

Technical Document

# OPC UA Driver Guide

February 3, 2025

niagara<sup>4</sup>

# Legal Notice

## Tridium, Incorporated

3951 Western 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 © 2025 Tridium, Inc. All rights reserved.

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

For an important patent notice, please visit: <http://www.honpat.com>.

# Contents

<b>About this Guide</b>	5
Document change log	5
Related documentation	6
<b>Chapter 1. Overview</b>	7
Driver component model	7
Requirements	8
Third-party simulation software	9
About the security architecture	9
Security profiles	10
<b>Chapter 2. OPC UA server tasks</b>	13
OPC UA server, client user authorization	13
Setting up the OPC UA server	13
Configuring OPC UA Server Security Modes	15
Configuring the server to support an OPC UA client user	15
Using Self-Signed Certificates from an allowed host	17
OpcUaServer alarm processing	17
Adding a server point with alarm and history extensions	18
Verifying server setup using client simulation software	19
Generating an OPC UA server certificate	21
Generate an OPC UA server certificate for Third-Party Clients	25
<b>Chapter 3. OPC UA client tasks</b>	31
Adding the OPC UA network	31
Connecting to an OPC UA server	31
Discovering OPC UA server points	36
Adding points containing OPC UA histories	40
Importing OPC UA Histories without using a control point	41
Important note about OPC UA Histories	43
OPC UA Client Histories	43
Adding a control point to invoke an OPC UA method	45
OpcUaServer alarm acknowledgment processing	46
Subscribing for OPC UA alarm events	46
Generating an OPC UA client certificate	49
Generate an OPC UA Client Certificate for Third-Party Servers	52
<b>Chapter 4. Opc Ua Reference</b>	57
Point type mapping	57
Point facet mapping	57
Components	58
opcUaServer-OpcUaServer	58
opcUaServer-OpcUaNamespace	61

opcUaServer-OpcUaServerPointDeviceExt .....	62
opcUaServer-OpcUaServerAlarmDeviceExt .....	62
opcUaServer-OpcUaServerDeviceFolder .....	64
opcUaServer-OpcUaServerPointFolder .....	64
opcUaServer-OpcUaAlarmClass .....	64
opcUaServer-OpcUaAlarmRecipient .....	66
opcUaServer-OpcUaAuthenticationScheme .....	68
opcUaClient-OpcUaNetwork .....	69
opcUaClient-OpcUaDevice .....	71
opcUaClient-OpcUaBuildInfo .....	73
opcUaClient-OpcUaClientAlarmDeviceExt .....	74
opcUaClient-OpcUaClientPointDeviceExt .....	75
opcUaClient-OpcUaDeviceFolder .....	77
opcUaClient-OpcUaClientHistoryDeviceExt .....	77
opcUaClient-OpcUaClientPointFolder .....	78
opcUaServer-OpcUaServerDeviceManager .....	78
OpC Ua Server Alarm Manager .....	79
OpC Ua Server Point Manager .....	79
OpC Ua Client Device Manager .....	82
OpC Ua Client Alarm Manager .....	83
OpC Ua Client Point Manager .....	83
OpC Ua Client History Import Manager .....	84
<b>Chapter 5. Glossary .....</b>	<b>87</b>

## 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 describes the procedures used to create an OPC UA Server in a station for the purpose of exposing control points, alarms, and histories to OPC UA clients. This document also describes the procedures used to create an OPC UA Client in the station which provides connectivity to OPC UA Servers for the purpose of integrating Opc UA data into the Niagara Framework for monitor and control operation.

**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.

### February 10, 2025

- Added new topic "OPC UA Client Histories" in the "OPC UA client tasks" chapter.
- Added update about OPC UA Server authentication changes with Niagara 4.14.u1 and Niagara 4.15.

### January 23, 2024

- Added new topic "Configuring OPC UA Server Security Modes" to the "OPC UA server tasks" chapter.
- Edited topic "Setting up certificate based authentication" to the "OPC UA server tasks" chapter.
- Added new topic "Generating an OPC UA server certificate" to the "OPC UA server tasks" chapter.
- Added new topic "Generate an OPC UA server certificate for Third-Party Clients" to the "OPC UA server tasks" chapter.
- Edited "Connecting to an OPC UA server" topic in the "OPC UA client tasks" chapter.
- Added new topic "Important note about OPC UA Histories" to the "OPC UA client tasks" chapter.
- Edited "Generating an OPC UA client certificate" to the "OPC UA client tasks" chapter.
- Edited "Generate an OPC UA Client Certificate for Third-Party Servers" to the "OPC UA client tasks" chapter.

### July 12, 2023

- Added column descriptions for all the manager views.
- Added new topic "Generate an OPCUA Client Certificate for Third-Party Servers" to the "Opc Ua Client tasks" chapter.

### August 22, 2022

Added Discovery Preferences to two topics "Discovering OPC UA Server server points" and "opcUaClient-OpcUaClientPointDeviceExt."

### May 24, 2022

- In the "opcUaServer-OpcUaServertopic" component topic added "Max Monitored Items Per Subscription" property description.
- Added new topic "Generating an OPC UA client certificate" to "OPC UA client tasks" Chapter
- Created new topic for setting up certificate based authentication.

### March 17, 2021

Edited the introductory chapters, and added screen captures and properties to component topics.

### October 27, 2020

In the "Connecting to an OPC UA Server" topic added the information about:

- How to set up the security settings and user name and password
- And in the "Third-party simulation software" updated OPC foundation's Sample Applications download.

### 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.

### December 19, 2017

Initial publication.

## Related documentation

This topic lists documents that are related to this guide.

- *Niagara Drivers Guide*
- *Niagara Platform Guide*

# Chapter 1. Overview

OPC is the interoperability standard for secure and reliable data exchange in the industrial automation space, as well as other industries. The OPC standard is a series of specifications that define the interface between clients and servers, as well as servers and servers, including access to real-time data, monitoring of alarms and events, access to historical data and other applications.

The OPC standard, initially restricted to the Windows operating system, derives its acronym from OLE (object linking and embedding) for Process Control. These specifications are now known as OPC Classic. The OPC Unified Architecture (UA) is a platform-independent, service-oriented architecture that integrates the functionality of the individual OPC Classic specifications into an extensible framework.

The Opc UA Driver is the Niagara implementation of this service-oriented architecture.

## Driver component model

Niagara provides a set of two OPC UA drivers: `opcUaClient` and `opcUaServer`. Both follow the Niagara Framework® model for drivers.

The OPC UA specification provides a platform-independent, service-oriented architecture. It integrates all functionality from the existing OPC Classic specification, providing a more secure and scalable solution. The specification is backwards compatible with OPC Classic.

### OpcUaServer driver component model

The two main components are the `OpcUaServer` and `OpcUaNamespace`. The `OpcUaServer` component models an OPC UA server instance in the driver framework. There should be only one instance of this driver in a station.

The `OpcUaServer` component models an Opc UA namespace in the parent OPC UA server. It typically provides a logical grouping of OPC UA variables, histories, and events that can be accessed by an OPC UA client.

### OpcUaClient driver component model

The primary `OpcUaClient` components are:

- `OpcUaNetwork` serves as the parent container for all of OPC UA client devices.
- `OpcUaDevice` represents an OPC UA client.
- `OpcUaClientPointDeviceExt` contains proxy control points used to proxy values to and from the OPC UA server.
- `OpcUaClientAlarmDeviceExt` subscribes the station to alarm events from the OPC UA server.
- `OpcUaClientHistoryDeviceExt` imports histories from the OPC UA server.
- `OpcUaClientProxyExt` is the control point proxy ext used to identify a specific OPC UA server data variable to proxy through this control point.
- `ImportHistoryExt` is the history point extension that imports the history data associated with the specific OPC UA server data variable defined in the proxy ext.

### Supported profiles

- UA Generic Client Profile
- UA Data Access Client Profile
- UA History Data Access Client Profile
- UA Alarm

### Supported facets

The following table lists the OPC UA client facets supported by the Niagara OPC UA drivers. This is a subset of

available OPC UA facets.

OPC UA Facet	Description
Base Client Behavior Facet	Defines client behavior for best use by operators and administrators. These behaviors include the ability to configure an endpoint for a server without using the discovery service set; support for manual security configuration and behavior with regard to security issues; support for automatic reconnection to a disconnected server.
Core Client Facet	Defines the core functionality required for any client. This Facet includes the core functions for security and session handling.
AddressSpace Lookup Client Facet	Defines the ability to navigate through the AddressSpace and includes the basic AddressSpace concepts: view and browse, and simple attribute read functionality.
Attribute Read Client Facet	This Facet defines the ability to read Attribute values of Nodes.
DataChange Subscriber Client Facet	This Facet defines the ability to monitor Attribute values for data change.
DataAccess Client Facet	This Facet defines the ability to utilize the DataAccess Information Model, i.e., industrial automation data like analog and discrete data items and their quality of service. Partially supported.
Alarm and Event Client Facet	This Facet defines the ability to subscribe for Event Notifications. This includes basic AddressSpace concept and the browsing of it, adding events and event filters as monitored items and adding subscriptions.
Method Client Facet	This Facet defines the ability to call arbitrary Methods.
Historical Access	This Facet defines the ability to read, process, and update historical data.

## Requirements

The OPC UA drivers require licensing and a set of core modules.

### Platforms and software

The OPC UA drivers function on any QNX-based platform running on latest Niagara version, such as the JACE-8000, as well as Windows-based platforms (Windows 7 Service Pack One and later) running on latest Niagara version.

The OPC UA drivers function on any Ubuntu Core 20 platform running on latest Niagara version, such as the JACE-9000, as well as Windows-based platforms (Windows 7 Service Pack One and later) running on latest Niagara version.

In addition to the framework software, the OPC UA server requires software, such as Prosys OPC UA Simulation Server, installed on your PC. You use this to configure and manage the OPC UA server.

### License requirements

The opcUaClient module requires a license for the "opcUaClient" feature. This license may have other OPC UA client device, proxy point, and history limits.

The opcUaServer module requires a license for the "opcUaServer" feature. This license may have other OPC UA client device, proxy point, and history limits.

To check to see if your software installation is licensed for opcUaClient and/or opcUaServer, open the license file from the **License Manager** view. The feature name is present only if your platform is licensed for the feature, as shown here:

```
<feature name="opcUaClient" expiration="2020-12-31" history.limit="" point.limit="" device.limit="" />
<feature name="opcUaServer" expiration="2020-12-31" history.limit="" point.limit="" device.limit="" />
```

### Server modules

The server driver requires these modules in the host station:



- `opcUaCore-rt` contains Java Class files and resources common to both OPC UA client and OPC UA server functionality.
- `opcUaServer-rt` contains Java Class files and resources that support OPC UA server run-time functionality.
- `opcUaServer-wb` contains Java Class files and resources that support OPC UA server Workbench functionality.

### Client modules

The client driver requires these modules in the host station:

- `opcUaCore-rt` contains Java Class files and resources common to both OPC UA client and OPC UA server functionality.
- `opcUaClient-rt` contains Java Class files and resources that support OPC UA client run-time functionality.
- `opcUaClient-wb` contains Java Class files and resources that support OPC UA client Workbench functionality.

## Third-party simulation software

Third-party OPC UA simulation servers and client devices are available for purposes of testing and evaluation. This section provides a few links.

If installing only the client driver, you will need to identify an OPC UA server for the client driver to communicate with. Following are a couple of simulators that are available.

- Prosys OPC UA Simulation Server download: <https://prosysopc.com/products/opc-ua-simulation-server/>
- OPC Foundation Sample Applications download: <https://opcfoundation.org/products/view/opc-ua-server-simulator>

If installing only the server driver, you will need to identify an OPC UA client device for the server driver to communicate with. Following are a couple of clients that are available.

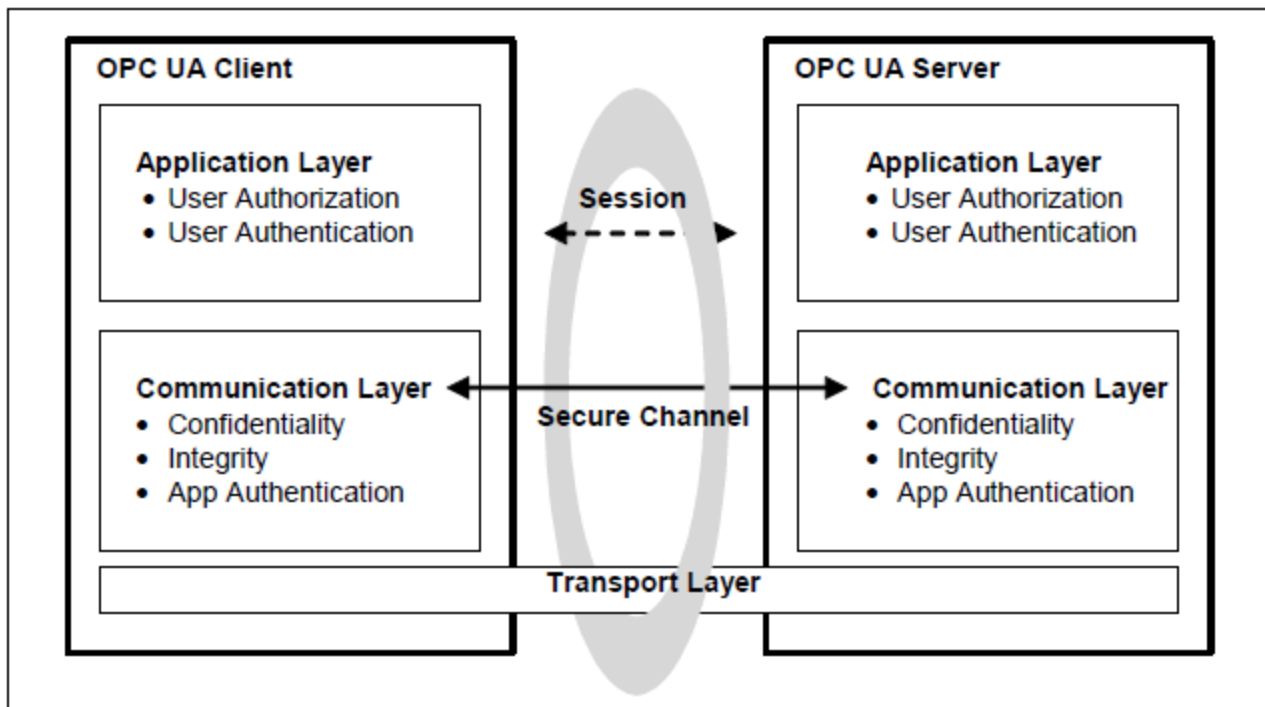
- Prosys OPC UA Client download: <https://www.prosysopc.com/products/opc-ua-client/evaluate/>
- Unified Automation's UA Expert OPC UA Client download: <https://www.unified-automation.com/downloads/opc-ua-clients.html>

### Related tasks

- [Verifying server setup using client simulation software](#)

## About the security architecture

The OPC UA security architecture is structured in an application layer and a communication layer on top of the transport layer.

**Figure 1.** OPC UA security architecture

A session in the application layer transmits information, settings and commands between a client application and a server application. The application layer also manages security objectives, user authentication and user authorization. The application layer communicates over a secure channel in the communication layer and relies upon it for secure communication. The secure channel provides encryption to maintain confidentiality, message signatures to maintain integrity and digital certificates to provide application authentication for data from the application layer, which it securely passes to the transport layer.

## Security profiles

In the Niagara OPC UA implementation the combination for security mode and security policy form a security profile. Different profiles specify different details, such as which encryption algorithms are required for which OPC UA functions. Some of the profiles specify security functions and others specify other functionality that is not related to security.

A Sign with Security Policy profile offers these options:

- `SignBasic128RSA15` provides medium channel security during commutation. It transports messages as plain text with a digital signature.
- `SignBasic256` provides medium to high channel security during commutation. It transports messages as plain text with a digital signature.
- `SignBasic256SHA256` provides high channel security during commutation. It transports messages as plain text with a digital signature.

A Sign and Encrypt with Security Policy profile offers these options:

- `SignEncryptBasic128RSA15` provides medium channel security during communication. It encrypts messages transports the data with a digital signature.
- `SignEncryptBasic256` provides medium to high channel security during commutation. It encrypts

messages and transports data with a digital signature.

- `SignEncryptBasic256SHA256` provides the highest channel security during commutation. It encrypts messages and transports data with a digital signature.

**NOTE:** The default profile for OPC UA is Sign and Encrypt with Security using the `SignEncryptBasic256SHA256` option for **Security Mode**.

The other profiles include a warning message that points out the security risk involved. You or an administrator must acknowledge the warning, which the driver logs in the system for audit purposes.



# Chapter 2. OPC UA server tasks

Setting up an OPC UA server involves configuring properties, adding server points with alarm and history extensions, and configuring OPC UA client users.

## OPC UA server, client user authorization

Starting with Niagara Niagara 4.14.u1 and Niagara 4.15, the OpcUaServer component provides access to specific aspects of the OPC UA Server based on user roles and permissions as they are configured in the User Service, Role Service and Category Service. These server-side settings are observed when granting access to specific aspects of the OPC UA Server and include more restrictive access for Anonymous users.

The Niagara OpcUaServer component performs user name and password authentication for users that are created and configured in the Niagara user service. In versions prior to Niagara 4.14.u1 and Niagara 4.15, this component does not consider the user roles and permissions as configured in the User Service, Role Service and Category Service when granting access to specific aspects of the OPC UA Server. Also, in the earlier versions, Anonymous users have the ability to subscribe to (monitor) point values.

Starting with Niagara 4.14.u1 and Niagara 4.15, based on individual user configuration, an OPC UA Client user experiences the following:

- the user sees only those points and point properties (including units, etc.) where read access is allowed.
- the user writes to only those points and point properties (including units, etc.) where write access is allowed.
- the correctly specified user access level types and restrictions are communicated to the Opc Ua client.

### Breaking change on version upgrade

OPC UA Server configurations prior to Niagara 4.14.u1 and Niagara 4.15, that rely on username and password authentication might start to see `User_Access_Denied` errors if the users are not assigned appropriate roles and permissions in the User Service. In addition, existing read subscriptions or monitors for Anonymous users will start to fail.

### Resolution

To resolve issues caused by the software update, configure any users that are created for OPC UA authentication with appropriate roles to access specific components of the driver tree.

## Setting up the OPC UA server

Setting up an OPC UA server follows a standard framework procedure.

### Prerequisites:

- The opcUaCore and opcUaServer modules are in the **My Modules** folder.
- The controller is licensed with the opcUaServer feature.
- You are connected to the Internet and working in Workbench running on a PC or laptop computer.
- The controller to use is on the network and powered on.

**NOTE:** If you intend to install only the OPC UA client driver, skip ahead to the chapter, "OPC UA client tasks."

The example client used in this procedure is the ProSys Simulation OPC UA Client. Many other third-party simulation OPC UA clients are available, which you can use to connect to an OPC UA server.

Step 1. Connect to the controller station.

Step 2. Open the opcUaServer palette (click the folder icon in the **Palette** sidebar).

- Step 3. Drag the **OpcUaServer** component from the palette to the **Config > Drivers** folder in the station. The driver adds the server to the station and enables it by default. The server **Status** likely reports {ok}.
- Step 4. To add a namespace (a container for point IDs), do one of the following:
- Drag an **OpcUaNamespace** component from the palette to the **OpcUaServer** node in the Nav tree.
  - Double-click **OpcUaServer** (this opens the **Opc Ua Server Device Manager**) and click **New > OK**.
- A second **New** window opens. It contains several properties.
- Step 5. Configure the **namespaceName** or use the default name and click **Save**. You can leave the rest of the properties at their default values. The driver adds the new namespace to the **Opc Ua Server Device Manager**. If you are updating the existing namespace, the changes will be reflected once the server restarts.
- Step 6. To retrieve the server's connection address, right-click the **OpcUaServer** node in the Nav tree and click **Views > AX Property Sheet**. The property sheet opens.

▶ Opc Tcp Endpoint	Opc Tcp Endpoint
▶ User Authentication Methods	<input checked="" type="checkbox"/> Anonymous <input checked="" type="checkbox"/> Username/Password
▶ Max Session Count	500
▶ Max Session Timeout	+00000h 05m 00s
▶ Max Subscription Count	50
▶ Opc Tcp Connection Address	opc.tcp://IE3BLT1GKS6C2.global.ds.honeyw
▶ Server Info	Opc Ua Build Info
▼ Session Info	No active sessions

- Step 7. Locate and make a note of the server's **Opc Tcp Connection Address**. Any OPC UA client attempting to connect to the server requires this connection address.
- NOTE:** The port specified in the **Opc Tcp Connection Address** may be blocked by your PC/ network firewall. The firewall settings may need to be adjusted to allow data transfer on this port.

## Result

The OPC UA server set up is complete. For any OPC UA client connection to succeed, the client must be configured with the server's **Opc Tcp Connection Address**, as well as a **Security Mode**, **Security Policy**, and **User Authentication** method as required by the server.

Username and password values also must be defined in the station's **UserService**.

## Related reference

- [opcUaServer-OpcUaServer](#)
- [opcUaServer-OpcUaNamespace](#)

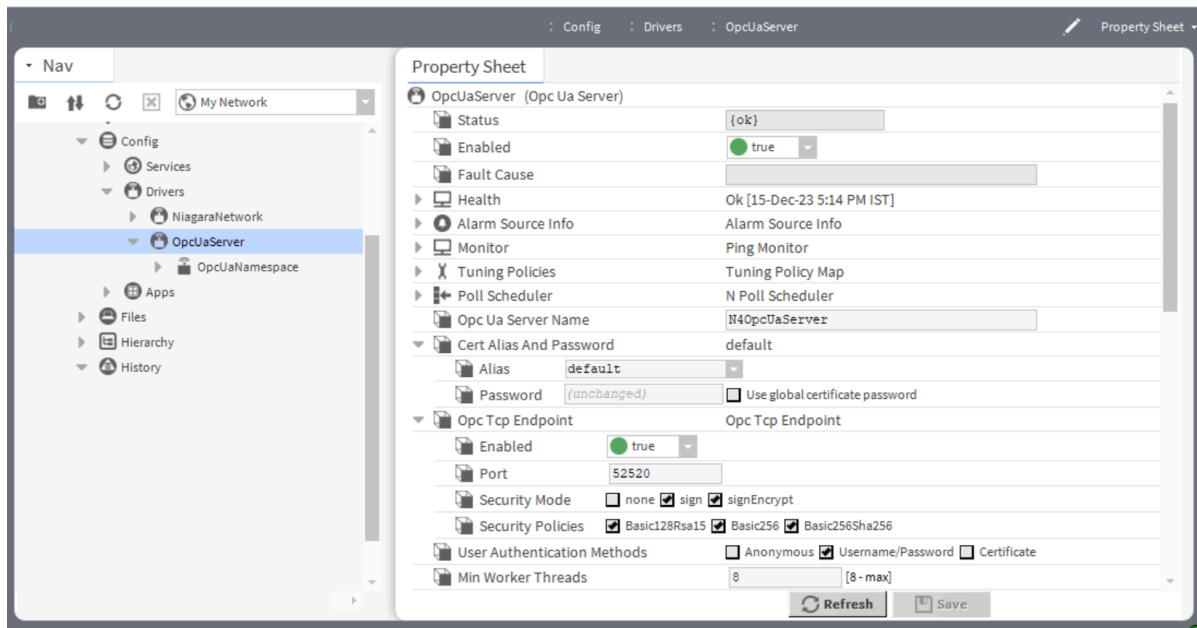
## Configuring OPC UA Server Security Modes

Starting with Niagara 4.14 to secure data, you need to configure the **Security Mode** and the Server Security Certificate used by the server to sign and encrypt communication messages. You must select the Server Security Certificate from the Niagara Key Store.

### Prerequisites:

- A self-signed certificate, or a CA signed certificate in the **User Key Store** is required for this procedure. For more details refer the “Generating an OPC UA Server Certificate” .
- Niagara 4.14 or later version is required to support certificate-based authorization.

Step 1. In the Workbench Nav tree, expand **Config > Drivers**, right-click the **OpcUaServer** and click **Views > Property Sheet**.  
The **Property Sheet** opens.



Step 2. Click the check box to select the **Security Mode** and **Security Policies**.

**NOTE:** It is recommended to use the **SignEncrypt** mode and **Basic256Sha256** policy for high security.

Step 3. In the **Cert Alias And Password** property, click the drop-down **Alias** to select the Server Security Certificate and enter the private key **Password**.

Step 4. To save your changes, click **Save**.

## Configuring the server to support an OPC UA client user

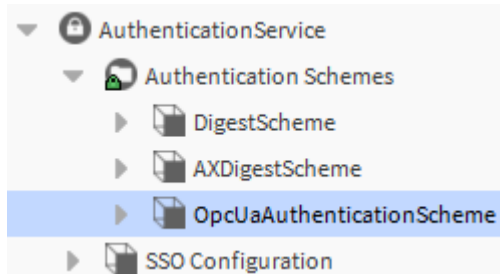
OPC UA clients require you to configure an authentication scheme. This scheme assures that the identity of each user can be verified. It may use roles to limit user access to only certain areas. The scheme authenticates the user when he or she enters a username and password, which are already configured for the user.

### Prerequisites:

- You are connected to a running station that is configured with the **OpcUaServer**.
- The **opcUaServer** palette is open.

**NOTE:** When making any server-side changes, you must first disable and then re-enable the server.

- Step 1. In the Nav tree, expand **Config > Drivers**, right-click **OpcUaServer** and click **Views > AX Property Sheet**.  
The server's AX Property Sheet opens.
- Step 2. Under the **Enabled** property, click the drop-down list and click **false**.  
The driver disables the **OpcUaServer**.
- Step 3. In the Nav tree, expand **Config > Services > AuthenticationService**.



- Step 4. Drag the **OpcUaAuthenticationScheme** component from the **opcUaServer** palette to the station's **Authentication Schemes** subfolder.
- Step 5. Expand **Config > Services** and double-click **UserService**.  
The **User Manager** opens.
- Step 6. To add a single new user, click **New** and click **OK**.  
A second **New** window opens.
- Step 7. Configure these properties and click **Ok**.:
  - For **Name** enter the user's name.
  - For **Authentication Scheme Name**, select a scheme from the drop-down list.
  - For **Password**, create a strong password and enter it a second time in the **Confirm** property.  
A strong password requires at least ten characters, plus at least one of each of the following characters: lowercase, uppercase, and a digit.
- Step 8. If you use the same default password for all new users set the **Force Reset At Next Login** value to **true**.
- Step 9. To enable the server, expand **Config > Drivers**, right-click **OpcUaServer** and click **Views > AX Property Sheet**.  
The **AX Property Sheet** opens.
- Step 10. Set **Enabled** to **true** and click **Save**.  
The server is ready to support an OPC UA client user.

#### Related reference

- [opcUaServer-OpcUaAuthenticationScheme](#)

## Setting up certificate based authentication

Certificate-based authentication is a technique that allows one machine to securely identify itself to another across a network connection, using a certificate called a public-key certificate.

#### Prerequisites:

- You need to have CA and Intermediate certificates, or a self-signed certificate to use in this procedure.



Refer to the “Generating an OPC UA Client Certificates” for instructions.

- Niagara 4.14 or later version is required to support the latest client certificate authentication in the OpcUa Server driver.

Using Signed Certificates

- Step 1. Import the CA and Intermediate certificates into the **User Trust Store** of the station's **CertManagerService** and the **User Trust Store** of Workbench.
- Step 2. Create a client certificate in the station's **CertManagerService** and sign the client certificate by any one of the CA or Intermediate certificates.
- Step 3. Export public key and private key of the client certificate.
- Step 4. For further communication with the server, send the public key and private key file to client. By default, the public key is in PEM format. The client needs to convert the public key into the required format by using openssl command and use that certificate as the user identity credential.

