (delimits) individual property values in the configuration files. |

Get Enrollment Summary window

This window

Figure 184 Get Enrollment Summary properties

To open this window, in the Supervisor station, expand **Config→Drivers**, double-click **BacnetAwsNetwork** and click **ESumm**.

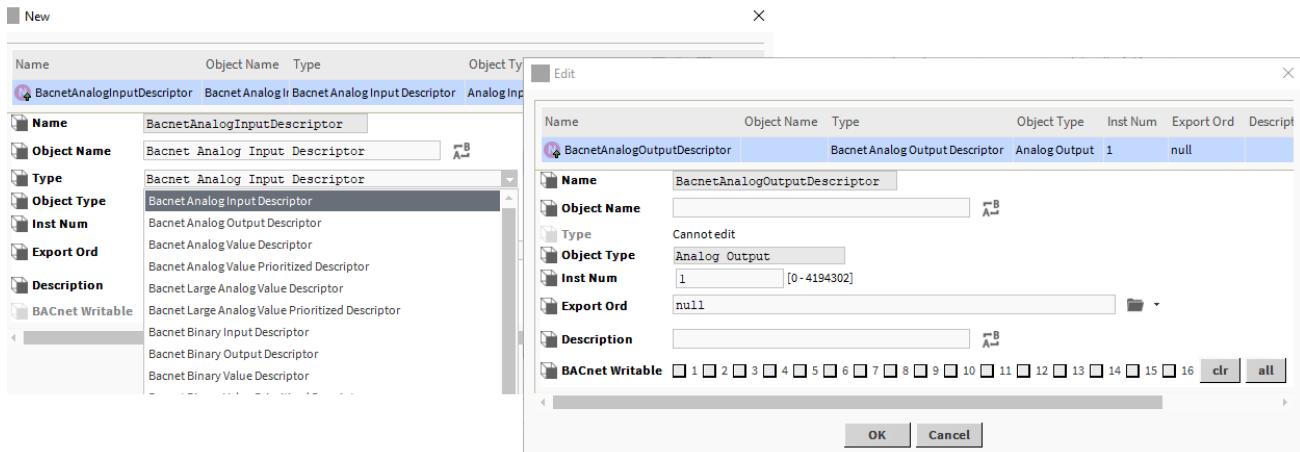
| Property | Value | Description |
|--------------------------------|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Acknowledgment | drop-down list (defaults to All) | Selects alarm records based on if they have been acknowledged or not. All Acked Not Acked |
| Enrollment, Recipient | chooser | Identifies the alarm recipient. |
| Enrollment, Process Identifier | number (defaults to zero (0)) | Specifies a number in the Recipient List for the station. This integer from the BACnet Notification Class object of the sending device, must match this ID in the Alarms device extension of the receiving device. |
| Event State | drop-down list | Reports the status of the event. |
| Priority, Min Priority | number (defaults to zero (0)) | Configures the lowest priority level recognized by the driver. |
| Priority, Max Priority | number (defaults to 255) | Configures the highest priority level recognized by the driver. |
| Notification Class | number (defaults to -1) | Specifies the notification class used for routing when handling and generating event notifications for this object. |

New (and Edit) Bacnet Descriptors

Descriptors configure import and export jobs.

The Export Table provides a set of descriptors. Each has the same set of properties, although, depending on the descriptor, the **Type** property and **BACnet Writable** properties may be unavailable for editing.

Figure 185 Bacnet Analog Input Descriptor window



To open one of these windows, expand **Config→Drivers→BacnetNetwork→Local Device**, double-click **Export Table**, click **New** and select the type of descriptor from the **Type** to Add drop-down list.

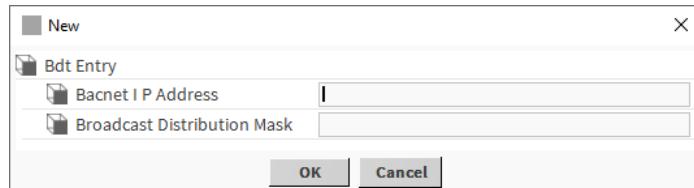
| Property | Value | Description |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name | drop-down list of BACnet objects | <p>Identifies the object (point, export folder, etc.) including an Object Id and Instance Number or property identifier, which ensures a unique name.</p> <p>This name is in the format: <code>objectType_<instance number></code> For example and Event State may be named: AHU1-Frz_Stat-eventState.</p> <p>Refer also to Object Id, Instance Number.</p> |
| Object Name | text in the format <code>name.name.name</code> (default includes the entire component path in the station using period (/) delimiters between parent.child levels) | <p>Identifies a BACnet object. This name should be unique within this specific device.</p> <p>For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device.</p> <p>For a proxy point, this is the default name in the station.</p> <p>You can shorten or edit Object Name, however, each Object Name must be unique among all server descriptors under the Export Table.</p> |
| Type | read-only or a drop-down list | <p>Identifies the type of BACnet object, for example, Analog Input, Analog Output, Calendar, Schedule, Structured View, File, Trend Log, BooleanWritable or other.</p> <p>Writable components typically offer more type selections than read-only points, such as:</p> <ul style="list-style-type: none"> Two read-only: input or value type, e.g. Binary Input or Binary Value object. Two writable: output or prioritized value type, e.g. Binary Output or Binary Value Prioritized object. <p>The screen capture shows an example of type selections when exporting an NdioBooleanWritable point.</p> <p>This property is dimmed if the export Type is read-only, such as for Analog Input or Binary Value.</p> |
| Object Type | read-only | Reports the type of object. |
| Inst Num | number in the range from 0 to 4194302 | <p>Provides a unique number to combine with a device's Object Id. Together, these properties form the Device Id.</p> <p>The Instance Number must be unique within the BACnet internetwork for each entity: components, devices, points, schedules, analog inputs, files, groups, multi state inputs, trend logs, SVO (Structured View Object), etc.</p> <p>The driver automatically increments this value when adding multiple entities.</p> <p>Refer to Object Id.</p> |
| Export Ord | ord (defaults to the format used is the numeric handle instead of slot, which is better if the | Defines the location of the source component or file using standard file Ord notation. In an Edit window, you can access a more meaningful ord by clicking the right-side folder control. This produces a popup Select Ord window, showing the component's location in the station's component tree hierarchy. |

| Property | Value | Description |
|-----------------|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | source object gets renamed) | |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. |
| BACnet Writable | read-only or an array of check boxes | <p>For writable types, provides an array of check boxes that configure the specific priority levels externally exposed to BACnet for the purpose of accepting writes. Included are all controls to clear or select.</p> <p>This property is dimmed if the export Type is read-only, such as for Analog Input or Binary Value.</p> <p>Each priority level (1-16) that you enable for BACnet writes results in that InN input on the source component to be linked to this Bacnet export descriptor component. These links appear as nubs when viewing the source writable point in its wire sheet view.</p> <p>Do not select any priority level already linked or in use by the system, otherwise control contention occurs.</p> <p>After you add a BACnet export descriptor you can configure additional properties accessible in its Property Sheet, along with those that are available in the Add and Edit windows.</p> |

New (and Edit) BBMD record

This window manually creates a new BBMD (BACnet/IP Broadcast Management Device) record in the **Broadcast Distribution Table** of the **Link Ip** layer under the **BacnetNetwork→Bacnet Comm→Network** node.

Figure 186 New BBMD window



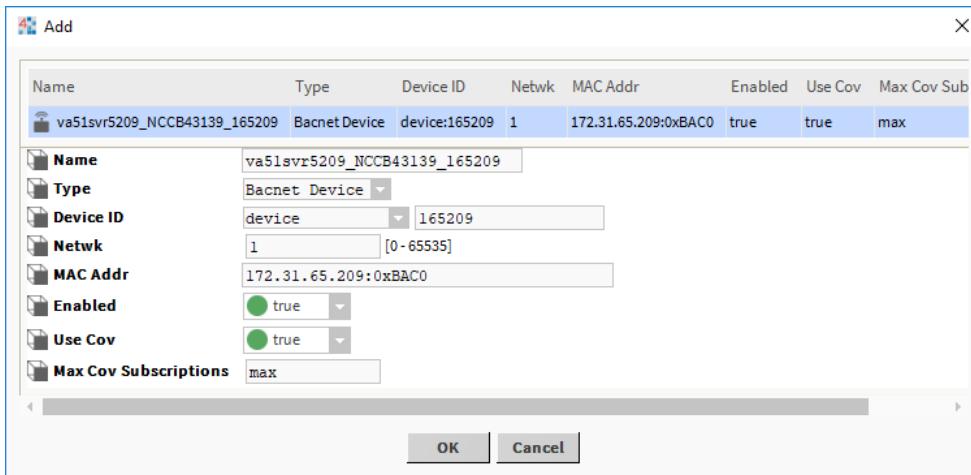
To open this window, expand **Config→Drivers→BacnetNetwork→Bacnet Comm→Network→Ip Port→Link**, double-click **Broadcast Distribution Table** and click **New** or select an existing BBMD record in the table and click **Edit**.

| Property | Value | Description |
|-----------------------------|------------|-------------------------------------------------------------------------------------------------------------------------------------|
| Bacnet IP Address | IP address | Identifies the BBMD IP address. |
| Broadcast Distribution Mask | numbers | Provides a bit mask to apply against the BBMD address. The combination of the two finds the address to which to broadcast packages. |

New (and Edit) device window

This window contains the properties for creating a new device.

