

Technical Document

Niagara Platform Guide

July 16, 2023

niagara⁴

Niagara Platform Guide

Tridium, Inc.

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

Confidentiality

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

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

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

Trademark notice

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

Copyright and patent notice

This document may be copied by parties who are authorized to distribute Tridium products in connection with distribution of those products, subject to the contracts that authorize such distribution. It may not otherwise, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior written consent from Tridium, Inc.

Copyright © 2023 Tridium, Inc. All rights reserved.

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

Contents

About this Guide	9
Document Change Log	9
Related documentation	10
Chapter 1 Platform Management	11
Supported platforms	11
Backup battery (or not)	13
Known 64-bit limitations and installation	13
Platform daemon (niagarad)	14
Installing and starting the platform daemon	14
Opening a platform connection to a host	14
Creating and updating the HSQL database password	18
Provisioning—a way to automate platform tasks	19
Platform Nav Container View	19
Exporting the Nav Container View	20
Chapter 2 Platform and station file systems	23
Supervisor homes	23
Controller homes	25
Windows user homes	26
Station homes	28
Shared file strategy	30
Running a station from the Workbench User Home	30
Chapter 3 Application Director (station management)	31
Starting and stopping a station	32
Standard output messages	33
Configuring station output	34
Configuring station log history	35
Setting up the station spy	36
Chapter 4 Certificate management (Platform security)	39
Certificate Wizard	39
Generating a CA certificate and signed Server certificate using the Certificate Wizard	40
Recommended verifications	44
Chapter 5 Distribution File Installer	47
Restoring a backup distribution file	47
Returning a controller to near factory defaults	51
Chapter 6 Lexicon Installer	53
Changing the language (Lexicon Installer)	53
Chapter 7 License Manager	55
About license files	55
Hardware features that can be licensed	57
Driver attributes	58

Driver types	59
Applications.....	60
Submitting a license request	62
Importing a host license from the license server.....	63
Installing a controller license	65
Exporting a license	67
Remote license management	68
Preparing the Supervisor's license database	69
Automating host license updates using provisioning	69
Updating licenses using the Network License Summary	70
Synchronizing with the Supervisor's license database.....	71
Subscription licensing.....	71
Requirements.....	71
Registering Niagara running in a container	72
Register Niagara using the NRE Command.....	74
SMA expiration reminder.....	76
Chapter 8 Platform Administration on a controller	79
Platform Administration on an embedded controller.....	80
Platform Administration on a workstation.....	81
Viewing platform details	82
Setting up the PC platform's users group (workstation).....	83
Creating a platform user account (controller).....	85
Changing a user's platform password (controller).....	86
Deleting a platform user account (controller)	87
System and file passphrases.....	87
Changing the platform's system passphrase	88
Editing the .bog file passphrase offline.....	89
System passphrase.....	90
Changing the platform daemon's HTTP port.....	90
Changing TLS Platform settings	91
System date, time and time zone	92
Time Zones	92
Daylight saving time	93
Changing a remote platform's date, time and time zone	94
Displaying current CPU usage	95
Viewing daemon output	97
Viewing controller system logs.....	99
Controller storage conservation.....	100
Changing a module's runtime profile	101
Configuring memory to improve performance	102
Backing up a station using Platform Administration.....	103
Commissioning a controller	103
Configuring TCP/IP settings.....	105
Upgrading a controller	107
Rebooting a controller station.....	108

Chapter 9 Software Manager	111
Viewing the PC's software database	111
Setting up the software database in the PC	113
Installing modules in a remote platform.....	114
Upgrading out-of-date modules.....	117
Removing modules from a remote platform.....	117
Chapter 10 Station Copier	119
Copying a station	120
Station Copier dependencies check	121
Station Transfer Wizard	122
Transferring a station to a controller	122
Backing up a station using Station Copier	126
Renaming a station.....	128
Deleting a station.....	130
Station installation troubleshooting.....	131
Chapter 11 File Transfer Client	133
Transferring files to and from a remote host	133
Chapter 12 Platform tools	137
Creating a new station.....	137
Copying a new station to the daemon user home	138
Importing a license to the local license database.....	139
Requesting a license from the license server	140
Exporting a license file using a tool	141
Deleting a license file using a tool	144
Using a tool to synchronize licenses	145
Chapter 13 Platform Services	147
Monitoring power to a controller	149
About the NtpPlatformService.....	151
Verifying access to an NTP server.....	151
Reverting to the legacy CPU usage for BACnet networks.....	152
Global capacity licensing	153
Example globalCapacity feature entry.....	154
Checking capacity licensing status	155
Capacity licensing and histories	158
Chapter 14 Supervisor components.....	159
Workbench License Manager (platform-LicenseDatabaseTool)	159
Chapter 15 Controller components	161
Data Recovery Service Editor (platDataRecovery- DataRecoveryService).....	161
Model-specific PlatformServiceContainer properties	162
NTP Platform Service (platform-NtpPlatformServiceNpsdk)	162
Platform Service Properties (platform-NtpPlatformServiceQnx).....	162
Platform Alarm Support (platform-PlatformAlarmSupport)	164

System Platform Service (platform-SystemPlatformServiceQnxJavelina)	164
System Platform Service (platform-SystemPlatformServiceQnxNpm6xx).....	164
Tcp/Ip Configuration (platDaemon-TcpIpConfiguration)	164
Hardware Scan Service View (platHwScan-HardwareScanService).....	165
Platform Service Properties (platMstp-BacnetMstpPlatformServiceQnx)	167
External SLA Battery (platPower-ExternalSlaBattery).....	167
Javelina Battery Platform Service (platPower-JavelinaBatteryPlatformService).....	167
NIMH Battery (platPower-NimhBattery)	168
NPM2 NIMH Battery (platPower-Npm2NimhBattery).....	168
NPM External SLA Battery (platPower-NpmExternalSlaBattery).....	168
Npm Dual Battery Platform Service (platPower-NpmDualBatteryPlatformService)	168
Power Monitor Platform Service (platPower-PowerMonitorPlatformServiceQnx).....	168
Serial Port Platform Service (platSerialQnx-SerialPortPlatformServiceQnx).....	169
Serial Port Service (platSerialQnx-SerialPortQnx)	170
Chapter 16 Shared components	173
Nav Container View (platCrypto-DaemonSecureSession)	173
Station Text Summary Editor view (platDaemon-StationTextSummaryEditor).....	175
Remote File System (platform-DefaultDaemonFileSpace).....	176
Daemon Session (platform-DaemonSession)	177
SyslogPlatformService (platform-PlatformServiceProperties).....	178
SystemMonitorService -(systemMonitor-SystemMonitorService)	180
Platform Monitor (systemMonitor-PlatformMonitor)	183
System Memory Monitor (systemMonitor-SystemMemoryMonitor)	183
Heap Memory Monitor (systemMonitor-HeapMemoryMonitor).....	185
Meta Space Memory Monitor (systemMonitor-MetaSpaceMemoryMonitor)	186
Loaded Classes Monitor (systemMonitor-LoadedClassesMonitor)	187
CPU Idle Cycles Monitor (systemMonitor-IdleCPUMonitor)	188
CPU Used Cycles Monitor (systemMonitor-UsedCPUMonitor)	189
Code Cache Memory Monitor (systemMonitor-CodeCacheMemoryMonitor)	191
RAM Disk Monitor (systemMonitor-RamDiskMonitor)	192
Socket State Monitor (systemMonitor-SocketStateMonitor)	193
Chapter 17 Plugin guides.....	195
Application Director view (platDaemon-ApplicationDirector).....	195

HTML-5 Certificate Management View	199
License Platform Service Plugin (platform- LicensePlatformService).....	202
Logger Configuration view (workbench-LoggerConfiguration)	202
Data Recovery Service Editor view (platDataRecovery- DataRecoveryServiceEditor)	203
Distribution File Installer (platDaemon-DistInstaller).....	204
File Transfer Client (platDaemon-FileTransferClient)	205
Hardware Scan Service view (platHwScan- HardwareScanServiceView).....	205
Javelina Battery Platform Service Plugin (platPower- JavelinaBatteryPlatformServicePlugin)	206
Lexicon Installer view (platDaemon-LexiconInstaller).....	206
License Manager view (platDaemon-LicenseManager)	206
License Platform Service Plugin view (platform- LicensePlatformServicePlugin)	207
Network License Summary view (provisioningNiagara- NetworkLicenseSummary)	208
Ntp Platform Service Editor (platform- NtpPlatformServiceEditorNpsdk).....	209
Ntp Platform Service Editor Qnx view (platform- NtpPlatformServiceEditorQnx)	209
Platform Administration view (platDaemon- PlatformAdministration)	211
Platform Certificate Management (platCrypto- CertManagerService).....	213
User Key Store tab	214
Trust Store tabs.....	215
Allowed Hosts tab	217
Platform Service Container Plugin (platform- PlatformServiceContainer).....	218
SystemService (under PlatformServices)	224
Power Monitor Platform Service Plugin (platPower- PowerMonitorPlatformServicePlugin).....	228
Software Manager view (platDaemon-SoftwareManager).....	228
Station Copier view (platDaemon-StationCopier).....	231
Syslog Platform Service Plugin view (platform- SyslogPlatformServicePlugin).....	232
System Date Time Editor view (platform-SystemDateTimeEditor).....	234
System Monitor Config (systemMonitor-SystemMonitorConfig)	234
System Platform Service Plugin (platform- SystemPlatformServicePlugin).....	235
System Platform Service Qnx Plugin (platform- SystemPlatformServiceQnxPlugin)	238
Tcp Ip Platform Service Plugin (platform-TcpIpPlatformService).....	239
Workbench Certificate Management (platCrypto- CertManagerTool).....	240
Workbench License Manager view (platform- WorkbenchLicenseManager).....	241

Chapter 18 Windows 243

- Advanced Platform Options..... 243
- Change TLS Settings window 244
- Configure NRE Memory Pools window 246
- Daemon Output Settings window 247
- Edit Filter window 248
- Export windows 250
- Import License window..... 251
- User Accounts 252
- New User window 253
- Set System Date/Time window 254
- Software Details window 255
- Sync Now window 255
- Syslog Configuration window 256
- View Details window..... 258
- Software Details window 259

Glossary 261

Index.....263

About this Guide

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

Product documentation

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

Document content

This document provides information about Niagara platform services, components and plugins, license tools and other topics related to the Niagara host.

Document Change Log

This topic summarizes the releases of this document.

July 16, 2023

For Niagara 4.13:

- Added references to JACE-9000 throughout.
- Added "Syslog Configuration window" topic to "Windows" chapter and "Syslog Platform Service Plugin view" to "Plugin Guides" chapter.
- Updated "Syslog Platform Service (platform-SyslogPlatformService)" component and "Platform Administration view".

May 24, 2023

Updated and reorganized for Niagara 4.13.

- Added new topics for "Niagara Subscription Licensing" to the "License Manager" chapter
- Added new topic "HTML-5 Certificate Management View" to the "Platform Plugins Guides" chapter
- In "Change TLS Settings", added "Certificate Password" options.
- In "User Key Store Tab", added "default" certificate.
- In "Change TLS Settings Window", added "Certificate Alias" and "Certificate Password" properties.
- Added "Syslog Configuration" topic to the "Platform Administration" chapter.

August 29, 2022

- In "Capacity licensing fault notifications" topics, changed estimated periodic recount time for global-Capacity count from 10 seconds to 10 minutes.
- Added "Reverting to legacy MS/TP processor" to the "Platform Services" chapter.
- Included "Components in the platMstp module" in the "Platform Component Guides" chapter.

December 14, 2021

Included in "Configure NRE Memory" the removal of the RAM Disk memory in the **Configure NRE Memory Pools** window as of Niagara 4.11.

September 29, 2021

Removed descriptions of support for **NTP Platform Service** on operating systems other than Qnx. Added a note describing removal of this platform service as of Niagara 4.9 and later.

October 8, 2020

Added component and plugin topics for the systemMonitor module.

August 7, 2020

Updated “Known limitations” noting the restriction requiring 64-bit PC.

January 9, 2020

Minor changes throughout to update for Niagara 4.9, replaced references to “applet” and “WebStart” with “Web Launcher”.

August 18, 2015

Initial release.

Related documentation

The following documents are related to the content in this guide and provide additional information.

- *Niagara 4 Installation Guide*
- *Niagara Station Security Guide*
- *Niagara AX to N4 Migration Guide*
- *JACE Niagara 4 Install and Startup Guide*
- *Niagara Drivers Guide*
- *Niagara Lexicon Guide*
- *Niagara LDAP Guide*
- *Niagara Lonworks Driver Guide*
- *NDIO Driver Guide N4*
- *Niagara 4 BACnet Driver Guide*
- *Niagara Modbus Driver Guide (N4)*
- *oBIX Guide – N4*
- *OPC UA Driver Guide*
- *Niagara Video Framework Guide*
- *Snmp V3 Driver Guide*
- *Niagara Data Recovery Service Guide*
- *Niagara Engineering Notes*

Chapter 1 Platform Management

Topics covered in this chapter

- ◆ Supported platforms
- ◆ Platform daemon (niagarad)
- ◆ Creating and updating the HSQL database password
- ◆ Provisioning—a way to automate platform tasks
- ◆ Platform Nav Container View
- ◆ Exporting the Nav Container View

At startup, Niagara, opens to the Web Browser View.

Figure 1 Web Browser View



The Nav tree on the left provides access to the available platforms and their stations. To see the platforms and stations, expand the **My Host** and IP address nodes.

From this Niagara home view, you open a connection to each platform and station.

This chapter introduces the supported platforms, explains the purpose of the platform daemon, and provides procedures for connecting to a platform, updating the HSQL database password and exporting the Nav Container View.

Supported platforms

A platform is everything that is installed on a host that is not part of a station. The platform interface addresses all the support tasks that allow you to set up, support and troubleshoot a host.

There are two types of platforms:

- Embedded controllers are purpose-built hardware devices shipped an operating system designed to optimize the controller functions. All use flash memory for file storage, Oracle's Sun Hotspot Java VM, and provide wired Ethernet connectivity.
- Supervisor PC hosts and/or engineering workstations that are Windows-based.

Among the two groups of controllers (embedded and Windows-based controllers), each model has a host model text descriptor, which you see in the **Station Manager** view of a **NiagaraNetwork** (Host Model column), and also in platform views, such as **Platform Administration**, as well as the **PlatformServices** container of a station running on a host.

This table lists controller models starting with the host model text descriptor. Models that are not compatible with Niagara 4 are so noted. Some models may exist on a job site where the Supervisor has migrated to Niagara 4 along with some number of controllers that are compatible. For background details, refer to the *Niagara AX to N4 Migration Guide*.

NOTE: For information about legacy platform controllers (JACE-6 and earlier) refer to earlier versions of this document.

Table 1 Platform host models

Model text descriptor	Model name	Notes	Compatible with Niagara 4?
ATLAS	JACE-9000	Linux-based, compact, embedded IoT (Internet of Things) controller and server platform for connecting multiple and diverse devices and sub-systems. With internet connectivity and web-serving capability, the JACE-9000 provides integrated control, supervision, data logging, alarming, scheduling and network management. It streams data and rich graphical displays to a standard web browser via an Ethernet or wireless LAN, or remotely over the internet. This controller supports Niagara 4. Refer to the <i>JACE-9000 Niagara 4 Install and Startup Guide</i> for commissioning details.	
TITAN	JACE-8000	QNX-based controller, with integral WiFi (802.11b/g) support, and a USB port for backup/restore usage using a USB flash drive. Plug-in option modules provide additional communications ports. Supports Niagara 4. Refer to the <i>JACE-8000 Install and Startup Guide</i> for commissioning details.	Yes, except when running older software

Model text descriptor	Model name	Notes	Compatible with Niagara 4?
Edge 10	Edge 10	is a QNX-based IO controller that harnesses the full power of the Niagara Framework® at the edge. Supports Niagara 4.7 and later. Ships with software to run a platform daemon, a tridium certificate, and a default station pre-installed. Edge devices run the full stack, with 10 points of on-board IO and IO-R-34 expansion capability. The controller features 512MB DDR SDRAM, 2GB total eMMC flash storage with user space set at 1GB, 5 Universal inputs, 2 Analog outputs, 3 Digital outputs, 2 10/100MB Ethernet ports capable of daisy chaining, 1 RS-485 serial port, real-time clock, and secure boot. Refer to the <i>Niagara Edge 10 Install and Startup Guide</i> for installation/commissioning details.	Yes
Workstation	User-supplied PC, for example, a Supervisor or engineering workstation.	Windows-based customer supplied PC that runs Workbench. The PC should be running a recent version of Windows. Most PC platforms use a 64-bit Windows OS (Win64-based). Although a Win64 Supervisor uses a 64-bit JVM (Java Virtual Machine) and different NRE core binaries, its platform interface is nearly identical to any Win32-based Supervisor (for sites using Niagara 4.8 and earlier). The 64-bit Java VM (Virtual Machine) does not have a 2 GB memory limit, unlike the Java VM on a Win32-based Supervisor. Typically, any PC with 64-bit Windows also has 4 GB or more of RAM, and, unlike a 32-bit Windows PC, the 64-bit OS can effectively use all of it. Therefore, if you are using an earlier Niagara version, a 64-bit Windows host may be the solution to improve the performance of your largest enterprise level Supervisor. File storage is, typically, a hard drive.	Yes

All controllers that can run Niagara 4 use the QNX operating system. Prior Windows-based controllers that run Windows XP Embedded are not supported.

Some platform views, for example, the **Platform Administration** view, differs depending on the type of controller.

NOTE: When connected to any Windows host, the **TCP/IP Configuration** platform view is always read-only. Intended configuration is for controllers only. On any Windows host, you configure TCP/IP and other network settings using the normal Windows Control Panel interface.

Backup battery (or not)

Controllers are designed to withstand brief power-loss events. A controller may or may not include a backup battery.

The Niagara Edge 10, JACE-8000, and JACE-9000 controllers include integral onboard memory, which preserves runtime data upon a power loss event. By default, these controller platforms do not have backup batteries.

A station running on a JACE-8000 or JACE-9000 has no **PowerMonitorService** under its **PlatformServices** container. For continuous operation across power events, an external battery-backed UPS is required to power the controller.

Backup memory support works via a station platform service, the **DataRecoveryService**. This platform service continuously records all database changes in memory, and upon reboot from a power event, restores these changes.

Known 64-bit limitations and installation

The latest Niagara versions require a 64-bit Windows-based PC. A Supervisor station running under a 64-bit Windows operating system may not support some legacy drivers. Although most limitations do not apply to a typical Supervisor, they should be understood before installation time.

NRE serial support is available for a 64-bit Windows platform. However, serial-based drivers (for example, modbusAsync, flexSerial, various legacy drivers) are not typically licensed on a Supervisor, and, therefore, are not fully tested or supported on a 64-bit platform.

Exceptions to such license rules can occur with 64-bit engineering workstations and demo machines. Again, 64-bit serial operation is not fully guaranteed.

A known issue with the 64-bit serial library may present itself in initialization phases, with usage of a 64-bit Niagara Serial Tunnel client. For related details see the *Niagara Drivers Guide*.

Installation of the Win64-based Supervisor is like the Win32-based installation, except that separate executables in the root of the Supervisor product image or CD are used to install (setup_x64.exe instead of setup_x86.exe, respectively).

A platform connection to a Win64-based Supervisor provides the identical collection of views as with a Win32-based host. Also, when opening a station running on a Win64 host, you see the same child services under its **PlatformServices** as with a station running on a Win32 host.

Platform daemon (niagarad)

The platform daemon is an executable that runs independently from Niagara core runtime, is pre-installed on every controller as factory-shipped, and runs whenever the controller boots up. The daemon is Java-based—running in its own Hotspot Java VM (Virtual Machine). An additional (and separate) Hotspot Java VM is used for the running the station process.

You need the platform daemon locally installed and running to host a station on your local PC, such as for a Supervisor. This lets you open the Workbench client platform connection to your local (**My Host**) platform. It also allows remote client platform connections to your PC.

When you install Niagara on your PC, one of the last “Would you like to?” install options is to **Install and Start Platform Daemon**. The default selection is to install.

The Niagara host’s platform daemon monitors a different TCP/IP port for client connections than does a running station.

By default, this TCP port is either:

- 5011 - for a secure (TLS)  **Platform** connection (if available).
- 3011 - for a **Platform** connection that is not secure (unencrypted) .

If necessary, you can change either TCP port monitored to a different (non-default) port during platform configuration.

Installing and starting the platform daemon

Your Workbench PC’s local platform daemon is not necessary for making client platform connections to other hosts, only to provide the ability to run a station locally on your PC. After Niagara installation on your PC, you can install and start the platform daemon at any time, if needed.

Prerequisites: You are using the Windows Start menu.

Step 1 To install and start the daemon, click **Start**, scroll down, expand the Niagara software version and click **Install Platform Daemon** (shortcut for `plat.exe installdaemon`).

The daemon opens a command prompt, runs and indicates successful installation.

Step 2 To view the platform daemon as a service, click the Windows Start menu, click **Services** and scroll down to **Niagara**.

Opening a platform connection to a host

A platform (host) connection differs from a station connection in that when connected to a platform, Workbench communicates (as a client) to the host’s platform daemon, niagarad (daemon), a server process. Unlike

a station connection that uses the Fox/Foxs protocol, a client platform connection ordinarily requires full Workbench, meaning it is unavailable using a standard Web browser or Web Launcher. Using a browser, a Supervisor station can connect to a remote platform through its **ProvisioningService**.

Prerequisites: The platform (PC localhost or controller) has been physically installed and connected. The host is licensed with the license server.

Step 1 Right-click **My Host** in the Nav tree and click **Open Platform**.

The **Connect** window opens with the name of your computer or remote controller as the **Host** name.

It is possible to make this type of secure TLS (Transport Layer Security), encrypted platform connection to any Niagara 4 host, provided it is properly configured.

NOTE: For best security, always use TLS. In Workbench, the default **Open Platform** and **Open Station** (Foxs) commands assume a secure connection. To make an unencrypted connection you must change the connection **Type** first.

If you received an e-mail with the License Key for a host, a pending unbound license already exists on the licensing server.

Step 2 If prompted, enter the license key along with the part number to activate the host's license, and make it immediately available.

Step 3 To accept the host name, click **OK**.

The **Authentication** window opens.

Once the platform is connected, the available platform functions are identical—regardless of connection method.

Step 4 Do one of the following:

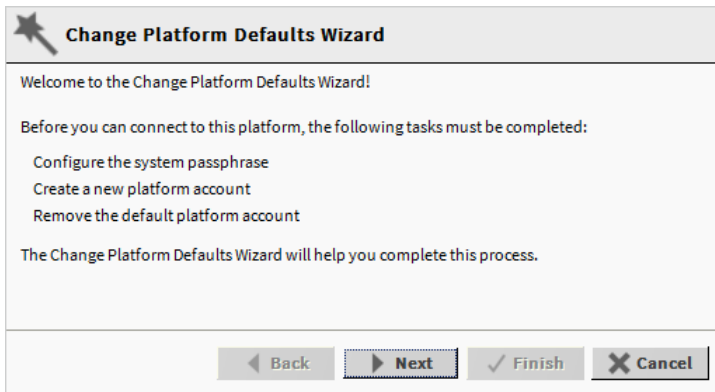
- If connecting to a controller, enter the credentials (user name and password) required by the controller.
- If connecting to your PC localhost, enter the credentials you use to log in to your computer.

As a platform client, you log in to either type of host using host level credentials for authentication. This means a user account and password separate from any station user account. Consider these credentials the highest level access to that specific host.

CAUTION: A new controller ships with default platform credentials that are widely known—and if left unchanged the controller is extremely susceptible to being hacked. Starting with the Niagara 4 startup commissioning process, you must change the default user name and password to something known only to your company and/or customers.

Step 5 Enable **Remember these credentials** and click **OK**

A new controller ships with default platform credentials that are widely known—and if left unchanged leave the controller extremely susceptible to being hacked. If Workbench detects factory default credentials when connecting to a remote platform it launches the **Change Platform Defaults Wizard** (shown here) which forces you to change the factory defaults prior to completing your platform connection.



Change Platform Defaults Wizard

Welcome to the Change Platform Defaults Wizard!

Before you can connect to this platform, the following tasks must be completed:

- Configure the system passphrase
- Create a new platform account
- Remove the default platform account

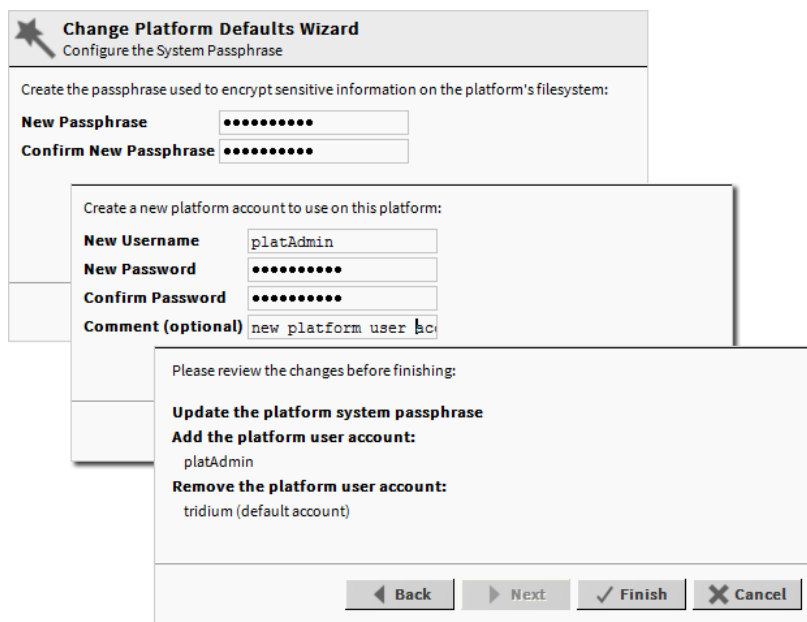
The Change Platform Defaults Wizard will help you complete this process.

◀ Back **▶ Next** ✓ Finish ✕ Cancel

If platform defaults are not detected the platform connection completes. If platform defaults are detected then proceed with the following substeps.

NOTE: If the Workbench FIPS property **Show FIPS Options** is set to `true` certain FIPS options become visible in this window. If selected, the framework enforces FIPS-strength password requirements.

- In the **Change Platform Defaults Wizard**, click **Next** to step through creating a system passphrase, creating a new platform account, and removing the default platform account, as shown below.



Change Platform Defaults Wizard
Configure the System Passphrase

Create the passphrase used to encrypt sensitive information on the platform's filesystem:

New Passphrase [.....]
Confirm New Passphrase [.....]

Create a new platform account to use on this platform:

New Username [platAdmin]
New Password [.....]
Confirm Password [.....]
Comment (optional) [new platform user ac]

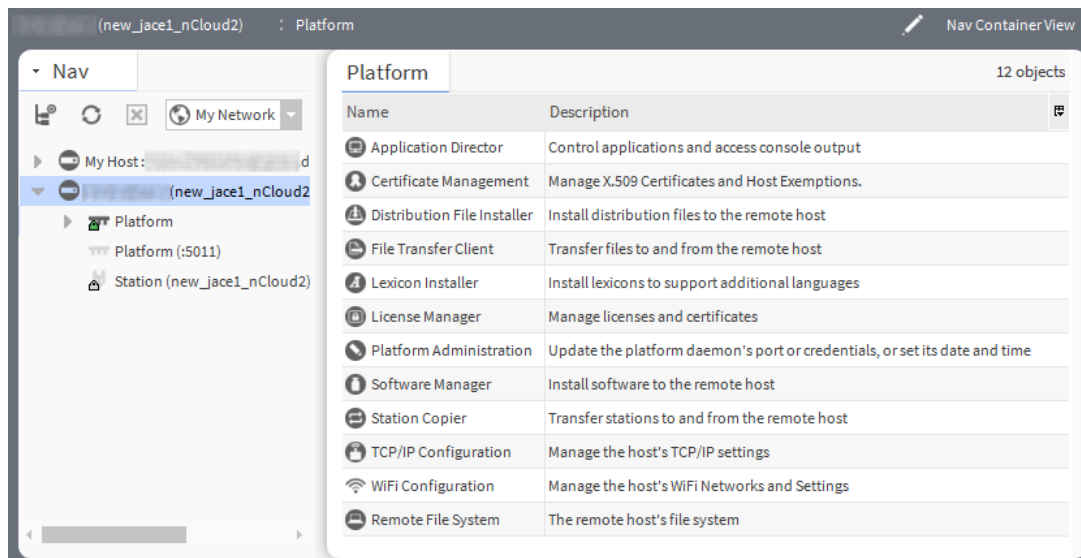
Please review the changes before finishing:

- Update the platform system passphrase**
- Add the platform user account:**
platAdmin
- Remove the platform user account:**
tridium (default account)

◀ Back ▶ Next ✓ Finish ✕ Cancel

- Click **Finish** to complete these changes.

The system completes making the connection between the host and Workbench, and displays the **Nav Container View**.

Figure 2 Platform functions listed in platform's Nav Container View

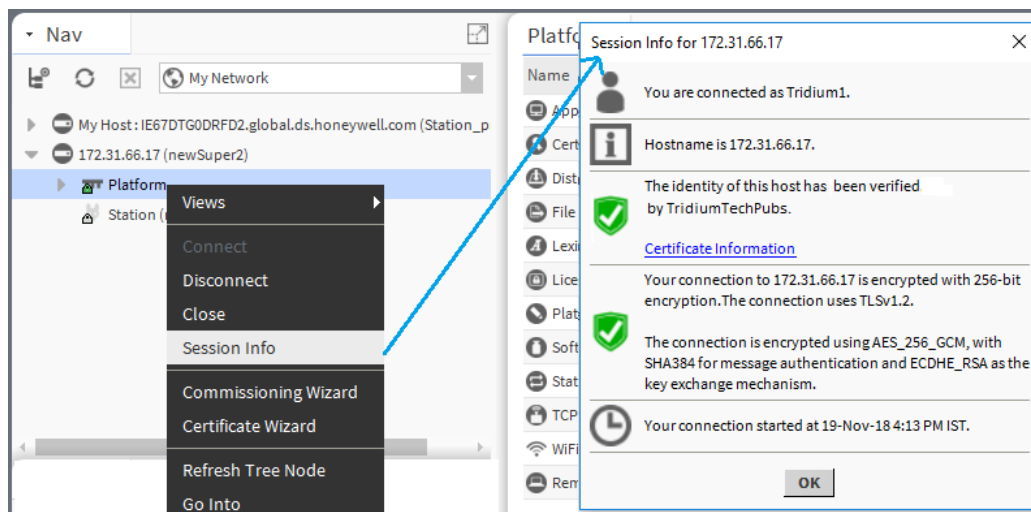
The platform-connection session icon appears in the Nav tree with a small padlock to indicate the connection type, that is: either for secure TLS encryption, or for an unencrypted connection.

Each platform function has its own Workbench view (plugin), which you access by double-clicking the view name. Most of the same platform views exist for both a platform connection to a controller and a Supervisor, with these exceptions:

- If you open a local platform connection at your computer, some platform views appear to be missing, for example the **Distribution File Installer** and **Software Manager** are not in the list. These views have no application when working at your computer—instead, you simply use Windows Explorer.
- A few of the platform views differ depending on platform type.

Step 6 To view information about the current session, right-click **Platform**→**Session Info**.

This same information is available when right-clicking **Station**→**Session Info**



This is an example of this client session information from a secure (TLS) platform connection. The identity of the (server) has been verified by a signed certificate, and all data sent over this connection are encrypted.

