

Technical Document

Niagara Cloud Service Guide

January 8, 2025



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

About this guide	7
Document change log	7
Related documentation	8
Chapter 1. Getting started	9
Overview	9
Requirements	9
NiagaraCloudService	12
Installing software modules	13
Setting up device Internet access	14
Niagara Remote	15
Chapter 2. Install and configure	17
Adding the NiagaraCloudService	17
Registering a device	17
Re-registering a device	20
Re-registering from a station	21
Niagara Cloud Service configuration	22
Alarms channel	23
Cloud Link Alarm Recipient	23
Messaging channel	24
Heartbeat channel	24
Model channel	24
Histories channel	25
Cloud History Export Manager	25
Exporting Histories data	26
Backup channel	27
Chapter 3. Data management	29
Cloud Archive History Provider	29
Assigning cloud Ids to station components using Cloud Id Manager	29
Excluding components and histories from cloud upload	30
Cloud upload exclusions using nc:excluded tag	32
Finding excluded components and histories	32
Model uploads to the cloud	33
Exporting Model data to the cloud	33
History uploads to the cloud	34
Re-including excluded components and histories	35
Re-including components	35
Re-including histories	36
Backup Channel	37
Uploading station backups to the cloud	37
Creating a cloud backup	37
Restoring a station for a controller	38

Restoring a Supervisor station	39
Restoring Supervisor backup to the same computer	39
Restoring Supervisor backup to a new computer	39
Chapter 4. Components and other references	41
cloudLink-CloudArchiveHistoryProvider	41
cloudLink-HistoryArchiveCache	42
cloudLink-CertTrustMapping	43
cloudLink-CloudAuthenticationScheme	44
cloudLink-CloudConnectionService	44
cloudLink-CloudIdManager	47
cloudLink-CloudLinkAlarmRecipient	49
cloudLink-CloudTrustManager	50
cloudLink-FederatedIdentityAuthenticator	50
cloudLinkForge-RpkAuthenticator	52
cloudLink-AmqpTransport	54
cloudLink-HttpTransport	56
cloudLink-NiagaraRemoteTransport	57
cloudLink-AlarmsChannel	58
cloudLinkForge-ForgeAmqpAlarmChannelConfig	59
cloudLink-BackupChannel	60
cloudLinkNcs-NcsBackupChannelConfig	60
cloudLink-CloudBackupPolicyContainer	61
cloudLink-CommandsChannel	65
cloudLink-ForgeAmqpCommandChannelConfig	66
cloudLink-EventsChannel	68
cloudLinkForge-ForgeAmqpEventChannelConfig	68
cloudLink-HeartbeatChannel	70
cloudLinkForge-ForgeAmqpHeartbeatChannelConfig	70
cloudLink-HistoriesChannel	71
cloudLinkForge-ForgeHistoryChannelConfig	72
cloudLink-MessagingChannel	75
cloudLinkForge-ForgeMessagingChannelConfig	75
cloudLink-ModelChannel	77
cloudLinkForge-ForgeModelChannelConfig	78
cloudLink-CloudIdExportPolicy	79
cloudLink-PointsChannel	80
cloudLinkForge-ForgeAmqpPointChannelConfig	83
cloudLink-PointsCovExportPolicy	84
cloudLink-JwksTrustMapping	86
cloudLink-RoleMappings	87
cloudLink-UserMappings	88
cloudLink-CloudHistoryExportConfigContainer	89
cloudLink-CloudHistoryExportManager	90
cloudLink-CloudHistoryAutoExportConfig	91

	Event messages and system commands	92
Chapter 5. Troubleshooting		99
	When an incident occurs	99
	What logs to collect	99
	What files to collect	101
	Authenticator information	101
	Network sanity checks	101
	Checking endpoint availability	102
	Registration issues	103
	Device Registration view does not load	103
	Cannot reach device registration web service	104
	Connection issues	105
	Cannot connect to the cloud	105
	Authenticator keys are lost	106
	Niagara Cloud Service does not attempt connection	108
	Proxy server preventing connection	108
	AMQP blocked	109
	Cloud backup channel issues	110
	Setting cloud backup size in Workbench	111
	Workaround for Directory Ord Chooser issue	112
Chapter 6. Tuning		115
	NiagaraCloudService	115
	Authenticators	115
	Transports	115
	Channels	116
	Commands	116
	Heartbeat	116
	Histories	117
	Messaging	117
	Model	117
	Points	118
Chapter 7. Appendix		119
	Configuring the station to receive commands	119
	Adding a CertTrustMapping	119
	Adding a JwksTrustMapping	120
	About Role Mappings	121
	Configuring Role Mappings (optional)	122
Chapter 8. Glossary		125

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 how to use the cloudLink module and the integrated NiagaraCloudService. You learn about the functionality to send and receive data to and from the cloud as well as to push data from a collection of network capable devices (thermostats, HVAC units, Niagara stations while they can be securely managed and controlled from the cloud.

Document change log

Updates (changes and additions) to this document are listed below.

January 8, 2025

- Added "Workaround for Directory Ord Chooser issue" topic in the "Troubleshooting" chapter.

November 5, 2024

Updated "Registration Code" and "Registration URL" properties in "cloudLink-FederatedIdentityAuthenticator".

August 26, 2024

- Update - Renaming: The product name "Niagara Cloud Service" is replacing "CloudLink".
- Update - Renaming: The component name "NiagaraCloudService" is replacing "CloudConnectionService".

July 24, 2024

- Added "System Command Configuration" section to the "Event messages and system commands" chapter.
- Added "Restoring a Supervisor station" topic to the "CloudLink Backup Channel" chapter.
- Updated "Histories channel" and "cloudLink-HistoriesChannel" topics to include new functionality.

January 31, 2024

- Updated the "Software modules" section in the "Requirements" chapter.
- Added "Cloud Link Alarm Recipient" topic to "CloudLink configuration" chapter.

October 23, 2023

- Added "Niagara Remote" topic and "cloudLink-NiagaraRemoteTransport" component (available with Niagara 4.10u7 and Niagara 4.13).
- Added "Assigning cloud Ids to station components using Cloud Id Manager" chapter.
- Added "cloudLink-CloudIdManager" component.
- Added a warning to "History uploads to the cloud" chapter to avoid the creation of duplicate data.

May 3, 2023

- Added "Cloud History Export Manager" chapter.
- Added "cloudLink-ComponentExportPolicy", "cloudLink-CloudHistoryAutoExportConfig" components

and "cloudlink-CloudHistoryExportManager" view.

- Added "Niagara Cloud Service Backup Channel" to the "Data management" chapter.
- Added "cloudLink-BackupChannel", "cloudLinkForge-ForgeBackupChannelConfig" and "cloudLinkForge-CloudBackupPolicyContainer" components to the "Components and other references" chapter.

January 20, 2023

- Removed "Security Recommendations" chapter.

October 20, 2022

- Updated "cloudLink-PointsChannel" with "No query is present by default in Point COV policy."
- Added sections on "Cloud upload exclusions using nc:excluded tag" in the "Data management" chapter.
- Added "cloudLink-CloudHistoryExportConfigContainer" component and "Cloud History Export Manager" view.
- Added "Cloud Archive History Provider" to the "Data management" chapter, and included two new components: "cloudLink-CloudArchiveHistoryProvider" and "cloudLink-ArchiveHistoryCache" in the "Components and other references" chapter.

July 25, 2022

- Added the following properties to "cloudLinkForge-ForgeHistoryChannelConfig" container: "Backfill Record Threshold", "Backfill Reconnect Min Time", and "Onboarding".
- Added "Activate Channel" action to "cloudLink-HistoriesChannel" component.
- Added the "FederatedIdentityAuthenticator" component to the "Components and other references" chapter.

May 11, 2021

- Initial release document.

Related documentation

Additional information is available in the following documents.

- *Niagara Cloud Suite (NCS) Partner Guide*
- *Niagara Cloud Suite (NCS) Customer Guide*
- *Niagara Data Service (NDS) API Guide*

Chapter 1. Getting started

This section provides an overview of Niagara Cloud Service. It describes license and software requirements, as well as procedures to configure the remote station for cloud integration and to register the device.

To use Niagara Cloud Service, you must have a properly licensed Workbench PC. Additionally, you must have a Niagara station that is properly licensed for Niagara Cloud Service. All systems must be running Niagara 4.10 or later, and should be on the latest available update for that Niagara version. Internet access is required for all systems as well.

Overview

The cloudLink module provides the functionality to send and receive data to and from the cloud. The main purpose of the module is to provide a mechanism to push data from a collection of network capable devices (thermostats, HVAC units, NiagaraStations) such that they can be securely managed and controlled from the cloud.

This module acts as an adapter, bridging Niagara's internal data to a cloud-specific format. To connect to Niagara Cloud Suite cloud platform, you need the NiagaraCloudService from the cloudLinkNcs palette. This service provides a set of authenticators, transports, and data channels that enable communication with the cloud platform.

NOTE: Only the Niagara Cloud Suite cloud platform is supported at this time. However, the cloudLink module has been created with extensibility in mind. In cases where you want to connect a single station to multiple cloud platforms (such as Azure, AWS or Google), you would need one instance of the Niagara Cloud Service per cloud platform.

For a specific cloud platform with known capabilities and requirements, some parts of the service are fixed, and configured by the choice of the palette. For example, the cloudLinkNcs palette contains a NiagaraCloudService that is pre-configured to communicate with the Niagara Cloud Suite. Once installed, the NiagaraCloudService component provides device registration and a secure connection to that platform.

Upon device registration, some components are automatically added from the Niagara Cloud Suite provisioning cloud service based on your subscriptions.

The station running Niagara Cloud Service contains references to the devices on the network. This station also includes the Niagara Cloud Service component that has an established connection to the Niagara Cloud Suite platform. Data is sent from the station to this cloud platform. As data enters the platform, it is stored in databases and/or Event Hubs. Separately, there are cloud-based web applications, which consume the data, retrieve and read data from either the Event Hub or REST APIs. Additionally, data can flow out of the cloud platform via a mechanism called System Commands. These are executed on the station running Niagara Cloud Service and have the ability to read, write, or edit values on the station depending on the correct security configurations.

Requirements

This topic describes the platform, licensing, and software requirements for using Niagara Cloud Service and the Niagara Cloud Management Portal.

Platform and application requirements

- The NiagaraCloudService requires a compatible Niagara version.
- A Workbench connection is required to install the cloudLink modules and configure the NiagaraCloudService.
- A browser is required to access the Niagara Cloud Management Portal.

License requirements

- A cloudLink license must be enabled on the host.
- You must have an active SMA (Software Maintenance Agreement).
- An active subscription to one or more Niagara Cloud services.

Niagara Community credentials

To register a device using the Niagara Cloud Management Portal, you must be a registered user of the Niagara Community and your Partner Admin must have given you access to a particular customer. A user without access will be redirected to Niagara Community.

Software modules

NCS requires a core set of modules. Some modules are optional. The following table shows the required and optional modules that are needed for each version of Niagara.

NOTE: In the Software Manager, carefully select the correct modules based on the table below, especially if you wish to perform a Niagara upgrade because some module names have changed.

Selecting the “select first” modules, cloudLinkNcs-rt.jar also automatically selects some of the other modules in the table, but not all of them. As a result, you will need to install some manually.

NOTE: When upgrading, ensure that you delete the unused modules so that these unnecessary modules will not be included in the station backups. This would make a restore from backup difficult as the old unnecessary modules must be obtained for a restore to work.

Table 1. Niagara versions 4.10.6, 4.12.2, 4.13.0

Module	Required	Software Installation	Upgrade to: 4.10.7+, 4.13.2+, 4.14+	Notes
cloudLink-rt.jar	yes	automatic		
cloudLink-ux.jar	yes	manually select		
cloudLinkForge-ux.jar	yes	manually select		
okhttp-rt.jar	yes	automatic		Niagara version 4.10.6+ only

Table 2. Niagara versions 4.10.7+, 4.13.2+, 4.14+

Module	Required	Software Installation	Upgrade from: 4.10.6, 4.12.2, 4.13.0	Notes
cloudLink-rt.jar	yes	automatic		
cloudLink-ux.jar	yes	manual		
cloudLinkAzure-rt.jar	yes	automatic		

Module	Required	Software Installation	Upgrade from: 4.10.6, 4.12.2, 4.13.0	Notes
cloudLinkForge-ux.jar	yes	manual		
cloudLinkNcs-rt.jar	yes	select first	When upgrading from Niagara 4.10.6, 4.12.2, 4.13.0, this module replaces cloudLinkNds-rt.jar, which should be manually deleted.	Use the palette in this module to install the NiagaraCloudService .
clUtils-rt.jar	yes	automatic	When upgrading from Niagara 4.10.6, 4.12.2, 4.13.0, this module replaces modelDiscovery-rt.jar, which should be manually deleted.	
clUtilsBacnet-rt.jar		manual	When upgrading from Niagara 4.10.6, 4.12.2, 4.13.0, this module replaces modelDiscoveryBacnet-rt.jar, which should be manually deleted.	Install this module for Cloud support of Bacnet network devices.
clUtilsNiagara-rt.jar		manual	When upgrading from Niagara 4.10.6, 4.12.2, 4.13.0, this module replaces modelDiscoveryNiagara-rt.jar which should be manually deleted.	Install this module for Cloud support of Niagara network devices.
okhttp-rt.jar	yes	automatic		Niagara version 4.10.7+ only

Internet access

Internet access is required for all stations and clients. For more information, refer to [Setting up device Internet access](#).

Security precautions

Station security is a must-have for all Niagara applications. Adequate security involves these best practices:

- Restricted physical access to each device (controller) and computer: do not make it easy for unauthorized individuals to access your devices. Users should be trained not to walk away from the PC while a sensitive view is open for others to see. Any user who has access to a dashboard should be configured for auto-logout.
- User authentication with strong passwords: a minimum of 10 characters that include numbers, upper and lower-case letters and special characters (! @ # \$ %); do not reuse passwords; establish a password policy that includes periodic password changes.
- Limited role assignments that configure access permissions: giving any user broad permissions on the **RoleService** is risky. A user with admin write permissions can create, edit, rename or delete any role. Such permission should be limited to only appropriately-authorized users.
- Client/server authenticated TLS communication at all levels: internal Foxs communication, HTTPS network communication, and external links to the Internet using VPN. TLS certificates must be signed by a third-party Certificate Authority. Self-signed certificates do not provide communication authentication.
- Components that support strong passwords, encryption, and authentication: replace older components, such as cameras, that do not support secure communication with components that support TLS.
- Encrypted data transmission over all communication channels.
- Signed program code (all Niagara modules are signed). Third-party modules should also be signed. Do not sign a module on behalf of a third party except as a last option, and then only if you trust the module authors.
- Separate locations for the Daemon User Home and Workbench User Home.

NiagaraCloudService

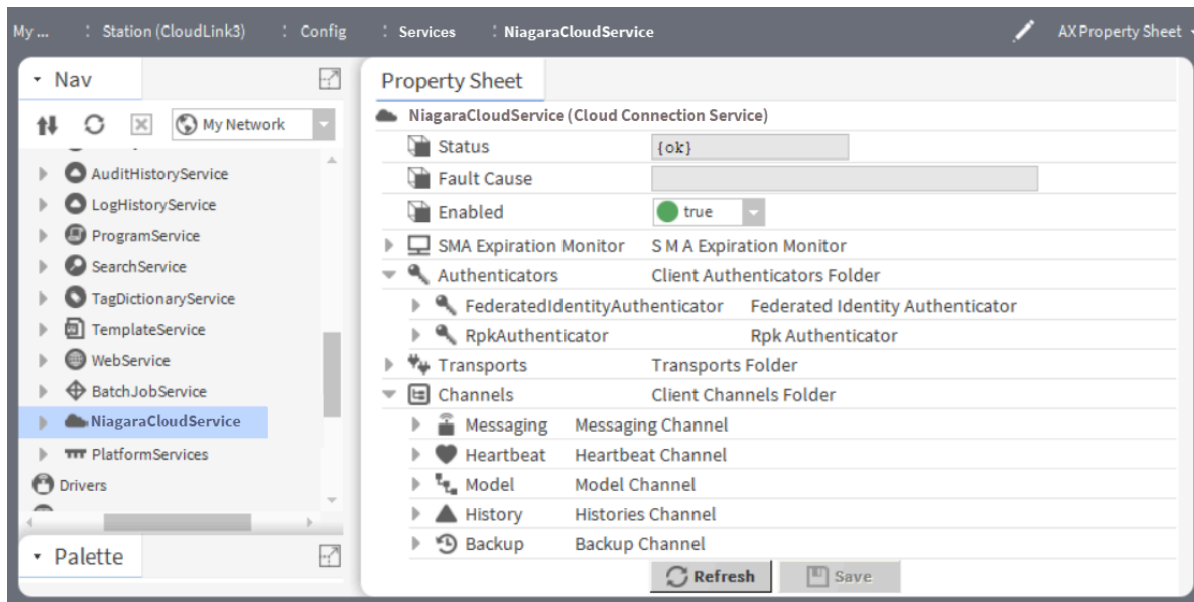
The cloudLinkNcs palette provides the **NiagaraCloudService**, which can send and receive data to and from the cloud. The main purpose of the service is to provide a mechanism to push data to the cloud platform from a collection of network capable devices (for example, thermostats, HVAC units, NiagaraStations) for secure management.

Niagara Cloud Service was designed to minimize the number of required configuration options to optimally set up the service. For this reason, there may not be many things to change for most users. This tuning guide touches on some of the configuration options, and discusses when you might need to adjust them.

NiagaraCloudService has three basic partitions:

- Authenticators
- Transports
- Channels

Figure 1. NiagaraCloudService properties for Authenticators, Transports, and Channels



Authenticators

This folder contains the mechanisms for authenticating to a specific cloud platform. For example, the `FederatedIdentityAuthenticator` authenticates to the Niagara Cloud Suite. Additionally, a developer may create their own authenticator, configured to authenticate to a different cloud platform. Furthermore, each authenticator has a unique authenticator Id. It is used by the transport to obtain the necessary authentication to communicate, such as a bearer token or connection string.

The specific authenticator is responsible for providing the means of authentication. For example, if a file were to be uploaded via HTTP request, the authenticator would need to provide a valid token to the HTTP request.

Transports

This folder contains the mechanisms for transporting data. Such transports can include:

- HTTP transport (the mechanism for sending data via HTTP)
- AMQP transport (the mechanism for sending data via AMQP)

The different transports contain properties that resemble common tuning parameters, like timeouts, retries,

and limits. However, in most cases, you should not have to adjust these properties as the defaults have been selected to optimize the bandwidth usage.

Channels

This folder contains the independent features that enable the **NiagaraCloudService** to function once provisioning is completed. The channels come pre-configured with default settings. Some channels also include default export policies, which are used to control the data and frequency at which the data is sent to the cloud.

- Alarms *
- Commands *
- Events
- Messaging (installed by default)
- Heartbeat *
- Model *
- Histories *
- Backup *
- Points

* Automatically added for Niagara Cloud Suite after device registration if the device is subscribed.

A channel queue holds messages relevant to the channel. Each channel handles communication for one application layer, point or history data. To communicate with the outside world, the channel uses pointers to the specific transport and authenticator it needs. For example, the Heartbeat channel maintains connectivity with the cloud platform by periodically sending heartbeat messages via AMQP. As such, the configuration for the Heartbeat channel in Niagara Cloud Suite has entries for a transport type of AMQP and an authenticator ID of RpkAuthenticator. This means that when the Heartbeat channel needs to send a heartbeat message it does the following:

1. Queries the specific authenticator for the needed credentials.
2. Creates an AMQP message.
3. Sends the message to the AMQP transport along with the associated details to ensure the message can be sent.

Each channel has a Channel Config object, where most tuning parameters are contained. As with the rest of Niagara Cloud Service, there should not be much tuning needed because the defaults are chosen to optimize communication for most scenarios. There are a few common configuration options for most channels.

Installing software modules

If the cloudLink modules are not part of your Niagara image, use this procedure to install them. You can skip this procedure in cases where Niagara Cloud Service is packaged inside a docker image. In that scenario, the act of creating the docker image handles downloading and installing Niagara Cloud Service.

Prerequisites:

You are working in Workbench and are connected to a station. The station is connected to the Internet. You have a user account on the Niagara Community Software portal.

Only the system being registered with the cloud needs the cloudLink modules. Subordinate stations do not need the modules. If a subordinate station itself needs to communicate directly to the cloud, you will need to install the modules and register that station separately.

NOTE: If the Workbench platform is used to install the Niagara Cloud Service to a JACE, you need to also install the modules on the JACE using the platform's **Software Manager** view. For more information about the **Software Manager** view, see the *Niagara Platform Guide*.

Step 1. Open a web browser and log in to the Niagara Community Software portal.

The address is `https://www.niagara-community.com`.

- Step 2. Click **Software** in the upper right of the home page.
- Step 3. Scroll down to locate CloudLink and click the appropriate zip file link.
The choice depends on your Niagara version. You should choose the same major/minor/update version as the Niagara version that you currently use. For example, if you use Niagara 4.10.9, the filename of the zip file would contain "4.10.9" in it. The build version may be different for Niagara Cloud Service and Core Niagara).
The zip file downloads to your system.
- Step 4. Navigate to your Windows downloads folder (`c:\Users\<UserName>\Downloads`) where `<UserName>` is unique for your computer.
- Step 5. Right-click the zip file in the downloads folder and extract its contents to your SysHome installation folder (for example, `Niagara/Niagara-4.10.x`).
NOTE: If the system prompts you to Overwrite any existing previous versions?, click OK.
This installs the modules. The palettes are included in the modules.
- Step 6. Restart the station and restart Workbench.
- Step 7. To install the software on any remote platform (JACE), use the Platform Administration Commissioning tool or the Software Manager tool.

Result

Once the station restart is complete, you can proceed to install and configure Niagara Cloud Service.

Setting up device Internet access

Internet access is required for all stations and clients. If your device is on an internal (closed) network, this is done by setting up proxy server settings typically handled by the on-site IT department. Your proxy server must allow access to the Niagara Cloud Suite. Explicit (named) proxy support is provided through the **net-HttpProxyServer** from the **net-rt** module and configure it to your proxy server settings. For information on how to set up the **proxyService**, refer to the *Getting Started with Niagara*.

Prerequisites:

You are working in Workbench with a platform connection to the controller. For each device behind a network firewall, appropriate DNS Host name and DNS Server IP address(es) are available for your network. Your platform's clock is synchronized with the cloud platform.

If the **proxyService** is available and configured, Niagara Cloud Service automatically uses it. If you are using a proxy server with the **net-HttpProxyServer**, the **proxyService** must be able to access the following domains, which are part of the Niagara Cloud Suite:

- *.azure-devices.net
- *.force.com
- *.honeywell.com
- *.honeywellcloud.com
- *.niagara-cloud.com
- *.niagara-community.com
- *.pingone.com
- *.tridium.com
- *.windows.net

- Step 1. In the platform **TCP/IP Configuration** view, enter the appropriate values for the following properties:

- **DNS Domain** (for example: company.net)
- **DNSv4Servers** (add a field for one or more DNS Servers; enter the appropriate IP address for each)

Step 2. Click **Save**.

On saving your changes you are prompted to reboot the device.

Next steps

CAUTION: From a cyber security perspective, it is crucial that your station is not exposed on the Internet. Communications via Niagara Cloud Service require only an outbound connection from your station to the Internet. Follow the best practices in the *Niagara 4 Hardening Guide* which is available on: <https://www.tridium.com/us/en/services-support/library>.

Niagara Remote

Niagara Remote connects you directly to the station virtual machine (VM), which runs on premise through Niagara Cloud Suite without the need for a separate on-premise VPN installation.

Most functionality is performed in the cloud. On the station side, you will see the **Niagara Remote Transport** component.

This is what you should know about it:

- Before it automatically installs upon station registration in NCS, you have met the following requirements:
 - You have purchased the Niagara Remote license.
 - You have installed the Niagara cloudLink modules. They are available at **Niagara Community** (<https://www.niagara-central.com>) > **Software**. Be sure that the modules version matches the Niagara version you are using.
- In general, there is no need for configuration as the NCS Device Provisioning Service does it for you.
- You can disable the Niagara Remote connectivity by setting the **Enabled** property to `False`.

Chapter 2. Install and configure

The following procedures describe the steps to add the NiagaraCloudService to your station, register the device with Niagara Cloud Suite and begin sending data.


Most components required for communicating to the cloud will be contained under the NiagaraCloudService. This service provides the authentication, transports, and channels necessary to communicate with Niagara Cloud Suite.

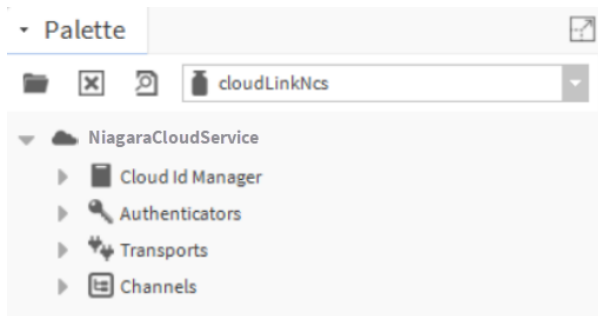
Adding the NiagaraCloudService

The **NiagaraCloudService** component under the **Services** container connects the station to the Niagara cloud.

Prerequisites:

You are working in Workbench and are connected to a station. The cloudLinkNcs module is installed.

- Step 1. To open the **Palette** side bar from the **Menu** bar, click **Window > Side Bars > Palette**. The **Palette** side bar opens on the lower left of the page.
- Step 2. Click on the Open Palette (folder) icon (). The **Open Palette** window opens.
- Step 3. Enter `cloud` in the filter box, select the cloudLinkNcs palette and click **OK**.



The palette opens in the side bar.

- Step 4. Expand your station and drag **NiagaraCloudService** to the **Services** container in the Nav tree. The **Name** window opens.
- Step 5. Accept the default name or enter the different name and click **OK**.

Registering a device

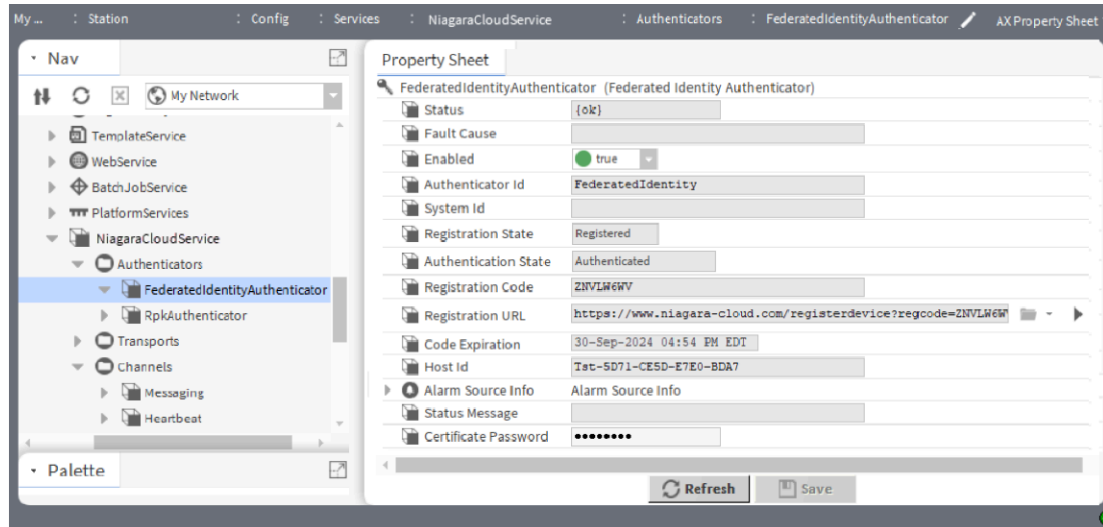
This procedure registers devices (stations) with specific customer projects. It is required for the use of Niagara Data Service, Niagara Recover, and Niagara Remote.

Prerequisites:

- You are using Workbench and are connected to a station to which you added the NiagaraCloudService.
- You have a Niagara Community account.
- You have set up projects in the Niagara Cloud Management Portal.

- Step 1. Expand **NiagaraCloudService > Authenticators** and double-click **FederatedIdentityAuthenticator**.

The **FederatedIdentityAuthenticator** Property Sheet opens.



- Step 2. Right-click the authenticator name and click **Actions > Start Registration**. This action announces the station to the cloud registration service from which it receives the **User Code**, and populates the **User Code**, **User Registration Url** and **Code Expiration** properties.
- NOTE:** The **Registration Code** is good for 15 minutes. If you take longer than that to complete registration, an error occurs and you must start again.
- The expiration time displays as **Code Expiration**. You need to complete the next step in the portal before the time is up or you will have to start again.
- Step 3. Click the link arrow to the right of the **User Registration Url** property or copy the URL and paste it into a browser. For more information about how to configure the web-browser whitelist (allowlist), see "Configuring the web-browser whitelist (allowlist)" in the *Getting Started with Niagara* guide and "Adding cloud endpoints to the Workbench browser allowlist." in the *Niagara Cloud Suite (NCS) Partner Guide*. The Niagara Community log-in window opens.
- Step 4. Log in to the Niagara Cloud Management Portal using your Niagara Community account.

The **Register new device** window opens showing the **Registration Code**.

Register new device

Registration Code
7R93E9NY

Device Name
My Device

License

LICENSE ID	CUSTOMER NAME	FEATURES
10564088	NCS Test Customer 1	Remote, Recover, NDS ↗

Project Name
Project 1 (selected)
Project 1
Project 2

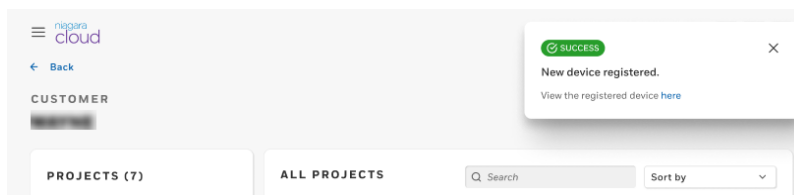
Location
Location

Cancel **Done**

Step 5. Enter a **Device Name**, select a license from the available licenses if there are more than one licenses, select a project for the customer from the **Project Name** list, enter the **Location** and click **Done**.

- **Device Name** can be the station name. However, you can change it to make it more descriptive of the project or location.
- **Licenses** selects the desired license from all available licenses to determine what features and functionality, which will be based on ordered subscriptions, are authorized to use.
- **Project Name** is the name of the Niagara Cloud Suite project to which this device belongs. It is selected from a dropdown of all the projects to which the user has access.
- **Location** identifies the building's geographic location.

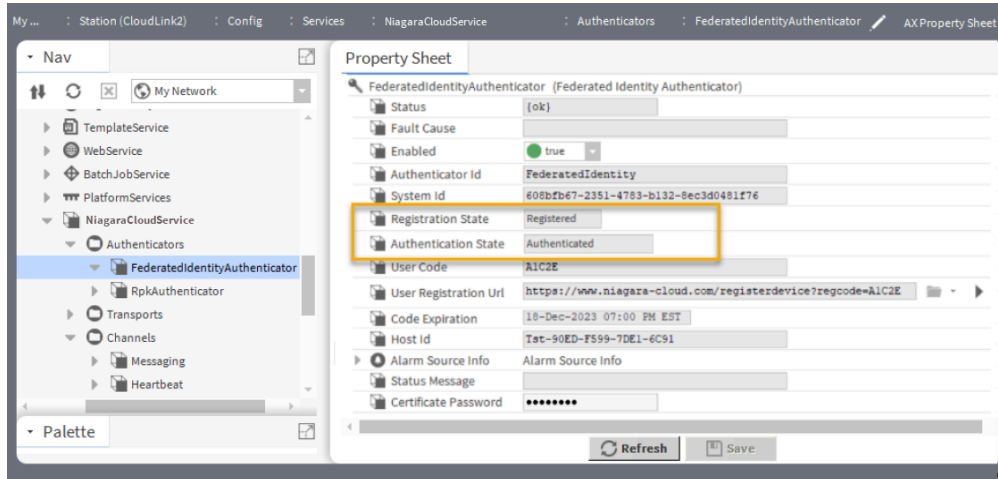
The success pop-up confirms the device registration.



The system registers the device with the Niagara Cloud.

Step 6. To confirm the federated registration and connection, go back to the station's **FederatedIdentityAuthenticator Property Sheet**.

The property sheet opens.



The device is registered and, after a moment authenticated, which means that it has its station certificate, and that the software has provisioned Niagara Cloud Service.

NOTE: The provisioning process sometimes can take a few minutes before everything is set up in the station and fully registered. For example, the RPK Authenticator takes time to become authenticated. Until then, it may be present but it is disabled. It will not work if you try to enable it.

The provisioning of the components takes place based on the device subscriptions you ordered in Niagara Licensing. As an example, if you order Recover, under **Channels**, the **Backup** channel will be automatically added.