New window of the Bacnet Device Manager



This window opens when you expand **Config→Drivers**, double-click **BacnetNetwork** and click **New** or select a row and click **Edit** in the **Bacnet Device Manager**.

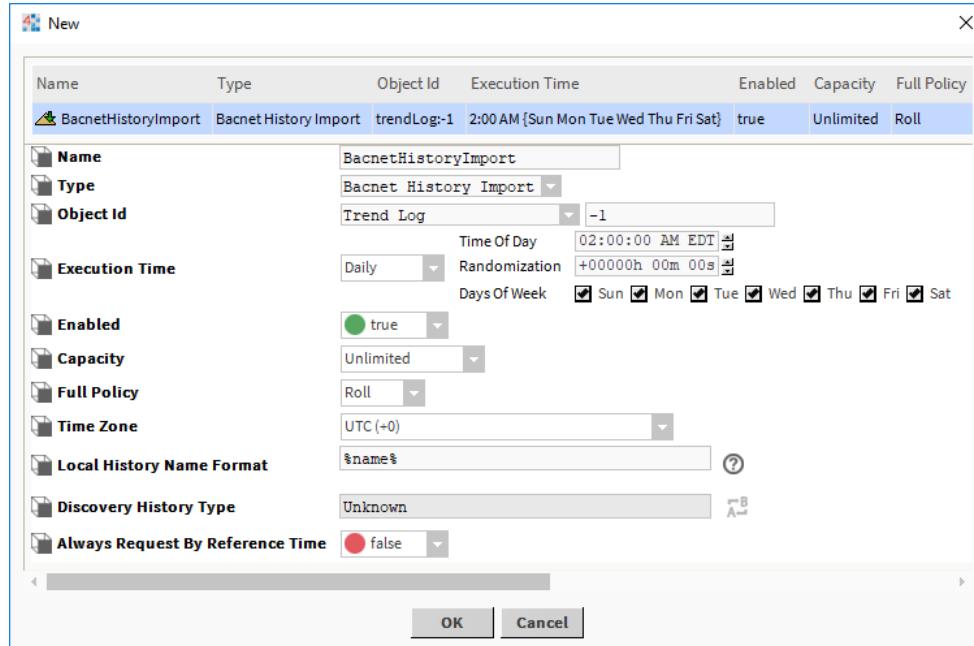
| Property | Value | Description |
|-----------|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name | text | Provides a name for the device. |
| Type | drop-down list | <p>Identifies the type of BACnet object, for example, Analog Input, Analog Output, Calendar, Schedule, Structured View, File, Trend Log, BooleanWritable or other.</p> <p>Writable components typically offer more type selections than read-only points, such as:</p> <ul style="list-style-type: none"> Two read-only: input or value type, e.g. Binary Input or Binary Value object. Two writable: output or prioritized value type, e.g. Binary Output or Binary Value Prioritized object. <p>The screen capture shows an example of type selections when exporting an NdioBooleanWritable point.</p> |
| Device ID | additional properties | <p>Identifies the object as a device and adds a unique instance number.</p> <p>These properties do not affect the device's ID in the station database.</p> |
| Netwk | additional properties | Contains network-related properties. Refer to separate topics in this document. |
| MAC Addr | text | Displays the Media Access Control (MAC) address. |
| Enabled | true (default) or false | Activates (true) and deactivates (false) use of the object (network, device, point, component, table, schedule, descriptor, etc.). |

| Property | Value | Description |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Cov | true or false If the device was discovered, and station database determined that the device indicates support for server-side COV, this property defaults to true. Otherwise, it defaults to false indicating that no proxy points under the device use COV. | <p>Enables (true) and disables (false) a device's support for COV (change of value) as a way to monitor proxy point values.</p> <p>Assuming the device supports subscription to the COV service, true triggers the driver to attempt the necessary updates (proxy subscriptions) to the value of each point using the BACnet COV subscription to the device. If the subscription attempt succeeds, the Read Status property of the point's BacnetProxyExt displays COV. If the subscription attempt fails, the driver uses normal polling and the Read Status property shows Polled.</p> <p>When true, individual proxy points under the device may use COV subscriptions, depending on their assigned tuning policy.</p> <p>When false, the driver ignores any proxy subscription updates.</p> |
| Max Cov Subscriptions | text (defaults to max) | Specifies the maximum number of COV (change of value) subscriptions that the database attempts to use with this device. This restriction prevents the station from consuming all of the available subscription space in the device. Applies only if Use Cov is true. |

New (and Edit) device history window

This window configures device history records.

Figure 187 New window for Bacnet History Import Manager



This window opens when you expand **Config→Drivers→BacnetNetwork→BacnetDevice**, double-click **Trend Logs** and click **New** or select a row and click **Edit** in the **Bacnet History Import Manager**.

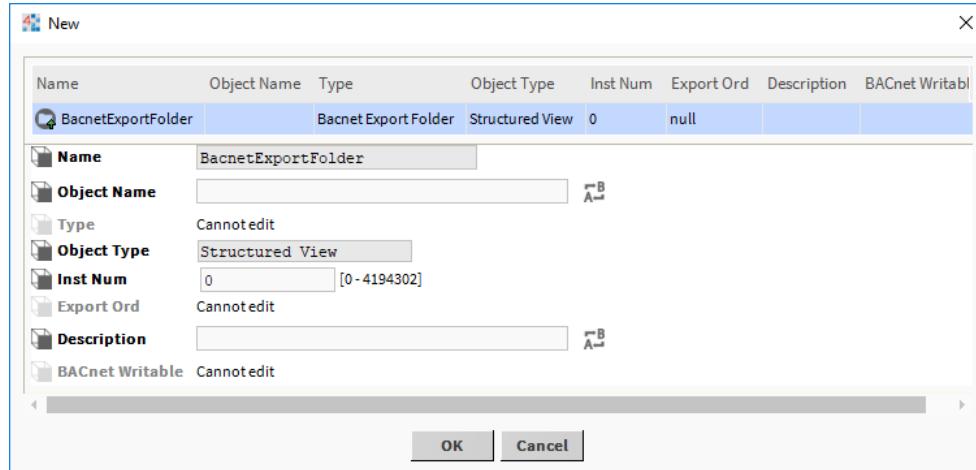
| Property | Value | Description |
|----------------|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name | text | Identifies the history. |
| Type | drop-down list (defaults to Boolean Writable) | <p>Identifies the type of BACnet object, for example, Analog Input, Analog Output, Calendar, Schedule, Structured View, File, Trend Log, BooleanWritable or other.</p> <p>Writable components typically offer more type selections than read-only points, such as:</p> <ul style="list-style-type: none"> Two read-only: input or value type, e.g. Binary Input or Binary Value object. Two writable: output or prioritized value type, e.g. Binary Output or Binary Value Prioritized object. <p>The screen capture shows an example of type selections when exporting an NdioBooleanWritable point.</p> |
| Object ID | Object Id and Instance Number | <p>Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number.</p> <p>For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device.</p> <p>In this case, the object is a schedule.</p> |
| Execution Time | additional properties | <p>Configures the frequency with which data are imported from a remote trend log (or trend log multiple) into a history (or histories). Data import may occur at regular intervals (Interval), Daily or when manually requested (Manual).</p> <p>Time Of Day specifies a specific daily time.</p> <p>Randomization configures an import based on time.</p> <p>Days Of Week specifies on which days to apply the import time.</p> |
| Enabled | true (default) or false | Activates (true) and deactivates (false) use of the object (network, device, point, component, table, schedule, descriptor, etc.). |
| Capacity | drop-down list (defaults to Unlimited) | Specifies the number of trend log records (histories) to store in the histories database. When capacity is reached, newer records overwrite the oldest records. |
| Full Policy | drop-down list | <p>Specifies what happens when a trend log (history) reaches capacity.</p> <p>Applies only if Capacity is set to Record Count. When capacity reaches record count, the newest records overwrite the oldest records.</p> <p>Roll ensures that the latest data are recorded.</p> <p>Stop terminates recording when the number of stored records reaches capacity.</p> <p>Full policy has no effect if Capacity is Unlimited.</p> |
| Time Zone | drop-down list | Sets up the time zone using the Set System Date/Time, which you access either using a platform connection and Platform |

| Property | Value | Description |
|----------------------------------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | Administration→Change Date/Time or using one of the station's PlatformServices views (Platform Service Container plugin or System Date and Time Editor). Otherwise, the time zone is displayed for information only. |
| Local History Name Format | BFormat | Identifies the local history file. |
| Discovery History Type | read-only | |
| Always Request By Reference Time | true or false (default) | Configures if the driver should always use the Reference Time to request records. |

New (and Edit) export folder window

This window creates a new export folder in a station.