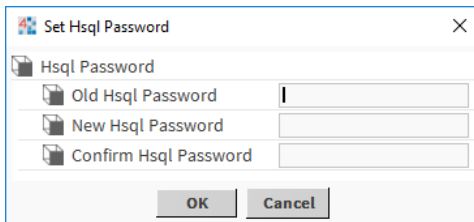
Creating and updating the HSQL database password

Each controller ships from the factory with a default HSQL database password and upgrading an AX station to an N4 station resets the HSQL database password back to the factory default. While the default password initially works, as soon as you install or upgrade a controller and its software, you should change this password to a unique and strong string.

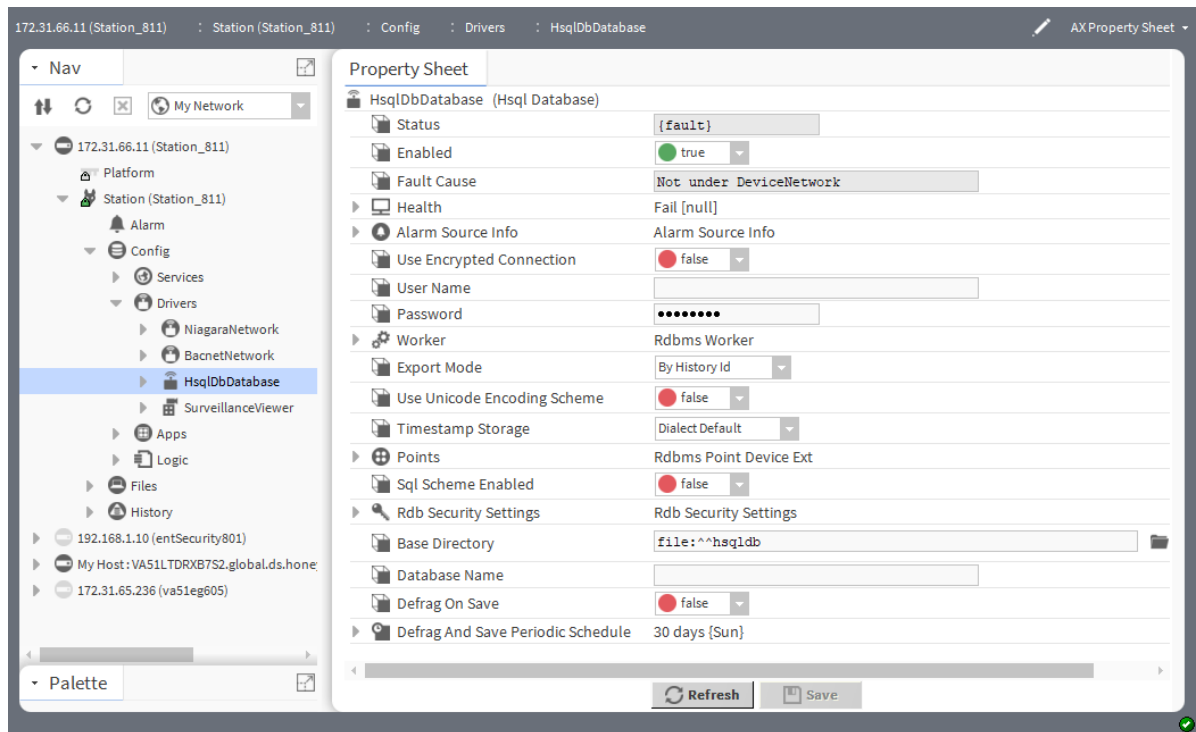
Prerequisites: You have just installed a new controller or upgraded an existing controller from AX to N4. You are using Workbench running on a PC that is connected to the network that services the controller.

The password must contain a minimum of 10 alphanumeric-only characters along with these special characters: @ # ! \$ % & + > <] [) (. The password strength check requires at least: 1 digit, 1 upper case character and 1 lower case character.

- Step 1 Connect to the controller station using Workbench.
- Step 2 Expand **Config**→**Drivers**→**RdbmsNetwork**.
- Step 3 Right-click **HsqlDbDatabase** and click **Actions**→**Set Hsql Password**.
The **Password** window opens.



- Step 4 Enter the Old Hsql Password, configure a strong new password and click **OK**.
If you are changing the password after installing a new controller or after upgrading from AX to N4 the default password is Niagara123.
- Step 5 Make a note of the new password and store it in a safe place.
As a best practice, consider changing this password on a regular basis.
- Step 6 Follow these steps, if you are using an existing N4 station. Open the device's **AX Property Sheet** by right-clicking on **HsqlDbDatabase** followed by clicking **Views**→**AX Property Sheet**.
The **AX Property Sheet** view opens.



Step 7 Enter the **User Name** as SA, **Password** should be blank, and click **Save**.

Step 8 Right-click **HsqlDbDatabase** and click **Actions→Ping**.

The **Health** property updates based on the ping results.

Provisioning—a way to automate platform tasks

Provisioning applies to subordinate controllers, which are represented in the Supervisor station as devices under the **NiagaraNetwork**. Provisioning from a Supervisor station automates some platform tasks.

The *Niagara Provisioning Guide* explains how to configure and provision remote hosts.

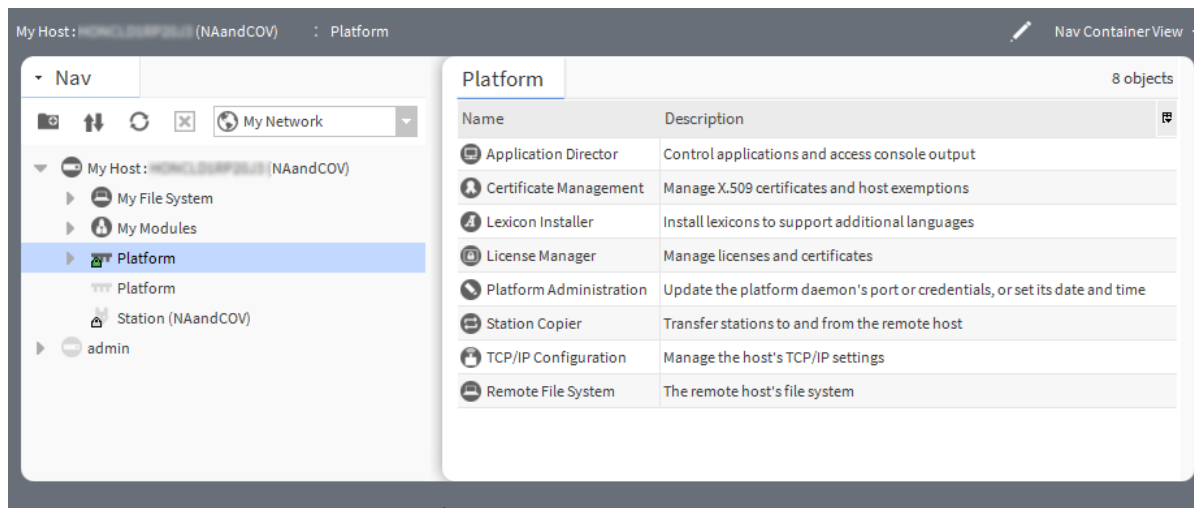
Some provisioning views provided by a Supervisor are nearly identical to platform views described in this document, including the **Software Manager** and **Application Director**, and work in the same fashion. If you are new to Niagara, become familiar with the direct platform views described in this document, before using provisioning from a Supervisor.

Platform Nav Container View

The **Nav Container View** opens every time you double-click the **Platform** node in the Nav tree of a remote controller or PC platform.

This view contains platform objects (functions) that are available before the platform contains a station.

Figure 3 Nav Container View for a PC platform



Some objects are only available in a controller station.

- **Application Director** manages one or more station. You use this object for troubleshooting the station connection.
- **Certificate Management** manages the signed PKI certificates that secure communication.
- **Distribution File Installer** on a remote host restores a backup distribution (.dist) file to the target platform, or installs a clean .dist that restores a controller to a near-factory minimum state.
- **File Transfer Client** copies files between your Workbench PC and a remote platform (in either direction).
- **Lexicon Installer** provides language support for Workbench.
- **License Manager** manages a platform's licenses and certificates.
- **Platform Administration** configures, provides status, and enables the troubleshooting of the platform daemon.
- **Software Manager** manages the software modules (.jars) that support the framework.
- **Station Copier** installs (copies) a station from your Workbench User Home to a remote platform (or if a Supervisor, to the local PC's daemon User Home), backs up (copies) a station to your Workbench User Home, and provides other station BOG file services.
- **TCP/IP Configuration** configures the TCP/IP settings for the platform's network adapter(s).
- **Remote File System** provides read-only access to folders and files on the remote platform, including those under its system home (Sys Home) and daemon User Home.

Exporting the Nav Container View

You can use the contents of the Nav Container View outside of Niagara perhaps for training or other purposes.

Prerequisites: You are working in Workbench and are connected to a local or remote platform.

Step 1 Double-click the **Platform** node in the Nav tree.

The **Nav Container View** opens.

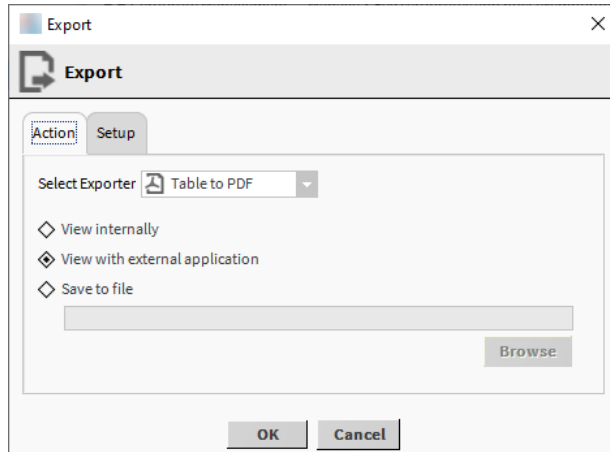
Step 2 Click the list icon () in the upper right corner of the view.

The list opens with two actions: **Reset Column Widths** and **Export**.

Reset Column Widths (the default action) enables an arrow icon for dragging the width of the columns.

Step 3 To export the table, click **Export**.

The Export window opens.



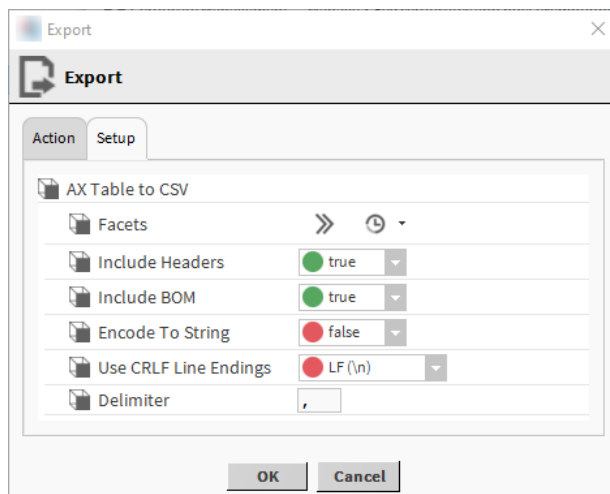
The default action is to a PDF for viewing outside of Niagara.

Step 4 Select how to use the exported data using the radio buttons.

Selecting **Save to file** enables the **Browse** property and button.

Step 5 Select the action from the **Select Exporter** drop-down list

The contents of the **Setup** tab changes based on the exporter you select. The exporter with the most Setup options is AX Table to CSV.



Step 6 If you select AX Table to CSV, confirm or change to setup options and click **OK**.

Clicking **OK** on this window is the same as clicking **OK** on the **Export** window. The framework performs the requested action.

Chapter 2 Platform and station file systems

Topics covered in this chapter

- ◆ Supervisor homes
- ◆ Controller homes
- ◆ Windows user homes
- ◆ Station homes
- ◆ Shared file strategy
- ◆ Running a station from the Workbench User Home

During installation and platform commissioning, the software differentiates between two types of files and stores them in separate locations (homes) based upon the content of the files: configuration and runtime data.

Configuration data, which can be changed by users, include stations, templates, registry, logs, and other data. Runtime data include core software modules, the JRE, and binary executables. Maintaining separate file locations enhances security by denying general access to runtime files (runtime folders are read-only) and allowing each user access to only their personal configuration files.

Multiple home directories serve to separate configuration and runtime data. Each platform has a User Home for configuration data and a System Home (Sys Home) for runtime data. Several other folders under these homes serve specific functions.

The platform's System Home (Sys Home) is sometimes identified by its alias, `niagara_home`. It has a `security` subfolder that contains license files and license certificates. Except when it is time to upgrade, the System Home's runtime files are read-only.

The platform's User Homes contain all configurable data. Referred to by the alias `niagara_user_home`, the separation of these files from the runtime files stored in the System Home folder is new in Niagara 4. The impact of this change is mostly felt when manipulating stations. When Workbench creates a new station, it puts the station in the platform's User Home directory. To start Niagara, the station must be copied from the station User Home to the platform's daemon User Home.

Due to application differences, there are some minor differences between the complete list of files in a user's User Home and the daemon's User Home. For instance, `daemon.properties` only exists in the daemon's User Home and `navTree.xml` only exists in the logged in user's User Home.

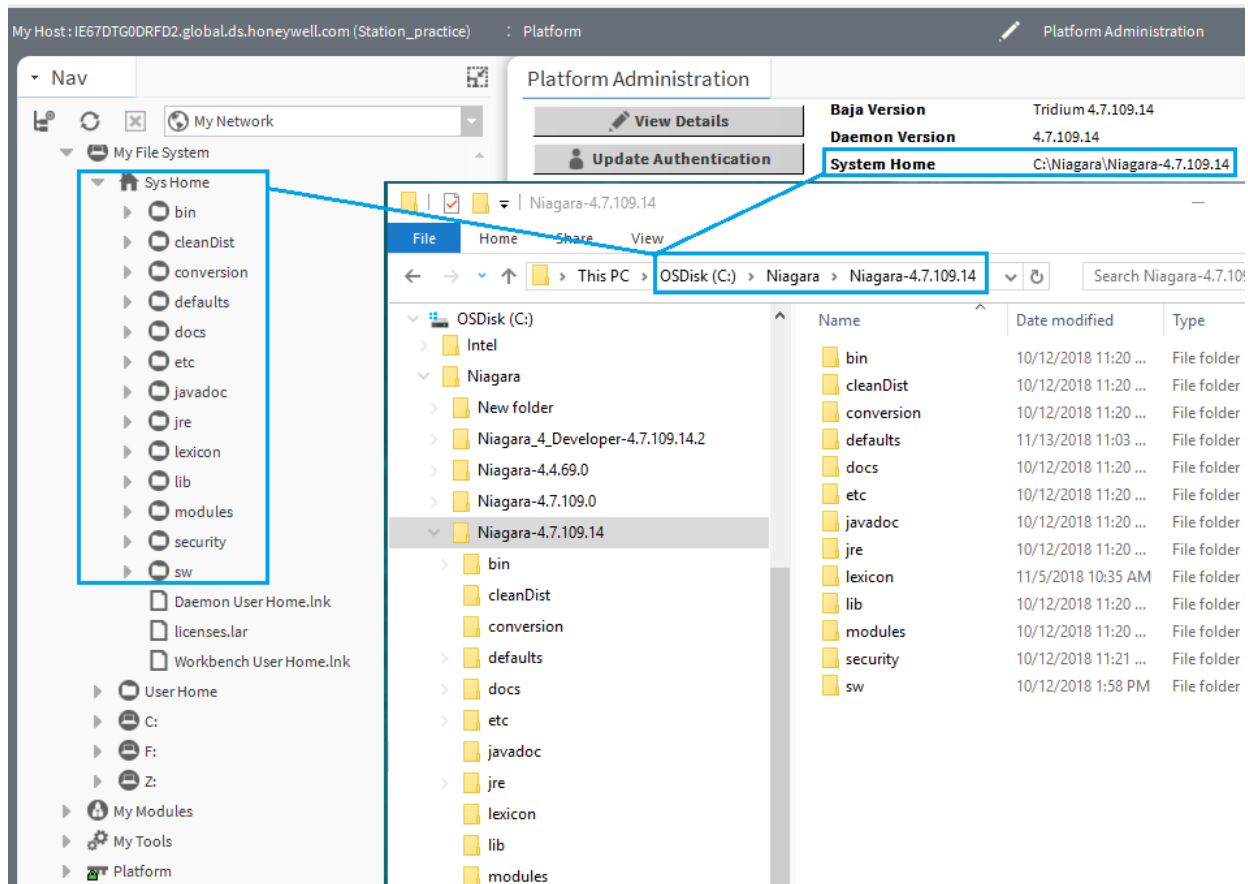
Supervisor homes

The homes on a Supervisor or engineering workstation support supervisory functions.

Supervisor homes include:

- A Sys Home, which contains runtime files, such as core software modules, the JRE, and binary executables.
- A User Home for each user. These are known as Workbench User Homes. They contain station configuration data, including option files, and registries.
- A platform daemon User Home, which contains the Supervisor station and platform configuration data.
- Two station homes: a Protected Station Home and a Station Home. These are located on the computer's drive C.

Figure 4 Example Sys Home (niagara_home) on a Supervisor platform



The example above shows the file system for Niagara 4 Supervisor running on a Windows PC. In the example, the actual location of the System Home folder on this PC is:

C:\niagara\niagara-4.7.109.14.

The following table provides a summary of the Supervisor or engineering workstation homes with shortcut information.

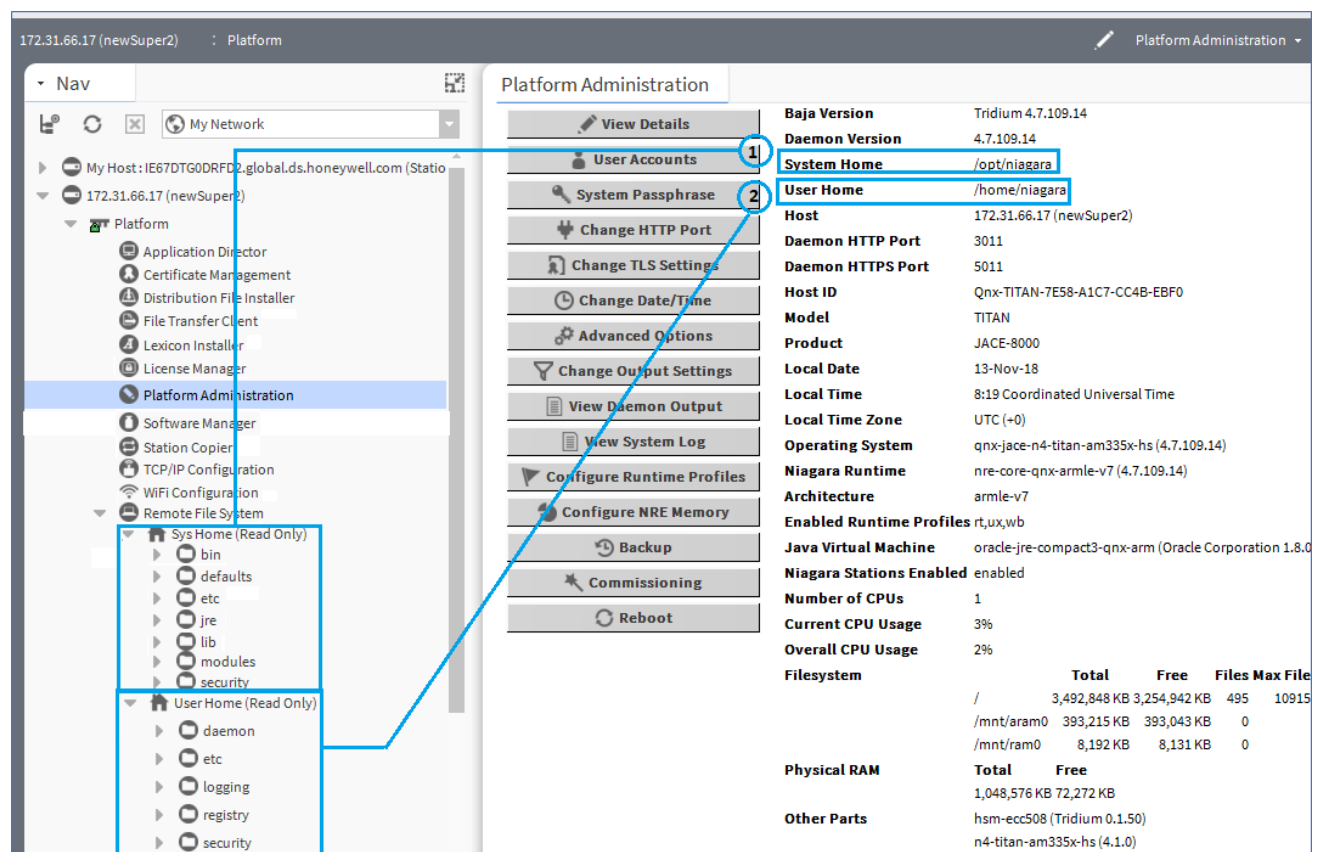
Home in the Workbench Nav tree	Home in the Platform Administration view	Niagara 4 alias	Windows folder location and contents	File ORD shortcut
My Host→My File System→Sys Home	System Home	niagara_home	C:\niagara\niagara-4.x.xx where 4.x.xx is the software version that contains executable and software files.	! (as in Niagara4.x)
My Host→My File System→User Home	N/A	niagara_user_home	C:\Users\username\N4-4.x.xx\tridium where: username is your name to identify you as the user of your computer. 4.x.xx is a software version. The Workbench User Home for each human user contains that user's unique configuration files.	~ (unique to Niagara 4)
shared folder	N/A	station_home	C:\Users\username\N4-4.x\tridium\shared where:	^ (as in Niagara4.x)

Home in the Workbench Nav tree	Home in the Platform Administration view	Niagara 4 alias	Windows folder location and contents	File ORD shortcut
			username is your name to identify you as the user of your computer. 4.x.xx is a software version.	
stations folder	N/A	protected_station_home	C:\ProgramData\N4-4.x\tridium\stations\<stationName>	^^ (unique to Niagara 4)
N/A	User Home	niagara_user_home	C:\ProgramData\Niagara4.x\<brand> Platform daemon user home (non-human user) holds platform daemon configuration files. Requires a local platform connection to view in the Platform Administration view.	~ (unique to Niagara 4)

Controller homes

A controller has one System Home and one User Home. The System Home on a controller appears as **System Home** in the **Platform Administration** view.

Figure 5 Controller System Home (niagara_home) and User Home (niagara_user_home) locations



1. Identifies a controller's System Home (alias: niagara_home) in both the Nav tree and the **Platform Administration** view. In the Nav tree, you find the controller's System Home by expanding **Platform→Remote File System**. The actual location of the System Home folder for a controller is: /opt/niagara.

- Identifies the controller's User Home or daemon User Home (alias: `niagara_user_home`) that contains the installed and running station and other configuration files. The actual folder for the daemon User Home is: `/home/niagara`.

Home in the Platform Administration view	Home in the Platform Administration view	Niagara 4 alias	OFD location and contents	File ORD shortcut
Platform→Remote File System→Sys Home (Read Only)	System Home	niagara_home	<code>/opt/niagara</code> Contains operating system data.	! (as in Niagara4.x)
Platform→Remote File System→User Home (Read Only)	User Home	niagara_user_home	<code>/home/niagara</code> Contains configuration data and the installed and running station.	~ (unique to Niagara 4)

Windows user homes

For security reasons, each person that is a user of a Windows platform, has their own user home. This means that each Supervisor platform has at least two user home locations: Workbench User Home (for people), and a platform daemon User Home (for the daemon server processes).

The Supervisor engineering workstation that is licensed to run a station has a daemon User Home. The daemon is a server process and represents a (non-human) user that manages the Supervisor's running station. The Supervisor's daemon User Home contains daemon-specific configuration information. The actual location of the Supervisor's daemon user home is `C:\ProgramData\Niagara4.x\<brand>`. The platform daemon is installed to this location and started from this location as a Windows service.

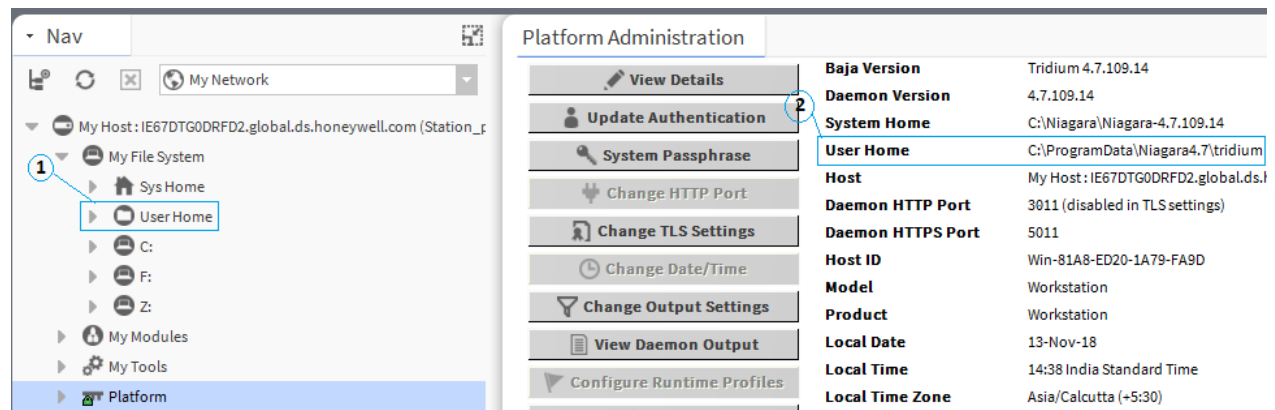
In Niagara, the installation wizard provides the default daemon User Home location, which you can change if you wish. In the step to select the daemon User Home location you have the option to either accept the default location or specify a different location.

CAUTION: The daemon and Workbench User Homes are intended to be installed in distinctly separate locations. This separation of homes is for security reasons but it also prevents certain unintended results. For example, when the two homes are installed in the same location the **Station Copier** becomes unavailable, and you will not be able to make a portable copy of the station.

In addition to this daemon User Home, a Windows host has a separate Workbench User Home for each person (operator, administrator) who logs in with credentials to a Windows-based platform licensed for Workbench, meaning a Supervisor or engineering workstation. Any given PC or workstation has at least one, and may contain multiple Workbench User Homes.

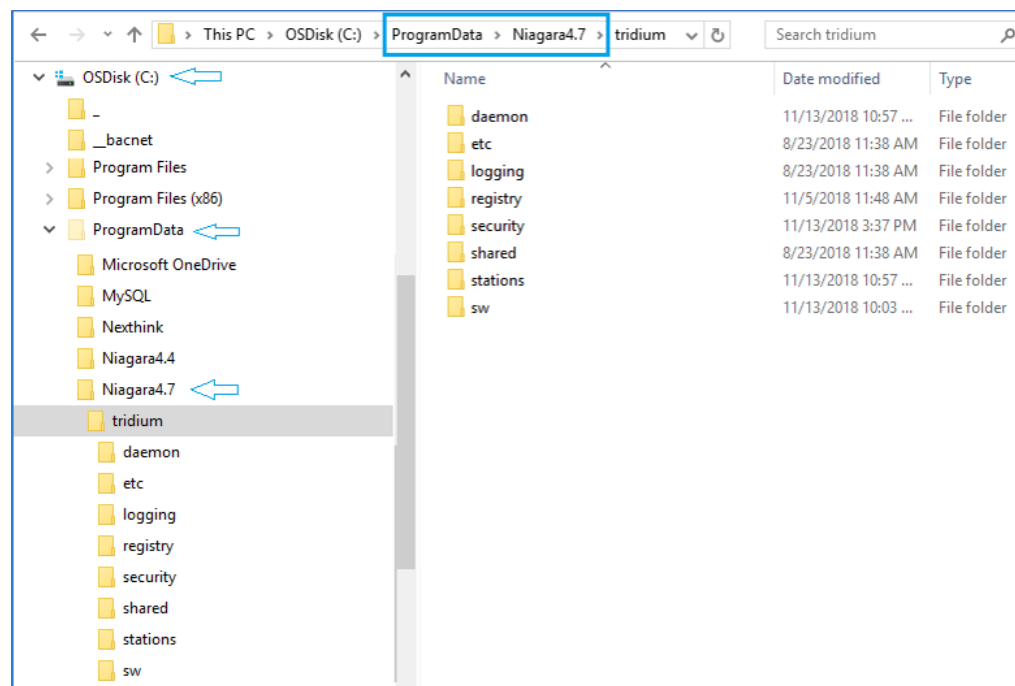
Each person's Workbench User Home is available in the Nav tree as a node under **My Host→My File System** and contains unique configuration information that is not shared. This is where to find any new Workbench station, as well as any remote station backups, templates and other configuration files. The actual location of each person's User Home is in the `Niagara4.x` folder under your Windows `User` account.

To see both types of User Homes on the same view, open a local platform connection to your Supervisor PC, expand **My File System** in the Nav tree, and double-click on the **Platform Administration** view.

Figure 6 Local platform connection to a Supervisor station with Workbench and daemon User Homes

1. Identifies the Workbench User Home.
2. Identifies the daemon User Home.

When you first install Niagara 4 on your PC and start the daemon (by choosing the install option **Install** and **Start Platform Daemon** on installation), the installation program creates this daemon User Home (Niagara4.0 folder). Initially, it contains an empty **stations** sub-folder, until you copy a station to it.

Figure 7 Example of a daemon User Home location in Windows Explorer

You can do this station copy in different ways. In Niagara 4, you can let the **New Station** wizard initiate this copy from its last Finish step. Or, as needed, you can manually open a local platform connection and use the **Station Copier**.

The actual location of each user's home folder is under that user's personal Windows account. Some example Workbench user home locations are:

C:\Users\John\Niagara4.x\<brand>

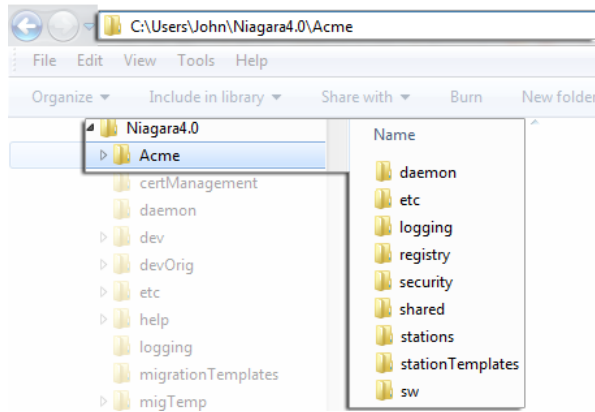
C:\Users\Mike\Niagara4.x\<brand>

where “John” and “Mike” are separate Windows user accounts. The first time a Windows user starts Workbench, the system automatically creates that user’s unique User Home folder.

The person that installs Niagara 4 on a PC acquires the first such User Home. If no other Windows users log on to that PC, this may be the only Workbench User Home on the platform. However, if another person logs on to Windows on that computer and starts Workbench, that user also acquires their own Workbench User Home.

The following figure shows an example Workbench user home location in Windows Explorer.

Figure 8 Example of an automatically-created Workbench User Home in Windows Explorer



Station homes

Niagara 4 uses the Java **Security Manager** to protect against malicious actors who may attempt to access station or platform data and APIs. The **Security Manager** uses signed policy files that specify the permissions to be granted for access to code from various sources. Included are tighter controls about which applications have access to parts of the file system. Two folders under the Workbench User Home serve to protect sensitive data while allowing authorized access to data that can be shared.