Using Trusted Self-Signed Certificates

- Step 1. Import the client certificate into the **User Trust Store** of the station's **CertManagerService**.
- Step 2. Create a connection from the client using that certificate as the user identity credential.

Using Self-Signed Certificates from an allowed host

- Step 1. Attempt a connection from the client using a certificate as the user identity credential.
- Step 2. **Approve** the certificate exemption from host in the **Allowed Hosts** section of the station's **CertManagerService**.
- Step 3. Ping the server from the device again to establish connection.

OpcUaServer alarm processing

When started, the OPC UA server automatically adds to the station's **AlarmService** **OpcUaServerAlarmClass** and **OpcUaServerAlarmRecipient** components (if they do not already exist).

The driver routes an OPC UA event from the Niagara alarms to the **OpcUaServerAlarmRecipient** via the linked **OpcUaServerAlarmClass**. Alarms routed to the recipient must come from a control point that also contains a server proxy extension. The **OpcUaServerProxyExt** relates the **OpcUaServerAlarmRecipient** to a **UA Node Id**. This **UA Node Id** is a property of the **OpcUaServerProxyExt**.

The **OpcUaServerAlarmRecipient** also has an **Opc Ua Severity** property. This property maps the alarm's **Alarm State** to the OPC UA alarm severity index.

The following image shows the default **Opc Ua Severity** mapping.

▼ ● Opc Ua Severity	700, 900, 500, 600
● To Offnormal	<input type="text" value="700"/> [1 - 1000]
● To Fault	<input type="text" value="900"/> [1 - 1000]
● To Normal	<input type="text" value="500"/> [1 - 1000]
● To Alert	<input type="text" value="600"/> [1 - 1000]

## Adding a server point with alarm and history extensions

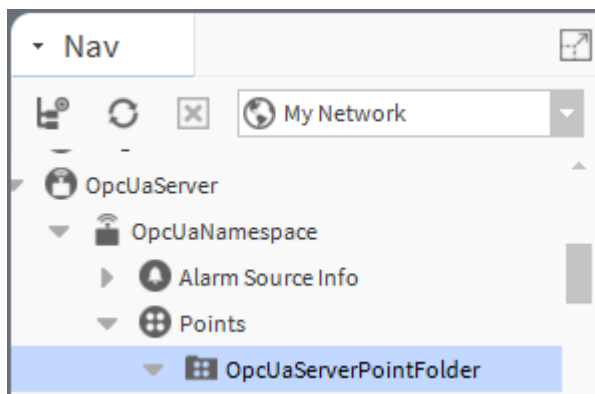
Server points provide live and historical data as well as alarm events. Server points are visible to connected OPC UA clients.

### Prerequisites:

- You are connected to a running station that is configured with the **OpcUaServer**.
- The **opcUaServer** palette is open.
- The station does not already have control points available.

**NOTE:** Adding a history extension to a control point in the OPC UA server's namespace makes the associated history visible to a connected OPC UA client. Similarly, adding an alarm extension to a control point in the OPC UA server's namespace and setting the alarm extension's **Alarm Class** to **OpcUaAlarmClass** sends OPC UA events to a connected OPC UA client that subscribes to these events.

Step 1. In the Nav tree, expand **Config > Drivers > OpcUaServer > OpcUaNamespace**.

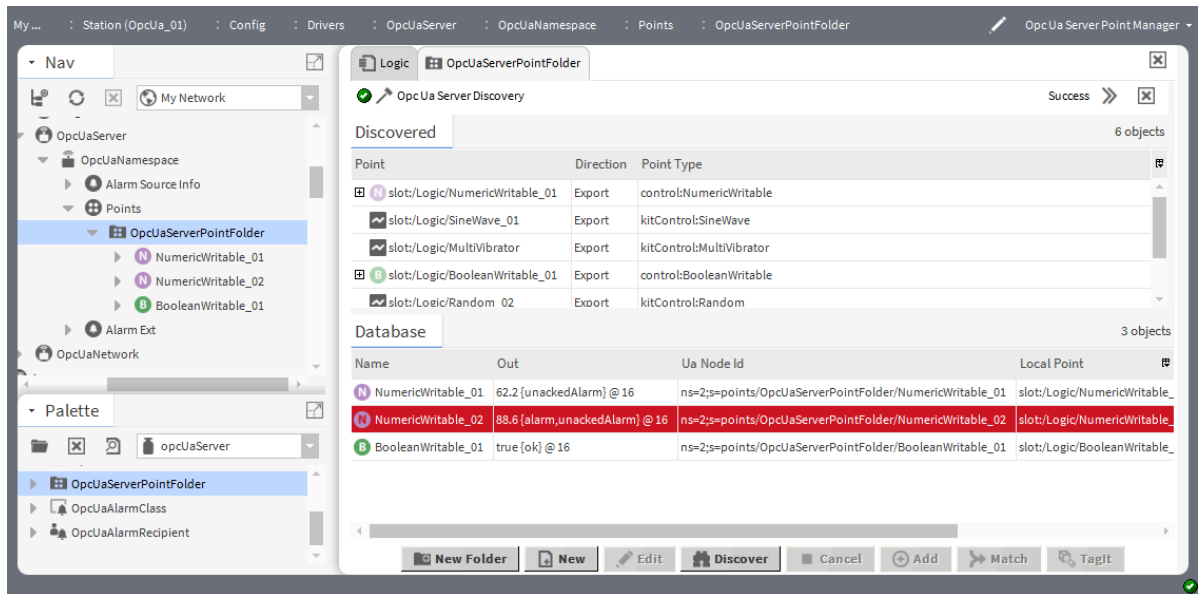


Step 2. From the palette, add an **OpcUaServerPointFolder** to the **Points** container.

Step 3. Under the station's **Config > Drivers** folder, add a new **NumericWritable** point. Configure it with a **NumericCov** history extension and an **OutOfRangeAlarmExt**.

Step 4. To open the **Opc Ua Server Point Manager**, double-click the **OpcUaServerPointFolder**, click **Discover**, then, in the **ORD** property, select the folder to search and click **OK**.

The **Discovered** pane presents a list of control points that exist in the station. You can map these to the **OpcUaServerPointFolder**.



Step 5. Select the **NumericWritable** in the **Discovered** pane and click **Add**.

The driver adds the point to the **Database** pane where it becomes visible under the **OpcUaServerPointFolder**.

By default, the driver exports a discovered writable point from the server as read-only. However, you can add a writable point that you can import. The driver imports the point (which the client writes) to the server. The server reads the data from the client.

After adding the point in the server, you can add history or alarm extensions to enable histories or alarms for the point.

Step 6. To do this, expand the writable point in the **Discovered** pane, select the point with a **Direction** value of **Import** and click **Add**.

## Result

The driver serves up the server points in this **OpcUaServerPointFolder** as consumable data including live and historical data and alarm events. These data are visible to a connected OPC UA client.

## Related reference

- [opcUaServer-OpcUaServerAlarmDeviceExt](#)

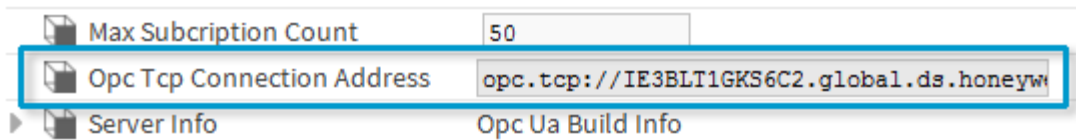
## Verifying server setup using client simulation software

This procedure uses the ProSys OPC UA Client software to verify the OPC UA server setup. Several other OPC UA clients are available. The procedure remains basically the same for all clients.

### Prerequisites:

- You are connected to a running station that is configured with the **OpcUaServer**.
- OPC UA client simulation software is installed.
- The station does not already have control points available.
- You have an Internet connection.
- Client user credentials (username and password) exist in the station's **UserService**.

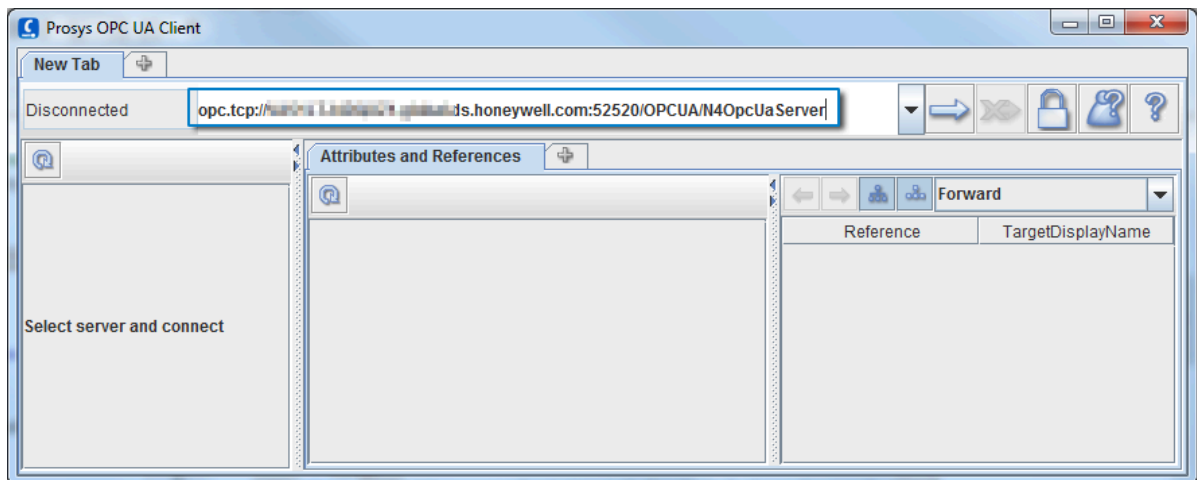
- Step 1. Expand **Config > Drivers**, right-click **OpcUaServer** and click **View > AX Property Sheet**.  
The **AX Property Sheet** opens.




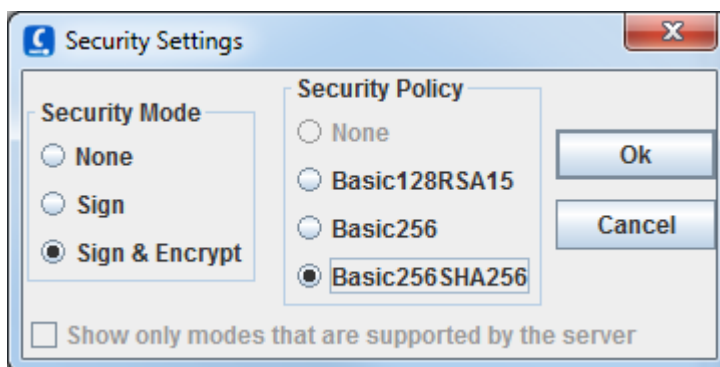
- Step 2. Copy the **Opc Tcp Connection Address**.

**NOTE:** The port specified in this address may be blocked by the PC/network firewall. The firewall settings may need to be adjusted to allow data transfer on this port.

- Step 3. Open the simulation software.  
The Prosys OPC UA Client window opens.

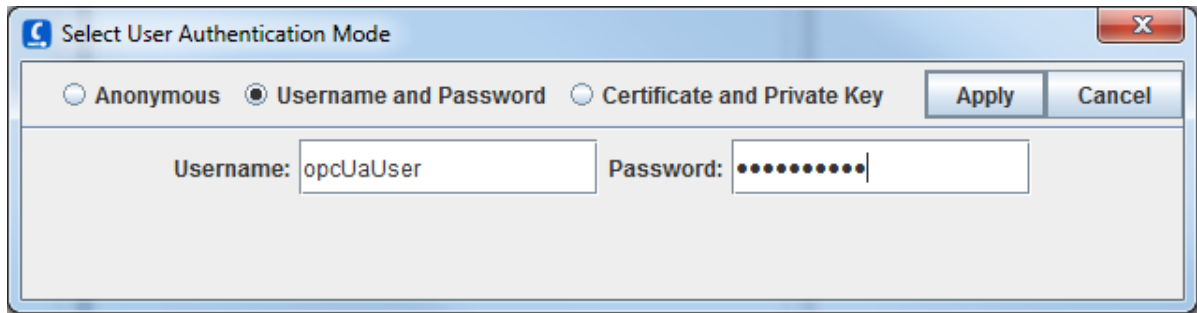



- Step 4. Paste the **Opc Tcp Connection Address** into the available field and click the security options button (  ).  
The **Security Settings** window opens.

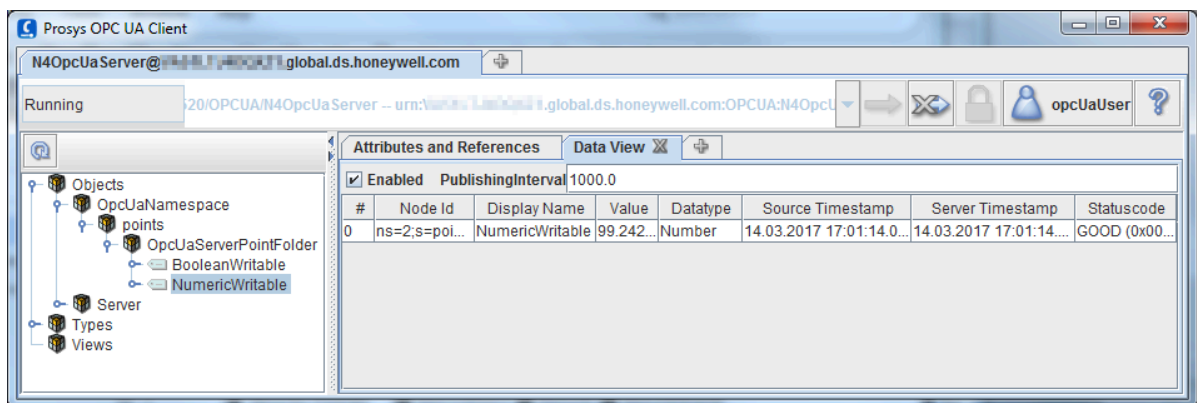


- Step 5. Select the **Security Mode** and **Security Policy** settings required by the server, for example **Sign & Encrypt** and **Basic256SHA256**, and click **Ok**.  
The focus returns to the **Prosys OPC UA Client** window.

- Step 6. To define the user, click a user ID button (  ).  
The **Select User Authentication Mode** window opens.



- Step 7. Select a user authentication mode as required by the server, for example: Username and Password, enter a valid **Username** and **Password** and click **Apply**.  
The focus returns to the **Prosys OPC UA Client** window.
- Step 8. To connect to the server, click the connect button (  ).  
The client connects to the OPC UA server.



In the example, expanded **Objects** reveal the OPC UA namespace with server points. The right-side **Data View** tab monitors data values for the selected points.

## Related reference

- [Third-party simulation software](#)

## Generating an OPC UA server certificate

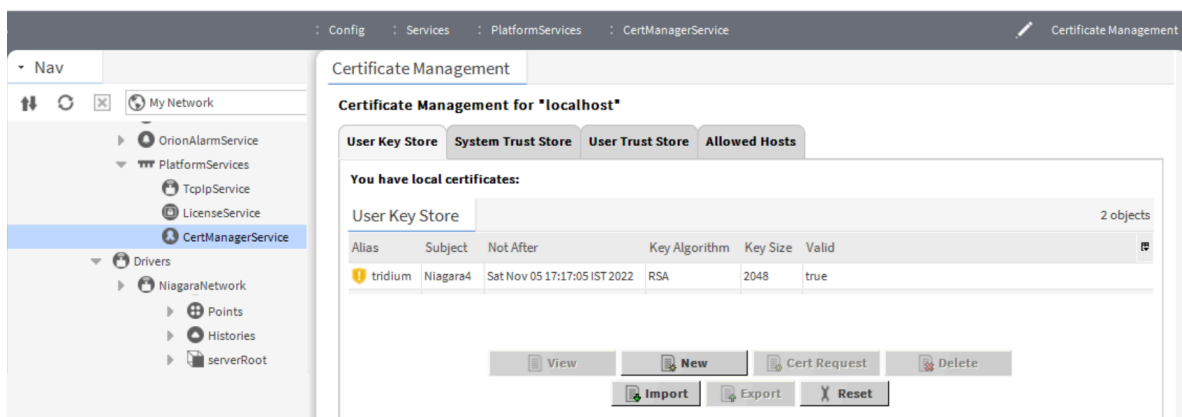
The topic explains how to generate a server certificate using the added URI field in the certificate to establish a secure server connection from the client to the server.

### Prerequisites:

- The certificate is used for OPC UA users only.
- Workbench running on a PC or laptop computer.

- Step 1. To generate a certificate, expand **Station > Config > Services > PlatformServices** and double-click **CertManagerService**

The **Certificate Management** view opens.



Step 2. Click the **New** button at bottom of the view.

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

Generate Self Signed Certificate

**Generate Self Signed Certificate**  
Generates a self signed certificate and inserts it into the keystore

Alias: OPCUA (required)

Common Name (CN): OPCUA (required)  
\* this may contain the host name or address of the server

Organizational Unit (OU):

Organization (O): Acme (required)

Locality (L):

State/Province (ST):

Country Code (C): IN (required)

Not Before: 09-Jan-2024 04:43 PM IST

Not After: 08-Jan-2025 04:43 PM IST

Key Size: ☒ 1024 bits ☒ 2048 bits ☐ 3072 bits ☐ 4096 bits

Certificate Usage: ☒ Server ☐ Client ☐ CA ☐ Code Signing

Alternate Server Name:

Alternate Server URI: 3.global.acmecorp.com:OPCUA:N4OpcUaServer

Email Address:

Key Usage: ☒ Digital signature ☐ Non-repudiation ☒ Key encipherment  
☒ Data encipherment ☐ Key agreement ☐ Certificate signing  
☐ CRL signing ☐ Encipher only ☐ Decipher only

OK Cancel

- Step 3. Give the certificate at least an **Alias**, **Common Name (CN)**, **Organization**, **Locality**, **State/Province**, and **Country Code**.

- Use **Alias** to identify this as an OPCUA certificate.
- The **Common Name (CN)** becomes the **Subject** (also known as the Distinguished Name). For OPCUA certificate, the **Common Name (CN)** may be the same as the **Alias**.
- **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** 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](http://countrycode.org)).
- **Not Before** and **Not After** define the period of validity for the certificate.
- For **Certificate Usage**, the radio button should be set to **Server**.

Step 4. Enter the URI in the **Alternate Server URI** field; it should be in the format: `urn:<full.computer.name>:OPCUA:N4OpcUaServer` and the select checkbox in the **Key Usage** set to **Data encryption** in the certificate.

**NOTE:**

- The full computer name can be found in **Control panel > System & Security > System**; it is of the type `hostname.domain` or just `hostname` in some cases.
- While connecting from client to server, the URI provided in the client certificate should match in the Application URI for the server. If doesn't match the URI, it sends an error message as `Bad_CertificateUriInvalid 0x80170000`. The URI specified in the Application Description does not match the URI in the certificate.

Step 5. When you have filled in the required fields, click **OK**.  
The **Private Key Password** window opens.

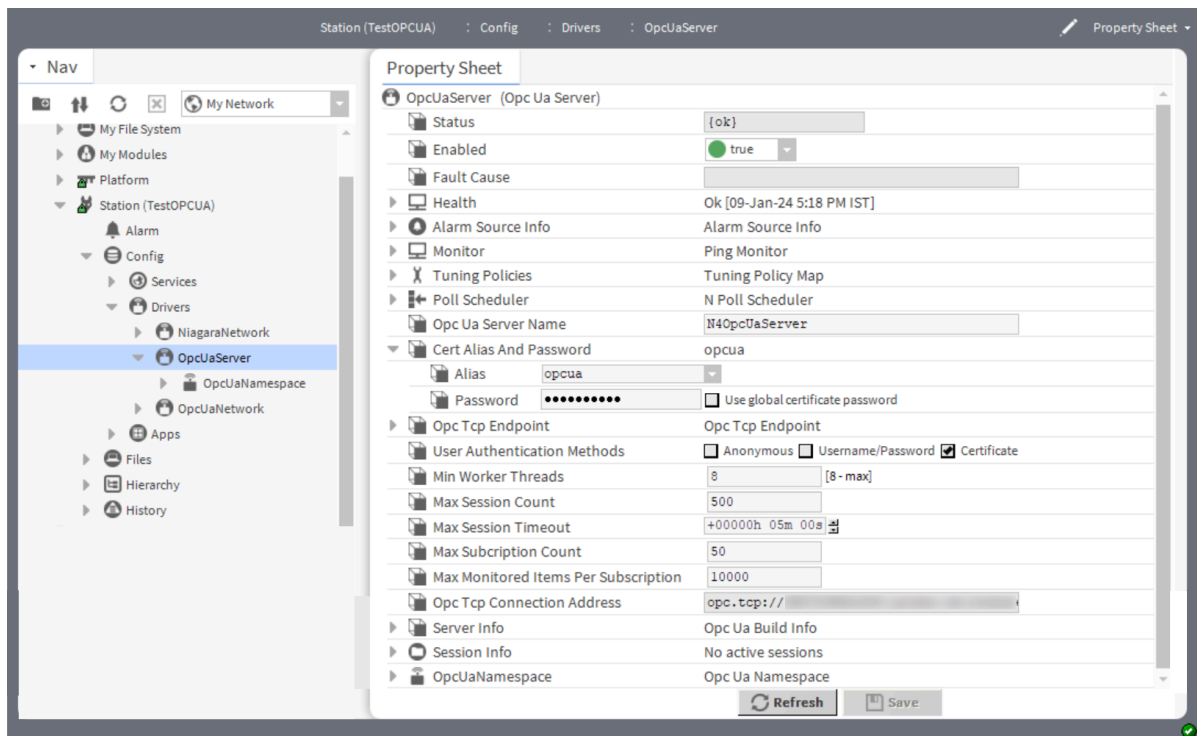


Step 6. Enter and confirm the password, then click **OK**.  
The system submits the certificate for processing in the background. A pop-up window appears on your screen advising you regarding it may take time to generate the certificate. The length of time it takes depends on the key size and the platform's processing capability. When created, the certificate appears as a row in the User Key Store table.

Step 7. To configure the certificate, open the device's **Property Sheet** by right-clicking on **OpcUaServer** followed by clicking **View > Property Sheet**.



The Property Sheet view opens.



- Step 8. Expand **Cert Alias And Password** property, select the certificate from the **Alias** drop-down menu, enter the private key **Password** and click **Save**.

## Generate an OPC UA server certificate for Third-Party Clients

The topic describes how to generate a server certificate using scripts by running the appropriate commands in the command prompt or Git Bash to establish a secure server connection from the client to the server. To ensure compatibility with OpcUaClient implementation, automatically uses a certificate for signing purposes with the keyCertSign usage while generating the certificate.

### Prerequisites:

- OpenSSL is installed on your system so that you can use the script file from the default windows command line.
- Niagara station is running.
- The hostname is the full device name. To find the full device name in the Windows menu, choose **Start > Settings > System > About**, and in **Device Specifications** you can find **Full device name** or from a command line, type the following net config workstation, and you can find the string **Full Computer name**.

Step 1. To generate a certificate, follow one of the choices below.

- If you are using windows, open the command prompt, type the following command and press Enter.

```
gen-opc-server-cert.bat
```

- If you are using Git Bash application, open the Git Bash prompt, type the following command and press Enter.

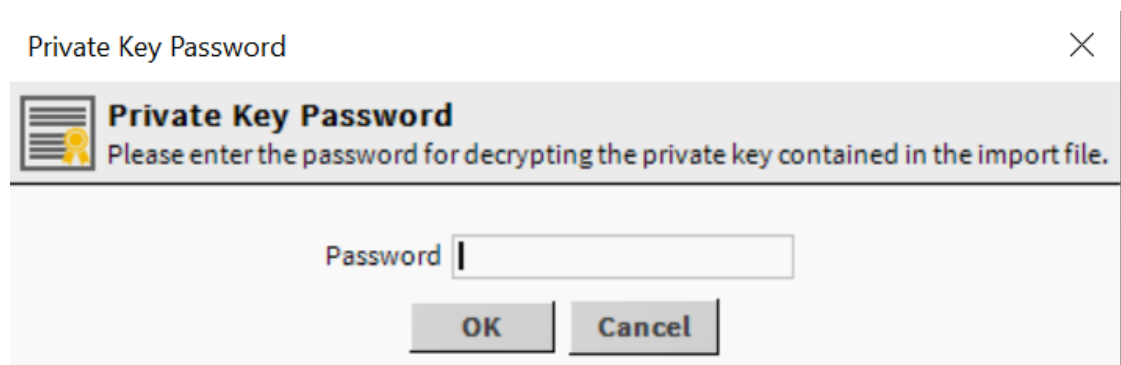
```
./gen-opc-server-cert.bat
```

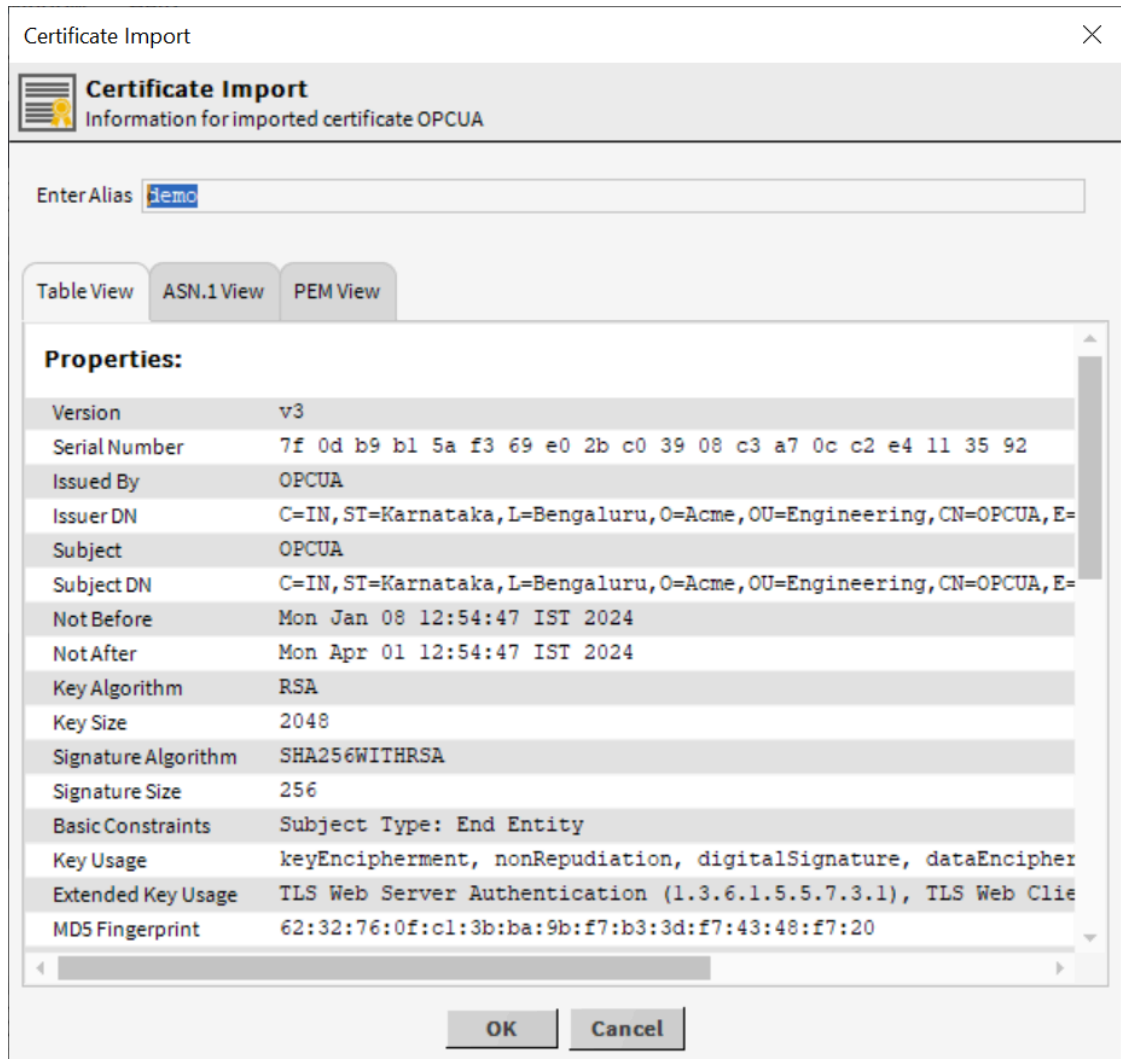


- The two-character **Country Code** is required and must be a known value, such as: US, IN, CA, FR, DE, ES, etc. (refer to the ISO CODE column at [countrycode.org](http://countrycode.org)).
- **State/Province**
- **Locality Name**
- **Organization Name** is the name of the company.
- **Organizational Unit Name**
- **Common Name (CN)**
- **Email Address**

It displays **Cert** written to the destination and generates a certificate in the given destination file.