Result

The platform and station are now fully registered with the Niagara Cloud Suite. They have a certificate for the federated identity and are connected to the IoT Hub (the cloud). However, no data have been sent to the cloud.

Re-registering a device

You need a station's cloud identity to connect to Niagara Cloud Service and it's related data stored in the cloud. If the cloud identity is lost or the station is moved to a new device (device replacement), you need to re-register the device to restore the station's cloud identity.

Examples of scenarios in which a re-registration is needed:

- Restoring a JACE station from a backup (DIST tool) or just a station copy to a new JACE .
- Restoring a Supervisor from a backup with a station copy (DIST tool is not supported on a Supervisor) to a new computer.
- If the cloud identity of a station is lost, for example, because the NiagaraCloudService was deleted, or authenticators or certificates expired.

NOTE: If the station is restored from a backup or copied to the same device, you do not need to re-register the device or change the license.

Device restore (same device): If the station is restored from a backup or copied to the same device, you may not need to re-register the device. Device re-registration is needed only if the station is not connecting to the cloud, for example, because certificates expired. Since the station is running on the same host, you do not need to change the license.

Device replacement (new device): This is the case where a previously registered station is copied or restored from a backup to a new device. In this case, the prerequisite is that you move the subscription from the old

license bound to the old Host ID to a new license that is bound to the new Host ID. Otherwise, the re-registration process will not find a subscription for the new Host ID and the re-registration process cannot be completed.

NOTE: It is possible in some cases that a device replacement that was restored from a cloud backup will still retain its cloud identity. However, you should only consider this as temporary and re-register the device to avoid future subscription issues with the new Host ID.

Re-registering from a station

Step 1. In the station:

- Expand **Config > Services > NiagaraCloudService > Authenticators**, and delete **FederatedIdentityAuthenticator** and **RpkAuthenticator**.
- In the cloudLinkNcs palette, expand the **NiagaraCloudService** component, and expand the **Authenticators** folder.
- Drag a new **FederatedIdentityAuthenticator** into the station under **Config > Services > NiagaraCloudService > Authenticators**.
- To start the re-registration process, right-click on **FederatedIdentityAuthenticator** and select **Actions > Start Registration**.

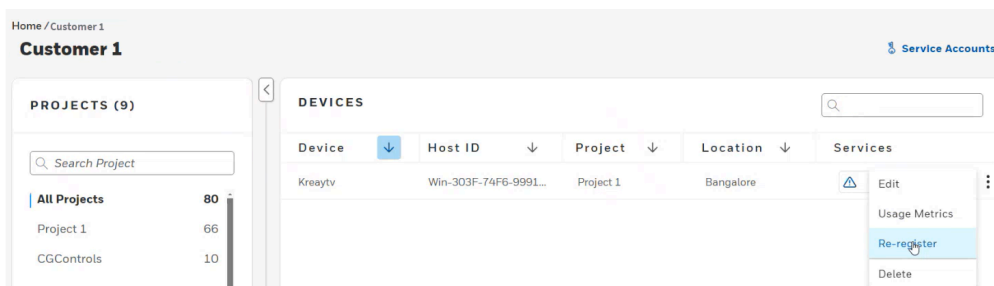
Property Sheet	
FederatedIdentityAuthenticator (Federated Identity Authenticator)	
Status	{ok}
Fault Cause	
Enabled	true
Authenticator Id	FederatedIdentity
System Id	
Registration State	Unregistered
Authentication State	Unauthenticated
User Code	LNQ9XHNQ
User Registration Url	https://www.qa.niagara-cloud.com/registerdevice?regcode=LNQ9XHNQ
Code Expiration	25-Jul-2024 08:10 PM IST
Host Id	Win-303F-74F6-9991-D810
Alarm Source Info	Alarm Source Info
Status Message	
Certificate Password	*****

The user code automatically populates the **User Code** field.

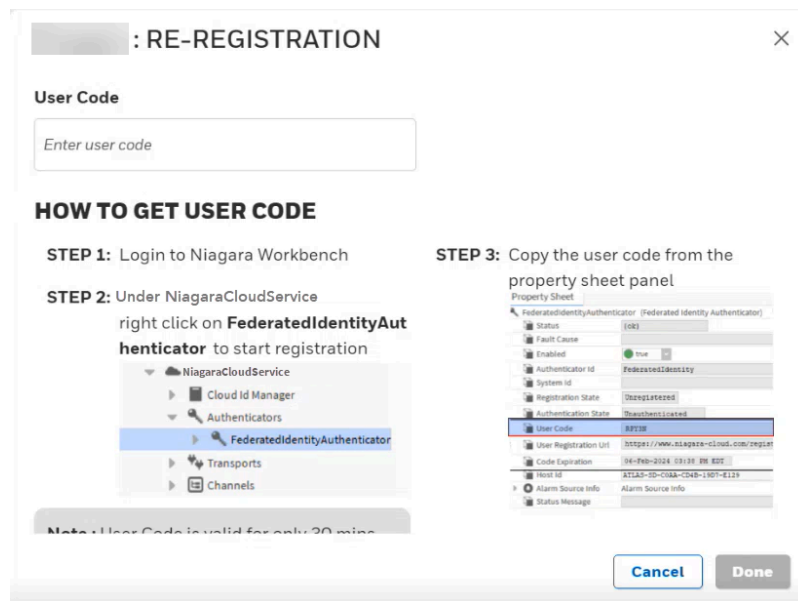
- Copy the code from the **User Code** field as you will paste it in the **User Code** field in the Niagara Cloud Management Portal Re-Registration window.

Step 2. In the Niagara Cloud Management Portal:

- Navigate to the device you wish to re-register and click the vertical ellipsis (three vertical dots).



- From the drop-down menu, select **Re-register**.



The Re-Registration window opens containing the **User Code** field in which you enter the user code and the step-by-step instructions on how to obtain the user code.

- c. Paste the user code that you copied earlier from the station's **User Code** field in the **User Code** field of the Re-Registration window, and click **Done**.

Step 3. In the station, check the status of the **FederatedIdentityAuthenticator** and the **RpkAuthenticator** as shown below.

FederatedIdentityAuthenticator: In Config > Services > NiagaraCloudService > Authenticators, the **FederatedIdentityAuthenticator** should show the **Registration State** as **Registered** and **Authentication state** as **Authenticated**. The **Host Id** property should show the current Host Id.

RpkAuthenticator: In Config > Services > NiagaraCloudService > Authenticator, the **RpkAuthenticator** screen should have a green check next to **Device Registered**, **Authenticator Enabled**, and **Device Connected**. The **RpkAuthenticator** should have values for **System Type** and **System Guid**.

Step 4. Restart the station.

Niagara Cloud Service configuration

Since Niagara Cloud Service is not a driver, it differs from the typical Niagara driver structure, that is, a network containing one or more devices. Instead, Niagara Cloud Service configuration is controlled by the Channels and Transports within the NiagaraCloudService.

NOTE: Not all channels, transports, and authenticators are present in all Niagara Cloud Service installations. The population of your NiagaraCloudService depends on the Niagara Cloud Management Portal subscriptions you have purchased for the station.

- **Alarms Channel**

This channel manages delivery of alarm messages to the cloud. It installs a **CloudLinkAlarmRecipient** in the **Alarm Service** which should be connected to an Alarm Class in order to route alarms to the cloud. See the [Cloud Link Alarm Recipient](#) section for more details regarding alarm configuration.

- **Backup Channel**

This channel uploads station backups to the cloud according to the configuration in the backup policy. For more information regarding backup configuration, see "cloudLink-BackupChannel" in the Components section of this guide.

- **Commands Channel**

This channel comes pre-configured with various commands which can be also configured individually. See “CloudLink-CommandsChannel” for detailed information on configuring commands. For more information regarding command configuration, see “cloudLink-CommandChannel” in the Components section of this guide.

- **Events Channel**

This channel installs a CloudLinkEventRecipient in the **Event Service** which should be connected to an Event Source in order to route events to the cloud. For more information regarding event configuration, see “cloudLink-EventsChannel” in the Components section of this guide.

- **Heartbeat Channel**

This channel delivers heartbeat messages to the cloud platform at regular intervals. For more information regarding heartbeat configuration, see “cloudLink-HeartbeatChannel” in the Components section of this guide.

- **Histories Channel**

This channel handles history delivery to the cloud platform. For more information regarding histories configuration, see “cloudLink-HistoriesChannel” in the Components section of this guide.

- **Messaging Channel**

This channel delivers messages to the cloud platform that have already been serialized. For more information regarding messaging configuration, see “cloudLink-MessagingChannel” in the Components section of this guide.

- **Model Channel**

This channel handles model export to the cloud platform. For more information regarding model configuration, see “cloudLink-ModelChannel” in the Components section of this guide.

- **Points Channel**

This channel has a Default Export Policy, which is disabled by default. It is recommended to use Histories to send telemetry data to the cloud. If this channel has export policies which are enabled, both points and any history data for the point will be sent to the cloud for the same cloud id.

This channel also has a Cov Export Policy that subscribes to the selected points and then sends batch updates. The default batch time is about one second.

NOTE: For Points channel, the Model Channel cloud Id export must occur first. The Cov Export Policy fails without nc:cloudId tags on the points.

Alarms channel

This channel installs a CloudLinkAlarmRecipient in the **Alarm Service**. To route alarms to the cloud, it should be connected to one or more Alarm Classes.

Cloud Link Alarm Recipient

The **CloudLinkAlarmRecipient** is an extension to the standard **AlarmRecipient** component and routes alarms to the cloud platform.

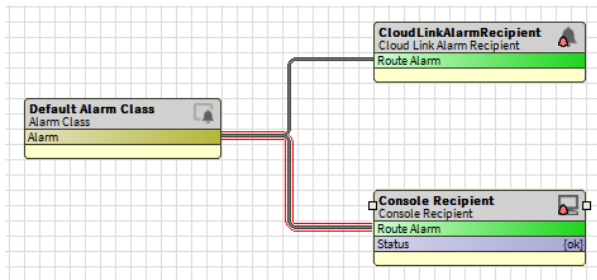
You must configure this component to route alarms to the cloud using the registered and configured **NiagaraCloudService**. The protocol used to send alarms is determined by the alarms channel of the specified **NiagaraCloudService**.

To send alarms to the cloud, an alarm class of the alarm sources must be selected, which is routing to the **CloudLinkAlarmRecipient**. One or more alarm classes can be linked to the **CloudLinkAlarmRecipient**.

As seen in the below figure, under the **Station > AlarmService**, the wire sheet view shows that the Default Alarm Class is routing to the **CloudLinkAlarmRecipient**. The figure also shows the optional connection of the Default Alarm Class to a standard alarm Console Recipient, which you can use to view the alarms.

NOTE: When you add the **NiagaraCloudService** to the station, an instance of the **CloudLinkAlarmRecipient** is created but no alarm classes will be connected to it. Connecting one or more alarm classes to the **CloudLinkAlarmRecipient** must be performed manually.

Figure 2. CloudLink Alarm Recipient added to AlarmService Wire Sheet



Messaging channel

This channel allows for direct messaging to the transports. It handles sending individual messages to the cloud platform using one or more transports, and is used by the authenticators to authenticate to the cloud platform.

Heartbeat channel

This channel is used to maintain an active link with the cloud platform. It periodically sends a heartbeat message according to its frequency setting and is enabled by default.

Model channel

This channel sends detailed component information to the cloud and includes points, histories, and log histories. The information includes type, facets, properties, tags, and relations. The channel comes with one **CloudIdExportPolicy**, which sends components that have a **cloudId** tag but not an **nc:excluded** tag. The Cloud Identity Manager automatically executes the policy if new components are added to the station. If you make changes without adding any new components, you can manually execute the export policy to upload the changes to the Niagara Cloud.

What you should know about the Model channel.

NOTE: It is required to use the Cloud Id Manager to ensure that the proper cloud Ids are added to the new components, otherwise the model will not send them to the cloud.

- **Temporary Model files**

During Model export, the system only creates temporary files in the station folder (named **cloudLinkModel**) by default if configured accordingly in the Channel Config. It then uploads the files to the cloud. In normal operation, there is no need to review the files, however, it may be necessary to view these files for troubleshooting purposes.

The Model export creates two types of files, one CSOM file for tag dictionary data and one or more component files for control points, histories, and others. The file names include the station's global Id, an export number, a file number (component file only) and are in GZIP-compressed JSON format. If you set **Model Channel > Channel Config > Delete Model Files** property to **false**, you can retain the created files. They will not be deleted after upload for each Model export.

NOTE: The **Upload Model Files** property controls whether or not the Model data files are sent to the cloud for each export. For normal operations, set this property to **true** so the Model is uploaded to the cloud. For troubleshooting purposes, you can temporarily change it to **false**.

The model files are stored in the folder **<station>\cloudLinkModel**.

- **Cloud Id**

The cloud Id is a read-only direct tag (**nc:cloudId**) and is the component's unique identifier in the cloud.

The Model channel only exports components that have cloud Ids. The Component Identity Worker under the Cloud Id Manager adds the cloud Ids. To add the cloud Ids, execute the Cloud Id Manager, which will automatically trigger the model if there are new components. When components are added to or removed from the station, or if the nc:excluded tag is added to or removed from any component, execute the Cloud Id Manager. Only a select set of components will receive a cloudId tag including control points, history imports, audit histories, log histories, devices, and networks.

- **Excluding components**

The Model channel provides a mechanism to exclude components from the model as well as prevent telemetry from being sent to the cloud. This is accomplished by adding an nc:excluded direct tag to the component. The nc:excluded tag can be added to a parent component and all of its descendents will also receive an nc:excluded implied tag. The nc:excluded tag is available in the Niagara Cloud tag dictionary.

- **Proxy point and history imports**

When the Model export process starts, it first performs a matching process between device proxy points and device history imports. Matching proxy points and device history imports will share a new nc:telemetryId tag. This new Id is used to identify the histories in the cloud. If there is no matching proxy point for a history import, then the model will contain the history import.

Histories channel

This channel sends history records to the cloud according to a configuration made in a history export policy. It comes with an export folder with a pre-configured autoExport policy, which is disabled by default. The autoExport policy defaults to send all histories to the cloud but allows for the exclusion of individual histories. To enable history exports, either enable the autoExport policy and configure its execution time or add additional custom history export policies. See **CloudLink-HistoriesChannel** for detailed information on configuring history export configurations.

NOTE: It is recommended that you only use an autoExport for large stations without selecting individual histories for exclusion. Configuring custom export policies on large stations can result in slow response times on the history export policy screens.

Cloud History Export Manager

The Cloud History Export Manager view allows you to discover histories and assign them to **CloudHistoryExportConfigs**, or include/exclude them from **Auto Export Configs**.

The Cloud History Export Manager is the default view of the Exports component in the Histories Channel.

Figure 3. Auto Export Configuration

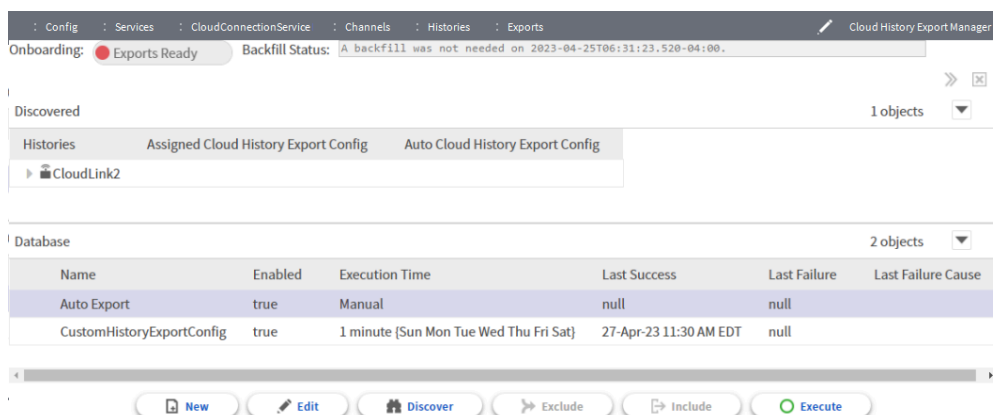
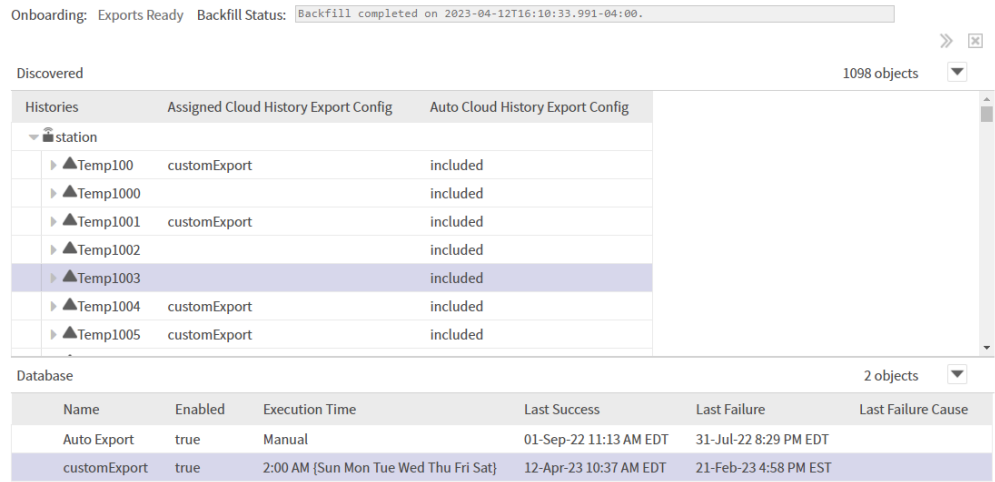


Figure 4. Custom History Export Configuration



Backfill Status

The backfill status field displays the status of a backfill operation, which is run when the number of pending history records exceeds the Backfill Record Threshold value. The histories are sent to the cloud by bulk upload rather than via AMQP transport. The bulk upload mechanism is designed to support large numbers of histories. When a backfill is running, no histories are sent to the cloud with the AMQP transport and the Onboarding field will display Exports Suspended.

The following table explains the functionality of the Cloud History Export Manager buttons.

Button	Description
New	Creates additional custom CloudHistoryExportConfig objects.
Edit (or double-clicking on the entry)	Opens a dialog window for changes to the CloudHistoryExportConfig details.
Discover	Discovers all eligible histories in the station.
Assign (when a CloudHistoryExportConfig is selected)	Adds the history to the CloudHistoryExportConfig if a CloudHistoryExportConfig in the Database section and a history in the Discovered section are both selected. NOTE: The Assign/Exclude buttons change text depending on the selection of a config entry in the Database section.
Unassign (when a CloudHistoryExportConfig is selected)	Removes the history from the CloudHistoryExportConfig. NOTE: The Unassign/Include buttons change text depending on the selection of a config entry in the Database section.
Exclude (visible when Auto Export is selected)	Excludes the history from the Auto Export when the Auto Export config is selected in the Database section and one or more histories are selected in the Discovered section. NOTE: The Exclude button for Auto Export becomes Assign if you select the CloudHistoryExportConfig.
Include (visible when Auto Export is selected)	Includes the history in the Auto Export when a history is selected. NOTE: The Include button for Auto Export becomes Unassign if you select the CloudHistoryExportConfig
Execute	Executes the selected CloudHistoryExportConfig if it is enabled. When a History Config changes from disabled to enabled, a backfill check will be performed, which may trigger a backfill if the number of pending histories exceeds the Backfill Record Threshold.

Exporting Histories data

Follow the below steps to ensure that a station’s histories are successfully sent to the cloud.

Prerequisites:

- Your station is registered.
- You have executed the Cloud ID Manager.

Step 1. Select **NiagaraCloudService > Channels > Histories > Exports > Auto Export** and set **Enabled** to **true**.
 Enabling an export policy checks if bulk history files should be uploaded due to **Backfill Record Threshold** and if necessary, start the bulk file upload process. After the backfill check is complete, the **Onboarding** mode is set to **Export Ready** and new histories records begin to flow to the cloud based on history export policies.

Step 2. Optional: Create new custom history export configurations (recommended for small stations).

Backup channel

This channel allows you to make Niagara backups of the station according to your backup policy configuration and stores the backup in the cloud within the Niagara Cloud Suite.

The policy allows you to configure the backup times and what is included within the backup. It contains the time at which the backups will run with a trigger mode that you can change as needed. The default backup policy is configured to back up around 2 a.m. on Sundays with a randomization of 1 hour. See Niagara Cloud Management Portal to learn more about the limit on the number of backups.

Your backup is stored securely within the Niagara cloud. It is encrypted with an encryption key, which is either your station platform's system passphrase or a password that you entered before the backup was uploaded to the cloud.

NOTE: It is important to remember the system passphrase or user-created password and keep it safe. If you lose the system passphrase or password, you will not be able to recover the backup. The backups are secured with the current system passphrase or password. If the system passphrase or password is changed, new backups will use the new system passphrase or password but previous backups in the cloud will still use the previous system passphrase or password.

Only you or someone you give the passphrase or password can decrypt and use the backup. You can annotate the backup with notes and retrieve it anytime for restoring into the platform or into a new platform in the event of hardware failure.

The file name of the backup has to following format:

```
backup_<device name>_<YYYYMMDDHHMMSS>_<device uuid>.edist2.
```

- **device_name**: the name entered when the station was registered or changed using the Niagara Cloud Management Portal
- **<YYYYMMDDHHMMSS>**: the date of the backup in the indicated format
- **device_uuid**: the unique identifier assigned to the device when registered

The cloud backup is run as a Niagara job. You can view the progress and log in the Job Service. To enter backup notes that are associated with the backup in the cloud, select **NiagaraCloudService > Channels > Backup > Policies > Default Backup Policy > Backup Note**.

NOTE: If a cloud backup fails, an alarm will be created if the **Alarm on Failure** property is set to **true** at **NiagaraCloudService > Channels > Backup > Policies > Default > Backup Policy > Alarm on Failure**.

Niagara Cloud Service for Niagara Cloud Suite uses an authentication certificate that will be updated approximately every 60 days. The system will trigger an automatic backup when the certificate is updated so that the correct certificate is stored in the backup `.dist` file.

Backup files can be downloaded through the Niagara Cloud Management Portal.

Decrypting an encrypted backup file

You can decrypt an encrypted backup file by navigating to the downloaded file within Workbench.

In Workbench, double-click the desired backup file. A window will open indicating the path to the file. Click **Decrypt DIST file** to open the **Encryption Key** window, where you will be prompted to enter either the system passphrase or the password that was used at the time the backup was made. Click **OK**. The backup will be decrypted to the same directory as the encrypted backup.

Chapter 3. Data management

This chapter provides information about services you can use to manage data in Niagara Cloud Service.

NOTE:

- Niagara Cloud Service currently does not support the **NiagaraSystemHistoryImports** and history exports, including the **NiagaraHistoryExports** and **NiagaraSystemHistoryExports** components. As a result, their associated components and histories are also currently not supported in Niagara Cloud Service.
- Optional modules: If you use NiagaraNetwork devices, ensure that the optional module **clUtilsNiagara-rt** is installed. If you use Bacnet devices, ensure that the optional module **clUtilsBacnet-rt** is installed.

Cloud Archive History Provider

As of Niagara 4.13 or later, the **CloudArchiveHistoryProvider** allows queries against local history records to be supplemented by archived history records that have been previously exported to a cloud platform using Niagara Cloud Service.

The **CloudArchiveHistoryProvider** is automatically added to the station's **HistoryService** in the **Archive History Providers** container after the **NiagaraCloudService** is registered to the cloud. The **CloudArchiveHistoryProvider** makes it possible for any existing views that query histories to benefit from both local and archived records. While the station still stores local histories, once Niagara Cloud Service exports those history records to a cloud platform, you can reduce the capacity of those local histories to free up resources in your station. At history query time, the **CloudArchiveHistoryProvider** can easily retrieve those exported (older) archived history records residing in a database located in the cloud platform.

License update requirements

To use the **CloudArchiveHistoryProvider**, your license requires the following updates:

- A general historyArchive license feature that covers any archive history provider implementation
- A general Niagara Cloud Service license feature that covers moving history data to or from the cloud platform
- An active SMA agreement

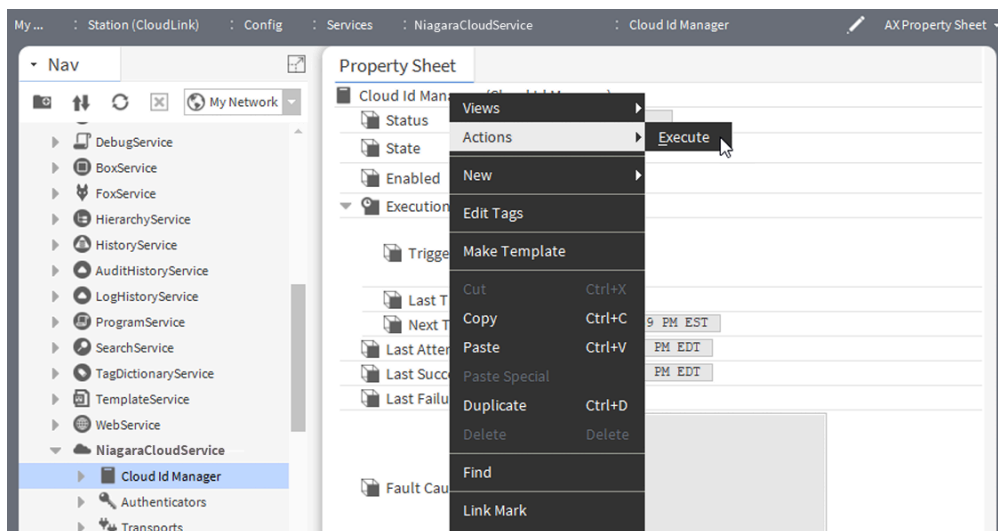
Assigning cloud Ids to station components using Cloud Id Manager

You can use the **Cloud Id Manager** component to assign cloud Ids to the components in the station and track cloud Ids for histories. As histories can be imported into the station without an existing component in the station, this allows for storing the history's cloud Id. The **Cloud Id Manager** detects changes to components in the system and automatically executes the Model Channel's Cloud Id Export Policy.

Prerequisites:

- You have added the **NiagaraCloudService** to your station.
- If needed and as part of the initial onboarding process, mark any components that you want to exclude from being uploaded to the cloud by using the **nc:excluded** tag.
- Optional modules: If you use NiagaraNetwork devices, ensure that the optional module **clUtilsNiagara-rt** is installed. If you use Bacnet devices, ensure that the optional module **clUtilsBacnet-rt** is installed.

Step 1. To run the **Cloud Id Manager** component, expand **Config > Services > NiagaraCloudService**.



Step 2. Right-click on **Cloud Id Manager** and select **Actions > Execute**.

NOTE: The Model Channel's Cloud Id Export Policy is automatically linked to the **Cloud Id Manager**, so it is executed when new cloud ids are assigned.

Excluding components and histories from cloud upload

You can exclude components and histories from being uploaded to the cloud using the `nc:excluded` tag. The `nc:excluded` tag allows you to select station's components and histories **not** to be uploaded to the cloud.

Prerequisites:

The Niagara Cloud tag dictionary has been installed in the station, which happens automatically when adding the **Niagara Cloud Service**.

- Step 1. To exclude a component by adding the `nc:excluded` tag as a direct tag, right-click the component, select **Edit Tags** and click **Add Tag**.
The **Add Tag** dialog box opens.
- Step 2. For **TagId (nn:tt)**, enter `nc:excluded`, click **OK** and click **Save**. **Type** defaults to `baja:Marker`, which does not require changing. Alternatively, select the dictionary from the drop-down menu and double-click on the excluded marker tag.
Alternatively, select the Niagara Cloud dictionary in the **Edit Tags** window, double-click on the excluded tag, and click **Save**.

Figure 5. The nc:excluded tag added as a direct tag on a station under the NiagaraNetwork

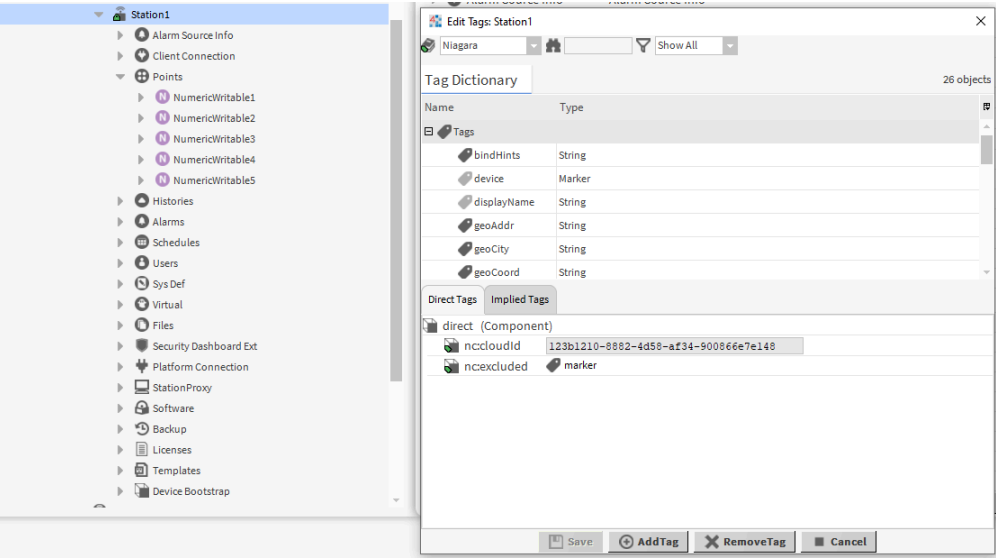
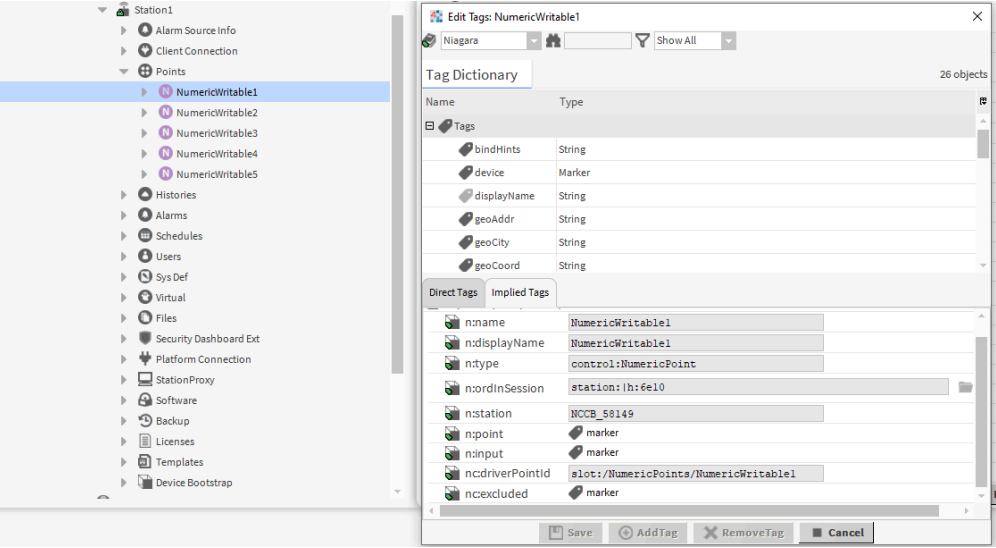


Figure 6. The nc:excluded tag added as implied tag to the components in the station



Tagging a component with the nc:excluded tag also adds an implied nc:excluded tag to all of the component's descendants. This means that tagging a device, for instance, excludes that device and everything it contains (such as proxy points and history imports).

As another example, tagging a BacnetNetwork excludes the network, along with all Bacnet devices, Bacnet points, Bacnet trend logs, etc., contained within the network. If any excluded components have associated histories, such as history imports or points with history extensions, those histories are also excluded from being upload to the cloud.

NOTE:

On an individual component, the nc:excluded tag may also be added as an implied tag using tag rules. Its descendants do not automatically get tagged with the nc:excluded tag.

Cloud upload exclusions using nc:excluded tag

Placing the nc:excluded tag on a component prevents the component from being uploaded to the cloud during future model upload jobs.
After running the model upload job triggered by the Cloud Id Manager, the component's associated history (if existing) is excluded from future history uploads.

NOTE: The Cloud Id Manager will trigger the model upload and ensure that the nc:excluded tags are processed. If you manually trigger a model upload instead of using the Cloud Id Manager to run the upload, the upload will have no effect on the interaction between histories and the nc:excluded tag. The histories will continue to be exported.