- The `stations` sub-folder, otherwise known as the Protected Station Home (alias: `protected_station_home`) contains the running station’s file system, and may be accessed only by core Niagara software modules. Station items that are always in Protected Station Home, that is, items that are not under the `shared` sub-folder include the following folders, as applicable:

- `alarm`
- `history`
- `niagaraDriver_nVirtual`
- `provisioningNiagara`
- `dataRecovery`

All files in the `stations` folder root (`config.bog`, `config.backup.timestamp.bog`, etc.) are always in the Protected Station Home. For this reason, in Niagara 4 it is no longer necessary to blacklist or whitelist station files or folders.

- The `shared` sub-folder, otherwise known as the Station Home (alias: `station_home`), allows all modules to have read, write, and delete permissions.

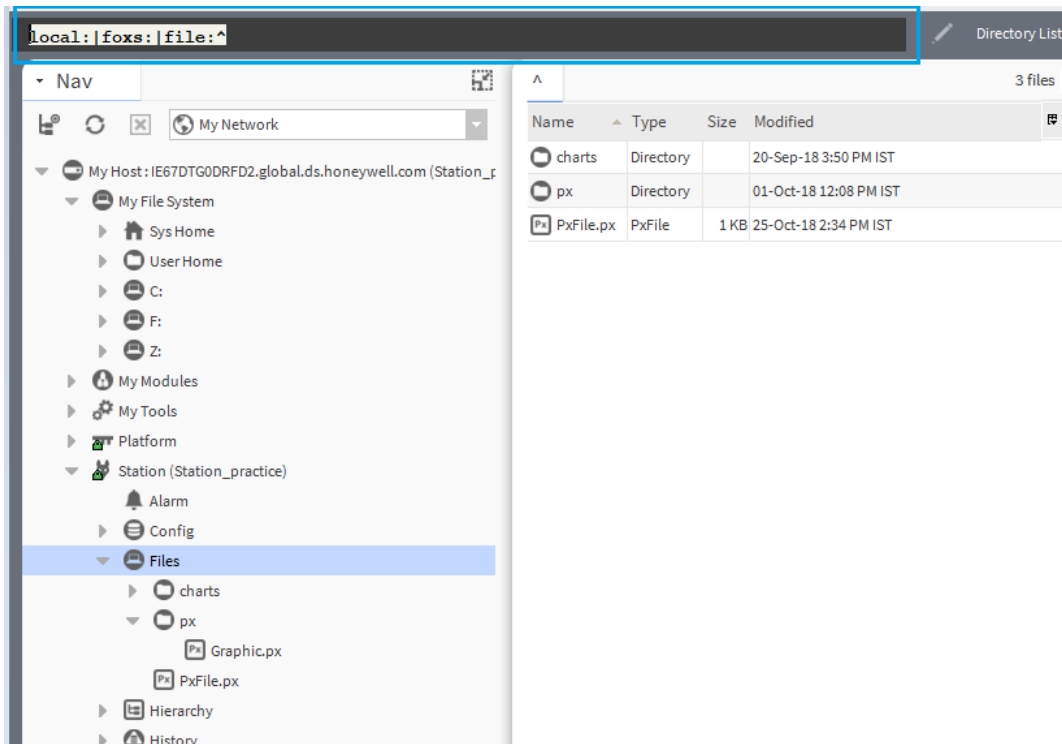
The alias `station_home` retains the same file ORD shortcut (^) as used in NiagaraAX—only in Niagara 4 it points to the station’s `shared` sub-folder.

Figure 9 Example NiagaraAX station file folders compared to Niagara 4 station file folders

AX station folder	N4 station folder
- stations - demo - alarm - dev - frames - history - html - httpd - images - nav - niagaraDriver_nVirtual - public - px - secure - tree - demoToN4	- stations - demo - demoToN4 - alarm - history - niagaraDriver_nVirtual - shared - dev - frames - html - httpd - images - nav - public - px - secure - tree

As shown in the figures above, comparing an AX station file folder structure (left side) to the same station migrated to Niagara 4, a number of folders are under this `shared` sub-folder. Included are folders and files used in graphical (Px) views or navigation, such as `images`, `px`, `nav` and so on. Modules that are prevented from writing to the station root by the **Security Manager** must write to the `shared` sub-folder.

Figure 10 File ORD for the Station Home in Niagara 4 now points to the shared folder



As shown in a station running above, the Station Home (alias: `station_home`) file ORD (^) now points to the contents of the `shared` sub-folder. Other items in protected Station Home are no longer accessible or visible.

Shared file strategy

A sharing strategy makes it possible for multiple users of a single Supervisor or engineering workstation to share station files including backups. This type of sharing is different from the `shared` station sub-folder. This is a folder you create for the purpose of sharing backups and distribution files.

If multiple people log in (differently) to Windows on a Niagara 4 host and use Workbench, each person has their own separate Workbench User Home.

Windows users require permissions to access other users' files; yet it's possible that different users of a system (Supervisor or engineering workstation) may need to share items, such as station backups, station copies, saved template files, and so on. Such items may be in multiple Workbench User Home locations in Niagara 4.

Therefore, in some scenarios you may need to establish a sharing strategy, perhaps re-copying such items to a commonly-accessible folder location on the Niagara 4 Windows PC. For example, you might create a shared folder under the Niagara 4 System Home location (the Workbench User Home is not shareable).

Running a station from the Workbench User Home

Instead of running a station out of the daemon **User Home**, you can run a station directly from your Workbench **User Home** (outside of normal platform daemon control).

You do this using the Niagara 4 console command:

```
station stationName
```

This is not a recommended way to run a production station, but instead more of a developer utility that allows quick access to station debug messages in the console window. If you run the station this way, be mindful of possible port conflicts with any other station that the daemon user may be running locally (in daemon **User Home**), meaning Fox ports, Web ports, and so on.

Chapter 3 Application Director (station management)

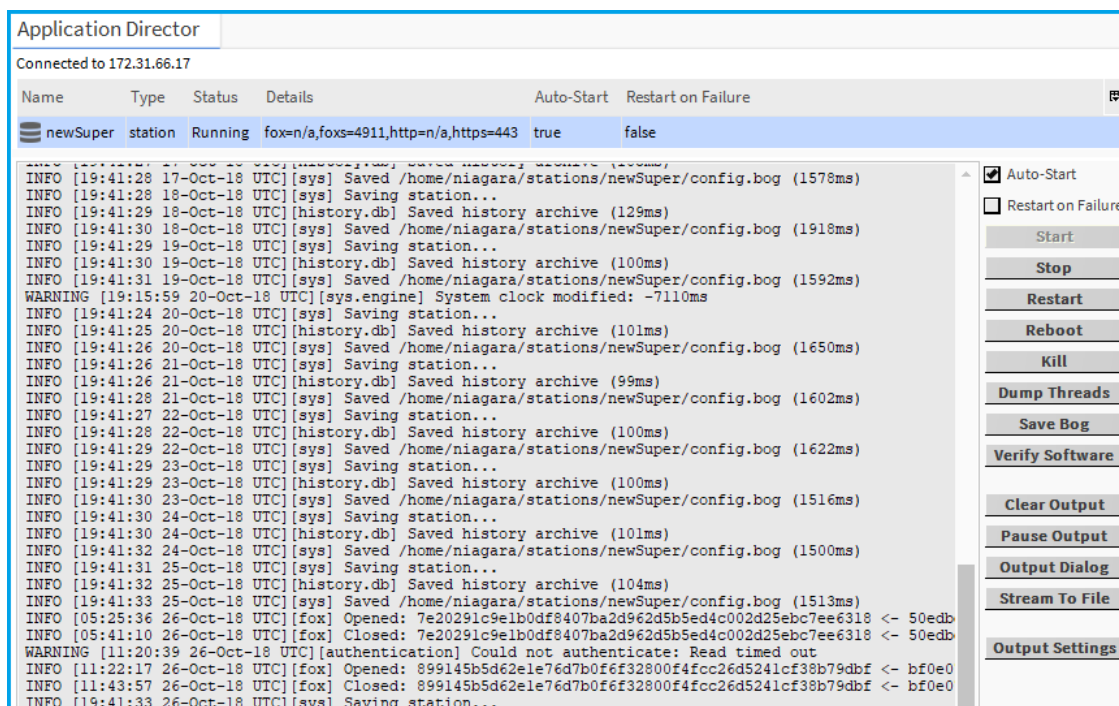
Topics covered in this chapter

- ◆ Starting and stopping a station
- ◆ Standard output messages
- ◆ Configuring station output
- ◆ Configuring station log history
- ◆ Setting up the station spy

The **Application Director** (🔍) is the platform view used to start and stop a station running in any host (whether a remote controller, a local, or a remote Supervisor PC).

The term application refers to an installed station. In addition to starting and stopping, you use the **Application Director** to examine standard station output, for troubleshooting and debug purposes. From it, you define a station's restart settings, plus have access to other station actions.

Figure 11 Application Director view, looking at a station



If, when you open the **Application Director** you cannot see the station row, drag down the horizontal space just above the daemon output.

Every 1.5 seconds, the platform daemon fetches data about the station(s) and updates the **Application Director**.

The options and buttons on the right perform a variety of station functions.

- **Auto-Start**, when enabled, configures the station to start automatically following platform daemon startup.

- **Restart on Failure**, when enabled, causes the platform daemon to restart the station if its process exits with a non-zero return code (for example, the engine watchdog had killed the station because of a deadlock condition). In Niagara 4, controllers can have a station restart without a reboot. Therefore, if this option is enabled, and the station fails (terminates with error), it restarts. If a controller has three automatic restarts within 10 minutes (or however many specified in the station's **PlatformService Failure Reboot Limit** property, the station remains in a failed state, regardless of the setting above.
- **Start** causes the host's platform daemon to immediately start the station, clear the text in the station output, and display messages for the station session.
- **Stop** opens a confirmation window. If you confirm, the host's platform daemon saves the station's configuration to its config.bog file, and potentially saves history data, then shuts the station down.
- **Restart** when pressed, opens a confirmation window. If you confirm, the host's platform daemon shuts the station down gracefully, then restarts it.
- **Reboot** opens a confirmation window. If you confirm, the framework reboots the selected host. This is considered a drastic action.
- **Kill** opens a confirmation window. If you confirm, the host's platform daemon terminates the station process immediately.
- **Dump Threads** causes the host's platform daemon to send from the station a VM thread dump to its station output.
- **Save Bog** causes the host's platform daemon to locally save the station's configuration to its config.bog.
- **Verify Software** causes Workbench to parse the station's config.bog and the host's platform.bog files looking for module references. It then checks to see if those modules, and any other software upon which they depend, are installed.
- **Clear Output** removes the output.
- **Pause Output** freezes the output from updating further (no longer in real time). When you freeze the output, the button changes to **Load Output**.
- **Output Dialog** produces a separate, non-modal output window displaying the same output text as in the **Application Director**'s standard output area. Included are buttons to **Dump Threads**, **Pause Output**, **Clear Output**, and **Close** the window.
- **Stream To File** opens a window for assigning a file name. Once open, the system saves all application output to this file.
- **Output Settings** opens a window for specifying how the platform daemon buffers the output from the station.

Starting and stopping a station

A station must be running before you can connect to it.

Prerequisites: You are connected to a platform that will host or already hosts a station.

- Step 1** Expand the **Platform** node in the Nav tree or double-click **Platform**.
The contents of the platform node opens in the tree or in the main view.
- Step 2** Double-click the **Application Director** (🔍).
The **Application Director** view opens.
- Step 3** To select a station, click the row in the table.
This action highlights the station. When a station is selected, the framework displays its standard output and enables the right-side buttons that apply to the station.
- Step 4** To access the station's shortcut menu, right-click the row in the table.
The shortcut menu (a subset of the application and output actions buttons) opens.

Step 5 To open the Workbench (Fox) connection to a running station, do one of the following:

- To open the connection in the current tab, double-click the station row in the table.
- To open the connection in a new tab, press **Ctrl** and double-click the station row in the table

If the station is not running, a double-click does not change the view.

Step 6 To perform other station functions, use the buttons on the right.

- To configure the station to start automatically any time the host computer is re-booted, enable the **Auto-Start** option. Clear this option to disable auto-start.
- To automatically attempt a restart any time a station fails, enable **Restart on Failure**. Clear this option to disable automatic restart.
- To start the selected station, click **Start**.

The station starts and displays standard output and error messages in the window. Depending on the status of the station selected, the standard output text consists of one of the following:

- If the station is running, the output updates in real time. As more text is written by the station, the system appends it to the bottom of the output area.
- If the station is not running, the output text is from the most recent execution of that station.
- If no station is selected, the output text area is blank.

Step 7 To view all the daemon output, use the scroll bars.

Step 8 To copy the output to the clipboard for further analysis, use the Windows copy shortcut (**Ctrl + C**)

Standard output messages

Station output log messages include errors and warnings that let you know why something is not working, as well as simple informational messages about events as they occur. If needed, you can also change the log level of station output.

The general format of a station output log message is:

```
TYPE [timestamp] [station_process] message_text
```

For example:

```
INFO [17:05:18 16-Feb-15 EST] [fox] FOXS server started on port [4911]
```

Message log types seen in station output include the following, by leading text descriptor:

- **INFO** is typical of most default station output log messages. Usually, each message lets you know some process milestone was started or reached, such as a service or the station itself.
INFO is equivalent to the **MESSAGE** level in the station log output.
- **WARNING** informs you of a potential problem, such as an inability to open a specific port. Typically, warnings do not keep a station from starting.
WARNING is equivalent to the (same) **WARNING** level in the station log output.
- **SEVERE** informs you of a problem that might keep the station from starting. Or, if it can start, an error that prevents some function of the station from operating correctly. Often an exception is produced.
SEVERE is equivalent to the **ERROR** level in the station log output.
- **FINE** is a verbose debug-level message that may be generated upon every process transaction. Typically, this is useful only in advanced debugging mode. You see these for station processes only if you have set the log level at **FINE** or even finer (**FINER**, **FINEST**, **ALL**) level.

Such levels, **FINE**, **FINER**, **FINEST**, are equivalent to the **TRACE** level in the station log output.

In addition to the verbose output messages, occasionally you may see a string of Java exception text in the station's output. This indicates an unforeseen station execution issue, which can range from a licensing problem, a mis-configuration, or some other unexpected problem. If an unexplained exception reoccurs, copy the exception text and report the problem to Systems Engineering.

Station output logs in Niagara 4 use a standard Java logging API (java.util.logging). Any Niagara 4 station has a standard **DebugService** (LoggingService) for making changes.

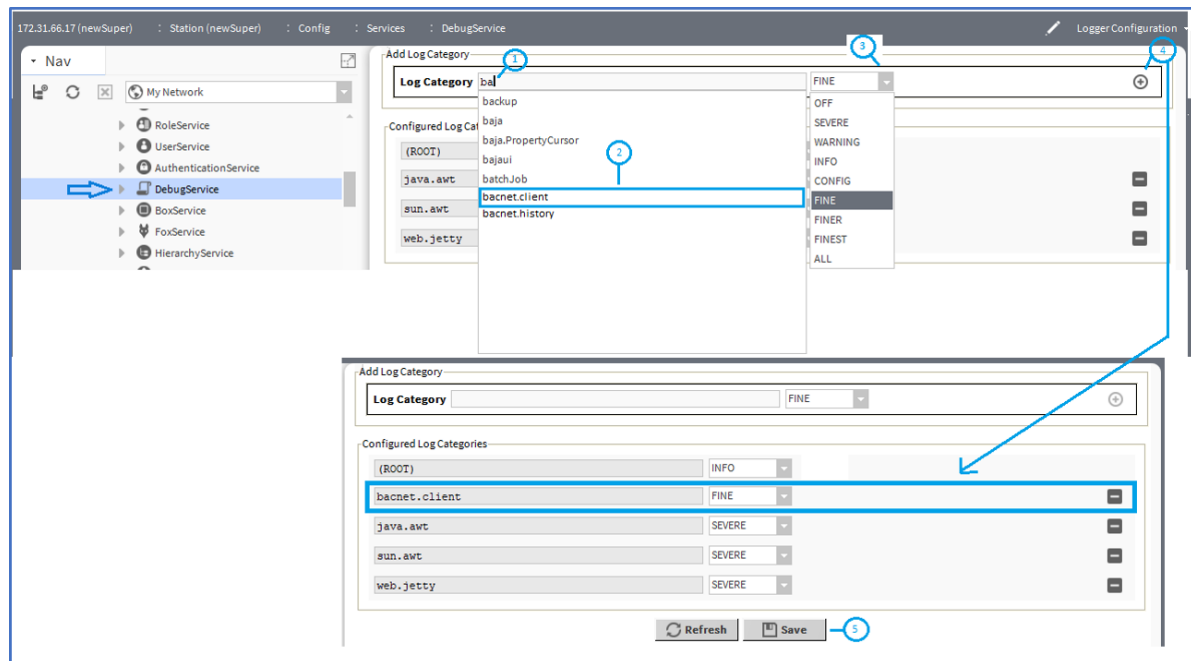
Configuring station output

Using the station's **DebugService** (LoggingService), you can review and change the log level of the station processes of interest. This tunes the station output seen in the **Application Director**.

Prerequisites: You are connected to a platform and its running station. You have admin write permissions on the station's **DebugService**.

Step 1 Expand **Station→Config→Services** and double-click **DebugService**.

The **Logger Configuration** view opens.



Step 2 Begin to type a **Log Category** (module or module.process), such as bacnet (1).

A drop-down list of log categories opens.

Step 3 To select a category, double-click its entry (2) in the list.

The category displays in the **Log Category** property.

Step 4 Select the level (3) from the drop-down list and click the add icon (+) (4) in the upper right of the **Log Category** row.

The **DebugService** adds the category to the **Configured Log Categories** list.

Step 5 Repeat these steps to add other categories and configure their levels.

Step 6 To save these settings to the host's `~/logging/logging.properties` file, click **Save** (5).

Settings become immediately active, affecting station output as seen in the **Application Director**.

CAUTION:

Be aware that persisted log settings are not part of a station's configuration, even though you access them through a station's **DebugService**. Settings apply to any station run on the host, until changed and saved again. Therefore, be sure to return settings back to normal levels and/or delete additions after concluding a debug session. Otherwise, excessive station output could adversely impact station performance.

Workbench has a similar log interface for its console, available in the **Tools** menu (**Tools→Logger Configuration**). This log affects output seen in the console window when you start Workbench with the shortcut **Workbench (Console)**, for (wb.exe). Changes to it are stored in your **User Home** ~/logging/logging.properties file.

Configuring station log history

When looking at a station's output, you are usually troubleshooting. As part of troubleshooting, you should always check the station's log histories, which should contain recently recorded station errors and (if configured) warnings. This information may help when evaluating live output from the station.

Prerequisites: You are connected to a platform and its running station. The station has been configured with the LogHistoryService.

Step 1 Expand **Station→Config→Services** and double-click **LogHistoryService**.

The **LogHistoryService's** AX Property Sheet opens.

Step 2 Expand **History Config** and **Last Record**.

All properties are available for configuration.

Property Sheet

LogHistoryService (Log History Service)

Enabled ☒ true

Minimum Severity Info

History Config Interval: irregular, Record Type: log reco...

Id /Titan_110/LogHistory

Source station: h:56

Time Zone UTC (+0)

Record Type history LogRecord

Capacity Record Count 500 [0 - max] records

Full Policy Roll

Interval irregular

System Tags

Last Record 07-Jul-15 1:01 PM UTC[INFO]Saved /hom...

Timestamp 07-Jul-2015 01:01:18 PM UTC

Log Name sys

Severity 800

Message Saved /home/niagara/stations/Titan_110/c...

Exception

Refresh Save

By default, the log level property **Minimum Severity** is set to **Info**. This is the minimum message type to be sent from station output to the log.

Step 3 You may wish to change this to **Warning**.

The station maintains a buffered history (LogHistory) of some of the messages seen in the station's standard output.

For more information about log history, refer to the *Niagara Histories Guide*.

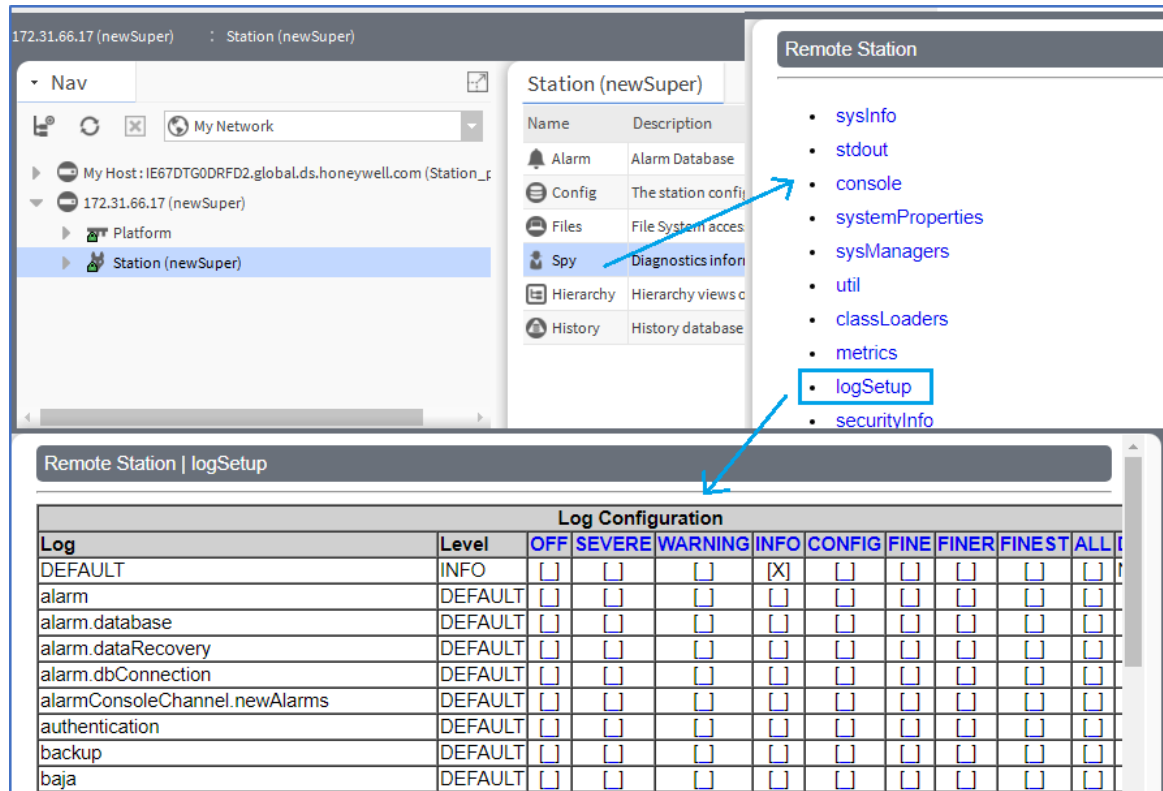
Setting up the station spy

As an alternative to using the station's **DebugService** to tune station log output, you can use the station spy HTML interface for log setup.

Prerequisites: You are running Workbench, are connected to a platform and to its running station.

Step 1 Double-click the running station in the Nav tree.

Its **Station Summary** view opens.



Step 2 Click the **logSetup** link.

A table of the log setup processes and configuration options opens in the main view. Each process shows its current log level, and starts with a (new) **DEFAULT** level. The following are the updated properties:

- Level selection columns are ordered left-to-right in increasing order of message volume.
- The number of severity levels is nine (9) in N4.
- The log levels persist each time you click to set or clear a check box, saved in the host's `~/logging/logging.properties` file. There is no separate **Save To File** in N4.0.
- The level given to the top **DEFAULT** row is global to any row with the far-right **DEFAULT** box set.

The following figure shows the top of the spy **logSetup** page after the **DEFAULT** level has been changed from **INFO** to **WARNING**, and then the weather process set to the non-default level **FINE**.

Remote Station | logSetup | FINE-weather

Changed weather log to level 'FINE'. 1

Log	Level	OFF	SEVERE	WARNING	INFO	CONFIG	FINE	FINER	FINEST	ALL	DEFAULT
DEFAULT	WARNING	☐	☐	☒	☐	☐	☐	☐	☐	☐	Not Applicable
plat.serial	DEFAULT	☐	☐	☐	☐	☐	☐	☐	☐	☐	[X]
sys.engine	DEFAULT	☐	☐	☐	☐	☐	☐	☐	☐	☐	[X]
weather	FINE	☐	☐	☐	☐	☐	☒	☐	☐	☐	[X]
history.db	DEFAULT	☐	☐	☐	☐	☐	☐	☐	☐	☐	[X]
bacnet.schedule	DEFAULT	☐	☐	☐	☐	☐	☐	☐	☐	☐	[X]

Callouts in the figure above show:

1. Last change made, reflected in this status line area (Changed weather log to level 'FINE' .)
2. The **DEFAULT** log level, which in this case has been set to **WARNING**. Note this log level now applies to all rows where one of the 9 non-default levels (OFF to ALL) has not been set.

Increasing station output by assigning various log levels above INFO consumes extra station resources and may exact a performance penalty! After troubleshooting, always return log levels to default values.

You can also easily review, and if necessary, adjust log levels from the station's

Chapter 4 Certificate management (Platform security)

Topics covered in this chapter

- ◆ Certificate Wizard
- ◆ Generating a CA certificate and signed Server certificate using the Certificate Wizard
- ◆ Recommended verifications

The platform's Certificate Management feature is for the management of PKI certificate stores and/or allowed host exceptions, which are used in certificate-based TLS connections between the station/platform and other hosts.

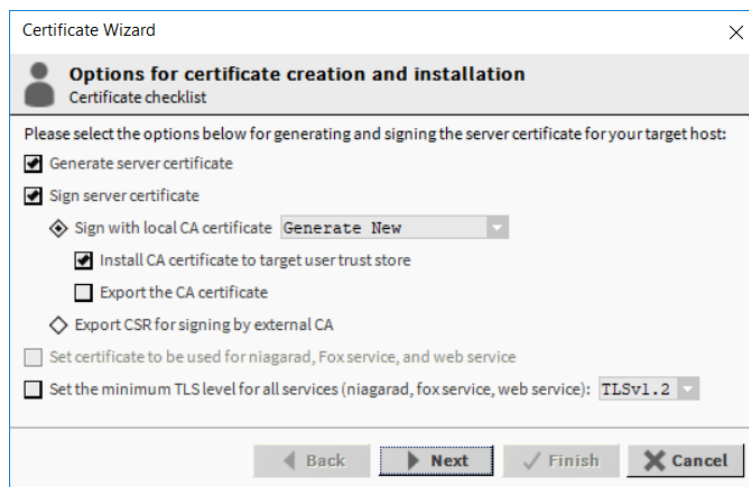
A separate guide, the *Niagara Station Security Guide* documents all aspects of station security.

Certificate Wizard

The latest Niagara version supports a **Certificate Wizard**. This wizard, available as a view on the platform root, provides a complete, continuous workflow that helps you properly set up certificates to harden a platform and station against cyber attack.

The wizard assumes prior experience in various types of certificate setup and a reasonable level of confidence in performing such procedures as are commonly done using the **Certificate Management** tool. That being the case, you will find that the **Certificate Wizard** simplifies the certificate setup process for a station by combining several steps into a continuous workflow. If you are unfamiliar with certificates, work through individual setup procedures using the **Certificate Management** tool as described elsewhere in this guide. The individual procedures will help you gain a better understanding of the steps involved.

Figure 12 Certificate Wizard platform tool with default selections



The screenshot shows the 'Certificate Wizard' window with the title 'Options for certificate creation and installation' and a 'Certificate checklist' icon. The main text reads: 'Please select the options below for generating and signing the server certificate for your target host:'. The options are as follows:

- ☒ Generate server certificate
- ☒ Sign server certificate
 - ◆ Sign with local CA certificate: Generate New (dropdown menu)
 - ☒ Install CA certificate to target user trust store
 - ☐ Export the CA certificate
 - ◆ Export CSR for signing by external CA
- ☐ Set certificate to be used for niagarad, Fox service, and web service
- ☐ Set the minimum TLS level for all services (niagarad, fox service, web service): TLSv1.2 (dropdown menu)

At the bottom, there are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

As an alternative to using the **Certificate Management** tabs to create and install a CA root certificate, the **Certificate Wizard** generates the root CA certificate and exports it with only its public key in preparation to install in a browser.

NOTE: The **Certificate Wizard** is intended to be used for a single platform at a time, not for provisioning multiple platforms.

The **Certificate Wizard** may be configured to perform some or all of the following tasks:

- Generate a new root CA certificate on the local host that can be used to sign all server certificates.
- Generate a new server certificate for a host platform.
- Sign a server certificate with a new or existing root CA certificate.
- Export a server certificate CSR (signing request) for signing by an external certificate authority.
- Install a root CA certificate into the **User Trust Store** of a platform/station selected from the daemon directory.

Once the **Certificate Wizard** generates the files, they must still be installed into all of the appropriate devices. This is accomplished via the **Certificate Management** tool found in the Workbench **Tools** menu.

Generating a CA certificate and signed Server certificate using the Certificate Wizard

This procedure describes how to use the **Certificate Wizard** workflow to complete a series of certificate-related steps for a platform and/or station.

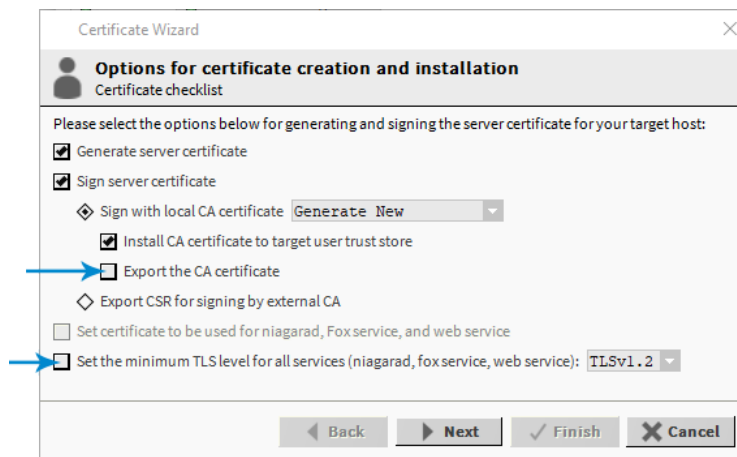
Prerequisites:

You have the required authority to create certificates. You are working in Workbench on a computer that is dedicated to certificate management, is not on the Internet or the company's LAN and is physically secure in a vault or other secure location. You have a thumb drive ready to which to copy the root CA certificate for safe keeping.

Step 1 In Workbench, open a localhost platform connection and in the **Application Director** view click **Stop** to stop any station that is running.

Step 2 In the Nav tree, right-click on the platform and click **Certificate Wizard**.

The **Certificate Wizard** window opens displaying options for certificate creation and installation.



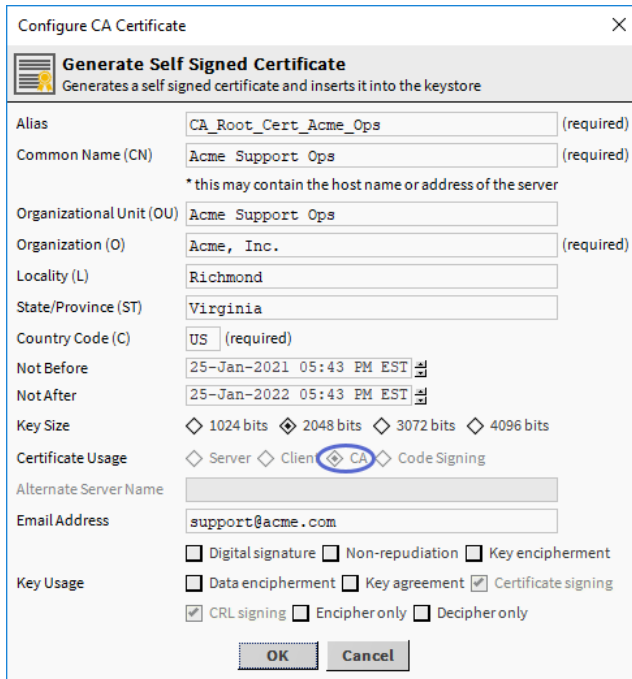
Step 3 In addition to the default selections, configure two optional properties.