- Step 7. To import the PEM certificate, open Workbench, expand **Config > Services > PlatformServices**, and click **CertManagerService**.
- In the **Certificate Management** view, click **Import**, browse to the destination file and enter the password for decrypting the private key.





**Certificate Import**  
Information for imported certificate OPCUA

Enter Alias

Table View ASN.1 View PEM View

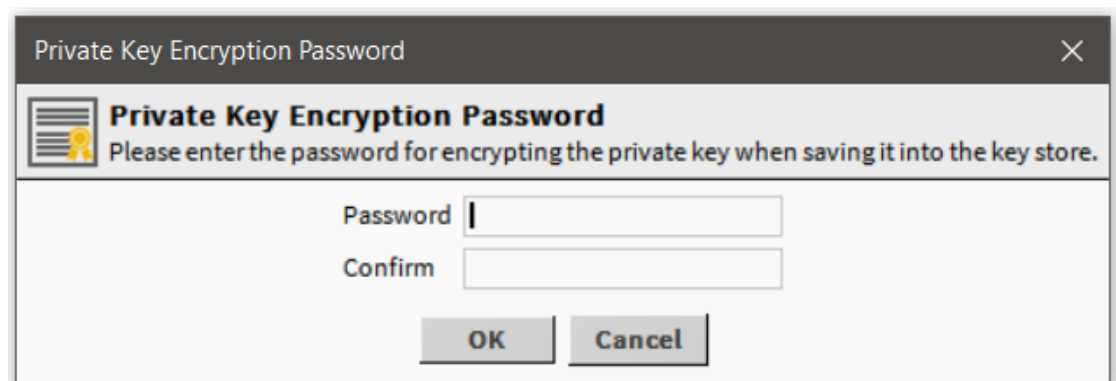
**Properties:**

Version	v3
Serial Number	7f 0d b9 b1 5a f3 69 e0 2b c0 39 08 c3 a7 0c c2 e4 11 35 92
Issued By	OPCUA
Issuer DN	C=IN, ST=Karnataka, L=Bengaluru, O=Acme, OU=Engineering, CN=OPCUA, E=
Subject	OPCUA
Subject DN	C=IN, ST=Karnataka, L=Bengaluru, O=Acme, OU=Engineering, CN=OPCUA, E=
Not Before	Mon Jan 08 12:54:47 IST 2024
Not After	Mon Apr 01 12:54:47 IST 2024
Key Algorithm	RSA
Key Size	2048
Signature Algorithm	SHA256WITHRSA
Signature Size	256
Basic Constraints	Subject Type: End Entity
Key Usage	keyEncipherment, nonRepudiation, digitalSignature, dataEncipher
Extended Key Usage	TLS Web Server Authentication (1.3.6.1.5.5.7.3.1), TLS Web Clie
MD5 Fingerprint	62:32:76:0f:c1:3b:ba:9b:f7:b3:3d:f7:43:48:f7:20

OK Cancel

The **Certificate Import** wizard opens.

- b. To change the existing Alias, enter the new Alias name or continue with the existing Alias and click OK.



**Private Key Encryption Password**

Please enter the password for encrypting the private key when saving it into the key store.

Password

Confirm

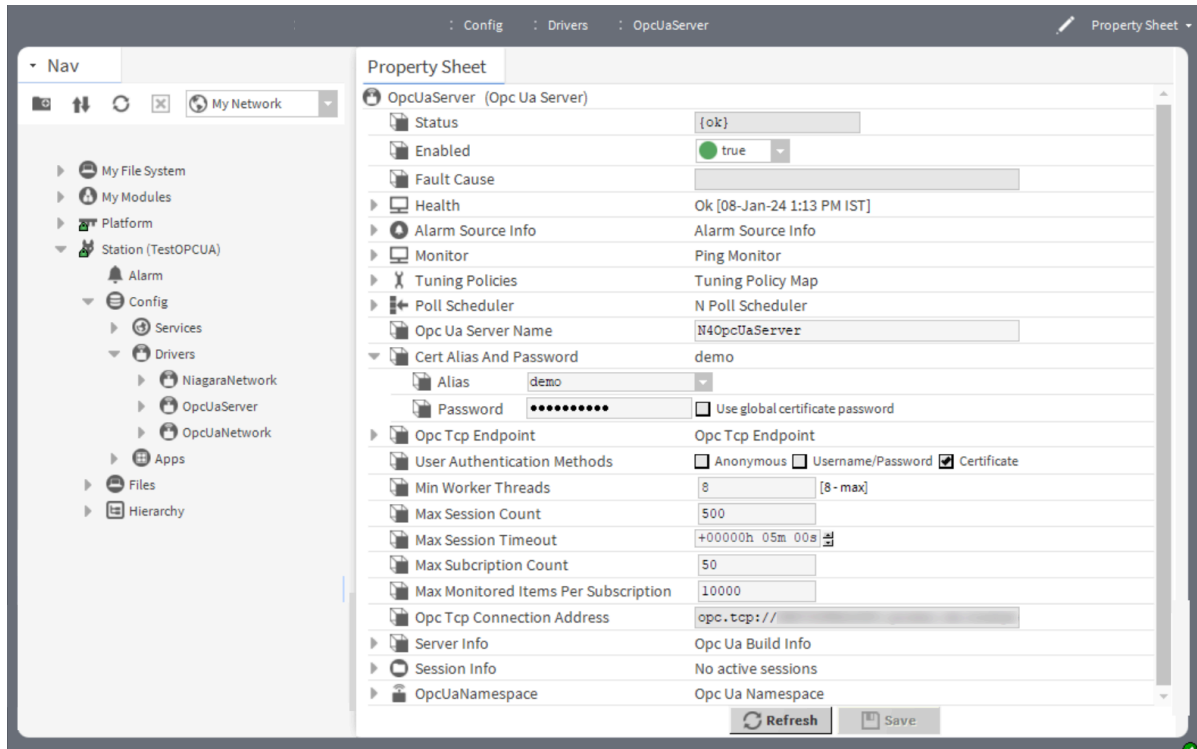
OK Cancel

The **Private Key Encryption Password** window opens.

- c. Type the **Password**, verify the **Confirm** password for encrypting the private key when saving it into the key store and click **OK**.

The certificate appears as a row in the **User Key Store** table.

- Step 8. To configure the certificate, open the device's **Property Sheet** by right-clicking on **OpcUaServer** followed by clicking **View > Property Sheet**.



The **Property Sheet** view opens.

- Step 9. Expand the **Cert Alias And Password** property, select a certificate from the **Alias** drop-down menu, enter the private key **Password** and click **Save**.  
The certificate is now available for Third-Party Clients.



# Chapter 3. OPC UA client tasks

The following topics describe how to set up the OPC UA client, connect to an OPC UA server, discover and add points with alarm and history extensions and subscribe for OPC UA alarm events.

## Adding the OPC UA network

An OPC UA network manages OPC UA devices, points, alarms and histories.

### Prerequisites:

- The `opcUaCore` and `opcUaClient` modules are in the **My Modules** folder.
- The controller is licensed with the `opcUaClient` feature.
- You are connected to the Internet and working in Workbench running on a PC or laptop computer.

Step 1. Connect to the controller station.

Step 2. Open the `opcUaClient` palette (click the folder icon in the **Palette** sidebar).

Step 3. In the Workbench Nav tree, expand **Config** and double-click on the **Drivers** folder.  
The **Driver Manager** opens.

Step 4. To add the network, click **New**.  
The **New** window opens.

Step 5. Select **OpcUaNetwork** from the drop-down list and click **OK**.

### Result

The driver adds the **OpcUaNetwork** to the station's **Driver Manager**. The driver enables the network by default. If this is an initial network setup, network **Status** will likely be `{down}` until you add an OPC UA device and configure the server connection.

### Related reference

- [opcUaClient-OpcUaNetwork](#)

## Connecting to an OPC UA server

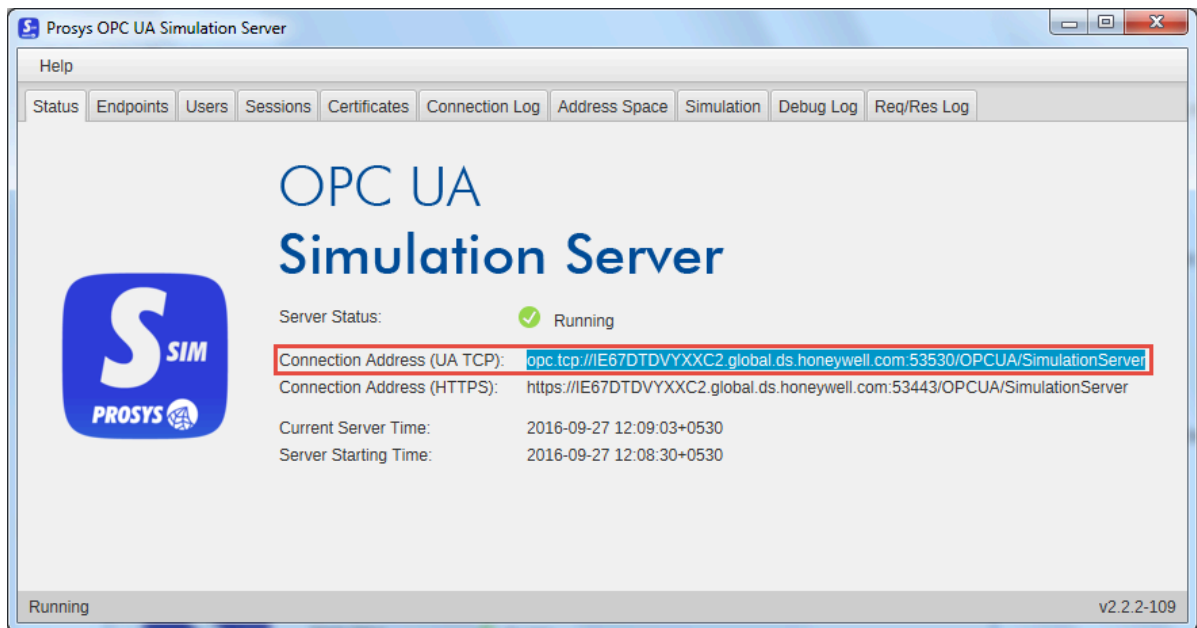
The next step is to connect to the OPC UA server and add an `OpcUaDevice` to the network.

### Prerequisites:

- You are working in Workbench running on a PC or laptop computer.
- Your OPC UA device is on the network and ready to connect.
- Prosys OPC UA Simulation Server is installed on the PC.
- Client Security Certificate (signed or self-signed) is generated or imported into the client station's **CertManagerService**. (For more details refer to the topics "Generating an OPC UA Client Certificate" and "Generating an OPC UA Client Certificate for Third-Party Server").
- Client User Identity Certificate (signed or self-signed) is generated or imported into the client station's **CertManagerService**. (For more details refer the topics "Generating an OPC UA Client Certificate" and "Generating an OPC UA Client Certificate for Third-Party Server").

Step 1. Open the OPC UA server software and navigate to the **Status** tab.

The software opens.



The example server software above is the ProSys Simulation Server. However, it is more likely that you will open a connection to your OPC UA server using software known to you that serves-up actual historical and live data.

- Step 2. Locate the **Connection Address** and copy it along with the required security mode and user authentication method.

In the example above, this address

is:opc.tcp://IE67DTDVYXXC2.global.ds.honeywell.com:53530/OPCUA/  
SimulationServer

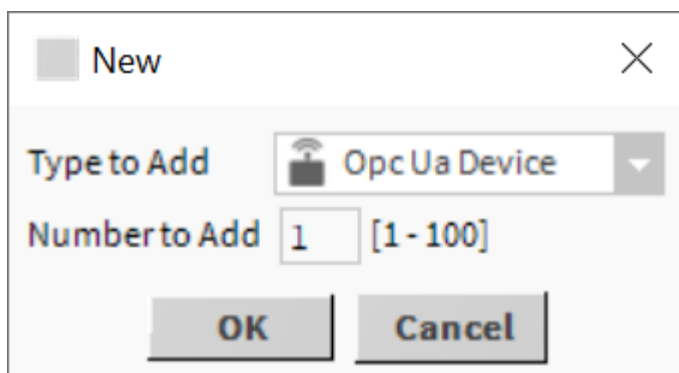
Within Niagara the default configuration for the OPC UA server and client connections is security mode `Sign` and `SignEncrypt` and security policy `Basic256SHA256`. These are the recommended settings for high security. A warning accompanies the other security policies and modes that the driver supports. You or an administrator must acknowledge this alert to proceed. The driver logs the acknowledgment in the system for audit purposes.

**NOTE:** If the device fails to find the server at the requested address, it may be because the client does not recognize the hostname. Consider adding the Hostname (IP address) to the hosts file. You may also use the IP address instead of Hostname in the connection address, for example: `opc.tcp://127.0.0.1:53530/OPCUA/SimulationServer`.

- Step 3. In the Workbench Nav tree, expand **Config > Drivers** and double-click **OpcUaNetwork**. The **Opc Ua Client Device Manager** opens.

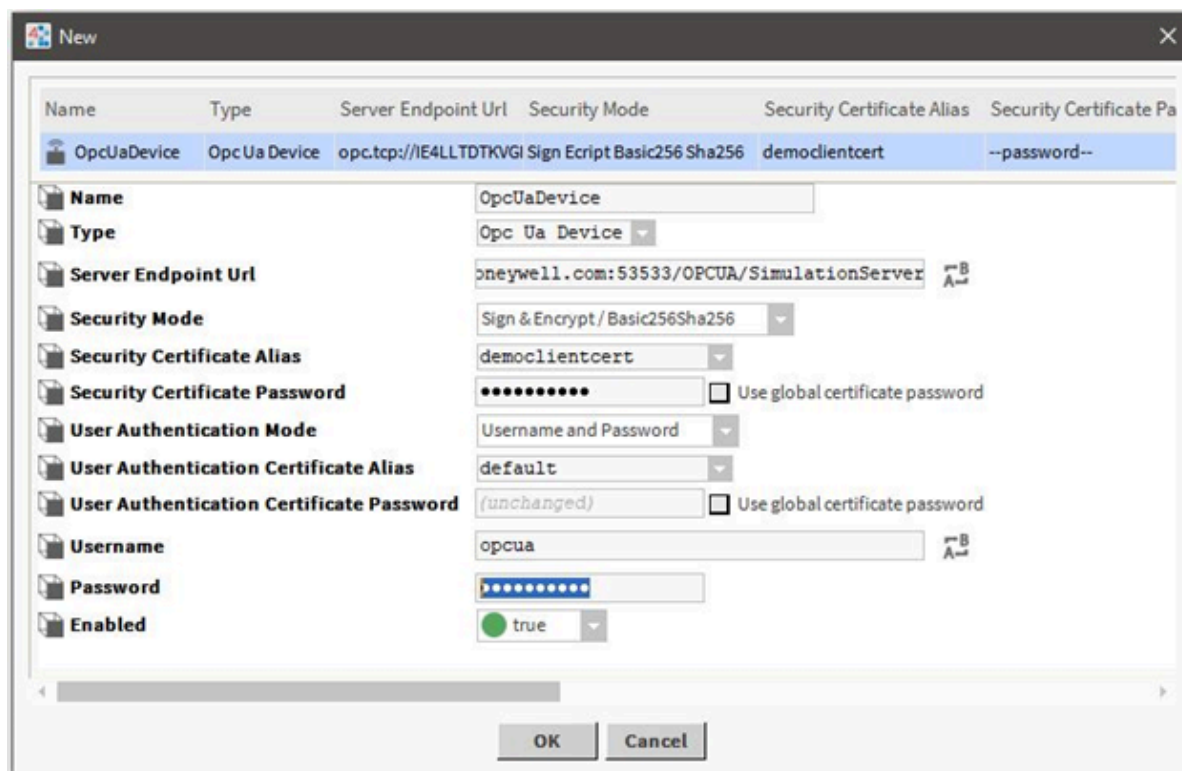
- Step 4. To add a new device, click **New**.





The New window opens.

Step 5. Select OpcUaDevice from the drop-down list and click OK.

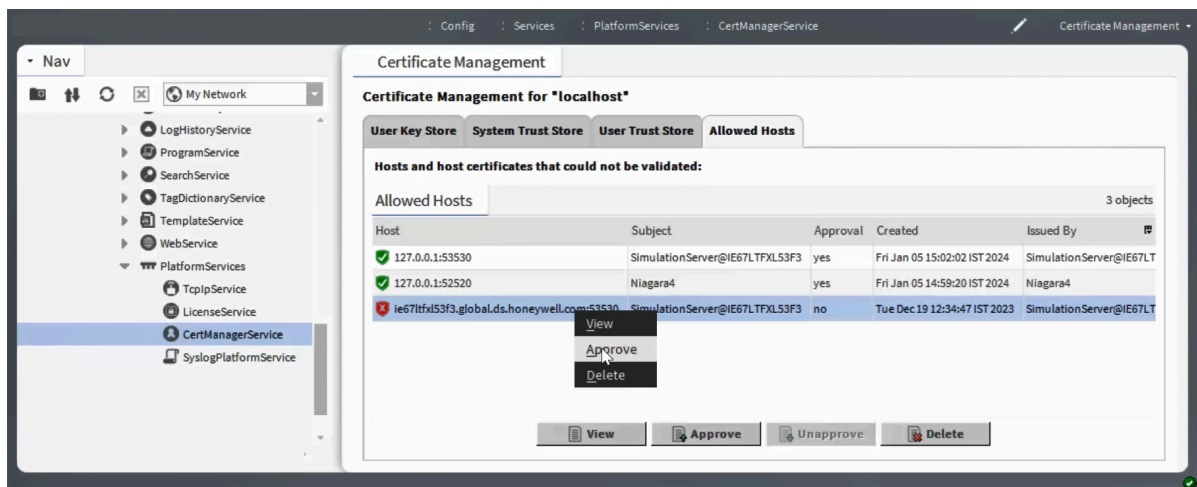


A second New window opens.

Step 6. Configure the following required properties and click OK.

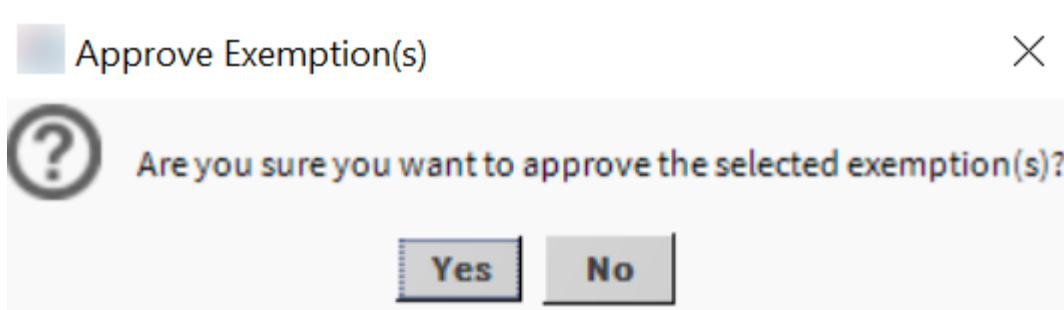
- **Server Endpoint Url** is the **Connection Address** you copied earlier.
- **Security Mode**, by default, is set to **Sign Encrypt Basic256 Sha256**. This value must match the server's **Security Mode** configuration. The default **Security Mode** for both OPC UA server and device is **Sign Encrypt Basic256 Sha256**, which enables signing and encryption with security policy **EncryptBasic256SHA256**.
- **Security Certificate Alias** defaults to the self-signed **tridium** certificate. For higher security, use a signed client certificate that matches the root CA certificate in the station's Trust Store.
- If **Security Mode** is selected other than **None**, then security certificates must be selected from the station's Key Store using the **Security Certificate Alias** drop-down list. You must enter the private key password for the selected certificate in the **Security Certificate Password** field.
- If **Security Mode** is selected as **None**, Security Certificate credentials remains same cannot be modified.
- By default the **User Authentication Mode** is set to **Username and Password**. This value must match the server's supported authentication modes.
- If **Username and Password** is selected, enter the username in the **User Authentication Certificate Alias**, password in the **User Authentication Certificate Password** and the credentials remains same.
- If **Certificate** is selected, click the **User Authentication Certificate Alias** drop-down to select the certificate and enter the private key password in the **User Authentication Certificate Password** field.

Step 7. Expand **Config > Services > PlatformServices** and double-click **CertManagerService**.



The **Certificate Management** view opens.

Step 8. To approve the Security Certificate sent by the server as an exemption, click the **Allowed Hosts** tab, right-click the host and click **Approve**.

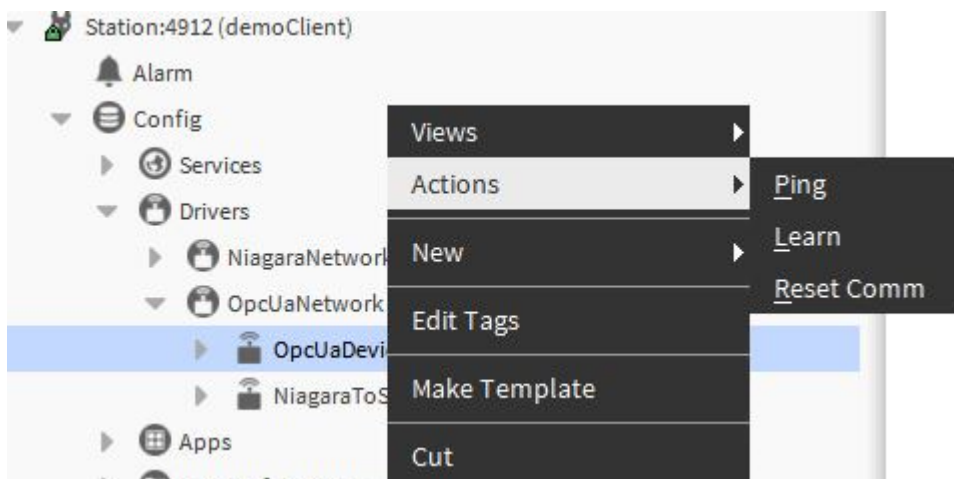


The **Approve Exemption(s)** window opens.

You do not need to approve the exemption if the Server's Security Certificate or the Signing Certificate has been imported into the **User Trust Store**.

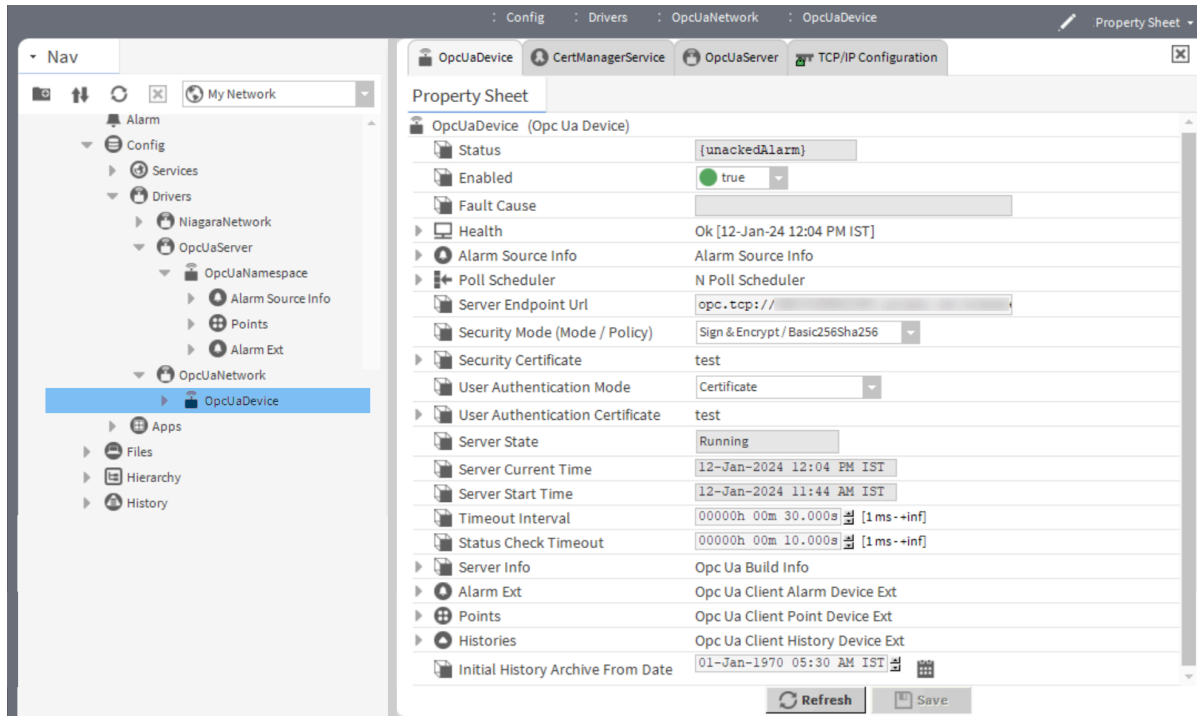
- a. Select **Yes** to continue.

Step 9. To ping the server, right-click the **OpcUaDevice** and click **Actions > Ping**.



**NOTE:** If the **OpcUaDevice** is still unable to create a secure channel to the server, the server might be rejecting the client's certificate. You may need to approve or trust the client certificate in the server's certificate store. Once you select the certificate, ping the server again.

Step10. Go to **Property Sheet** view of **OpcUaDevice**, and check the status of the following properties.



The system should have populated these properties with current values.

- **Server State** shows Running.
- **Server Current Time** shows current timestamp. For example, 12-Jan-2024 12:04 PM IST.
- **Server Start Time** shows server start time, For example, 12-Jan-2024 11:44 AM IST
- **Server Info** shows complete information of the server.
  - a. **Product Name**
  - b. **Product Uri**
  - c. **Manufacturer**
  - d. **Software Version**
  - e. **Build Number**
  - f. **Build Date**

#### Related reference

- [opcUaClient-OpcUaDevice](#)

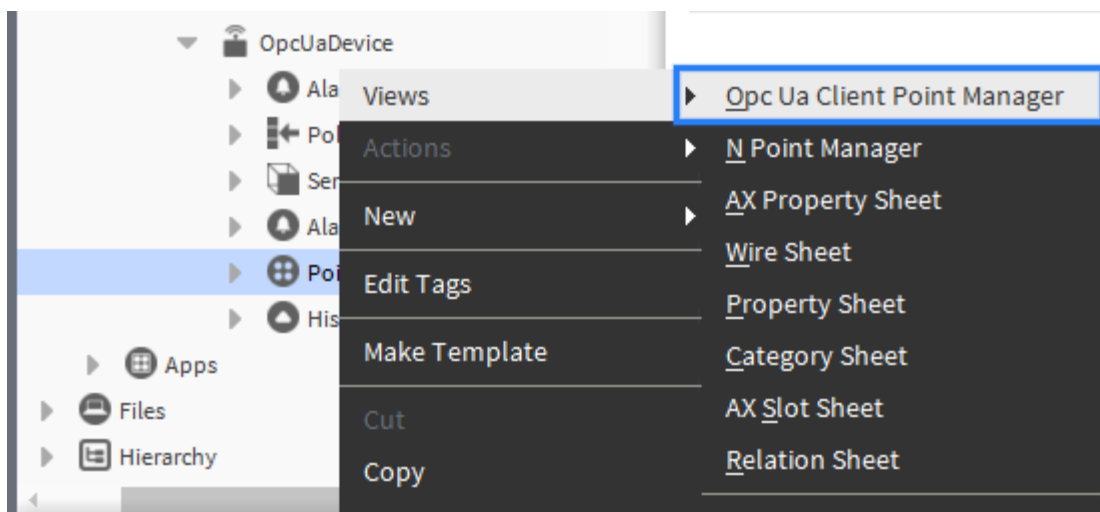
## Discovering OPC UA server points

Server point discovery interrogates the connected OPC UA server to discover the OPC UA objects.