Finding excluded components and histories

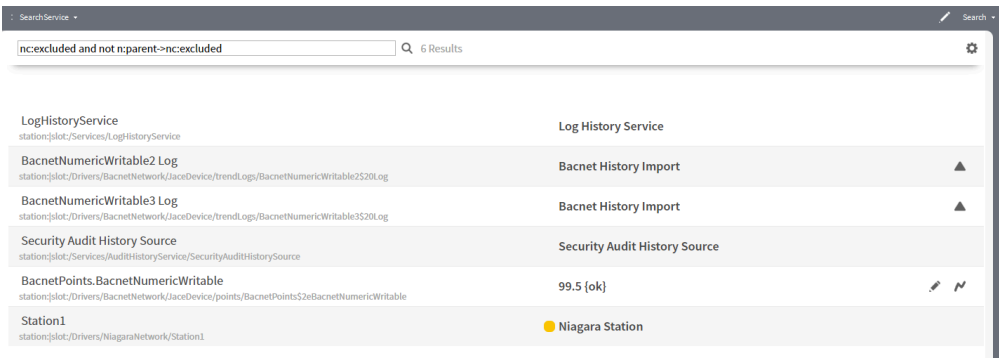
You can search for excluded components in a station using two different NEQL queries: the "nc:excluded" NEQL query and the "nc:excluded and not n:parent->nc:excluded" NEQL query. Here is an overview of the workflows that are available to you depending on the desired target information.

The "nc:excluded" query finds all components in the station that have an nc:excluded tag whether direct or implied. The search results often contain more information than needed. For instance, if you have tagged the BacnetNetwork with nc:excluded, all descendants of the BacnetNetwork will get an implied nc:excluded tag. As a result, running the "nc:excluded" query lists the BacnetNetwork and every one of its descendants as a search result, which can quickly become too much generated output if all you were interested in was the BacnetNetwork.

A common use case is the ability to find all the components that have been manually excluded. The "nc:excluded and not n:parent->nc:excluded" NEQL query searches only for components that are tagged with nc:excluded and whose parents are not tagged with nc:excluded. In the previous example, this query will only list the BacnetNetwork and not all of its descendants, which in many instances is the preferred search method.

Using the SearchService, to find manually excluded components, that is, excluded components whose parents are not excluded only, you can use the "nc:excluded and not n:parent->nc:excluded" NEQL query. This query is recommended if you want to see a list of all top-level components excluded in the station. If you want a list of all excluded components, execute the first query and then the second query, for which you need to infer that a point is excluded because it being under one of the excluded top-level components.

Figure 7. NEQL query showing all top-level excluded components in the station



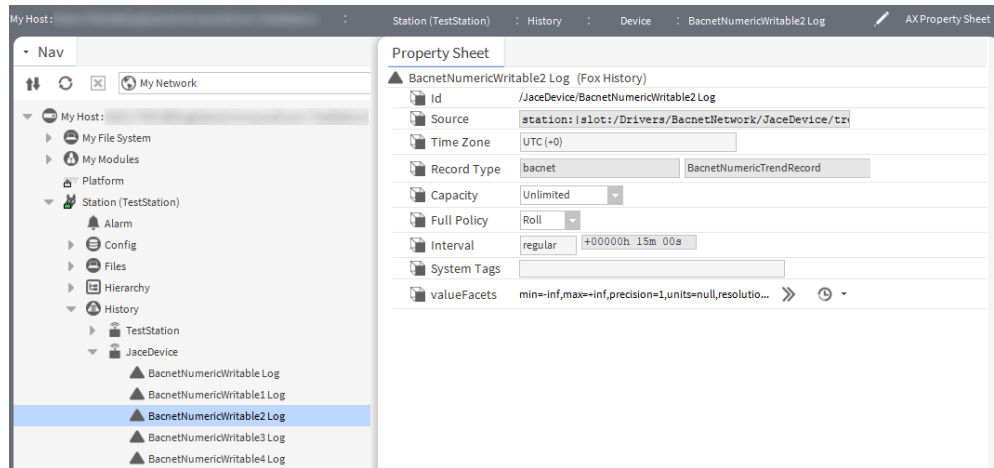
To determine if an individual component is excluded or not, check if the component is tagged with nc:excluded. If the tag is present, whether as a direct or implied tag, the component will be excluded from model uploads.

It is currently not possible to view a list of all histories that have been excluded due to their components being excluded.

However, you can determine if an individual history is excluded because its component is excluded as follows:

- If a history is listed in the history space but not listed in the **Cloud History Export Manager**, it is excluded because its component is excluded.
- Alternatively, look at the history in the history space and navigate to its **AX Property Sheet** view. You can find its component by looking at the **Source** property. If the component has an `nc:excluded` tag (direct or implied), then that is the reason why the history is excluded.

Figure 8. History's Source property on AX Property Sheet to confirm that the history `/JaceDevice/BacnetNumericWritable2 Log` is excluded because its component is excluded



Model uploads to the cloud

The `nc:excluded` tag is processed during the model upload job. Adding or removing the `nc:excluded` tag does not have an effect until you have run the model upload job.

When you run the model upload job, any components that would normally be uploaded are checked for the `nc:excluded` tag. If a component with an `nc:cloudId` tag has the `nc:excluded` tag, it will not be uploaded to the cloud.

NOTE: Do not manually remove the `nc:cloudId` tag. If the `nc:cloudId` tag is manually removed and the excluded component is included again, a new `nc:cloudId` tag will be generated and the component (and its associated history if it has one) will be treated as if it were a new and different point. As a consequence, the point's identity and previously uploaded data is lost. Whatever history still remains in the station will be sent up anew, but any data in the cloud is no longer associated with this point.

Exporting Model data to the cloud

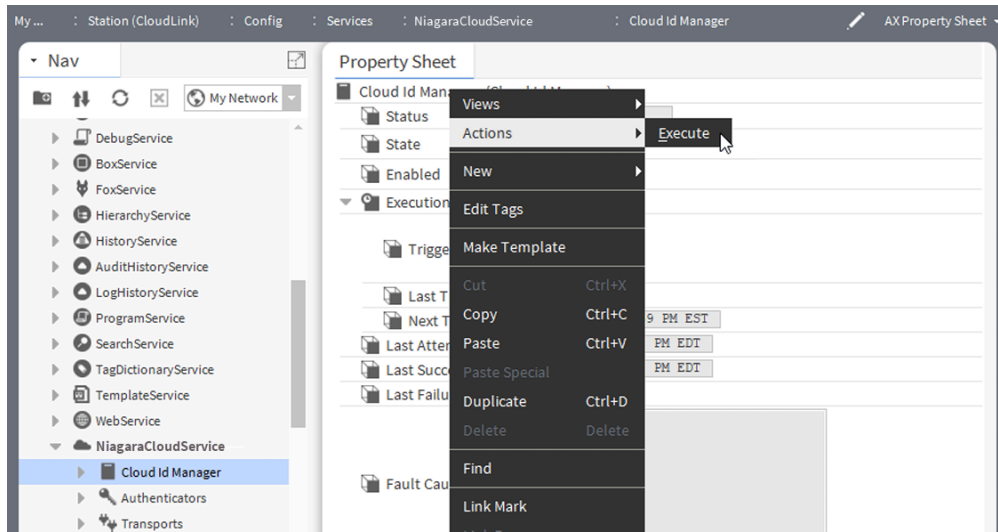
Before the station can upload data to the cloud, it must run the Cloud Id Manager, which triggers a model upload after it has assigned cloud Ids. The following steps describe how to export model data to the cloud by executing the Cloud Id Manager, which adds cloud Ids and telemetry Ids. This also triggers a model export if there are new components, otherwise the model will not be sent. This process ensures that a station's model is kept up-to-date in the cloud.

Prerequisites:

- Your station is registered.
- You have configured your networks:
 1. Add networks.
 2. Add drivers.
 3. Add proxy points (optional for model data).
 4. Add history imports (required for model and telemetry data).
- You have configured local components.

1. Add control points.
 2. Add history extensions (required for telemetry data).
- You have configured other histories as needed.
 1. Audit Service history
 2. Log Service history
 3. System Monitor Service history
 - If needed, you have tagged with nc:excluded any components or folders that you do not wish to send to the cloud.

Step 1. To run the **Cloud Id Manager** component, expand **Config > Services > NiagaraCloudService**.



Step 2. Right-click **Cloud Id Manager** and select **Actions > Execute**.

Result

The amount of time the export takes depends on the number of components in your station. You can monitor the status of the model export job in the Job Service.

After every configuration update of your network, local components, and histories, export the Model again.

NOTE: The recommended execution sequence is that you first configure the History exports, and then enable them.

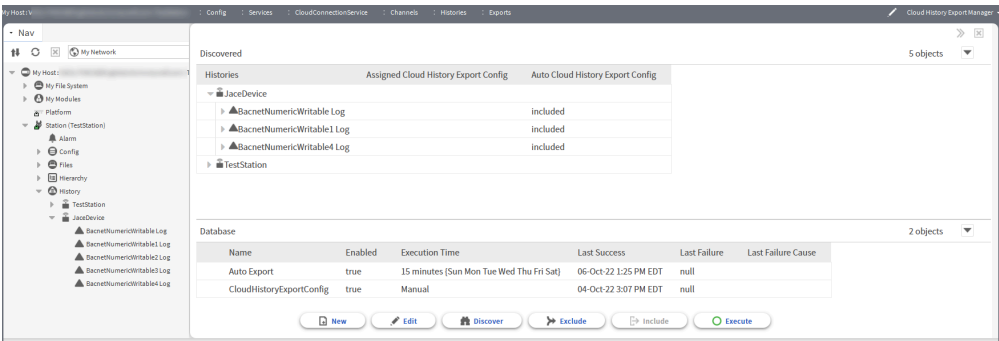
History uploads to the cloud

History exclusions are processed as part of the Cloud Id Manager execution.

WARNING: Avoid the creation of duplicate data: If you use Niagara a with Niagara Cloud Service versions prior to 4.10.6.24 for long-term support (LTS) release or 4.13.0.170 (for non-LTS release), do **not** configure multiple history extensions on a single point to send data to Niagara Cloud. Prior to these versions, stations configured with multiple history extensions on a single point would use the same cloud Id for both of the history streams, which leads to duplicated data in the Niagara Cloud Suite telemetry database.

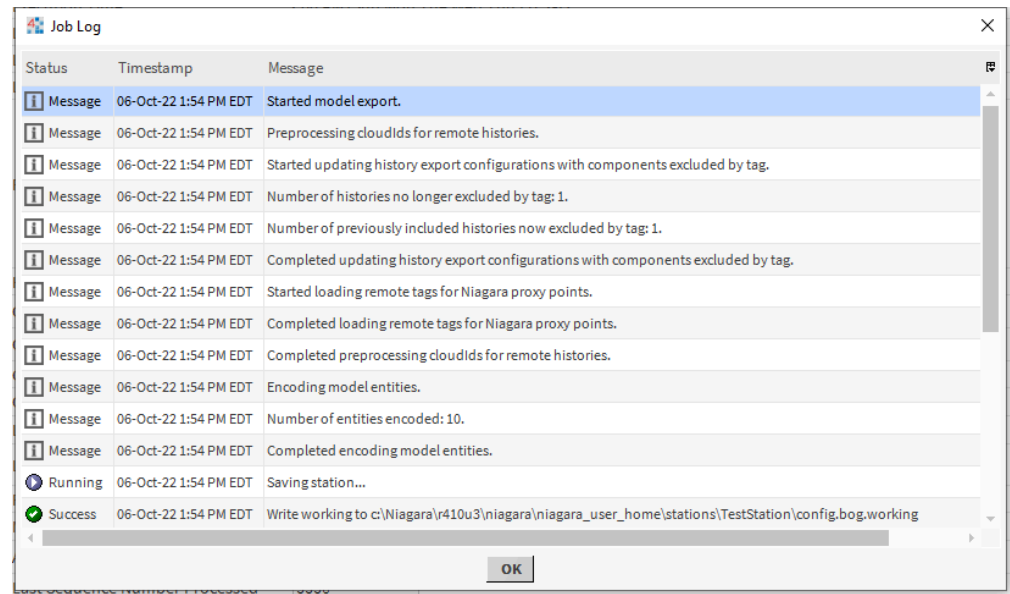
If you exclude a component with an associated history and run the Cloud Id Manager, the history will also be excluded from future history uploads. This includes history uploads via backfill and histories uploaded via the **CloudHistoryExportConfig** component. If a history is excluded in this way, it is no longer listed in the **Cloud History Export Manager**. If it was previously assigned to a **CloudHistoryExportConfig**, it is also unassigned from that **CloudHistoryExportConfig**.

Figure 9. In the Cloud History Export Manager, after excluding BacnetNumericWritable2 Log and BacnetNumericWritable3 Log, their histories are no longer displayed



If you previously tagged a component associated with a history using `nc:excluded` and then removed the tag, running the Cloud Id Manager allows the history to be included in history uploads again. The history is displayed in the **Cloud History Export Manager** and may be freely included/excluded from the auto export and assigned/unassigned from other **CloudHistoryExportConfigs**. Changes to history exclusions are logged in the station log. Specifically, the log shows the number of histories that were excluded and the number of histories that were re-included as a result of running the model upload job.

Figure 10. Model upload job log after adding one `nc:excluded` tag and removing one `nc:excluded` tag

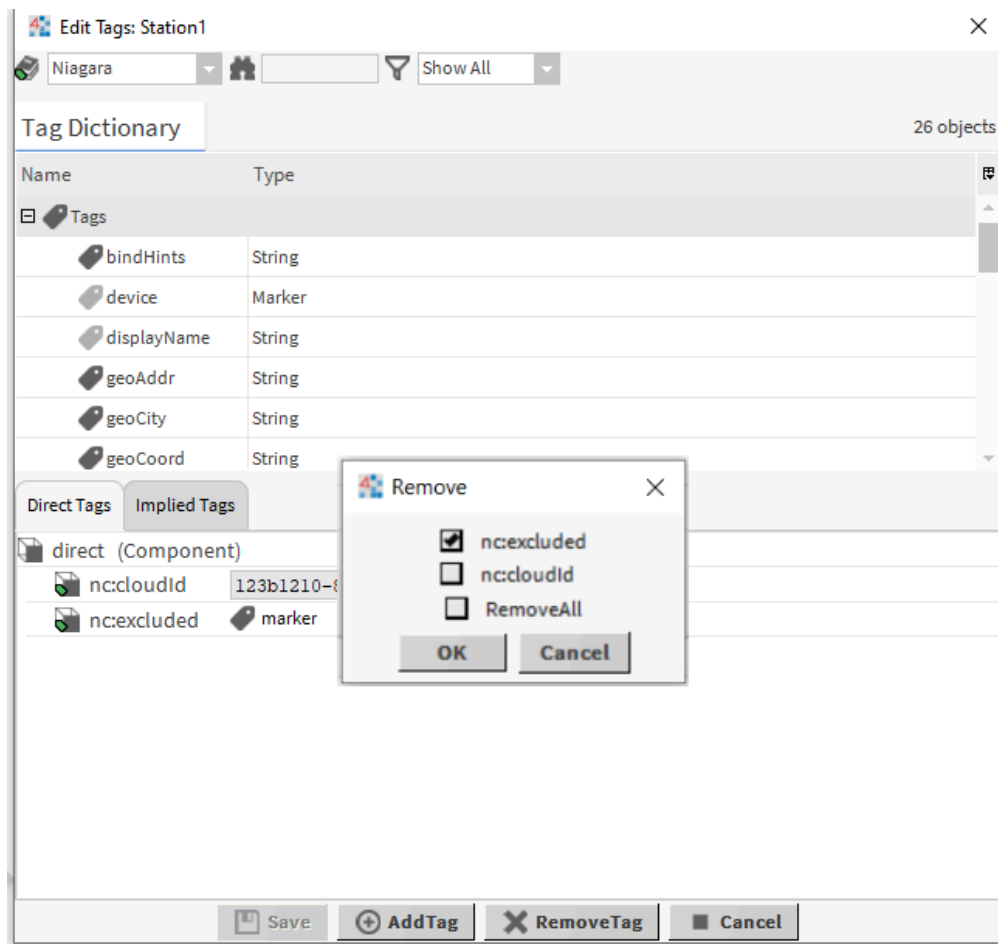


Re-including excluded components and histories

The following steps allow you to include previously excluded components and histories to be uploaded to the cloud.

Re-including components

- Step 1. To re-include a component that is currently excluded, remove the `nc:excluded` tag from the component using the **Edit Tags** window.



Step 2. From the **Remove** window, select the **nc:excluded** checkbox, click **OK**, and click **Save**.

NOTE: If you want to include the component in the model upload, the **nc:excluded** tag cannot be present as a direct or implied tag. If the **nc:excluded** tag is on the component as an implied tag because its ancestor component is tagged with **nc:excluded**, then re-tag its ancestor and/or sibling components to ensure that the components that you wish to exclude still have the **nc:excluded** tag (direct or implied), while the component that you want to include does not. Once the component no longer has the **nc:excluded** tag, the next Cloud Id Manager execution will assign the component an **nc:cloudId** tag if it does not already have one and will upload the component to the cloud.

Re-including histories

Step 1. To re-include a history that is currently excluded, re-include the component associated with that history.
This allows the history to be included in history uploads again.

Step 2. Run the model upload job to process the change.
The history is now listed in the **Cloud History Export Manager** again.

If a history is listed in the history space but not in the **Cloud History Export Manager**, while its component does not have the **nc:excluded** tag, it is likely that changes were made to the **nc:excluded** tags in the station and the model upload job was not run afterwards. Running the model upload job allows the history to once again be listed in the **Cloud History Export Manager** and included in history uploads.

Backup Channel

The Niagara Cloud Service Backup Channel uploads station backups to the cloud according to the configuration in the backup policy. The backups are similar to those made by Niagara's **Backup Service**, however, they are encrypted with an encryption key, which can be either the system passphrase or a password that you entered before they were uploaded to the cloud (also see "Backup channel" in the "Install and configure" chapter). In addition, the Backup Channel has different exclusions than the Backup Service.

Key Backup Policy features:

- The policy controls the time at which the backups will run with a trigger mode, which you can change as needed.
- The default backup policy is set to back up on Sundays at 2 a.m. with a randomization of 1 hour.
- Each device is limited to a certain number of backups (see "*Cloud backup and restore strategy*" in the Niagara Cloud Suite (NCS) Partner Guide for details).
- The file name of the backup has the following format: `backup_<device name>_YYYYMMDDHHMMSS_<device uuid>.edist2`.
 - `device_name` : the name entered when the station was registered or changed using the Niagara Cloud Management Portal
 - `YYYYMMDDHHMMSS`: the date of the backup in the indicated format
 - `device_uuid` : the unique identifier assigned to the device during registration (FederatedIdentityAuthenticator - System Id)
- The cloud backup is run as a Niagara job. You can view the progress details and log in the **JobService**.
- Niagara Cloud Service for NCS uses an authentication certificate that will be updated approximately every 60 days. The system will trigger an automatic backup when the certificate is updated so that the correct certificate is stored in the backup `.dist` file.

Uploading station backups to the cloud

Creating a cloud backup

The **Backup Channel** is used to upload station backups to the cloud. Unlike the Niagara Backup Service, these backups are encrypted with the system passphrase or user-created password.

Prerequisites:

- You have access to the Niagara Cloud Management Portal.
- The station is registered with the Niagara Cloud Suite.
- You know the system passphrase or password.

Step 1. Under **NiagaraCloudService**, expand **Channels > Backup > Policies > Default Backup Policy** and in the Backup Note property enter a note text that will be associated with the cloud backup.

Step 2. Expand **Channels > Backup > Default Backup Policy** and in the **Exclude Files** and **Exclude Folders** properties, enter additional files or folders that you want to exclude from the cloud backup.

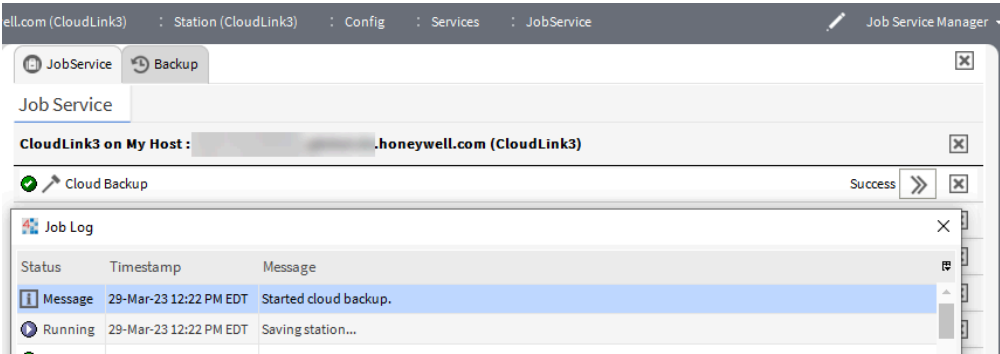
The files to be excluded are separated by a semicolon with extensions either with the complete name or using the wildcard asterisk (*). Refer to "*Workaround for Directory Ord Chooser issue*" in the "*Troubleshooting*" chapter to ensure that you enter the correct folder path for the **Exclude Folders** property.

Step 3.

- For a manual backup, expand **Channels > Backup > Policies**, right-click on **Default Backup Policy** and select **Actions > Execute**.
- For a recurrent backup, expand **Channels > Backup > Default Backup Policy** and schedule the desired time in the **Time Trigger** property.

Result

Upon executing a backup, a cloud backup job is launched and a station backup is sent to the cloud. You can view the cloud backup job under **Services > JobService**. The details of the backup can be viewed in the cloud backup entry's job log.



The backup file is an encrypted `.dist` file, which is encrypted with the system passphrase or the configured password at the time the cloud backup was made. The cloud backups can be viewed and downloaded from the Niagara Cloud Management Portal (see Niagara Cloud Suite (NCS) Partner Guide for more details).

NOTE: If a cloud backup fails, an alarm will be created if the **Alarm on Failure** property is set to `true` at **Channels > Backup > Default Backup Policy**.

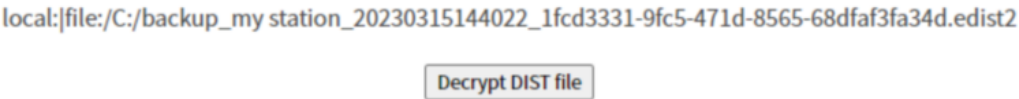
Restoring a station for a controller

You restore an individual station from a backup distribution (dist) using Workbench.

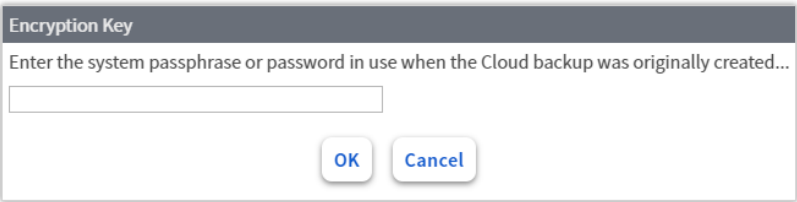
Prerequisites:

- Using the Niagara Cloud Management Portal, you have downloaded the backup file to a location on your computer's hard drive.
- You have opened Workbench and made a connection to the platform where you are restoring the station.
- You know the station's passphrase or password used at the time of backup creation.

- Step 1. Navigate to and open the downloaded backup file under **My Host > My File System**. Niagara Recover displays a table with a row for each backup.
- Step 2. Double-click the file name and click the **Decrypt DIST file** button.



NOTE: The file must be decrypted before it can be used to restore a station. The **Encryption Key** window opens.



- Step 3. Enter the station's unique passphrase or password used at the time of backup and click **OK**. Niagara decrypts the `.edist2` file, which results in a `.dist` file. The backup is decrypted to the same directory as the encrypted backup.
- Step 4. Use the platform tool, **Dist File Installer**, to restore the station.

Restoring a Supervisor station

The Niagara Distribution File Installer tool does not support restoring a Supervisor `.dist` file. However, you can manually restore the Supervisor station from a backup to the same or different computer.

Restoring Supervisor backup to the same computer

You wish to restore a previously backed-up version of a station to the same computer where the backup was made.

Prerequisites:

- You have shut down the currently running Supervisor station.
- The computer to which you wish to restore a backup version has the same Host Id, Niagara version, and Niagara Cloud Service version.
- You know the system passphrase or password in use at the time of backup. If you do not know the system passphrase, you may need to perform extra steps. You may also use the alternative password if the backup channel was configured to use this instead of the system passphrase.

Step 1. Open the Niagara **Platform > Station Copier** and copy the Supervisor station from the local or remote host to the Workbench host.

Step 2. Delete the old Supervisor station from the local or remote host.

Step 3. Using the Niagara Cloud Management Portal, download the cloud backup for the Supervisor station.

Step 4. In Workbench > **My File System**, locate the backup file ending with the `.edist2` file extension, double-click on the `.edist2` file, and follow the instructions.

NOTE: You must enter the system passphrase in use at the time of backup. For hosts using Niagara 4.10u8 or Niagara 4.13.3 and later, you may use the alternative password if the backup channel was configured to use this instead of the passphrase.

The file will be decrypted to a new file with a `.dist` extension.

Step 5. Open the `.dist` file with a Zip tool.
You may need to change the file extension of the decrypted backup file from `.dist` to `.zip`.

Step 6. Extract the contents of the `.zip` file to a folder using the Zip tool.

Step 7. Copy the station from the expanded folder from the backup `niagara_user_home\stations\` to your Niagara Workbench User Home folder.

Step 8. Using the Niagara platform Station Copier tool, copy the station to the local or remote host.

Step 9. Copy the alarms and histories if needed, and click **Finish**.

Step 10. Start the new station on the local or remote host.
The station has been restored to an older version from backup and will connect to the cloud with the same identity.

Restoring Supervisor backup to a new computer

You wish to restore a previously backed-up version of a station to a new computer.

Prerequisites:

- The computer has the same Niagara and Niagara Cloud Service versions that were installed on the

computer where the backup was made.

- You know the system passphrase or password in use at the time of backup. If you do not know the system passphrase, you may need to perform extra steps. You may also use the alternative password if the backup channel was configured to use this instead of the system passphrase.
- Refer to the "*System and file passphrases*" and "*Editing the .bog file passphrase offline*" topics in the Niagara Platform Guide if necessary.

NOTE: The new restored station will have a new identity and will not assume the identity of the old station because you add and register a new Niagara Cloud Service identity.

Step 1. Install Workbench and copy the cloudLink modules to the Niagara home modules folder.

NOTE: Ensure that a Niagara license has been installed.

Step 2. Use the Niagara Cloud Management Portal to download the cloud backup for the Supervisor station.

Step 3. In Workbench > **My File System**, locate the backup file ending with the .edist2 file extension, double-click on the .edist2 file and follow the instructions.

NOTE: The encryption key must be entered, which is either the system passphrase or a password that was in use at the time of backup.

The file will be decrypted to a new file with a .dist extension.

Step 4. Open the .dist file with a Zip tool.

You may need to change the file extension of the decrypted backup file from .dist to .zip.

Step 5. Extract the contents of the .zip file to a folder using the Zip tool.

Step 6. Copy the station from the expanded folder from the backup `niagara_user_home\stations\` to your Niagara Workbench User Home folder.

Step 7. Using the Niagara platform Station Copier tool or Station Transfer wizard, copy the station to the local or remote host.

Step 8. Start the new station on the local or remote host.

Step 9. Open the station and navigate to **Config > Services** and locate the **NiagaraCloudService**.

Step 10. Delete this existing **NiagaraCloudService** as it will not work correctly on the new computer.

Step 11. To add the **NiagaraCloudService** to your station and register the device with Niagara Cloud Suite, follow the steps in the "Install and configure" topic of this guide.

Chapter 4. Components and other references

Components include services, folders and other model building blocks associated with a module. 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**

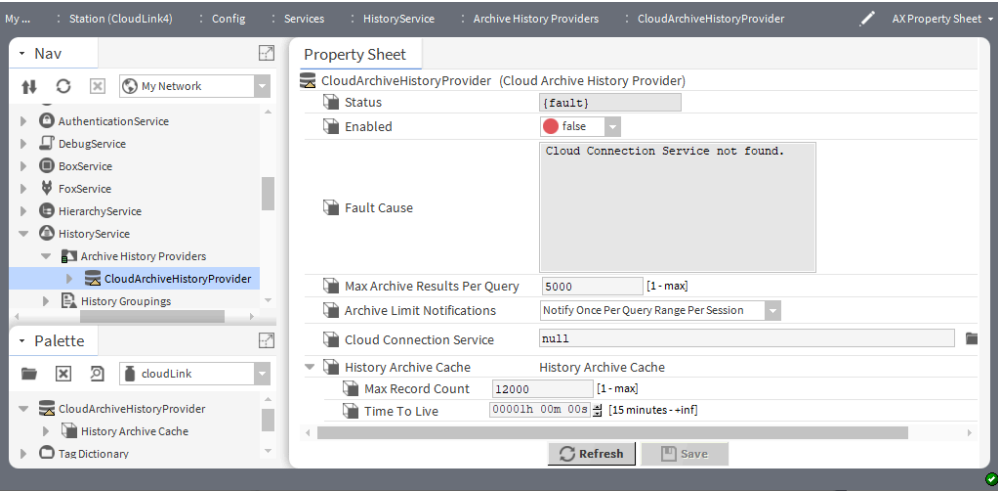
Also discussed in this section are cloud-specific event messages and commands, as well as extensibility.

cloudLink-CloudArchiveHistoryProvider

As of Niagara 4.13 and later, the CloudArchiveHistoryProvider component allows queries against local history records to be supplemented by archived history records that have been previously exported to a cloud platform using Niagara Cloud Service.

This component is automatically installed if you have a history channel and the historyArchive license feature, but you can also find it in the cloudLink palette. To access the component's properties, expand **Config > Services > HistoryService > ArchiveHistoryProviders**.

Figure 11. CloudArchiveHistoryProvider properties



Property	Value	Description
Status	read-only	<div>Indicates the condition of the CloudArchiveHistoryProvider at the last check.</div> <ul style="list-style-type: none">• {ok} indicates that the CloudArchiveHistoryProvider component is licensed• {disabled} indicates that the Enable property is set to false• {fault} indicates another

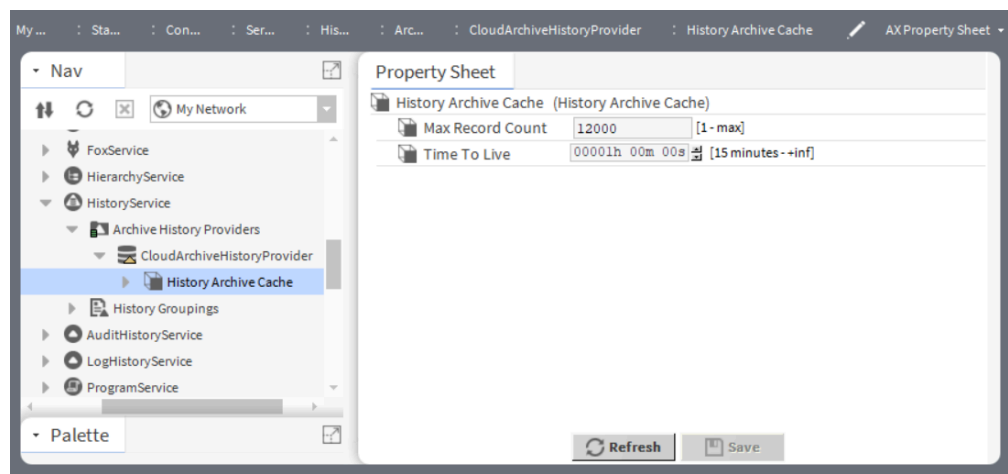
Property	Value	Description
		problem. Check the <code>FaultCause</code> property for more information
Enabled	<code>true</code> or <code>false</code> (default)	Activates and deactivates the use of the component and all its subcomponents.
Fault Cause	read-only	Indicates the reason why the <code>CloudArchiveHistoryProvider</code> is in fault. This field is empty unless a fault exists.
Max Archive Results Per Query	5000 (default)	The maximum number of history records to return for a query.
Archive Limit Notifications	<code>Notify Once Per Query Range Per Session</code> or <code>Never Notify</code> or <code>Always Notify</code>	Defines the behavior of a subset of Workbench views, but not all of them. <code>Web Chart</code> and <code>HTML5 History Table</code> views (accessible from the browser and Workbench) provide their own notification when a history query exceeds this limit.
Cloud Connection Service	<code>NiagaraCloudService</code> ord	Indicates the <code>NiagaraCloudService</code> to use to query for archived history records.
History Archive Cache	additional properties	Contains a set of properties for configuring the caching of history records queried from the cloud.

cloudLink-HistoryArchiveCache

`HistoryArchiveCache` is a subcomponent of the `CloudHistoryArchiveProvider` component.

To access the component's properties, expand **Config > Services > HistoryService > ArchiveHistoryProviders > CloudArchiveHistoryProvider**.

Figure 12. History Archive Cache



Property	Value	Description
Max Record Count	1-max (defaults to 12000)	The maximum number of history records to store in the cache.
Time To Live	hours, minutes, seconds (defaults to 1 hour)	The duration to keep an entry in the cache after its last access time.