- To export the root CA certificate with its private key, click on **Export the CA certificate**. It is a good idea to back up this certificate for archival storage in a secure location.
- To configure the TLS version, **Set minimum TLS level for all services**→**TLSv1.2**.

NOTE:

As of Niagara 4.13, TLSv1.0 and TLSv1.1 are still supported for backwards compatibility, but it is recommended to use TLSv1.2 and higher.

The **Configure CA Certificate** window opens for you to enter the root CA certificate information.



Configure CA Certificate

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

Alias: CA_Root_Cert_Acme_Ops (required)

Common Name (CN): Acme Support Ops (required)
* this may contain the host name or address of the server

Organizational Unit (OU): Acme Support Ops

Organization (O): Acme, Inc. (required)

Locality (L): Richmond

State/Province (ST): Virginia

Country Code (C): US (required)

Not Before: 25-Jan-2021 05:43 PM EST

Not After: 25-Jan-2022 05:43 PM EST

Key Size: 1024 bits 2048 bits 3072 bits 4096 bits

Certificate Usage: Server Client **CA** Code Signing

Alternate Server Name:

Email Address: support@acme.com

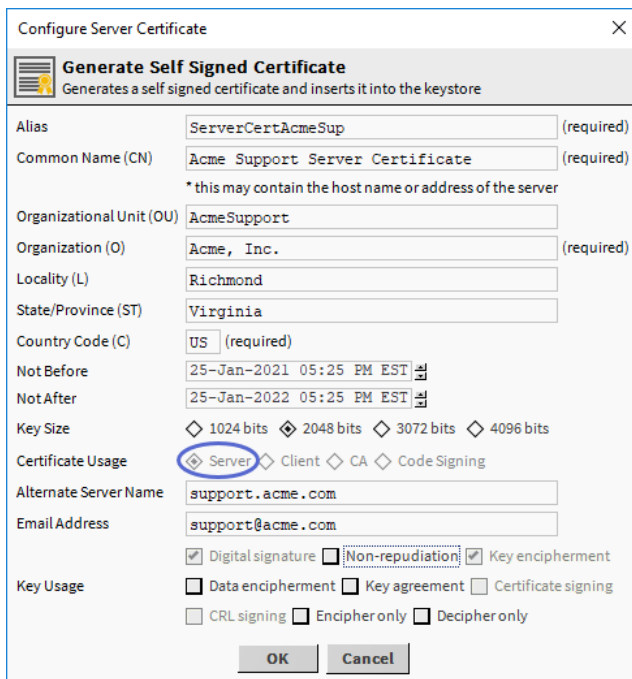
Key Usage: ☐ Digital signature ☐ Non-repudiation ☐ Key encipherment
☐ Data encipherment ☐ Key agreement ☒ Certificate signing
☒ CRL signing ☐ Encipher only ☐ Decipher only

OK Cancel

Step 4 In the **Configure CA Certificate** window, fill in the form, and click **OK**.

Step 5 When prompted for a **Private Key Password**, enter and confirm a strong password (minimum 10 characters, include at least one of each: a number, lowercase, and uppercase character), and click **OK**. For example, Private123%.

The software creates the new root CA certificate in the background. When complete, the wizard opens another **Configure CA Certificate** window. This one is for the server certificate.



Configure Server Certificate

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

Alias: ServerCertAcmeSup (required)

Common Name (CN): Acme Support Server Certificate (required)
* this may contain the host name or address of the server

Organizational Unit (OU): AcmeSupport

Organization (O): Acme, Inc. (required)

Locality (L): Richmond

State/Province (ST): Virginia

Country Code (C): US (required)

Not Before: 25-Jan-2021 05:25 PM EST

Not After: 25-Jan-2022 05:25 PM EST

Key Size: 1024 bits 2048 bits 3072 bits 4096 bits

Certificate Usage: **Server** Client CA Code Signing

Alternate Server Name: support.acme.com

Email Address: support@acme.com

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

OK Cancel

Step 6 In the **Configure Server Certificate** window, fill in the form, and click **OK**.

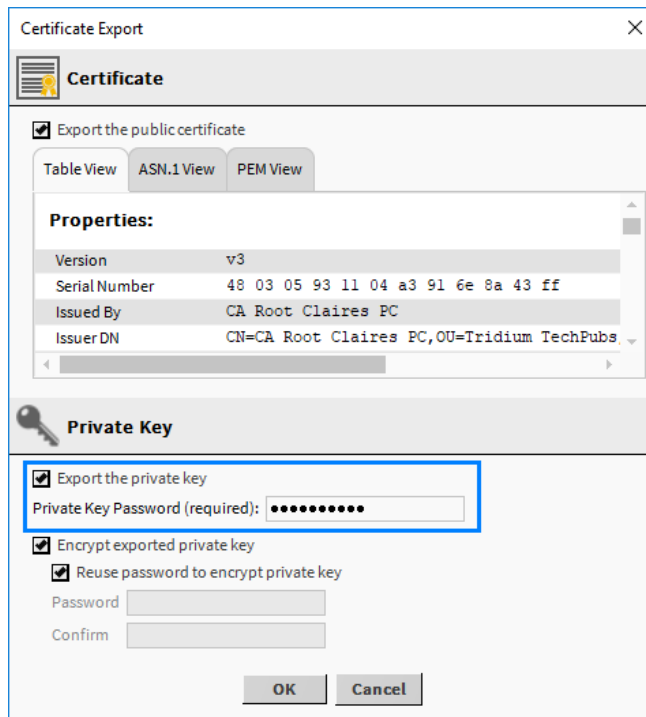
This process generates a server certificate that is ready to be signed. The platform will never be a client, but the station will routinely function as a one, and, since the platform and the station share the same trust store, only one server certificate is required. You will need to run the wizard again when this certificate expires.

Server certificate generation occurs in the background. When complete the wizard opens the **Certificate Signing** window.

NOTE: The server certificate that is about to be signed is already selected. You cannot change the selection. Also, the root CA certificate and the CA password are already identified. There is no need to make other selections or entries.

Step 7 In the **Certificate Signing** window, review the details (similar to the example shown) and click **OK** to continue.

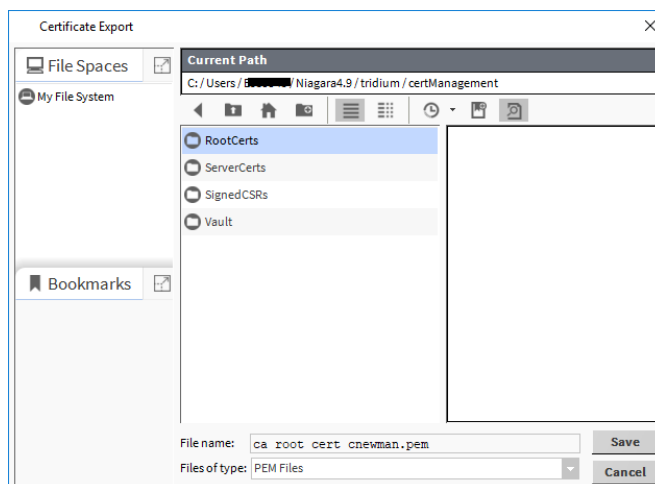
Since we did not choose to export the CSR, the wizard does not display it but proceeds directly to import the signed CSR into the Supervisor station's **User Key Store** and the new root CA certificate into its **User Trust Store**. When complete the wizard opens the **Certificate Export** window.



Step 8 In the **Certificate Export** window, in addition to the default selection, click the optional check box: **Export the private key**, enter the private key password, and click **OK**.

By default, the wizard exports the root CA certificate with only its public key. This is appropriate for distributing the root CA certificate, which must be imported to the **User Trust Store** of every platform/station throughout the enterprise, any PC that hosts an instance of the Workbench, and any browser used to monitor and control the system. You export a root CA certificate with its private key only for the purpose of backing it up to a secure location.

The wizard opens the **Certificate Export** window.



Step 9 Use the folder icon to locate the storage location for the exported root CA certificate in the local-host file system, such as an added subfolder in your `certManagement` folder (as shown) or a thumb drive, and click **Save**.

Within the `certManagement` folder, you can create subfolders for storing certificates and certificate signing requests (CSRs). In the above example, the `RootCerts` folder is a suitable location for

the root CA certificate with its public key, while the Vault folder simulates a secure storage location for the root CA certificate with its private key, which should be kept under lock and key.

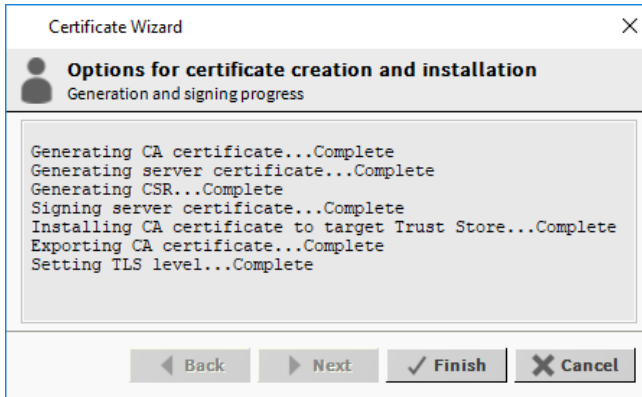
On completion, the wizard acknowledges that the export was successful.

Step 10 To continue, click **OK**.

The **Select Station** window opens.

Step 11 In the **Select Station** window, click the drop-down list of all stations in the Platform Daemon Home, and select the station to Set the TLS levels on, and click **OK**.

The wizard displays a progress summary as you complete the various steps.



Step 12 When prompted with the message, "All operations are complete", click **OK** and **Finish**.

The wizard modifies the station's .bog file in the Platform Daemon Home.

The **Certificate Wizard** successfully generated the new server certificate for the Supervisor PC, and the new root CA certificate for use in signing other server certificates. Those certificates are exported to the certManagement folder in the local file system for subsequent use. Additionally, the wizard set the TLS levels on the selected station to the selected value: TLSv1.2.

Recommended verifications

The platform and the station share the same certificate management stores, while the Workbench application has its own stores. Once you have set up certificates, confirm that the stores contain the certificates you expect.

Verify that the new certificates are installed into Certificate Management.

From the remote controller platform, double-click on **Certificate Management** and verify that the certificates were installed:

- The new server certificate appears in the **User Key Store**.
- The new root CA certificate appears in the **User Trust Store**. This is a copy of the root CA certificate exported with its public key.

Certificate Management for "localhost"

User Key Store System Trust Store User Trust Store Allowed Hosts

You have local certificates:

User Key Store 3 objects

Alias	Subject	Not After	Key Algorithm
supervisor-ncloud	N4:supervisor-nCloud:Tst-992F-13CF-3E1B-2348	Mon Jul 26 13:50:13 EDT 2027	EC
default	Niagara4	Fri Dec 22 10:38:44 EST 2023	RSA
newmansup0servercert	Sup0 Server Cert Claires PC	Fri Sep 24 07:56:53 EDT 2021	RSA

Certificate Management for "localhost"

User Key Store System Trust Store User Trust Store Allowed Hosts

You have user certificates that identify these certificate authorities:

User Trust Store 1 objects

Alias	Subject	Not After	Key Algorithm	Key Size	Valid
ca root cert cnewman	CA Root Claire's PC	Fri Dec 22 10:38:44 EST 2023			true

From the Workbench, click **Tools→Certificate Management**.

- Confirm that the new root CA certificate created by this instance of the Workbench (for use by the wizard) is found in the **User Key Store** of the Workbench, alongside the self-signed **default** server certificate. The Workbench is a client when connecting to platforms and stations, so it is worth pointing out that this is not a Server Certificate. The root CA certificate is available to the Workbench for use in signing server certificates.

Certificate Management for Niagara Workbench

User Key Store System Trust Store User Trust Store Allowed Hosts

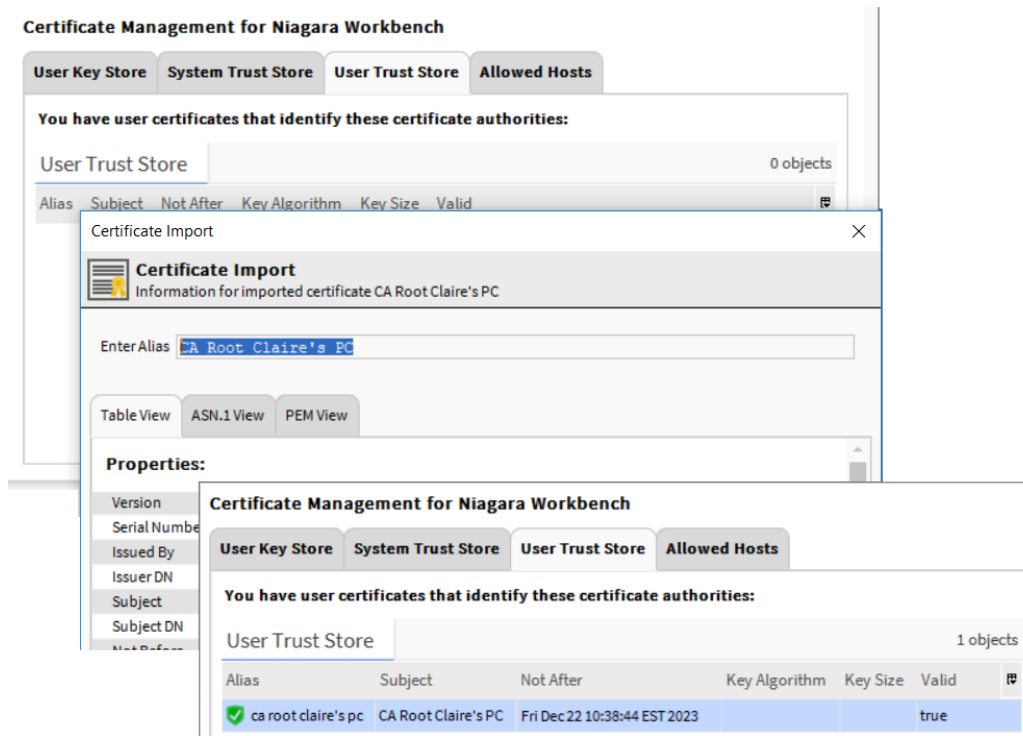
You have local certificates:

User Key Store 2 objects

Alias	Subject	Not After	Key Algorithm	Key Size	Valid
tridium	Niagara4	Sun Sep 06 07:46:33 EDT 2020	RSA	2048	true
ca root cert cnewman	CA Root Claire's PC	Fri Dec 22 10:38:44 EST 2023	RSA	2048	true

- Notice that the root CA certificate was not imported into the **User Trust Store** of the Workbench by the **Certificate Wizard**. You need to import the root CA certificate into this location to ensure that this instance of the Workbench can validate server certificates while handshaking with platforms and stations on the network.

To do this within **Certificate Management**: Click **Import**, locate and select the new root CA certificate in `~certManagement`, and click **OK**.



Verify that the TLS level for each of the following is set to TLSv1.2:

NOTE:

As of Niagara 4.13, TLSv1.0 and TLSv1.1 are still supported for backwards compatibility, but it is recommended to use TLSv1.2 and higher.

- Platform TLS Settings — Using **Platform Administration**, view the Change TLS Settings option and verify the Protocol value.
- Station Web Service — In a station connection open a Property Sheet view on the Web Service and verify the Https Min Protocol property value.
- Station Fox Service — Open a Property Sheet view on the Fox Service and verify the Foxs Min Protocol property value.

Additional recommendations

If you are currently reading the *Niagara Platform Guide*, typically, you need to select the appropriate Server Certificate for use in secure platform and station connections.

- Set the new server certificate to be used for secure platform (niagarad) communications.
- Set the new Server Certificate to be used for secure station communications via Fox and Web Services.

For details, refer to the *Niagara Station Security Guide*.

Chapter 5 Distribution File Installer

Topics covered in this chapter

- ◆ Restoring a backup distribution file
- ◆ Returning a controller to near factory defaults

This platform view is used to install .dist (distribution) files. A Supervisor PC prepared to install distribution files is sometimes referred to as an engineering workstation.

A backup .dist file includes not only the entire station folder, but all other configuration information that may be customized for the platform. This allows for a complete restoration of a station from the one backup file.

Typically, you make a station backup from a Workbench station connection (a station is running, and has the **BackupService**). In the Nav tree, right-click the opened station, and select **Backup Station**.

Less typical is an offline backup from the **Platform Administration** view.

By default, the framework saves station backup .dist files in your Workbench **User Home**, in a ~/backups folder.

Use this view for either of these two tasks:

- To install a clean .dist file. This downgrades a controller to an older Niagara 4 release level, or restores it to a known empty state. Following a clean .dist install, you must commission the controller again, as this wipes out the file system—almost all software, as well as all station files—leaving the controller in an empty near-factory state.

NOTE: Do not use this view to upgrade a controller. Instead, use the **Commissioning Wizard** in the controller. The **Commissioning Wizard** is a right-click option on a platform when opened in Workbench.

Restoring a backup distribution file

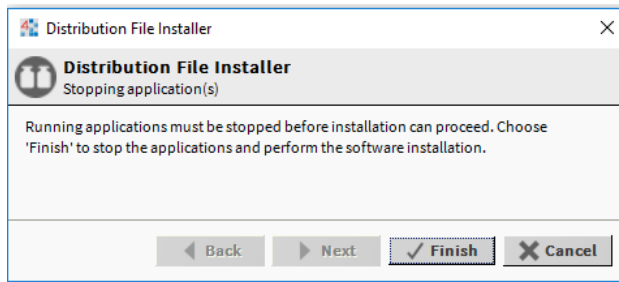
This procedure restores a controller to a factory default state.

Prerequisites:

- A backup .dist file of the station on the target controller exists.
- The software database of your Niagara 4 installation includes matching versions of all software modules used by the station when the station backup was made. Without these modules, restoring the backup .dist will fail.
- Any controlled equipment, which might be adversely affected by the station stopping (and the removal of software) is put in a manually controlled state.
- If the .dist file is protected with a file passphrase, you know this passphrase.

Step 1 Using Workbench, open a platform connection to the remote host.

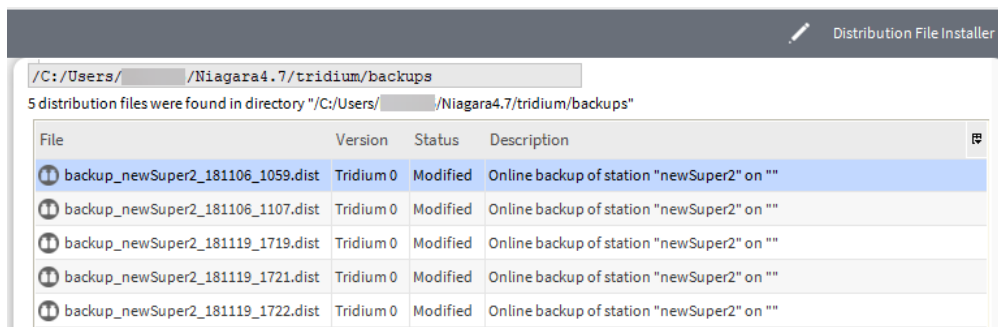
Step 2 If a station is already running on the remote host, use the **Applications Director** to stop the station.



Step 3 To locate the `.dist` file, do one of the following:

- In the **Distribution File Installer** click the **Backups** button (📁). This opens the `! /backups` folder.
- Click the **Choose Directory** button to point to another backup `.dist` file location.

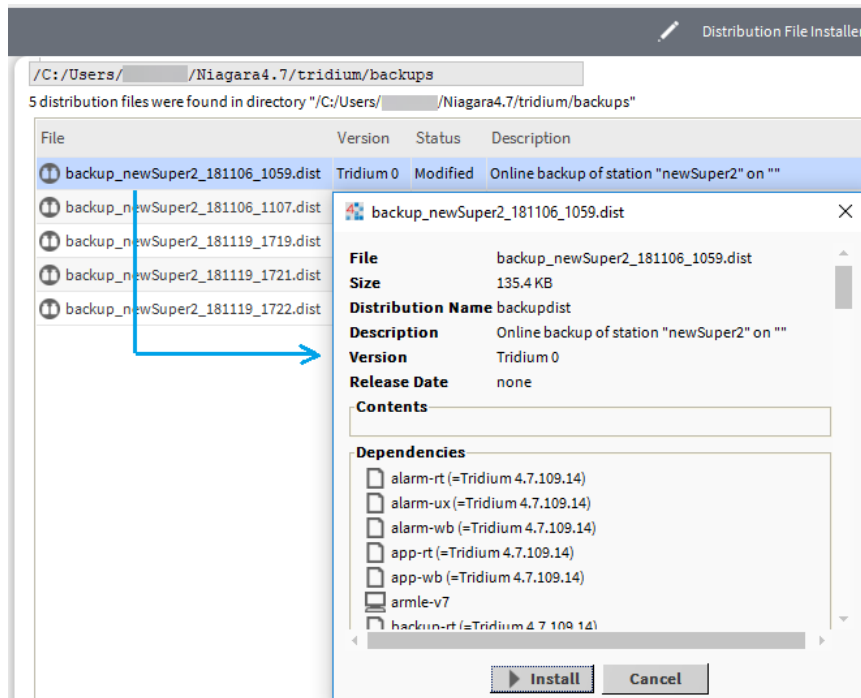
The Installer parses through the distribution files, and makes selectable only those files that are compatible with the opened platform. When done parsing, available backup `.dist`s open in a list.



Distribution files that are inappropriate, for example those that are for a different target platform or have unmet dependencies, are dimmed and the **Install** button does not become active if you select one of them.

Step 4 For details on any `.dist` file, double-click it.

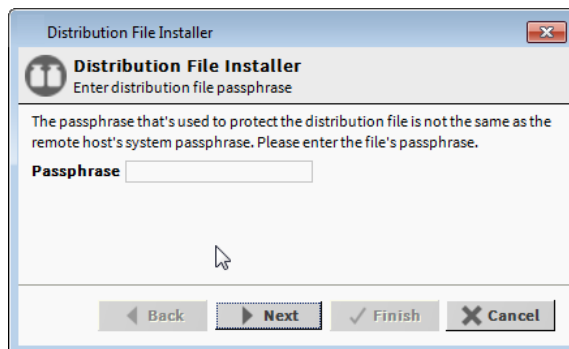
The system opens a popup that includes a list of dependencies.



The details window provides information about the selected distribution file, including all contents and any dependencies.

Step 5 To restore any selected backup, click **Install**.

When you click **Install**, the system attempts to validate the file's passphrase. If the file passphrase and system passphrase are the same, the process continues without prompting for a file passphrase. If the file passphrase and system passphrase are different, the distribution file installer prompts you for the passphrase.

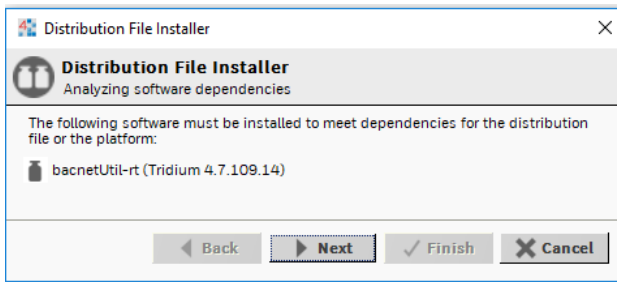


NOTE: If prompted for the `.dist` file passphrase and you do not know it, you cannot install the file.

Step 6 If you are prompted for the **Passphrase**, enter it and click **Next**.

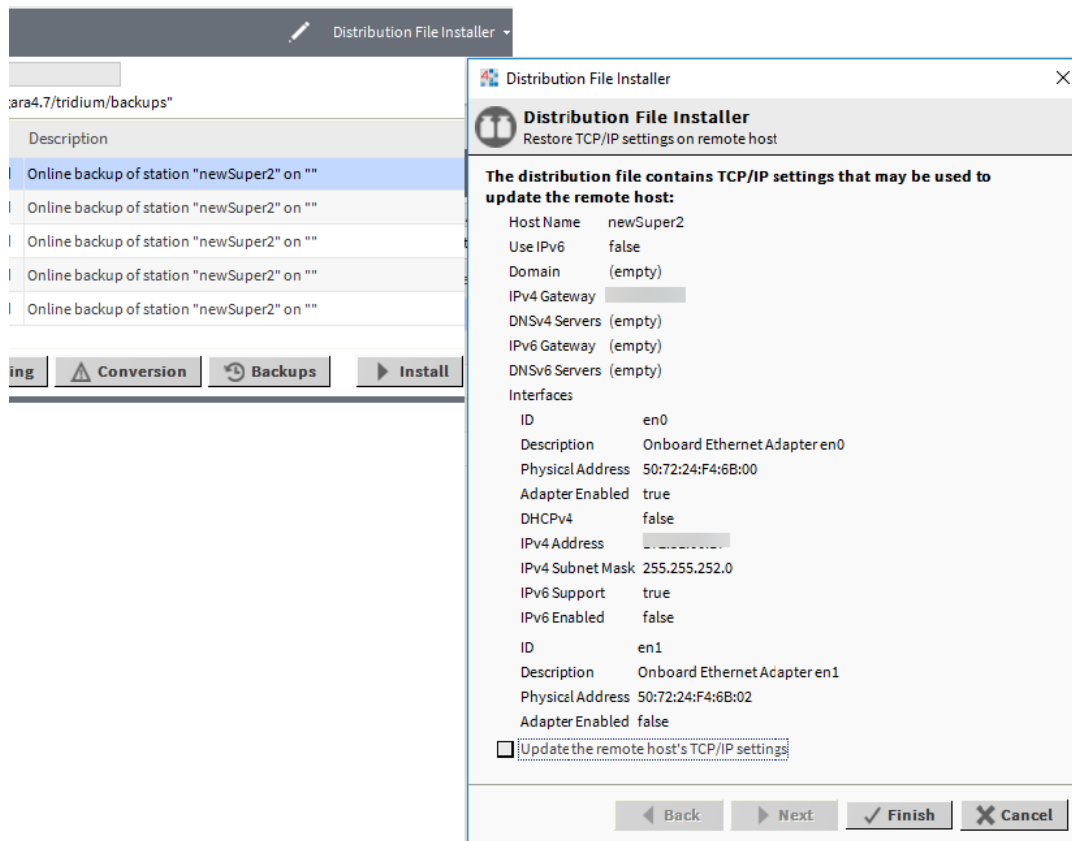
If the host is already running a station, a window opens telling you that the station must be stopped.

If the station backup `.dist` file contains software modules that are different from (or in addition to) those already installed in the remote host, another window opens:



Step 7 To continue, click **Next**.

Another window asks if you wish to restore the TCP/IP settings stored in the `.dist` file (as displayed) into the remote host.



The TCP/IP settings contained in the `.dist` file are listed, and by default, the check box **Update the remote host's TCP/IP settings** is cleared.

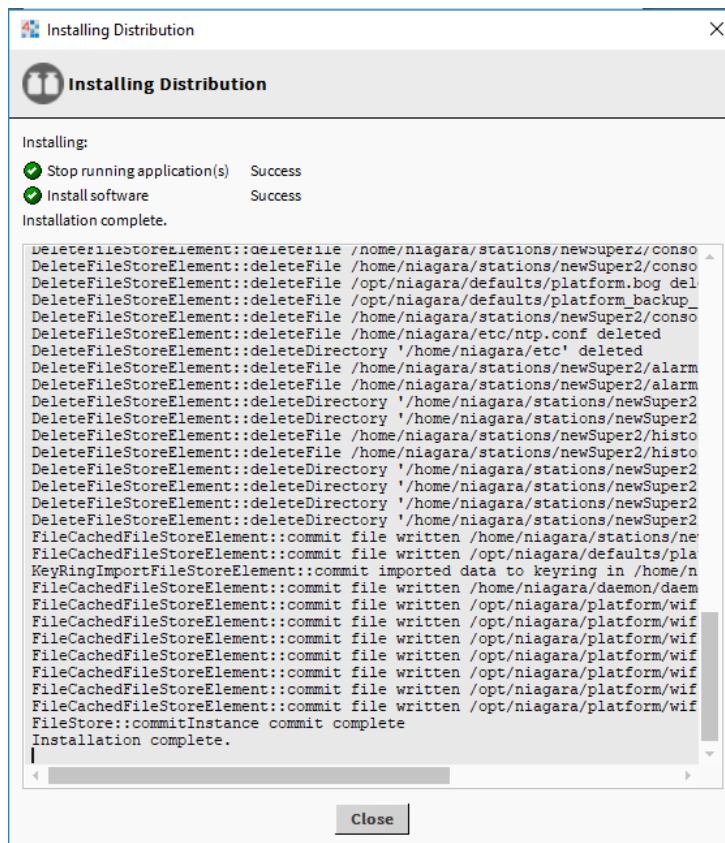
Step 8 Do one of the following:

- To use the same `.dist` file on differently addressed hosts, leave this check box cleared.
- To use the TCP/IP settings stored in the `.dist` file, enable the **Update the remote host's TCP/IP settings** check box.

Depending on your choice, after the `.dist` file installs and the host reboots, it retains its current TCP/IP settings or uses the TCP/IP settings stored in the `.dist` file.

Step 9 To begin the installation, click **Finish**.

The `.dist` installation process opens a window that tracks its progress.



The installer automatically stops the station, then continues with the distribution file install process, which overwrites the station. After the distribution file (and modules, if selected) are installed on the platform, the controller reboots, and the progress window indicates complete.

Step 10 To continue, click **Close** and open a new platform connection, perhaps to view output in the **Application Director**.

Returning a controller to near factory defaults

At times it may be necessary to restore an controller to a known good empty state, either to recommission it with the current release build, or before recommissioning it with an earlier build. To do this, you can install a clean .dist (distribution) file.

Prerequisites: The platform is running Niagara 4. You have backed up any station files as well as any other files needed later, for example digital certificate keys. Always export certificate keys for any TLS-configured unit and store the exported keys in a safe place, such that if the controller needed to be replaced (a hardware swap-out), you could re-import the keys.

Wiping a controller clean is, typically, unnecessary if you are upgrading an operational controller to a later software build. Using the **Commissioning Wizard** should be all that is necessary. However, if you are downgrading a controller to an earlier build, you should install a clean .dist file first, to avoid compatibility problems. This applies especially to JACE controllers, as binaries for the (QNX) OS are included in .dist files.