#### Prerequisites:

You are working in Workbench running on a PC or laptop computer.

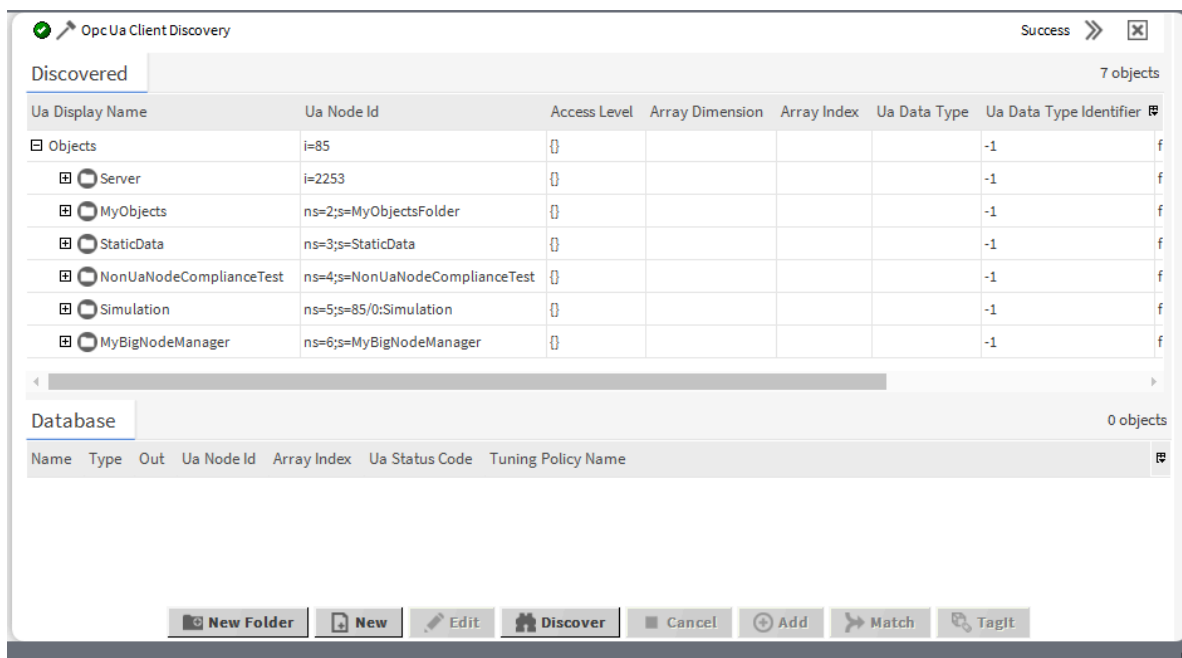
Step 1. In the Nav Tree, expand **Config > Drivers > OpcUaNetwork > OpcUaDevice**.



Step 2. To open the **Opc Ua Client Point Manager**, do one of the following:

- Right-click the **Points** node and click **Views > Opc Ua Client Point Manager** and click **Opc Ua Client Point Manager**.
- Double-click the **Points** node.

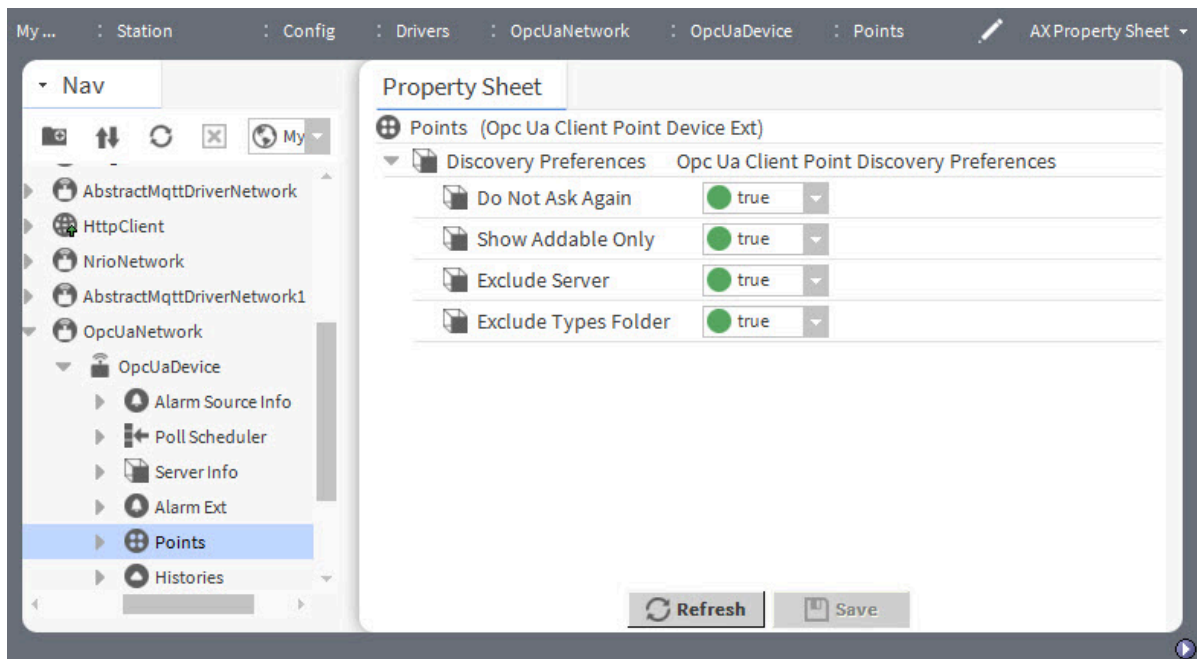
The **Opc Ua Client Point Manager** opens.



Step 3. To configure **Discovery Preferences** before discovering points, right-click **Points**, click **Views > AX Property Sheet** and expand **Discovery Preferences**.

**NOTE:** Be sure to configure **Discovery Preferences** properties before attempting to discover points. It is possible to unintentionally filter out points or data. Refer to the “opcUaClient-OpcUaClientPointDeviceExt” topic for details on these properties.

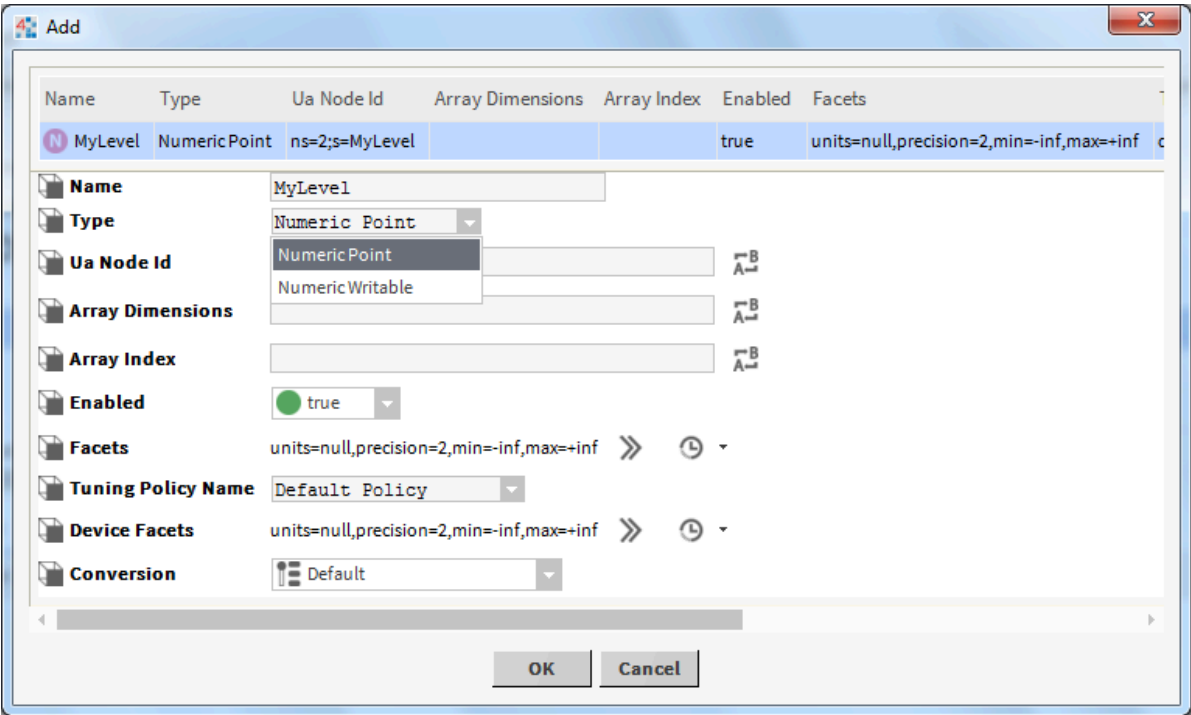
The **Property Sheet** opens.



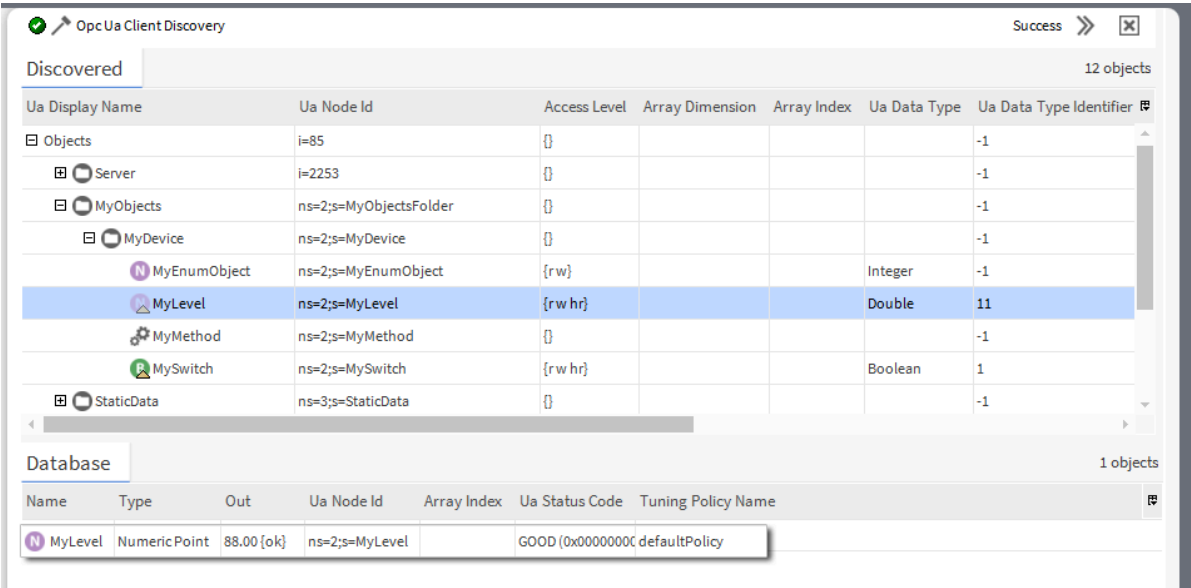
These properties control when the discovery prompt appears and which points a discovery job leaves out of the resulting **Discovered** table.

- Step 4. Configure preferences and click **Save**.
- Step 5. To start the discovery job, double-click **Points** in the Nav tree and click **Discover**.  
The discovery job communicates with the OPC UA server to discover the server's object structure.
- Step 6. To browse the discovered objects, expand the items in the **Discovered** pane.
- Step 7. To add points to the station database, select the points to add and click **Add** or double-click on the selected object(s).

The Add window opens.



Step 8. If the point is writable, change the point **Type** to a writable type as shown above and click OK. The driver adds the point(s) to a subscribed state and updates them with real-time values.



If a point has engineering units, precision, and/or range information, the driver attempts to configure the point's facets to match. This may not always be possible.

**Related reference**

- [opcUaClient-OpcUaDevice](#)


## Adding points containing OPC UA histories




If the OPC UA point you are adding has an OPC UA history, the point automatically includes a history extension (`ImportHistoryExt`).

**Prerequisites:**

Discovered objects with histories are visible in the **Opc Ua Client Point Manager** view.

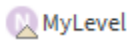
- Step 1.
- To open the **Opc Ua Client Point Manager**, expand **Config > Drivers > OpcUaNetwork > OpcUaDevice** and double-click the **Points** folder.
- Step 2.
- To run a discovery job, click **Discover**.
- Step 3.
- To confirm that the points have histories, examine them in the **Opc Ua Client Point Manager**.

 Opc Ua Client Discovery

Discovered		
Ua Display Name	Ua Node Id	Access L
[-] Objects	i=85	{}
[-] Server	i=2253	{}
[-] MyObjects	ns=2;s=MyObjectsFolder	{}
[-] MyDevice	ns=2;s=MyDevice	{}
MyEnumObject	ns=2;s=MyEnumObject	{rw}
 MyLevel	ns=2;s=MyLevel	{rw hr}
 MyMethod	ns=2;s=MyMethod	{}
 MySwitch	ns=2;s=MySwitch	{rw hr}

Here are two ways to identify OPC UA objects that have histories:

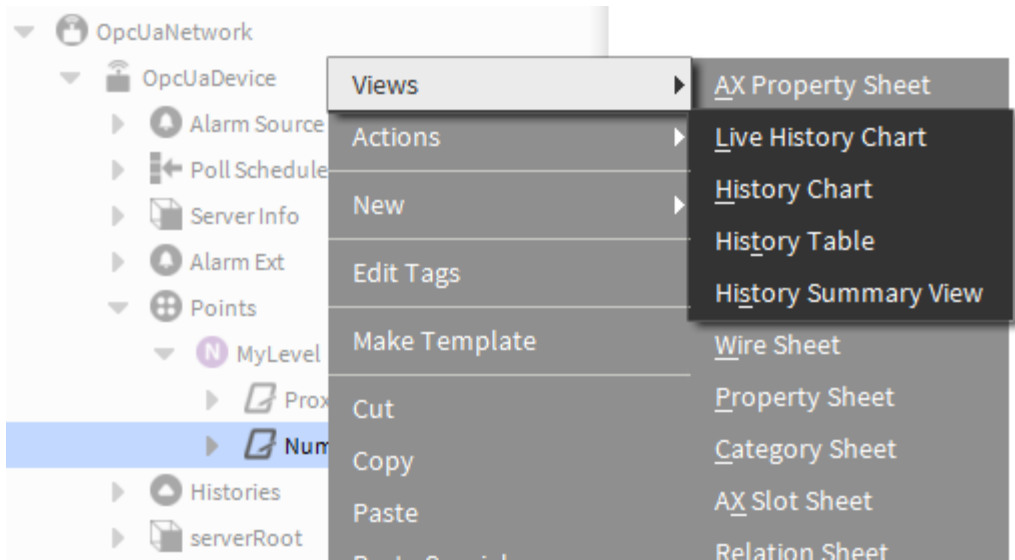
- A triangular history badge appears over the icon of the OPC UA object in the **Discovered** pane, for example:



- The value under the **Historizing** column in the **Discovered** pane reports `true`.

- Step 4.
- Select one or more objects with histories and click **Add**.  
The default history extension period for uploading history data is once in 10 minutes. To modify this period, configure the **Execution Time** property by changing the **Interval** value.
- Step 5.
- To view histories on the property sheet, right-click the point and click **Views > AX Property Sheet**.





Related reference

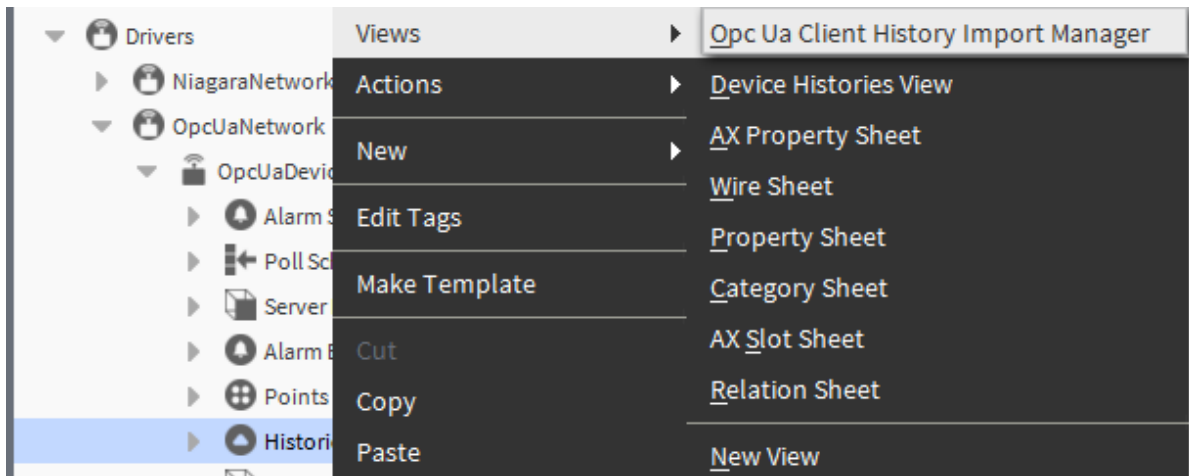
- [opcUaClient-OpcUaClientHistoryDeviceExt](#)

Importing OPC UA Histories without using a control point

The `OpcUaDevice` component’s histories extension (`OpcUaClientHistoryDeviceExt`) can import OPC UA histories without creating a control point.

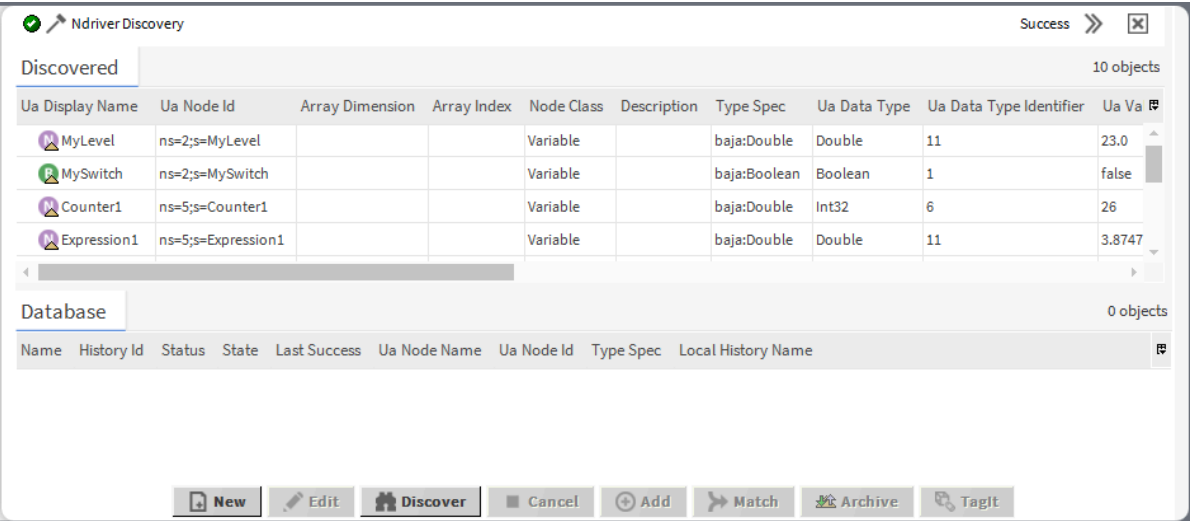
**WARNING:** Be aware that you can duplicate an OPC UA history import by unintentionally using both the `OpcUaClientHistoryImport` component and a control point with an `ImportHistoryExt` for the same discovered object. Refer to “Adding points containing OPC UA histories.”

- Step 1. In the Nav tree, right-click on `Histories` under the `OpcUaDevice` and click `Views > Opc Ua Client History Import Manager`.

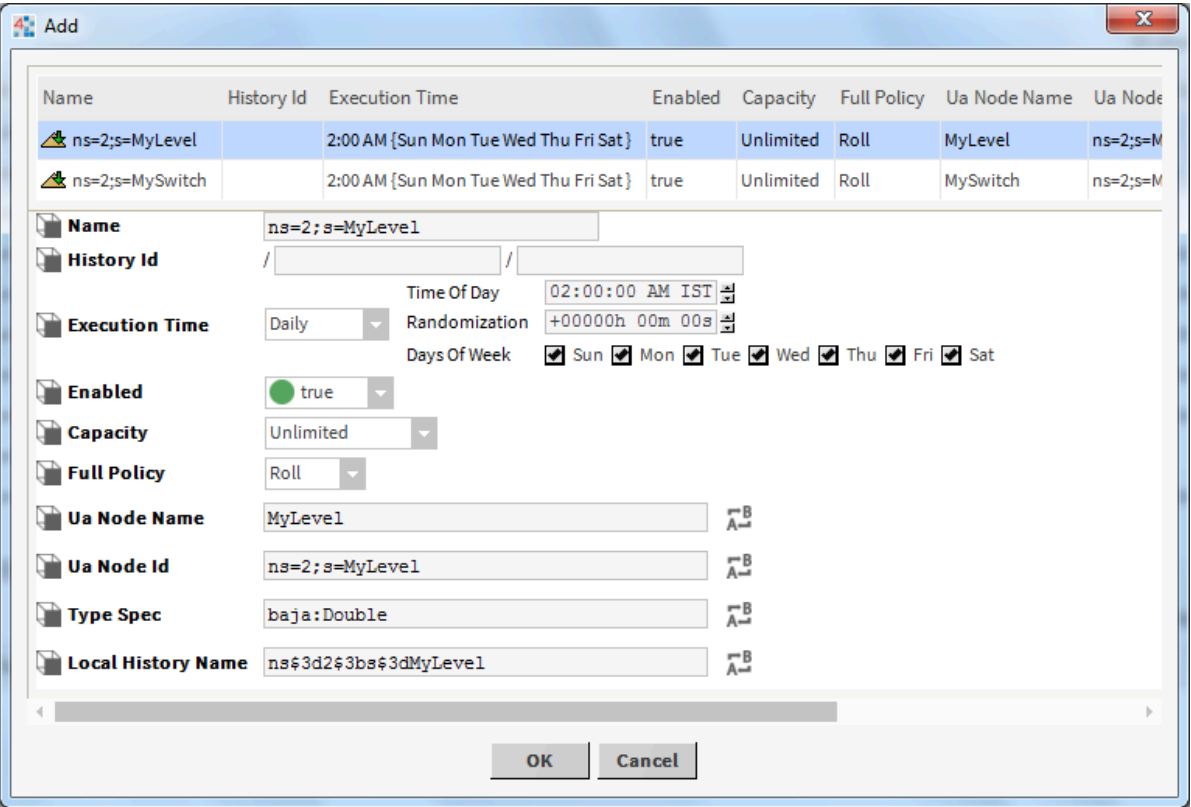


- Step 2. In the view, click **Discover** and wait a few moments for the job to run.

On completion, the discovery job displays a list of histories in the **Discovered** pane.



- Step 3. Select one or more histories to be imported and click **Add**.  
The **Add** window opens.



- Step 4. Modify properties to set up the history import as desired and click **OK**.

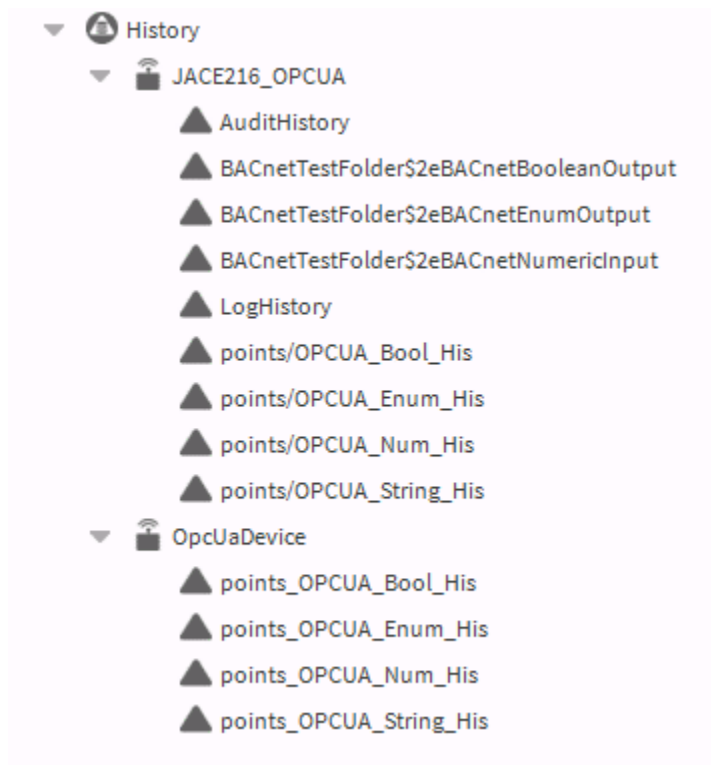
**Result**

The driver imports the histories to the station's history database at the time specified in the **Execution Time** property.

## Important note about OPC UA Histories

When you discover and add histories in the OPC UA client, they are automatically stored in two locations. Adding histories using the History extension places them under the respective driver name within the history section. Similarly, when you add a point with history to the Point device extension, it appears under the station history.

**Figure 2.** OpcUa Histories



Following is an explanation of two methods for importing histories from the OpcUa server into the client.

- **OpcUaClientPointDeviceExt:** When you identify a point with a history on the discovered OPC UA server in the **OpcUaClientPointManager** view, adding this point to the station as a Niagara point will automatically create an **ImportHistoryExt** as a subordinate to the control point. By default, these histories are located under the station History.
- **OpcUaClientHistoryDeviceExt:** Each **OpcUaDevice** includes an **OpcUaClientHistoryDeviceExt**. This device extension's **OpcUaClientHistoryImportManager** view allows you to uncover all histories collected by the connected OPC UA Server. When you discover and add these histories, the system retrieves and stores the history of that particular point under the client driver name within the History.

Preferably, you should add histories to the Points device extension. However, it is possible to collect histories without creating a Control Point by utilizing the Histories device extension..

## OPC UA Client Histories

This topic explains how the OPC UA collects histories using the timestamps for the points.

Starting with the Niagara 4.15 and Niagara 4.14.u1 releases, the OPC UA driver enables users to create and manage historical data by storing time-stamped values. When an OPC UA server gathers historical data, it collects two timestamps for each data point: the Source Timestamp and the Server Timestamp.

The OPC UA Client receives the Source and Server timestamps for proxy points. These two timestamps are defined as follows:

**Source Timestamp:**

This timestamp is generated by the device or system, reflecting the time when the data was generated by the source.

**Server Timestamp:**

This timestamp is generated by the server, indicating the time when the server received and stored the data.

You can configure history records to use the Server Timestamp by setting the **Use Server Timestamp** property.