Actions

Clear Cache: Removes all recorded data from the cache.

cloudLink-CertTrustMapping

CertTrustMapping is required to configure the station to receive commands sent from the cloud platform. This component is added to the Trust Manager in the CloudAuthenticationScheme in the AuthenticationService. This component is available in the Authentication folder in the cloudLink palette.

CertTrustMapping (Cert Trust Mapping)

App Id

Expected Jwt Audience

Expected Jwt Issuer

Certificate Alias

NiagaraCloudService

Type	Value	Description
App Id		Value of the Forge application id.
Expected Jwt Audience	CloudLink (default)	Value of the token audience "aud" field. By default, "Niagara Cloud Service", but this may be changed to match the value present in the JWT for those providers that do not have a fully configurable

Type	Value	Description
		audience field. For example, Salesforce prepends the Salesforce application Id (not to be confused with the Forge application Id) onto the audience.
Expected Jwt Issuer		Typically, the URL of the user identity provider. NOTE: This value is required. The value for the Token issuer "iss" field must to be provided by the Developer/Integrator during Certificate Trust Mapping configuration.
Certificate Alias		Alias of your token provider's public certificate file that was imported.

cloudLink-CloudAuthenticationScheme

An authentication scheme verifies that a user is authorized to access a station. Schemes are added to or removed from the AuthenticationSchemes container in the AuthenticationService. All authentication requests are routed through the system's AuthenticationService.

NOTE: When the NiagaraCloudService is added to the station the Cloud Authentication Scheme is added as well (if it is not already present).

cloudLink-CloudConnectionService

The NiagaraCloudService is the service that provides connectivity to the Niagara Cloud Suite. Most of the other components are located under the NiagaraCloudService.

The NiagaraCloudService is a specific implementation of the CloudConnectionService class that is designed to communicate with the Niagara Cloud Suite cloud platform services.

NOTE: NiagaraCloudService is a licensed feature, and requires a valid SMA.

Property Sheet

NiagaraCloudService

(Cloud Connection Service)

Status

{ok}

Fault Cause

Enabled

true

Platform Type

Forge

SMA Expiration Monitor

S M A Expiration Monitor

Mode

Warning

Warn Below

30

day [1 - 180]

Alarm Source Info

Alarm Source Info

Remaining

-1 day {stale}

Authenticators

Client Authenticators Folder

Transports

Transports Folder

Channels

Client Channels Folder

Property	Value	Description
Status	read-only	<p>Reports the condition of the entity or process at last polling.</p> <p>{ok} indicates that the component is licensed and polling successfully.</p> <p>{down} indicates that the last check was unsuccessful, perhaps because of an incorrect property, or possibly loss of network connection.</p> <p>{disabled} indicates that the Enable property is set to <code>false</code>.</p> <p>{fault} indicates another problem. Refer to Fault Cause for more information.</p>
Fault Cause	read-only	<p>Indicates the reason why a system object (network, device, component, extension, etc.) is not working (in fault). This property is empty unless a fault exists.</p>
Enabled	true or false	<p>Activates (<code>true</code>) and deactivates (<code>false</code>) use of the object (network, device, point, component, table, schedule, descriptor, etc.).</p>
Platform Type	read-only	<p>Indicates which cloud platform this instance of the NiagaraCloudService</p>

Property	Value	Description
		is configured to interact with.
SMA Expiration Monitor	additional properties	Contains a set of properties for configuring warning of impending SMA expiration.
Authenticators	folder	Contains one or more authenticators used with the cloud platform.
Transports	folder	Contains transports used to communicate with the cloud platform.
Channels	folder	Contains channels used to interact with the cloud platform.

SMA Expiration Monitor

The default configuration for these properties are configured to warn you when the current Software Maintenance Agreement (SMA) has 30 days or less remaining before it expires. If your organization requires more time to process license changes, you may wish to increase the expiration warning to avoid service interruption. The warning only appears in the station's **Application Director** output. To send an external notification, use an Alarm Source Info configuration to deliver a message to a recipient via email or mobile phone.

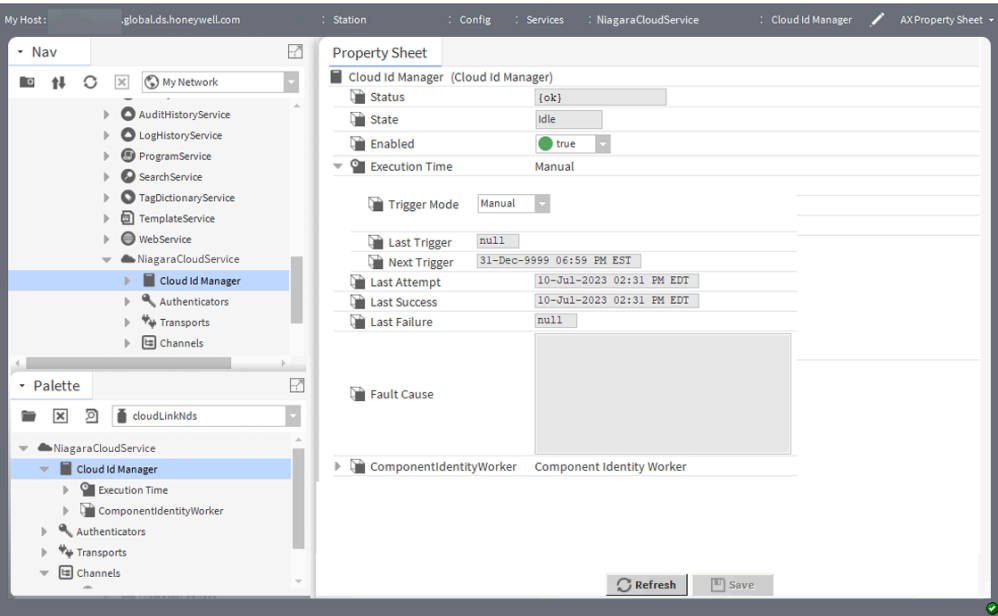
Property	Value	Description
Mode	drop-down list	<p>Determines when to raise awareness that the SMA is expiring.</p> <ul style="list-style-type: none"> Warning indicates that the monitor should raise an alarm if there are fewer than Warn Below days remaining in SMA. Expired indicates that the monitor should raise an alarm when the SMA expires. Disabled indicates that the monitor should not raise an alarm.
Warn Below	number of days [1–180] (defaults to 30)	Specifies the number of days before SMA expiration to raise an alarm.
Alarm Source Info	additional properties	<p>Contains a set of properties for configuring and routing alarms when this component is the alarm source.</p> <p>For property descriptions, refer</p>

Property	Value	Description
		to the <i>Niagara Alarms Guide</i>
Remaining	read-only	Indicates the days remaining in the current SMA.

cloudLink-CloudIdManager

The **Cloud Id Manager** component assigns cloud Ids to the components in the station and tracks cloud Ids for histories. As histories can be imported into the station without an existing component in the station, this allows for storing the history's cloud Id.

After you have marked any components to be excluded from the cloud upload using the nc:excluded tag, execute the **Cloud Id Manager** component as part of the initial onboarding process. It is recommended to run the **Cloud Id Manager** periodically, at least weekly or even daily, to pick up any new points that have been added to the station. You set the trigger time by selecting the desired **Trigger Mode** (Manual, Daily, Interval).



This component is a child of the NiagaraCloudService.

Property	Value	Description
Status	read-only	Indicates the condition of the Cloud Id Manager. <ul style="list-style-type: none">ok: indicates that the Cloud Id Manager component is operational.disabled: indicates that the Enable property is set to false.fault: indicates another problem. Check the Fault Cause property for more

Property	Value	Description
		information.
State	read-only	Indicates the current execution state of the Cloud Id Manager.
Enabled	true or false (defaults to true)	Activates or deactivates the component and all its subcomponents.
Trigger Mode	Manual, Daily, Interval (defaults to Manual)	Controls when the Cloud Id Manager should check the station for new components or histories that need to have a cloud Id assigned to them. Initially, this value is manual to provide an opportunity to get the station properly configured. Once the station has been configured, this value should be changed to periodically check for new components or histories.
Last Trigger	time	Reports time at which the Cloud Id Manager was last run.
Next Trigger	date, time	Reports day and time at which the Cloud Id Manager will run again.
Last Attempt	read-only	Timestamp of the last time the Cloud Id Manager tried to assign cloud Ids.
Last Success	read-only	Timestamp of the last successful execution.
Last Failure	read-only	Timestamp of the last failed execution.
Fault Cause	read-only	Indicates the reason why the Cloud Id Manager is in fault. This field is empty unless a fault exists.
Component Identity Worker	text	Displays the worker that does the actual assignment of cloud Ids. There can be different types of workers depending on how or to what you assign cloud Ids. Currently, the Component Identity Worker is the only worker that has been implemented.

Actions

Execute: Runs the Cloud Id Manager component.

cloudLink-CloudLinkAlarmRecipient

CloudLinkAlarmRecipient component is an extension to the standard **AlarmRecipient** and routes alarms to the cloud platform. It functions the same as the standard **AlarmRecipient** component does except that the **CloudLinkAlarmRecipient** component can be used to send alarms to a cloud platform. The **CloudLinkAlarmRecipient** component is automatically installed. It is important that you connect Alarm Classes to it to enable that alarms are routed to the cloud.

NOTE: A configured NiagaraCloudService object must exist in the station in order for the CloudLinkAlarmRecipient to function properly.

Figure 13. CloudLink Alarm Recipient properties

Property Sheet

CloudLinkAlarmRecipient (Cloud Link Alarm Recipient)

Time Range

12:00 AM - 12:00 AM

Start Time

12:00:00 AM EST

End Time

12:00:00 AM EST

Days Of Week

☒ Sun ☒ Mon ☒ Tue ☒ Wed ☒ Thu ☒ Fri ☒ Sat

Transitions

☒ toOffnormal ☒ toFault ☒ toNormal ☒ toAlert

Route Acks

☒ true

Cloud Connection Service

station:|slot:/Services/CloudConnectionService

Enable Batch Alarms

☒ Enabled

Alarm Batch Delay

+000000h 00m 30s

Alarm Batch Size

100 [2-512]

Last Sent To Cloud

21-Dec-2023 03:10 PM EST

Name	Value	Description
Time Range	start time and end time	Indicates the hours during which the alarms will be sent to the cloud.
Days of Week	check boxes	Indicates the days of the week alarms will be sent to the cloud.
Transitions	check boxes	Indicates the alarm transitions that will be sent to the cloud.
Route Acks	true (default) or false	Enables (true) and disables (false) the routing of alarm acknowledgements to the recipient. If it is set to false, trap acknowledgements are not routed.
Cloud Connection Service	null (default),	Indicates the NiagaraCloudService to route alarms through. This is automatically configured.
Enable Batch Alarms	Enabled (default), Disabled	Enables and disables the sending of multiple alarms to the cloud in one message. If set to true, alarms are accumulated and all the pending alarms are sent to

Name	Value	Description
		the cloud in one message. If set to <code>false</code> , each alarm is immediately sent to the cloud in a separate message.
Alarm Batch Delay	00000h 00m 30s (default)	Specifies the amount of time to delay before sending batch alarms to the cloud. During this delay, the alarms received are accumulated.
Alarm Batch Size	100 (default)	Specifies the maximum number of alarms to be included in a batch.
Last Sent To Cloud	null (default)	Shows the last time an alarm was sent to the cloud.

cloudLink-CloudTrustManager

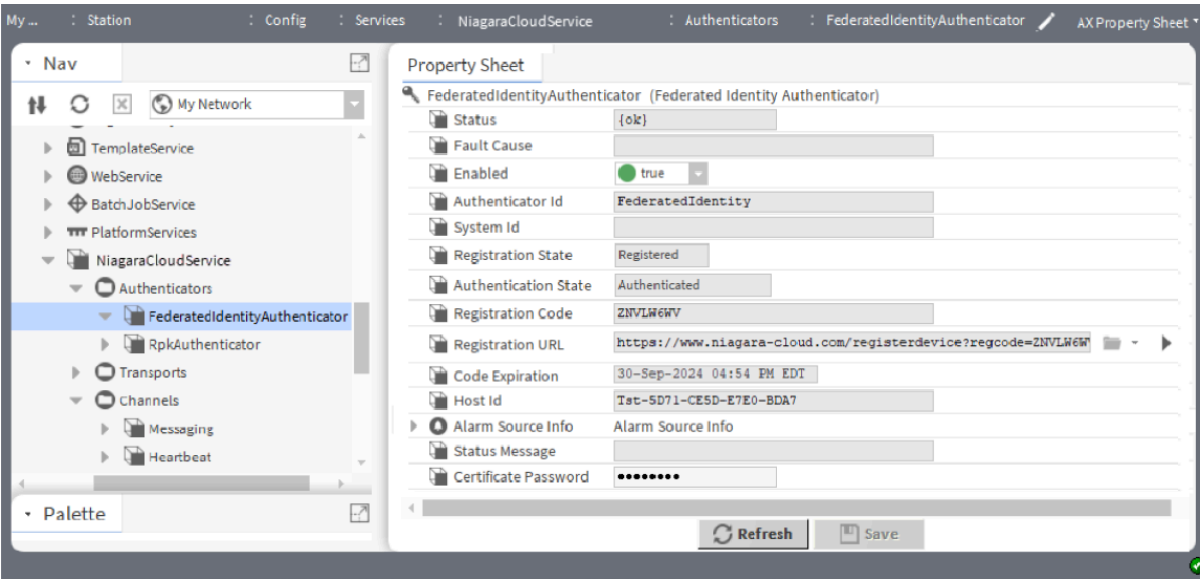
CloudTrustManager is a container for the added CertTrustMapping and JwksTrustMapping components.

cloudLink-FederatedIdentityAuthenticator

The FederatedIdentityAuthenticator is the authenticator for the Niagara Cloud Suite (NCS). It handles the station-side registration with the Federated Identity Service and provides a secure connection to the NCS identity provider.

This component is located in the cloudLinkNcs palette.

Figure 14. Federated Identity Authenticator properties



Property	Value	Description
Authenticator Id	read-only	The identity of the authenticator used by other Niagara Cloud Service components to identify this authenticator.
System Id	read-only	Contains a logical identifier for the station outside of the provisioning lifecycle, and is used for registering the system with the cloud platform.
Registration State	read-only	Current state of the device's registration with the cloud platform.
Authentication State	read-only	Current state of the authentication process with the cloud platform.
Registration Code	read-only	The code that you receive after you invoked the Start Registration action and which will be provided to the cloud.
Registration Url	read-only	The Url that you receive after invoking the Start Registration action. It is needed to complete the registration process.
Code Expiration	read-only	Specifies the expiration time of the code.
Host Id	read-only	Displays the alphanumeric code unique to the specific host.

Property	Value	Description
Alarm Source Info	additional properties	Configures how to handle alarms from the Federated Identity Authenticator. NOTE: Alarms are sent only when the <code>Alarm On Failure</code> property is set to <code>true</code> .
Status Message	read-only	Contains status messages during the registration process.
Certificate Password	text	Secures the RPK Authenticator certificate stored in the User Key Store to protect it from unauthorized access. The certificate is used to authenticate with the Forge identity provider. It is recommended to set a password and securely retain it as no password is set by default (as of Niagara 4.14)

Actions

Start Registration: Invokes the device registration process with Niagara Cloud Suite.

cloudLinkForge-RpkAuthenticator

This component handles device authentication to the Niagara Data Service and provides any necessary tokens that the device needs to send data to the cloud.

The authenticator is pre-configured with the items you need, such as the platform type and connection URLs. The different configurations needed for different platforms and environments are provided by specialized palettes.

This component is added automatically during provisioning. It is a sub component of the `NiagaraCloudService` and is automatically added when the federated device registration is complete.

Figure 15. Rpk Authenticator properties

Property Sheet

RpkAuthenticator (RpkAuthenticator)

Status	{ok}
Fault Cause	
Enabled	<input checked="" type="checkbox"/> true
Authenticator Id	RpkAuthenticator
System Id	N4:station:Tst-90ED-F599-7DE1-6C91
System Guid	ba52ef98-deld-4d31-9ee1-282537995189
System Type	pgsdemoniagara-cloud-customer-project-qa
Registration State	Registered
Authentication State	Authenticated
System Ownership Code	961892b7023bb1318c20105b80ede04aac396e72
Certificate Password	••••••••

The following properties support the `RpkAuthenticator`.

Property	Value	Description
Status	read-only	Indicates the condition of the authenticator at the last check: <ul style="list-style-type: none"> • {ok} indicates that the <code>RpkAuthenticator</code> component is licensed. • {disabled} indicates that the <code>Enable</code> property is set to <code>false</code>. • {fault} indicates another problem. Check the <code>FaultCause</code> property for more information.
Fault Cause	read-only	Indicates the reason why the <code>RpkAuthenticator</code> component is in fault. This field is empty unless a fault exists.
Enabled	true or false (defaults to true)	Activates and deactivates use of the component.
Authenticator Id	read-only	Identifies the authenticator. It is used by other Niagara Cloud Service components to identify this authenticator.
System Id	read-only	Contains a logical identifier for the station outside of the provisioning lifecycle, and is used in registering the system with the cloud platform's identity provider.
System Guid	read-only	Serves as a globally unique identifier for this device. The Identity Provider with which the connector is registered supplies this value.
System Type	read-only	Groups systems of similar origin/brand for data segregation purposes. You must configure this property before registering a device.
Registration State	read-only	Indicates the current state of device's registration with the identity provider.
Authentication State	read-only	Current state of the authentication process with the identity provider.
System Ownership Code	read-only	Proves physical ownership and/or possession of the device using a unique code.

Property	Value	Description
Certificate Password	text	Secures the RPK Authenticator certificate stored in the User Key Store to protect it from unauthorized access. The certificate is used to authenticate with the Forge identity provider. It is recommended to set a password and securely retain it as no password is set by default (as of Niagara 4.14)

Action

Reauthenticate forces the authenticator to immediately authenticate with the cloud platform again instead of authenticating according to its normally-scheduled refresh.

cloudLink-AmqpTransport

This transport handles sending and receiving data with the AMQP messaging protocol.

AMQP Transport

Amqp Transport

Status

{ok}

Fault Cause

Enabled

true

Message Retries

2

[0 - 10]

Compression

Gzip

Message Throttling Limit

5

[0 - max]

Default Message Timeout

00000h 01m 00s

[1second - 5 minutes]

Authenticator Id

RpkAuthenticator

Status Message

Connected

Connect Retry Interval

+00000h 00m 20s

Connection Type

AMQP WebSocket

Type	Value	Description
Status	read-only	<div>Indicates the condition of the AmqpTransport.</div> <ul style="list-style-type: none">• {ok} indicates that the AmqpTransport component is successfully connected.• {down} indicates that the AmqpTransport is not connected to the Cloud platform, perhaps it is not registered, or possibly loss of network connection.• {disabled} indicates that the Enable property is set to false.• {fault} indicates another problem. Check the Fault

Type	Value	Description
		Cause property for more information.
Fault Cause	read-only	Indicates the reason why the AmqpTransport component is in fault. This field is empty unless a fault exists.
Enabled	true or false	Activates and deactivates use of the component.
Message Retries	2 (default)	The number of times a failed message should re-attempt delivery before notifying the sender that the message cannot be delivered.
Compression	GZip (default), None	This is used to specify the type of compression to use with this transport.
Message Throttling Limit	5 (default), 0 (no limit)	Maximum number of messages/second from station to cloud platform. Default is recommended for Forge starter environment instance.
Default Message Timeout	0000h 01m 00s(default)	Defines the waiting time before the system times out on messages sent using AMQP client.
Authenticator Id	read-only	The identity of the authenticator to use to get authentication information.
Status Message	read-only	The current state of the AMQP connection.
Connect Retry Interval	0000h 00m 20s (default)	Amount of time between attempts to establish a connection upon a failure to connect.
Connection Type	AMQP WebSocket (default), AMQP	Specify the type of connection to make to the cloud platform. The AMQP WebSocket transport option connects to the cloud platform using the HTTPS web port 443. The AMQP transport option connects to the cloud platform on port 5671. In both

Type	Value	Description
		<p>case the connection is established using TLS.</p> <p>NOTE: You will need to configure firewalls to allow outbound connections from Niagara Stations to the hosts specified in the prerequisites for the procedure, "Setting up device internet access", and for the port indicated by the selected Transport value (either AMQP WebSocket or AMQP).</p>

Actions

Reconnect – Forces the AMQP transport to close its existing connection and reconnect to the cloud platform.

cloudLink-HttpTransport

This transport handles sending data with the HTTP protocol.

The screenshot shows the configuration for the HTTP Transport. It includes fields for Status (set to {ok}), Fault Cause (empty), Enabled (checked), Message Retries (set to 2), Compression (set to None), Message Throttling Limit (set to 0), Connect Timeout (set to 15000 ms), Read Timeout (set to 30000 ms), and Write Timeout (set to 15000 ms).

In addition to the standard properties (Status, Fault Cause and Enabled), these properties are unique to the Histories Channel.

Property	Value	Description
Message Retries	2 (default)	Configures the number of times a failed message should re-attempt delivery before notifying the sender that the message cannot be delivered.
Compression	GZip, None (default),	Specifies the type of compression to use with this transport.
Message Throttling Limit	0 (default, no limit)	Defines the maximum number of messages/second from station to cloud platform.
Connect Timeout	15000 (default)	Specifies the time in milliseconds the

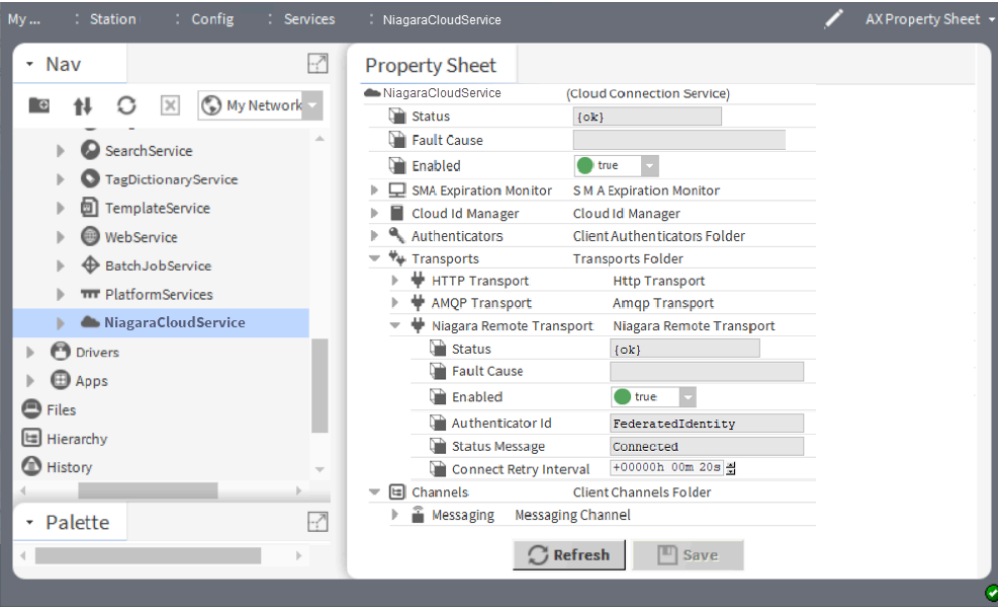
Property	Value	Description
		transport should wait when establishing a connection.
Read Timeout	30000 (default)	Specifies the time in milliseconds the transport should wait when reading a response.
Write Timeout	15000 (default)	Specifies the time in milliseconds the transport should wait when writing a request.

cloudLink-NiagaraRemoteTransport

This component allows you to directly connect to the station virtual machine (VM), which runs on premise through Niagara Cloud Suite without requiring a separate on-premise VPN installation. **Niagara Remote Transport** enables the station to communicate with the Niagara Remote server thereby allowing client browsers to establish a Fox session with the station for browser-based station configuration and viewing. After you have purchased the Niagara Remote service, it installs automatically when you register the station in Niagara Cloud Suite.

The Niagara Cloud Suite Device Provisioning Service performs the component configuration for you. You can disable the Niagara Remote connectivity by setting the **Enabled** property to `False`.

Once the station is onboarded, the **Niagara Remote Transport** component is available under **NiagaraCloudService > Transports**.



Property	Value	Description
Status	read-only	Indicates the condition of the Niagara Remote Transport . <ul style="list-style-type: none">• {ok} indicates that the transport is successfully connected.• {down} indicates that the transport is not connected

Property	Value	Description
		<p>to the Cloud platform, perhaps because it is not registered or has lost a network connection</p> <ul style="list-style-type: none"> • {disabled} indicates that the Enable property is set to false. • {fault} indicates another problem. Check the Fault Cause property for more information.
Fault Cause	read-only	Indicates the reason why the Niagara Remote Transport component is in fault. This field is empty unless a fault exists.
Enabled	true (defaults to true) or false	Activates and deactivates use of the component.
Authenticator Id	read-only	Indicates the identity of the authenticator, which is used to get authentication information.
Connect Retry Interval	hours, minutes, seconds (defaults to 0000h 00m 20s)	Specifies the amount of time between attempts to establish a connection after failing to connect.

Actions

Reconnect: re-establishes a connection if needed.

cloudLink-AlarmsChannel

This channel handles alarm delivery to the cloud platform.

Alarms (Alarms Channel)

Status {ok}

Fault Cause

Enabled true

Channel Type Alarm

Channel Config Amqp Alarm Channel Config

In addition to the standard properties (Status, Fault Cause and Enabled), these properties support the **AlarmsChannel**.

Property	Value	Description
Channel Type	read-only	Identifies the type of channel.
Channel Config	additional properties	Contains specific channel

Property	Value	Description
configuration information.		

cloudLinkForge-ForgeAmqpAlarmChannelConfig

This channel config holds information for the alarms channel when used with the Forge platform and communicating over the AMQP transport.

Property Sheet

Channel Config (Forge Amqp Alarm Channel Config)

Transport Information

Transport Config

Transport Type

AMQP

Authenticator Id

RpkAuthenticator

Channel Queue

In Memory Message Queue

Queue Size

5000000

[1024 - max]

Weight

1

[1 - max]

Max Message Size

1000000

B [10000 - 10000000]

Property	Value	Description
Transport Information	additional properties	Contains the AMQP-specific part of the configuration.
Transport Type	read-only	Identifies which transport should be used with this configuration
Authenticator Id	read-only	Identifies which authenticator should be used, note this is only needed for connectionless transports.

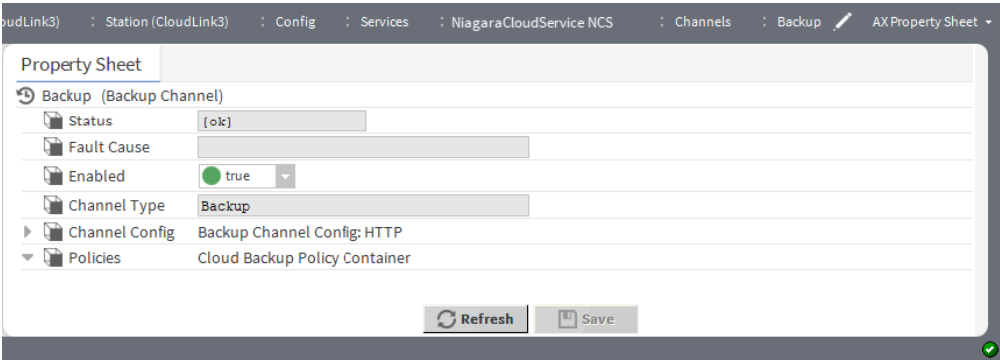
Property	Value	Description
Channel Queue	additional properties	Contains the Queue information to use for outgoing messages.
Queue Size	number (defaults to 5000000)	Defines amount of data (bytes) a queue can hold before it starts to reject additional messages.
Weight	number (defaults to 1)	<div>Configures the relative weight of the queue in the round robin scheduling algorithm that selects the next message for the transport to send.</div> <div>The transport layer uses a weighted round robin approach to accept messages, so increasing the weight on one queue cause its messages to be sent more frequently than messages from other queues.</div>

Property	Value	Description
		This may be useful to ensure that certain message types like alarms are sent with priority over telemetry data. Do not modify these parameters unless you determine that needed messages are being held up by less important ones.
Max Message Size	1000000 (default)	Defines the message size limit, before compression, to use with this channel and transport.

cloudLink-BackupChannel

This channel uploads station backups to the cloud according to the configuration in the backup policy.

This component is automatically added by the provisioning service, which tells the station what to configure.

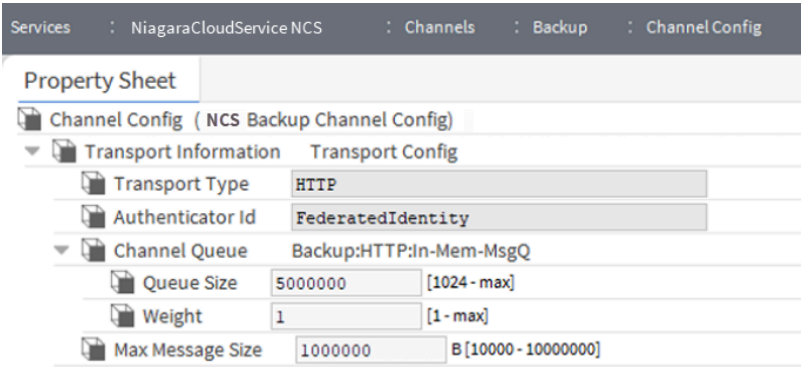


In addition to the standard properties (Status, Fault Cause, and Enabled), these properties are unique to the BackupChannel.

Property	Value	Description
Channel Type	read-only	Specifies the type of channel.
Channel Config	additional properties	Contains specific channel configuration information for cloud platform.
Policies	additional properties	Contains properties for backup policies.

cloudLinkNcs-NcsBackupChannelConfig

The ChannelConfig holds information for the backup channel when used with the NCS platform.



Properties specific to Transport Config

Property	Value	Description
Transport Type	read-only	Identifies which transport should be used with this configuration.
Authenticator Id	read-only	Identifies which authenticator should be used.
Channel Queue	additional properties	Contains the queue information to use for outgoing messages.
Max Message Size	[10000–10000000] (defaults to 1000000)	Limits the message size in bytes (before compression) to use with this channel and transport.

Properties specific to InMemoryMessageQueue

Property	Value	Description
Queue Size	[1024–max] (defaults to 5000000)	Specifies the amount of data in bytes the queue can hold before it starts to reject additional messages.
Weight	[1–max] (defaults to 1)	Specifies the relative weight of the queue in the round robin scheduling algorithm that selects the next message from the transport to send. The higher the number, the more times messages from this queue will be processed relative to other lower weighted queues.

cloudLink-CloudBackupPolicyContainer

This container holds backup policies which define the execution time of the backup and files and folders to exclude from the backup.

Property Sheet

Policies (Cloud Backup Policy Container)

Default Backup Policy

Cloud Backup Policy

Retry Trigger

15 minutes {Sun Mon Tue Wed Thu Fri Sa...

Interval

00000h 15m 00s [1 ms - +inf]

Trigger Mode

Interval

Time Of Day

☐ Start Time12:00:00 AM EDT

End Time11:59:59 PM EDT

Days Of Week

☒ Sun☒ Mon☒ Tue☒ Wed☒ Thu☒ Fri☒ Sat

Last Trigger

05-Jun-2024 04:24 PM EDT

Next Trigger

05-Jun-2024 04:39 PM EDT

Property	Value	Description
Default Backup Policy	backup policy container	Contains policy properties and is installed by default.
Retry Trigger	time (defaults to 15 minutes)	Defines the time interval a backup is rerun after a failure. NOTE: If a backup has failed when the station is stopped, the backup will run according to the retry trigger which may be immediately.

Properties specific to Default Backup Policy

In addition to the standard properties (Status, Fault Cause and Enabled), these properties support the **Default Backup Policy**.