Installing a clean .dist wipes the entire file system and installs an appropriate version of the Niagara platform daemon, resetting the unit to a near factory state. If the controller came with an appliance installed, installing a clean .dist also removes the appliance. Only the following settings are preserved:

- TCP/IP settings
- license files
- brand.properties

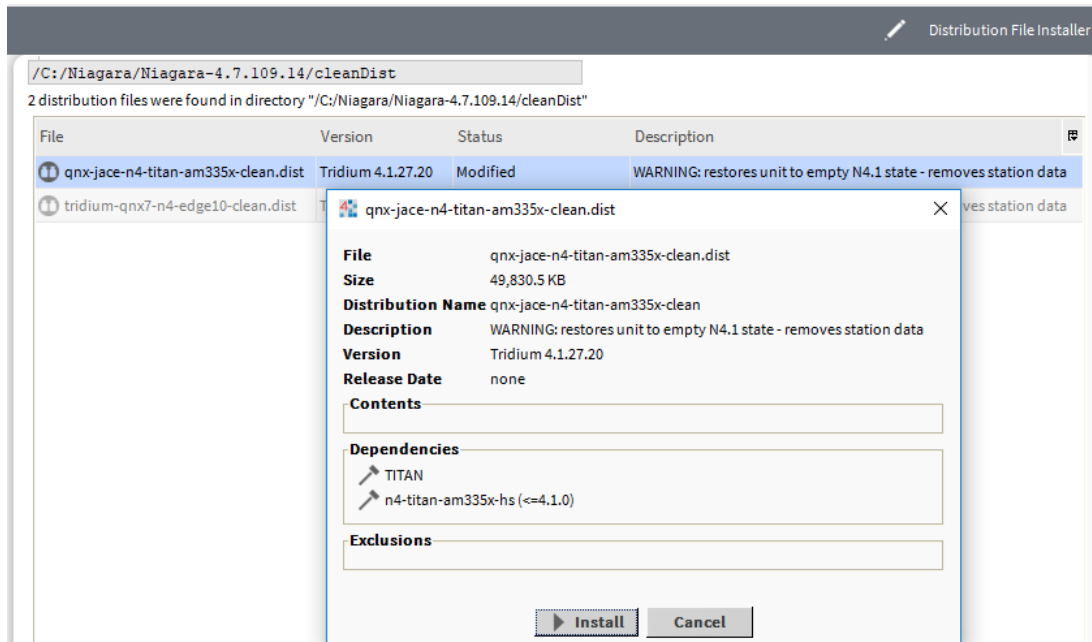
- most secure communication (TLS) configuration

All other data are deleted from the file system, including station bog files, Px files, modules, etc. The unit's TLS private key information is also deleted. In addition, installing a clean .dist deletes all configured platform users, restoring the factory-default platform credentials and port (3011).

Step 1 Using Workbench, open a platform connection to the controller.

Step 2 To access the !cleanDist directory, open the **Distribution File Installer** and click the **Cleaning** button.

Each clean dist file has the suffix -clean in its name. Clean distribution files are located in your Sys Home !cleanDist folder—apart from other dist files under your software database.



Clean dist files appear listed with a WARNING in the **Description**. You can select only the appropriate file for the currently opened platform.

Step 3 Select the appropriate clean dist file for the platform and click **Install**.

Removing a file system takes a few minutes, then the controller automatically reboots. Wait for the reboot to complete.

NOTE: After reboot from a clean dist install, the controller requires port (3011).

Step 4 Do one of the following:

- To re-install the software versions to the controller, open a version of Workbench that uses the same software version that you want on the controller, and use the platform **Commissioning Wizard** to install the desired software build.
- If you have a backup dist file for the controller that was made when it had the desired prior N4 software version, use the **Distribution File Installer** to install it.

Chapter 6 Lexicon Installer

Topics covered in this chapter

- ◆ Changing the language (Lexicon Installer)

The Lexicon Installer lets you install file-based sets of the text used by the software from your Workbench PC to a remote platform.

CAUTION: In Niagara 4, usage of this view and file-based lexicon sets are typically not recommended. Instead, make one or more modules of customized lexicons using the **Lexicon Module Builder**, and install them in a remote platform using the **Software Manager**. Otherwise, issues may occur when accessing a host station using a browser.

Lexicons can also be installed as modules (.jar files), in which case you use the platform **Software Manager** (instead) for installation in remote platforms. In fact, standard lexicons are distributed as modules, using a module file name convention of:

```
niagaraLexiconLc-rt.jar
```

where **Lc** is the two-character language code, such as **Fr** for French and **Es** for Spanish. Workbench provides a **Lexicon Tool** with a special **Lexicon Module Maker** view that you can use to modify or make new lexicon modules, from edited text-based lexicon files. For complete details, refer to the *Niagara Lexicon Guide*.

Lexicons typically have one of two uses, depending on job location:

- International locations provide non-English language support
- Domestic (U.S.) locations where you have modified the English (en) lexicon to change the wording used in default labels.

Beforehand, use the **Lexicon Editor** view of the **Lexicon Tool** in Workbench to review and edit entries (or keys) in the individual lexicon files with localized values needed for language support.

Changing the language (Lexicon Installer)

This step installs one or more text-based lexicon file sets in the host controller. Lexicons provide support for non-English languages. A locale code identifies each lexicon. For example, "fr" identifies the French lexicon and "de" the German lexicon. In some domestic (U.S.) installations, an English lexicon ("en") is added and configured to globally customize items, such as the property descriptions in Workbench.

Prerequisites: The lexicon file(s) to install are in the lexicons folder under your Niagara 4 **Sys Home** (niagara_home)

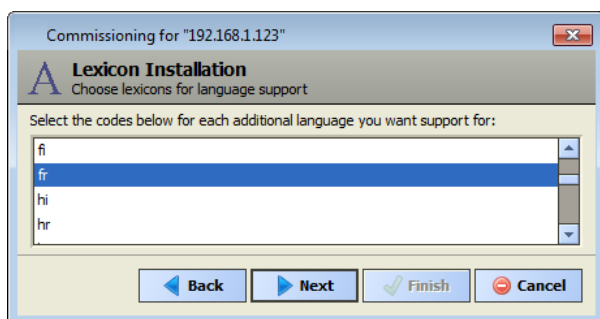
NOTE: The recommendation for Niagara 4 is to skip this step. Instead, make one or more modules of customized lexicons and install them in the next (Select modules) step. Otherwise, issues may occur when using a browser to access of the hosted station. For complete details on working with lexicons and the **Lexicon Module Builder**, refer to the *Niagara Lexicon Guide*.

Step 1 Expand or double-click **Platform** and double-click the **Lexicon Installer**.

Any existing file-based lexicon sets (already installed in that platform) are listed in the view pane.

Step 2 Click a language code to select it, as shown.

To install more than one lexicon, hold down the **Ctrl** key while you click.



Step 3 Click the **Next** button and follow the wizard, and click **OK**.

The selected lexicon directory or directories are installed in the remote platform. When all files are transferred, an **Installation Complete** window opens.

Chapter 7 License Manager

Topics covered in this chapter

- ◆ About license files
- ◆ Submitting a license request
- ◆ Importing a host license from the license server
- ◆ Installing a controller license
- ◆ Exporting a license
- ◆ Remote license management
- ◆ Subscription licensing
- ◆ SMA expiration reminder

License management involves two separate databases: an online license server and a local database that resides on a Supervisor computer. The **License Manager** installs (imports) licenses and certificates to a remote platform, sourced either from your Workbench PC or the Niagara licensing server. You can also view the contents of licenses and certificates, and if desired, delete them from a remote platform.

The Workbench management of licenses uses a structured local license database with a license archive file format. In addition, the **Workbench License Manager** is available, which does not require a platform (or station) connection to use. Instead, it uses a platform tool to manage licenses.

- The Supervisor platform's local license database contains all the licenses assigned to the remote controller platforms that are part of the Supervisor's **NiagaraNetwork**. This structured collection of subfolders and files is under the Supervisor platform's Sys Home `! /security/licenses/db` directory. Each subdirectory has a unique host ID name that matches a remote host platform.

The local license database design makes it easier to store licenses for multiple host platforms without inadvertently overwriting one license file with another. This saves you from having to make special license folders (subdirectories), and/or rename license files uniquely. The related license archive storage file format (.lar) facilitates the exchange of licenses among different PCs, and is used when updating and synchronizing licenses to the online licensing server, as well as with provisioning features for **NiagaraNetworks**.

Workbench creates and manages the local license database automatically and updates it when you perform license operations from platform connections, **PlatformServices** and when using platform tools. The same directory and file structure is visible using your PC's Windows Explorer.

- The online licensing server is a database of licenses and certificates. As the final license authority, it contains the most current version of each host platform's license. This includes licenses for controllers, Supervisors, and workstation-only applications. The platform's **License Manager** provides the interface between the Supervisor platform and the licensing server.

Multiple views (plugins) access the licensing server and local database including: the browser's **License Request** form and the platform's **License Manager**. Other views use the licensing server to confirm that a feature is licensed. Examples include the **Local License Database** tool and the **Network License Summary** view of the Licenses slot on the **NiagaraNetwork's ProvisioningNwExt**.

The procedures in this guide distinguish between the server and database depending on the requirements of the task.

About license files

The Niagara license file is a structured XML file that has a .license file extension. It enables a set of vendor specific features. Each license file is valid for one specific host platform (controller, PC), matched by that host's unique host ID. License files are digitally signed by the vendor to prevent tampering.

These items are common to all license files:

Item	Syntax	Description
license	<code><license></code> <code></license></code>	These are the first and last lines in the license file. All contents (lines) in between are <code><feature></code> elements, plus one signature element. <code><license vendor="Tridium" expiration="2025-03-31" hostId="ATLAS-SD-F84C-2E6D-D8"</code>
vendor	<code>vendor="Tridium"</code>	This value is always Tridium.
expiration	<code>expiration="never"</code>	Defines the expiration date of the license file. After the expiration date Workbench fails to start due to a license expired error. Typically, engineering copies of Workbench have expiration dates that expire on an annual basis. License files for actual projects are issued with non-expiring licenses, where this attribute value is:
version	<code>version="4.x"</code>	Identifies the Niagara software version where 4.x is highest release version of software that can be installed in the controller. If a newer version of software is installed, the controller may fail on start up with a license version error. Niagara 4 licenses, starting at version 4.0, are not backward compatible with NiagaraAX (version 3.x) software.
hostid	<code>hostid="x"</code>	Identifies with an alphanumeric code the unique identify of the host where x is the number. For example: <code>hostId="ATLAS-SD-XXXX-XXXX-XXXX-XXXX"</code> <code>hostId="Qnx-TITAN-XXXX-XXXX-XXXX-XXXX"</code> <code>hostId="Win-XXXX-XXXX-XXXX-XXXX"</code> The hostId in the license file must match the hostId of the controller, otherwise the controller cannot run a station. A hostId that begins with Win is for a PC.
serialNumber	<code>serialNumber="n"</code>	Designates a controller's unique serial number assigned at the factory where n is a number. The serial number in the license file must match the serial number of the controller. For example: <code>serialNumber="329696"</code>
generated	<code>generated="<date>"</code>	Records the <code><date></code> upon which the license file was generated. For example: <code>generated="2023-01-27"</code>
brand	<code>"brand"</code>	For any license with <code>vendor="Tridium"</code> , the NiCS (Niagara Compatibility Structure) provides a schema that OEMs can use to define the various levels and types of Niagara interoperability that their products support. For example: <code><feature name=accept.station.in="*" accept.station.out="*" accept.wb.out="*" "brand" brandId="tridium"</code>
accept.station.in	<code>accept.station.in="*" "brand"</code>	Provides a list of brands that this local station allows Niagara data to come in from. From the controller's perspective, this is the list of brands that it can accept data from. The "*" is a wildcard designation to allow all brands.
accept.station.out	<code>accept.station.out="*" "brand"</code>	Provides a list of brands to which this local station allows Niagara to share data. This is the list of brands that the controller can share data with.
accept.wb.out	<code>accept.wb.out="*" "brand"</code>	Provides a list of brands that this tool is allowed to connect to and engineer. This is the list of brands that the station can engineer.
brandId	<code>brandId="tridium"</code>	Holds a text descriptor that acts as the identifier for the product line. Every licensed station and tool has a Brand Identifier (BrandID). Each station or tool can have only one BrandID entry.
accept.wb.in	<code>accept.wb.in="*" "brand"</code>	Provides a list of brands that this station allows to be connected to it for engineering of its application. This is the list of brands that can engineer the station.
about	<code>"about"</code>	Designates optional information and does not affect station operation in any way. This information can be useful for filtering records when searching the license database. Two attributes in this feature are typically designated when ordering product: <code><feature name="about" project="Testing" owner="Tech Pubs"/></code>

Item	Syntax	Description
project	project="x"	Is an optional attribute where x designates the name of a project. This grouping should typically be assigned to all controllers used for a particular project. For example: project="Tech Pubs"
owner	owner="x"	Is an optional attribute where x identifies the name of a person or group responsible for the project, or possibly an end user. For example: owner="Tech Pubs"
signature	<signature> </signature>	This ending element contains a digital <signature> that is created when the license file is generated. It prevents tampering with the license file. Attempts to edit the license file to enable additional features render the license file useless. Typically, the signature element is the last element contained in the license, so it is followed by the closing license tag as the last line in the license file. For example: <signature>MCwCFF0dq4wJcYgvhTVtrf0oSyuCDCWjAhRj+H9pNxQGStBnhEkIqK8rONB10g==</signature> </license>

Hardware features that can be licensed

Some license features are specific to JACE controllers.

Alphabetically, these features can be included in a license: dataRecovery, jre8qnx, mstp, ndio, nrrio, and serial.

Feature name	Feature as it appears in the license file	Description
dataRecovery	<feature name="dataRecovery" expiration="never" parts="NDB-SRAM">	Licenses a station's DataRecoveryService , which is sourced from its dataRecovery module. This is required to support installed SRAM (Static RAM), whether integral onboard SRAM, such as for more recent controllers, or another JACE controller with an installed SRAM option card.
jre8qnx	<feature name="jre8qnx" expiration="never">	Licenses the (Oracle) Sun Hotspot Java 8 virtual machine (VM) to be able to run on the Niagara 4 JACE controller. There are no attributes.
mstp	<feature name="mstp" expiration="never" port.limit="5" parts="DR-MSTP-AX"/>	Determines how many of the available serial ports may be used for BACnet MS/TP communications. Features bacnet and serial must also exist in the license file. port.limit="5" specifies the number of serial ports that may be used for MSTP communications. Typically, this number matches the number of physical ports. Some JACE controller models have option card modules or slots with serial ports. If additional ports are added, the port limit may be less than the number of physical ports (if the port activation has not been ordered as well).
ndio	<feature name="ndio" expiration="never" device.limit="none" history.limit="none" point.limit="none">	Enables the NDIO (Niagara Direct Input Output) driver, required to configure and use the JACE controller's Ndio-type I/O modules. Not all JACE controllers support such I/O modules, which attach as a chain directly to the controller using 20-pin connectors. Refer to the specific controller's data sheet to confirm whether this is an available option. In the ndio features line, a device equates to an Ndio Board. History and schedule limits have no practical application. Refer to the <i>NDIO Driver Guide N4</i> for related details.

Feature name	Feature as it appears in the license file	Description
nrio	<code><feature name="nrio" expiration="never" device.limit="16" history.limit="none" point.limit="none" schedule.limit="none" /></code>	Enables the NRIO (Niagara Remote Input Output) driver, which is required to configure and use the JACE controller's Nrio-type I/O modules and/or any onboard I/O of a controller. Most QNX-based JACE controllers support NRIO modules, which communicate via RS-485. In the nrio features line, a device equates to an Nrio16-Module, and that history and schedule limits have no practical application. Refer to the <i>NRIO Driver Guide (N4)</i> for related details.
serial	<code><feature name="serial" expiration="never" /></code>	Enables the use of JACE serial ports for various drivers, for example aapup or modbusAsync. The JACE license needs this serial feature in addition to any specific driver feature. Only one serial feature line is needed regardless of the number of serial-based drivers. In the case of the JACE used for BACnet MS/TP, a license would require this serial feature and the driver features bacnet and mstp.

Driver attributes

Each driver is enabled by a feature line (element) in the license file. Most of the drivers use the same attributes within that feature.

The most common driver attributes are:

```
<feature name="driverName" expiration="expirationDate"
device.limit="none" history.limit="none" point.limit="none"
schedule.limit="none">
```

The various limit attribute values can be either "none" or a numerical (limit) value, for example `device.limit=32`. A limit value of `none` means unlimited, whereas a limit value of `0` means none allowed.

For many drivers, only the `point.limit` and `device.limit` attributes are applicable; yet most drivers include all `.limit` attributes. For example, due to the simplicity of the Modbus protocol, none of the Modbus-related drivers have any history or schedule import/export capability. Thus, `history.limit` and `schedule.limit` values have no significance in the feature for a Modbus driver.

In Niagara 4, and depending on the license, limit attributes in individual drivers may be superseded by global capacity limits using a licensing model that sets limits, which span across multiple drivers. This allows more flexibility to allocate the number of devices, points, and so on—without requiring ongoing license changes.

Attribute	Description
name	Defines the name of the driver, often the same as the actual module (.jar file) name, for example: bacnet, lonworks, etc.
expiration	Defines the date when the driver expires. This is, typically, the same as the expiration property of the license feature. In some cases, such as beta testing agreements, individual drivers may be set to expire even though the main license file is non-expiring.
device.limit	Designates a license limit on the number of devices that may be added to this specific driver network in the station database. Above this limit, any added device component (and all its child components) are in fault. This limit has no impact on the actual physical limitation of a field bus. For example just because the lonworks feature is set to <code>device.limit="none"</code> does not mean that you can exceed the normal limit of 64 devices per segment.
history.limit	Limits the number of histories that can be imported from remote histories (logs or trends) into the station's history space, and/or exported from station histories to appear as histories in remote devices. Above this limit, any added history import descriptor (or history export descriptor) is in fault, and the associated import/export is not successful.
point.limit	Designates the maximum number of proxy points that may be added to the station database for a particular driver. Above this limit, any added proxy point are in fault.

Attribute	Description
schedule.limit	Limits the maximum number of schedules that can be imported from remote schedules into the station's database, and/or exported from station schedules to appear as schedules in remote devices. Above this limit, any added schedule import descriptor (or schedule export descriptor) is in fault, and the associated import/export is not successful.
parts	This is an alphanumeric part code is automatically assigned when generating the license file and is for internal use.

Driver types

Each driver type is enabled by a separate feature element (or line, starting with the `name` attribute), and has common attributes. New Niagara drivers are continually being developed and offered as products. This topic includes some, but not all of the drivers that are available. It is included to illustrate how driver features appear in licenses.

Driver	Description
aaphp	Enables the American Auto-Matrix Public Host Protocol (PHP) driver. The <code>serial</code> feature is also required.
aapup	Enables the American Auto-Matrix Public Unitary Host (PUP) driver. The <code>serial</code> feature is also required.
bacnet	Enables the functionality of the BACnet driver for BACnet/Ethernet and BACnet/IP. If the JACE controller's other features can be added to enable BACnet MS/TP communications over serial ports: <code>mstp</code> and <code>serial</code>. <pre><feature name="bacnet" expiration="never" device.limit="none" export="true" history.limit="none" point.limit="none" schedule.limit="none"/></pre> <code>export="true"</code> enables BACnet server operation. <code>export="false"</code> permits only BACnet client operation. When BACnet export is enabled, any station histories and/or schedules that are exported to BACnet do not count towards any <code>history.limit</code> or <code>schedule.limit</code> values in the license (if any). Refer to the <i>Niagara 4 BACnet Driver Guide</i> for details on all BACnet integration.
bacnetAws	Provides added functionality as a BACnet AWS Supervisor with BTL-certification as described in the BACnet "Advanced Operator Workstation" specification (B-AWS). Available for PC platforms only (not JACE platforms), this BACnet feature is also required in the license. More details are available in an appendix of the <i>Niagara 4 BACnet Driver Guide</i> .
bacnetOws	Provides added functionality as a BACnet OWS Supervisor with BTL-certification as described in the BACnet "Operator Workstation" specification (B-OWS). Available for PC platforms only (not JACE platforms), more details are available in an appendix of the <i>Niagara 4 BACnet Driver Guide</i> .
fileDriver	Enables the driver used to import comma or tab delimited text files and convert them into histories. For more details, see the <code>file-FileNetwork</code> topic in the <i>Niagara Drivers Guide</i> .
lonworks	Enables the Lonworks driver. Using this driver requires a LON interface on the JACE controller. Most controller models require an optional Lonworks interface card to be installed. More details are available in the <i>Niagara Lonworks Driver Guide</i> .
modbusAsync	Enables the Modbus Master Serial driver. The JACE controller operates as the Modbus Master device communicating via an available serial port using either Modbus RTU or Modbus ASCII. The <code>modbusCore</code> and <code>serial</code> features are also required.
modbusCore	Required by the JACE controller or Modbus Supervisor host for any of the Modbus drivers (Async, Slave, TCP, TCP Slave). For details on any Modbus driver, refer to the <i>Niagara Modbus Driver Guide (N4)</i> .
modbusSlave	Enables the Modbus Slave Serial driver. The JACE controller operates as a Modbus Slave communicating via an available serial port using either Modbus RTU or ASCII to a Modbus Master device. The <code>modbusCore</code> and <code>serial</code> features are also required.
modbusTcp	Enables the Modbus Master TCP driver. The JACE controller or Modbus Supervisor operate as a Modbus Master device communicating via Modbus TCP/IP. The <code>modbusCore</code> feature is also required.
modbusTcpSlave	Enables the Modbus Slave TCP driver. The JACE controller or Modbus Supervisor operates as a Modbus Slave device communicating via Modbus TCP/IP. The <code>modbusCore</code> feature is also required.

Driver	Description
obixDriver	Enables the oBIX driver. The driver supports the oBIX protocol, which is M2M (Machine-to-Machine) communications via XML over TCP/IP. A license is required for this feature. The license reads as follows: <pre><feature name="obixDriver" expiration="never" device.limit="none" export="true" history.limit="none" point.limit="none" schedule.limit="none"/></pre> export="true" enables oBIX server operation. export="false" permits only oBIX client operation. Refer to the <i>oBIX Guide – N4</i> for related details.
opc	Enables the OPC client driver, and is only available on Windows-based platforms because of the protocol's dependency on Windows. Refer to the <i>OPC UA Driver Guide</i> for related details.
niagaraDriver	Enables communication via the Fox protocol to other NiagaraStations, and allows creation of a NiagaraNetwork including proxy points, importing/exporting histories and schedules, and routing alarms. <pre><feature name="niagaraDriver" expiration="never" virtual="true" schedule.limit="none" point.limit="none" history.limit="none" device.limit="none" parts="ENG-WORKSTATION"/></pre> For more details, refer to the <i>Niagara Drivers Guide</i>.
rdbOracle	Enables the Relational Database Driver using the Oracle database format. This driver allows exporting of histories from the NiagaraStation to an Oracle database. The driver does not include the Oracle software, which must be purchased separately from a third-party source. <pre><feature name="rdbOracle" expiration="never" parts="ENG-WORKSTATION"/></pre>
rdbSqlServer	Enables the Relational Database Driver using the Microsoft SQL database format. This driver allows importing and exporting of histories to and from the NiagaraStation, and to and from a Microsoft SQL database. The driver does not include the Microsoft SQL software, which must be purchased separately from a third-party source. The driver does work with the MSDE version, which is free from Microsoft; however, the normal Microsoft imposed limitations on the MSDE version still apply. <pre><feature name="rdbSqlServer" expiration="never" history.limit="10" historyImport="true" parts="ENG-WORKSTATION"/></pre>
snmp	Enables the SNMP (Simple Network Management Protocol) driver, which allows sending and receiving SNMP messages. <pre><feature name="snmp" expiration="never" device.limit="none" history.limit="none" point.limit="500" schedule.limit="none"/></pre> Refer to the <i>Niagara Drivers Guide</i> and <i>Snmp V3 Driver Guide</i> for related details.
videoDriver	Enables the Niagara Video Framework driver (modules <code>nvideo</code>, <code>videoDriver</code>, <code>nDriver</code>) that provide the foundation to integrate select commercial off-the-shelf video surveillance and recording systems into the Niagara station. Depending on the specific video hardware used, one or more vendor-specific license feature entries are also required. Refer to the <i>Niagara Video Framework Guide</i> for related details.

Applications

Alphabetically, application types listed here include box, email, ldapv3, mobile, provisioning, search, template, station, web, and workbench. Applications station, web, and workbench have special importance, and are summarized first.

All Niagara 4 hosts are capable of TLS operation without this license feature. For details, refer to the *Niagara Station Security Guide*.

Application	Description
station	Enables a station to run. This application is present in every JACE platform and Supervisor platform. It may not be present in a license for an engineering workstation (PC) unless specifically ordered with it. Optional attributes are listed below. <pre><feature name="station" expiration="2025-04-01" resource.limit="none" guestEnabled="true"/></pre>
resource.limit	Establishes limits on resources in thousands of resource units (kRUs). <pre>resource.limit="none" enforces no limit.</pre> If the <code>resource.limit</code> flag is specified and the actual resource units exceed the limit, the station displays a warning on startup. If the resource units exceed the limit by 110%, the station will not boot at all. In Niagara 4, the <code>resource.limit</code> flag is superseded by global capacity limits, which uses a licensing model that sets limits on the numbers of devices, points, histories, and so on that span across multiple drivers in the station. This allows a clearer measure of resource capacity in a license than the <code>resource.limit</code> method.
guestEnabled	Enables the guest in the UserService. <code>guestEnabled="true"</code> is required, otherwise the station's UserService has its built-in user guest hidden upon first station start up as a security measure. Only hosts licensed as demo hosts can enable and use the guest user. It is unavailable on any host with a non-expiring license.
web	Enables the WebService in a running station to access the web server via a browser HTTP connection. If not licensed, the server is set to fault with an appropriate <code>faultCause</code>. <pre><feature name="web" expiration="never" ui="true" ui.wb="true" ui.wb.admin="true"/></pre> Full Workbench can connect to a station (via a Fox connection) even if the web feature is missing or expired.
ui	Enables and disables browser access to users with an HTML5 Hx profile. <code>ui="true"</code> allows browser access to users with an HTML5 Hx Profile. <code>ui="false"</code> prevents access to the browser UI with either HTML5 Hx or Wb web profiles. No browser access is allowed, except for Spy pages.
ui.wb	Enables and disables browser access to users with a Wb web profile. <code>ui.wb="true"</code> allows browser access to users with a Wb web profile. <code>ui.wb="false"</code> allows browser access with an HTML5 Hx web profile as long as <code>ui="true"</code>.
ui.wb.admin	Enables and disables browser users with a Wb web profile to access admin-only views on components. <code>ui.wb.admin="true"</code> allows browser users with a Wb web profile access to admin-only views on components, provided they have admin permissions on components with such views. Admin-only views include most types of views, except for property sheet views. For example, wire sheets and most manager views require this option. Browser access to such views is unavailable for any user with an HTML5 Hx web profile. <code>ui.wb.admin="false"</code> renders such views unavailable to Wb web profile users.. Users still have access to the station with a browser subject to the <code>ui</code> and <code>ui.wb</code> flags. Property sheet views are available on components. Slot sheets may be available too, provided a user has admin-level permissions on the components.
workbench	Enables the use of Workbench. This feature must be present to start the full version of Workbench. If the admin flag is false, all views requiring admin access are unavailable. This feature is included for PC platforms only, with the sole exception of the SoftJACE. <pre><feature name="workbench" expiration="never" admin="true"/></pre>
box	Enables a host for Bajascript and a Javascript API (read and write) for Niagara data access from a Javascript enabled environment, such as a web browser. Along with the mobile feature, this feature is required for mobile application support. <pre><feature name="box" expiration="never" session.limit="none" parts="ENG-WORKSTATION"/></pre>
email	Enables the devices monitored by the system and the services that do the monitoring to communicate status, alarms and reports as needed using direct email and SMS messaging. The email feature enables a station to communicate with an SMTP server:

Application	Description
	<pre><feature name="email" expiration="never"/></pre> If the feature is not present, the system marks the EmailService and all incoming and outgoing accounts as in {fault}. The SMS messaging feature enables a station to send text messages to a phone.
ldapv3	Enables a host to use the authentication schemes of LDAP and/or Kerberos for station users under the standard UserService. This allows LDAP to authenticate users using the site's existing Active Directory server or LDAPv3 server. This departs from the former usage of special user services, such as the LdapUserService and LdapV3ADU-serService in place of the standard UserService in Niagara 4. If <code>kerberos="true"</code>, the Niagara 4 host is licensed for Kerberos authentication with LDAPv3. <pre><feature name="ldapv3" expiration="never" kerberos="true" parts="ENG-WORKSTATION"/></pre> Refer to the <i>Niagara LDAP Guide</i> for complete details.
mobile	Enables the host to support the Mobile application framework for station support of web browser access from mobile devices, such as cell phones and tablets. The host requires the <code>box</code> feature for Bajascript support. <pre><feature name="mobile" expiration="never" history="true" schedule="true" alarm="true" px="true" propsheet="true" parts="ENG-WORKSTATION"/></pre>
provisioning	Enables the operation of host provisioning, typically used to automate routine maintenance, such as JACE software upgrades, file distribution and backups. It applies to a Supervisor platform only. Provisioning uses the BatchJobService and a network extension model (for example, a provisioning extension under the NiagaraNetwork) sourced respectively from modules batchJob and provisioningNiagara. <pre><feature name="provisioning" expiration="never"/></pre>
search	Enables the SearchService and the use of searches in the station. Without this feature, the SearchService remains in fault, and the Quick Search box and Search sidebar are unavailable. The <code>local</code> attribute must be <code>true</code> to allow searches local to this station. The <code>system</code> attribute allows searches that can span multiple stations. <pre><feature name="search" local="true" system="true" expiration="never"/></pre>
template	Enables the TemplateService and the use of templates in the station. Without this feature, the TemplateService and templates remain in fault. <pre><feature name="template" expiration="never"/></pre>

Submitting a license request

If you are connected to a host that has not yet been assigned a license by the server (or has a pending license), you can submit a license request to the server.

Prerequisites: You are using Workbench running on a PC with access to the Internet. A host platform does not have a license yet.

Step 1 Connect to the host.

A license request form opens in your computer's default browser.

The screenshot shows a web browser window with the address bar displaying `axlicensing.tridium.com/license/request?params=YnJhbm1`. The page title is "Bind License | NiagaraCen". The Niagara licensing logo is at the top left. The main content area is titled "Request/Bind License" and contains two sections: "License Details" and "Requester Details".

License Details

Host Id* : Win-5BE1-B094-FC24-3440

License Key* :

Requester Details

Name* :

Company* :

E-mail* :

At the bottom of the form are two buttons: "Cancel" and "Submit".

Step 2 Enter the **License Key**, your **Name**, **Company**, **E-mail** address and click **Submit**.

Upon approval, the license server sends the host's license file, typically in a zipped format, by e-mail back to the entered address. The license is also available for automatic retrieval using the corresponding licensing server operations from various views, such as the **License Manager** view, and so forth.

Step 3 To activate that license, enter the license key you received in e-mail along with the part number.

Importing a host license from the license server

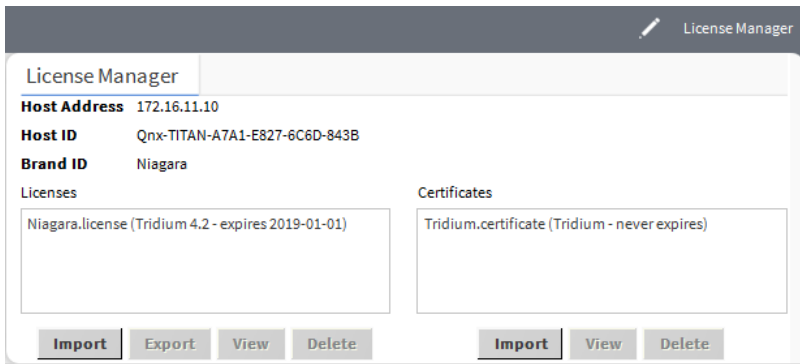
If your PC currently has Internet access while running a platform connection to any host, the **License Manager** automatically retrieves and installs individual licenses. You can also retrieve and install a license using the **Import** button, then select the license server option. The system automatically updates your local license database.

Prerequisites: The host to which you are connecting is licensed.

You can use the **Import** button to install a license file from a local file, from the licensing server, or from a local license database.