When importing histories, if you set **Use Server Timestamp** to `true`, the system will use the Server Timestamp. If it is set to `false` the Source Timestamp will be employed instead.

When adding a new point with a history, navigate to the added point in the point database to configure the **Use Server Timestamp** property within the point's history extension.

Property Sheet	
N Random (Numeric Point)	
Facets	units=null,precision=2,min=-inf,max=+inf >> ⌚
Proxy Ext	Opc Ua Client Proxy Ext
Out	0.00 {stale}
hist	Numeric Import History Ext
Status	{ok}
Fault Cause	
Enabled	<input checked="" type="radio"/> true
History Name	%parent.name% ?
History Config	Interval: irregular, Record Type: numeric...
Last Record	08-Sep-1977 11:19 AM EDT <input type="checkbox"/> Hidden -2.00
State	Idle
Execution Time	10 minutes {Sun Mon Tue Wed Thu Fri Sa...}
Last Attempt	19-Nov-2024 07:16 AM EST
Last Success	19-Nov-2024 08:04 AM EST
Last Failure	19-Nov-2024 07:07 AM EST
Use Server Timestamp	<input checked="" type="radio"/> true
Change Tolerance	0.00
Precision	32 bit
Min Rollover Value	<input checked="" type="checkbox"/> null 0.00
Max Rollover Value	<input checked="" type="checkbox"/> null 0.00

## Adding a control point to invoke an OPC UA method

The driver provides a means of invoking OPC UA method objects defined in a OPC UA server. My Method icon

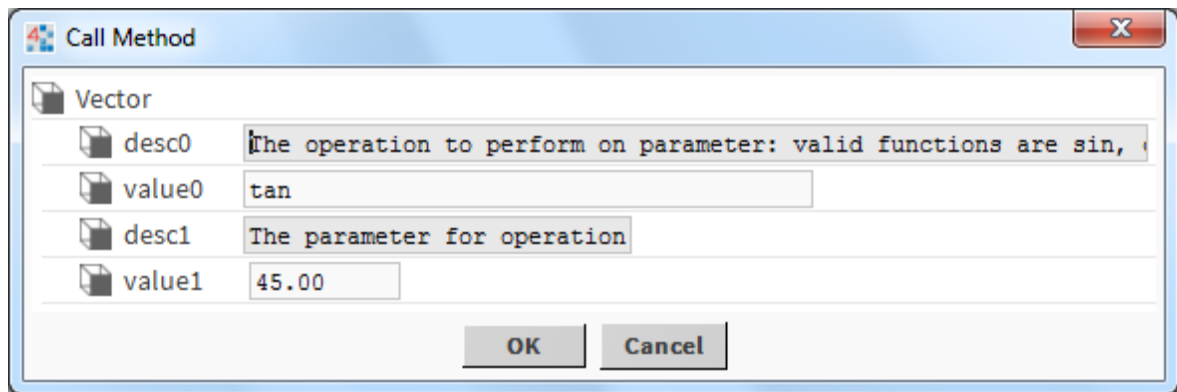
() identifies discovered OPC UA methods.

### Prerequisites:

The discovered objects on the OPC UA server are visible in the **Opc Ua Point Manager** view.


- Step 1. To open the **Opc Ua Client Point Manager**, expand **Config > Drivers > OpcUaNetwork > OpcUaDevice** and double-click the **Points** folder.
- Step 2. To run a discovery job, lick **Discover**.
- Step 3. Select a **MyMethod** and click **Add**.  
The driver adds the method in the **Discovered** pane to the database. The added control point now has a **callMethod** action.
- Step 4. Right-click on the method and click **Actions > callMethod**.

If the method has defined input arguments, a **Call Method** window opens with the arguments.



A descriptor displays each argument. You may need to scroll the descriptor to see the complete text.

**Step 5.** Enter the desired argument values and click **OK**.

Database						
Name	Type	Out	Ua Node Id	Array Index	Ua Status Code	Tuning Policy Name
 MyMethod	Opc Ua Method	1.00 [ok]	ns=2;s=MyMethod			defaultPolicy

The `OpcUaMethod` control point is a subclass of `BStringPoint`. The `out` property is a `StatusString` type. The data type of the method return value may not be a string. In addition to setting the `Out` property by converting the method results to a string, the driver adds a `Result` property of the appropriate `BStatusValue` type.

This invokes the method in the OPC UA server with the provided arguments.

## Result

On completion, the `Out` property reflects and displays the control point.

## OpcUaServer alarm acknowledgment processing

An OPC UA client can subscribe and receive OPC UA events from an OPC UA server. The **OPC UA client Alarm Manager** may allow you to acknowledge a received event back to the OPC UA server. It may also allow the user to command the OPC UA server to enable or disable the source of the received event.

The OPC UA server's **OpcUaServerAlarmDeviceExt** processes the Acknowledge, Enable, and Disable commands. **OpcUaServerAlarmDeviceExt** is a child of each namespace component. Each OPC UA server namespace has a frozen slot named **AlarmExt**. This extension is a type of **OpcUaServerAlarmDeviceExt**. It keeps track of all of the alarmable points under this namespace. When the extension receives an Acknowledge command for the OPC UA client, it attempts to locate the alarm record being acknowledged and acknowledges the alarm. When the extension receives a Disable command, it disables the alarm from generating any more OPC UA events from the associated control point until it receives an Enable command.

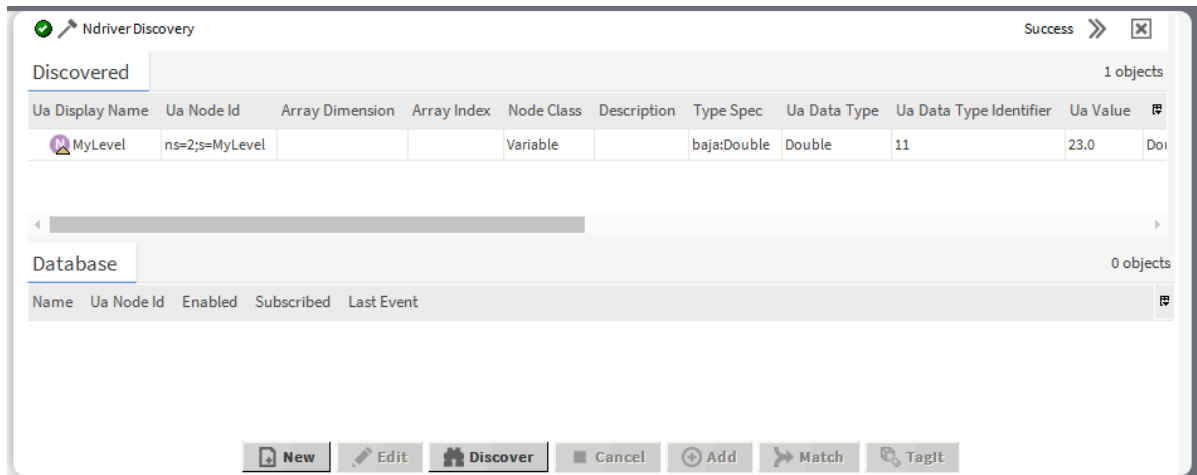
## Subscribing for OPC UA alarm events

The **OpcUaDevice** component's **Alarm Ext** (**OpcUaClientAlarmDeviceExt**) component subscribes to OPC UA alarm events.

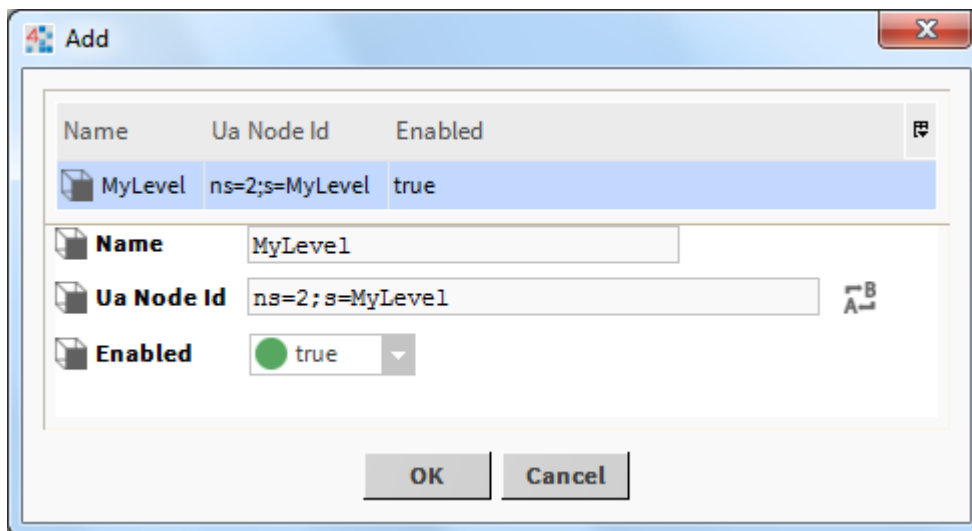
Step 1. In the Nav tree, right-click on **Alarm Ext** under the **OpcUaDevice** and click **Views > Opc Ua Client Alarm Manager**.

Step 2. In the view, click **Discover**.

Discovery scans the Opc UA server looking for OPC UA variables that are capable of generating alarms and displays them in the **Discovered** pane.

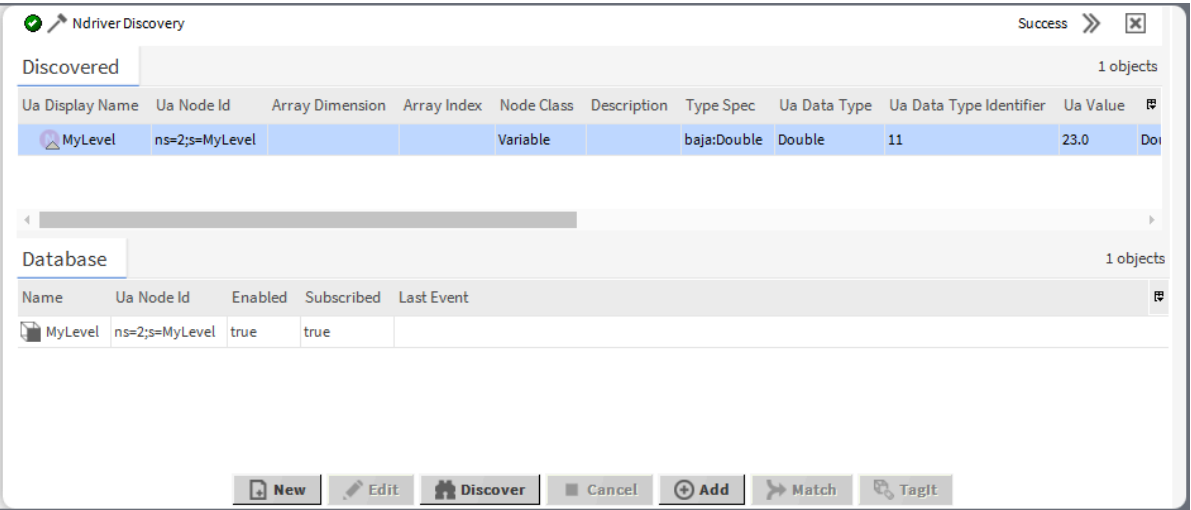


Step 3. Select the variable to be imported and click **Add**.  
The **Add** window opens.



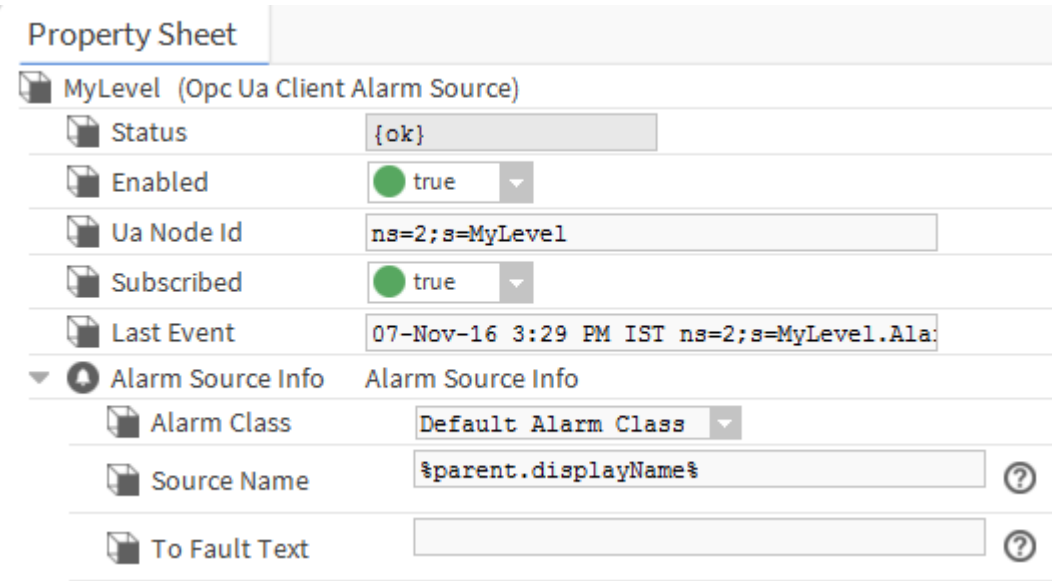
Step 4. Modify properties as desired and click **OK**.

The driver adds an **OpcUaClientAlarmSource** component to the station database.



The Last Event column displays a summary of the last alarm event received from the OPC UA server for the specified OPC UA node.

The added **OpcUaClientAlarmSource** component contains an **AlarmSourceInfo** component.



Step 5. Configure the **AlarmSourceInfo** to route the received OPC UA alarm events to alarm recipients.

The **Enabled** property subscribes and unsubscribes to receive OPC UA alarm events on the given OPC UA node.

When the node receives a new OPC UA alarm event it routes the alarm event through the specified alarm class in the station's **AlarmService**. The system then treats it as any other alarm.

**Related reference**

- [opcUaClient-OpcUaClientAlarmDeviceExt](#)



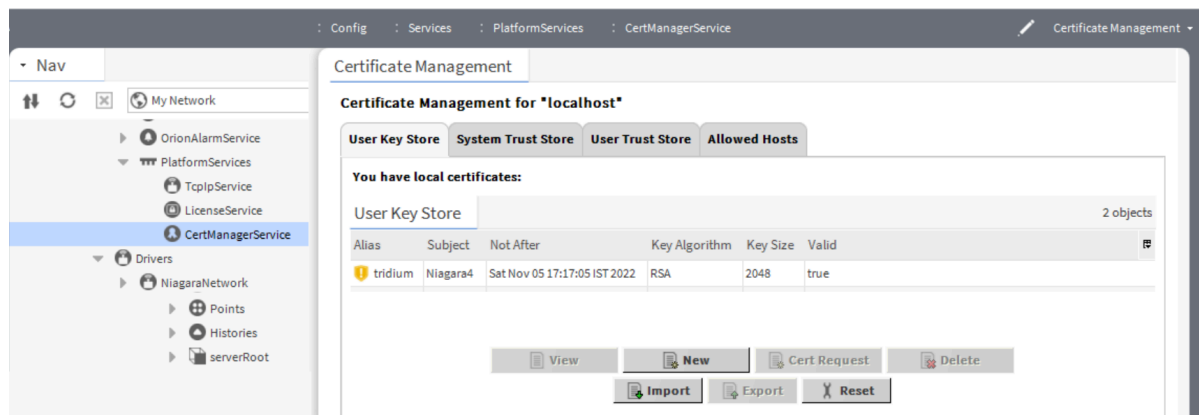
## Generating an OPC UA client certificate

This topic explains how to generate a client certificate using the added URI field in the certificate to establish a secure server connection from the client to the server.

### Prerequisites:

- The certificate is used for only OPC UA users only.
- Workbench running on a PC or laptop computer.

Step 1. To generate a certificate, expand **Station > Config > Services > PlatformServices** and double-click **CertManagerService**. The **Certificate Management** view opens.



Step 2. Click the **New** button at bottom of the view.

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

Generate Self Signed Certificate

**Generate Self Signed Certificate**  
Generates a self signed certificate and inserts it into the keystore

Alias: OPCUA (required)

Common Name (CN): OPCUA (required)  
\* this may contain the host name or address of the server

Organizational Unit (OU):

Organization (O): Acme (required)

Locality (L):

State/Province (ST):

Country Code (C): IN (required)

Not Before: 11-Mar-2022 07:09 PM IST

Not After: 11-Mar-2023 07:09 PM IST

Key Size: ☐ 1024 bits ☒ 2048 bits ☐ 3072 bits ☐ 4096 bits

Certificate Usage: ☒ Server ☐ Client ☐ CA ☐ Code Signing

Alternate Server Name:

Alternate Server URI: opal.acmecorp.com:OPCUA:NiagaraOpcUaClient

Email Address:

Key Usage: ☒ Digital signature ☐ Non-repudiation ☒ Key encipherment ☒ Data encipherment ☐ Key agreement ☐ Certificate signing ☐ CRL signing ☐ Encipher only ☐ Decipher only

OK Cancel

Step 3. Give the certificate at least an **Alias**, **Common Name (CN)**, **Organization**, **Locality**, **State/Province**, and **Country Code**.

- Use **Alias** to identify this as an OPCUA certificate.
- The **Common Name (CN)** becomes the **Subject** (also known as the Distinguished Name). For OPCUA certificate, the **Common Name (CN)** may be the same as the **Alias**.
- **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** 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](http://countrycode.org)).
- **Not Before** and **Not After** define the period of validity for the certificate.
- For **Certificate Usage**, the radio button should be set to **Server**.

Step 4. Enter the URI in the **Alternate Server URI** field; it should be in the format: `urn:<full.computer.name>:OPCUA:NiagaraOpcUaClient` and the select checkbox in the **Key Usage** set to **Data encipherment** in the certificate.

**NOTE:**

- The full computer name can be found in **Control panel > System & Security > System**; it is of the type `hostname.domain` or just `hostname` in some cases.
- While connecting from client to server, the URI provided in the client certificate should match in the Application URI for the server. If doesn't match the URI, it sends an error message as `Bad_CertificateUriInvalid 0x80170000`. The URI specified in the Application Description does not match the URI in the certificate.

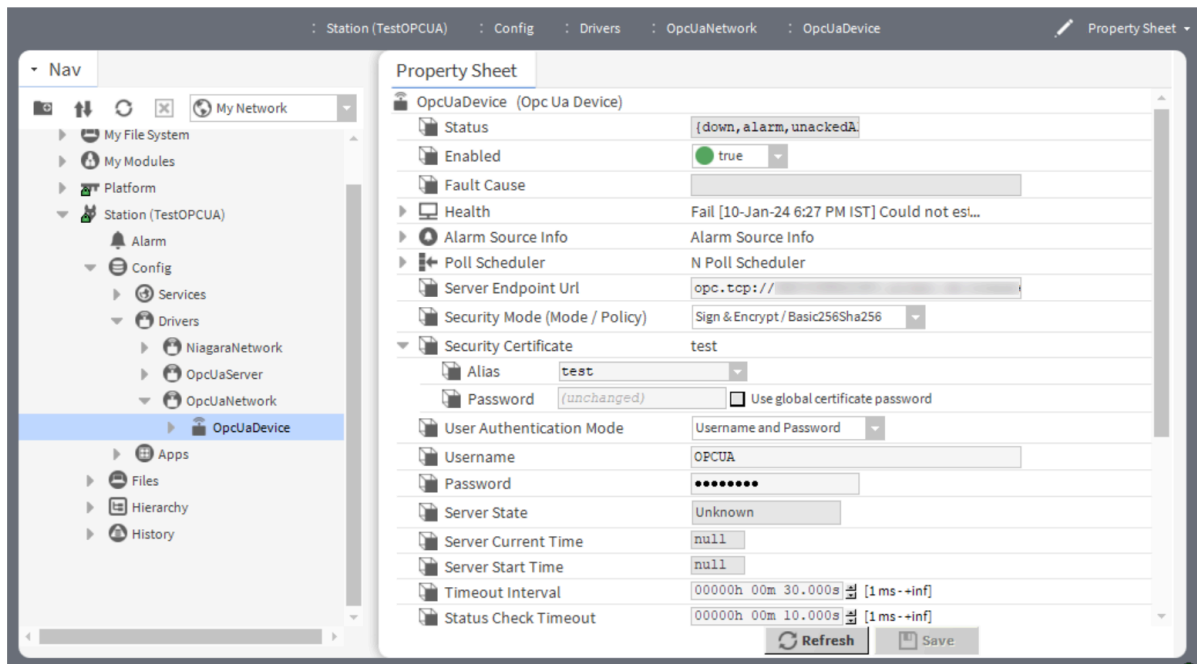
Step 5. When you have filled in the required fields, click **OK**.  
The **Private Key Password** window opens.



Step 6. Enter and confirm the password, then click **OK**.  
The system submits the certificate for processing in the background. A pop-up window appears on your screen advising you that it may take time to generate the certificate. The length of time it takes depends on the key size and the platform's processing capability. When created, the certificate appears as a row in the User Key Store table.

Step 7. To configure the certificate, open the device's **Property Sheet** by right-clicking on **OpcUaDevice** followed by clicking **View > Property Sheet**.

The Property Sheet view opens.



- Step 8. Expand **Security Certificate** property, select the certificate from the **Alias** drop-down menu, enter the **Password** and click **Save**.

## Generate an OPC UA Client Certificate for Third-Party Servers

The topic describes how to generate a client certificate using scripts by running the appropriate commands in the command prompt or Git Bash to establish a secure server connection from the client to the server. To ensure compatibility with OpcUaClient implementation, automatically uses a certificate for signing purposes with the keyCertSign usage while generating the certificate.

### Prerequisites:

- OpenSSL is installed on your system so that you can use the script file from the default windows command line.
- Niagara station is running.
- The hostname is the full device name. To find the full device name in the Windows menu, choose **Start > Settings > System > About**, and in **Device Specifications** you can find **Full device name** or from a command line, type the following net config workstation, and you can find the string **Full Computer name**.

Step 1. To generate a certificate, follow the below choices.

- If you are using windows, open the command prompt, type the following command and press Enter.

```
gen-opc-client-cert.bat
```

- If you are using Git Bash application, open the Git Bash prompt, type the following command and press Enter.

```
./gen-opc-client-cert.bat
```

- If you are using Linux or WSL, open the command prompt, type the following command and press Enter.

gen-opc-client-cert.sh

Prompts should appear in the Command Prompt (Terminal or Shell) window.

- Step 2. Follow the instructions displayed in the window, type the client hostname and press Enter.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