Figure 188 New export folder window



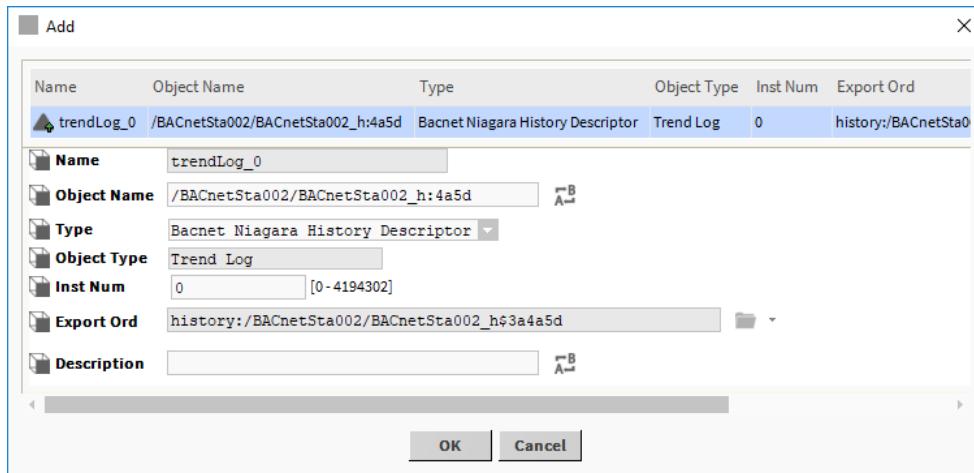
| Property | Value | Description |
|-------------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name | read-only | <p>Identifies the object (point, export folder, etc.) including an Object Id and Instance Number or property identifier, which ensures a unique name.</p> <p>This name is in the format: <code>objectType_<instance number></code> For example and Event State may be named: AHU1-Frz_Stat-eventState.</p> <p>Refer also to Object Id, Instance Number.</p> |
| Object Name | text | Defines the exposed-to-BACnet name for this object. In an export folder this is the name for the SVO (Structured View Object) that the export folder represents. This name must be unique at this level in the structured view hierarchy. |
| Type | cannot be edited | Identifies the record as an export folder. |
| Object Type | read-only | Reports the type of object. |

| Property | Value | Description |
|-----------------|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Inst Num | number in the range from 0 to 4194302 | <p>Provides a unique number to combine with a device's Object Id. Together, these properties form the Device Id.</p> <p>The Instance Number must be unique within the BACnet internetwork for each entity: components, devices, points, schedules, analog inputs, files, groups, multi state inputs, trend logs, SVO (Structured View Object), etc.</p> <p>The driver automatically increments this value when adding multiple entities.</p> <p>Refer to Object Id.</p> |
| Export Ord | cannot be edited | Defines the location of the source component or file using standard file Ord notation. In an Edit window, you can access a more meaningful ord by clicking the right-side folder control. This produces a popup Select Ord window, showing the component's location in the station's component tree hierarchy. |
| Description | | Provides additional information, which could include the camera's geographical location or other unique information. |
| BACnet Writable | cannot be edited | <p>For writable types, provides an array of check boxes that configure the specific priority levels externally exposed to BACnet for the purpose of accepting writes. Included are all controls to clear or select.</p> <p>This property is dimmed if the export Type is read-only, such as for Analog Input or Binary Value.</p> <p>Each priority level (1-16) that you enable for BACnet writes results in that InN input on the source component to be linked to this Bacnet export descriptor component. These links appear as nubs when viewing the source writable point in its wire sheet view.</p> <p>Do not select any priority level already linked or in use by the system, otherwise control contention occurs.</p> <p>After you add a BACnet export descriptor you can configure additional properties accessible in its Property Sheet, along with those that are available in the Add and Edit windows.</p> <p>This property is dimmed if the export Type is read-only, such as for Analog Input or Binary Value.</p> |

Add (and Edit) export history window

This window prepares histories for export.

Figure 189 Add window in Bacnet Log Export Manager



To open this window expand **Config→Drivers→BacnetNetwork→Local Device**, select a history in the Local Objects pane and click **Add**. The Add window contains a number of properties with either default or empty values.

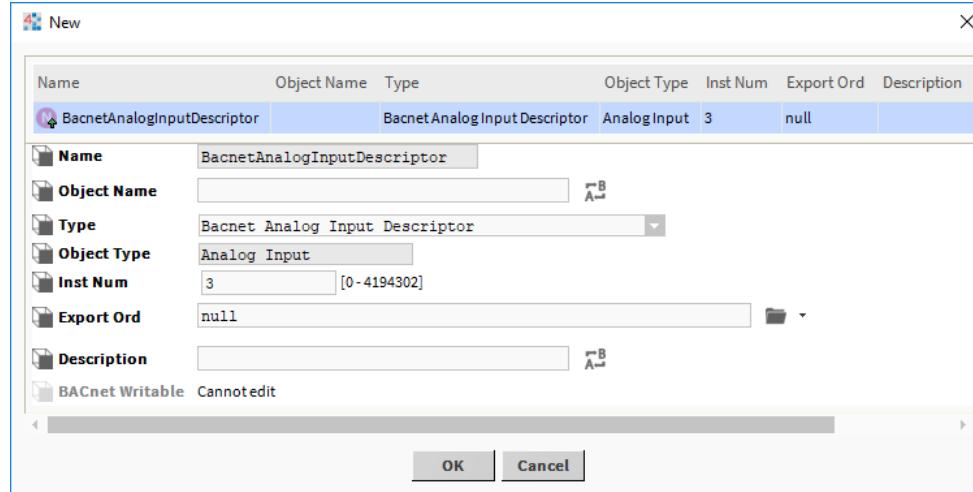
| Property | Value | Description |
|-------------|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name | read-only | <p>Identifies the object (point, export folder, etc.) including an Object Id and Instance Number or property identifier, which ensures a unique name.</p> <p>This name is in the format: <code>objectType_<instance number></code> For example and Event State may be named: <code>AHU1-Frz_Stat-eventState</code>.</p> <p>Refer also to Object Id, Instance Number.</p> |
| Object Name | text | <p>Identifies a BACnet object. This name should be unique within this specific device.</p> <p>For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device.</p> <p>For a proxy point, this is the default name in the station.</p> <p>You can shorten or edit Object Name, however, each Object Name must be unique among all server descriptors under the Export Table.</p> <p>In an export folder this is the name for the SVO (Structured View Object) that the export folder represents. This name must be unique at this level in the structured view hierarchy.</p> |
| Type | drop-down list (defaults to Boolean Writable) | <p>Identifies the type of BACnet object, for example, Analog Input, Analog Output, Calendar, Schedule, Structured View, File, Trend Log, BooleanWritable or other.</p> <p>Writable components typically offer more type selections than read-only points, such as:</p> <ul style="list-style-type: none"> Two read-only: input or value type, e.g. Binary Input or Binary Value object. |

| Property | Value | Description |
|-------------|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | <ul style="list-style-type: none"> Two writable: output or prioritized value type, e.g. Binary Output or Binary Value Prioritized object. <p>The screen capture shows an example of type selections when exporting an NdioBooleanWritable point.</p> |
| Object Type | read-only | Reports the type of object. |
| Inst Num | number in the range from 0 to 4194302 | <p>Provides a unique number to combine with a device's Object Id. Together, these properties form the Device Id.</p> <p>The Instance Number must be unique within the BACnet internetwork for each entity: components, devices, points, schedules, analog inputs, files, groups, multi state inputs, trend logs, SVO (Structured View Object), etc.</p> <p>The driver automatically increments this value when adding multiple entities.</p> <p>Refer to Object Id.</p> |
| Export Ord | chooser | Defines the location of the source component or file using standard file Ord notation. In an Edit window, you can access a more meaningful ord by clicking the right-side folder control. This produces a popup Select Ord window, showing the component's location in the station's component tree hierarchy. |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. |

New (and Edit) export objects window

This window prepares components for export to BACnet objects.

Figure 190 New window in Bacnet Export Manager



This window opens when you expand **Config→Drivers→BacnetNetwork→BacnetDevice**, double-click **ExportFolder** and click **New** or select a row and click **Edit** in the **Bacnet Export Manager**.

All properties in this window apply separately to each highlighted (for export) component.

NOTE:

Select **Type** first before entering other properties, such as **Object Name** and **Description**. Entering type clears the other properties and you will need to enter them again. **Type** is the only property you cannot change after adding an export descriptor.