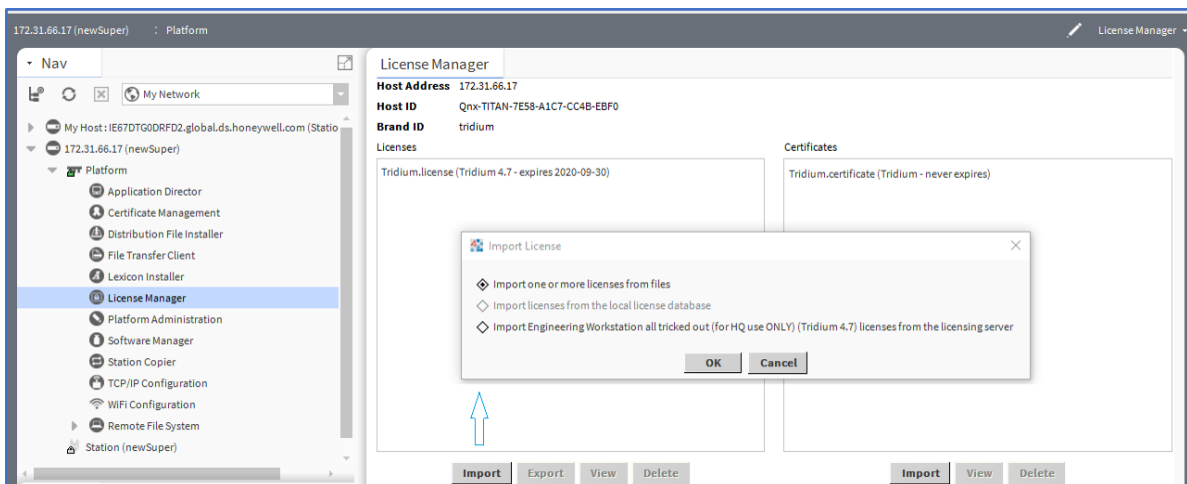
Step 1 In the platform's **Nav Container View**, double-click **License Manager** and supply your credentials.

The **License Manager** view opens.



Step 2 Under the **Licenses** pane, click **Import**.

The **Import License** window opens.

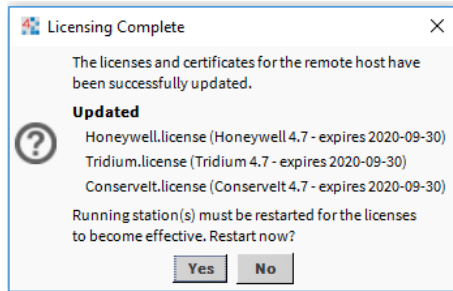


Step 3 Select an option and click **OK**.

- **Import one or more licenses from files** opens a **Select File** window in which you can navigate to either a source license archive (.lar) file or an unzipped license file. When you select a license or license archive file, the software attempts to install the license in the host platform.
- **Import licenses from the local license database** is unavailable (dim) if the host's license file is not in your local license database, or if the license in your local license database already matches the currently installed license. With this option selected, the license is immediately installed in the remote host platform.
- **Import licenses from the licensing server** is available if the PC you are working from has Internet connectivity. When you select this option, the system silently searches the licensing server and installs the license.

Depending on the option chosen and the success of the import attempt after you clicked **OK**, one of several windows may open to signal completion, as follows:

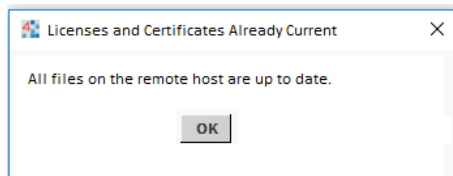
- **Licensing Complete**



Indicates that the license was successfully added.

If a station is running on the host platform, this window informs you that the station must be restarted for the license(s) to become effective, and provides a **Yes** button to do this now. Or, you can select **No** and do this manually later.

- **Licenses and Certificates Already Current**



Indicates that the license currently installed on the host already matches the source license (whether specifying any of the license import options). A window opens.

- **File Not Installed**

Indicates that the software found no appropriate license (by host ID) in either the license file or the license archive specified when importing by file.

- **(License Request Form, in browser)**

If importing from the license server, and an existing license was not found for this host platform, a separate window (of your default browser) opens with a license request form, showing the host ID.

Installing a controller license

If you did not license the controller during commissioning, you either need to access the licensing server over the Internet or have downloaded the license from the server in advance. You may use Workbench or the web UI to install a license in a remote controller. This procedure documents installation using Workbench.

Prerequisites: Your PC is connected to the controller platform and running Workbench. The PC is connected to the Internet, or you downloaded and have available the license file.

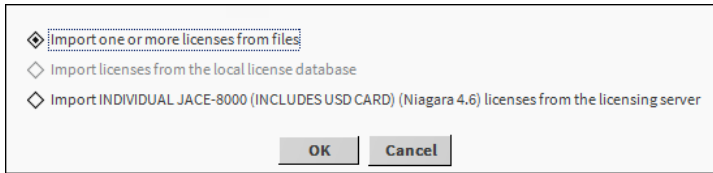
Initial licensing needs to be done with a platform connection because the station will not run until it is licensed.

Step 1 Connect to the controller platform and double-click the **License Manager** row in the **Nav Container View**.

The **License Manager** view opens.

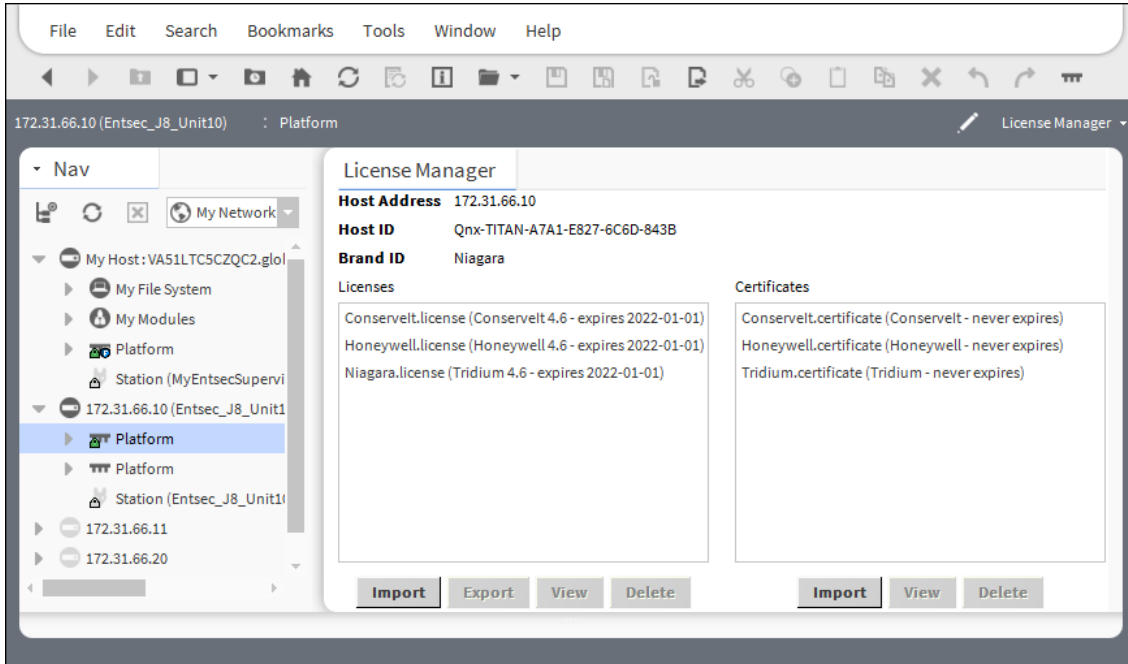
Step 2 Click the **Import** button under the **Licenses** pane.

The **Import License** window opens.



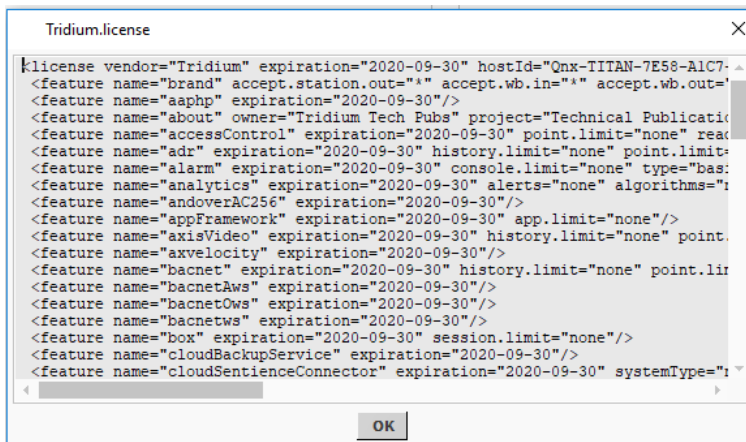
Step 3 Select the import option to use and click **OK**.

After the software imports the license you see it in the **License Manager**.



Step 4 To view a license, select it and click **View** or double-click the license in the table.

The license file opens.



A license and a certificate are each a digitally-signed text file, with differences briefly as follows:

- A license file is unique to a specific host, and enables a set of vendor features. All hosts require a branded Tridium license. If third-party modules are installed, one or more additional licenses may be needed.

- A certificate file varies by vendor, and matches that vendor to a public key used for encryption and to verify the authenticity of license file. All hosts require a default certificate. If third-party modules are installed, one or more additional certificates may be needed.

CAUTION: Do not delete an existing license or certificate without a specific reason, as you will likely render the controller inoperable until a proper license or certificate is reinstalled!

Step 5 After installing a license, open the **Application Director**, confirm that both **Auto Start** and **Restart on Failure** are selected, then start the station by clicking **Start**.

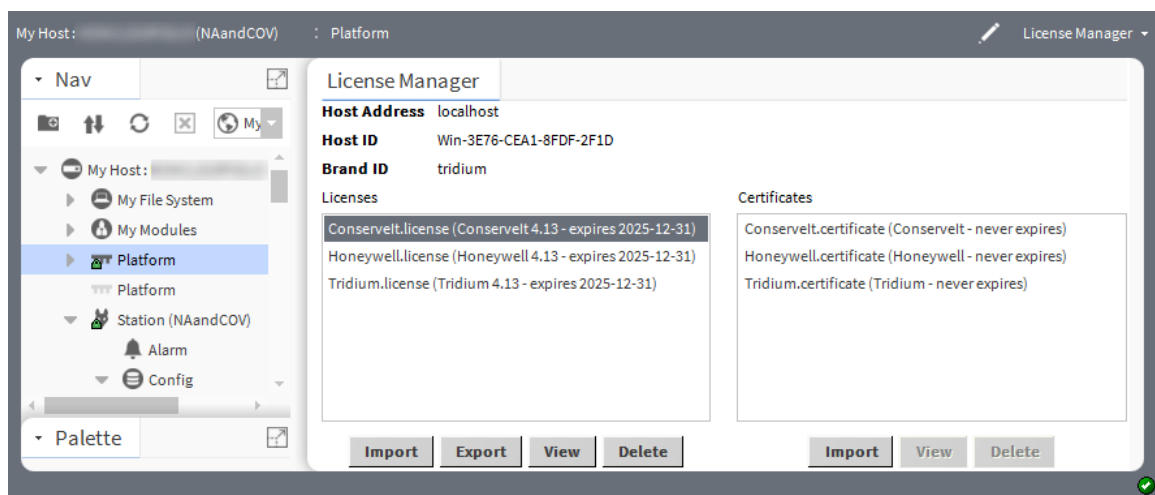
Exporting a license

The ability to export a license using the **License Manager** is always available if you have a license selected, to save locally as a license archive file. This procedure differs from the procedure to export a license without first establishing a platform connection.

Prerequisites: You are connected to the controller or PC and running Workbench.

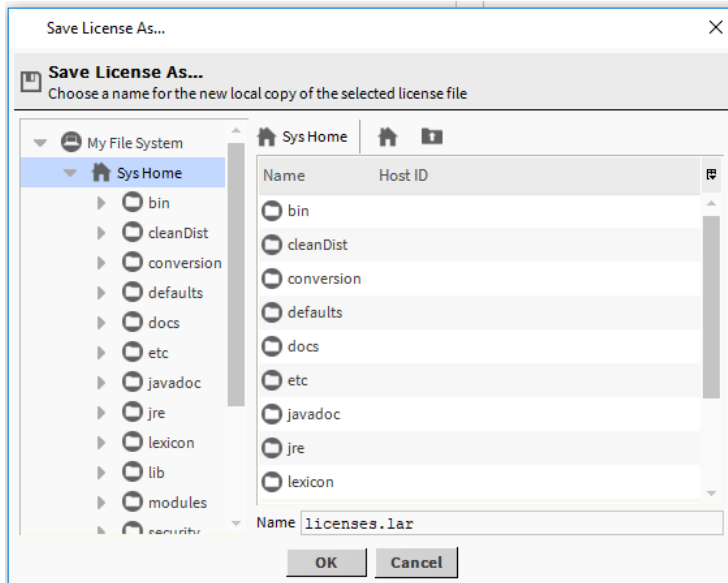
Step 1 Expand **Platform** and double-click **License Manager**.

The **License Manager** view opens.



Step 2 Select the license to export and click **Export**.

A **Save License As...** window opens.



By default, the system saves a license archive file in the root of your Niagara release directory.

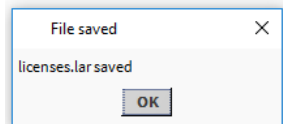
Step 3 If needed, use the window's navigation controls to specify another target folder or drive.

Step 4 Before saving, you can also rename the license archive file, to make it more identifiable.

For example, instead of: `licenses.lar`, you could rename it `My6E.lar`.

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

A notification window opens.



The system saves the license in a compressed (zip-compatible) format known as a license archive. This is a file with a `.lar` file extension. It includes the complete `licenses/hostID` folder (subdirectory) structure (relative to `sys home`) for any included licenses.

Step 6 To view the license zip file, use Windows Explorer to navigate to the folder that contains the file, right-click the `.lar` file and open it with a utility, such as 7-zip.

The `.lar` file contains only the license(s) for the host to which you are connected.

Remote license management

In addition to the `!security/licenses/db` folder, there is also a `!security/licenses/inbox` folder. This inbox allows you to drag a license into your license database of both individual license files and license archive (`.lar`) files, which may have been saved or exported from other PCs, or perhaps sent to you from the licensing server.

After copying license files and/or `.lar` files into your `inbox` subfolder, you close and restart Workbench. The appropriate host ID named subfolders are automatically created in your local license database, each with the appropriate license file(s). The contents of the `inbox` folder are then deleted.

After you restart Workbench, you can provision remote licenses and use the synchronize online feature of the **Supervisor License Manager** to ensure you have the latest version of all your licenses.

Preparing the Supervisor's license database

Before provisioning licenses in multiple remote hosts, the database of controller licenses in the Supervisor PC must be updated.

Prerequisites: You are working in Workbench connected to a Supervisor PC. All licenses for to be provisioned are available: you have downloaded licenses from the license server, received them in an email attachment or exported license archive files (.lar files) from another PC.

Step 1 Using Windows Explorer, locate the `!security/licenses/inbox` folder on your Supervisor PC.

This folder provides a single location from which to update the Supervisor's license database.

Step 2 Copy or drag the individual remote licenses or license archives (.lar files) into this folder.

Step 3 Close and restart Workbench.

Workbench automatically creates the appropriate host-ID-named subfolders in your local license database, each with the appropriate license file(s). Then it deletes the contents of the `inbox` folder.

Now you can use the synchronize online feature of the **Supervisor License Manager** to ensure you have the latest version of all your licenses.

Automating host license updates using provisioning

This procedure uses the **Niagara Network Job Builder** to create a one-time provisioning job to automate updating the license in one or more host controllers.

Prerequisites: The **BatchJobService** is available under **Services** and the **ProvisioningNwExt** component is available under your **NiagaraNetwork**.

Step 1 Double-click **ProvisioningNwExt**.

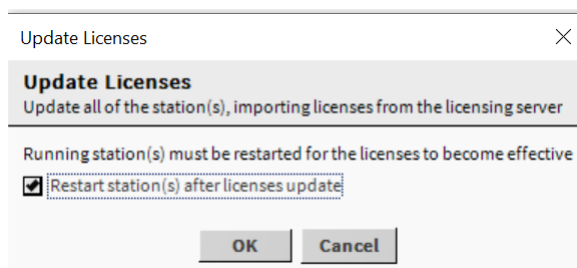
The system opens the **Niagara Network Job Builder** view.

Step 2 In the top pane, **Provisioning steps to run**, click add (+).

The **New Job Step** window opens.

Step 3 Click the **Update Licenses** step and click **OK**.

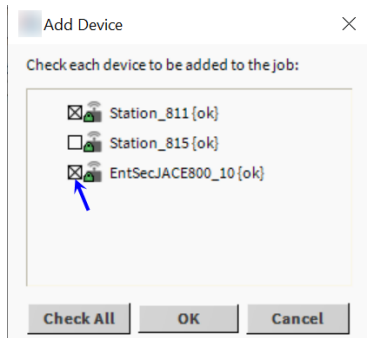
The **Update Licenses** window opens.



Step 4 To continue, click **OK** (or to exit the License Update, click **Cancel**).

Step 5 In the bottom pane, **Stations to include in the job**, click add (+).

The **Add Device** window opens.



Step 6 To select the stations to upgrade, click one or more of the boxes next to the station names and click **OK**.

Step 7 To initiate the provisioning job, review your choices and click **Run Now** at the bottom of the **Niagara Network Job Builder View**.

The view changes to the **Niagara Network Job View**, where steps and results appear as they are executed.

The system updates the licenses for all selected hosts. To make updating licenses a regular, automatic event, you need to create a job prototype.

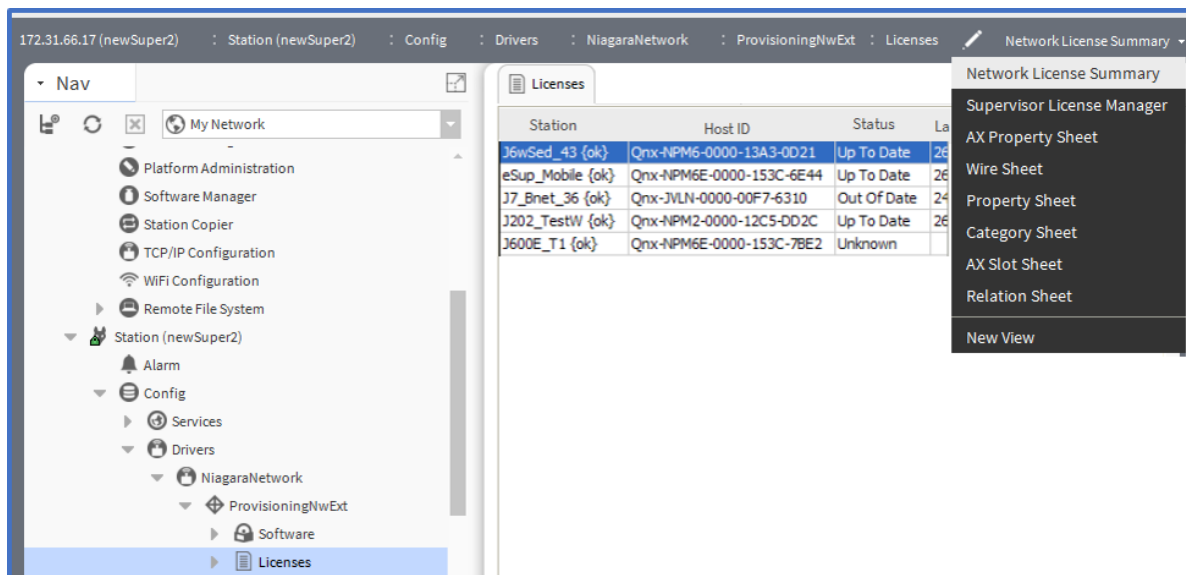
Updating licenses using the Network License Summary

Rather than create a one-time provisioning job, you can update the license on one or more remote hosts using the **Network License Summary**.

Prerequisites: You have synchronized the Supervisor's license database with the host controllers in your network, purchased a license upgrade for each host, and the upgrades are available on the online licensing server.

Step 1 Select the **Licenses** slot on the **ProvisioningNwExt**.

The system displays the **Network License Summary**.



Step 2 Select one or more stations and click **Update**.

If a newer license is found (than that already installed), the system installs it in the remote host (s), updates the license(s) in the Supervisor's local license database, and resets the `Last Updated` timestamp to the time of the update.

Synchronizing with the Supervisor's license database

The local Supervisor maintains a database that includes information about each host's license. Periodically, it is a good idea to interrogate each Supervisor and update this license database. You use this license database when provisioning remote controller licenses.

Prerequisites: You have a network of licensed hosts. The `ProvisioningNwExt` component is available under the `NiagaraNetwork`.

Step 1 Expand the **Config→Drivers→NiagaraNetwork→ProvisioningNwExt** in the Nav tree to see its



Licenses node.

Step 2 Right-click **Licenses** and select **Views→Supervisor License Manager**.

The **Supervisor License Manager** window opens.

Step 3 Click **Synchronize**.

The system prompts with the option to **Synch All Licenses?**

Step 4 Click **Yes** and, at the `Synchronization Complete` prompt, click **OK**.

The Supervisor's license database contains the license identifier for each host in the network.

Subscription licensing

Subscription licensing is available in Niagara 4.13 and later. This feature allows you to regularly pay for a Niagara instance license rather than pay the total up-front cost of a standard license. Niagara instances with Subscription licensing periodically access the Niagara Licensing Server to check the license and get updated subscription information. As long as you keep the subscription up to date, the license continues to allow any licensed feature to run.

Subscription licensing works at the top-license level and does not support subscription-licensed features on perpetual licenses or pools of licenses that can be handed out on demand.

This feature makes it easier to run Niagara on virtual machines and informs you if you accidentally attempt to run multiple VMs with the same license.

Requirements

Successfully registering a subscription license requires several things.

Requirements for subscription licensing include the following:

- A valid Niagara Central account with Subscription Licensing entitlement enabled
- A Niagara instance
- A Niagara 4.13 version of Workbench to connect to and from with which to commission the subscription-licensed Niagara device (the Niagara instance)
- Access by the subscription-licensed device to the following URLs and ports:
 - `https://www.niagara-community.com:443`
 - `https://www.niagaracentralapis.honeywell.com: 443`
- A subscription key that is provided with your Niagara container purchase

In addition, the subscription licensing procedures assume that you are an experienced Niagara and Workbench user.

Registering Niagara running in a container

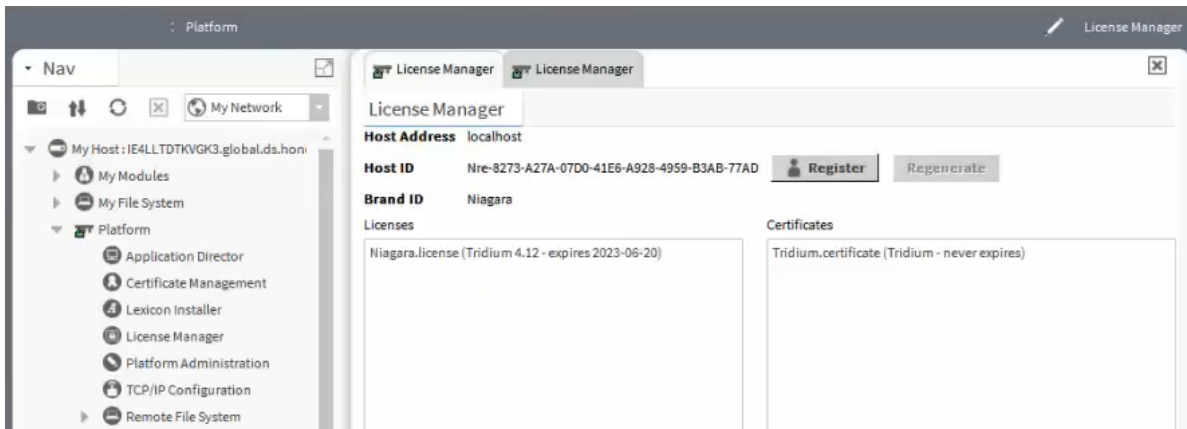
You register a Niagara container instance through the platform **License Manager**.

Prerequisites: Your version of Niagara is 4.13 or higher. You have a subscription key.

Step 1 Start the Niagara container instance.

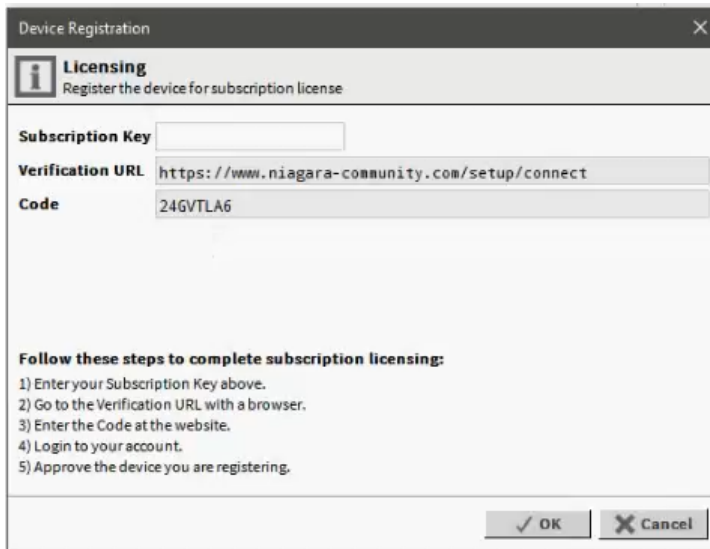
Step 2 Launch Workbench, open a platform connection to the container instance, expand the platform node, and double-click **License Manager**.

The **License Manager** view opens.



Step 3 To register the instance, click the **Register** button to the right of **Host ID**.

The **Device Registration** window opens.

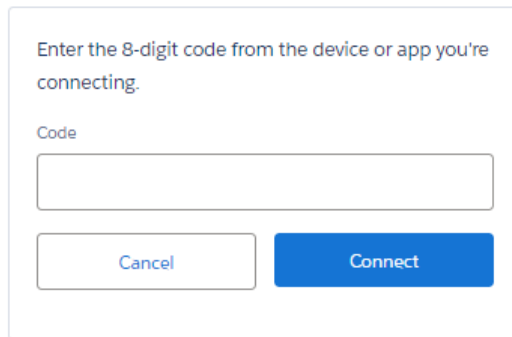


Step 4 Enter the **Subscription Key** provided with the Niagara container purchase and copy the **Code**.

Step 5 To open the Niagara Community page, click the **Verification URL**.

The **Enter Code** window opens.

Enter Code

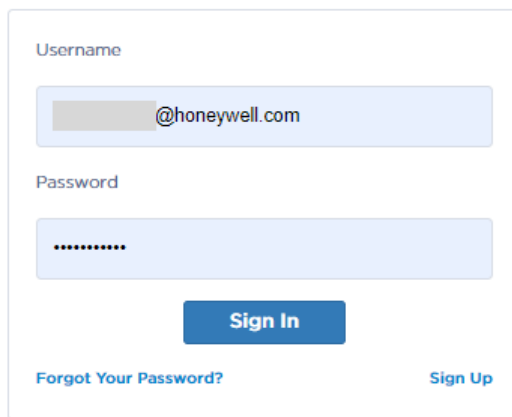


Enter the 8-digit code from the device or app you're connecting.

Code

[Cancel](#) [Connect](#)

- Step 6** Paste the **Code** you copied and click **Connect**.
The Niagara Community Login window opens.



Username

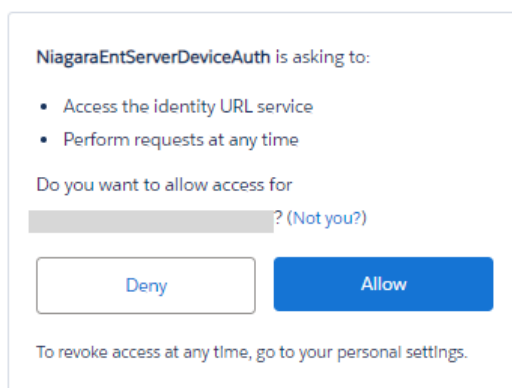
Password

[Sign In](#)

[Forgot Your Password?](#) [Sign Up](#)

- Step 7** Enter your Niagara Central username and password and click **Sign In**.
The **Allow Access?** window opens.

Allow Access?



NiagaraEntServerDeviceAuth is asking to:

- Access the identity URL service
- Perform requests at any time

Do you want to allow access for

? (Not you?)

[Deny](#) [Allow](#)

To revoke access at any time, go to your personal settings.

- Step 8** To authorize yourself as the user, click **Allow**.
This allows the subscription licensing server to use your Niagara Central credentials for authorization.

The **License Manager** fetches the license attached to the host ID, and starts Workbench.

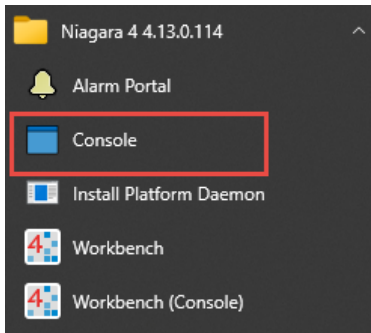
Register Niagara using the NRE Command

If you do not wish to use Workbench to register your Niagara instance, you can use a command line.

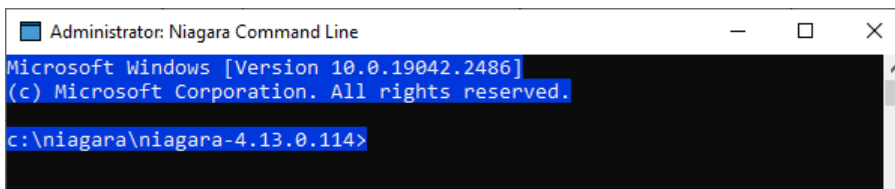
Prerequisites: Your version of Niagara is 4.13 or higher. You have a subscription key.

Step 1 Open a command prompt.

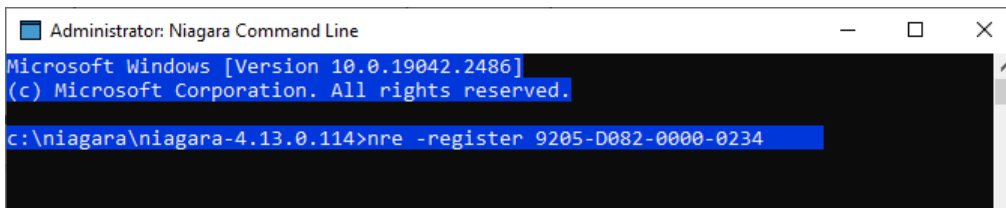
On **Windows**, this can be done by opening the **Console** application in the Niagara installation, or on **Linux**, by opening the Niagara Shell application.



The **Administrator: Niagara Command Line** window opens.



Step 2 Type `nre -register SubscriptionLicenseKey` and press enter.