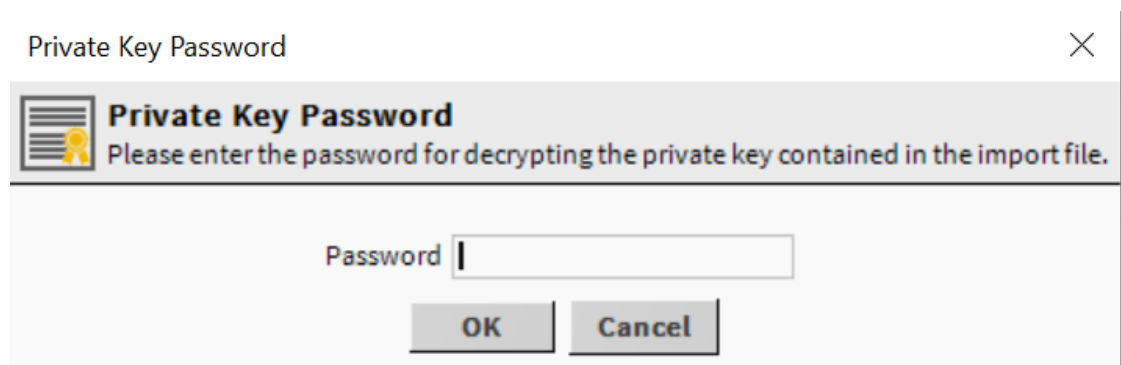
C:\Users\██████\Documents>gen-opc-client-cert.bat
Enter the client hostname [██████]: Acme.com
Enter certificate validity in days [365]: 364
Enter destination file [.\client.pem]: .\test.pem
.....+..+.....+*****+*..+.....
.+.....+.....+.....+.....+
..+..+.....+..+.....+.....+.....+.....+
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [US]:IN
State or Province Name (full name) []:Karnataka
Locality Name (eg, city) []:Bengaluru
Organization Name (eg, company) []:Acme
Organizational Unit Name (eg, section) []:Engineering
Common Name (e.g. server FQDN or YOUR name) [Acme.com]:OPCUA
Email Address []:XYZ@Acme.com
Cert written to .\test.pem
```

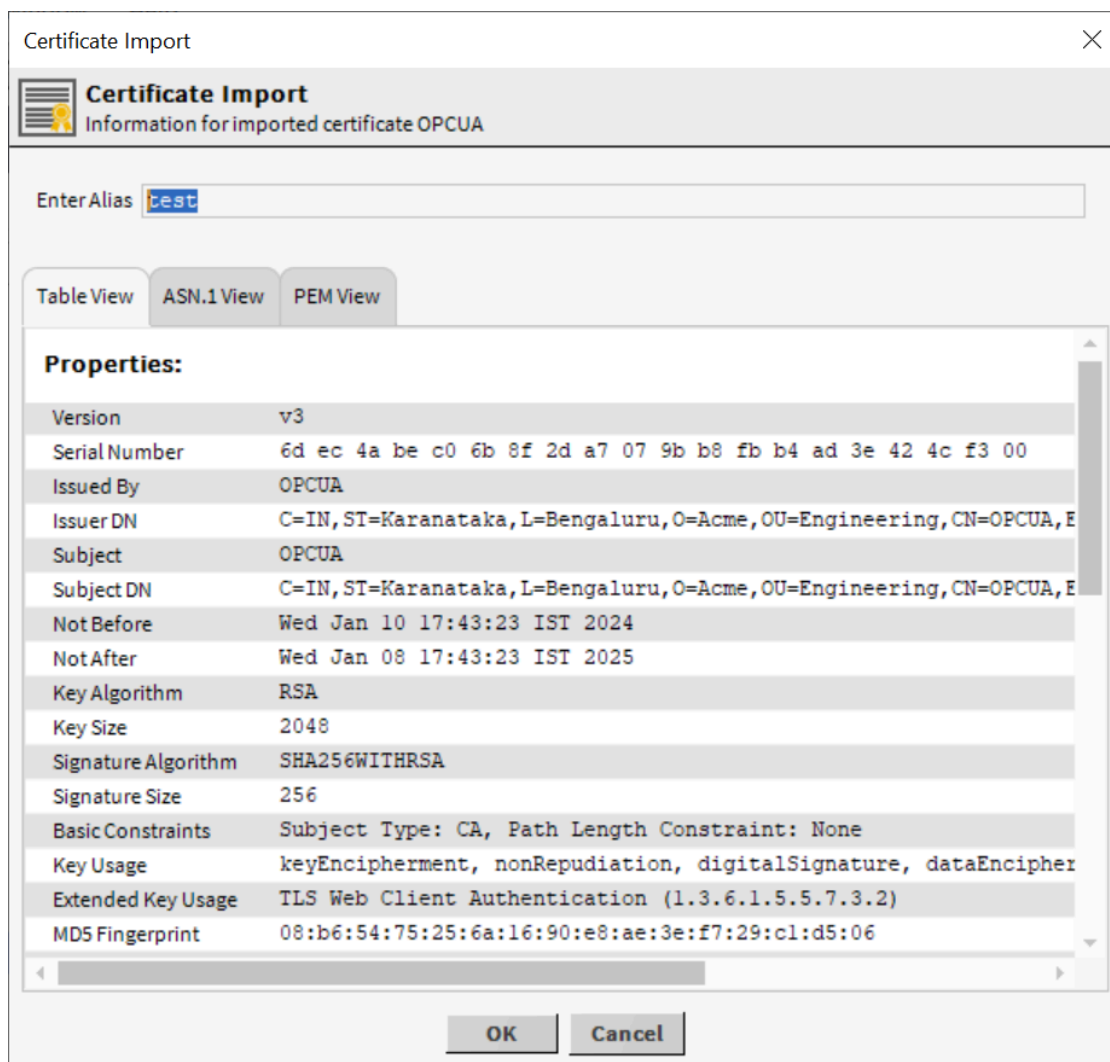
- Step 3. Type the certificate validity in days and press enter.  
**NOTE:** Based on requirements, you can specify duration for certificate validity.
- Step 4. Type the destination file name in the format `.\yourCertName.pem` and press Enter.
- Step 5. Type a random PEM Pass Phrase, press enter and verify the PEM Pass Phrase, press Enter.  
 The PEM passphrase can be any random passphrase with sufficient strength. Use the same PEM passphrase throughout the procedure.
- Step 6. Type the following information and press Enter after each step.

- 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](http://countrycode.org)).
- **State/Province**
- **Locality Name**
- **Organization Name** is the name of the company.
- **Organizational Unit Name**
- **Common Name (CN)**
- **Email Address**

It displays **Cert** written to the destination and generates a certificate in the given destination file.

- Step 7. To import the PEM certificate, open Workbench, expand **Config > Services > PlatformServices**, and click **CertManagerService**.
- In the **Certificate Management** view, click **Import**, browse to the destination file and enter the password, for decrypting the private key.





The **Certificate Import** wizard opens.

- b. To change the existing Alias, enter the new Alias name and click **OK**.



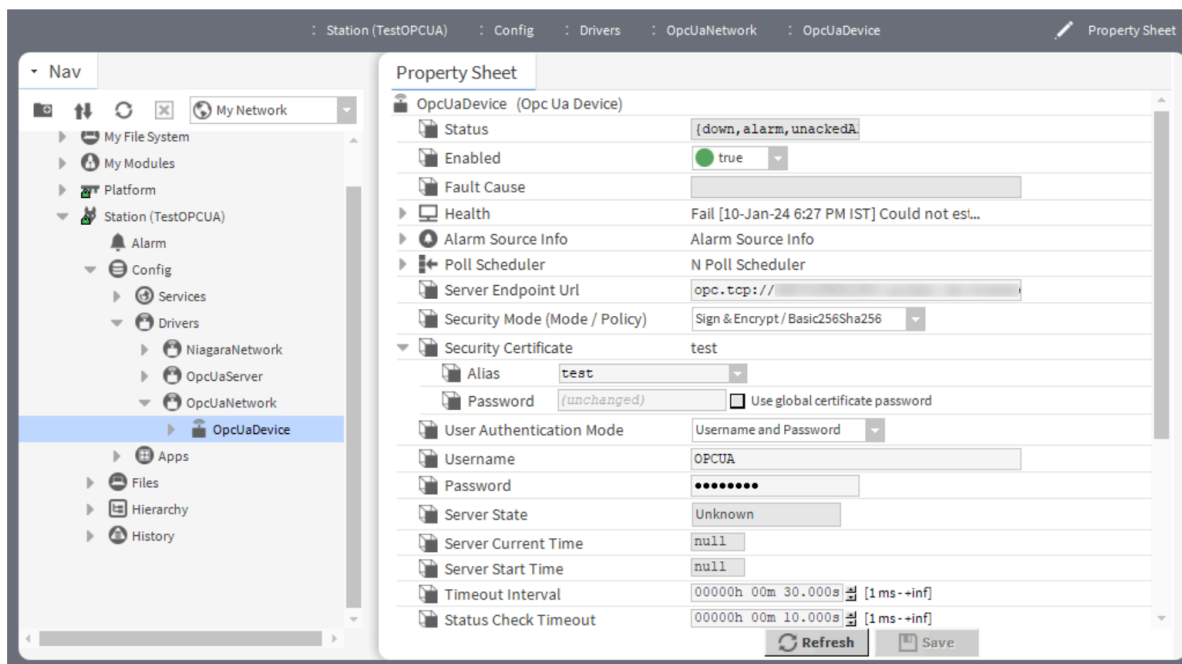
The **Private Key Password** window opens.

- c. Type the **Password**, verify the **Confirm** password for encrypting the private key when saving

it into the key store and click **OK**.

The certificate appears as a row in the User Key Store table.

- Step 8. To configure the certificate, open the device's **Property Sheet** by right-clicking on **OpcUaDevice** followed by clicking **View > Property Sheet**.



The **Property Sheet** view opens.

- Step 9. Expand the **Security Certificate** property, select the certificate from the **Alias** drop-down menu, enter the **Password** and click **Save**.  
The certificate is now available for Third-Party servers.



# Chapter 4. Opc Ua Reference

The following topics provide conceptual information on the OPC UA security architecture, including descriptions of the security policies, security mode, security profiles and user authentication.

Also covered is information on point type and facet mapping, server alarm processing and alarm acknowledgment processing, links for third-party simulation software, as well as descriptions of the components and views present in the opcUaServer and opcUaClient modules.

## Point type mapping

The following table maps the OPC UA built-in data types to Niagara point types.

OPC UA Data Type	Boolean	Enum	Numeric	String
Boolean	X			
SByte			X	
Byte			X	
Int16			X	
UInt16			X	
Int32			X	
UInt32			X	
Int64			X	
UInt64			X	
Float			X	
Double			X	
String				X
DateTime				X
Guid				X
ByteString				
XmlElement				X
NodeId				X
ExpandedNodeId				
StatusCode				
LocalizedText				X
ExtensionObject				
Variant (array of)	X	X	X	X
DiagnosticInfo				
Enumeration		X		

## Point facet mapping

The OpcUaClient driver will attempt to initialize facets for a point based on the setup of the OPC UA Variable being proxied.

Control Point	Description
BooleanPoint	Will set the TrueText and FalseText if the OPC UA variable has "TrueState" and "FalseState" properties.
EnumPoint	Will set the EnumRange if the OPC UA variable has an "EnumStrings" property.
NumericPoint	Will set the Units if the OPC UA variable has an EUInformation property and the Unit maps to a system Unit. Will set the Precision, Max, and Min values based on the OPC UA variable data type or a "Range" property if it exist for the variable.

## Components

Components include services, folders and other model building blocks associated with a module. You drag them to a property or wire sheet from a palette. Views are plugins that can be accessed by double-clicking a component in the Nav tree or right-clicking a component and selecting its view from the **Views** menu. The component and view topics that follow appear as context-sensitive help topics when accessed by:

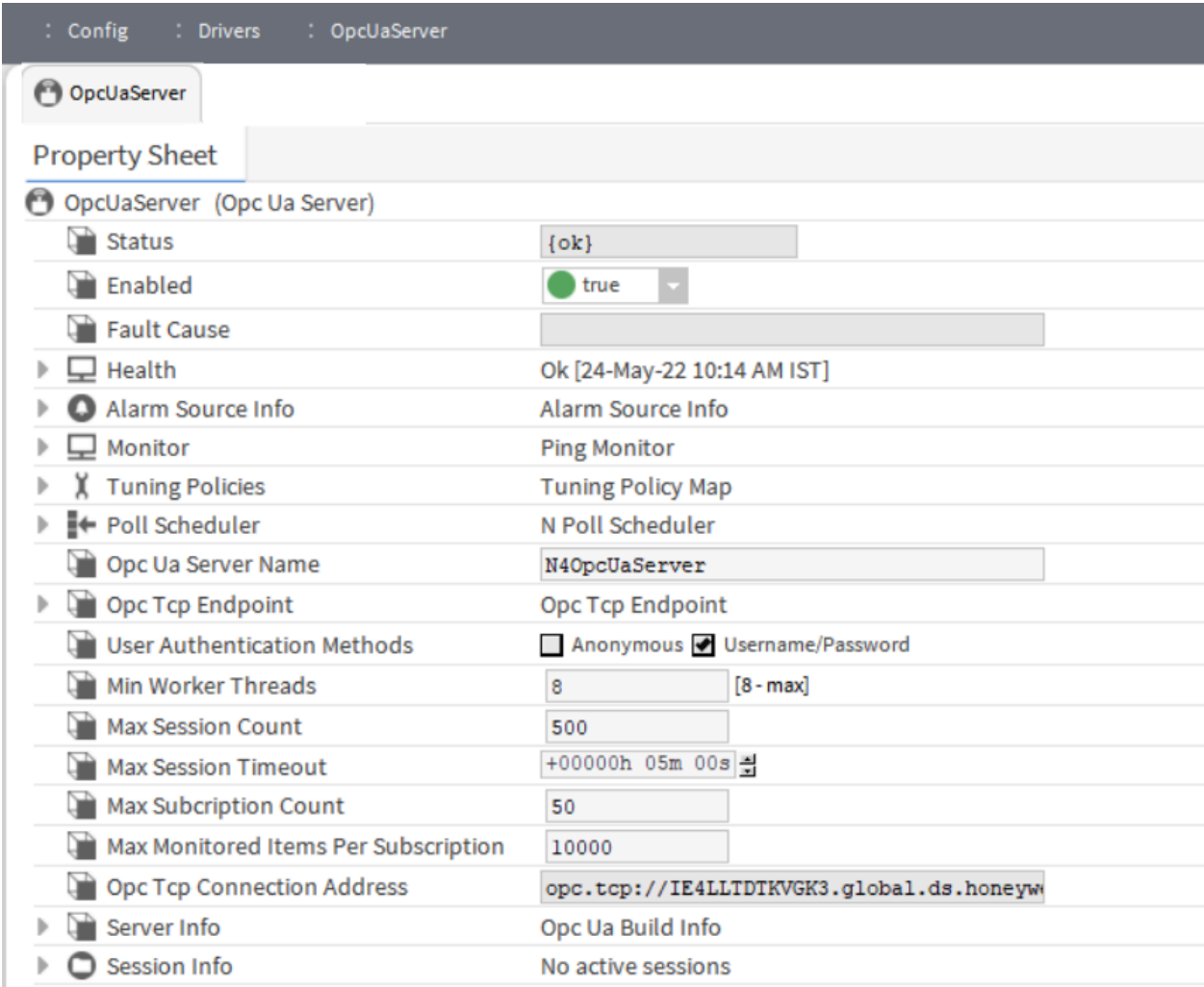
- Right-clicking on the object and selecting **Views > Guide Help**
- Clicking **Help > Guide On Target**

### opcUaServer-OpcUaServer

This component contains the device-level **OpcUaNamespace** component and configuration parameters necessary for communications with OPC UA client devices. Additionally, the OPC UA server point device extension (**OpcUaServerPointDeviceExt**) of the OPC UA namespace adds points to the server. The OpcUaServer palette houses this component.

**NOTE:** Starting with Niagara Niagara 4.14.u1 and Niagara 4.15, the OpcUaServer component provides access to specific aspects of the OPC UA Server based on user roles and permissions as they are configured in the User Service, Role Service and Category Service. These server-side settings are observed when granting access to specific aspects of the OPC UA Server and include more restrictive access for Anonymous users. See [OPC UA server, client user authorization](#) for more details.

Figure 3. opcUaServer properties



To access these properties, expand **Config > Drivers**, right-click **OpcUaServer** and click **Views > AX Property Sheet**.

In addition to the standard property (Enabled), these properties are unique to this component.

Property	Value	Description
Opc Ua Server Name	text string	Defines the name of the server. You may edit it or use the default name: N4OpcUaServer.
	additional properties	Configures the Tcp Endpoint.
Opc Tcp Endpoint, Enabled	true (default) or false	Enables (true) and disables (false) the use of a Tcp Endpoint.
Opc Tcp Endpoint, Port	52520 (default)	Defines the port number for Opc Tcp connections.

Property	Value	Description
<p><b>NOTE:</b> The port specified in the Opc Tcp Connection Address may be blocked by PC/network firewall. The firewall settings may need to be adjusted to allow data transfer on this port.</p>		
Opc Tcp Endpoint, Security Mode	check boxes (defaults to <code>sign</code> and <code>signEncrypt</code> )	Security mode of the application defines the security level for a message being exchanged during communication between client and server. The mode can be set to <code>None</code> , <code>Sign</code> , or <code>SignAndEncrypt</code> .
Opc Tcp Endpoint, Security Policies	check boxes (defaults to all)	Specifies which security mechanisms are to be used for the Secure Channel between client and server.
User Authentication Methods	check boxes (defaults to all except for <code>Anonymous</code> )	<p>Selects a method for authenticating the user. These settings must match the server's settings for a successful connection.</p> <p>Username and password values must be defined in the station's <code>UserService</code>.</p> <p><b>Certificate:</b> Choose the certificate authentication and select the <code>CA</code> or <code>Intermediate Certificate</code> (stored in User trust store) from the drop down list. Please follow Setting up certificate based authentication topic for more information.</p> <p><b>NOTE:</b> For Certificate creation and saving procedure, refer to <a href="#">Station Security Guide</a>.</p>
Max Session Count	500 (default)	All communications are done through sessions, which must be alive all the time in normal cases. This property displays the maximum number of session that can be live.
Max Session Timeout	00001h 00m 00s (default)	The server can close sessions that have not been active (no message of any type from the client received for a more extended period than <code>SessionTimeout</code> ).
Max Subscription Count	50 (default)	Displays the max number of subscription for a session.
Max Monitored Items Per Subscription	10000 (default)	Specifies the maximum number of monitored items per subscription.
Opc Tcp Connection Address	read-only	Reports the connection address for the Opc Ua Server. The station must be running for this field to be populated. This address includes the specified port number for the Opc Tcp Endpoint, for example <code>opc.tcp://ABCD1234.global.ds.honeywell.com:52520/OPCUA/N4OpcUaServer</code>
Server Info	additional properties	A separate topic documents these properties.
Session Info	additional properties	A separate topic documents these properties.

Actions

- **Ping** sends a message to a network object (device, database, etc). The message provokes a response, which indicates the current state of the object.

Related tasks

- [Setting up the OPC UA server](#)

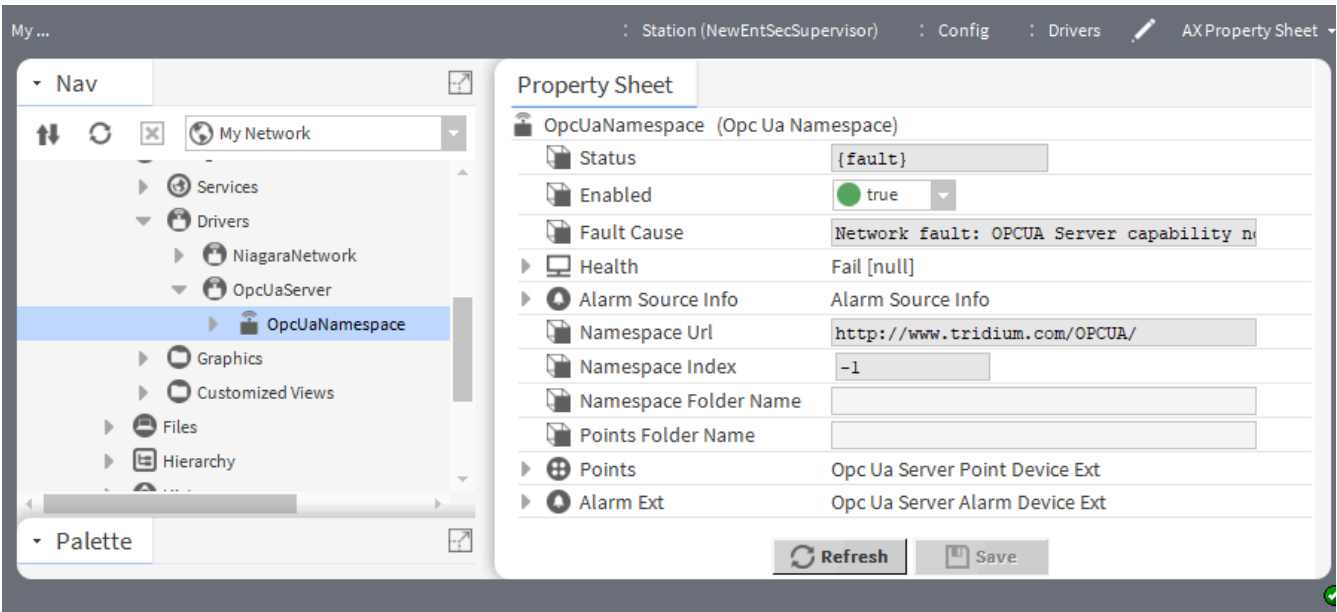
opcUaServer-OpcUaNamespace

This component provides a standard method for an OPC UA server to represent objects to OPC UA clients. It defines objects in terms of variables and applicable attributes. It typically provides a logical grouping of OPC UA variables, histories, and events that can be accessed by an OPC UA client.

To add server points to the OPC UA server, first add a namespace component. Within this namespace discover points and add then points to the namespace. Once that is done, an OPC UA client can connect to the OPC UA server and subscribe to all these points for monitoring.

This component is found in the opcUaServer palette.

Figure 4. OpcUaNamespace properties



This device-level component resides under the **OpcUaServer** node.

In addition to the common properties (Status, Enabled, Fault Cause, Health and Alarm Source Info), these properties are unique to this component.

Property	Value	Description
Namespace Url	read-only	Reports the URL of the namespace.
Namespace Index	read-only	
Namespace Folder Name	text	Defines the location of a folder.

Property	Value	Description
Points Folder Name	text	Defines the points folder name.
Points	container	Opens the OpcUaServer Point Manager.
Alarm Ext	container	Opens the OpcUaServer Alarm Manager.

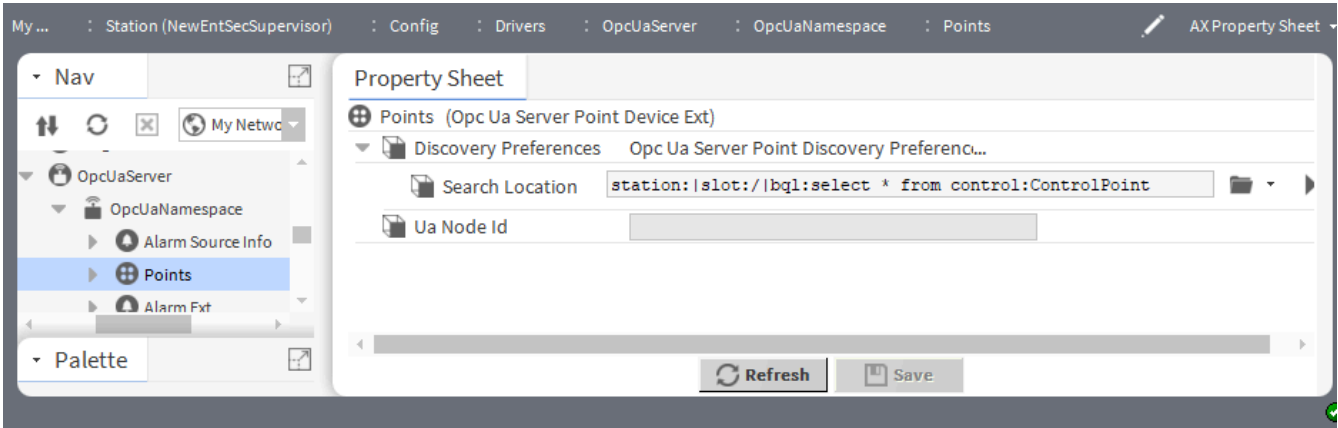
Related tasks

- Setting up the OPC UA server

opcUaServer-OpcUaServerPointDeviceExt

This component configures and provides the **Opc Ua Server Point Manager**. It is the OPC UA implementation of the standard PointDeviceExt, a frozen device extension under every OPC UA server namespace.

Figure 5. OpcUaServerPointDeviceExt properties



To access these properties, expand **Config > Drivers > OpcUaServer**, right-click **Points** and click **Views > AX Property Sheet**.

Property	Value	Description
Discovery Preferences, Search Location	ORD	Identifies the location of the points container in the station.
Ua Node Id	read-only	Reports the ID of the UA node.

opcUaServer-OpcUaServerAlarmDeviceExt

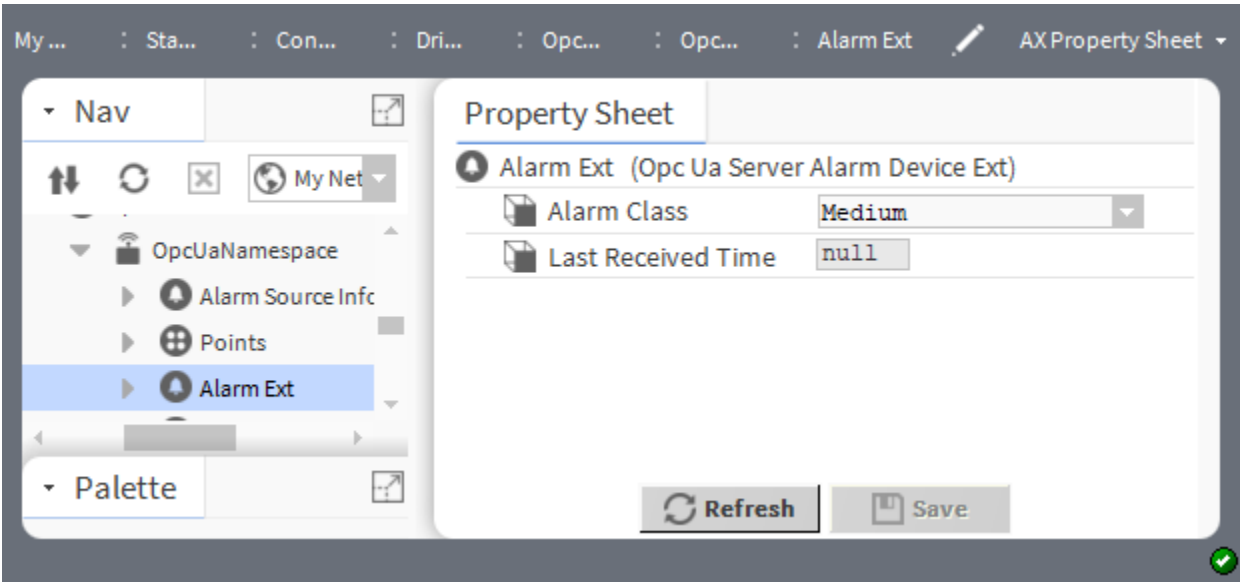
This component is a child of each Namespace component. Each OPC UA nerver namespace has a frozen slot named **AlarmExt** that keeps track of all of the alarmable points under this namespace.

The default view for for this component is the **Opc Ua Server Device Manager**.

When the OPC UA client driver receives an acknowledgement, it attempts to locate the alarm record being acknowledged and acknowledge the alarm. When the driver receives a disable command, it disables the alarm

from generating any more OPC UA events from the associated control point until it receives an enable command

**Figure 6.** OpcUaServerAlarmDeviceExt properties



To view these properties, expand **Config > Drivers > OpcUaServer**, right-click **Alarm Ext** and click **Views > AX Property Sheet**

Property	Value	Description
Alarm Class	drop-down list	<p>Specifies the alarm routing option for the component.</p> <p>Replace provides a selection list of a local alarm classes, from which to select one to use for all alarms received from this device.</p> <p>Use Existing routes alarms from this remote station to any matching alarm class, that is, one with an identical name as that in each alarm record. If the program finds no local matching alarm class, it uses the station's default alarm class.</p> <p>Prepend adds leading text (as specified) to the incoming alarm class string, then routes it to any local matching alarm class in the station.</p>

Property	Value	Description
		<code>Append</code> adds trailing text (as specified) to the incoming alarm class string, then routes it to any local matching alarm class in the station.
Last Received Time	read-only	Reports the last time this extension received an alarm.

### Related tasks

- [Adding a server point with alarm and history extensions](#)

#### opcUaServer-OpcUaServerDeviceFolder

This component is the OPC UA server implementation of a standard folder under the **OpcUaServer** container. Typically, you add such folders using the **New Folder** button in the **Opc Ua Server Device Manager** of the **OpcUaServer**. This component is also available in the **opcUaServer** palette.

#### opcUaServer-OpcUaServerPointFolder

This component is the OPC UA server implementation of a standard folder under the **OpcUaNamespace > Points** container.