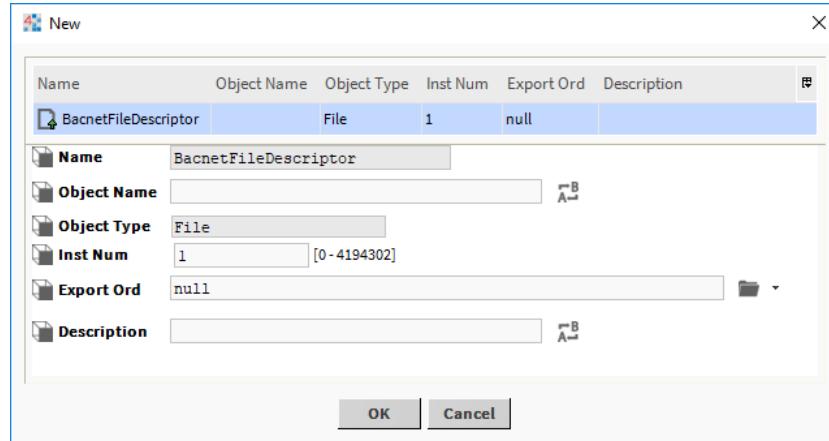
| Property | Value | Description |
|-------------|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name | read-only | <p>Identifies the object (point, export folder, etc.) including an Object Id and Instance Number or property identifier, which ensures a unique name.</p> <p>This name is in the format: <code>objectType_<instance number></code> For example and Event State may be named: <code>AHU1-Frz_EventState</code>.</p> <p>Refer also to Object Id, Instance Number.</p> |
| Object Name | read-only | <p>Identifies a BACnet object. This name should be unique within this specific device.</p> <p>For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device.</p> <p>For a proxy point, this is the default name in the station.</p> <p>You can shorten or edit Object Name, however, each Object Name must be unique among all server descriptors under the Export Table.</p> <p>In an export folder this is the name for the SVO (Structured View Object) that the export folder represents. This name must be unique at this level in the structured view hierarchy.</p> |
| Type | drop-down list (defaults to Boolean Writable) | <p>Identifies the type of BACnet object, for example, Analog Input, Analog Output, Calendar, Schedule, Structured View, File, Trend Log, BooleanWritable or other.</p> <p>Writable components typically offer more type selections than read-only points, such as:</p> <ul style="list-style-type: none"> • Two read-only: input or value type, e.g. Binary Input or Binary Value object. • Two writable: output or prioritized value type, e.g. Binary Output or Binary Value Prioritized object. <p>The screen capture shows an example of type selections when exporting an NdioBooleanWritable point.</p> |
| Object Type | read-only | Reports the type of object. |
| Inst Num | number in the range from 0 to 4194302 | <p>Provides a unique number to combine with a device's Object Id. Together, these properties form the Device Id.</p> <p>The Instance Number must be unique within the BACnet internetwork for each entity: components, devices, points, schedules, analog inputs, files, groups, multi state inputs, trend logs, SVO (Structured View Object), etc.</p> <p>The driver automatically increments this value when adding multiple entities.</p> <p>Refer to Object Id.</p> |

| Property | Value | Description |
|-----------------|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Export Ord | chooser | Defines the location of the source component or file using standard file Ord notation. In an Edit window, you can access a more meaningful ord by clicking the right-side folder control. This produces a popup Select Ord window, showing the component's location in the station's component tree hierarchy. |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. |
| BACnet Writable | cannot be edited | <p>For writable types, provides an array of check boxes that configure the specific priority levels externally exposed to BACnet for the purpose of accepting writes. Included are all controls to clear or select.</p> <p>This property is dimmed if the export Type is read-only, such as for Analog Input or Binary Value.</p> <p>Each priority level (1-16) that you enable for BACnet writes results in that InN input on the source component to be linked to this Bacnet export descriptor component. These links appear as nubs when viewing the source writable point in its wire sheet view.</p> <p>Do not select any priority level already linked or in use by the system, otherwise control contention occurs.</p> <p>After you add a BACnet export descriptor you can configure additional properties accessible in its Property Sheet, along with those that are available in the Add and Edit windows.</p> |

New (and Edit) file export window

This window prepares files for export.

Figure 191 Add window in Bacnet File Export Manager



To open this window, expand **Config→Drivers** in the Nav tree followed by expanding the BACnet network **Local Device**, double-clicking the **Export Table** node and clicking **New**.

| Property | Value | Description |
|-------------|----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name | read-only | <p>Identifies the object (point, export folder, etc.) including an Object Id and Instance Number or property identifier, which ensures a unique name.</p> <p>This name is in the format: <code>objectType_<instance number></code> For example and Event State may be named: <code>AHU1-Frz_Stat-eventState</code>.</p> <p>Refer also to Object Id, Instance Number.</p> |
| Object Name | text | <p>Identifies a BACnet object. This name should be unique within this specific device.</p> <p>For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device.</p> <p>For a proxy point, this is the default name in the station.</p> <p>You can shorten or edit Object Name, however, each Object Name must be unique among all server descriptors under the Export Table.</p> <p>This is the name of the export file, which, by default, includes only the filename and extension.</p> |
| Object Type | read-only | Reports the type of object. |
| Inst Num | number in the range from 0 to 4194302 | <p>Provides a unique number to combine with a device's Object Id. Together, these properties form the Device Id.</p> <p>The Instance Number must be unique within the BACnet internetwork for each entity: components, devices, points, schedules, analog inputs, files, groups, multi state inputs, trend logs, SVO (Structured View Object), etc.</p> <p>The driver automatically increments this value when adding multiple entities.</p> <p>Refer to Object Id.</p> |
| Export Ord | ord (defaults to the format used is the numeric handle instead of slot, which is better if the source object gets renamed) | Defines the location of the source component or file using standard file Ord notation. In an Edit window, you can access a more meaningful ord by clicking the right-side folder control. This produces a popup Select Ord window, showing the component's location in the station's component tree hierarchy. |
| Description | text | Provides additional information, which could include the camera's geographical location or other unique information. |

New (and Edit) foreign device record

This window manually creates a new foreign device record in the **Foreign Device Table** of the **Link Ip** layer under the **BacnetNetwork→Bacnet Comm→Network** node.

Figure 192 New BBMD window



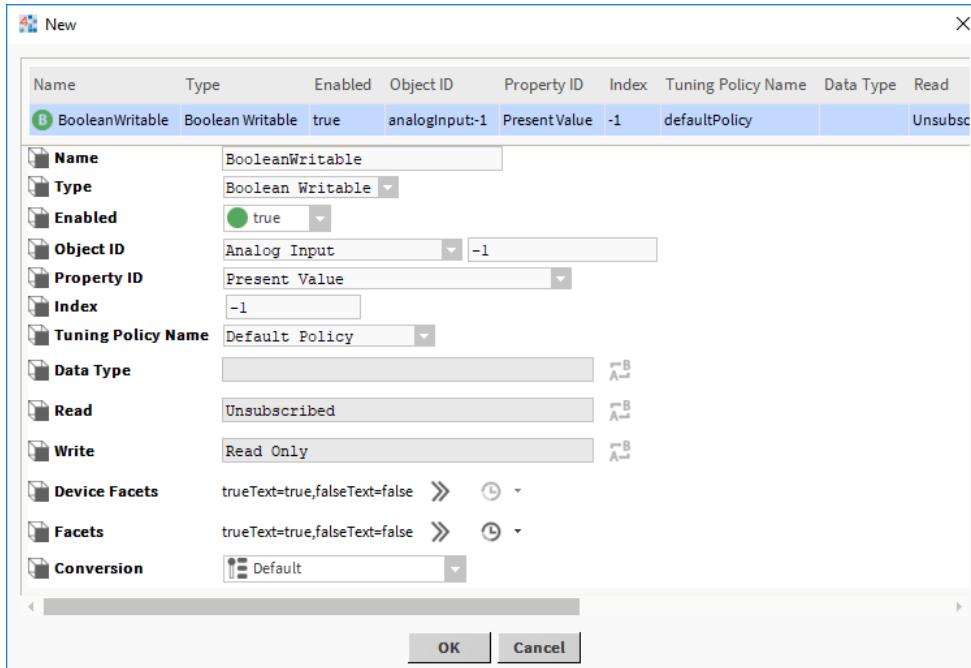
To open this window, expand **Config**→**Drivers**→**BacnetNetwork**→**Bacnet Comm**→**Network**→**Ip Port**→**Link**, double-click **Foreign Device Table** and click **New** or select an existing BBMD record in the table and click **Edit**.

| Property | Value | Description |
|-------------------|-----------------------|-----------------------------------------------------------------------------------------------------------|
| Bacnet IP Address | IP address | Identifies the foreign device IP address. |
| Time To Live | time (defaults to -1) | Defines the period of time within which a foreign device must re-register or be purged from the database. |
| Purge Time | read-only | Reports when the device is scheduled to be removed from the database. |

New (and Edit) point window

This window configures the point properties.

Figure 193 New window for the Bacnet Point Manager



This window opens when you expand **Config**→**Drivers**→**BacnetNetwork**→**BacnetDevice**, double-click **Points** and click **New** or select a row and click **Edit** in the **Bacnet Point Manager**.