The license server connects and returns the a Verification URL and User Code:

```
Administrator: Niagara Command Line - nre -register 9205-D082-0000-0234
INFO [14:11:19 02-Feb-23 IST][org.bouncycastle.jsse.provider.PropertyUtils] Found string security property [jdk.tls.disabledAlgorithms]: SSLv3, RC4, DES, MD5withRSA, DH keySize < 1024, EC keySize < 224, 3DES_EDE_CBC, anon, NULL
INFO [14:11:19 02-Feb-23 IST][org.bouncycastle.jsse.provider.PropertyUtils] Found string security property [jdk.certpath.disabledAlgorithms]: MD2, MD5, SHA1 jdkCA & usage TLS, RSA keySize < 1024, DSA keySize < 1024, EC keySize < 224, SHA1 usage SignedJAR & denyAfter 2019-01-01
WARNING [14:11:19 02-Feb-23 IST][org.bouncycastle.jsse.provider.DisabledAlgorithmConstraints] Ignoring unsupported entry in 'jdk.certpath.disabledAlgorithms': SHA1 jdkCA & usage TLS
WARNING [14:11:19 02-Feb-23 IST][org.bouncycastle.jsse.provider.DisabledAlgorithmConstraints] Ignoring unsupported entry in 'jdk.certpath.disabledAlgorithms': SHA1 usage SignedJAR & denyAfter 2019-01-01
INFO [14:11:20 02-Feb-23 IST][org.bouncycastle.jsse.provider.PropertyUtils] Found boolean security property [keystore.type.compat]: true
INFO [14:11:20 02-Feb-23 IST][org.bouncycastle.jsse.provider.PropertyUtils] Found string system property [java.home]: c:\niagara\niagara-4.13.0.114\jre
INFO [14:11:20 02-Feb-23 IST][org.bouncycastle.jsse.provider.PropertyUtils] Found string system property [java.home]: c:\niagara\niagara-4.13.0.114\jre
INFO [14:11:20 02-Feb-23 IST][org.bouncycastle.jsse.provider.PropertyUtils] Found string system property [java.home]: c:\niagara\niagara-4.13.0.114\jre
INFO [14:11:22 02-Feb-23 IST][sys.registry] Up-to-date [105ms]
*****
*** Please go to the Verification URL and enter the User Code to approve
*** the subscription for this Niagara instance
***
*** Verification URL: https://www.niagara-community.com/setup/connect
*** User Code: 24GVBJPT
***
INFO [14:11:25 02-Feb-23 IST][licensing.subscription] Polling for registration status (will poll for 5 minutes)
```

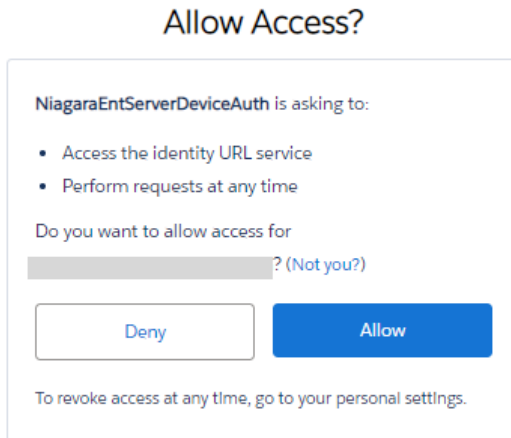
- Step 3 Copy the Verification URL, paste it in a browser and press Enter.
Do not close the command prompt.
The **Enter Code** window opens.

Enter Code

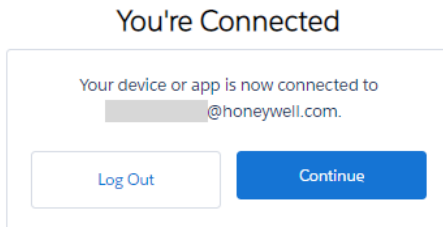
Enter the 8-digit code from the device or app you're connecting.

Code

- Step 4 Go back to the command prompt, copy and paste the User Code and click **Connect**.
The **Allow Access** window opens.



Step 5 To authorize yourself as the user, click **Allow**.
The **You're Connected** window opens.



Step 6 Click **Continue**.
The **License Manager** fetches the license attached to the host ID, and starts Workbench.

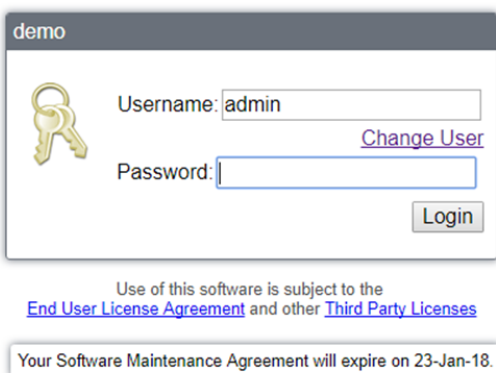
SMA expiration reminder

In Niagara for a browser-based connection to a properly licensed station, the **Login** window displays a Software Maintenance Agreement (SMA) expiration reminder.

The SMA expiration reminder is a licensed feature, which functions as described here:

- SMA Expiration reminder, including expiry date, displays in the **Login** window.

Figure 13 SMA expiration reminder in browser-based station Login window



To connect using Java Web Start [click here](#)

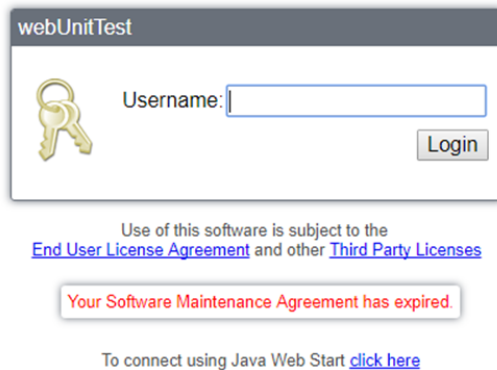
- You can hide this initial expiration notice via the **showExpirationDate** property in the **UserService**.
- SMA expiration reminder displays at the specified number of days before expiration.
 - By default, this occurs at 45 days prior to the expiration date. However this number is configurable (range 30–365 days) via the **expirationReminder** property in the **UserService**.
- When the number of days before expiration is less than or equal to the number set in the **expirationReminder** property, the SMA expiration reminder displays and cannot be dismissed or hidden.

Figure 14 SMA expiration reminder at less than 45 days before expiry date



- Once the SMA is expired, the SMA expiration reminder displays and cannot be dismissed or hidden.

Figure 15 SMA expiration reminder upon expiry date



For details on the **SMA Notification Settings**, see “baja-UserService” in *Getting Started with Niagara*.

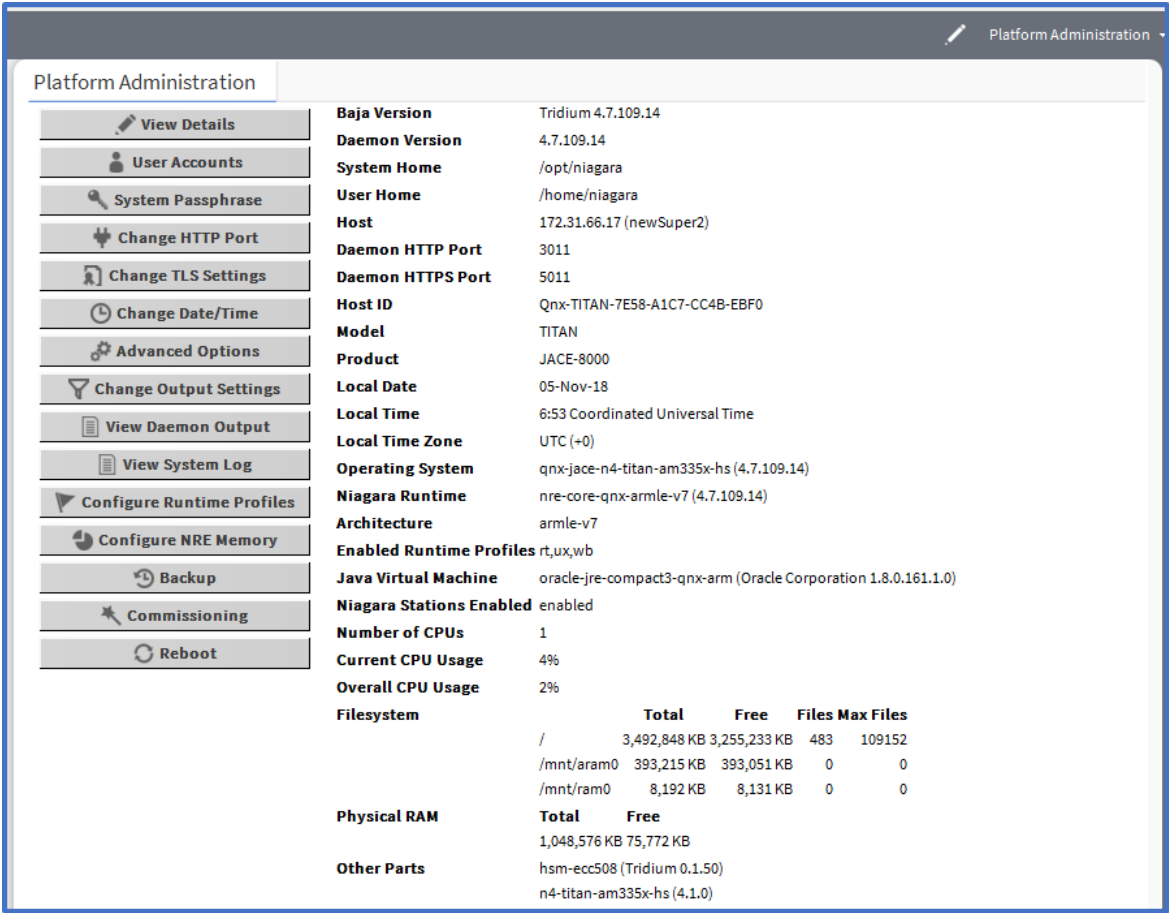
Chapter 8 Platform Administration on a controller

Topics covered in this chapter

- ◆ Platform Administration on an embedded controller
- ◆ Platform Administration on a workstation
- ◆ Viewing platform details
- ◆ Setting up the PC platform's users group (workstation)
- ◆ Creating a platform user account (controller)
- ◆ System and file passphrases
- ◆ Changing the platform daemon's HTTP port
- ◆ Changing TLS Platform settings
- ◆ System date, time and time zone
- ◆ Displaying current CPU usage
- ◆ Viewing daemon output
- ◆ Viewing controller system logs
- ◆ Controller storage conservation
- ◆ Configuring memory to improve performance
- ◆ Backing up a station using Platform Administration
- ◆ Commissioning a controller
- ◆ Rebooting a controller station

The **Platform Administration** view provides access to various platform daemon (and host) settings and summary information.

Figure 16 Platform Administration view on a controller platform



Available functions appear as buttons on the left side, and summary information is listed in the right side. Typical use is when commissioning a new controller, or to troubleshoot platform or host problems.

During a platform connection, upon first access to **Platform Administration**, a small delay occurs while downloading data about that platform's installed modules. You may briefly see a **Loading Modules** window before the main view appears.

Platform Administration on an embedded controller

The **Platform Administration** views available on an embedded controller differ from those that are available on a Windows-based platform.

Figure 17 Platform Administration for a controller platform

Platform Administration

Navigation Options:

- View Details
- User Accounts
- System Passphrase
- Change HTTP Port
- Change TLS Settings
- Change Date/Time
- Advanced Options
- Change Output Settings
- View Daemon Output
- View System Log
- Configure Runtime Profiles
- Configure NRE Memory
- Backup
- Commissioning
- Reboot

System Details:

Baja Version	Tridium 4.7.109.14
Daemon Version	4.7.109.14
System Home	/opt/niagara
User Home	/home/niagara
Host	172.31.66.17 (newSuper)
Daemon HTTP Port	3011
Daemon HTTPS Port	5011
Host ID	Qnx-TITAN-7E58-A1C7-CC4B-EBF0
Model	TITAN
Product	JACE-8000
Local Date	29-Oct-18
Local Time	9:26 Coordinated Universal Time
Local Time Zone	UTC (+0)
Operating System	qnx-jace-n4-titan-am335x-hs (4.7.109.14)
Niagara Runtime	nre-core-qnx-armle-v7 (4.7.109.14)
Architecture	armle-v7
Enabled Runtime Profiles	rt,ux,wb
Java Virtual Machine	oracle-jre-compact3-qnx-arm (Oracle Corporation 1.8.0.161.1.0)
Niagara Stations Enabled	enabled
Number of CPUs	1
Current CPU Usage	3%
Overall CPU Usage	1%

Filesystem

	Total	Free	Files	Max Files
/	3,492,848 KB	3,255,222 KB	483	109152
/mnt/aram0	393,215 KB	392,987 KB	0	0
/mnt/ram0	8,192 KB	8,131 KB	0	0

Physical RAM

Total	Free
1,048,576 KB	75,384 KB

Other Parts

hsm-ecc508 (Tridium 0.1.50)
n4-titan-am335x-hs (4.1.0)

- **User Accounts** manages who can access the controller platform.
- **Advanced Options** enable and disable three properties: SFTP/SSH Port, Daemon Debug, and USB Backup.
- **Configure Runtime Profiles** control which modules are running at any time. Limiting the running modules can conserve memory.
- **Commissioning** performs initial Niagara installation and startup in a remote controller, or when upgrading a controller.
- **Reboot** restarts the platform.
- Various data in the view (repeated in "View Details") differ greatly from that for Windows hosts.

Platform Administration on a workstation

Platform Administration for a Windows-based platform is different from the same for a controller.

Figure 18 Platform Administration for Windows-based platform

Platform Administration							
Baja Version	Tridium 4.7.109.14						
Daemon Version	4.7.109.14						
System Home	C:\Niagara\Niagara-4.7.109.14						
User Home	C:\Niagara\New folder						
Host	My Host : IE67DTG0DRFD2.global.ds.honeywell.com (Station_practice)						
Daemon HTTP Port	3011 (disabled in TLS settings)						
Daemon HTTPS Port	5011						
Host ID	Win-81A8-ED20-1A79-FA9D						
Model	Workstation						
Product	Workstation						
Local Date	29-Oct-18						
Local Time	15:49 India Standard Time						
Local Time Zone	Asia/Calcutta (+5:30)						
Operating System	Windows 10 Enterprise (10.0)						
Niagara Runtime	nre-core-win-x64 (4.7.109.14)						
Architecture	x64						
Enabled Runtime Profiles	rt,se,ux,wb						
Java Virtual Machine	oracle-jre-win-x64-es (Oracle Corporation 1.8.0.181.0)						
Niagara Stations Enabled	enabled						
Number of CPUs	4						
Current CPU Usage	7%						
Overall CPU Usage	79%						
Filesystem	<table border="1"> <thead> <tr> <th></th> <th>Total</th> <th>Free</th> </tr> </thead> <tbody> <tr> <td>C:\</td> <td>482,864,124 KB</td> <td>419,626,972 KB</td> </tr> </tbody> </table>		Total	Free	C:\	482,864,124 KB	419,626,972 KB
	Total	Free					
C:\	482,864,124 KB	419,626,972 KB					
Physical RAM	<table border="1"> <thead> <tr> <th></th> <th>Total</th> <th>Free</th> </tr> </thead> <tbody> <tr> <td></td> <td>8,270,364 KB</td> <td>3,908,740 KB</td> </tr> </tbody> </table>		Total	Free		8,270,364 KB	3,908,740 KB
	Total	Free					
	8,270,364 KB	3,908,740 KB					
Other Parts	None						

- No **User Accounts** button is available, as platform authentication is handled differently, with credentials managed only in Windows.
- No **SFTP/SSH** button is available (equivalent configuration can be done using Windows, if needed). More typically, the "Remote Desktop Connection" feature of Windows is used.
- The **Change Date/Time** button is dimmed (unavailable). To change the date and time in a PC, use Windows.
- The **Configure Runtime Profiles** button is dimmed (unavailable). Windows hosts are invariably enabled for all runtime profiles.
- The **Commissioning** button is dimmed. The **Commissioning Wizard** is intended only for initial Niagara installation and startup in a remote controller, or when upgrading a controller.
- The **Reboot** button is dimmed. This is intended only for remote platforms. Any Windows host must be rebooted using Windows.
- Various data in summary information (repeated in View Details) differ greatly from QNX hosts.

Viewing platform details

The View Details summary includes all the information shown in main **Platform Administration** view, plus installed modules, and other data. Generally, information in this view is helpful when troubleshooting or asking for technical support.

Prerequisites: You are working in Workbench and are connected to the remote controller platform.

Step 1 Expand the **Platform** node in the Nav tree or double-click **Platform**.

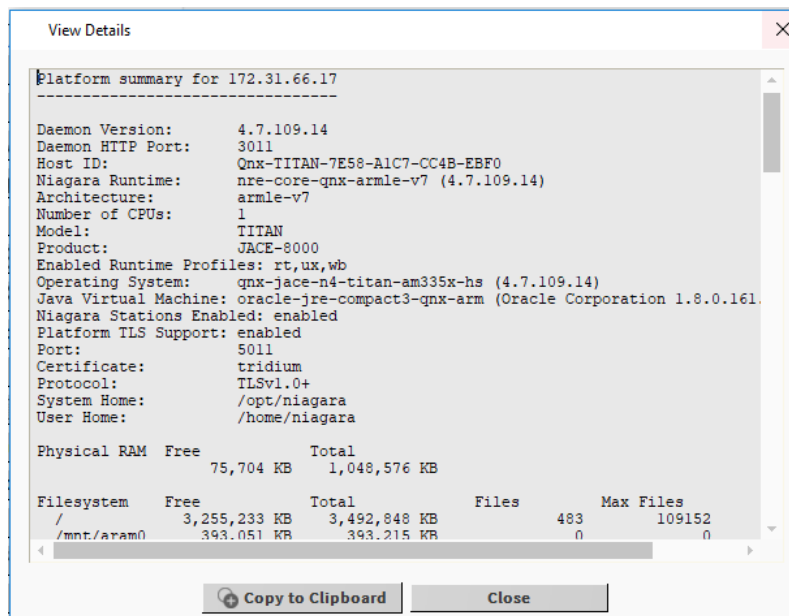
The contents of the Nav Container View opens in the tree or in the main view.

Step 2 Double-click the **Platform Administration**.

The **Platform Administration** view opens.

Step 3 Click **View Details**.

The **View Details** window opens.



Step 4 To copy all details to the Windows clipboard, click **Copy To Clipboard**.

Setting up the PC platform's users group (workstation)

Specifying the Windows users group for platform administrator access is only available on Windows-based hosts. Niagara 4 uses only basic, native Windows OS user authentication for platform access. There is no platform **User Manager** view. Instead, you must use native Windows functions to create and manage Windows OS users and groups. This theme also applies to the **TCP/IP Configuration** view for a Niagara 4 Windows host, which is available, but is read-only (allowing you to review current TCP/IP settings). Also, there is only one level of platform access for any Niagara 4 host—admin level. This level applies to Windows platforms as well as to controller platforms.

Prerequisites: You are working in Workbench and are connected to a Supervisor PC.

Step 1 Expand the **Platform** node in the Nav tree or double-click **Platform**.

The contents of the Nav Container view opens in the tree or in the main view.

Step 2 Double-click the **Platform Administration**.

The **Platform Administration** view opens.

Step 3 Click **Update Authentication**.

An **Authentication** login window opens.

Authentication

Platform Daemon Authentication
Platform daemon credentials

Please enter the username and password you use when logging into the host's domain:

User Name

Password

◀ Back ▶ Next ✓ Finish ✕ Cancel

- Step 4 Use your standard Windows login credentials—if the host is on a Windows domain, log in using the credentials you use when logging in to that domain.

This is necessary to limit the number of possible domain groups to only those groups in which you are a member. Such groups are selectable in the next window to choose the sole Windows users group for platform daemon access, as shown in the figure below.

Authentication

Platform Daemon Authentication
Platform daemon access control

Please choose the Windows group which will have the ability to use the platform daemon:

Platform Daemon Group

Authentication

Platform Daemon Authentication
Platform daemon access control

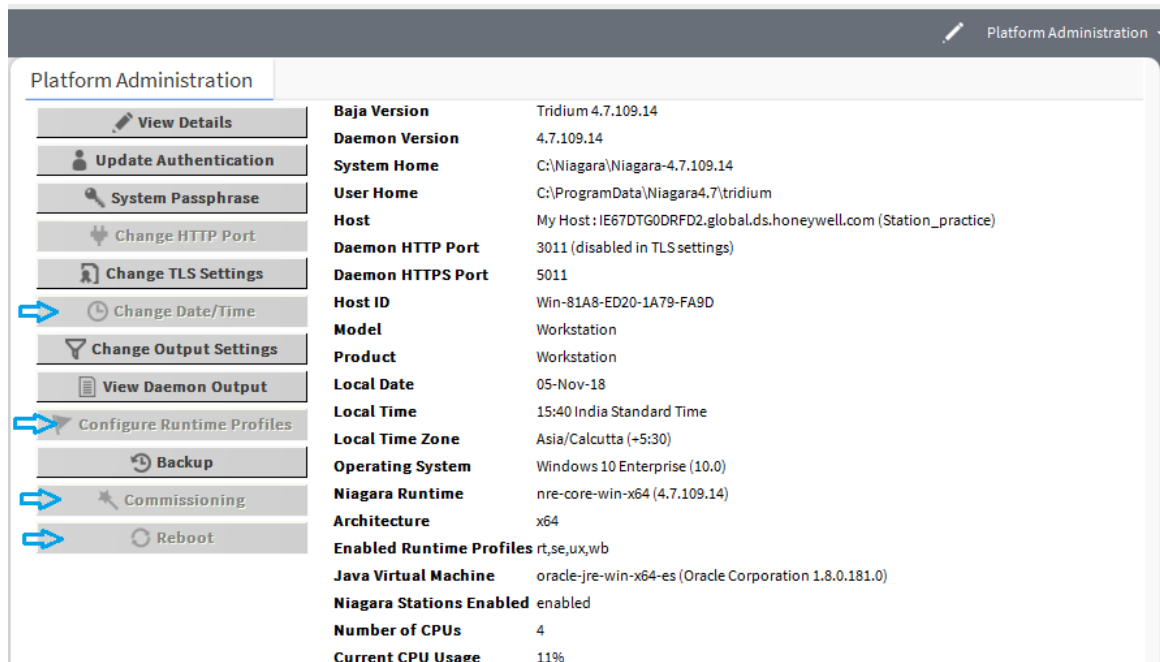
Please choose the Windows group which will have the ability to use the platform daemon:

Platform Daemon Group

- BUILTIN\Administrators
- BUILTIN\Backup Operators
- BUILTIN\Cryptographic Operators
- BUILTIN\Distributed COM Users
- BUILTIN\Event Log Readers
- BUILTIN\Guests
- BUILTIN\Hyper-V Administrators
- BUILTIN\IIS_IUSRS
- BUILTIN\Network Configuration Operators
- BUILTIN\Performance Log Users
- BUILTIN\Performance Monitor Users
- BUILTIN\Power Users
- BUILTIN\Remote Desktop Users
- BUILTIN\Remote Management Users

This window lets you select the one Windows users group that can make platform connections to this host. Groups include Windows built-in user group with the “BUILTIN” or “NT AUTHORITY” prefix), as well as any locally-defined user groups. If the host has been added to a Windows domain, groups defined in that domain are also listed and available. Domain groups are limited to only those in which the login user is a member.

When platform-connected to a remote Niagara 4 Windows host, some **Platform Administration** view buttons are unavailable, as shown in the figure below.



As shown above, **Change Date/Time**, **Commissioning**, and **Reboot** are unavailable when the platform is connected to any Niagara 4 Windows host (remote or local). For a local platform connection, **Configure Runtime Profiles** is also unavailable.

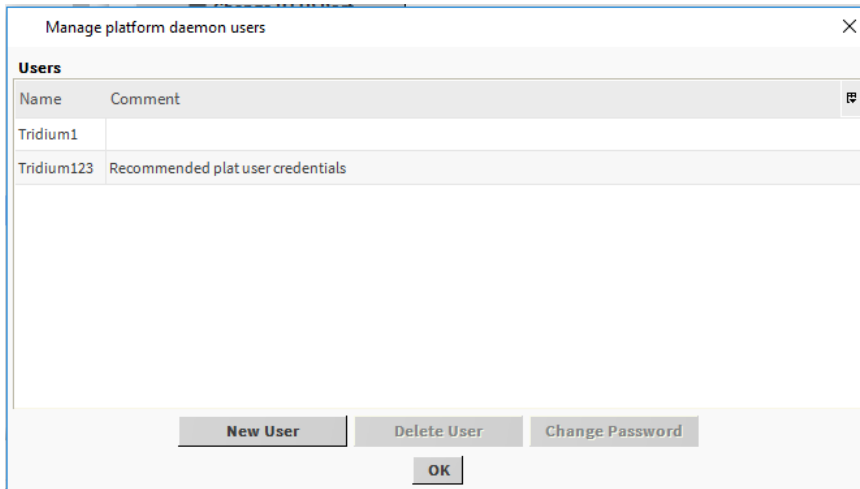
Creating a platform user account (controller)

The Commissioning Wizard initially guides the creation of the first admin platform user. The User Accounts button is only available on a remote controller's **Platform Administration** view. You can create multiple platform administrator users (up to 20 maximum). All have the same full administrator permissions, can create additional users, and can change the password of their own account.

Prerequisites: You are working in Workbench and are connected to a remote controller platform.

If you are commissioning a new unit, or a controller that has had a cleanDist file installed, only a well-known default platform admin account exists. Any unit with the default platform admin user is extremely susceptible to unauthorized intrusion. Therefore, before you can complete other commissioning tasks, the **Commissioning Wizard** requires you to first replace the default platform user account in a wizard step.

- Step 1 Expand the **Platform** node in the Nav tree or double-click **Platform**.
The contents of the Nav Container view opens in the tree or in the main view.
- Step 2 Double-click the **Platform Administration**.
The **Platform Administration** view opens.
- Step 3 Click the **User Accounts** button.
The **Manage platform daemon users** window opens.



Step 4 Click **New User** and fill in the form.

Changing a user's platform password (controller)

Strong user passwords are required for all remote platform users.

Prerequisites: You are working in Workbench and are connected to the remote controller platform.

Step 1 Expand the **Platform** node in the Nav tree or double-click **Platform**.

The contents of the Nav Container view opens in the tree or in the main view.

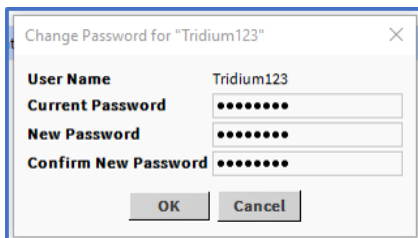
Step 2 Double-click the **Platform Administration**.

The **Platform Administration** view opens.

Step 3 Click **User Accounts**.

Step 4 Select a user and click **Change Password**.

The **Change Password for** user name window opens.



Step 5 Enter the current password, then enter the new password (twice) in the popup window and click **OK**.

NOTE: If the Workbench FIPS property **Show FIPS Options** is set to `true` certain FIPS options become visible in this window. If selected, the framework enforces FIPS-strength password requirements.

If you enter an incorrect current password, an **Invalid login credentials** error popup opens. After clicking **OK** you return back to the change password window above.

If you are changing your password used in your current platform session, your new credentials become immediately effective upon clicking **OK**. If you previously had **Remember these credentials**, selected in the **Authentication** login window, the cached credentials are automatically updated.

Deleting a platform user account (controller)

As a security precaution, platform users should be limited to only valid users. Others should be deleted.

Prerequisites: You are working in Workbench and are connected to a remote controller platform.

Any platform user can delete any other platform user except:

- The user active in the current platform session
- The original platform user, meaning the one created in the **Remove platform default user account** step when using the **Commissioning Wizard** to commission the controller. Such users cannot be selected to delete.

Step 1 Expand the **Platform** node in the Nav tree or double-click **Platform**.

The contents of the Nav Container view opens in the tree or in the main view.

Step 2 Double-click the  **Platform Administration**.

The **Platform Administration** view opens.

Step 3 Click the **User Accounts** button.

The **Manage platform daemon users** window opens.

Step 4 Select the user account to delete, click **Delete User** and click **OK**.

Workbench deletes the user account.

System and file passphrases

All Niagara 4 platforms have a system passphrase (password), used to encrypt sensitive information, such as client passwords stored in BOG files and station databases (config.bog files) or station backup distribution (.dist) files. This passphrase increases security for the files that contain critical information. In various operations, you are prompted to enter the passphrase, such as when copying stations or restoring station backups in remote platforms.

This system passphrase applies to the JACE-8000 and JACE-9000 controllers.

The following areas of the framework are affected by passphrase implementation:

- Provisioning
- **Distribution File Installer** to restore a backup .dist file. If you do not know the passphrase for a .dist file you cannot install it.
- File Transfer Client
- **Station Copier** to transfer a local file.
- Back up
- Commissioning
- Export Tags

The sensitive information in files is protected with encryption, either by encrypting the information within the file or by encrypting the whole file. How encryption is applied depends on the expected portability of the file. Files located under the daemon User Home (files that belong to the system) are encrypted using a strong, randomly generated key that exists only on that system. While files located under the Niagara User Home (that is, portable files that can be sent to many systems) are encrypted using a key derived from the user-defined system passphrase entered during software installation or when the system passphrase is changed.

Due to the different types of encryption that are used for the system and for portable locations, when transferring files between the daemon User Home and another User Home you must use the Workbench platform tools (**Station Copier**, **File Transfer Client** or **Backup**) which convert files to use the correct encryption key for the target location.

CAUTION: Do not use Windows Explorer to copy files between the daemon User Home and other User Homes because without the proper encryption those files may not be readable.

If the file passphrase and system passphrase are the same, a station copy proceeds without prompting for a passphrase.

If the file passphrase is not the same as the target host system passphrase then you are prompted to enter the file passphrase, as shown.

Figure 19 Station Transfer Wizard prompt for bog file passphrase



- **For system-to-portable transfers**

You can get portable copies of files located under the daemon User Home by any of these methods:

- Make a backup from the **Platform Administration** view
- Make a backup from a running station
- Use either **Station Copier** or **File Transfer Client** from the **Platform Administration** view

The resulting local, portable copies and backup files are protected with a file passphrase.

- **For portable-to-system transfers**

Alternately, when you use the **Distribution File Installer** to restore a backup .dist file, or **Station Copier** to transfer a station from your Workbench directory to a controller, the file's passphrase is validated and used to translate the data back into the proper system encryption format for use under the daemon User Home.

CAUTION: It is important to remember the system passphrase and keep it safe. If you lose the system passphrase, you will lose access to encrypted data.

Changing the platform's system passphrase

A unique system passphrase is associated with each remote controller platform. When commissioning a new remote controller platform, Workbench enforces the requirement to change the platform's default system passphrase. After commissioning, you may change the system passphrase at any time.

Prerequisites: You are working in Workbench and are connected to a remote controller platform.

To change the system passphrase you may run the **Commissioning Wizard** or use the **Platform Administration** view as described here.

Step 1 Expand the **Platform** node in the Nav tree or double-click **Platform**.

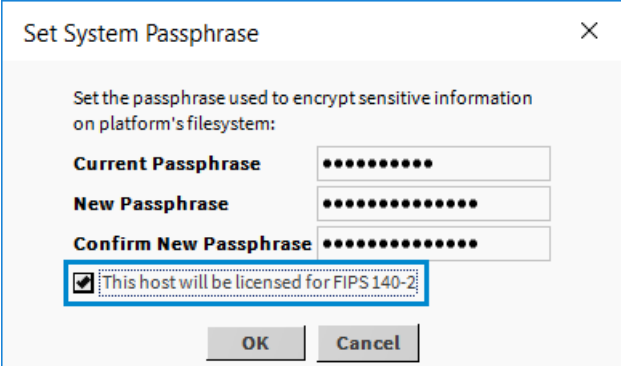
The contents of the Nav Container view opens in the tree or in the main view.

Step 2 Double-click the **Platform Administration**.

The **Platform Administration** view opens.

If you are using the default passphrase, a yellow warning box displays in the bottom right of the **Platform Administration** view. This warning indicates a problem with the passphrase.

Step 3 Click **System Passphrase**.