You add such folders using the **New Folder** button in the **Opc Ua Server Point Manager** of the OPC UA server point device extension. ThisFolder is also available in the **opcUaServer** palette.

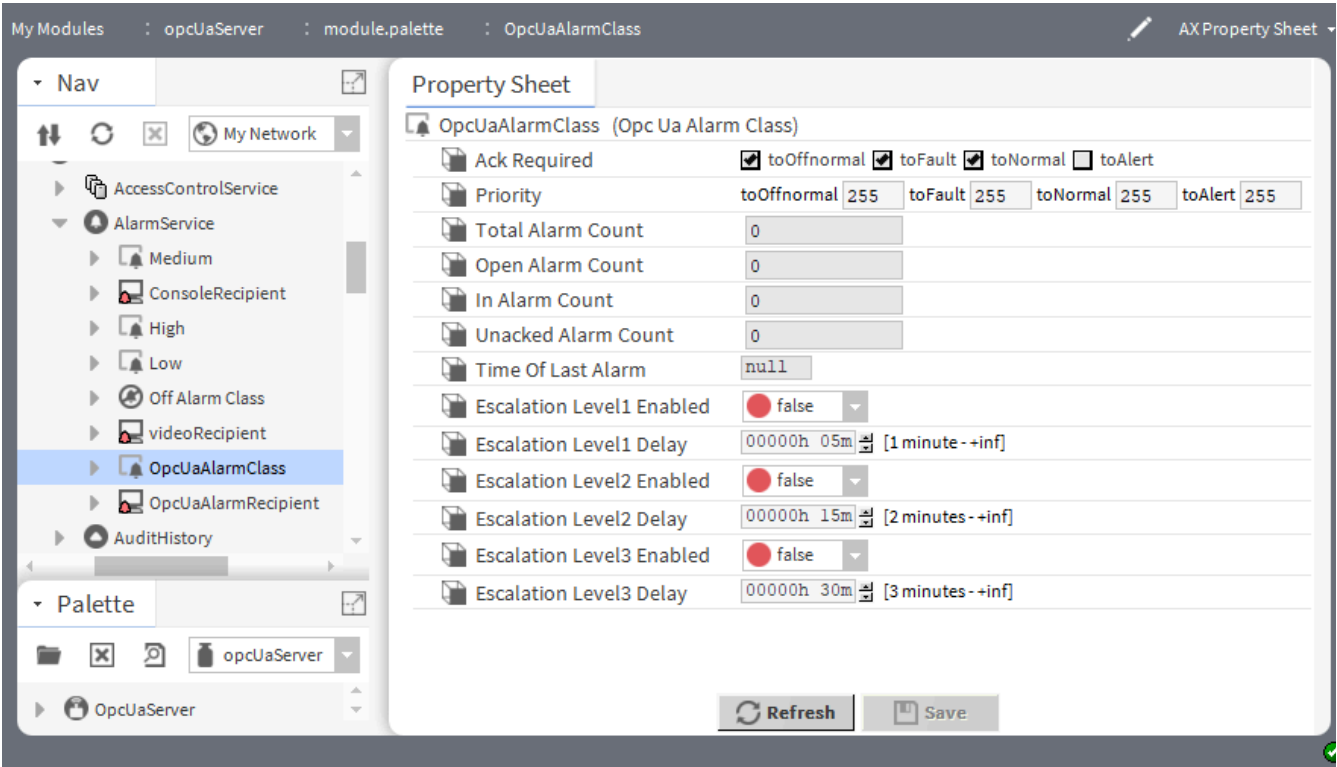
#### opcUaServer-OpcUaAlarmClass

This component groups alarms that have the same routing. It collects the alarms of OPC UA server points if the alarm extension is configured.

This component is found in the **opcUaServer** palette. It goes on the **AlarmService**.



Figure 7. OpcUaAlarmClass properties



To access these properties, expand **Config > Services > AlarmService** and double-click **OpcUaAlarmClass**.

Property	Value	Description
Ack Required	check boxes (default to toOffNormal, otFault and toNormal)	Selects which conditions require acknowledgment.
Priority	four fields for configuring importance	<p>Define the priority level to assign to the alarm class for each component state transition (from normal to offnormal, from normal to fault, from normal to alert, from offnormal to fault and from alert to normal.</p> <p>The lower the number, the more significant the alarm. The highest priority alarm is number 1.</p>
Total Alarm Count	read-only	Displays the total number of alarms assigned to the <b>Alarm class</b> from all sources.
Open Alarm Count	read-only	Displays the current total number

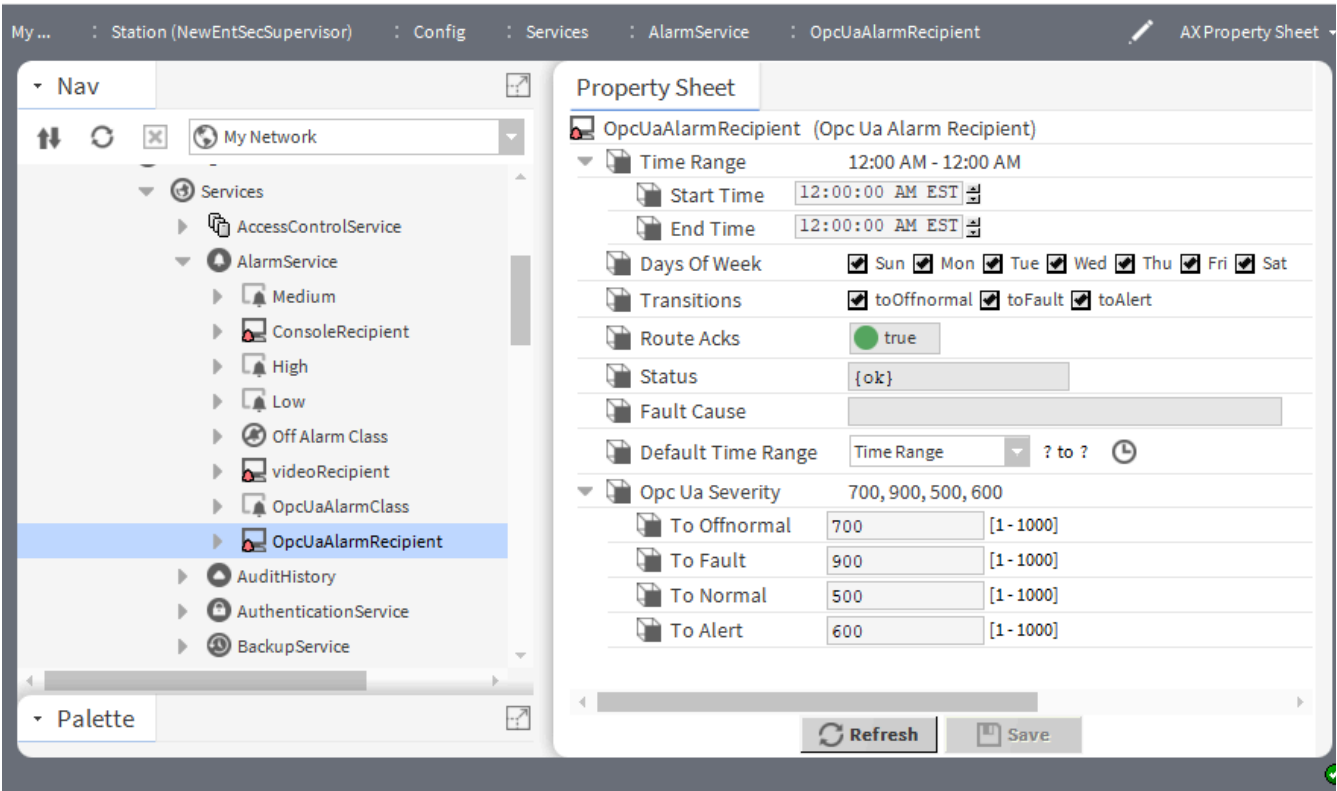
Property	Value	Description
		of alarms that are unacknowledged and normal or unacknowledged and an alert.
In Alarm Count	read-only	Displays the total number of alarm conditions.
Unacked Alarm Count	read-only	Displays the total number of unacknowledged alarms.
Time Of Last Alarm	read-only	Displays when the system generated the last alarm assigned to this <code>Alarm Class</code> .
Escalation Level(n) Enabled	true or false (default)	Turns this escalation level on (true) and off (false).
Escalation Level(n) Delay	hours minutes (defaults to 3 minutes)	Sets the time between alarm generation and escalation. It is not the time between escalation levels. Set a time to allow an unacknowledged alarm to remain unacknowledged before you escalate it to the next level.

### opcUaServer-OpcUaAlarmRecipient

This component routes alarms to OPC UA clients and maps alarm conditions based on OPC UA Severity settings.

This component is found in the opcUaServer palette.

Figure 8. Opc Ua Alarm Recipient properties



To access these properties, expand **Config > Services > AlarmService** and right-click **OpcUaAlarmRecipient** and click **Views > AX Property Sheet**.

In addition to the standard properties (Status and Fault Cause), these properties are specific to this component.

Property	Value	Description
Time Range, Start and End Times	time of day	Sets the time of day to begin and stop the function (for example, trigger schedule, alarm event)
Days of the Week	check boxes (all default to enabled)	Specifies the days of the week to include.
Transitions	check boxes ( all default to enabled)	Selects which alarm transitions to display in the console. Only those transitions selected display although the station saves all transitions in alarm history.  Options are: toOffnormal, toFault, toNormal, toAlert

Property	Value	Description
Route Acks	read-only	Enables ( <code>true</code> ) and disables ( <code>false</code> ) the routing of alarm acknowledgements to the recipient. The framework does not route trap (event notification) acknowledgements if you select <code>false</code> .
Default Time Range	drop-down list	Selects the time range to use for reporting alarms.
Opc Ua Severity, To Offnormal	number (defaults to 700)	Assigns a severity index (integer) based on the urgency of an alarm event. By default, To Offnormal is urgent.
Opc Ua Severity, To Fault	number (defaults to 900)	Assigns a severity index (integer) based on the urgency of an alarm event. By default, To Fault is the most urgent.
Opc Ua Severity, To Normal	number (defaults to 500)	Assigns a severity index (integer) based on the urgency of an alarm event. By default, To Normal is the least urgent.
Opc Ua Severity, To Alert	number (defaults to 600)	Assigns a severity index (integer) based on the urgency of an alarm event. By default, To Alert is moderately urgent.

### opcUaServer-OpcUaAuthenticationScheme

This component authenticates the user on an OPC UA server, and ensures the password requirements are adhered to. Any new OPC UA user should be associated with this scheme.

This component must be added to the station's **Authentication Service**. Then, when creating a new OPC UA-specific user, you configure the user's authentication scheme as: `OpcUaAuthenticationScheme`.

**NOTE:** When making any changes on OPC UA server side, you must disable and then re-enable the OPC UA server.

Property	Value	Description
Global Password Configuration	container	Contains additional subproperties
Password Strength	container	Contains the subproperties configured for a strong password which requires a minimum of 10 characters; and at least 1 of each of the following characters: lowercase, uppercase, digit.
Expiration Interval	365d 00h 00m 00s (default)	Defines the period of time during which the password is valid.

Property	Value	Description
Warning Period	030d 00h 00m 00s	Defines the period of time prior to password expiration when a warning is issued.
Password History Length	1–10	Configures the number of previously used passwords to be retained. This setting restricts the user from reusing a set number of previously used passwords. The range is 1–10, with the default being 2.

### Related tasks

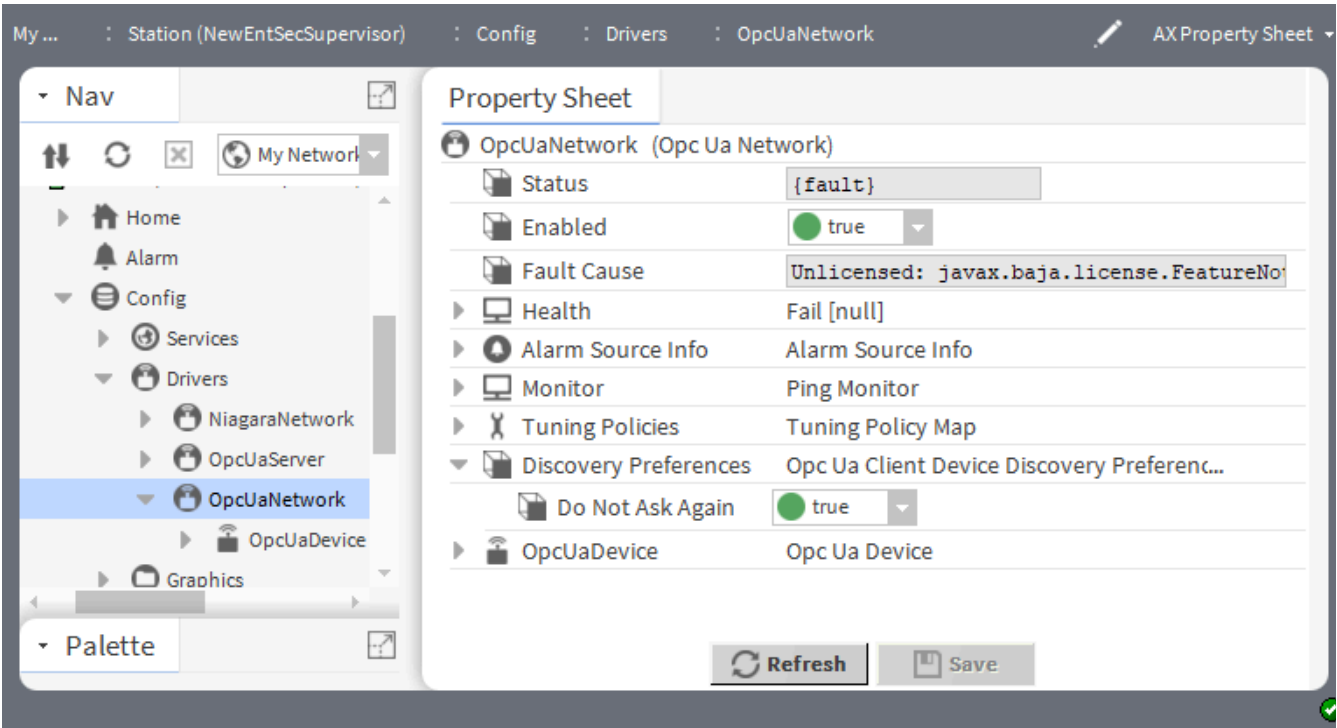
- [Configuring the server to support an OPC UA client user](#)

### opcUaClient-OpcUaNetwork

The **OpcUaNetwork** is a top-level container component for an OPC UA network in a station. It represents a network of manageable OPC UA client devices.

Within a network you can create multiple OPC UA client devices. This network provides a common monitor that monitors the status of connected OPC UA servers. This component is available in the opcUaClient palette.

Figure 9. OpcUaNetwork properties



**NOTE:** Starting with Niagara Niagara 4.14.u1 and Niagara 4.15, the OpcUaServer component provides access to specific aspects of the OPC UA Server based on user roles and permissions as they are configured in the User Service, Role Service and Category Service. These server-side settings are observed when granting access to specific aspects of the OPC UA Server and include more restrictive access for Anonymous users. See [OPC UA server, client user authorization](#) for more details.

Property	Value	Description
Discovery Preferences, Do Not Ask Again	true or false (defaults to true)	
OpcUaDevice	additional properties	Documented in a separate topic.

Actions

- Ping sends a message to a network object (device, database, etc). The message provokes a response, which indicates the current state of the object.
- Submit Discover Job discovers server objects.
- Dump Units

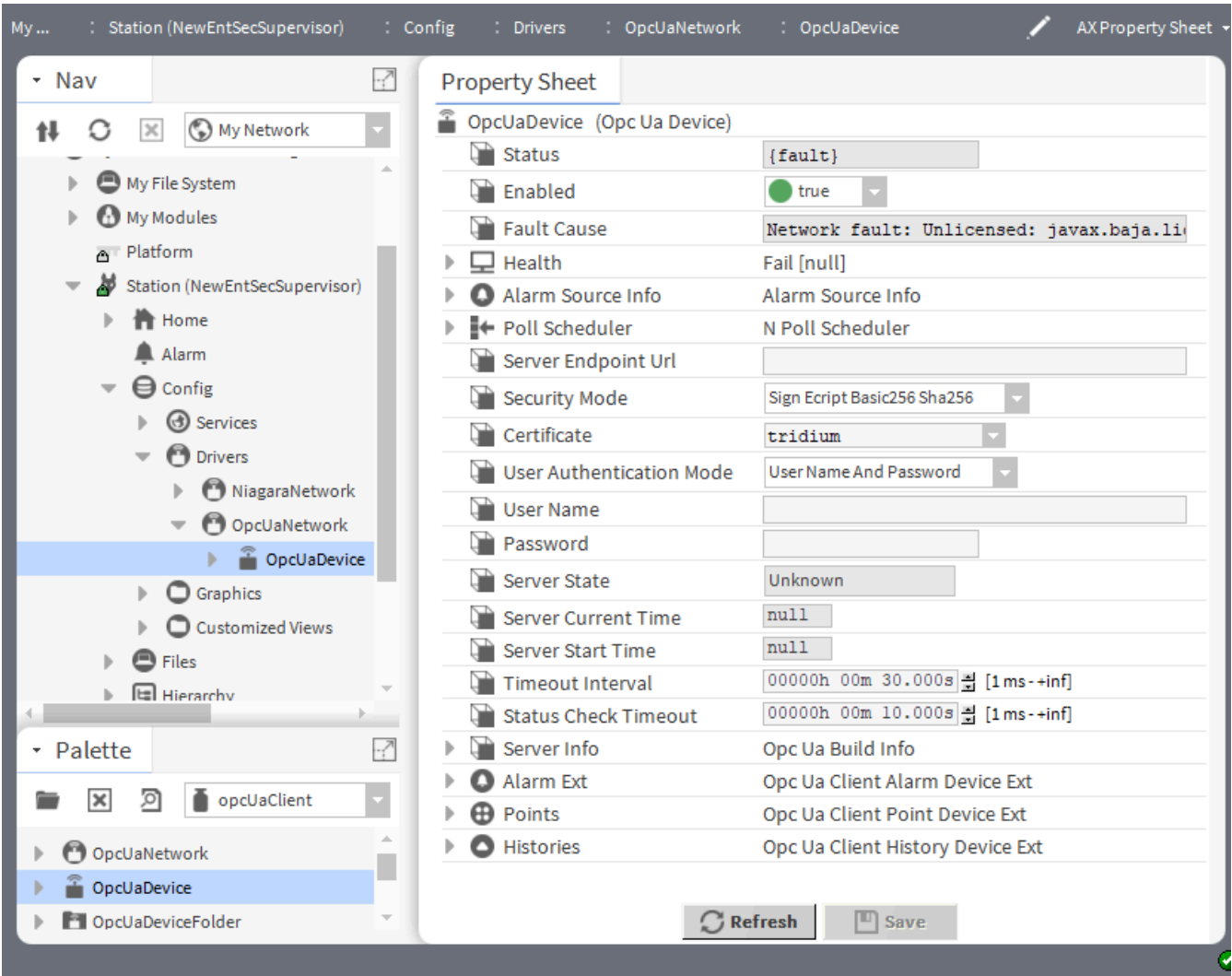
Related tasks

- [Adding the OPC UA network](#)

opcUaClient-OpcUaDevice

This device-level component in an OPU UA network represents a client connection to a specific OPC UA server and contains the configuration properties necessary for the driver to communicate with the server. The **OpcUaDevice** has a points device extension (**OpcUaClientPointDeviceExt**) that contains all subscribed proxy points.

Figure 10. OpcUaDevice properties



The default view for an OpcUaDevice is the **Opc Ua Client Manager**.

In addition to the standard properties (Status, Enabled, Fault Cause, Health, Alarm Score Info, and Poll Scheduler), these properties configure an OPC UA device.

Property	Value	Description
Server Endpoint Url	URL	Defines the connection address (URL) for the Opc Ua Server, for example: opc.tcp://IE67DTDVYXX.honeywell.com:52520/OPCUA/N4OpcUaServer

Property	Value	Description
Security Mode	drop-down list (defaults to Sign Ecript Basic256 Sha256)	<p>Specifies what security should be applied to message exchanges during a session.</p> <p>Sign Ecript Basic256 Sha256 signs and encrypts all messages. The signature or signing of the message detects if it has been manipulated by any third party.</p> <p>Sign Ecript Basic256</p> <p>Sign Basic256 Sha256</p> <p>Sign Basic256</p> <p>Sign Ecript Basic128Rsa15</p> <p>Sign Basic128 Rsa15</p> <p>None</p>
Certificate	drop-down list (defaults to tridium)	Selects the client's TLS certificate. The default self-signed certificate should be approved and used only until a signed certificate is available.
User Authentication Mode	drop-down list (defaults to User Name and Password)	<p>Configures the type of user authentication. For a successful connection, these settings must match the OPC UA server's <b>User Authentication Methods</b> settings.</p> <p>Anonymous</p> <p>User Name and Password</p>
User Name	text string	Configures the user's name. User authentication by User Name and Password requires this property be set. The username and password must be defined in the station's <b>UserService</b> .
Password	text string	Configures the user's password. User authentication by User Name and Password requires this property be set. The username and password must be defined in the station's <b>UserService</b> .
Server State	read-only	Reports the current condition of the server.
Server Current Time	read-only	Reports the current time as maintained in the server.
Server Start Time	read-only	Reports when the server came online.
Time Interval	hours minutes seconds (defaults to 30 seconds)	
Status Check Timeout	hours minutes seconds (defaults to 10 seconds)	
Server Info	additional properties	Displays read-only data read from the OPC UA server



Property	Value	Description
		configured for the device. These data are documented in a separate topic.
Alarm Ext	additional properties	Opens the Opc Ua Client Alarm Manager. A separate topic documents this component's properties.
Points	additional properties	Opens the Opc ua Client Point Manager that contains device point data. A separate topic documents this folder's related discovery preferences.
Histories	additional properties	Opens the <Opc Ua Client History Import Manager>. The Histories AX Property Sheet configures the driver's Retry Trigger. This typical set of properties are documented in the <i>Niagara Drivers Guide</i> .

Actions

- **Ping** sends a message to a network object (device, database, etc). The message provokes a response, which indicates the current state of the object.
- **Learn** discovers device objects.
- **Reset Comm** enables and disables the device (closes and reopens server connection).

Related tasks

- [Connecting to an OPC UA server](#)
- [Discovering OPC UA server points](#)

opcUaClient-OpcUaBuildInfo

This component is a child of each **OpcUaDevice** component. Each OPC UA device has a frozen slot named **Server Info (OpcUaBuildInfo)** on the **OpcUaDevice** component.

This component displays read-only data read from the OPC UA server and configured for the device.

Figure 11. OpcUaBuildInfo properties

Server Info (Opc Ua Build Info)	
Product Name	N4_OpcUaServer
Product Uri	urn:tridium.com:OPCUA:N4OpcUaServer
Manufacturer	Tridium
Software Version	4.3.58.16
Build Number	16
Build Date	05/10/17 18:46:26.2290000 GMT

Property	Value	Description
Produce Name	read-only	Reports the name of the OPC UA server.

Property	Value	Description
Product Uri	read-only	Reports the URI (Uniform Resource Identifier) for the server.
Manufacturer	read-only	Reports the name of the manufacturer of the server.
Software Version	read-only	Reports the version of software for the server.
Build Number	read-only	Reports a number for the server data type.
Build Date	read-only	Reports a date and time of the software build for the server.

Action

**Update** retrieves the device data from the server.

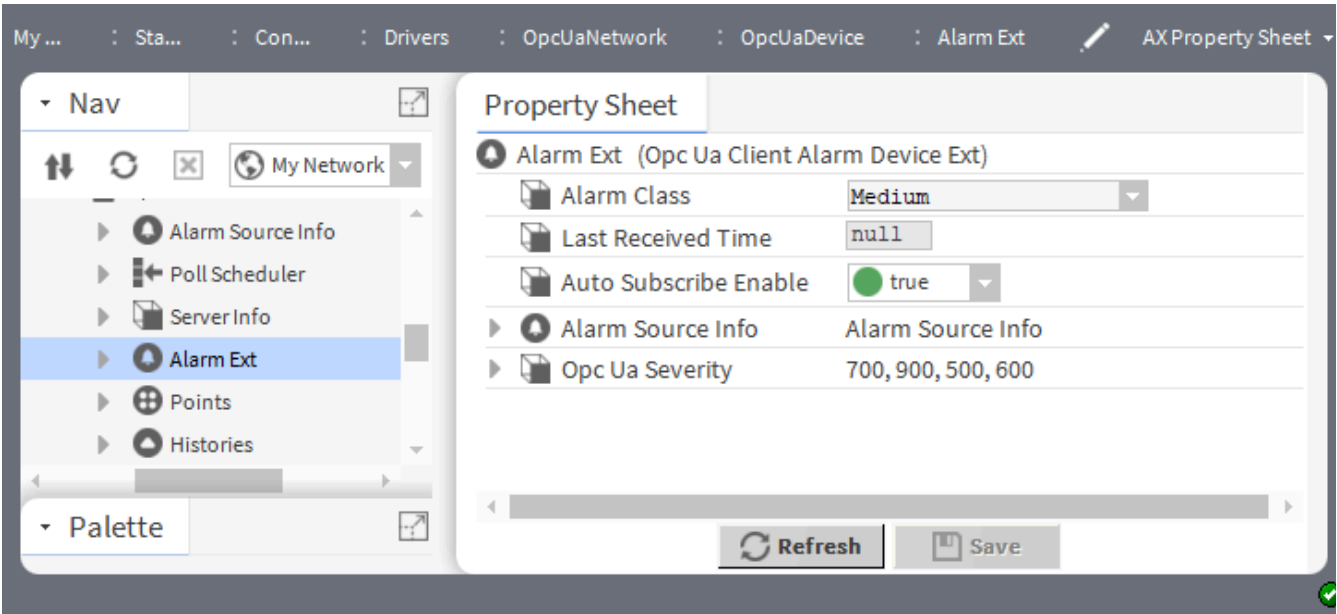
opcUaClient-OpcUaClientAlarmDeviceExt

This component is a child of each OpcUaDevice component. Each OPC UA device has a frozen slot named AlarmExt that is of type OpcUaClientAlarmDeviceExt.

The default view is the Opc Ua Client Alarm Manager.

This component provides the ability to subscribe to OPC UA alarm events. The view allows you to scan the connected OPC UA server for OPC UA variables that can generate an alarm.

**Figure 12.** OpcUaClientAlarmDeviceExt properties



To view these properties, expand **Config > Drivers > OpcUaNetwork > OpcUaDevice**, right-click **Alarm Ext** and

click **Views > AX Property Sheet**.

In addition to the standard properties that configure Alarm Source Info, these properties support the OPC UA client alarm device extension.

Property	Value	Description
Alarm Class	drop-down list (defaults to <code>Medium</code> )	<p>Specifies the alarm routing option for the component.</p> <p><code>Replace</code> provides a selection list of a local alarm classes, from which to select one to use for all alarms received from this device.</p> <p><code>Use Existing</code> routes alarms from this remote station to any matching alarm class, that is, one with an identical name as that in each alarm record. If the program finds no local matching alarm class, it uses the station's default alarm class.</p> <p><code>Prepend</code> adds leading text (as specified) to the incoming alarm class string, then routes it to any local matching alarm class in the station.</p> <p><code>Append</code> adds trailing text (as specified) to the incoming alarm class string, then routes it to any local matching alarm class in the station.</p>