Property Sheet

Default Backup Policy (Cloud Backup Policy)

Status

{ok}

State

Idle

Enabled

●

true

Execution Time

2:00 AM {Sun} +~1 hour

Time Of Day

02:00:00 AM EDT

Randomization

+00001h 00m 00s

Days Of Week

☒Sun

☐Mon

☐Tue

☐Wed

☐Thu

☐Fri

☐Sat

Last Trigger

03-Jun-2024 06:03 PM EDT

Next Trigger

09-Jun-2024 02:37 AM EDT

Last Attempt

03-Jun-2024 06:03 PM EDT

Last Success

03-Jun-2024 06:03 PM EDT

Last Failure

23-May-2024 09:57 AM EDT

Fault Cause

Backup Note

Encryption Key

System Passphrase

Encryption Key Type

System Passphrase

Password

●●●●●●

Confirm

●●●●●●

Exclude Files

.lock;.backup*;console.*;config.bog.b*;

Exclude Folders

file:^^webFileCache

file:^^cloudLinkModel

file:^^cloudLinkHistory

file:^^orientSystemDb

Alarm On Failure

●

true

Alarm Source Info

Alarm Source Info

Initial Retry Interval

1

[1 - max]

Max Retry Interval

96

Property	Value	Description
Execution Time	time (defaults to 2 a.m. on Sundays)	Defines the execution time. You can set it to up manually, daily or at a time interval.
Last Attempt	date, time (read-only)	Indicates the last time the backup was run.
Last Success	date, time (read-only)	Indicates the last time the backup was successful.
Last Failure	date, time (read-only)	Indicates the last time the backup failed.
Fault Cause	read-only	Indicates the reason why the Backup Policy failed. The field stays empty unless a backup failed at some point in the past.
Backup Note	text	Defines the text (note) that will be associated with

Property	Value	Description
		the backup in the cloud.
Encryption Key	additional properties	Indicates the selected encryption key type used to encrypt the backup prior to uploading it to the cloud. NOTE: If the encryption key is changed, subsequent backups will use the new encryption key. Previous backups in the cloud still use the prior encryption key, which was in use at the time the backups were made.
Encryption Key Type	drop-down menu	Specifies the type of the encryption key, which is either the <code>System Passphrase</code> or the <code>Password</code> . If you select the password, you are prompted to enter the custom password into the <code>Password</code> property below.
Password	Secure text with confirmation	If you select <code>Password</code> for Encryption Key Type , a password, which meets default security standards, must be entered. In the <code>Confirm</code> field, you must re-enter the created password a second time.
Excluded Files	file names	Contains files separated by a semicolon with extensions either with the complete name or using the wildcard asterisk (*). The default Backup Policy contains recommended files to exclude (<code>*.lock;*.backup*;console.*;config.bog.b*;config_backup*</code>).
Excluded Folders	folder name	Contains a list of folders. The default Backup Policy contains recommended folders to exclude which are used for system temporary files (<code>file:^^webFileCache, file:^^cloudLinkModel, file:^^cloudLinkHistory</code>).
Alarm On Failure	<code>true</code> or <code>false</code> (defaults to <code>true</code>)	If <code>true</code> , alarms will be created for backup failures otherwise no alarms will be created.
Alarm Source Info	additional properties	Contains standard Niagara alarm configuration information. For more information, see Niagara Alarms Guide)
Initial Retry Interval	number (defaults to 1)	Defines the starting number of intervals before failed backups are retried. Failed backups are retried periodically at intervals that increase over time so that the frequency of backup attempts decreases. For subsequent backup failures, the cumulative count is multiplied by a fixed factor of 2 so that the backups attempts are performed less frequently over time. The first backup is tried after 1 interval, the next at 2 intervals, the next at 4 intervals, and so on. Once a backup is successful, the internal count is reset to the initial retry interval.
Max Retry Interval	number	Specifies the maximum number of intervals to wait before attempting another backup. When the

Property	Value	Description
		internal count reaches this limit without a successful backup, all subsequent backup attempts will be at this number of intervals.

Actions

Execute: Invokes a manual backup to the cloud.

cloudLink-CommandsChannel

This channel handles command and control functions from the cloud platform.

System commands

The cloud platform can send system commands down to the station. Handling for these system commands is done by extending a specific class. This is done via the CommandsChannel.

When a system command is received via a connected transport, the CommandsChannel is called to determine if it has a registered command to handle the incoming message. If there is one, the appropriate registered command is called to process the message.

Currently, there are 11 commands to read/write to the points that are in Niagara Cloud Service. The commands are listed here.

- RetrieveCloudPointsCommand — Lists the names of all the points in the station that are accessible from the cloud.
- CloudMultiPointReadCommand — Returns the values of a list of cloud accessible points in the station.
- CloudMultiPointWriteCommand — Sets the values of a list of cloud accessible points in the station.
- CloudPointReadCommand — Returns the values of an individual cloud accessible point in the station.
- CloudPointWriteCommand — Sets the value of an individual cloud accessible point in the station.
- ReadMultiPointInputsCommand — Returns all the cloud inputs for a given set of points in the station.
- CloudMultiPointClearCommand — Releases the values of a list of cloud accessible points in the station that were previously set with a CloudPointWriteCommand or CloudMultiPointWriteCommand.
- AlarmAckCommand — Acknowledges an alarm.
- BatchAlarmAckCommand — Acknowledges a list of alarms.
- ListCloudCommandsCommand — Lists the names of all commands that are available on this system.
- InvokeCommand - allows invocation of an action on a component.

With Cloud Command Queues the JACE/Supervisor now responds as soon as the command has been placed in the queue. When the command executes its output is sent to the cloud via NewEventMessage(s) and when the command exits another NewEventMessage is sent.

The CloudCommandsDeviceExt must be enabled before any individual command can run.

Custom commands

In addition to providing many "out-of-the-box" commands, Niagara Cloud Service also provides the capability to handle custom commands. This ability to invoke custom code provides greater flexibility, however, there are multiple restrictions and security requirements.

In addition to the standard properties (Status, Fault Cause and Enabled), these properties are unique to the Commands Channel.

Property	Value	Description
Channel Type	read-only	Identifies the type of channel.
Channel Config	additional properties	Contains specific channel configuration information.
Commands	additional properties	Holds all the commands that have been installed for this cloud platform.
Default Command Queue	additional properties	Identifies the queue for pending commands that have been sent to the station. Additional command queues can be added.

Channel configuration information

For channel configuration details, see “cloudLinkForge-ForgeAmqpCommandChannelConfig”.

Property	Value	Description
Command Timeout	+00000h 01m 00s (default)	Defines the maximum time to wait for a command to complete.

Commands information

Each individual Command property provides the **Enabled** subproperty which is configured either `true` or `false`.

InMemoryCommandQueue properties

Command queues hold pending commands that have been sent to the station. Commands are sent with a command priority between 1 and 255 with lower numeric values being more important than higher values. Commands are placed in the queue with the lowest number higher than the command priority. By default there is one queue; however, additional command queues can be added.

Property	Value	Description
Priority	number (defaults to 255)	Establishes the priority of the queue. Queues with a lower numeric priority are drained before queues with higher numeric values.
Max Size	number (defaults to 20)	Defines the number of pending commands.

cloudLink-ForgeAmqpCommandChannelConfig

This channel config holds information for the commands channel when used with the Forge platform and communicating over the AMQP transport.

Channel Config (Forge Amqp Command Channel Config)

Transport Information

Transport Config

Transport Type

AMQP

Authenticator Id

RpkAuthenticator

Channel Queue

In Memory Message Queue

Queue Size

5000000

[1024 - max]

Weight

1

[1 - max]

Max Message Size

1000000

B [10000 - 10000000]

Property	Value	Description
Transport Information		Contains the AMQP specific part of the Transport configuration.
Transport Type	read-only	Identifies which transport should be used with this configuration.
Authenticator Id	read-only	Identifies which authenticator should be used, note this is only needed for connectionless transports.
Channel Queue	additional properties	Contains the InMemoryMessageQueue. queue information to use for outgoing messages.
Queue Size	number (defaults to 1)	Defines amount of data (bytes) a queue can hold before it starts to reject additional messages.
Weight	1000000 (default)	<p>Configures the relative weight of the queue in the round robin scheduling algorithm that selects the next message for the transport to send.</p> <p>The transport layer uses a weighted round robin approach to accept messages, so increasing the weight on one queue cause its messages to be sent more frequently than messages from other queues. This may be useful to ensure that certain message types like alarms are sent with priority over telemetry data. Do not modify these parameters unless you determine that needed messages are being held up by less important ones.</p>

Property	Value	Description
Max Message Size	1000000 (default)	Defines the message size limit (bytes), before compression, to use with this channel and transport.

cloudLink-EventsChannel

This channel handles event delivery to the cloud platform. Note that if the station does not currently have the event service installed the events channel will add it to the station.

The screenshot shows the configuration for the 'Events (Events Channel)' property. It includes the following fields:

- Status:** {ok}
- Fault Cause:** (empty text field)
- Enabled:** true (with a dropdown arrow)
- Channel Type:** Event
- Channel Config:** Forge Amqp Event Channel Config

In addition to the standard properties (Status, Fault Cause and Enabled), these properties are unique to the EventsChannel.

Property	Value	Description
Channel Type	read-only	Identifies the type of channel.
Channel Config		Contains specific channel configuration information.

Event messages

An event message is a method of pushing data to the cloud. Specific logic triggers an event message. For example, an alarm triggers a NewAlarm event message. The NiagaraCloudService contains channels to accomplish such things as getting message types to send to the cloud. The types of event messages are listed here:

- NewAlarms - sent to the cloud provider
- AlarmAckRequests - from the cloud provider
- AlarmAckResponses - to the cloud provider
- As well as others such as DeviceAckRequests.

cloudLinkForge-ForgeAmqpEventChannelConfig

This channel config holds information for the events channel when used with the Forge platform and communicating over the AMQP transport.

Channel Config (Forge Amqp Event Channel Config)

Transport Information

Transport Config

Transport Type

AMQP

Authenticator Id

RpkAuthenticator

Channel Queue

In Memory Message Queue

Queue Size

5000000

[1024 - max]

Weight

1

[1 - max]

Max Message Size

1000000

B [10000 - 10000000]

Property	Value	Description
Transport Information	read-only	Contains the AMQP specific part of the configuration.
Transport Type	read-only	Identifies which authenticator should be used, note this is only needed for connectionless transports.
Authenticator Id	read-only	Identifies which authenticator should be used, note this is only needed for connectionless transports.
Channel Queue	additional properties	Contains the queue information to use for outgoing messages.
Queue Size	number (defaults to 1)	Defines amount of data (bytes) a queue can hold before it starts to reject additional messages.
Weight	number (defaults to 1000000)	<p>Configures the relative weight of the queue in the round robin scheduling algorithm that selects the next message for the transport to send.</p> <p>The transport layer uses a weighted round robin approach to accept messages, so increasing the weight on one queue cause its messages to be sent more frequently than messages from other queues. This may be useful to ensure that certain message types like alarms are sent with priority over telemetry data. Do not modify these parameters unless you determine that needed messages are being held up by less important ones.</p>
Max Message Size	number (defaults to 1000000)	Defines the message size limit

Property	Value	Description
		(bytes), before compression, to use with this channel and transport.

cloudLink-HeartbeatChannel

This channel delivers heartbeat messages to the cloud platform at regular intervals.

Heartbeat (Heartbeat Channel)

Status

{ok}

Fault Cause

Enabled

true

Channel Type

Heartbeat

Channel Config

Forge Amqp Heartbeat Channel Config

Frequency

00000h 05m 00s

[1 minute - 1 day]

In addition to the standard properties (Status, Fault Cause and Enabled), these properties are unique to the HeartbeatChannel.

Property	Value	Description
Channel Type	read-only	Identifies the type of channel.
Channel Config	additional properties	Contains specific channel configuration information.
Frequency	hours minutes seconds (defaults to 00000h 05m 00s)	Specifies how often the channel should generate a heartbeat message.

cloudLinkForge-ForgeAmqpHeartbeatChannelConfig

This channel config holds information for the heartbeat channel when used with the Forge platform and communicating over the AMQP transport.

Channel Config (Forge Amqp Heartbeat Channel Config)

Transport Information

Transport Config

Transport Type

AMQP

Authenticator Id

RpkAuthenticator

Channel Queue

In Memory Message Queue

Queue Size

5000000

[1024 - max]

Weight

1

[1 - max]

Max Message Size

1000000

B [10000 - 10000000]

Property	Value	Description
Transport Information		Contains the AMQP specific part of the Transport configuration.

70

January 8, 2025

Property	Value	Description
Transport Type	read-only	Identifies which transport should be used with this configuration.
Authenticator Id	read-only	Identifies which authenticator should be used. Note this is only needed for connected transports.
Channel Queue		Contains the queue size and weight information to use for outgoing messages.
Queue Size	number (defaults to 1)	Defines amount of data (bytes) a queue can hold before it starts to reject additional messages.
Weight	1000000 (default)	<p>Configures the relative weight of the queue in the round robin scheduling algorithm that selects the next message for the transport to send.</p> <p>The transport layer uses a weighted round robin approach to accept messages, so increasing the weight on one queue cause its messages to be sent more frequently than messages from other queues. This may be useful to ensure that certain message types like alarms are sent with priority over telemetry data. Do not modify these parameters unless you determine that needed messages are being held up by less important ones.</p>
Max Message Size	number (defaults to 1000000)	Defines the message size limit (bytes), before compression, to use with this channel and transport.

cloudLink-HistoriesChannel

This channel handles history delivery to the cloud platform.

The screenshot displays the configuration window for the 'Histories (Histories Channel)' component. The interface includes several sections:

- Status:** A text field containing '[ok]'.
- Fault Cause:** An empty text field.
- Enabled:** A toggle switch set to 'true'.
- Channel Type:** A dropdown menu showing 'History'.
- Channel Config:** A section titled 'Forge History Channel Config'.
- Exports:** A section titled 'Cloud History Export Config Container'.
 - Max Concurrent Export Executions:** A numeric field set to '10' with a '[2-max]' limit.
 - Retry Trigger:** A section with a '15 minutes' interval and a 'Sun Mon Tue Wed Thu Fri Sa...' schedule.
 - Trigger Mode:** A dropdown menu set to 'Interval'.
 - Time Of Day:** A section with 'Start Time' (12:00:00 AM EST) and 'End Time' (11:59:59 PM EST).
 - Days Of Week:** A row of checkboxes for Sun, Mon, Tue, Wed, Thu, Fri, and Sat, all of which are checked.
 - Last Trigger:** A timestamp field showing '18-Feb-2021 04:41 PM EST'.
 - Next Trigger:** A timestamp field showing '18-Feb-2021 04:56 PM EST'.
- Auto Export:** A section titled 'Cloud History Auto Export'.

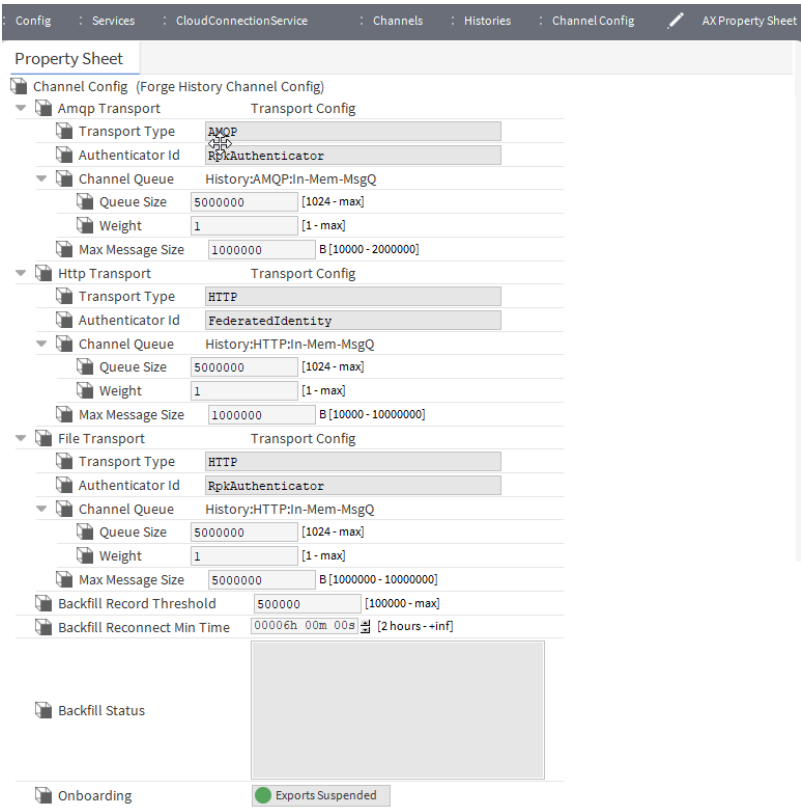
To access these properties, double-click the **Histories Channel** component.

In addition to the standard properties (Status, Fault Cause and Enabled), these properties are unique to the **HistoriesChannel**.

Property	Value	Description
Channel Type	read-only	Identifies the type of channel.
Channel Config	additional properties	Contains specific channel configuration information.
Exports	additional properties	Provides a container for History export policies.

cloudLinkForge-ForgeHistoryChannelConfig

This component holds information for the histories channel when used with the Niagara Cloud Suite platform.



To access these properties, expand **Histories** and double-click **Channel Config**.

Property	Value	Description
Amqp Transport	additional properties	Contains the AMQP-specific properties of the configuration.
Http Transport	additional properties	Contains the HTTP-specific properties of the configuration.
File Transport	additional properties	Contains the file upload-specific properties of the configuration.
Backfill Record Threshold	number (defaults to 500000)	<p>Defines the number of pending history records at which point the station sends histories with the file upload transport rather than the AMQP Transport.</p> <p>Sending large numbers of histories with a file upload is much more efficient than sending AMQP messages. Once the backfill of histories is completed, the station resumes sending histories with the AMQP transport.</p>

Property	Value	Description
Backfill Reconnect Min Time	hours minutes seconds	Configures the minimum amount of time that the station must be disconnected before the station checks for backfill. For example, if this is set to 2 hours, and the station is only disconnected for 30 minutes, it performs no backfill check is performed. The station refers to this property only when the transport is disconnected. When the station starts, it always performs a backfill check.
Backfill Status	read-only	Indicates the status of a backfill operation which is run when the number of pending records exceeds the <code>Backfill Record Threshold</code> value. The histories are sent to the cloud by bulk upload rather than with the AMQP Transport. The bulk upload mechanism is designed to support large numbers of histories. When a backfill is running, no histories are sent to the cloud with the AMQP transport and the Onboarding section will indicate <code>Exports Suspended</code> .

TransportConfig properties

Property	Value	Description
Transport Type	read-only	Identifies which transport should be used with this configuration.
Authenticator Id	read-only	Identifies which authenticator should be used. NOTE: This is only needed for connectionless transports.
Channel Queue	additional properties	This contains the queue information to use for outgoing messages.
Max Message Size	number (defaults to 1000000)	Defines the message size limit (bytes) before compression to use with this channel and transport.


In-Mem-MsgQ (InMemoryMessageQueue) properties


Property	Value	Description
Queue Size	number (defaults to 5000000)	Defines the amount of data in bytes the queue can hold before it starts to reject additional messages.
Weight	number (defaults to 1)	Defines the relative weight of the queue in the round robin scheduling algorithm that selects the next

Property	Value	Description
		message from the transport to send. The higher the number, the more times messages from this queue will be processed relative to other lower weighted queues.


cloudLink-MessagingChannel


This channel delivers messages to the cloud platform that have already been serialized.

 Messaging (Messaging Channel)


 Status

{ok}


 Fault Cause

 Enabled

true

 Channel Type

Messaging

 Channel Config

Forge Message Channel Config

In addition to the standard properties (Status, Fault Cause and Enabled), these properties are unique to the **MessageChannel**.

Property	Value	Description
Channel Type	read-only	Identifies the type of channel.
Channel Config	additional properties	Contains specific channel configuration information.

cloudLinkForge-ForgeMessagingChannelConfig

Forge Messaging Channel Config

Properties

Property Sheet

Channel Config (Forge Message Channel Config)	
Authenticator Id	FederatedIdentity
HTTP Transport Config	
Transport Type	HTTP
Authenticator Id	FederatedIdentity
Channel Queue Messaging:HTTP:In-Mem-MsgQ	
Queue Size	5000000 [1024 - max]
Weight	1 [1 - max]
Max Message Size	1000000 B [10000 - 10000000]
AMQP Transport Config	
Transport Type	AMQP
Authenticator Id	RpkAuthenticator
Channel Queue Messaging:AMQP:In-Mem-MsgQ	
Queue Size	5000000 [1024 - max]
Weight	1 [1 - max]
Max Message Size	1000000 B [10000 - 2000000]

Property	Value	Description
Authenticator Id	read-only	Identifies which authenticator should be used, note this is only needed for connectionless transports.
HTTP	additional properties	This contains the HTTP specific part of the configuration.
AMQP	additional properties	This contains the AMQP specific part of the configuration.

TransportConfig properties

All of these properties are specific to the TransportConfig.

Property	Value	Description
Transport Type	read-only	Identifies which transport should be used with this configuration.
Authenticator Id	read-only	Identifies which authenticator should be used, note this is only needed for connectionless transports.
Channel Queue	additional properties	This contains the Queue information to use for outgoing messages.
Max Message Size	number (defaults to 1000000)	Defines the message size limit (bytes), before compression, to use with this channel and

Property	Value	Description
		transport.

InMemoryMessageQueue properties


All of these properties are specific to the InMemoryMessageQueue.


Property	Value	Description
Queue Size	number (defaults to 1)	Defines amount of data (bytes) a queue can hold before it starts to reject additional messages.
Weight	1000000 (default)	<p>Configures the relative weight of the queue in the round robin scheduling algorithm that selects the next message for the transport to send.</p> <p>The transport layer uses a weighted round robin approach to accept messages, so increasing the weight on one queue cause its messages to be sent more frequently than messages from other queues. This may be useful to ensure that certain message types like alarms are sent with priority over telemetry data. Do not modify these parameters unless you determine that needed messages are being held up by less important ones.</p>

cloudLink-ModelChannel


This channel handles model export to the cloud platform.


Property Sheet

 Model (Model Channel)


 Status

{ok}


 Fault Cause

 Enabled


true

 Channel Type

Model

 Channel Config

Forge Model Channel Config

 Policies

Model Export Policy Container

In addition to the standard properties (Status, Fault Cause and Enabled), these properties are unique to the ModelChannel.

Property	Value	Description
Channel Type	read-only	Identifies the type of channel.
Channel Config	additional properties	Contains specific channel configuration information.
Policies	additional properties	Container for different model export strategies.

cloudLinkForge-ForgeModelChannelConfig

This channel config holds information for the model channel when used with the Forge platform.

Properties

Property Sheet

Channel Config (Forge Model Channel Config)

Http Transport Transport Config

Transport Type HTTP

Authenticator Id RpkAuthenticator

Channel Queue Model:HTTP:In-Mem-MsgQ

Queue Size 5000000 [1024 - max]

Weight 1 [1 - max]

Max Message Size 5000000 B [1000000 - 10000000]

Upload Model Files ☒ true

Delete Model Files ☒ true

Property	Value	Description
Http Transport	additional properties	Contains the HTTP-specific part of the configuration.
Upload Model Files	true (default) or false	Indicates if the model data should be uploaded to the cloud.
Delete Model Files	true or false (default)	Indicates if the model data should be saved to local files. Only use this setting to retain the data in cases of support purposes.

InMemoryMessageQueue properties

All of these properties are specific to the InMemoryMessageQueue.

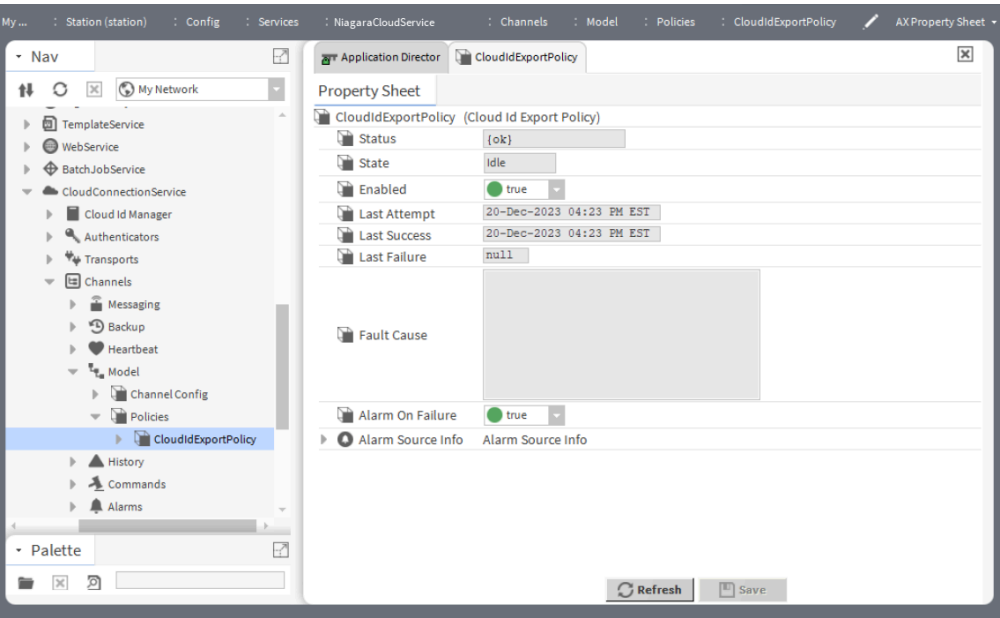
Property	Value	Description
Queue Size	5000000 (default)	Defines amount of data (bytes) a queue can hold before it starts to reject additional messages.
Weight	number (defaults to 1)	Configures the relative weight of the queue in the round robin scheduling algorithm that selects

Property	Value	Description
		the next message for the transport to send.
		The transport layer uses a weighted round robin approach to accept messages, so increasing the weight on one queue cause its messages to be sent more frequently than messages from other queues. This may be useful to ensure that certain message types like alarms are sent with priority over telemetry data. Do not modify these parameters unless you determine that needed messages are being held up by less important ones.

cloudLink-CloudIdExportPolicy

This component defines the process that exports the model to the cloud. It sends components that have a cloudId tag but not an nc:excluded tag.

It is automatically added by the provisioning service, which tells the station what to configure.



In addition to the standard properties (Status, State and Enabled), these properties are unique to the Cloud Id Export Policy.

Properties	Value	Description
Last Attempt	read-only	Indicates the date and time of the last Model export.

Properties	Value	Description
Last Success	read-only	Indicates the date and time of the last successful Model export.
Last Failure	read-only	Indicates the date and time of the last failed Model export.
Fault Cause	read-only	Indicates the reason as to why the Cloud Id Export Policy is in fault. This field is empty unless a current fault exists.
Alarm On Failure	true (default), false	Indicates if the Cloud Id Export Policy should trigger an alarm when the model export fails.
Alarm Source Info	additional properties	Configures standard Niagara Alarm Source Info on how alarms from the Cloud Id Export Policy are to be handled. NOTE: The alarms are sent only when the above Alarm On Failure property is set to true.

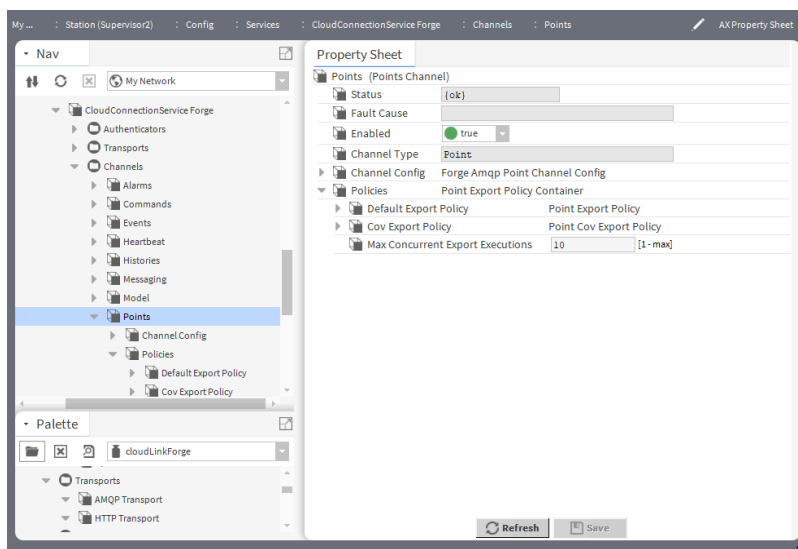
Actions

Execute: Starts a model export process. The status of the operation can be viewed in the Job Service.

cloudLink-PointsChannel

This channel handles point snapshot delivery to the cloud platform.

Figure 16. Points Channel properties



To access these properties, expand **Config > Services > NiagaraCloudService > Channels** and double-click on **Points**.

In addition to the standard properties (Status, Fault Cause and Enabled), these properties are unique to the

PointsChannel.

Property	Value	Description
Channel Type	read-only	Identifies the type of channel.
Channel Config	additional properties	Contains specific channel configuration information.
Policies	additional properties	Contains different point export strategies.
Max Concurrent Export Executions	number (defaults to 10)	Configures the number of export policies that can execute in parallel.

Point Export Policy properties

This container can hold any number of `PointExportPolicy` objects. It starts with two default policies, which are disabled by default. Additional policies can be added to export sets of points at different rates.

To access these properties, expand **Config > Services > CloudConnectionServiceForge > Channels > Points > Policies** and double-click on **Default Export Policy** or **Cov Export Policy**.

Default Export Policy

In addition to the standard properties (Status, Fault Cause and Enabled), these properties are unique to the **PointsChannel**.

Property	Value	Description
Execution Time	Interval, Time of Day, Days of Week	Controls the frequency with which the <code>PointExportPolicy</code> should send data to the cloud platform.
Trigger Mode	Manual, Daily, Interval	Controls when this export should try to send data to the cloud.
Last Attempt	read-only	Reports the date and time of the last attempted execution.
Last Success	read-only	Reports the last time the station successfully performed this function.
Last Failure	read-only	Reports the last time the system failed to perform this function. Refer to <code>Fault Cause</code> for details.
Custom Point Queries	list of queries	Lists the bql and/or neql queries with which to select the points to be exported.
Alarm on Failure	true (default) or false	Controls the recording of ping failure alarms. true records an alarm in the

Property	Value	Description
		<p>station's AlarmHistory for each ping-detected device event (down or subsequent up).</p> <p><code>false</code> ignores device down and up events.</p>
Alarm Source Info	additional properties	<p>Contains a set of properties for configuring and routing alarms when this component is the alarm source.</p> <p>For property descriptions, refer to the <i>Niagara Alarms Guide</i></p>

Cov Export Policy

In addition to the standard properties (Status, Fault Cause and Enabled), these properties are unique to the **PointsChannel**.

Property	Value	Description
Last Attempt	read-only	Reports the date and time of the last attempted execution.
Last Success	read-only	Reports the last time the station successfully performed this function.
Last Failure	read-only	Reports the last time the system failed to perform this function. Refer to Fault Cause for details.
Custom Point Queries	list of queries	Lists the bql and/or neql queries with which to select the points to be exported.
Alarm on Failure	<code>true</code> (default) or <code>false</code>	<p>Controls the recording of ping failure alarms.</p> <p><code>true</code> records an alarm in the station's AlarmHistory for each ping-detected device event (down or subsequent up).</p> <p><code>false</code> ignores device down and up events.</p>
Alarm Source Info	additional properties	Contains a set of properties for configuring and routing alarms

Property	Value	Description
		when this component is the alarm source. For property descriptions, refer to the <i>Niagara Alarms Guide</i>
Cov Batch Delay	hours, minutes, seconds	Defines the delay of sending batch updates.
Max Concurrent Export Executions	numeric	Specifies how many simultaneous threads are used to export history data. Using more threads (up to the number of export policies that are used) allows the export to proceed more quickly, but may tax the resources of either the gateway, or the connection bandwidth.

cloudLinkForge-ForgeAmqpPointChannelConfig

This channel config holds information for the points channel when used with the Forge platform and communicating over the AMQP transport.

Properties

Channel Config (Forge Amqp Point Channel Config)

- Transport Information (Transport Config)
 - Transport Type: AMQP
 - Authenticator Id: RpkAuthenticator
 - Channel Queue: In Memory Message Queue
 - Queue Size: 5000000 [1024 - max]
 - Weight: 1 [1 - max]
 - Max Message Size: 1000000 B [10000 - 10000000]

Property	Value	Description
Transport Information	read-only	Contains the AMQP specific part of the Transport configuration.
Transport Type	read-only	Identifies which transport should be used with this configuration.
Authenticator Id	read-only	Identifies which authenticator should be used, note this is only needed for connectionless transports.
Channel Queue	additional properties	Contains the queue size and weight information to use for outgoing messages.
Max Message Size	number (defaults to 1000000)	Defines the message size limit (bytes), before compression, to

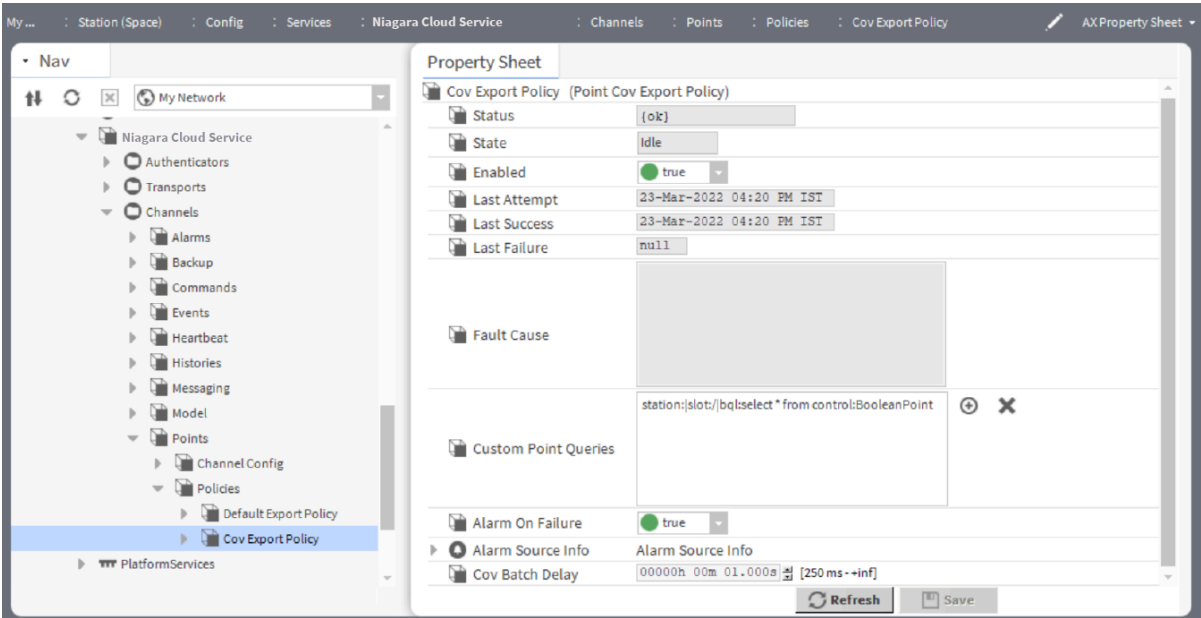
Property	Value	Description
		use with this channel and transport.