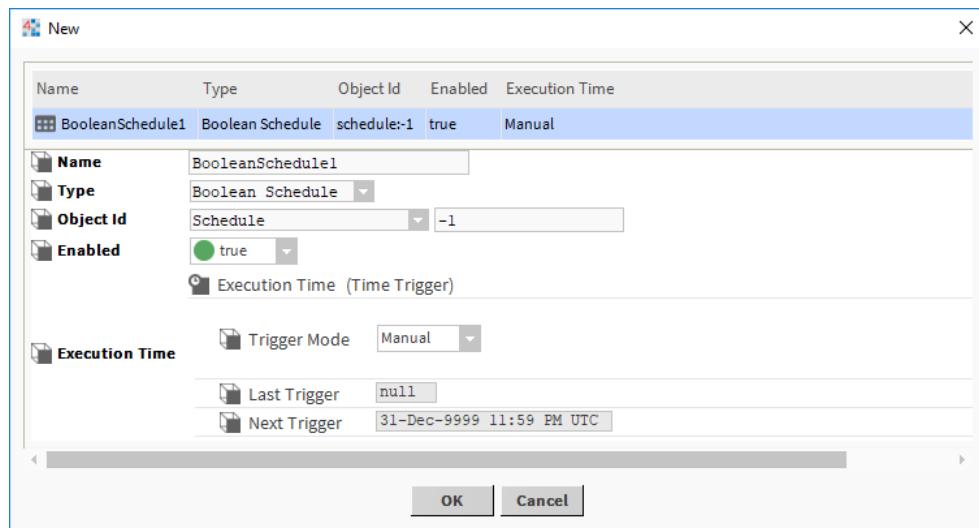
| Property | Value | Description |
|--------------------|----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name | text (defaults to the name in the station for the proxy point) | <p>Identifies the object (point, export folder, etc.) including an Object Id and Instance Number or property identifier, which ensures a unique name.</p> <p>This name is in the format: <code>objectType_<instance number></code> For example and Event State may be named: <code>AHU1-Frz_Stat-eventState</code>.</p> <p>Refer also to Object Id, Instance Number.</p> <p>The exception to this rule is the name for presentValuePoints, where the system uses only the Object Name for the point name. For example: <code>AHU1-Frz_Stat</code>.</p> <p>The system replaces each slash (/) character, if any, with a period (.) in the name. This enforces (externally) the BACnet requirement for unique names for all objects in a device.</p> <p>This is a point-name only. It does not affect the name of the remote BACnet device.</p> |
| Type | drop-down list of BACnet objects | <p>Identifies the type of BACnet object, for example, Analog Input, Analog Output, Calendar, Schedule, Structured View, File, Trend Log, BooleanWritable or other.</p> <p>Writable components typically offer more type selections than read-only points, such as:</p> <ul style="list-style-type: none"> Two read-only: input or value type, e.g. Binary Input or Binary Value object. Two writable: output or prioritized value type, e.g. Binary Output or Binary Value Prioritized object. <p>The screen capture shows an example of type selections when exporting an NdioBooleanWritable point.</p> |
| Enabled | true (default) or false | Activates (true) and deactivates (false) use of the object (network, device, point, component, table, schedule, descriptor, etc.). |
| Object ID | Object Id and Instance Number | <p>Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number.</p> <p>For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device.</p> |
| Property ID | drop-down list | Specifies a number in the Recipient List for the station. This integer from the BACnet Notification Class object of the sending device, must match this ID in the Alarms device extension of the receiving device. |
| Index | number | Provides a numeric index into the property array. Applies only to an arrayed property, such as Priority Array, otherwise it defaults -1 for no index. |
| Tuning Policy Name | drop-down list | Selects a network tuning policy by name. This policy defines stale time and minimum and maximum update times. |

| Property | Value | Description |
|---------------|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | During polling, the system uses the tuning policy to evaluate both write requests and the acceptability (freshness) of read requests. |
| Data Type | read-only | Reports the ASN (Abstract Syntax Notation) data type for a device property. The driver automatically converts values to this appropriate type. For example, an analog input present value is an ASN REAL that can be interpreted by a string point as a character string. ENUMERATED is another example. |
| Read | read-only | Indicates if the point's value was polled, subscribed or resulted from a change of value. |
| Write | read-only | Indicates if the proxy extension is read-only or can be written to. read only indicates the proxy extension cannot be written to. writable or ok indicates that the proxy extension can be written to. For writable proxy extensions, this property indicates either that the last write occurred within the effective tuning policy period or, if a write operation failed, it provides descriptive text. For example, if you create a writable point for a read-only object (say, a NumericWritable for presentValue of an Analog_Input object) and attempt to write from the system, the write status may show: Property:Write Access Denied. If the error is actually a BACnet error, the colon-separated format contains the error class and error code returned by the device. |
| Device Facets | additional properties | Opens a Config Facets window for defining device facets. |
| Facets | additional properties | Opens a Config Facets window for defining binary facets. |
| Conversion | drop-down list (defaults to Default) | Configures the facets to use when converting values from device facets to point facets. |

New (and Edit) window for schedule import

This window can configure a schedule and calendar. You typically leave these properties at their default values.

Figure 194 New window for importing schedules and calendars



This window opens when you expand **Config**→**Drivers**→**BacnetNetwork**→**BacnetDevice**, double-click **Schedules** and click **New** or select a row and click **Edit** in the **Bacnet Schedule Import Manager**.

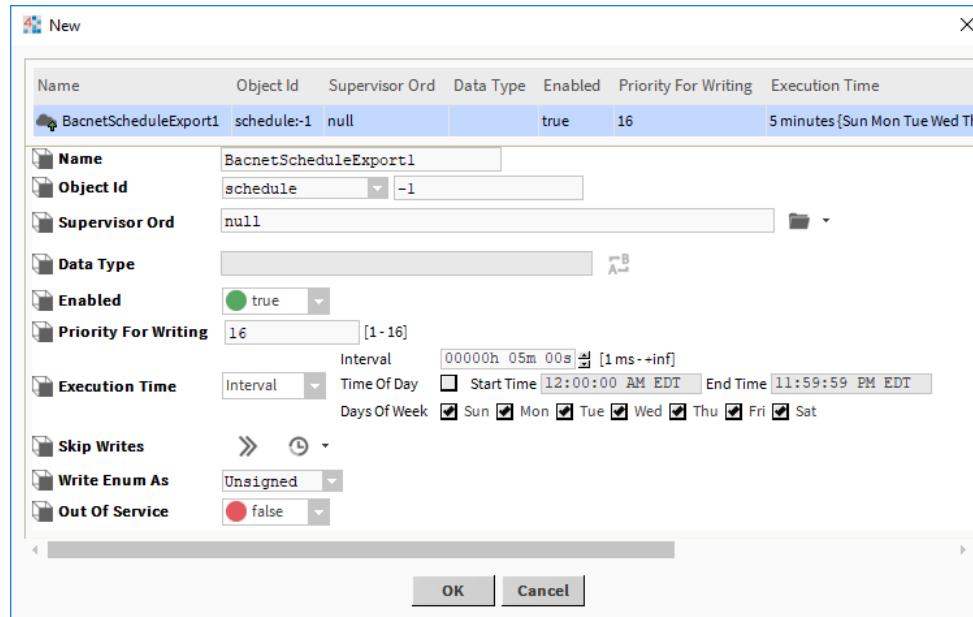
| Property | Value | Description |
|-----------|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name | text | Identifies the Boolean writable point. |
| Type | drop-down list (defaults to Boolean Writable) | <p>Identifies the type of BACnet object, for example, Analog Input, Analog Output, Calendar, Schedule, Structured View, File, Trend Log, BooleanWritable or other.</p> <p>Writable components typically offer more type selections than read-only points, such as:</p> <ul style="list-style-type: none"> Two read-only: input or value type, e.g. Binary Input or Binary Value object. Two writable: output or prioritized value type, e.g. Binary Output or Binary Value Prioritized object. <p>The screen capture shows an example of type selections when exporting an NdioBooleanWritable point.</p> |
| Object ID | Object Id and Instance Number | <p>Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number.</p> <p>For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device.</p> |

| Property | Value | Description |
|----------------|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Enabled | true (default) or false | Activates (true) and deactivates (false) use of the object (network, device, point, component, table, schedule, descriptor, etc.). |
| Execution Time | additional properties | <p>Configures the frequency with which data are imported from a remote trend log (or trend log multiple) into a history (or histories). Data import may occur at regular intervals (Interval), Daily or when manually requested (Manual).</p> <p>Time Of Day specifies a specific daily time.</p> <p>Randomization configures an import based on time.</p> <p>Days Of Week specifies on which days to apply the import time.</p> |

New (Edit) window for schedule export

This window configures a schedule for export.

Figure 195 New window in the Bacnet Schedule Export Manager



This window opens when you expand **Config→Drivers→BacnetNetwork→BacnetDevice**, right-click **Schedules**, click **Views→Bacnet Schedule Export Manager** and click **New** or select a row and click **Edit** in the **Bacnet Schedule Export Manager**.

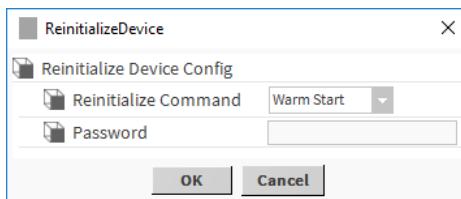
| Property | Value | Description |
|-----------|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name | text | Identifies the schedule |
| Object ID | Object Id and Instance Number | <p>Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number.</p> <p>For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device.</p> |

| Property | Value | Description |
|----------------------|---------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | In this case, the object is a schedule. |
| Supervisor Ord | additional properties | Identifies the local station ORD for the schedule. |
| Data Type | read-only | Reports the ASN (Abstract Syntax Notation) data type for a device property. The driver automatically converts values to this appropriate type. For example, an analog input present value is an ASN REAL that can be interpreted by a string point as a character string. ENUMERATED is another example. |
| Enabled | true (default) or false | Activates (true) and deactivates (false) use of the object (network, device, point, component, table, schedule, descriptor, etc.). |
| Priority for Writing | numeric text values 1 through 16 (defaults to 16) | Sets up schedule priority. |
| Execution Time | additional properties | Configures the frequency with which data are imported from a remote trend log (or trend log multiple) into a history (or histories). Data import may occur at regular intervals (Interval), Daily or when manually requested (Manual). Time Of Day specifies a specific daily time. Randomization configures an import based on time. Days Of Week specifies on which days to apply the import time. |
| Skip Writes | additional properties | Adjusts which property types in a target BACnet schedule object the system writes to upon an export from a weekly schedule component. |
| Write Enum As | drop-down list (defaults to Unsigned) | Defines how to write Enum data as Unsigned: ENUMERATED or as an INTEGER. |
| Out Of Service | true (default) or false | Affects only the BACnet exposure and access to the source component. If the source (exported) component is a proxy point, the equivalent Enabled property in its ProxyExt is not affected. true writes the point's current value to the Present Value property. Often, a remote BACnet client sets (or clears) this Out of Service state. You can set this property to false if needed. |

ReinitializeDevice window

This window configures a password, which the driver requires when someone reboots a Supervisor station.