Last Received Time	read-only	Reports the time the last alarm was received.
Auto Subscribe Enable	<code>true</code> (default) or <code>false</code>	Turns automatic device subscription on ( <code>true</code> ) and off ( <code>false</code> ).
Opc Ua Severity	read-only	Reports the configuration for the four severity settings. For more information, refer to <i><a href="#">opcUaServer-OpcUaAlarmRecipient</a></i> .

**Related tasks**

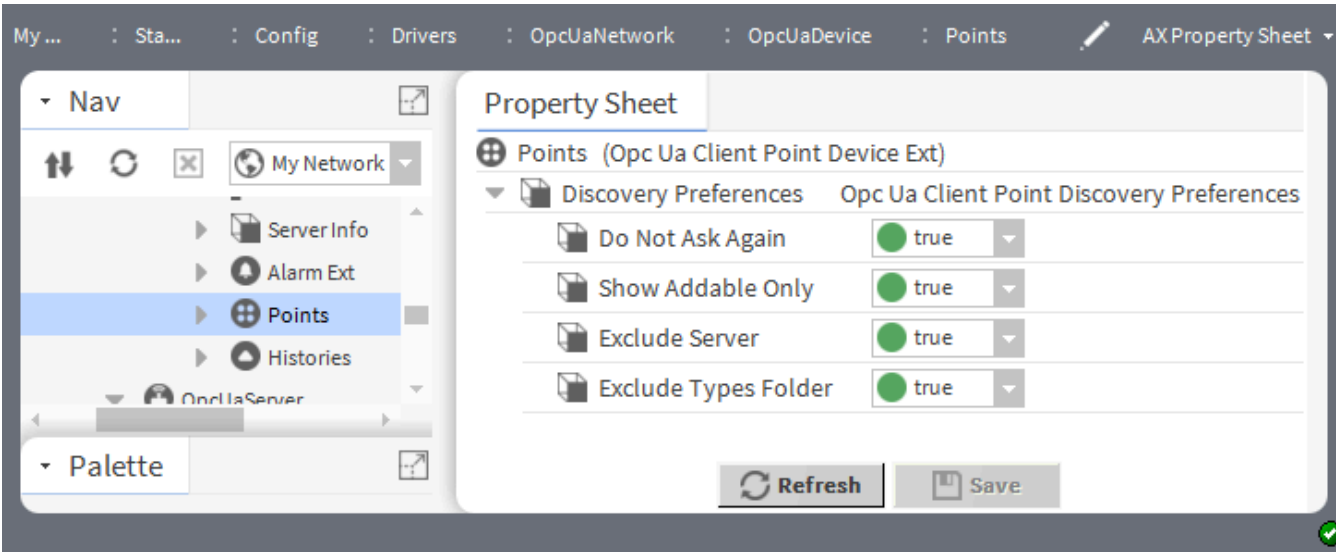
- [Subscribing for OPC UA alarm events](#)

**opcUaClient-OpcUaClientPointDeviceExt**

This component is the OPC UA implementation of `PointDeviceExt`, a frozen device extension under every OPC UA client device. Its primary view is the **Opc Ua Client Point Manager**.

The OpcUaClientPointDeviceExt (Points) is also available in the opcUaClient palette.

**Figure 13.** OpcUaClientPointDeviceExt properties



To access these properties, right-click **Points**, click **Views > AX Property Sheet** and expand **Discovery Preferences**.

Property	Value	Description
Do Not Ask Again	true (default) or false	<p>Controls when the discovery prompt displays.</p> <p>true displays the prompt when you click the Discover button on the Device Manager view.</p> <p>false displays the prompt before the system initiates the search.</p>
Show Addable Only	true (default) or false	<p>Controls the display of points that cannot be added to the database.</p> <p>true excludes the points that cannot be added to the database.</p> <p>false displays the points that cannot be added to the database.</p>
Exclude Server	true (default) or false	<p>Controls the display of points associated with the server.</p>

Property	Value	Description
		<code>true</code> excludes the points associated with the server.
		<code>false</code> includes the points associated with the server.
Exclude Types Folder	<code>true (default) or false</code>	Controls the display of points in the Types folder.
		<code>true</code> excludes the points in the Types folder.
		<code>false</code> includes the points in the Types folder.

opcUaClient-OpcUaDeviceFolder

This component is the OPC UA client implementation of a folder under an OpcUaNetwork.

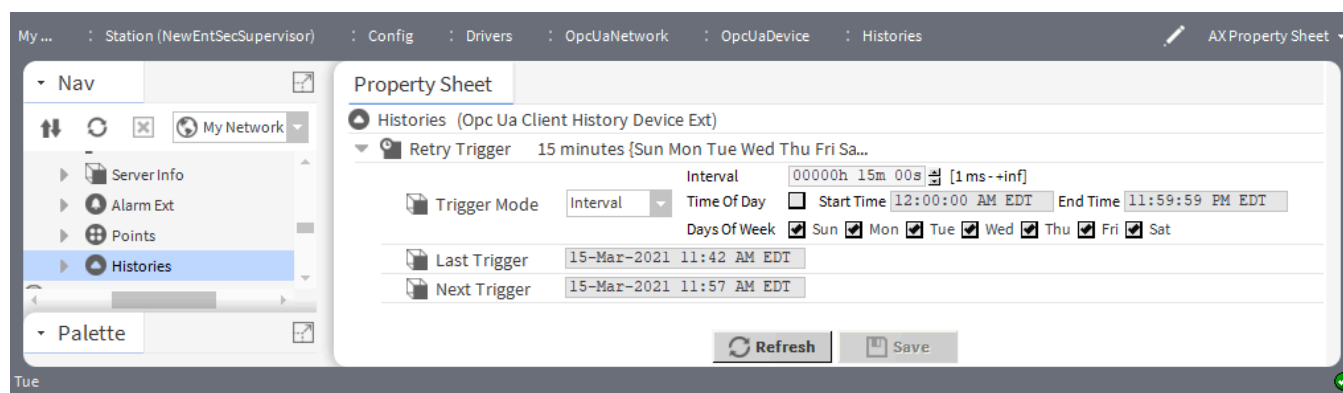
Typically, you add such folders using the **New Folder** button in the **Opc Ua Client Device Manager** view of the OPC UA network. The component is also available in the opcUaClient palette.

opcUaClient-OpcUaClientHistoryDeviceExt

This component, whose default name is Histories, is a frozen device extension on the OPC UA client component. It imports historical data from the OPC UA server into the station’s history space.

You use this components **Opc Ua Client History Import Manager** to add OPC UA client historyImport descriptors.

Figure 14. OpcUaClientHistoryDeviceExt properties



For information about **Retry Trigger** properties, refer to the *Niagara Drivers Guide*.

Action

**Retry** downloads histories again.

Related tasks

- [Adding points containing OPC UA histories](#)

opcUaClient-OpcUaClientPointFolder

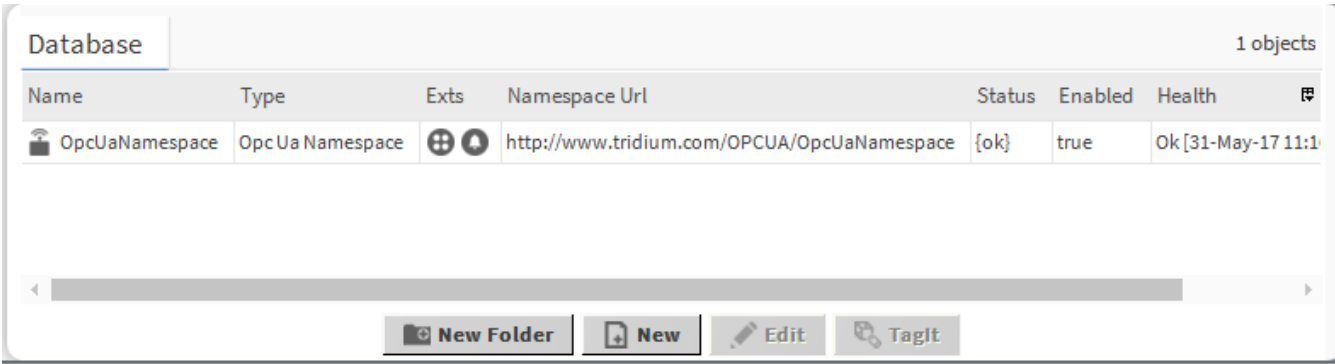
This component is the OPC UA cClient implementation of a folder under an OPC UA device **Points** extension. You add such folders using the **New Folder** button in this component’s **Opc Ua Client Point Manager**. This folder is also available in the opcUaClient palette.

This folder is also available in the opcUaClient palette.

opcUaServer-OpcUaServerDeviceManager

This manager is the primary view for the **OpcUaServer** component. It manages OPC UA namespace components in the station.

**Figure 15.** Opc Ua Server Device Manager with added OpcUaNamespace



To view, either double-click the **OpcUaServer** or right-click the OpcUaServer and select **Views > Opc Ua Server Device Manager**.

Column	Description
Name	Displays the name of the namespace.
Type	Displays the type of the namespace.
Exts	Displays the device extension’s hyperlinks, including: Points, Alarms, Schedules, Trend Logs and Config.
Namespace Uri	Displays the namespace Uri.
Status	Reports the current condition of the entity as of the last refresh: {alarm}, {disabled}, {down}, {fault}, {ok}, {stale}, {unackedAlarm}
Enabled	Indicates if the network, device, point or component is active or inactive.
Health	Displays the current status of the device.

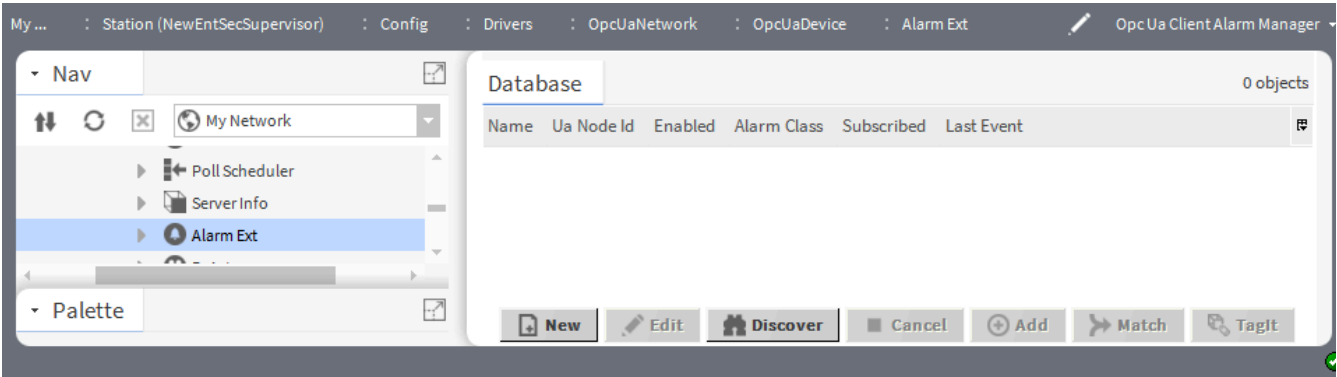
Buttons

- **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.
- **TagIt** associates metadata, such as location or unique configuration with the object.

### Opc Ua Server Alarm Manager

This manager is the default view of the server alarms device extension (`OpcUaServerAlarmDeviceExt` as well as the client alarms device extension (`OpcUaClientAlarmDeviceExt`).

**Figure 16.** Opc Ua Client Alarm Manager



To view, double-click the Alarms extension, or right-click and select **Views > Opc Ua Server Alarm Manager**.

Column	Description
Name	Reports the name of the entity or logical grouping.
Ua Node Id	Displays the ID of UA Node.
Enabled	Indicates if the network, device, point or component is active or inactive.
Alarm Class	Displays the alarm routing options and priorities.
Subscribed	Displays whether the device is subscriber or not.
Last Event	Displays the summary of the last alarm event received from the OPC UA server for the specified OPC UA node.

#### 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.
- **TagIt** associates metadata, such as location or unique configuration with the object.

### Opc Ua Server Point Manager

This manager is the default view for the `OpcUaServerPointDeviceExt` (Points container) under an OPC UA namespace. This is also the default view for any `OpcUaServerPointFolder` under the **Points** container of an OPC UA namespace.

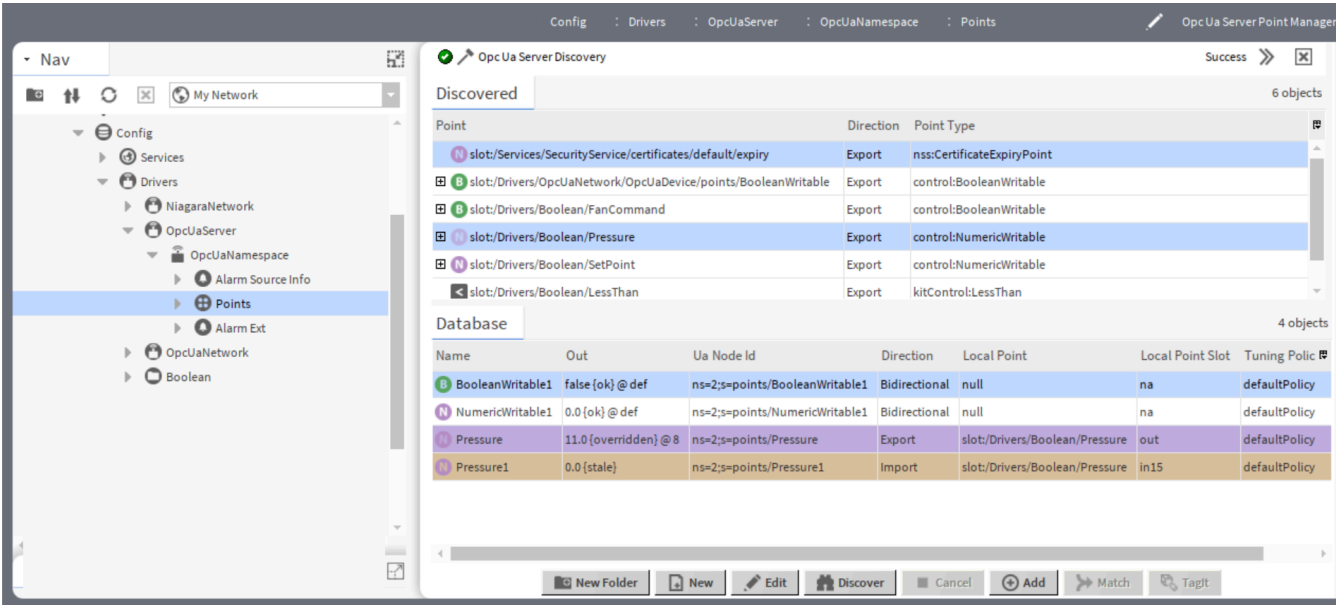
#### Bi-directional points

Starting in Niagara 4.15, the OPC UA Server supports bi-directional communication. One point is responsible for sending data from the OPC UA Server, while another point is required to receive data from the client. With this enhancement, a single point will manage incoming and outgoing data values.

Two different ways to add the bi-directional points in the Opc Ua Server Point Manager

- While adding the bidirectional points from the **Discovered** pane, the import and export points are linked to the local points
- Another way is adding the bi-directional points in the **Database** pane, by clicking through the **New** button. These points are not linked to any local points.

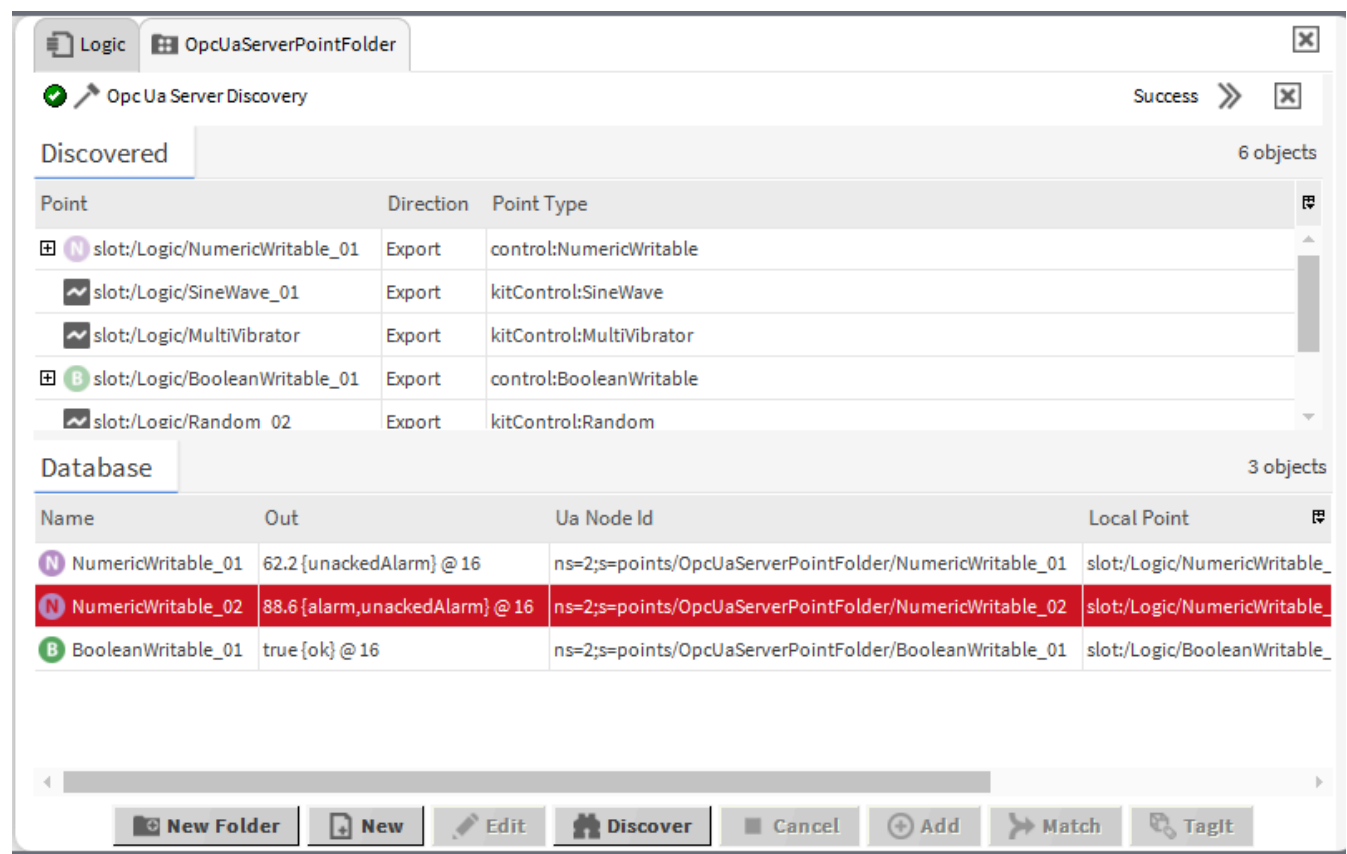
Figure 17. Bi-directional points



Use this view to discover server points that exist in the station, which can be mapped to the **OpcUaServerPointFolder**. Such server points are visible to a connected OPC UA client. Also, server points provide live and historical data as well as alarm events.



Figure 18. Opc Ua Server Point Manager view with discovered points added to station database



To view, right-click a `OpcUaServerPointDeviceExt` or `OpcUaServerPointFolder` and select **Views > Opc Ua Server Point Manager**

Column	Description
Point	Display the details of points.
Direction	Displays the direction value.
Point Type	Displays the type of the point added.

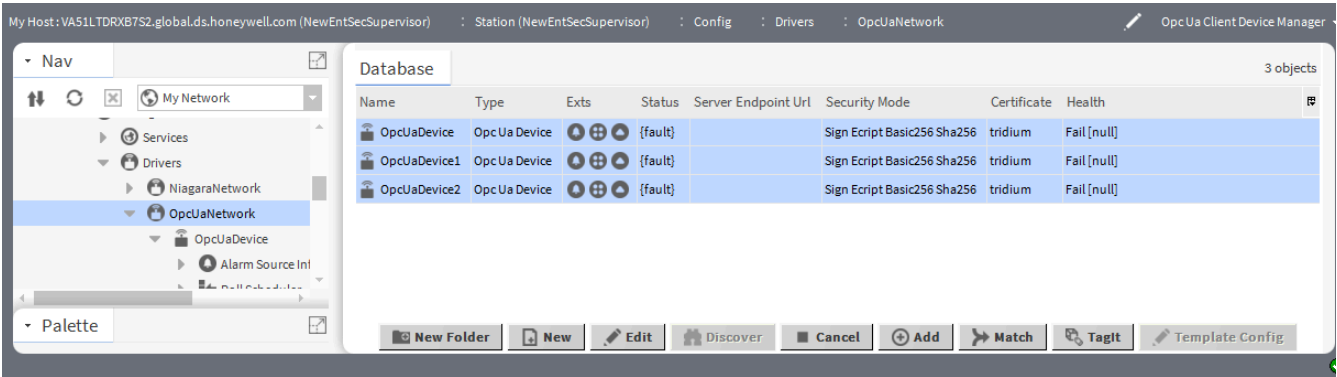
Buttons

- **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.

Opc Ua Client Device Manager

The **Opc Ua Device Manager** is the default view of a **OpcUaNetwork**, which helps to access OPC UA device components.

**Figure 19.** Opc Ua Client Device Manager



To view, right-click a **OpcUaNetwork** and select **Views > Opc Ua Client Device Manager**.

Column	Description
Name	Reports the name of the entity or logical grouping.
Type	Displays the type of device.
Exts	Displays the device extension’s hyperlinks, including: Points, Alarms, Schedules, Trend Logs and Config.
Status	Reports the current condition of the entity as of the last refresh: {alarm}, {disabled}, {down}, {fault}, {ok}, {stale}, {unackedAlarm}
Server Endpoint Url	Displays the connection address (URL) for the Opc Ua Server.
Security Mode	Displays the security applied to message exchange during a session.
Certificate	Displays the selected certificate.
Enabled	Indicates if the network, device, point or component is active or inactive.
Health	Reports the status of the network, device or component. This advisory information, including a time stamp, can help you recognize and troubleshoot problems but it provides no direct management controls.  The <i>Niagara Drivers Guide</i> documents the these properties.

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.
- **Template Config** accesses the station template that defines configuration options. You would select a template to set up the device with pre-configured properties.

Opc Ua Client Alarm Manager

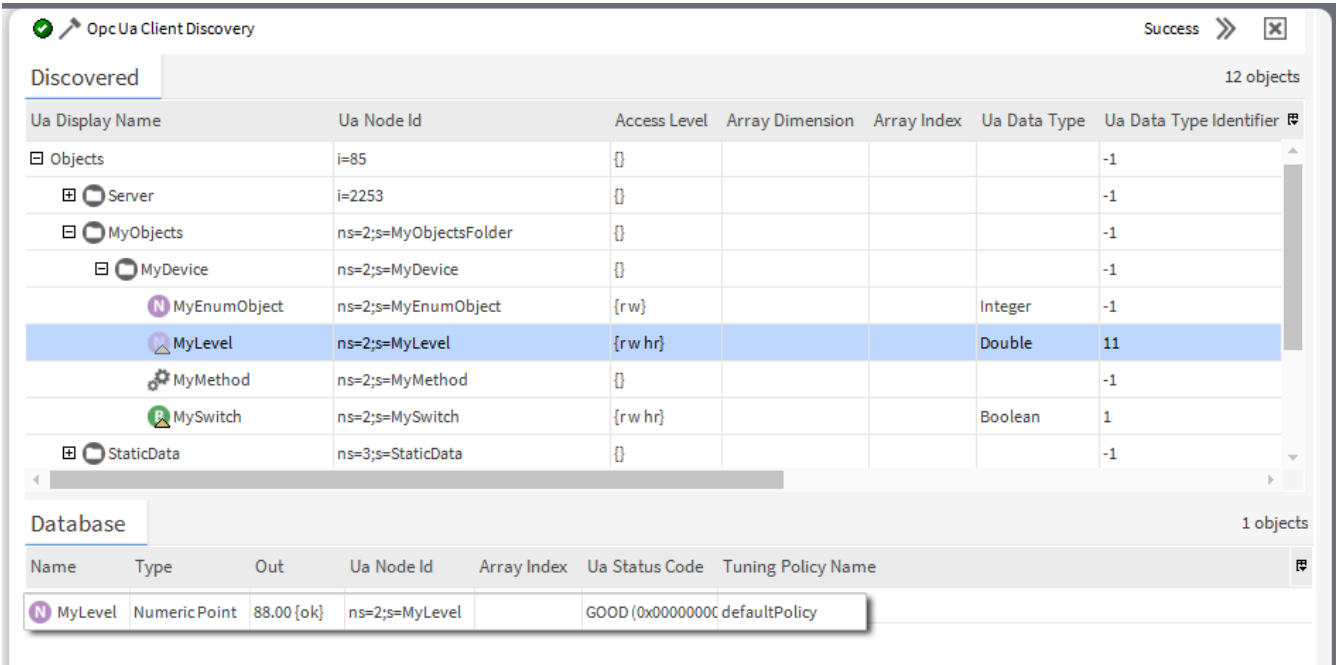
This manager is the default view of the client alarms device extension (**OpcUaClientAlarmDeviceExt**) as well as the server alarms device extension (**OpcUaServerAlarmDeviceExt**).

This view allows you to scan the OPC UA server to discover and subscribe to OPC UA alarm events. For more information, refer to “Opc Ua Server Alarm Manager.”

Opc Ua Client Point Manager

This manager is the default view for the **OpcUaClientPointDeviceExt** (Points container) under an OPC UA device. This is also the default view for any **OpcUaClientPointFolder** under the **Points** container of an OPC UA Device. Use this view to discover available points and add them to your station database.

**Figure 20.** Opc Ua Client Point Manager view with selected discovered points added to station database



To view, right-click a **OpcUaClientPointDeviceExt** or **OpcUaClientPointFolder** and select **Views > Opc Ua Client Point Manager**.

Column	Description
Ua Display Name	Reports the name of the entity or logical grouping.
Ua Node Id	Displays the ID of the UA node.
Access Level	Indicates the value of a variable can be accessed (read/write) and if it contains current and/or historic data.
Array	Displays the maximum supported length of each dimension. If the maximum is unknown the value shall be 0.

Column	Description
Dimension	
Array Index	Displays the integer marker of the index of a particular value in an array type Node.
Ua Data Type	Displays the <b>DataType</b> attribute to a node of the data type <b>NodeClass</b> in the Server.
Ua Data Type identifier	Displays the numeric identifier from the Data Type Node Id.

Buttons

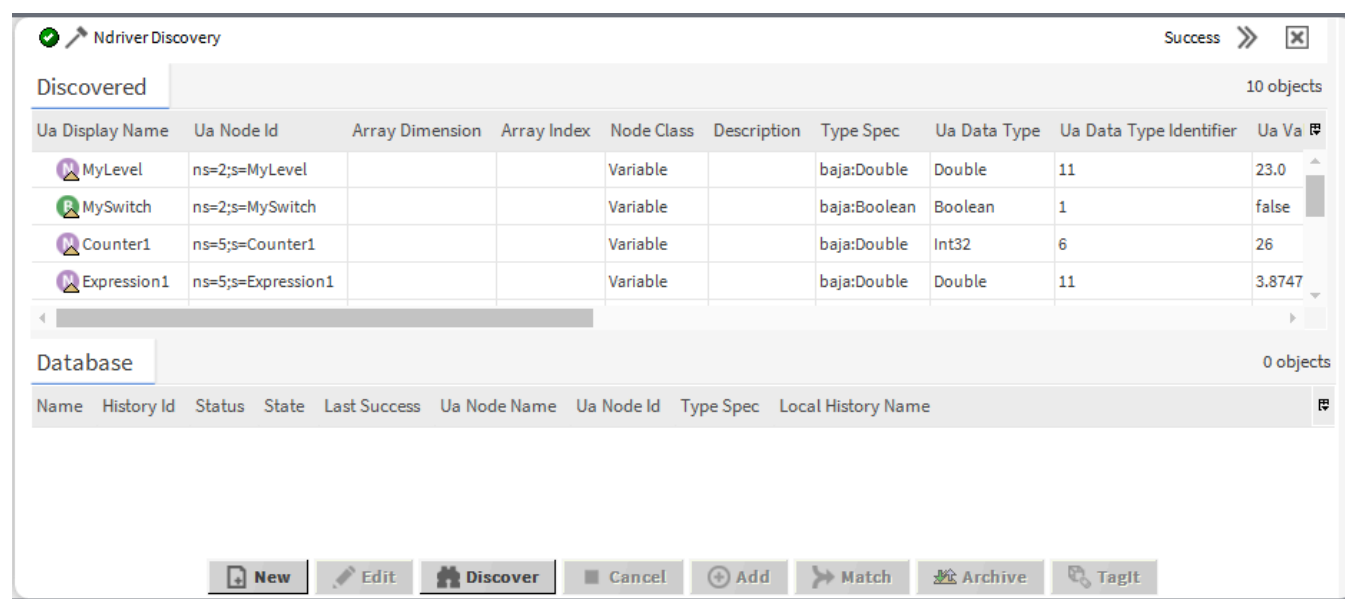
- **New Folder** creates a new folder for devices. Each such folder provides its own set of manager views.
- **New** creates a new device record in the database.
- **Edit** opens the device’s database record for updating.
- **Discover** runs a discover job to locate installed devices, which appear in 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.

Opc Ua Client History Import Manager

This manager is the default view for the **OpcUaClientHistoryDeviceExt** component, which discovers and imports OPC UA histories without the creation of a control point.

To view, right-click a **OpcUaClientHistoryDeviceExt** and select **Views > OpcUa Client History Import Manager**.

**Figure 21.** OpcUa Client History Import Manager lists discovered points with histories



Column	Description
Ua Display Name	Reports the name of the entity or logical grouping.

Column	Description
Ua Node Id	Displays the ID of the UA node.
Array Dimension	Displays the maximum supported length of each dimension. If the maximum is unknown the value shall be 0.
Array Index	Displays the integer marker of the index of a particular value in an array type Node.
Node Class	Displays the class of a Node in an AddressSpace.
Description	Displays the information in the description attribute of a Node.
Type Spec	Displays the Niagara Type Specification given to an OPCUA objects based on the type of data it encodes.
Ua Data Type	Displays the <b>DataType</b> attribute to a node of the data type <b>NodeClass</b> in the Server.
Ua Data Type Identifier	Displays the numeric identifier from the Data Type Node Id.

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.
- **TagIt** associates metadata, such as location or unique configuration with the object.
- **Archive** archives the data.



# Chapter 5. Glossary

The following glossary entries relate specifically to the topics that are included as part of this document.

To find more glossary terms and definitions refer to glossaries in other individual documents.

## Alphabetical listing

### namespace

A container for node IDs (points in the framework). The OPC foundation specifies a predefined namespace with index 0. Many more namespaces belong to specific OPC UA specifications, for example, DI and PLCopen. Every OPC UA specification defines its own node IDs. (from <https://stackoverflow.com>).

The **OpcUaNamespace** component under the **OpcUaNetwork** node in the Nav tree configures the namespace. The **OpcUaServerPointDeviceExt** component uniquely identifies each specific node.

### severity

In OPC UA, alarm severity is specified by an integer range between 1 and 1000. An alarm severity map assigns a level of urgency to each alarm event. An event that transitions:

- To Offnormal is urgent. By default it receives a severity index of 700.
- To Fault is the most urgent. By default it receives a severity index of 900.
- To Normal is the least urgent. By default it receives a severity index of 500.
- To Alert is moderately urgent. By default it receives a severity index of 600.

You can configure the severity index for each level of urgency.