InMemoryMessageQueue properties

Property	Value	Description
Queue Size	number (defaults to 5000000)	Defines amount of data (bytes) a queue can hold before it starts to reject additional messages.
Weight	number (defaults to 1)	<p>Configures the relative weight of the queue in the round robin scheduling algorithm that selects the next message for the transport to send.</p> <p>The transport layer uses a weighted round robin approach to accept messages, so increasing the weight on one queue cause its messages to be sent more frequently than messages from other queues. This may be useful to ensure that certain message types like alarms are sent with priority over telemetry data. Do not modify these parameters unless you determine that needed messages are being held up by less important ones.</p>

cloudLink-PointsCovExportPolicy

This topic describes the **Cov Export Policy** component . By default, the **Cov Export policy** is disabled. However, when you enable it and manually add one or more queries, the points from the driver are sent to the cloud with the same cloud id.



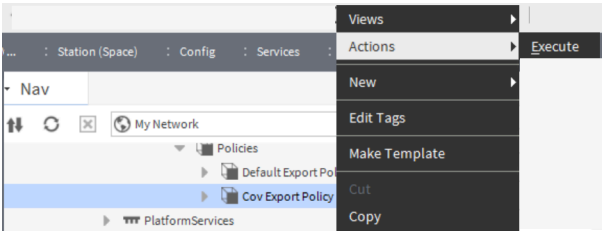
To access these properties, expand **Config > Services > NiagaraCloudService > Channels > Points > Policies** and double-click on **Cov Export Policy**.

In addition to the standard properties (Status, Fault Cause and Enabled), these properties are unique to the **PointsCovExportPolicy**.

Property	Value	Description
Last Attempt	read-only	Reports the date and time of the last point value export data.
Last Success	read-only	Reports the last successful date and time of the point value export data.
Last Failure	read-only	Reports the timestamp of the last failed export.
Fault Cause	read-only	Indicates the reason why the PointExportPolicy is in fault. This field is empty unless a fault exists.
Custom Point Queries	list of queries	Provides a list of bql queries. Click the browser icon to open File chooser, ord chooser to select the query for the points value to be exported. NOTE: Queries are not automatically added to this field. You must specify one or more queries manually.
Alarm on Failure	true (default) or false	Controls the recording of ping failure alarms.

Property	Value	Description
		<p><code>true</code> records an alarm in the station's AlarmHistory for each ping-detected device event (down or subsequent up).</p> <p><code>false</code> ignores device down and up events.</p>
Alarm Source Info	additional properties	<p>Contains a set of properties for configuring and routing alarms when this component is the alarm source.</p> <p>For property descriptions, refer to the <i>Niagara Alarms Guide</i></p>
Cov Batch Delay	00000h 00m 01.000s (default)	<p>Specifies the amount of time to delay before sending batch Cov points to the cloud.</p>

Actions



Execute executes the query for the selected points to be exported.

cloudLink-JwksTrustMapping

Jwks Trust Mapping is required to configure the station to receive commands sent from the cloud platform. This component is added to the Trust Manager in the CloudAuthenticationScheme in the AuthenticationService.

This component is available in the Authentication folder in the cloudLink palette.

JwksTrustMapping (Jwks Trust Mapping)

App Id	
Expected Jwt Audience	NiagaraCloudService
Expected Jwt Issuer	
Jwks Endpoint	

Type	Value	Description
App Id		Value of the Forge application Id.

Type	Value	Description
Expected Jwt Audience	NiagaraCloudService (default)	<p>Value of the token audience "aud" field.</p> <p>By default, "Niagara Cloud Service", but this may be changed to match the value present in the JWT for those providers that do not have a fully configurable audience field. For example, Salesforce prepends the Salesforce application Id (not to be confused with the Forge application Id) onto the audience.</p>
Expected Jwt Issuer		<p>Typically, the URL of the user identity provider.</p> <p>NOTE: This value is required. The value for the Token issuer "iss" field must to be provided by the Developer/Integrator during Certificate Trust Mapping configuration.</p>
Certificate Alias		<p>Alias of your token provider's public certificate file that was imported.</p>

cloudLink-RoleMappings

Roles are used to specify the authorization to station resources for System Commands.

For Niagara Cloud Service, the role mappings are installed and configured automatically. You can add additional role mappings manually. A **Role Mapping** component is available in the cloudLink palette under the Authentication folder.

Property Sheet

Role Mappings (Cloud Role Mappings)

Reassert Missing Standard Roles

true

CustomerAdminMapping

Role Mapping

Cloud Role

customer_admin

Station Role

CloudAdmin

CustomerUserMapping

Role Mapping

Cloud Role

customer_user

Station Role

CloudUser

NdsOperatorMapping

Role Mapping

Cloud Role

nds_operator

Station Role

CloudOperator

PartnerAdminMapping

Role Mapping

Cloud Role

partner_admin

Station Role

CloudAdmin

PartnerUserMapping

Role Mapping

Cloud Role

partner_user

Station Role

CloudUser

Once configured, the station is ready to receive commands with the specified cloud roles.

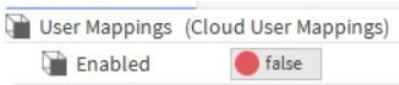
Property	Value	Description
Reassert Missing Standard Roles	true (default), false	<p>Enables/disables creation of the standard pre-configured roles. Setting this to "false" will prevent the standard roles from being created upon station start. When set to "true" a check is made (only on station startup), and for each of the three standard roles:</p> <ul style="list-style-type: none">• if the role is missing, it is created with the default permissions• if the role exists with the default permissions, nothing is done <p>If role permissions are changed during station operation, nothing is done until the next station start.</p>

cloudLink-UserMappings

In general, "user mapping" is used for Single Sign-On (SSO) to back-end systems, such as a cloud platform. User mapping maps a portal user ID to the user ID of the back-end system.

In Niagara Cloud Service user mapping is used when an application Id from Forge is mapped directly to a Niagara User (no authentication checks). Disabled by default, this component provides actions to enable/disable user mappings.

WARNING: Due to the inherent security risk, the use of UserMappings is not recommended. If enabled, a confirmation window appears prompting you to acknowledge that you wish to proceed with the non-recommended configuration.



Property	Value	Description
Enabled	true or false (default)	<p>Available as a read-only property, it is not directly editable.</p> <p>Disabled by default. If set to "true" (via Actions), UserMappings are enabled.</p> <p>If "false", cloud login attempts will fail, throwing the FailedLoginException to inform the user.</p>

Actions
The following actions are available via right-clicking the component.

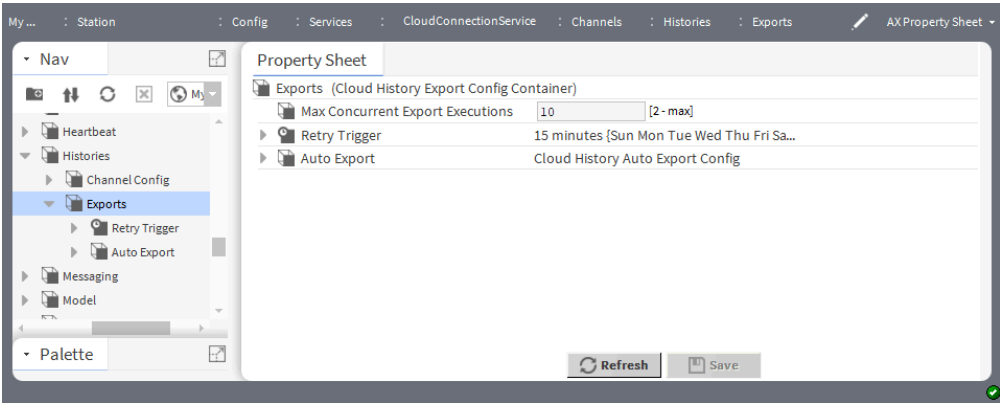
- Enable
- Disable

cloudLink-CloudHistoryExportConfigContainer

This container component allows you to configure export policies. The **Cloud History Export Config Container**can hold any number of export policies.
The auto export policy uses an opt-out model and when the **Auto Export** is enabled, it exports all histories in the station to the cloud according to the execution time. When enabled, make sure to exclude what you do not wish to send to the cloud. Its default view is the **Cloud History Export Manager**.

NOTE: It is recommended to only use an autoExport for large stations without selecting individual histories for exclusion. Configuring custom export policies on large stations can result in slow response times on the history export policy screens.

To access the **Cloud History Export Config Container** property sheet, right-click on **Exports** in the Nav tree and select **Views > Property Sheet**.



Property	Value	Description
Max Concurrent Export Execution	10 (default)	The number of export policies that can execute in parallel.
Retry Trigger	15 minutes (default)	Specifies how long to wait before retrying to execute failed exports.
Auto Export	additional properties	Configures an export policy that will export all histories in the station. If enabled, the auto export policy uses an opt out model that, by default, includes all histories. Multiple export policies can be used to control the frequency at which histories data are exported to the cloud.

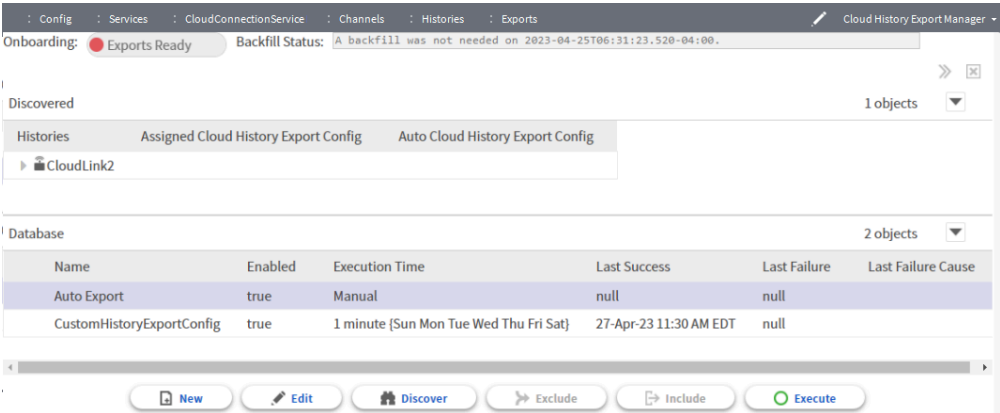
Actions

- **Execute All:** Executes all export policies.
- **Retry Failed Exports:** Retries to execute failed exports.

cloudLink-CloudHistoryExportManager

The **Cloud History Export Manager** view allows you to discover histories and assign them to **CloudHistoryExportConfigs**, or include/exclude them from **Auto Export Configs**. More actions are described below.

This component is the default view of the **Export (Cloud History Export Config Container)** component. To access the view, double-click **Exports** in the **Nav** tree.



Cloud History Export Manager buttons

Button	Description
New	Creates additional custom CloudHistoryExportConfig objects.
Edit (or double-clicking on the entry)	Opens a dialog window for changes to the CloudHistoryExportConfig details.
Discover	Discovers all eligible histories in the station.
Assign (when a CloudHistoryExportConfig is selected)	Adds the history to the CloudHistoryExportConfig if a CloudHistoryExportConfig in the Database section and a history in the Discovered section are both selected. NOTE: The Assign/Exclude buttons change text depending on the selection of a config entry in the Database section.
Unassign (when a CloudHistoryExportConfig is selected)	Removes the history from the CloudHistoryExportConfig . NOTE: The Unassign/Include buttons change text depending on the selection of a config entry in the Database section.
Exclude (visible when Auto Export is selected)	Excludes the history from the Auto Export when the Auto Export config is selected in the Database section and one or more histories are selected in the Discovered section. NOTE: The Exclude button for Auto Export becomes Assign if you select the CloudHistoryExportConfig .
Include (visible when Auto Export is selected)	Includes the history in the Auto Export when a history is selected. NOTE: The Include button for Auto Export becomes Unassign if you select the CloudHistoryExportConfig .
Execute	Executes the selected CloudHistoryExportConfig if it is enabled. When a History Config changes from disabled to enabled, a backfill check will be performed, which may trigger a backfill if the number of pending histories exceeds the Backfill Record Threshold.

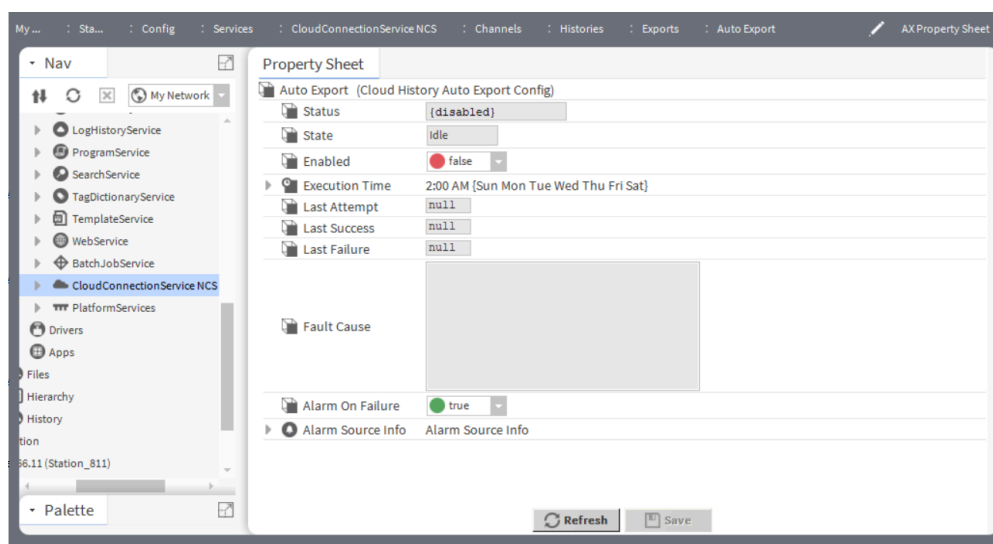
NOTE: To remove a custom export config, right-click on the entry and from the context menu select **delete**.

cloudLink-CloudHistoryAutoExportConfig

This component configures an export policy that automatically exports all histories in the station to the cloud. The auto export policy uses an opt-out model. When enabled, it sends all histories to the cloud according to the **Execution Time**. Multiple export policies can be used to control the frequency at which histories data are exported to the cloud.

NOTE: It is recommended that you only use an Auto Export for large stations without selecting individual histories for exclusion.

To access this component, double-click **Auto Export** in the Nav tree.



In addition to the standard properties (Status, State, and Enabled), these properties are unique to the **Cloud History Auto Export Config**.

Property	Value	Description
Execution Time	interval (15 minutes), days of the week (all)	Controls the frequency at which the CloudHistoryAutoExportConfig sends data to the cloud platform. Auto Export is configured for 15 minute interval, enabled for all days of the week. You can modify this setting for more or less frequent updates to Niagara Cloud Suite.
Last Attempt	read-only	Indicates the last time this export tried to export data.
Last Success	read-only	Indicates the time of the last successful export.
Last Failure	read-only	Indicates the time of the last failed export.
Fault Cause	read-only	Indicates the reason why the CloudHistoryAutoExportConfig is in fault. This field is empty unless a fault exists.
Alarm On Failure	true (default), false	Indicate if an alarm should be raised if there are problems during exporting data.
Alarm Source Info	additional properties	Configures information for alarms generated by this component.

Actions

Execute: Executes auto export policy.

Event messages and system commands

An event message is a method of pushing data to the cloud. Specific logic triggers an event message. For example, an alarm triggers a NewAlarm event message.

Event messages

The NiagaraCloudService contains channels to accomplish such things as getting message types to send to the cloud. The types of event messages are listed here:

- NewAlarms - sent to the cloud provider
- AlarmAckRequests - from the cloud provider
- AlarmAckResponses - to the cloud provider
- As well as others such as DeviceAckRequests.

System Commands

The cloud platform can send system commands down to the station. Handling for these system commands is done by extending a specific class. This is done via the CommandsChannel.

When a system command is received via a connected transport, the CommandsChannel is called to determine if it has a registered command to handle the incoming message. If there is one, the appropriate registered

command is called to process the message.

CloudLink supports the following commands but only the commands in bold are supported by NCS via the control APIs:

- ListCloudCommandsCommand — Lists the names of all commands that are available on this system.
- CloudPointReadInputsCommand — Returns all the cloud inputs for a point in the station.
- CloudMultiPointReadInputsCommand — Returns all the cloud inputs for a given set of points in the station.
- CloudPointReadCommand — Returns the value of an individual cloud accessible point in the station.
- **CloudMultiPointReadCommand** — Returns the values of a list of cloud accessible points in the station.
- RetrieveCloudPointsCommand — Lists the names of all the points in the station that are accessible from the cloud.
- **CloudMultiPointReadInputPropertiesCommand** — Returns the input slot status values for a given set of points in the station.
- CloudPointWriteCommand — Sets the value of an individual cloud accessible point in the station.
- **CloudMultiPointWriteCommand** — Sets the values of a list of cloud accessible points in the station.
- CloudMultiPointClearCommand — Releases the values of a list of cloud accessible points in the station that were previously set with a CloudPointWriteCommand or CloudMultiPointWriteCommand.
- CloudMultiAlarmWriteCommand — Adds a new note to the existing notes facet of a set of alarms in the station.
- AlarmAckCommand — Acknowledges an alarm.
- BatchAlarmAckCommand — Acknowledges a list of alarms.
- InvokeCommand — allows invocation of an action on a component.

With Cloud Command Queues the JACE/Supervisor now responds as soon as the command has been placed in the queue. When the command executes its output is sent to the cloud via NewEventMessage(s) and when the command exits another NewEventMessage is sent.

You can disable individual commands if necessary. The default is enabled

System Command Configuration

Most system commands do not require configuration outside the command component's properties. However, some commands do have built-in restrictions or restrictions that are configurable by tag.

CloudMultiPointWriteCommand Configuration

This configuration also applies to CloudPointWriteCommand , but that command is not provided by the Niagara Cloud Suite control API.

Below is a screencap of the MultiPointWrite command configuration. There are a few properties to consider. The default priority and durations. "Accept Writes Outside Facets Range" governs whether writes that are outside the min/max range defined by the point's facets will be used to reject invalid write values. If *false*, writes outside the range are rejected. If *true*, it will accept those writes where the framework allows.

Default Priority: To write at the command's default priority setting, you must omit the `inputPriority` item from each point's request data when sending a command with the NCS control API. In the following example, the first point will write at the command's default priority while the second point will write at priority 7.

NOTE: The `nc:writableLevels` tag (see below) on the point must include the command's default priority level in order for the write to be accepted.

Point Write Command:

```
{
```

```

"points": [
{
  "cloudId": "8126c229-e35a-4bd8-982c-b7e7789c2937",
  "value": 12.54,
  "duration": 5
},
{
  "cloudId": "3d732f48-2a7e-40ea-9a08-84d99fd1094e",
  "value": "88",
  "duration": 3,
  "inputPriority": 7
}
],
"requestProcessingPriority": 255
}

```

Priorities 2-16: A write command received at priorities 2-7 and 9-16 will add a temporary dynamic slot with a **BCloudWriteInfo** to the component. The **BCloudWriteInfo** will be set with the value and duration specified by the command. In addition, a link is added from the **BCloudWriteInfo**'s value property to the specified priority level input. When the command's duration has expired, both the **BCloudWriteInfo** dynamic slot and the link to the priority input are removed.

Priority 8: A write command received at priority 8 invokes an **Override** action.

Built-in priority write level restrictions:

- The point write command prevents writing at priority 1, priority 6 (for BooleanWritables) and fallback.
- The point write command prohibits writing when there is a link at a priority level except if the link is from a **BCloudWriteInfo**, which was created by a previous point write command.
- The ability to write to the remaining priorities is controlled by an nc:writableLevels tag described below.

nc:writableLevels tag: A write command is controlled by an nc:writableLevels tag on a control point. This tag's value contains a string of numbers between 2 and 16 separated by commas. The numbers indicate the priority levels for which a point write command will be allowed. For example, 2,4,5,6,7,8,9,10,11,13,14,15,16 will allow point write commands at all priority levels except for 3 and 12 as these priorities are not in the list. If a command specifies a priority of 3 or 12, it will be rejected. There are two versions of the nc:writableLevels tag:

- *nc:writableLevels tag as a direct tag:*

You can add the nc:writableLevels tag as a direct tag to a component with a string value in the format described above. This direct tag will override any nc:writableLevels indirect tag added by the tag rule. Since there will already be an nc:writableLevels indirect tag added to all **BIWritable** components, the nc:writableLevels direct tag must be added by selecting **Edit Tags window > Direct Tags tab**, and clicking **AddTag**. In the **Add Tag** window for Tag Id, enter nc:writableLevels and for **Type**, choose **baja:String**. The value of the nc:writableLevels tag should be entered as a string of numbers between 2 and 16 separated by commas.

Figure 17. nc:writableLevels tag definition in the Niagara Cloud Tag dictionary

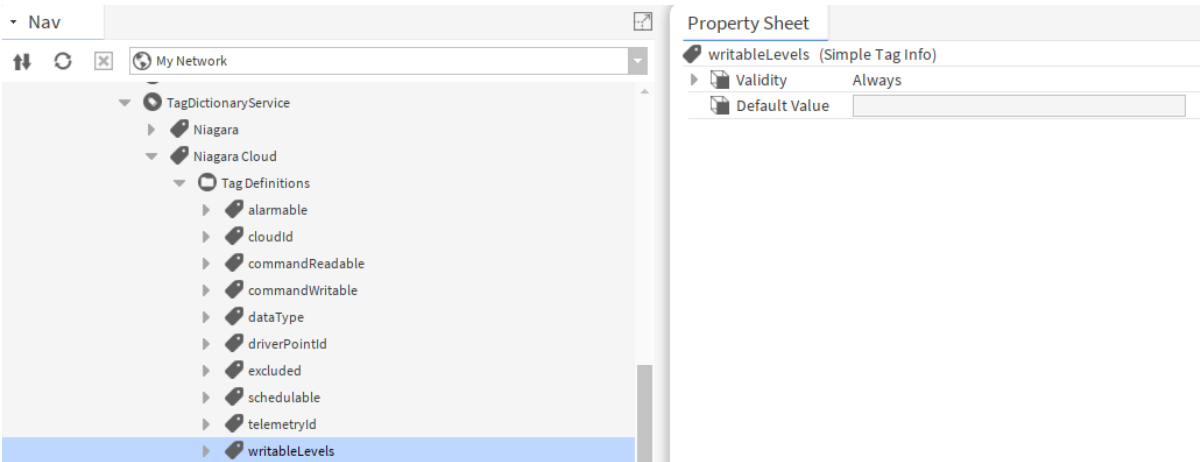
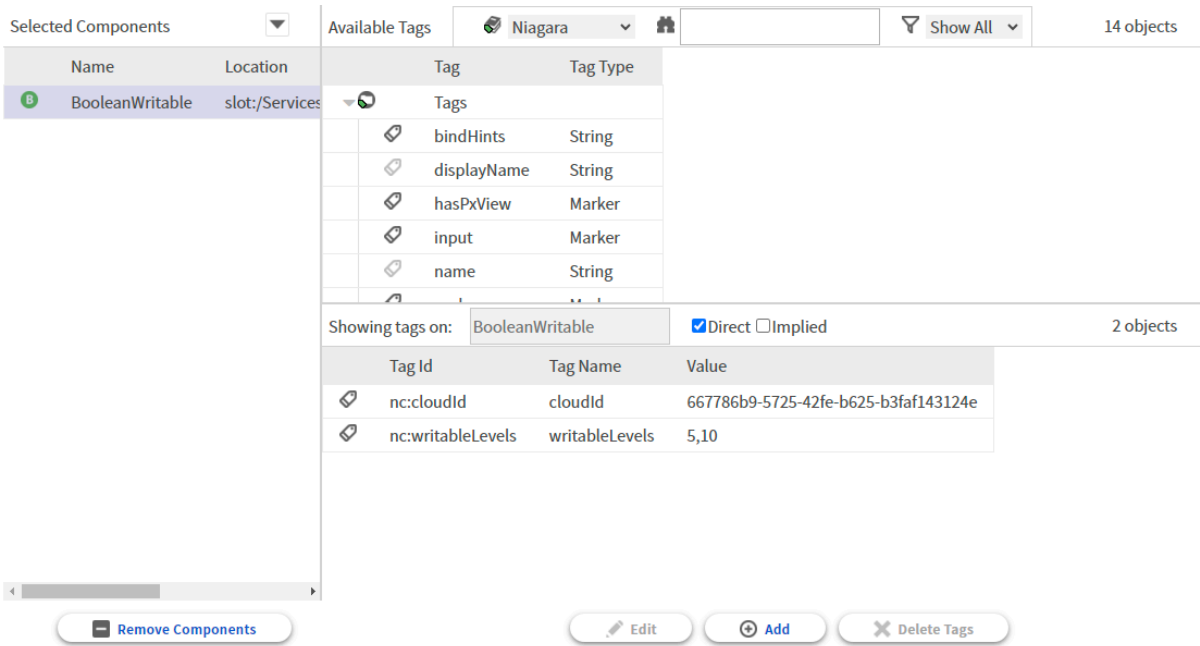


Figure 18. nc:writableLevels tag on a control point



- nc:writableLevel tag as an indirect tag:*

The Niagara Cloud Tag dictionary contains a tag rule, which automatically adds an nc:writableLevels indirect tag to all **BIWritablePoints**. The indirect tag's value will show the list of priority levels which are permitted on the point and takes into account any built-in restrictions. By default, this tag rule applies to all **BIWritable** components in the station, however only those control points that also have a nc:cloudId tag will have a value for the nc:writableLevels tag. Control points without a nc:cloudId tag cannot be commanded from the cloud and the tag value will be blank. The tag rule can be customized as necessary to restrict certain input priorities from being written to from the cloud.

In the property sheet of the WritablePriorityLevels custom tag (see figure below), the **Excluded** property is used to prohibit writes at certain priority levels. The value is a string of numbers between 2 and 16 separated by a commas. The numbers indicate the priority levels for which a write command will not be allowed. As configured in the figure, input priorities 11 and 12 will be excluded.

In the property sheet of the WritablePriorityLevels custom tag (see figure below), the **Strict** property further restricts point writes to the levels indicated in the **Excluded** property and those at a higher

priority. The `Strict` property will also further restrict point write commands to priorities at or at a higher priority level at which there is a link.

NOTE: The `BCloudWriteInfo` links placed by previous point write commands are not restricted, that is, a new point write command can overwrite a previous point write command at the same priority level, but those inputs at a higher priority will still be restricted.

Figure 19. Niagara Cloud tag dictionary Writable Point Tags rule

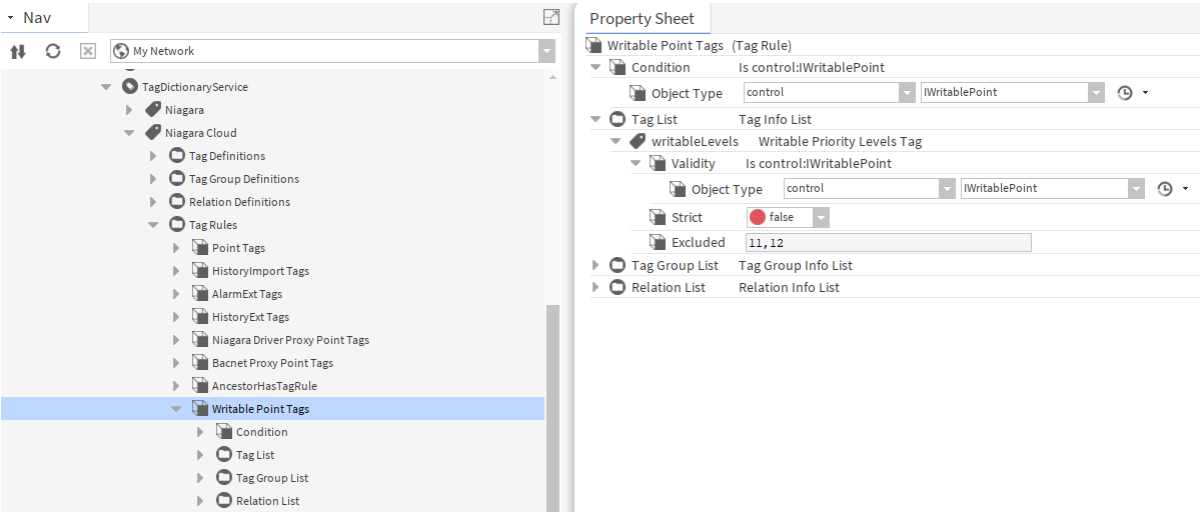
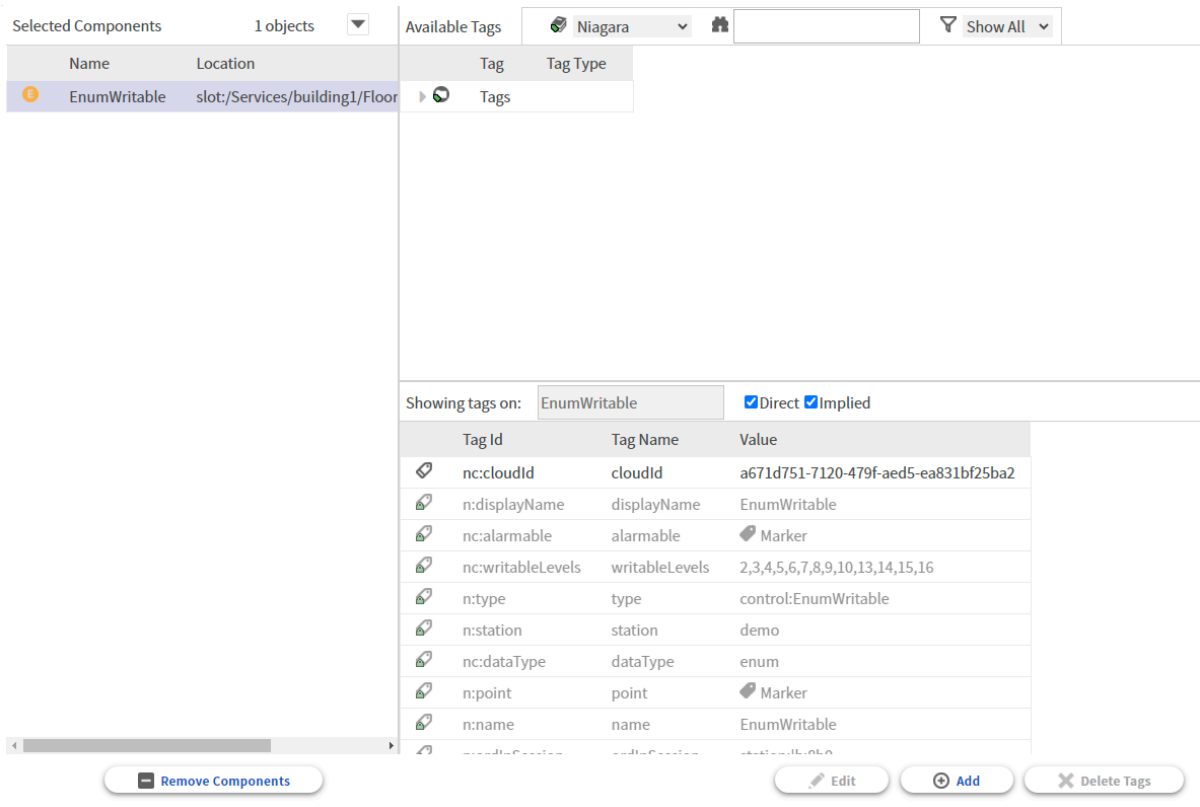


Figure 20. nc:WritableLevels indirect tag based on the Writable Point Tags rule

NOTE: The nc:writableLevels indirect tag only shows the correct priority levels when viewed with the Tag Manager view in Workbench. In the Edit Tags window, Implied Tags tab does not always show the correct value for the tag.