Figure 196 Reinitialize Device properties



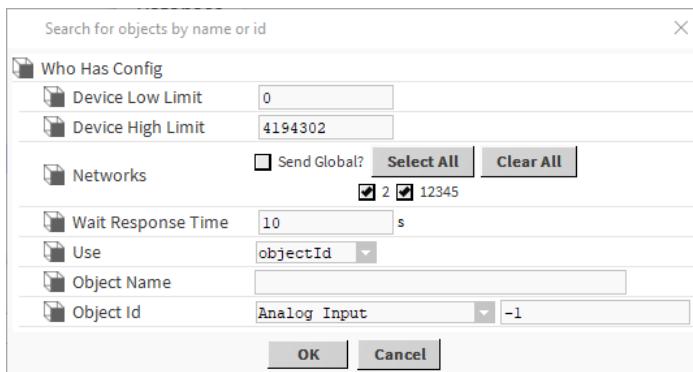
To open this window, in the Supervisor station, expand **Config→Drivers**, double-click **BacnetAwsNetwork** and click **Reinit**.

| Property | Value | Description |
|----------------------|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Reinitialize Command | drop-down list (defaults to Warm Start) | Selects the type of reboot that requires the password. Warm Start requires a password when someone clicks the Windows menu followed by Power→Restart . Cold Start requires a password when someone clicks the Windows menu followed by Power→Shutdown . |
| Password | text | Defines the password. |

Search for objects by name or id (who-has config) window

This window searches for specific objects by name or object ID (type and instance number).

Figure 197 Default who-has popup window



To open this window, click the **Manager** menu and click **Who Has**.

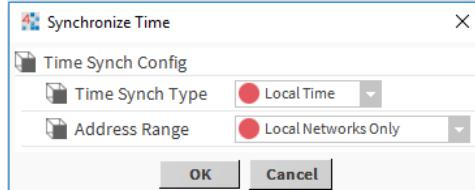
| Property | Value | Description |
|-------------------|----------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Device Low Limit | number (defaults to zero (0)) | Defines the device Instance Number below which discovery ignores devices. |
| Device High Limit | number (defaults to 4194302) (the maximum valid BACnet device Instance Number) | Defines the device Instance Number above which discovery ignores devices. |
| Networks | check boxes | Selects <Known networks (by network number)> - Choices reflect the number of network ports available under Bacnet Comm , network component, plus any additional remote networks discovered by receiving "I-am-router" messages. Click |

| Property | Value | Description |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | to include/exclude as needed. By default, all known networks are individually pre-selected. |
| Wait Response Time | seconds (defaults to 10) | Determines how many seconds the framework waits before determining that all BACnet devices that exist have responded to the discovery request. |
| Use | | |
| Object Name | text in the format name . name . name . name (default includes the entire component path in the station using period ("/") delimiters between parent.child levels) | <p>Identifies a BACnet object. This name should be unique within this specific device.</p> <p>For components, this name includes the entire component path under the station's Config using period (.) delimiters between parent.child levels. This enforces (externally) the BACnet requirement for unique names for all objects in a device.</p> <p>For a proxy point, this is the default name in the station.</p> <p>You can shorten or edit Object Name, however, each Object Name must be unique among all server descriptors under the Export Table.</p> |
| Object Id | Object Id and Instance Number | <p>Identifies each device and component by the type of entity (Object Id) combined with a unique Instance Number.</p> <p>For devices (controllers, instruments, meters, etc.), Instance Number must be unique within the BACnet internetwork. For components (points, schedules, analog inputs, files, groups, multi state inputs, etc.), it must be unique within the device.</p> |

Synchronize Time window

This window configures the time when synchronizing across a network.

Figure 198 Synchronize Time window



| Property | Value | Description |
|----------------|--------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Time Sync Type | drop-down list (defaults to Local Time) | <p>Selects local time or universal time.</p> <p>Local Time uses the current time at the Supervisor station location.</p> <p>UTC Time uses Greenwich Mean Time. It does not observe daylight saving time.</p> |
| Address Range | drop-down list (defaults to Local Networks Only) | Selects a range of addresses. |

Index

A

| | |
|----------------------------------------|-----|
| ACC | 321 |
| Accept Uris array | 176 |
| add device window | 386 |
| AI..... | 232 |
| alarms | 171 |
| configuring a station to receive | 39 |
| configuring a station to send..... | 40 |
| configuring reporting (sending)..... | 40 |
| exporting BACnet objects | 42 |
| receiving..... | 37 |
| testing | 42 |
| analog input | 232 |
| analog input descriptor..... | 263 |
| analog output..... | 234 |
| analog output descriptor | 265 |
| AO..... | 234 |
| architecture | 13 |
| array | 176 |
| AV..... | 236 |
| AWS..... | 16 |
| AwsConfigFolder..... | 315 |

B

| | |
|--------------------------------------------------|----------|
| backup | 43 |
| Bacnet access door..... | 319 |
| Bacnet Accumulator..... | 321 |
| Bacnet Analog Value | 236 |
| Bacnet Analog Value Descriptor | 267 |
| Bacnet Analog Value Prioritized Descriptor | 269 |
| BACnet array..... | 176 |
| Bacnet Aws Config Manager | 370 |
| Bacnet Aws Device Manager | 368 |
| Bacnet Binary Input..... | 237 |
| Bacnet Binary Input Descriptor | 271 |
| Bacnet Binary Output | 239 |
| Bacnet Binary Output Descriptor | 273 |
| Bacnet Binary Value | 241 |
| Bacnet Binary Value Descriptor..... | 275 |
| Bacnet Binary Value Prioritized Descriptor | 277 |
| Bacnet Boolean Schedule Descriptor | 279 |
| Bacnet Calendar | 243 |
| Bacnet Calendar Descriptor | 280 |
| Bacnet Comm properties | 133 |
| Bacnet command..... | 323 |
| Bacnet Config Manager | 366 |
| Bacnet descriptors..... | 384 |
| Bacnet Device Manager | 339, 378 |
| Bacnet Dynamic Schedule Descriptor | 299 |
| Bacnet Enum Schedule Descriptor | 282 |
| Bacnet Event Enrollment Descriptor | 282 |
| Bacnet event log..... | 317 |
| Bacnet Export Manager | 351 |

| | |
|-------------------------------------------------------|-----------|
| Bacnet File | 247 |
| Bacnet File Descriptor..... | 285 |
| Bacnet File Export Manager | 354 |
| Bacnet Group | 247 |
| Bacnet History Import Manager | 173, 363 |
| Bacnet Ip Link Layer..... | 141 |
| Bacnet Load Control | 325 |
| Bacnet Loop | 249 |
| Bacnet Loop Descriptor | 287 |
| bacnet module | 123 |
| Bacnet Multi State Input Descriptor..... | 288 |
| Bacnet Multi State Output Descriptor..... | 289 |
| Bacnet Multi State Value Descriptor | 291 |
| Bacnet Multi State Value Prioritized Descriptor | 293 |
| Bacnet Multistate Input..... | 251 |
| Bacnet Multistate Output..... | 253 |
| Bacnet Multistate Value | 255 |
| Bacnet Network | 378 |
| Bacnet Niagara History Descriptor | 295 |
| Bacnet Niagara Log Export Manager | 81, 356 |
| Bacnet Notification Class | 256 |
| Bacnet Notification Class Descriptor | 296 |
| Bacnet Numeric Schedule Descriptor..... | 298 |
| Bacnet Ows Device Manager..... | 374, 376 |
| bacnet palette | 31 |
| Bacnet Point Device Ext | 179 |
| Bacnet Point Manager | 344 |
| Bacnet Program | 258 |
| Bacnet pulse converter | 327 |
| Bacnet Schedule | 259 |
| Bacnet Schedule Export Manager | 361 |
| Bacnet Schedule Import Manager | 358 |
| Bacnet String Schedule Descriptor | 301 |
| Bacnet structured view | 329 |
| Bacnet Trend Log | 261 |
| Bacnet Trend Log Descriptor | 302 |
| Bacnet Trend Log Multiple | 263 |
| Bacnet Trend Multiple View | 366 |
| Bacnet Tuning Policy | 165 |
| BACnet user | 83 |
| bacnet-BacnetBitStringTrendLogExt | 230 |
| bacnet-BacnetExportFolder | 131 |
| bacnet-CrlDescriptor | 151 |
| bacnet-DirectInitiatingConnection | 155 |
| bacnet-HubConnector | 158 |
| bacnet-HubConnectorHealth | 159 |
| bacnet-HubFunctionConnections | 162 |
| bacnet-IssuerCertAndCrl | 150 |
| bacnet-JettyScWebSocketInitiator | 152 |
| bacnet-ScConfiguration | 147 |
| bacnet-ScCredentials | 149 |
| BACnet/IP | 90, 95–96 |
| BACnet/Ip diagrams | 88 |
| BACnet/SC..... | 95–96 |
| BacnetAlarmDeviceExt | 171 |

| | | | |
|------------------------------------------------|---------------------------|----------|-----|
| bacnetAws-BacnetDynamicObjects | 312 | BV | 241 |
| BacnetAwsConfigDeviceExt | 315 | | |
| BacnetAwsDevice | 313 | | |
| BacnetAwsDeviceFolder | 312 | | |
| BacnetAwsNetwork | 307 | | |
| BacnetBooleanCovTrendLogExt | 209 | | |
| BacnetBooleanIntervalTrendLogExt | 210 | | |
| BacnetBooleanTrendLogExt | 212 | | |
| BacnetBooleanTrendLogRemoteExt | 213 | | |
| BacnetConfigDeviceExt | 174 | | |
| BacnetDestination | 178 | | |
| BacnetDevice | 168 | | |
| BacnetDeviceFolder | 168 | | |
| bacnetEDE-WbEdeService | 123 | | |
| BacnetEnumCovTrendLogExt | 215 | | |
| BacnetEnumIntervalTrendLogExt | 216 | | |
| BacnetEnumProxyExt | 185 | | |
| BacnetEnumTrendLogExt | 218 | | |
| BacnetEnumTrendLogRemoteExt | 219 | | |
| BacnetEthernetLinkLayer | 144 | | |
| BacnetEventEnrollment | 245 | | |
| BacnetEventLogImport | 315 | | |
| BacnetHistoryImport | 197 | | |
| BacnetMultiPoll | 162 | | |
| BacnetNetwork | 124 | | |
| BacnetNumericCovTrendLogExt | 202 | | |
| BacnetNumericIntervalTrendLogExt | 203 | | |
| BacnetNumericProxyExt | 182 | | |
| BacnetNumericTrendLogExt | 205 | | |
| BacnetNumericTrendLogRemoteExt | 207 | | |
| BacnetObject | 232 | | |
| BacnetOwsDeviceFolder | 336 | | |
| BacnetOwsNetwork | 331, 376 | | |
| BacnetPointFolder | 170 | | |
| BacnetScheduleExport | 194 | | |
| BacnetScheduleImportExt | 193 | | |
| BacnetStact | 133 | | |
| BacnetStringCovTrendLogExt | 221 | | |
| BacnetStringIntervalTrendLogExt | 223 | | |
| BacnetStringProxyExt | 189 | | |
| BacnetStringTrendLogExt | 224 | | |
| BacnetStringTrendLogRemoteExt | 226 | | |
| BacnetTrendLogAlarmSourceExt | 228 | | |
| BacnetTrendLogExt | 81 | | |
| BacnetTrendLogMultipleImport | 199 | | |
| BacnetVirtualGateway | 171 | | |
| BacnetVirtualObject | 171 | | |
| BacnetVirtualProperty | 171 | | |
| BBMD | 90–91, 117, 143, 372, 386 | | |
| configuring | 94 | | |
| configuring a station | 91 | | |
| bdt manager | 372 | | |
| best practice | 19 | | |
| BI | 237 | | |
| BO | 239 | | |
| Boolean proxy extension | 179 | | |
| broadcast distribution table | 141 | | |
| Broadcast Distribution Table | 143 | | |
| BTL certification | 16 | | |
| | | | |
| | | C | |
| Cal | 243 | | |
| certificate | | | |
| architecture | 98 | | |
| importing signed into the User Key Store | 107 | | |
| site/issuer | 100 | | |
| certificate expiration | 117 | | |
| certificates | | | |
| signing multiple | 105 | | |
| client | 134 | | |
| client/server certificates | 98 | | |
| CMD | 323 | | |
| components | 337 | | |
| bacnetAws module | 307 | | |
| bacnetOws module | 331 | | |
| discovering to export | 75 | | |
| config | 174, 315 | | |
| Config | 232 | | |
| ConfigFolder | 232 | | |
| configuration files (EDE) | 46 | | |
| Configure Device Discovery window | 379 | | |
| confirmed worker | 134 | | |
| connections | 154 | | |
| controller | | | |
| configuration | 26 | | |
| controller station | | | |
| set up under the Supervisor network | 27 | | |
| COV | | | |
| subscribing proxy points | 60 | | |
| COV subscriptions | 59 | | |
| CSR, See Certificate Signing Request | | | |
| | | D | |
| data | | | |
| archive | 197 | | |
| data gathering | 59 | | |
| Default Policy | 165 | | |
| delete object | 74 | | |
| descriptor | 265 | | |
| descriptors | | | |
| for exporting data | 76 | | |
| device | | | |
| adding | 29–31 | | |
| configuration | 29 | | |
| configuring to send alarms | 39 | | |
| searching for | 57 | | |
| Device Communication Control window | 380 | | |
| Device Histories view | 365 | | |
| device history | 388 | | |
| device IDs | | | |
| configuring | 31 | | |
| device management | 57 | | |
| device object | 174 | | |

| | |
|----------------------------------|-----|
| device polling | 59 |
| devices | |
| discovering offline..... | 51 |
| matching..... | 30 |
| devices and points..... | 49 |
| diagrams | 88 |
| document change log | 11 |
| Door | 319 |
| dynamic creation of objects..... | 69 |

E

| | |
|---------------------------------|-----|
| EDE Bacnet Device manager | 342 |
| EDE bacnet Point Manager | 349 |
| EDE configuration | 49 |
| EDE configuration files..... | 46 |
| EDE configuration utility | 45 |
| EDE Configuration window | 382 |
| EDE Export window..... | 381 |
| edit device window..... | 386 |
| EE..... | 245 |
| Elog | 317 |
| Ethernet port | |
| configuration | 25 |
| EthernetPort | 140 |
| Event Enrollment..... | 282 |
| Event Handler | 135 |
| event parameters | 247 |
| Event Saver | 284 |
| export descriptors | 130 |
| deleting | 76 |
| export table | 129 |

F

| | |
|----------------------------|----------|
| Failover Connection..... | 160 |
| failover hub | |
| setting up | 113 |
| faq | 117 |
| fdt manager | 373 |
| file | |
| exporting..... | 79 |
| File..... | 247 |
| file descriptors..... | 130 |
| file export | 395 |
| foreign device | 373, 396 |
| configuring | 93 |
| foreign device table | 141 |
| Foreign Device Table | 143 |

G

| | |
|------------------------------------|-----|
| Get Enrollment Summary window..... | 383 |
| Grp | 247 |

H

| | |
|----------------------------------------------|---------|
| history..... | 61 |
| standard export | 82 |
| history export..... | 81, 391 |
| HistoryConfig..... | 337 |
| host | |
| installing on an IP subnet with a BBMD | 91 |
| rebooting..... | 120 |
| hub | 117 |
| failure | 118 |
| setting up security..... | 107 |
| hub connection | |
| enabling..... | 113 |
| hub function | |
| adding | 111 |
| Hub Function..... | 161 |

I

| | |
|--------------------------|--------|
| installation..... | 13, 15 |
| internetwork | |
| diagrams..... | 85 |
| network numbers | 85 |
| routers..... | 85 |
| internetwork rules | 89 |
| internetworks | 85 |
| IP subnet..... | 91 |
| IpPort | 140 |
| issuer certificate | 100 |

J

| | |
|---------------------|----|
| job log (EDE) | 53 |
|---------------------|----|

L

| | |
|-------------------------------|-----|
| LCO | 325 |
| license..... | 14 |
| link..... | 144 |
| Link (Ip) | 141 |
| Link (Mstp) | 145 |
| Link (ScPort) | 146 |
| list of | 132 |
| local bacnet AWS device | 309 |
| local device | 125 |
| local bacnet Ows device | 332 |
| log descriptors | 131 |
| Loop | 249 |