Examples of property settings

Strict	Excluded	Station Link	CloudWriteInfo	nc:writableLevels tag	Notes
false	4,10	none	none	2,3,5,6,7,8,9,11,12,13,14,15,16	
true	9	none	none	10,11,12,13,14,15,16	

Strict	Excluded	Station Link	CloudWriteInfo	nc:writableLevels tag	Notes
false	none	input 6	none	2,3,4,5,7,8,9,11,12,13,14,15,16	
true	none	input 6	none	7,8,9,11,12,13,14,15,16	Command is not permitted at or higher than priority input 6.
false	none	none	input 6	2,3,4,5,6,7,8,9,11,12,13,14,15,16	Command write is permitted at input 6.
true	none	none	input 6	6,7,8,9,11,12,13,14,15,16	Command write is not permitted higher than input 6.
false	5	input 6	input 6	2,3,4,7,8,9,11,12,13,14,15,16	Input 6 is restricted but input 9 is not restricted.
true	5	input 6	input 6	9,11,12,13,14,15,16	Inputs higher than 9 are restricted due to the CloudWriteInfo link.

Custom commands

In addition to providing many “out-of-the-box” commands, Niagara Cloud Service also provides the capability to handle custom commands. This ability to invoke custom code provides greater flexibility, however, there are multiple restrictions and security requirements.

Chapter 5. Troubleshooting

This information is provided to make the troubleshooting and diagnosis of the **NiagaraCloudService** as straightforward as possible.

This troubleshooting information is intended for anyone who may be using the **NiagaraCloudService**, or supporting those who are using it.

Most of the pieces of the **NiagaraCloudService** have individual enable flags, so they can be separately enabled or disabled. In most cases, you should not disable any parts of the service, as most cloud applications depend upon all data streams being in place. However, it may be easier to diagnose a problem with an individual component if you disable the other components that are in parallel with the component under investigation. Do not disable the component(s) used by the aspect of the **NiagaraCloudService** you are investigating.

When an incident occurs

Collecting the following recommended information helps the technical support team get to the root cause of the problem quickly, characterize defects fully, and address the problem for immediate and future users.

Information to collect

- Date and time of incident; be as accurate as you can with the time
- Customer or user in question, including brand
- Hardware platform (for example, OS, version)
- Core Niagara software version (and any additional patches beyond base).
- Niagara Cloud modules versions, not just the release but the specific version of each module
- Any third party modules in use
- Any relevant log output or stack traces; see "What Logs to Collect". More is better; extraneous information can be discarded if it is not important, but lost information cannot be recovered.
- Any relevant files; see "What Files to Collect".
- Authenticator information (for example, system ID, system type); see "Collecting Authenticator Information".

Questions to answer

- What steps were taken before and after the problem? Be as specific and complete as possible.
- Information about the network environment is critical in many cases. Is the host experiencing network disconnections (either intentional or not)? Is a proxy server in use? If yes, is it transparent or explicit (named). Is the Niagara **HttpProxyServer** service used?
- What steps were taken to resolve the problem?
- Was the **NiagaraCloudService** or authenticator disabled/enabled, did you do a **forceReconnect**, was the station restarted, did you attempt to reregister the authenticator? Ideally, if the station state can be left unchanged, the support team may suggest steps to correct the problem, or to learn more about it.

What logs to collect

There are several logs that can be enabled for diagnosing connection problems. The following tables list the logs.

Niagara Cloud Service logs

When you enable moderately or highly verbose logs, it is best practice to stream the station output to a file. For more information, see "What files to collect". This allows you to capture what may be a larger amount of data than can be saved from the regular station output window and is the recommended way to capture output data. Also, saving the log to a file gives you something to refer to later and to share with technical

support, if needed.

Deciding which logs to enable requires a bit of judgment. You could set every log to: ALL, but this would yield so much data it would be difficult to dig through it all to isolate a specific problem. You need to decide what might be the likely source. Each of the basic Niagara Cloud Service functions, such as histories, has a log level beginning with "cloudLink". These do not generate a giant amount of data, so they can usually be set to ALL for whatever the specific function calls for.

- For issues with message security and authentication, set cloudLink.security and authentication to ALL. The output level is usually low enough to be manageable.
- For Niagara Cloud Suite registration , set cloudLink.auth.federated to ALL; for NDS/RPK registration, use registration, use cloudLink.auth.forge.
- For IoT Hub concerns, set cloudLink.transport.amqp.client. This can be extremely verbose, especially for a large system with many points and histories.

CAUTION: Do not forget to return logger settings to their default INFO levels once your problem is corrected. Leaving the loggers at higher levels of debug can impact system performance, and hide any new problems under a wave of noisy station output. This is especially true for the cloudLink.transport.amqp.client logger.

Log Name	Description	Verbosity	Notes
authentication	Inbound command authentication logging	Low	User authentication; non-cloud, but may be useful in identifying failed command reason
cloudLink.alarm	Alarm recipient logging	Low	alarm message delivery
cloudLink.auth.federated	Federated Device Identity authenticator logging	Low	set to CONFIG for NCS registration trace
cloudLink.auth.key	Logging related to key retrieval	Low	set to CONFIG for information about key retrieval/generation
cloudLink.channel	Common channel logging	Low	
cloudLink.channel.alarm	Alarm channel configuration information	Low	
cloudLink.channel.command	Command Channel information	Moderate	Set to FINER for command tracing
cloudLink.channel.event	Event channel configuration information	Low	
cloudLink.channel.heartbeat	Heartbeat Channel information	Low	
cloudLink.channel.history	History Channel information	Low	
cloudLink.channel.messaging	Message Channel information	Low	
cloudLink.channel.model	Model Channel information	Low	
cloudLink.channel.Point	Point Channel information	Low	
cloudLink.channelConfigFactory		Low	
cloudLink.connectionService	NiagaraCloudService logging	Low	set to ALL for factory management logging
cloudLink.event	Event recipient logging	Low	event message delivery
cloudLink.licenseLimit	License check logging	Low	
cloudLink.model.batch		Low	
cloudLink.model.exportPolicy		Low	
cloudLink.point	Point export policy logging	Low	
cloudLink.queue.inMemory	Outbound message queue logging	Moderate	set to FINER for message queue tracing
cloudLink.security	Trust mapping logger	Low	
cloudLink.smaMonitor	SMA monitor logging	Low	
cloudLink.tag	CloudId tagger logging	Low	
cloudLink.transport	Common transport logging	High	set to FINER form message throttling and tracing information
cloudLink.transport.amqp	AMQP transport Logging	Moderate	set to FINE for inbound message tracing

Log Name	Description	Verbosity	Notes
cloudLink.transport.amqp.client	AMQP client Logging	High	set to ALL for AMQP event tracing
cloudLink.transport.file	Local file system transport logging	Low	Set to ALL for file lock events
cloudLink.transport.http	HTTP transport Logging	Low	
cloudLink.util	Utility Logging	Low	

CloudLinkForge

Log Name	Description	Verbosity	Notes
cloudLink.auth.forge	Forge Authenticator logging	Low	set to CONFIG for RPK trace
cloudLink.forge	Utility logging	Low	
cloudLink.forge.msg	Message serialization logging	High	

What files to collect

After setting logs, collecting the station output is critical to diagnosing the problem. To do this, it is best to stream the station output to a file on your Workbench PC. This can be done from the **Application Director** window.

If the incident has already happened, it can be useful to go into the host's file system and get the older console output. This will be in the User Home with the filename `"console.txt"`. Previous console logs from earlier station executions may also be useful. They will be listed under `"console_backup_YYMMDD_HHMM.txt"`.

The station database is always helpful, and may allow technical support to determine configuration problems that lead to the behavior being investigated. The station database is in the `config.bog` file. It may also be helpful to include the full station using the station copier.

As a diagnostic tool, it is a good idea to create a backup distribution file, which also contains the cloud certificates. You may use the Workbench BackupService to create this file or Niagara Cloud Service to archive backups in the cloud from where Niagara Recover can retrieve them.

NOTE: If you are providing technical support with the bog file or full station copy, be sure to provide the username and password for the station.

Authenticator information

This information is particularly important if support personnel need to make any modifications to the device registration.

The `FederatedIdentityAuthenticator` is the authenticator for the Niagara Cloud Suite. It handles the station-side registration with the Federated Identity Service and provides a secure connection to the NCS identity provider.

The RPK Authenticator is only relevant if Niagara Data Service is installed. The authenticator's **System Id** property is important if it has been populated. Knowing that the **System Id** is empty is also useful, so note if it is empty. The text field size often prevents full display of the values, so the best approach is to right-click on the **RPK Authenticator** and select **Views > Spy Remote**. Copy the entire text of that page and paste it into a text file.

Network sanity checks

If you are unable to register a controller with the Niagara Cloud, these sanity checks are intended to help you identify the source of the problem.

Unfortunately, a controller may not provide the full spectrum of tools available to probe the network environment; however, you can run all the basic checks below from the controller. If you can connect a laptop to the controller network you will be able to run the tests in the additional checks section.

Checking endpoint availability

This more advanced test attempts to reach the web endpoints required for device registration with a browser. If you are installing a Supervisor, these checks should provide additional information.

Prerequisites:

You have a Windows or Linux PC connected to the same network as the controller.

- Step 1. Open a browser and navigate to `https://api.niagara-cloud.com`.
This should return a JSON formatted response.
- Step 2. Navigate to `https://gaprodsystemauthentication.sentience.honeywell.com/api/authentication/rpkchallenge`.
This should return an XML formatted error message stating that the service does not support the GET method.
- Step 3. Navigate to `https://gaprodregui.sentience.honeywell.com/api/swagger/public`.
This should return a JSON formatted response.
- Step 4. If you cannot reach the endpoints, attempt to see where the problem is using the trace route (tracert) Windows command.

This shows the path through the network that packets are taking.

1	>tracert niagara-cloud.com				
2					
3	Tracing route to waws-prod-blu-075.api.niagara-cloud.com [13.82.101.179]				
4	over a maximum of 30 hops:				
5					
6	1	3 ms	3 ms	3 ms	137.19.60.3
7	2	1 ms	1 ms	1 ms	137.19.35.237
8	3	15 ms	16 ms	15 ms	10.160.16.2
9	4	21 ms	15 ms	15 ms	10.223.255.229
10	5	15 ms	15 ms	14 ms	10.223.255.65
11	6	16 ms	16 ms	16 ms	10.223.255.58
12	7	17 ms	15 ms	16 ms	10.221.192.36
13	8	15 ms	15 ms	18 ms	199.64.6.87
14	9	15 ms	15 ms	16 ms	199.64.6.52
15	10	15 ms	16 ms	16 ms	199.64.6.77
16	11	26 ms	58 ms	28 ms	12.249.243.109
17	12	22 ms	22 ms	23 ms	cr2.phlpa.ip.att.net [12.123.237.142]
18	13	24 ms	22 ms	22 ms	12.122.2.201
19	14	22 ms	22 ms	22 ms	gar3.rcmva.ip.att.net [12.122.135.173]
20	15	24 ms	20 ms	20 ms	12.122.135.109
21	16	23 ms	27 ms	32 ms	12.247.95.62
22	17	25 ms	24 ms	25 ms	be-74-0.ibr02.was05.ntwk.msn.net [104.44.9.42]
23	18	24 ms	24 ms	24 ms	be-1-0.ibr01.was05.ntwk.msn.net [104.44.4.18]
24	19	23 ms	23 ms	22 ms	be-5-0.ibr04.bl20.ntwk.msn.net [104.44.16.183]
25	20	23 ms	23 ms	22 ms	ae161-0.icr01.bl7.ntwk.msn.net [104.44.21.230]
26	21	*	*	*	Request timed out.
27	22	*	*	*	Request timed out.

Entries that get an asterisk (*) represent network messages that timed out. If a host gets three asterisks, the endpoint is either down or configured not to respond to ping traffic. Services running in the cloud are usually configured not to respond to ping traffic; however, you can see if our network traffic is making it out of the local network environment.

In the example above, lines 17 and 19 report a response from a server owned by AT&T. Lines 22 through 25 report responses from Microsoft owned machines. This tells us that the station is able to route out of the local environment onto the public Internet.

These traces provide other information. For example, if a host has one or two asterisks on its line, the host or a host leading up to it is dropping packets. This degrades performance and could lead to other problems. A big jump in response times from one line to the next could indicate a potential network problem.

Registration issues

Several problems can prevent the registration of devices with the Niagara Cloud.

Device Registration view does not load

If the device registration does not load there are several things you can do.

Cause

The typical cause of this registration problem is a corrupted cloudLinkForge-ux module. If you see a white screen of delay after double-clicking the **RpkAuthenticator**, the device registration widget did not load.

Tips

- First, wait about 10-15 seconds and refresh the view. Occasionally, the view presentation may take a little longer than expected to load, and can be resolved by a view refresh.
- Try restarting your browser.
- If you are using Workbench, close and restart it.

- If the widget still fails to load, open another **ux** view and confirm that another view, such as the **User Manager** on the **UserService** loads successfully.
 - If the **User Manager** view does not load, there may be a more general problem.
 - If the **User Manager** view loads successfully, you may have a problem with the **ux** module providing the **Device Registration** view. This could be because your **cloudLinkForge-ux** module has been corrupted. This can happen if an IT algorithm strips potentially dangerous files from within the **.jar** file when emailing it to other users. Install the modules again (see “Installing software modules”).

Solution

If your **cloudLinkForge-ux** module is corrupted, obtain a valid version of the module, either by downloading it directly from the Niagara Community Software portal, or from a trusted source.

Cannot reach device registration web service

This topic provides help when the **RpkAuthenticator** is prevented from reaching the device registration web service.

Cause

This registration problem typically occurs when you are connected to the station using Workbench or a browser on a machine that does not have sufficient access to the Internet. The station host must have Internet access to authenticate directly with the identity provider, which enables cloud communication. Lack of Internet access prevents device registration.

NOTE: Sufficient access means not only that the machine has access to the Internet, but that certain proxy and firewall limitations are not in effect. For details, see the “Requirements” topic in this guide. Your network configuration must satisfy the stated requirements.

Tip

The following error in the Workbench VM, not the station VM, confirms that your client Workbench or browser does not have Internet access, or is blocked by proxy or firewall rules from reaching a necessary destination:


```

1 >tracert api.niagara-cloud.com
2
3 Tracing route to api.forge.connected.honeywell.com [20.120.121.65]
4 over a maximum of 30 hops:
5
6  1    3 ms    3 ms    3 ms  137.19.60.3
7  2    1 ms    1 ms    1 ms  137.19.35.237
8  3   15 ms   16 ms   15 ms  10.160.16.2
9  4   21 ms   15 ms   15 ms  10.223.255.229
10 5   15 ms   15 ms   14 ms  10.223.255.65
11 6   16 ms   16 ms   16 ms  10.223.255.58
12 7   17 ms   15 ms   16 ms  10.221.192.36
13 8   15 ms   15 ms   18 ms  199.64.6.87
14 9   15 ms   15 ms   16 ms  199.64.6.52
15 10  15 ms   16 ms   16 ms  199.64.6.77
16 11  26 ms   58 ms   28 ms  12.249.243.109
17 12  22 ms   22 ms   23 ms  cr2.phlpa.ip.att.net [12.123.237.142]
18 13  24 ms   22 ms   22 ms  12.122.2.201
19 14  22 ms   22 ms   22 ms  gar3.rcmva.ip.att.net [12.122.135.173]
20 15  24 ms   20 ms   20 ms  12.122.135.109
21 16  23 ms   27 ms   32 ms  12.247.95.62
22 17  25 ms   24 ms   25 ms  be-74-0.ibr02.was05.ntwk.msn.net [104.44.9.42]
23 18  24 ms   24 ms   24 ms  be-1-0.ibr01.was05.ntwk.msn.net [104.44.4.18]
24 19  23 ms   23 ms   22 ms  be-5-0.ibr04.bl120.ntwk.msn.net [104.44.16.183]
25 20  23 ms   23 ms   22 ms  ae161-0.icr01.bl17.ntwk.msn.net [104.44.21.230]
26 21  *        *        *      Request timed out.
27 22  *        *        *      Request timed out.

```

Solution

Ensure that your client machine running Workbench or the browser has Internet access before attempting device registration. Also, make sure that the URLs specified in the “Requirements” topic of this guide are accessible to the client machine, and are not blocked by a network proxy or firewall configuration.

Connection issues

There are several reasons why a connector might not be able to send data to the Niagara Cloud through the IoT Hub. Many of them relate to issues outside of **NiagaraCloudService**.

Federated identity does a provisioning check every 15 minutes. If it finds an unregistered **RpkAuthenticator** at the end of that check, it tries to register the authenticator. The **RpkAuthenticator** is disabled until the process returns a success registration status response.

Cannot connect to the cloud

A message that indicates a failure to connect to the cloud may require special action.

```
WARNING [14:41:57 02-Oct-18 EDT][cloud.connector] Cannot connect to Cloud
```

- Step 1. Set the `cloudLink.transport.amqp` and `cloudLink.transport.amqp.client` log levels at least to FINE. You may set them to an even finer level, such as FINER, FINEST or ALL, although, the finer you set the log level, the more data the log produces.
- Step 2. Confirm that your device is enabled.

System Disabled

The Honeywell Forge Operations team tracks and aggressively manages the bandwidth usage of systems participating in the Forge Platform ecosystem. Devices that send too much data are subject to being disabled from the Forge IoT Hub. This means that the device is prevented from sending any data to the Forge IoT Hub. The device may even be prevented from establishing the IoT Hub connection in the first place. This may manifest in several different ways. One example is where the authenticator is able to authenticate to the identity endpoint, but cannot open the connection. You may see the Property Sheet of the RpkAuthenticator show "Connected", or possible "Pending Connect". The following message, or something similar, may show in the station output:

```

1  CONFIG [14:33:56 29-Jul-19 BST][cloud.connector.sentence] Starting RPK Challenge
2  CONFIG [14:33:56 29-Jul-19 BST][cloud.connector.sentence] Sending RPK Challenge request to URI https://gaprodsystemauthentication.s
3  FINE [14:33:57 29-Jul-19 BST][cloud.connector.http] HTTP Response Code:200
4  CONFIG [14:33:57 29-Jul-19 BST][cloud.connector.sentence] Checking for existing locally initialized keys
5  CONFIG [14:33:58 29-Jul-19 BST][cloud.connector.sentence] Authenticating using software keys
6  CONFIG [14:33:58 29-Jul-19 BST][cloud.connector.sentence] Sending RPK Challenge Response to URI https://gaprodsystemauthentication.s
7  FINE [14:33:59 29-Jul-19 BST][cloud.connector.http] HTTP Response Code:200
8  CONFIG [14:33:59 29-Jul-19 BST][cloud.connector.sentence] Completed RPK Challenge - 2438 ms
9  CONFIG [14:33:59 29-Jul-19 BST][cloud.connector.sentence] Starting System Connections
10 WARNING [14:33:59 29-Jul-19 BST][com.microsoft.azure.sdk.iot.device.transport.amqps.AmqpsDeviceAuthenticationCBSTokenRenewalTask] jav
11 FINE [14:34:02 29-Jul-19 BST][cloud.connector.http] HTTP Response Code:200
12 CONFIG [14:34:02 29-Jul-19 BST][cloud.connector.sentence] Completed System Connections: 3125 ms
13
14 FINEST [14:34:03 29-Jul-19 BST][cloud.connector] BCloudConnector.pingFail(Could not open the connection), notifying connectCallbacks
15 FINE [14:34:03 29-Jul-19 BST][cloud.connector] Connection fail
16 java.io.IOException: Could not open the connection
17     at com.microsoft.azure.sdk.iot.device.DeviceIO.open(DeviceIO.java:165)
18     at com.microsoft.azure.sdk.iot.device.DeviceClient.open(DeviceClient.java:369)
19     at com.tridium.cloud.client.iotdep.BIoTHubMessageClient.lambda$onConnect$4(BIoTHubMessageClient.java:441)
20     at java.security.AccessController.doPrivileged(Native Method)
21     at com.tridium.cloud.client.iotdep.BIoTHubMessageClient.onConnect(BIoTHubMessageClient.java:387)
22     at com.tridium.cloud.client.iothub.BAbstractIoTHubConnectorImpl.doConnect(BAbstractIoTHubConnectorImpl.java:79)
23     at com.tridium.cloud.client.sentence.BSentenceConnectorImpl.doConnect(BSentenceConnectorImpl.java:734)
24     at com.tridium.cloud.client.BConnectorImpl.connect(BConnectorImpl.java:118)
25     at com.tridium.cloud.client.BCloudConnector.reconnectSync(BCloudConnector.java:527)
26     at java.util.concurrent.FutureTask.run(FutureTask.java:266)
27     at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.access$201(ScheduledThreadPoolExecutor.java:180)
28     at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:293)
29     at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
30     at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
31     at java.lang.Thread.run(Thread.java:748)
32 Caused by: com.microsoft.azure.sdk.iot.device.exceptions.TransportException: Unknown transport exception occurred
33     at com.microsoft.azure.sdk.iot.device.transport.amqps.AmqpsIoTHubConnection.onLinkRemoteClose(AmqpsIoTHubConnection.java:729)
34     at org.apache.qpid.proton.engine.BaseHandler.handle(BaseHandler.java:176)
35     at org.apache.qpid.proton.engine.impl.EventImpl.dispatch(EventImpl.java:108)
36     at org.apache.qpid.proton.reactor.impl.ReactorImpl.dispatch(ReactorImpl.java:324)
37     at org.apache.qpid.proton.reactor.impl.ReactorImpl.process(ReactorImpl.java:291)
38     at com.microsoft.azure.sdk.iot.device.transport.amqps.IoTHubReactor.run(IoTHubReactor.java:28)
39     at com.microsoft.azure.sdk.iot.device.transport.amqps.AmqpsIoTHubConnection$ReactorRunner.call(AmqpsIoTHubConnection.java:824)
40     at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    ... 3 more

```

This log indicates that the IoT Hub has blocked your device due to sending too much traffic.

In the screen capture above, all the authentication steps return an HTTP Response Code of 200 (see lines 3, 7, and 11), indicating success. The exception occurs only when attempting to establish the IoT Hub connection (see lines 32– 40).

If this is happening, your device may be blocked (that is, throttled) by the Niagara Cloud due to sending too much traffic at some point. You should have received an email indicating that the device has been blocked.

- Step 3. If you received an email, contact support and request that your device's ability to connect with the cloud platform be re-enabled.
- Step 4. If you did not receive an email, and you are connecting for the first time to the cloud, there may be a problem with the email addresses on file for this system. Work with support to set the proper notification configuration and re-enable your device's ability to connect.

Authenticator keys are lost

This applies to all non-QNX stations. Controllers should use hardware encryption.

If your station output appears as shown here:

Figure 21. Station output

```

CONFIG [20:58:51 23-Feb-21 EST][cloudLink.auth.forge] Connecting to Forge identity provide
INFO [20:58:51 23-Feb-21 EST][cloudLink.auth.key] Starting to init keys with id of :N4:dem
INFO [20:58:51 23-Feb-21 EST][cloudLink.auth.key] Authenticator found existing local keys
FINEST [20:58:51 23-Feb-21 EST][cloudLink.auth.forge] getConnectionInfo called but no conn
CONFIG [20:58:52 23-Feb-21 EST][cloudLink.auth.forge] Starting RPK Challenge
CONFIG [20:58:52 23-Feb-21 EST][cloudLink.auth.forge] Sending RPK Challenge request to URL
niagara>INFO [20:58:52 23-Feb-21 EST][cloudLink.auth.key] Starting to generate a signed re
INFO [20:58:53 23-Feb-21 EST][cloudLink.auth.key] Generating signature with keyId of N4:de
CONFIG [20:58:53 23-Feb-21 EST][cloudLink.auth.forge] Sending RPK Challenge Response to UR
CONFIG [20:58:53 23-Feb-21 EST][cloudLink.auth.forge] Next RpkAuthenticator token renewal
WARNING [20:58:53 23-Feb-21 EST][cloudLink.auth.forge] Cannot reauthenticate: Could not au
com.tridium.cloudLink.auth.SystemAuthenticationException: Could not authenticate:java.util
    at com.tridium.cloudLink.forge.auth.BRpkAuthenticator.authenticate(BRpkAuthenticat
    at com.tridium.cloudLink.forge.auth.BRpkAuthenticator.reauthSync(BRpkAuthenticator
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.access$201
    at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(Schedu
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)
Caused by: java.util.concurrent.ExecutionException: com.tridium.cloudLink.transport.HttpSt
    at java.util.concurrent.CompletableFuture.reportGet(CompletableFuture.java:357)
    at java.util.concurrent.CompletableFuture.get(CompletableFuture.java:1908)
    at com.tridium.cloudLink.forge.auth.BRpkAuthenticator.rpkChallenge(BRpkAuthenticat
    at com.tridium.cloudLink.forge.auth.BRpkAuthenticator.authenticate(BRpkAuthenticat
    ... 7 more
Caused by: com.tridium.cloudLink.transport.HttpStatusException:
    at com.tridium.cloudLink.transport.BHttpTransport.lambda$send$1(BHttpTransport.jav
    at java.security.AccessController.doPrivileged(Native Method)
    at com.tridium.cloudLink.transport.BHttpTransport.send(BHttpTransport.java:251)
    at com.tridium.cloudLink.transport.BAbstractTransport.lambda$sendMessages$9(BAbstr
    at java.security.AccessController.doPrivileged(Native Method)
    at com.tridium.cloudLink.transport.BAbstractTransport.sendMessages(BAbstractTransp
    ... 3 more

```

In addition, your `RpkAuthenticator` properties appear as shown here:

Property Sheet	
RpkAuthenticator QA (Rpk Authenticator)	
Status	{ok}
Fault Cause	
Enabled	<input checked="" type="checkbox"/> true
Authenticator Id	RpkAuthenticator
System Id	N4:demo3:Tst-2647-CB53-252D-1220
System Guid	1f74ed16-8edf-475a-b1bd-5f56b3168318
System Type	n4-station
Registration State	Registered
Authentication State	Authentication Failed
System Ownership Code	e35147fa096aef36fa5672bc64b7bca29e704cc8
DevTestComp	Dev Test Component

your station does not have the required public/private key pair stored in its User Key Store that it needs to authenticate to the Niagara Cloud.

To confirm this, check the **User Key Store** tab of the **Certificate Manager** for a key with an alias matching the station name. This alias will be all lowercase, prefaced with "cloud_" and with hyphens replacing any underscores in the station name. If you do not see the alias for your station, your station cannot register because it does not have the necessary key pair. The station is registered with the cloud, but cannot authenticate.

NOTE: If the key pair is missing when the station starts, the startup process generates a new key pair, however, this key pair is not registered with the cloud so the device cannot authenticate using it.

Solution

If this is a new station, contact support to remove the registration with the Niagara Cloud, and register the device again.

Always keep a current backup distribution file of the station platform that contains the certificates. You may also export the certificates with their private keys so that you them in case of future need. Store any exported keys in a safe place, preferably off campus.

If this is an existing station and, at some point, you exported the keys from the station's **User Key Store**, try importing them back into your **User Key Store** using the **Certificate Manager**. If the file contains the correct keys, the authenticator should reconnect successfully.

Niagara Cloud Service does not attempt connection

If you registered your authenticator and received a success message, but your authenticator does not attempt to connect at all, that is, there is no confirmation message in the station output when your cloudLink.auth.forge log is set to ALL, your device remains disconnected.

Solution

Here is an example of the message that should appear in the log if succeeded: `FINE [11:57:26 26-Apr-24 EDT][cloudLink.auth.forge] Authenticated with the cloud identity provider`. If it does not appear, proceed with the following solution.

Try disabling the `RpkAuthenticator`, then re-enable it.

Proxy server preventing connection

If you are able to register the station with the Niagara Cloud but the station cannot connect (the connector's **Connection State** never displays `Connected` and the AMQP Transport's **Connection State** never displays `Connected`), there may be a problem with the local IT network's proxy settings, or with the firewall settings imposed upon the station.

Problem

The `NiagaraCloudService` requires Unauthenticated Proxy Access. Without this, the proxy server prompts for credentials, and asks you to approve exemptions for certificates in the Niagara **User Trust Store**. This process repeats itself frequently and does not provide a workable solution.

If network settings prevent the station from connecting properly, the station remains in the unregistered state even if registration reports that it is registered. If you received the "successfully registered!" message, your device is registered. If the device's RPK Authenticator still shows "Unregistered," registration cannot reach the authentication endpoint.

Solution

If you are using a proxy server that requires credentials, or an explicit (named) proxy server, install the `HttpProxyServer` service from the net-rt module and configure it with appropriate settings for your proxy server. Use the `HttpProxyServer` to direct traffic.

Review with your IT administrator your Internet connection (refer to “Setting up device internet access” in this guide), specifically regarding firewall access and unauthenticated transparent proxy access. This is a common problem with network setup, especially in a heavily restricted corporate or educational network. Ensure that the requirements in this section are met by the IT network configuration.

AMQP blocked

If you are using AMQP as your transport, you have registered your authenticator, and you are seeing the connector status stuck in Pending Connect, it may be because AMQP is blocked on your network.

Figure 22. NiagaraCloudService properties for AMQP blocked

The screenshot displays the configuration interface for NiagaraCloudService, organized into two main sections: Authenticators and Transports.

Authenticators Section:

- Client Authenticators Folder:** Contains 'RpkAuthenticator QA'.
- RpkAuthenticator QA Properties:**
 - Status: {ok}
 - Fault Cause: (empty)
 - Enabled: true
 - Authenticator Id: RpkAuthenticator
 - System Id: N4:demo3:Tst-8E63-85FB-F946-98D7
 - System Guid: 9b2c987f-637a-42f3-baac-769372f48d22
 - System Type: n4-station
 - Registration State: Registered
 - Authentication State: Authenticated
 - System Ownership Code: 9015d20278728ad99ae061ca615d377b8bb3798a
 - Registration Url: https://[redacted].cloud.tridium.com
 - Authentication Url: https://[redacted]
 - Registration Web Url: https://[redacted].cbp.honeywell.com
- DevTestComp:** Dev Test Component

Transports Section:

- Transports Folder:** Contains 'HTTP Transport' and 'AMQP Transport'.
- HTTP Transport:** Http Transport
- AMQP Transport:** Amqp Transport
 - Status: (down)
 - Fault Cause: (empty)
 - Enabled: true
 - Message Retries: 2 [0-10]
 - Compression: Gzip
 - Message Throttling Limit: 5 [0-max]
 - Authenticator Id: RpkAuthenticator
 - Status Message: client lost connection. Attempting to re
 - Connect Retry Interval: +00000h 00m 20s
 - Connection Type: AMQP

To confirm, look at the station output as a connection problem. Set the cloudLink.transport.amqp, cloudLink.transport.http, and cloudLink.auth logs to ALL.

Figure 23. Station output for AMQP blocked

```

CONFIG [21:35:17 23-Feb-21 EST][cloudLink.auth.forge] Completed RPK Challenge - 1656 ms
CONFIG [21:35:17 23-Feb-21 EST][cloudLink.auth.forge] Starting System Connections
CONFIG [21:35:18 23-Feb-21 EST][cloudLink.auth.forge] Completed System Connections: 765 ms
CONFIG [21:35:18 23-Feb-21 EST][cloudLink.auth.forge] Next RpkAuthenticator token renewal s
CONFIG [21:35:35 23-Feb-21 EST][cloudLink.auth.forge] Expiration time from token: 02-Mar-21
FINEST [21:35:35 23-Feb-21 EST][cloudLink.transport.amqp.client] EventImpl{type=REACTOR_INI
FINEST [21:35:35 23-Feb-21 EST][cloudLink.transport.amqp.client] EventImpl{type=CONNECTION_
FINEST [21:35:35 23-Feb-21 EST][cloudLink.transport.amqp.client] EventImpl{type=SESSION_LOC
FINEST [21:35:35 23-Feb-21 EST][cloudLink.transport.amqp.client] EventImpl{type=CONNECTION_
FINEST [21:35:36 23-Feb-21 EST][cloudLink.transport.amqp.client] EventImpl{type=LINK_INIT,
FINEST [21:35:36 23-Feb-21 EST][cloudLink.transport.amqp.client] EventImpl{type=LINK_INIT,
FINEST [21:35:57 23-Feb-21 EST][cloudLink.transport.amqp.client] EventImpl{type=TRANSPORT_E
INFO [21:35:57 23-Feb-21 EST][cloudLink.transport.amqp] AMQP client lost connection. Attempt
java.io.IOException: Error{condition=amqp:connection:framing-error, description='connection
    at com.tridium.cloudLink.transport.internal.AmqpClient.onTransportError(AmqpClient.
    at org.apache.qpid.proton.engine.BaseHandler.handle(BaseHandler.java:101)
    at org.apache.qpid.proton.engine.impl.EventImpl.dispatch(EventImpl.java:100)
    at org.apache.qpid.proton.reactor.impl.ReactorImpl.dispatch(ReactorImpl.java:324)
    at org.apache.qpid.proton.reactor.impl.ReactorImpl.process(ReactorImpl.java:291)
    at com.tridium.cloudLink.transport.internal.AmqpClient.lambda$null$0(AmqpClient.jav
    at java.security.AccessController.doPrivileged(Native Method)
    at com.tridium.cloudLink.transport.internal.AmqpClient.lambda$connect$1(AmqpClient.
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)

```

Solution

For a rapid solution, switch to AMQP over WebSocket by changing the setting in the **NiagaraCloudService > Transports > AMQP Transport**.