M

| | |
|-----------------------|-----|
| message flow | 97 |
| message wrapper | 98 |
| module | |
| bacnet | 123 |
| MS/TP | 145 |

| | |
|------------------------------------------------|------------|
| MS/TP devices | 16 |
| improving station startup | 120 |
| MS/TP port | 26 |
| MSI | 251 |
| MSO | 253 |
| MstpLinkLayer | 145 |
| MstpPort | 140 |
| MSV | 255 |
| N | |
| NC | 256 |
| network | |
| adding | 24 |
| adding in a Supervisor station | 20 |
| installing BACnet/SC | 115 |
| number | 85 |
| set up controller station under the Supervisor | |
| network | 27 |
| network layer | 138 |
| network port | 140 |
| network properties | 138 |
| new device window | 386 |
| New point window | 397 |
| New window | |
| for export folders | 390 |
| for history import | 388 |
| for object export | 393 |
| for schedule export | 401 |
| for schedule import | 399 |
| node certificates | 98 |
| node switch | |
| setting up | 114 |
| Node Switch | 152 |
| O | |
| object handler | 70 |
| object requests with initial values | 70 |
| objects | |
| dynamic creation | 69 |
| enabling write for exported objects | 83 |
| objects export | 393 |
| objects to export | 74 |
| Operational certificates | |
| client certificate | 103 |
| creating | 103 |
| server certificate | 103 |
| OutOfServiceExt | 192 |
| OWS | 16 |
| P | |
| palette (Supervisors) | 16 |
| PC | 327 |
| performance | |
| improving | 304 |
| PICS | 16 |
| plugins | 339 |
| points | |
| container | 179 |
| discovering offline | 52 |
| Points | 179 |
| Poll Service | 162 |
| port | |
| setting up security | 115 |
| ports | 25–26, 140 |
| prerequisites | 14 |
| Primary Connection | 160 |
| priority array | 176 |
| Prog | 258 |
| properties | |
| polling | 34 |
| Proxy Ext | |
| Enum | 185 |
| Proxy Ext (Boolean) | 179 |
| proxy point | |
| export descriptors | 76 |
| proxy point overwrite | 55 |
| proxy points | |
| adding | 33 |
| Px views and virtual points | 67 |
| R | |
| ReinitializeDevice window | 402 |
| related documentation | 12 |
| requirements | 14 |
| supervisor stations | 16 |
| restart notification recipients | 132 |
| router table | 139 |
| rules for internetworks | 89 |
| S | |
| Sc Link Layer | 146 |
| scalability | 118 |
| Sched | 259 |
| schedule | |
| exporting | 78 |
| importing | 68 |
| Schedule | 193 |
| schedules | 172 |
| ScHubPort | 107 |
| ScPort | 115 |
| secure | |
| hub, setting up | 107 |
| port | 115 |
| secure communication port | 146 |
| Secure Connect | 97 |
| server | |
| object handler | 136 |
| Server | 263 |
| server configuration | 69 |

| | |
|------------------------------------------------|---------|
| server descriptors..... | 129–130 |
| ServerAIDesc | 263 |
| ServerAODesc..... | 265 |
| ServerAVDesc | 267 |
| ServerAVPDesc | 269 |
| ServerBIDesc..... | 271 |
| ServerBODesc..... | 273 |
| ServerBoolSchedDesc..... | 279 |
| ServerBVDesc..... | 275 |
| ServerBVPDesc..... | 277 |
| ServerCalendarDesc | 280 |
| ServerEnumSchedDesc..... | 282 |
| ServerFileDesc | 285 |
| ServerLoopDesc | 287 |
| ServerMSIDesc | 288 |
| ServerMSODesc | 289 |
| ServerMSVDesc | 291 |
| ServerMSVPDesc..... | 293 |
| ServerNCDesc | 295 |
| ServerNumSchedDesc..... | 298 |
| ServerStringSchedDesc..... | 301 |
| sever properties | 134 |
| site certificate..... | 100 |
| slot | |
| configuring for polling | 35 |
| stack: unresolved device address | 53 |
| state texts support file..... | 48 |
| station | |
| configuring as a BBMD | 91 |
| configuring as a foreign device | 93 |
| set up controller station under the Supervisor | |
| network | 27 |
| Supervisor palettes..... | 16 |
| supervisor station requirements | 16 |
| SVO | 329 |
| Svo Subordinate Manager view | 353 |
| synchronization recipients | 132 |
| synchronize time window | 404 |

U

| | |
|--------------------------------------|-----|
| utc synchronization recipients | 132 |
|--------------------------------------|-----|

V

| | |
|----------------------------|---------|
| validation | 50 |
| views..... | 339 |
| virtual c omponents | 171 |
| virtual components | 63, 171 |
| virtual device properties | |
| configuring..... | 64 |
| virtual points in Px views | |
| configuring | 67 |
| virtual property | 171 |
| VPN | 118 |

W

| | |
|---------------------------|-----|
| Web Socket Acceptor | 153 |
| WebService | 112 |
| Who Has service | 57 |
| who-has function | 403 |
| windows..... | 379 |
| worker | 134 |
| Worker Pool | 304 |

T

| | |
|--------------------------------|------------------------|
| time | |
| synchronizing | 58 |
| time synchronization recipient | |
| adding | 58 |
| TLM | 263 |
| TLog | 261 |
| transport properties | 137 |
| trend log | 81 |
| trend log multiple | 263 |
| trend logs..... | 61, 173, 199, 315, 363 |
| add and edit..... | 388 |
| Trending | 197 |
| troubleshooting..... | 119 |
| EDE | 53 |
| tuning policies | 165 |
| configuring | 59 |
| tuning policy map | 165 |

Glossary

| | |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| APDU | Application (layer) Protocol Data Unit is a BACnet message—that is, a unit of data specified within any of the BACnet link-layer protocols. It consists of protocol control information and possibly application user data. |
| BACnet | Building Automation and Control NETworking protocol (ANSI/ASHRAE Standard 135- 2004) is an open communication protocol standard conceived by a consortium of manufacturers and system users under the auspices of ASHRAE. BACnet modles data a common set of objects, which can accessed using a standard set of services. |
| BACnet/Ethernet or B/ETH | BACnet over Ethernet, one of the original BACnet link-layer types. Since the introduction of BACnet/IP, its usage has become less common. |
| BACnet/IP or B/IP | BACnet over Ethernet IP was introduced in Annex J of the BACnet standard. It has become the most popular BACnet link-layer protocol (except in lowest-cost devices, which use MS/TP). |
| BACnet client | An operational mode for a BACnet system that makes use of a BACnet device for some particular purpose via a service request instance. |
| BACnet server | An Operational mode for a BACnet system that provides a service to a requesting client. |
| BBMD | BACnet Broadcast Management Device is a device that receives and redistributes broadcasttype BACnet messages (Who-Is, I-Am, etc.) to other B/ IP devices on its own subnet and sends broadcasttype messages to BBMDs on other subnets. By having one BBMD on each subnet, a B/IP network can span subnets (between IP routers, which otherwise typically block broadcast type messages). A station supports operation as a BBMD. |
| BTL | BACnet Testing Laboratories was established by BACnet International to support the compliance and interoperability testing of BACnet products. The BACnet AWS Supervisor and BACnet OWS Supervisor are each BTL-certified. |
| Config object | A model of a physical item, process, calculation or operation in a client BACnet device where you can see all the properties of the object together. |
| COV or Cov | Change-of-Value. Characterizes the option to track data based on when a value changes rather than at a consistent interval, such as every minute, 15 minutes, etc. |
| data link layer | This is the second layer of the OSI (Open Systems Interconnection) software model. This layer works between two hosts, which are directly connected (point-to-point or broadcast). At the sending end, the data link layer converts data into bit-by-bit signals that are compatible with the hardware. At the receiving end it re-assembles the electrical signals from the hardware into recognizable information for processing. The result hides the underlying hardware details, providing the communication medium to the upper layers of the model. (www.tutorialspoint.com) |
| device instance | The logical address on an MS/TP link or IP network that matters to BACnet. This number is unique across all subnets and routed links. (www.csimn.com) |
| Device object or device component | Describes any component that represents an external device. It has several device extensions, one being the Config Device Ext . By default, this extension contains a (frozen) device object component, which represents the |

| | |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>single BACnet device object in that device. You cannot delete this component. You can add and delete other Config objects as needed.</p> <p>On the server side, the configuration of a station's single Local Device component under its BacnetNetwork represents the station's device object.</p> |
| foreign device | <p>A device that exists on an IP subnet without a BBMD. This device can register with a BBMD on another (remote) subnet as a foreign device to explicitly receive BACnet broadcast messages. In no way does it imply any reduced functionality.</p> |
| internetwork | <p>To be read: inter-network. Two or more BACnet networks connected by a BACnet router, or essentially "everything on your site that can be accessed via BACnet."</p> <p>The single BacnetNetwork in a station often represents an internetwork if any external BACnet routers exist or if it has multiple ports under its BacnetComm→Network component. If the latter, the station can act as a BACnet router among its local networks.</p> <p>In a BACnet internetwork, each network must have a unique Network Number, from 0 to 65534. Each device must have a unique Object Id, from 0 to 4194302.</p> |
| MS/TP or MSTP | <p>BACnet link-layer protocol is used by lower-cost devices to pass master slave / tokens over RS-485 multidrop networks. QNX-based controllers support direct MS/TP (network) trunks, one per RS-485 port (if licensed for MS/TP).</p> |
| object identifier or object ID | <p>A BACnet method to identify a particular object within a device using a combination of its object type and an instance number (unique for that type, within that device). In the case of the single device object (type) per BACnet device, it must have a unique instance number across the BACnet internetwork on which it is installed.</p> |
| structured view object | <p>A presentation-related type of object. BACnet objects generally refer to sensors, actuators, and other functional elements that make up a BACnet device. Objects fall into categories specified by the protocol. Analog input and output objects are a couple of the most commonly used objects. (www.cismn.com)</p> |
| udp/ip | <p>User Datagram Protocol (UDP) is a networking protocol with minimal overhead that promotes network speed and efficiency. BACnet IP uses the UDP side of the IP stack. This is appropriate since messages are typically short and independent of each other.</p> |