If you really want to use AMQP, try working with your IT administration to modify the network settings to allow this protocol.

For most internal networks, AMQP is blocked by default. So, any device on the network needs to use AMQP over WebSocket.

Cloud backup channel issues

When you upload backups to the Niagara Cloud Service, variations in network performance may cause issues.

In these cases, the cloud backup will fail and an error message similar to the following will show in the station log:

Failed [24-Apr-24 2:27:08 PM] Job Failed

```

java.lang.RuntimeException: java.util.concurrent.ExecutionException: java.net.SocketTimeoutException: timeout
    at com.tridium.cloudLink.azure.file.AzureBlobOutputStream.close(AzureBlobOutputStream.java:158)
    at com.tridium.cloudLink.azure.file.AzureBlobFileUploader.lambda$doUpload$0(AzureBlobFileUploader.java:52)
    at java.security.AccessController.doPrivileged(Native Method)
    at com.tridium.cloudLink.azure.file.AzureBlobFileUploader.doUpload(AzureBlobFileUploader.java:47)
    at com.tridium.cloudLink.file.FileUploader.lambda$upload$2(FileUploader.java:51)
    at java.util.concurrent.CompletableFuture.uniWhenComplete(CompletableFuture.java:774)
    at java.util.concurrent.CompletableFuture$UniWhenComplete.tryFire(CompletableFuture.java:750)
    at java.util.concurrent.CompletableFuture$Completion.run(CompletableFuture.java:456)

```

```

at java.lang.Thread.run(Thread.java:750)
Caused by: java.util.concurrent.ExecutionException: java.net.SocketTimeoutException: timeout
at java.util.concurrent.CompletableFuture.reportGet(CompletableFuture.java:357)
at java.util.concurrent.CompletableFuture.get(CompletableFuture.java:1928)
at com.tridium.cloudLink.azure.file.AzureBlobOutputStream.close(AzureBlobOutputStream.java:152)
... 8 more
Caused by: java.net.SocketTimeoutException: timeout
at okhttp3.internal.http2.Http2Stream$StreamTimeout.newTimeoutException(Http2Stream.kt:675)
at okhttp3.internal.http2.Http2Stream$StreamTimeout.exitAndThrowIfTimedOut(Http2Stream.kt:684)
at okhttp3.internal.http2.Http2Stream$FramingSink.emitFrame(Http2Stream.kt:567)
at okhttp3.internal.http2.Http2Stream$FramingSink.write(Http2Stream.kt:532)

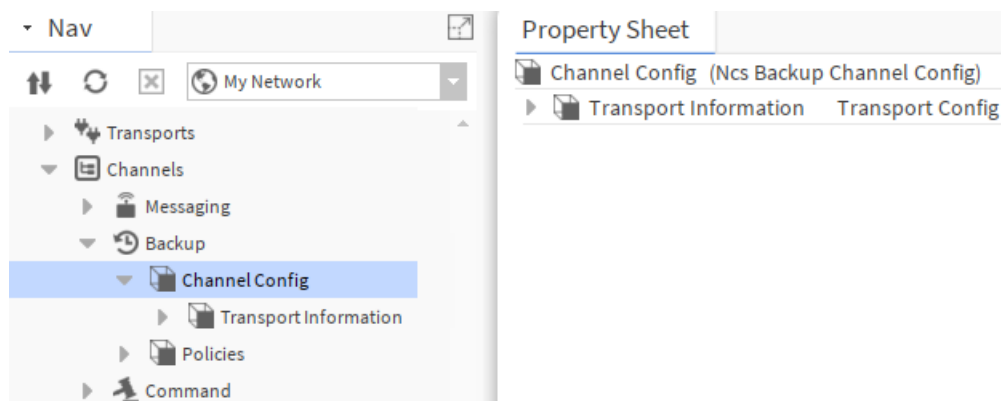
```

Since a cloud backup can result in a large file, the entire backup file is not sent to the cloud service in one request but rather split into smaller data items called blocks. These blocks are then sent to the cloud in multiple upload requests. The default block size is 4 megabytes but sometimes this is too large and needs to be reduced. **Solution:** Set the block size to 2 megabytes.

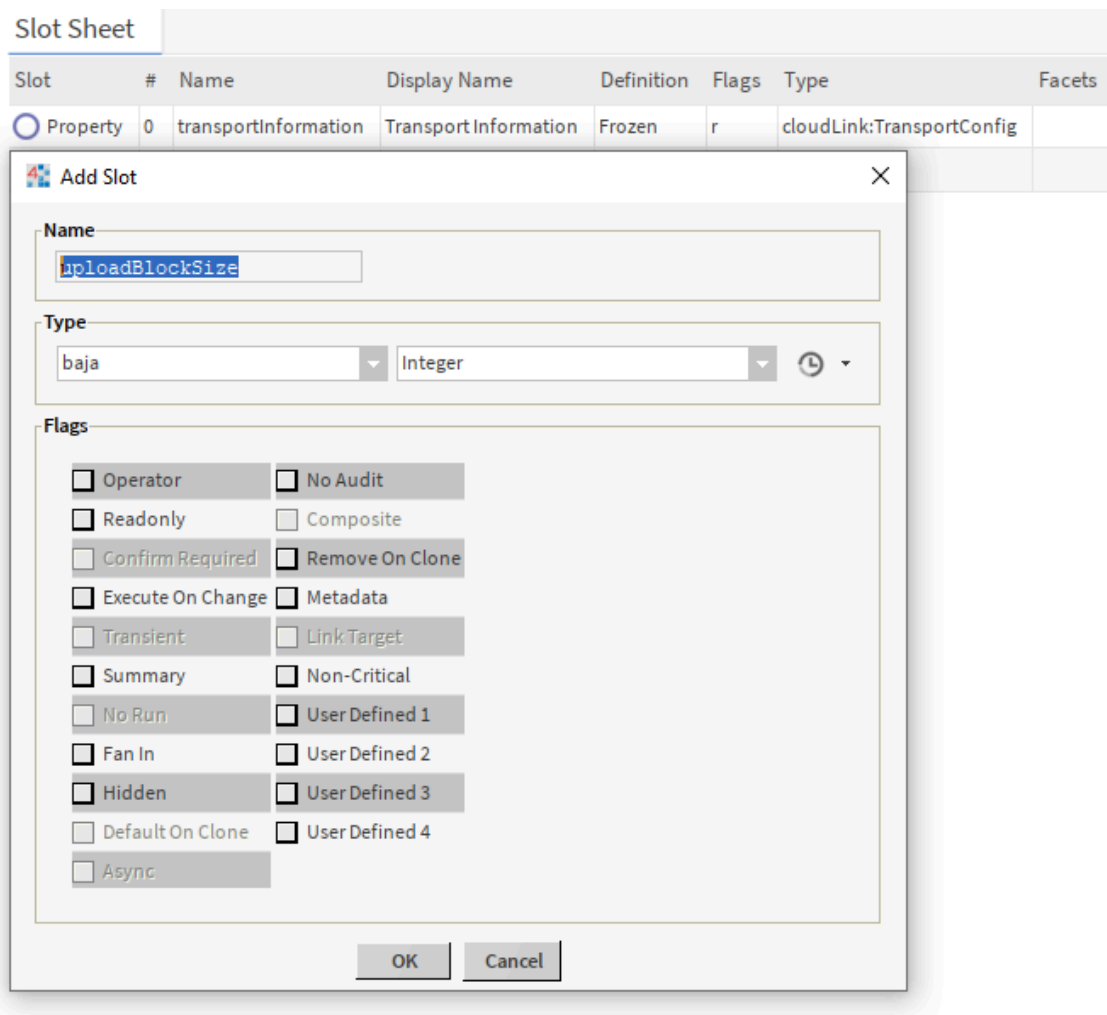
Setting cloud backup size in Workbench

The following steps describe how to set the cloud backup block size to 2 megabytes in Workbench.

- Step 1. In Workbench, expand Niagara Cloud Service > **Channels** > **Backup** and double-click on the **Channel Config** component to open the **Channel Config** property sheet.

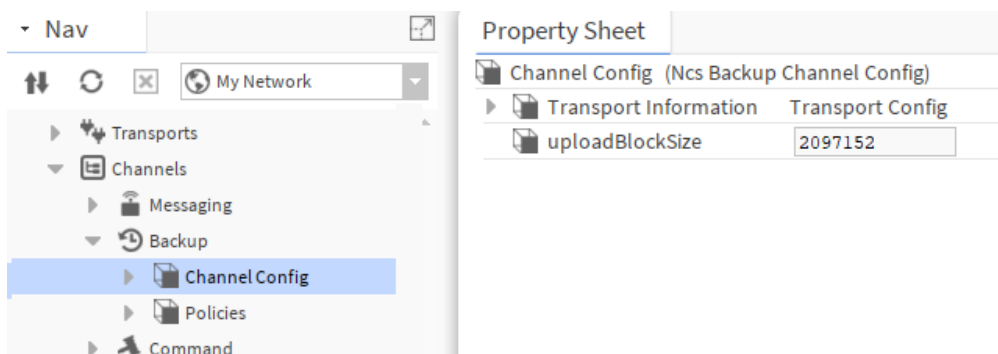


- Step 2. Right-click on Channel Config and select **Views** > **AX Slot Sheet** in the popup menu. The **Slot Sheet** view opens.
- Step 3. Right-click anywhere in the Slot Sheet view and select **Add Slot** from the popup menu. The **Add Slot** window opens.
- Step 4. In the **Add Slot** window, enter `uploadBlockSize` for **Name**.



Step 5. In the **Add Slot** window, select `baja:Integer` for **Type** and click **OK** to save the new slot.

Step 6. Navigate back to the **Channel Config** property sheet and set the value of the newly added property `uploadBlockSize` to 2097152 (2 megabytes).



Step 7. Click **Save** and execute the Default Backup Policy in the Backup Channel **Policies** folder to make a cloud backup.

Workaround for Directory Ord Chooser issue

The following steps ensure that you enter the correct folder path for the **Exclude Folders** property on the **Cloud Backup Policy** component.

Step 1. To the right of the **Exclude Folders** property, select the + icon.

Backup Note	backup notes
Encryption Key	System Passphrase
Exclude Files	*.lock;*.backup*;console.*;config.bog.b*;
Exclude Folders	<div> <div>file:^^webFileCache</div> <div>file:^^cloudLinkModel</div> <div>file:^^cloudLinkSchedule</div> <div>file:^^orientSystemDb</div> </div>
Alarm On Failure	<input checked="" type="radio"/> true

The Ord window opens.

The Ord window is a small dialog box with a title bar containing a Niagara logo and the text 'Ord'. It has a close button (X) in the top right corner. The main area contains a text field with the text 'null' selected. To the right of the text field is a folder icon with a dropdown arrow. At the bottom are 'OK' and 'Cancel' buttons.

Step 2. In the Ord window, click on the folder icon to open the Directory Chooser window.

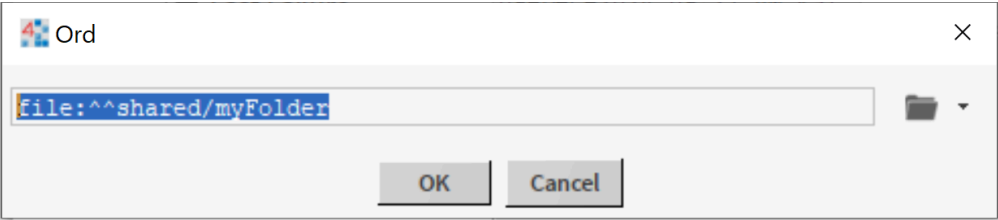
The Directory Chooser window has a title bar with a Niagara logo and the text 'Directory Chooser', and a close button (X) in the top right corner. It features a 'File Spaces' tab on the left with a tree view showing 'Files on Station (myStation)' and 'My Modules'. Below this is a 'Bookmarks' section. The main area on the right shows a list of files and folders under the 'Files' header, including 'myFolder' and 'myFolder2'. At the bottom are 'Choose' and 'Cancel' buttons.

Step 3. Navigate to and select the folder that you want to exclude and click **Choose**. The Ord window displays the selected folder. For example, `file:^myFolder` where `myFolder` is the intended excluded folder name. However, this value is incorrect. You must change it in order for the Cloud Backup Service to exclude it from the backup.

The Ord window is shown again, but now the text field contains the text 'file:^myFolder'. The folder icon to the right of the text field is still present. The 'OK' and 'Cancel' buttons remain at the bottom.

Step 4. To correct this, edit the folder path by replacing the leading caret ^ with the prefix ^^shared/

For example, file:^^shared/myFolder



- Step 5. Click **OK**.
The correct folder is listed in the **Excluded Folders** list and will be excluded from cloud backups.

Property Sheet

Default Backup Policy	Cloud Backup Policy
Status	{ok}
State	Idle
Enabled	<input checked="" type="checkbox"/> true
Execution Time	2:00 AM {Sun} +~1 hour
Last Attempt	18-Dec-2024 09:14 AM EST
Last Success	18-Dec-2024 09:14 AM EST
Last Failure	09-Dec-2024 09:27 AM EST
Fault Cause	
Backup Note	backup notes
Encryption Key	System Passphrase
Exclude Files	*.lock;*.backup*;console.*;config.bog.b*;...
Exclude Folders	file:^^webFileCache file:^^cloudLinkModel file:^^cloudLinkSchedule file:^^orientSystemDb file:^^shared/myFolder
Alarm On Failure	<input checked="" type="checkbox"/> true

Chapter 6. Tuning

This section provides basic guidelines for tuning the `NiagaraCloudService` part of the Niagara Cloud Service feature.

Tuning Considerations

Niagara Cloud Service was designed with lessons learned from Niagara Cloud Honeywell Sentience Driver, and with greater knowledge of the capabilities and limitations of the cloud platforms to which it might connect. Niagara Cloud Service was created to minimize the number of required configuration options to optimally set up the service. For this reason, there may not be many things to change for most users. This tuning guide touches on some of the configuration options, and discusses when you might need to adjust them.

Individual Stack Components

Most of the pieces of the `NiagaraCloudService` have individual enable flags, so they can be separately enabled or disabled. In most cases, you should not disable any parts of the service, as most cloud applications depend upon all data streams being in place. However, it may be easier to diagnose a problem with an individual component if you disable the other components that are in parallel with the component under investigation. Do not disable the component(s) used by the aspect of the `NiagaraCloudService` you are investigating.

NiagaraCloudService

SMA Expiration Monitor

The default configuration for this property is to warn you when the current Software Maintenance Agreement (SMA) has 30 days or less remaining in its valid period. This was taken from existing implementation, but if your organization might require a longer time to process license changes, you may wish to increase this, to avoid service interruption. Note that the warning only appears in the station's **Application Director** output, so for a notification that can be sent externally, there is an Alarm Source Info configuration available for delivery to recipients via email or mobile phone.

Authenticators

RpkAuthenticator (Honeywell Forge Platform)

This authenticator provides authentication to the Honeywell Forge platform using the platform mandated RPK Exchange over TLS. The authenticator is pre-configured with the items you will need, such as the platform type and connection URLs. The different configurations needed for different platforms and environments are provided by specialized palettes.

Transports

The different transports contain parameters that resemble common “tuning” parameters, like timeouts, retries, and limits. However, in most cases, you should not have to adjust these parameters as the defaults have been selected to optimize the bandwidth usage. There are a few items to note about transport properties.

Common properties

- **Message Retries**

This will enable the connection to be robust across minor glitches that drop individual messages. You can experiment with higher values if you experience frequent connection drops, or lower values if the network connectivity is extremely reliable.

- **Compression**

The compression choice is usually specified by the cloud platform. Sometimes there is an option whether to use compression or not. Most of the data messages sent by Niagara Cloud Service achieve high

compression rates, so if it is an option for the cloud platform, it is recommended to use it.

NOTE: Message size limits are pre-compression so if you turn compression off the maximum message size may need to be adjusted. For example the IoT Hub has a maximum message size of 250 KB but the channels that use it have a default max message size of 1 MB. So if compression is turned off then the maximum message size needs to be adjusted down to 250KB.

- **Message Throttling Limit**

Some cloud platforms may have a limit to the number of messages that can be sent within a time frame. In addition, connection over a narrow or metered bandwidth connection may require limiting the data flow rate. This limit will cause the NiagaraCloudService to throttle the number of messages sent through this transport to this number per second. This is a sliding window, and when the window is full, the transport will simply pause message delivery. A value of zero indicates no throttling.

Channels

Each channel has a Channel Config object, where most tuning parameters are contained. As with the rest of Niagara Cloud Service, there should not be much tuning needed because the defaults are chosen to optimize communication for most scenarios. There are a few common configuration options for most channels. Channels with configuration properties beyond the common ones are discussed next.

Common properties

- **Channel Queue**

- This queue holds messages relevant to the channel. The queue size can be configured to hold more or fewer messages. In addition, the queue weight can be configured. The transport layer uses a weighted round robin approach to accept messages, so increasing the weight on one queue will cause its messages to be sent more frequently than messages from other queues. This may be useful if you need to ensure that certain message types like alarms are sent with priority over telemetry data. It is not recommended to modify these parameters unless you have determined that needed messages are being held up by less important ones.

- **Max Message Size**

- This specifies the largest message that the channel will send. In general, bandwidth is optimized by fewer larger messages, than by many small messages, so this size is generally large.

Commands

The Commands channel allows configuration of the individual commands known by the system. All commands can be individually enabled or disabled. The default is enabled, and it is not recommended to disable them unless you are specifically having a problem with a command. Some commands have additional configuration parameters.

- **Application IDs (Honeywell Forge)**

There are some application ids that are configured in the Commands container, that allow Forge applications to send commands to the system. These should not be modified.

- **Write Commands**

Write commands have a default priority and duration that will be used when the calling application has not provided one. You can adjust this on each of the write commands to configure how writes will be handled.

- **Command Queue**

This queue is where incoming commands are stored to await execution. When a command comes from the cloud it is enqueued, and the result of the command is returned asynchronously. This allows for command execution to not hold up the response that it was received.

Heartbeat

The Heartbeat channel sends a message to inform the cloud that the system is still healthy; it is analogous to the "ping" message sent by Niagara drivers. The default frequency for this heartbeat message is 5 minutes,

but can be adjusted if needed. It would be recommended to set this as high as possible, subject to the cloud platform's needs for identifying failed systems, as it provides little value and consumes messaging bandwidth.

Histories

The Histories channel allows the export of histories from the Niagara history database to the cloud platform. This is done with Cloud History Export Configs, which are configured with an execution time trigger, and contain a list of histories. Export Configs are configured in the **Cloud History Export Manager** view, where histories can be discovered and assigned to Export Configs, or included/excluded from the Auto Export Config.

Max Concurrent Export Executions

If the Auto Export is enabled, it exports ALL histories in the station, except for those that are added to its exclusion list. When you enable this, make sure that you have excluded anything you do not want sent to the cloud platform.

Other Configs use an include approach, so you need to assign a history to that config for it to be exported. Note that a history can be assigned to at most one config.

CAUTION: Avoid data duplication. You can assign a history to an export config for exporting, and also exclude it in the Auto Export Config. In fact, that is the strongly recommended configuration. If the Auto Export Config is used in addition to regular Export Configs, you should make sure that histories in the Export Config are excluded from the Auto Export Config to avoid data duplication in the cloud platform.

NOTE: Histories that are sourced by points that are excluded from the cloud (see Points >Excluding points) are not included for selection in the user interface. The history will not be exported even if the history is in the Export Config.

This governs how many simultaneous threads will be used to export history data. Using more threads (up to the number of export policies that are used) will allow the export to proceed more quickly, but may tax the resources of either the gateway, or the connection bandwidth.

Messaging

There are no specific configuration properties in the Messaging channel, but it is worth noting that it uses multiple transports, because it is used by other components for sending individual messages.

Model

- The Model channel uses several configuration properties to control how model information is created and sent. You can control whether the model file is sent to the cloud, and whether a copy of it is stored locally. In addition, the status of the processing of the model information can be seen here. You should not need to adjust these parameters in most cases.
- **Cloud Id Export Policy**
The ComponentExportPolicy allows the export of station components to a model service. This can be configured with an execution time trigger, although in most cases model synchronization is a manual action done once the system is properly configured, and only repeated if the site/system configuration changes.
 - **Scope** — is used to control where components are exported from.
 - **Component Types** — is used to identify the station types that will be included in the model information exported to the cloud platform.

Points

The Points channel exports station control points to the cloud platform. In many cases, this ends providing a data stream similar to what the Histories channel provides, but it is a "snapshot" of the point's value. It could be used to reduce the history database size needs on the station, although it does not necessarily capture every change of value, as the point's instantaneous value is captured at each export. The point export is controlled with Point Export Policies. Each policy specifies points to be captured, and a frequency at which to export the data.

- **Custom Point Queries**

This field is an **Ord List** that specifies which points will be exported. You can use this to tailor the list of points specifically for the policy.

- **Excluding points**

Multiple policies can be used to capture very granular choices in which points to export and not export. Points can be excluded from the export using the `nc:excluded` tag.

- **Max Concurrent Export Executions**

This governs how many simultaneous threads will be used to export point data. Using more threads (up to the number of export policies that are used) will allow the export to proceed more quickly, but may tax the resources of either the gateway, or the connection bandwidth.

Chapter 7. Appendix

This section provides supplementary information and clarification topics.

Configuring the station to receive commands

The CloudAuthenticationScheme, which is added automatically to the station's AuthenticationSchemes node, is required to configure at least one **CertTrustMapping** or **JwksTrustMapping** component.

NOTE: For successful configuration, you need to have at least one **CertTrustMapping** or **JwksTrustMapping**. You can have more than one and you can have any combination you want in order to enable your station to accept tokens from all desired token providers.

Adding a CertTrustMapping

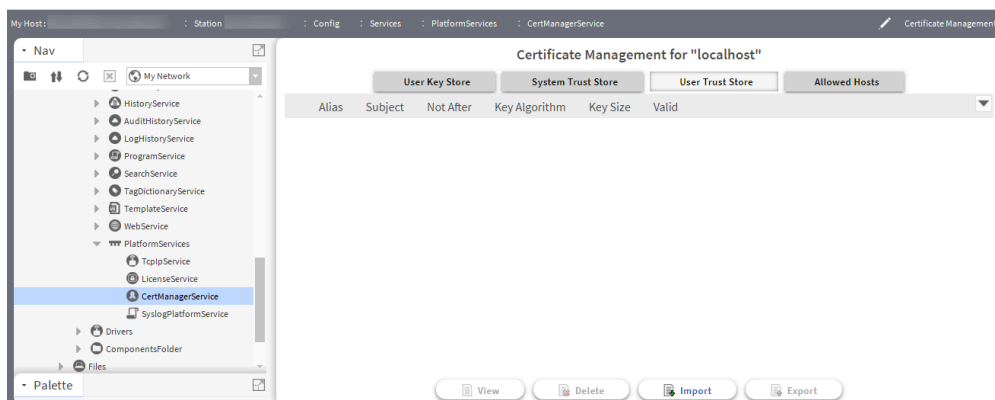
This procedure describes how to add a CertTrustMapping to a station.

Prerequisites:

- The public certificate of your token provider.
- The CloudAuthenticationScheme is already installed.
- The cloudLink palette is open.

Step 1. In the Nav pane, expand **Services > PlatformServices > CertManagerService**.

Step 2. Select the **User Trust Store** tab.



Step 3. Click the **Import** button.

Step 4. Browse to the location of your token provider's public certificate file and open it.

Step 5. Enter an alias for the certificate and click **OK**.

Step 6. In the Nav tree, expand **AuthenticationService > CloudAuthenticationScheme** component.

Step 7. From the cloudLink palette, drag a **CertTrustMapping** component onto the Trust Manager node under the CloudAuthenticationScheme in the Nav tree.

Step 8. Open the **Property Sheet** view on the new CertTrustMapping component.

Property Sheet	
JwksTrustMapping (Jwks Trust Mapping)	
App Id	
Expected Jwt Audience	NiagaraCloudService
Expected Jwt Issuer	
Jwks Endpoint	

NOTE: The app Id value is required for CertTrustMapping to work for the application ID.

Step 9. In the **App Id** field, enter the Honeywell Forge application ID.

NOTE: A valid value is required for the mapping to be successful.

Step10. In the **Expected Jwt Issuer** field, enter the value of the token issuer "iss" field (typically the URL of the user identity provider).

NOTE: The value for the Token issuer "iss" field must to be provided by the Developer/Integrator during Certificate Trust Mapping configuration.

Step11. In the **Expected Jwt Audience** field, enter the value of the token audience "aud" field.

By default, it is "Niagara Cloud Service", but this may be changed to match the value present in the JWT for those providers that do not have a fully configurable audience field. For example, Salesforce prepends the Salesforce application Id (not to be confused with the Honeywell Forge application Id) onto the audience.

Step12. In the **Certificate Alias** field, expand the dropdown and select the alias of the certificate that was imported above.

Step13. Save **CertTrustMapping**.

Result

Certificate Trust Mapping is now available and the station is configured to receive commands.

Adding a JwksTrustMapping

This procedure describes how to add JwksTrustMapping to a station.

Prerequisites:

- URL of the token issuers key service.
- The CloudAuthenticationScheme is already installed.
- The cloudLink palette is open.

Step 1. In the Nav pane, under **AuthenticationService**, expand **AuthenticationSchemes** > **CloudAuthenticationScheme** > **TrustManager**.

Step 2. From the cloudLink palette drag a **JwksTrustMapping** component onto the **Trust Manager** node under the **CloudAuthenticationScheme** in the Nav tree.

Step 3. Open the **Property Sheet** view of the added **JwksTrustMapping** component.

Property Sheet	
JwksTrustMapping (Jwks Trust Mapping)	
App Id	
Expected Jwt Audience	CloudLink
Expected Jwt Issuer	
Jwks Endpoint	

Step 4. In the **App Id** field, enter the Honeywell Forge application id.

NOTE: This value is required.

Step 5. In the `Expected Jwt Audience` field enter the value of the token audience "aud" field. By default, it is "Niagara Cloud Service", but this may be changed to match the value present in the JWT for those providers that do not have a fully configurable audience field. For example, Salesforce prepends the Salesforce application Id (not to be confused with the Honeywell Forge application Id) onto the audience.

Step 6. In the `Expected Jwt Issuer` field, enter the value of the Token issuer "iss" field (typically the URL of the user identity provider).

NOTE: The value for the Token issuer "iss" field must to be provided by the Developer/Integrator during `JwksTrustMapping` configuration.

Step 7. In the `Jwks Endpoint` field enter the URL of the token issuer's key service.

Step 8. Click **Save** to save the `JwksTrustMapping` property sheet settings.

Result

`Jwks Trust Mapping` is now available and the station is configured to receive commands.

About Role Mappings

Roles are used to specify the authorization to station resources for System Commands.

The roles that are authorized in the cloud application are contained in the security token sent with a System Command. These are in a claim called "cloudroles", which is a comma separated list of text strings. For example: "cloudroles": "CloudRole-Admin, CloudRole-Operator". The Role Mappings component provides a way to match the cloud roles to actual roles on the station. So if the cloud role is "CloudRole-Operator", it can be mapped to the role of "CloudOperator" on the station. . There should be one role mapping for each cloud role contained in your security token. More than one cloud role can be mapped to the same station role if necessary.

Once configured, the station is ready to receive commands with the specified cloud roles.

NOTE: You need to add one role mapping for each cloud role contained in your security token. More than one cloud role can be mapped to the same station role if necessary.

Standard pre-configured roles

The Role Mappings component creates three standard station roles as a convenience. These are `CloudUser`, `CloudOperator` and `CloudAdmin`.

By default, each cloud role is "enabled" and has `=viewable hierarchies` set to "none". The default values for the permissions of these roles are shown in the following table.

Standard role name	Default permissions
<code>CloudUser</code>	1=rR;
<code>CloudOperator</code>	1=rwiRWI
<code>CloudAdmin</code>	1=rwiRWI; 2=rwiRWI

These can be removed if necessary. Any role can be created for use with the Role Mappings component.

To prevent these standard roles from being created upon station start up, set the property **Reassert Missing Standard Roles** in the Role Mappings component to "false".

To recreate these standard roles, set the property **Reassert Missing Standard Roles** in the Role Mappings component to `true`. The standard roles will be created upon station start. If the standard roles are already present in the Role Service, they will not be replaced.

NOTE: If the permissions of the standard roles are modified, the modified permissions will remain in effect. If the Standard Role is being used by any RoleMapping in the Cloud Authentication Scheme, then a warning (like the example shown) is logged in the Application Director when this event occurs. Permissions for role CloudOperator (1=rwi;2=r) have been changed from default value of 1=rwiRWI. Please ensure that this is intentional by reviewing the station RoleService.

WARNING [12:18:24 28-Nov-18 EST][ncloud.security] Permissions for role CloudOperator (1=rwi;2=rwiRWI) have been changed from default value of 1=rwi;2=r. The RoleMapping (RoleMapping-Operator) that maps to this role has been disabled.



Configuring Role Mappings (optional)

This procedure describes how to configure roles that are used to specify the authorization to station resources for System Commands.

Prerequisites:

- CloudAuthenticationScheme and JwksTrustMapping are already configured
- The cloudLink palette is open.

NOTE:

- The NiagaraCloudService automatically adds the Role Mappings container to the CloudAuthenticationScheme. It also automatically adds several Role Mapping components in the container.
- You can add or change one role mapping for each cloud role contained in your security token (optional).
- More than one cloud role can be mapped to the same station role if necessary.

Step 1. In the Nav Tree under AuthenticationService, expand **Authentication Schemes > cloudAuthenticationScheme > Role Mappings**.

Step 2. In the cloudLink palette, expand the Authentication folder, and drag a **Role Mapping** component onto the **Role Mappings** component expanded in the previous step.

Step 3. For the **Cloud Role** property, enter the exact name of one of the Cloud Roles that will be in the claim of your security token. You can get the cloud roles from the pre-configured role mappings.

The roles that are authorized in the cloud application are contained in the security token sent with a System Command. These are in a claim called "cloudroles", which is a comma separated list of text strings. For example: "cloudroles": "customer_admin, customer_user".

The standard Niagara Cloud Service pre-configured roles and their default permissions are as follows:

- CloudUser: 1=rR
- CloudOperator: 1=rwiRWI
- CloudAdmin: 1=rwiRWI;2=rwiRW

For more information, see "cloudLink-RoleMappings" component topic.

Step 4. For the **Station Role**, enter the exact name of an existing role in the **RoleService** of the station.

NOTE: Do not enter the default "Admin" role for the station role. Any role mapping with a station role of "Admin" will be ignored for security reasons.

Step 5. Click **Save**.

Result

The station is now ready to receive commands with cloud roles specified.

Chapter 8. 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

Cloud Id

A component's unique identifier in the cloud indicated with a direct marker tag nc:cloudId.

dist

A Niagara backup distribution file that includes the station folder as well as other configuration information that can be customized for the platform.

edist

File extension for the legacy Backup as a Service product. These files are not compatible with Niagara Cloud Service.

edist2

A Niagara Cloud Service file extension for encrypted dist files sent the cloud.

Federated Device Registration

Provides a method to associate a single device with each cloud service. If preferred, registration can even occur at a later date.

Host ID

A host ID (or hostid) is an alphanumeric code that identifies the device that is authorized to run the software.

Niagara Cloud Management Portal

A platform on which one or more Systems Integrators (SI) are responsible for configuring the on-premise and cloud platforms of one of more customers. In this way, an SI has multiple customers as sub-tenants so that the SI can have a single cloud login to service all their customers.

Niagara Cloud Suite

Niagara Cloud Suite is a collection of user-facing (and some non-user-facing) web applications, running within the Tridium-provided Azure cloud space, providing a known set of service features.

Niagara Data Service

A per customer instance of service that collect telemetry (history) and model data from on-premise devices and allow API access to that data in the cloud.

Niagara Data Service (NDS)

Project (NCS)

An organizational unit created by a partner that contains a set of devices.

Telemetry Id

An identifier used to identify for time series data in the cloud including histories, point values, log data, and so on. It is indicated with a nc:telemetryId direct marker tag .