The **Set System Passphrase** window opens. It contains the following fields and options:

- Current Passphrase**: A text field with masked characters (dots).
- New Passphrase**: A text field with masked characters (dots).
- Confirm New Passphrase**: A text field with masked characters (dots).
- ☒ **This host will be licensed for FIPS 140-2**: A checkbox option.
- OK** and **Cancel** buttons at the bottom right.

The **Set System Passphrase** window opens.

NOTE: If the Workbench FIPS property **Show FIPS Options** is set to `true` certain FIPS options become visible in this window. If selected, the framework enforces FIPS-strength password requirements.

Step 4 Enter the old system passphrase, enter the new system passphrase and confirm it.

The system passphrase must contain at least 10 characters, 1 digit, 1 lower case character and 1 upper case character.

An error popup reminds you if you attempt to enter a password that does not meet minimum rules.

Step 5 Store this passphrase in a safe place.

CAUTION: It is important to remember the system password and keep it safe. If you lose the system passphrase, you will lose access to encrypted data.

Editing the .bog file passphrase offline

Files created in Workbench initially have no passphrase since they do not yet contain sensitive data. If you do not know the passphrase for a .bog file, you can edit it offline. This procedure describes the steps to edit the .bog file offline, set a new passphrase and change the admin user password. This procedure works for a Supervisor or controller station.

Prerequisites: The .bog file to edit is contained in a ZIP file. You are using Workbench running on a PC.

CAUTION: Changing the passphrase on a .bog file results in the loss of sensitive data in the file. Any password values that are encoded with the current passphrase will be cleared. So you will need to enter or re-enter the passwords for all platform and station users.

- Step 1 Using Windows File Explorer, extract the downloaded ZIP file and copy the station folder to your installation's User Home stations folder (C:\Users\UserName\Niagara4.x\brand\stations).
- Step 2 In the Workbench Nav tree, expand **My Host**→**My File System**→**User Home** and navigate to the station folder just copied to that location.
- Step 3 To open the .bog file, expand the station folder and double-click `config.bog`.
- Step 4 On the Workbench tool bar, click the Bog File Protection tool (🔒).
The **Bog File Passphrase** window opens.
- Step 5 Click **Force the file to start using a different passphrase that you specify**, enter the new **Passphrase** and **Passphrase Confirm** values, click **Update** and on the subsequent view, click **Close**.
- Step 6 In the Nav tree, expand **Config**→**Services**, and double-click **UserService**.
- Step 7 In the **AX User Manager**, double-click the admin user and under Authenticator, change the **Password** and **Password Confirm** values.

Repeat this step as needed to change passwords for other users.

Step 8 Right-click the `config.bog` file and click **Save**.

The framework prompts you to enter the file passphrase. You can **Save** only if you enter the correct passphrase.

Step 9 Open the **Station Copier** and copy the station from the your User Home to the target platform.

System passphrase

The JACE-8000 and JACE-9000 makes use of the system passphrase to encrypt sensitive information. For details on how these controllers use the passphrase, refer to JACE-8000 Install and Startup Guide and the JACE-9000 Install and Startup Guide.

When inserting the JACE-8000 SD card into a replacement unit, note the following:

- If the replacement unit is pre-configured with the same system passphrase, the unit starts.
- If the replacement unit has a different system passphrase, the unit will not boot, and the status LED flashes every second. To resolve, you must make a serial connection and, when prompted, select either: **Update the system passphrase**, or **Remove all encrypted data**.

Changing the platform daemon's HTTP port

HTTP port 3011 is monitored by the host's platform daemon for regular platform client connections (connections that are not secure). You may change this port for specific firewall reasons, or, perhaps, for additional security. This differs from the port used for station (Foxs) connections that is secure.

Prerequisites: Your network firewall will allow communication through the newly-configured port. You are working in Workbench and are connected to a remote controller platform.

Step 1 Expand the **Platform** node in the Nav tree or double-click **Platform**.

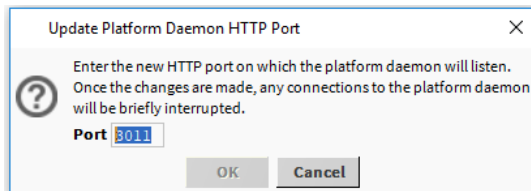
The contents of the Nav Container view opens in the tree or in the main view.

Step 2 Double-click  **Platform Administration**.

The **Platform Administration** view opens.

Step 3 Click **Change HTTP Port**.

The **Update Platform Daemon HTTP Port** window opens.



Step 4 Enter a different port number and click **OK**.

When you click **OK**, the platform daemon restarts, and your platform connection reopens (this does not affect the operation of any running station). If the platform was previously connected on the port without security, the platform icon shows in the Nav tree with the new HTTP port number (:n) in parenthesis.

Step 5 Before closing the host, which removes it from the Nav tree, carefully note the new (non-default) port number you entered.

You must specify this port number the next time you open a platform using a connection that is not secure.

- Step 6** To check this port number in a station running on the host, expand **Config→Services** and double-click **PlatformServices**.

The **Platform Daemon Port** property should reflect the change you just made.

Changing TLS Platform settings

This function on the **Platform Administration** view configures a secure (TLS) platform connection, as well as changes related to secure platform connection (platformtls) properties.

Prerequisites: You are connected to the platform.

- Step 1** Double-click the **Platform** node in the Nav tree or expand **Platform**, double-click **Platform Administration** and click **Change TLS Settings**.

The Platform TLS Settings window opens.

- Step 2** If you are configuring a remote controller you may change **State**, **Port**, and **Protocol**.

When connected to a PC, **State** is set to **TLS only**, the daemon HTTP **Port** requires 5011 (3011 is disabled in TLS settings), and Protocol must be **TLSv1.2+** (or later).

- Step 3** Select the **Certificate Alias** from the drop-down list and supply its password.

- Step 4** To continue, click **Save**.

When you click **Save** after making any changes, the changes are immediately applied. Often this means your current platform connection closes, and then opens in Workbench.

For example, if you change the **Port** from (default) 5011 to another port number, your reopened platform TLS connection uses the new port, shown in parentheses (nnnn) to indicate that a port other than the default is being used.

As of Niagara 4.13, you select the appropriate certificate alias from the drop-down list and provide the password that is assigned to it. If the selected certificate and the entered password do not match, the framework notifies you that the TLS changes you tried to save failed, and the default certificate is used instead. The default certificate is uniquely generated for each installation and cannot be deleted. This is the message displayed in the **Platform Administration** view.

Platform Administration

View Details

Update Authentication

System Passphrase

Change HTTP Port

Change TLS Settings

Change Date/Time

Change Output Settings

Syslog Configuration

View Daemon Output

Configure Runtime Profiles

Backup

Commissioning

Reboot

Baja Version

Daemon Version

System Home

User Home

Host

Daemon HTTP Port

Daemon HTTPS Port

Host ID

Host ID Status

Model

Product

Serial Number

Local Date

Local Time

Local Time Zone

Operating System

Niagara Runtime

Architecture

Enabled Runtime Profiles

Java Virtual Machine

Niagara Stations Enabled

Number of CPUs

Current CPU Usage

Overall CPU Usage

Filesystem

Physical RAM

Other Parts

Tridium 4.13.0.107

4.13.0.107

C:\niagara\r413\niagara\niagara_home

C:\niagara\r413\niagara\niagara_user_home

My Host:

3011 (disabled in TLS settings)

5011

Win-42F8-AE8A-D800-2658

Perpetual

Workstation

Workstation

None

16-Dec-22

10:24 Eastern Standard Time

America/New_York (-5/-4)

Windows 10 Enterprise (10.0)

nre-core-win-x64 (4.13.0.103)

x64

rt,se,ux,wb

oracle-jre-win-x64-es-dev (Oracle Corporation 1.8.0.351.0)

enabled

20

9%

23%

Total

Free

C:\ 493,342,716 KB 208,895,648 KB

Total

Free

66,772,024 KB 35,886,896 KB

None

Bad password for certificate 'pwd'. Using 'default' instead.

Step 5 If you changed the **Port**, before closing the host (removing it from the Nav tree), carefully note the new secure platform port number. In the future you must specify the port number when making a secure connection to this remote platform.

System date, time and time zone

To ensure accurate historical data, each remote platform's date, time and time zone should be correct and synchronized with the Supervisor PC.

To keep time synchronized across multiple platforms, you can use the **NtpPlatformService** in the **Platform-Services** of the station running on each platform, as appropriate.

Time Zones

Platform configuration of the Niagara host includes specifying its time zone. This affects both real time clock accuracy used in station control, and also how timestamps appear in items like histories and alarms.

A time zone is a region in the world that uses the same standard time, often referred to as the local time. There are many different time zones, owing to the combinations of geographic locations and political/cultural differences. Time zones calculate their local time as an offset from UTC (Coordinated Universal Time). In addition, many time zones apply DST (Daylight Saving Time).

The Java-sourced time zone database has an historical perspective, where a history of changes for applicable time zones are stored. Thus, multiple definitions for a time zone may exist, including past definitions as well as its current definition. The **Time Zone Database Tool** provides access to the historical time zone database on the local host.

This facilitates the display of a station's timestamped data (histories and alarms) collected in time zones under prior rules (typically DST-related). These timestamps display with their original (and correct) collected time.

NOTE: On all Niagara 4 controller platforms, the Java-sourced time zone database is historically accurate only back to the year 2010. Any pre-2010 historical data are displayed using 2010 rules. This was done to improve Java heap usage on these platforms.

However, the Java-sourced time zone database on Windows Niagara 4 platforms extends further back, for example, to the year 1995.

In Workbench, select **Tools→Time Zone Database Tool** to navigate the Java time zone database, where you can explore DST rules for any timezone. If a local station is running on the same host (Supervisor), this is the time zone database used. For more information, refer to the *Getting Started with Niagara*.

Daylight saving time

Where it is used, Daylight Saving Time (DST) maximizes daylight hours during normal waking hours. Many time zones, but not all, use this twice-yearly event to adjust the local time.

The start of DST adds an offset (typically 1 hour) to local time. During this period of the year, local time may be called daylight time. The end of DST removes the DST offset from local time. During this period of the year, local time may be called standard time.

Any time zone using DST has specific rules that define the exact days and times when DST starts and ends. Since DST policies are set by national and regional governments, these rules vary widely from zone to zone. Also, DST policies are subject to change for this same reason—as in the 2007 change for all U.S. time zones that observe DST.

For example, in the 2007 the United States changed the DST start time to the first Sunday on or after the 8th of March (from first Sunday on or after the 1st of April for 2006 and prior years). It changed the DST end time to the first Sunday on or after the 1st of November (from the last Sunday in October for 2006 and prior years).

NOTE: A change in DST rules for a time zone can cause issues in Niagara when displaying historical data (histories and alarm records), particularly when applying new (current) DST rules to records collected using prior (old) DST rules.

Java comes with a time zone database that has an historical perspective. This means that it stores a history of changes for applicable time zones. Thus, multiple definitions for a given time zone may exist including past definitions as well as its current definition. This facilitates the display of a station's time-stamped data (histories and alarms) collected in time zones under prior rules. These are usually DST-related rules. These historical definitions ensure that records display with the original time the data were collected.

In a Niagara 4 JACE controller, the Java source time zone database is historically accurate only back to the year 2010. Any pre-2010 historical data display using 2010 rules. This improves Java heap usage on these platforms.

On a Windows platform used by a Supervisor PC the Java source time zone database extends back to 1995.

To view the time zone DST rules for the station currently running on a host, select **Tools** and click **Time Zone Database Tool**. *Getting Started with Niagara* provides more information about this tool.

For former AX users, Niagara 4 requires no separately maintained `timezones.jar` distributed in the builds, nor associated entries in a platform's `system.properties` file. Instead, time zones are directly sourced from the Java VM (virtual machine) in the host platform. This means there is no requirement to update time zone definitions independently from Java updates that may be included in Niagara 4 updates.

Changing a remote platform's date, time and time zone

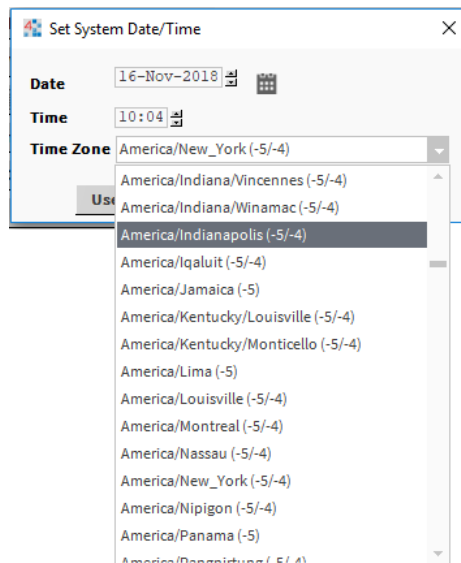
For a Windows host, you use Windows functions to specify the date, time and time zone. On a PC, views that include the time zone as well as TCP/IP configuration are read-only. Usually, you specify time zone in the controller when you run the **Commissioning Wizard**. You can change the time zone at any time using this procedure, which uses the **PlatformAdministration** object. You can also change the date, time and time zone using the **PlatformServices** view in the station.

Prerequisites: You are working in Workbench and are connected to the remote controller platform.

Step 1 Expand the **Platform** node in the Nav tree or double-click **Platform** and double-click **Platform Administration**.

Step 2 Click **Change Date/Time**.

The **Set System Date/Time** window opens.



Step 3 To set the date, do one of the following:

- Click in a day-month-year position, then click the up and down controls or click and type in numerals directly.
- Click the calendar icon for a popup window and select the date from a calendar.

Step 4 To set the time, do one of the following:

- Click in a hour or minute position then click the up and down controls.
- Click and type in numerals directly.

Step 5 To set the time zone, select it from the drop-down list and click **Save**.

The time zones appear on a selection list with a format such as:
Zone ID ($\pm$ hours UTC offset DST, $\pm$ hours UTC offset UST).

For example:

America/Chicago (-6, -5)

Europe/Berlin (+1, +2)

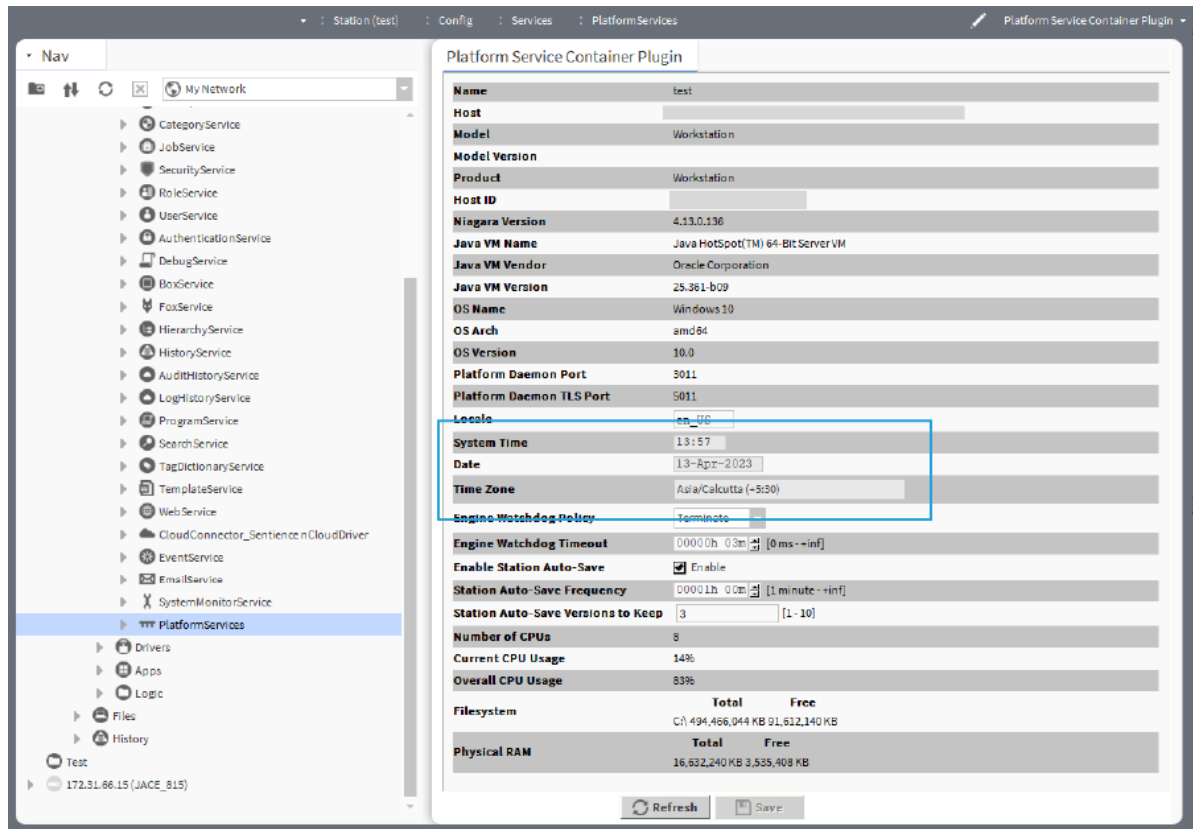
Asia/Tokyo (+9)

There is no DST observance in Japan, so the selection with zone ID Asia/Tokyo shows only the UTC offset of +9 hours. This selection list of time zones is from a historical time zone database.

The system saves the updated date, time and time zone.

- Step 6** To synchronize the remote host's date, time and time zone with your Workbench PC, click the **Use Local** and click **Save**.
- Step 7** To view the current platform's date, time and time zone, expand **Config→Services** and double-click **PlatformServices**.

The time, date and time zone are reported as rows in the **Platform Service Container Plugin** view.



The platform-NtpPlatformServiceQnx component is the Niagara interface to the NTP (Network Time Protocol) daemon of the QNX OS running on a controller. If enabled, this component provides client and server support for NTP.

- Step 8** To view NTP statistics, right-click the **NtpPlatformServiceQnx** node under the controller's **PlatformServices** and select **Views→SpyRemote**.

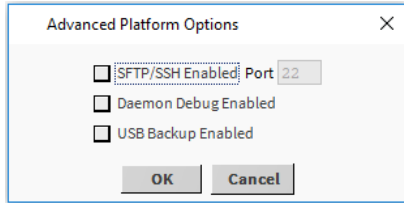
Keep in mind that the ntpd is a QNX process; thus Niagara has no control over what it reports.

Displaying current CPU usage

The QNX Diagnostics Servlet, disabled by default from running in a controller platform provides links to a variety of controller information.

Prerequisites: You are using Workbench running on a PC or laptop and are connected to a remote controller platform.

- Step 1** Double-click **Platform Administration** and click the **Advanced Options** button.
- The **Advanced Platform Options** window opens.



- Step 2 Check the box next to **Daemon Debug Enabled** and click **OK**.
- Step 3 Outside of Workbench, open a browser and connect to: `https://<ipAddress>:5011/qnx` where `<ipAddress>` identifies the remote controller.
The servlet prompts you for the remote host's platform credentials.
- Step 4 Enter the platform credentials and click **OK**.
The servlet displays a list of links.

QNX Diagnostics Servlet

[System information](#)
[pidin](#)
[pidin arg](#)
[pidin env](#)
[pidin family](#)
[pidin fds](#)
[pidin in](#)
[pidin mem](#)
[pidin pmem](#)
[pidin times](#)
[pidin ttimes](#)
[Current CPU usage](#)
[Reset CPU usage](#)
[View current system log](#)
[View historical system log_1](#)
[View historical system log_2](#)
[View network interface controllers](#)
[View network interface parameters](#)
[View network statistics](#)
[Report free disk space](#)
[Report disk usage](#)
[Report file descriptor usage](#)
[View linear flash strings](#)
[Niagara Daemon JMX Info](#)

- Step 5 Click the `Current CPU usage` link.
The servlet displays a text-formatted table.

Wed Sep 30 19:15:18 2020

Last 34d02h of system time in last 34d02h interval: 22 PIDs 67 TIDs

PID/TID	NAME	TIME	CPU%	DELTA	CPU%
1	procnto	32d23h	96%	32d23h	96%
1		0.000	0%	0.000	0%
2		0.000	0%	0.000	0%
3		0.000	0%	0.000	0%
4		0.000	0%	0.000	0%
5		0.000	0%	0.000	0%
6		0.000	0%	0.000	0%
7		5m56s	0%	5m56s	0%
8		7m31s	0%	7m31s	0%
9		0.000	0%	0.000	0%
10		23m00s	0%	23m00s	0%
11		18m32s	0%	18m32s	0%
12		16m00s	0%	16m00s	0%
13		20m35s	0%	20m35s	0%
14		0.131	0%	0.126	0%
15		22m27s	0%	22m27s	0%
16		27m37s	0%	27m37s	0%
-	[child procs]	11.509	0%		
2	proc/boot/random	0.000	0%	0.000	0%
1		5m44s	0%	5m44s	0%
2	Timer Thread	26.767	0%	26.767	0%
3	Syspoll Thread	2.869	0%	2.869	0%
4					
3	proc/boot/pipe	0.000	0%	0.000	0%
1		10m23s	0%	10m23s	0%
2		0.047	0%	0.045	0%
3		10m17s	0%	10m17s	0%
4					
-	[dead threads]	3m22s	0%		
4	proc/boot/devc-seromap	3m18s	0%	3m18s	0%
1					
4101	proc/boot/slogger2	0.039	0%	0.039	0%
1					

At the top of the table is the period of time covered by the table, for example, the Last 9d17h of system time. If this period of time is significant, presumably some threads that used some CPU threads have expired. In this case, some sections of the table may contain a row titled *dead threads*.

Suggestion: reset the accounting and get a new capture after 10 minutes or a few hours.

Step 6 To clear history, click **Reset CPU usage**.

Step 7 To revisit the usage statistics, click **Current CPU usage** after some time has passed.

For information on how to analyze the information provided by this platform diagnostic servlet, reach out to your Technical Support channel when the need arises.

Viewing daemon output

The **Platform Administration** view lets you examine standard output from the host's platform daemon in real time. This output is different from the output of a running station as seen in the **Application Director**. Daemon output is available for troubleshooting purposes in both Supervisor and remote controller platforms.

Prerequisites: You are working in Workbench and are connected to a remote controller platform.

Step 1 Expand the **Platform** node in the Nav tree or double-click **Platform**.

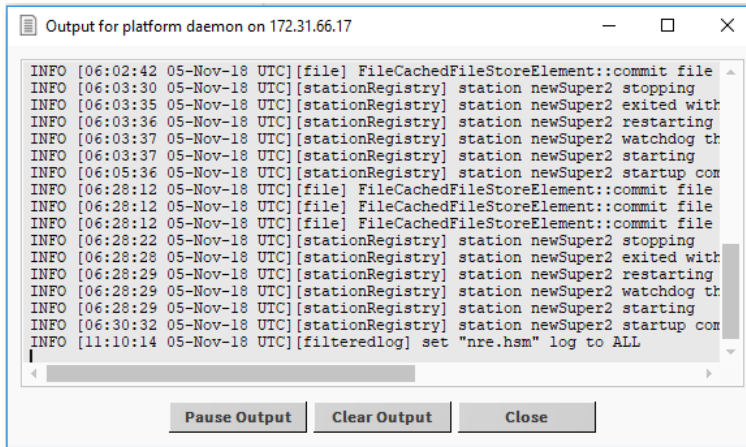
The contents of the Nav Container view opens in the tree or in the main view.

Step 2 Double-click  **Platform Administration**.

The **Platform Administration** view opens.

Step 3 Click **View Daemon Output**.

The **Output for platform daemon on [IP address]** window opens



Depending on the log filter settings set in Platform Administration's **Daemon Output Settings** window, the activity level in the output window varies. Output is non-modal, meaning that you can leave this window open and still do other Workbench operations (including change output settings).

Step 4 As needed, use the scroll bars to navigate through messages, which have headings TRACE, MESSAGE, WARNING, and ERROR. Each message includes a timestamp and a thread ID number.

Step 5 To copy text of interest to the Windows clipboard, use the Windows copy shortcut (**Ctrl + C**).

Step 6 To freeze the output from updating further (no longer in real time), click **Pause Output**.

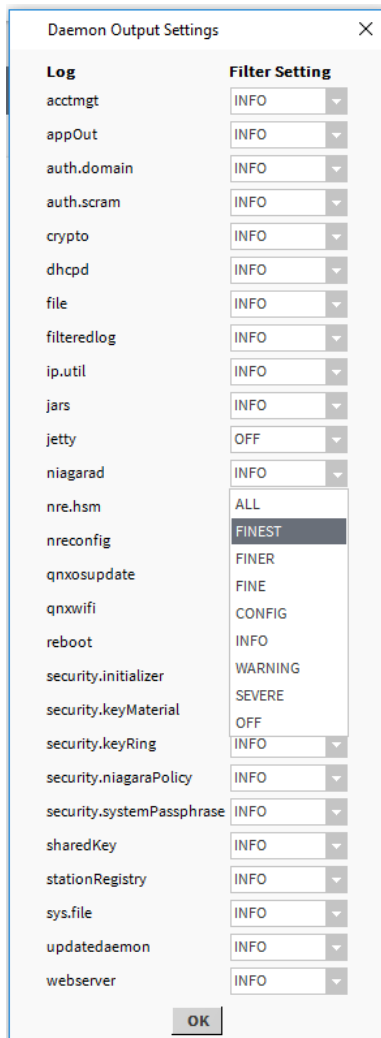
When you freeze the output, the button changes to **Load Output**. This means that daemon messages are still collected. When you click **Load Output**, the display loads the collected messages and continues again in real time.

Step 7 To clear all collected messages from the current daemon output window, click **Clear Output**.

This is not a destructive clear, as another (or new) daemon output window retains the daemon messages.

Step 8 To configure the information included in the daemon output, click **Close** and click **Change Output Settings**.

The **Daemon Output Settings** window opens.



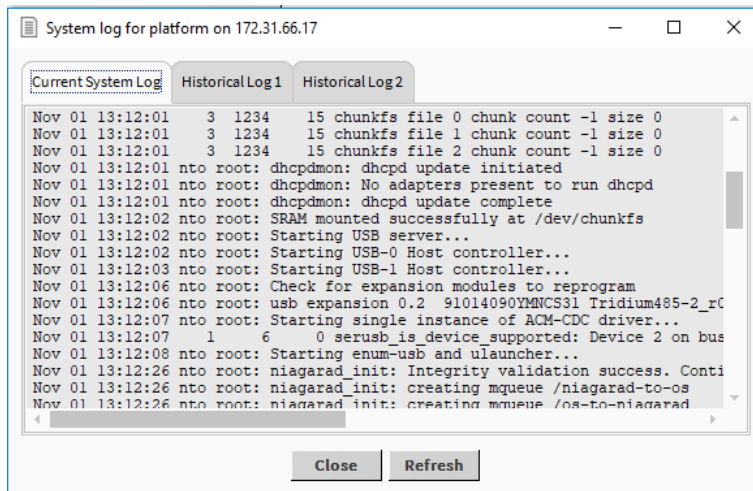
The Windows reference chapter in this guide documents these properties.

Viewing controller system logs

The **Platform Administration** view's **View System Log** button provides a easy and direct method for retrieving system logs for diagnostic purposes.

Prerequisites: You are working in Workbench and are connected to a remote controller platform.

- Step 1 Expand the **Platform** node in the Nav tree or double-click **Platform**.
The contents of the Nav Container view opens in the tree or in the main view.
- Step 2 Double-click  **Platform Administration**.
The **Platform Administration** view opens.
- Step 3 Click **View System Log**.
The **System log for platform <IPaddress>** window opens.



This window contains multiple tabs for viewing various system logs.

Step 4 To refresh, click **Refresh**.

Controller storage conservation

For controllers with limited flash-based file storage, you may conserve storage space by limiting the amount of space each software module requires. You do this by changing the module's runtime profile.

For any Windows-based host (provided it has a hard drive for file storage), you typically want all runtime profiles enabled—including DOC (documentation) modules if it hosts Workbench.

Software modules in Niagara 4 include a suffix on the module's .jar file name that identifies the runtime profile. Some may have multiple runtime profiles. For example, the **alarm** module is distributed as three separate .jar files:

- alarm-rt
- alarm-se
- alarm-wb

The runtime profile describes the contents of each .jar based on what systems are able to use them, where **rt** module .jars are a baseline among all Niagara 4 platforms. Each .jar file is digitally signed. This security measure ensures that the content cannot be changed at commissioning time.

The following table lists the types of software module runtime profile types. Only a very few modules do not have the **-rt** extension. One of those is **baja.jar**. JACE controllers use a Java 8 compact3-compliant VM, whereas Windows-based hosts use the full Java 8 Standard Edition (SE) VM.

Runtime Profile	Example module name	Minimum JRE Version Dependencies	Description
rt	alarm-rt	Java 8 compact3	Module JARs for data modeling and communications. These have core runtime Java classes only, with no user interface. This is the largest runtime profile group.
ux	webchart-ux	Java 8 compact3	Module JARs for BajaUX, any Java classes implementing lightweight HTML5, JavaScript, CSS user interface interaction, also theme modules.
wb	report-wb	Java 8 SE or Java 8 compact3	Module JARs with Java classes for Workbench or Web Launcher user interface; views, field editors, widgets, and so on. Includes Hx and HTML5 Hx views.
se	test-se	Java 8 SE	Module JARs with Java classes that use the full Java 8 Standard Edition (SE) platform API. Currently, these can run on Windows-based hosts only.
doc	platformguide-doc	not applicable	Module JARs without Java code (classes), typically for documentation.

The runtime profile type `rt` is by far the most common of Niagara 4 software .jars. An inventory of module .jar files by type in one Beta build !/modules folder (4.0.11.0) yielded counts of:

- *-rt: 378
- *-ux: 17
- *-wb: 116
- *-se: 6
- *-doc: 20

Where the majority of modules with two runtime profiles had both `rt` and `wb`, with only a few modules having three runtime profiles, as follows:

- alarm: rt, wb, se
- hierarchy: rt, ux, wb
- history: rt, ux, wb
- platCrypto: rt, se, wb
- search: rt, ux, wb
- seriesTransform: rt, ux, wb

Changing a module's runtime profile

Software modules in Niagara 4 are distributed with a runtime profile type, designated by a suffix on each module's .jar file name. In the refactoring of modules, many have multiple runtime profiles.

Prerequisites: You are working in Workbench and are connected to the remote controller platform.

Step 1 Expand the **Platform** node in the Nav tree or double-click **Platform**.

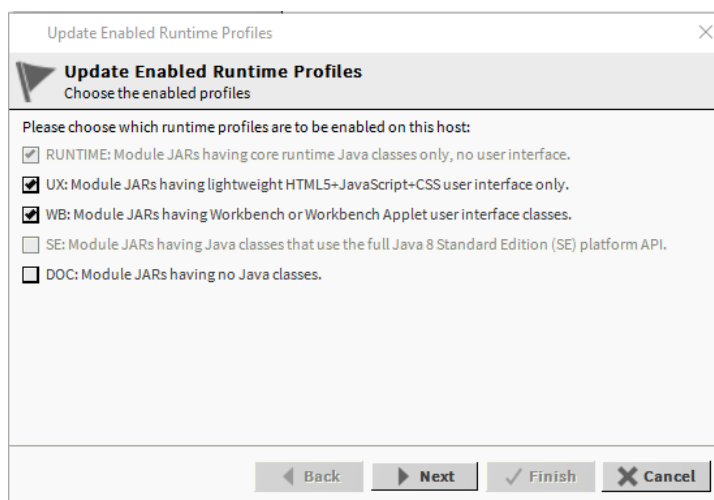
The contents of the Nav Container View opens in the tree or in the main view.

Step 2 Double-click the  **Platform Administration**.

The **Platform Administration** view opens.

Step 3 Click **Configure Runtime Profiles**.

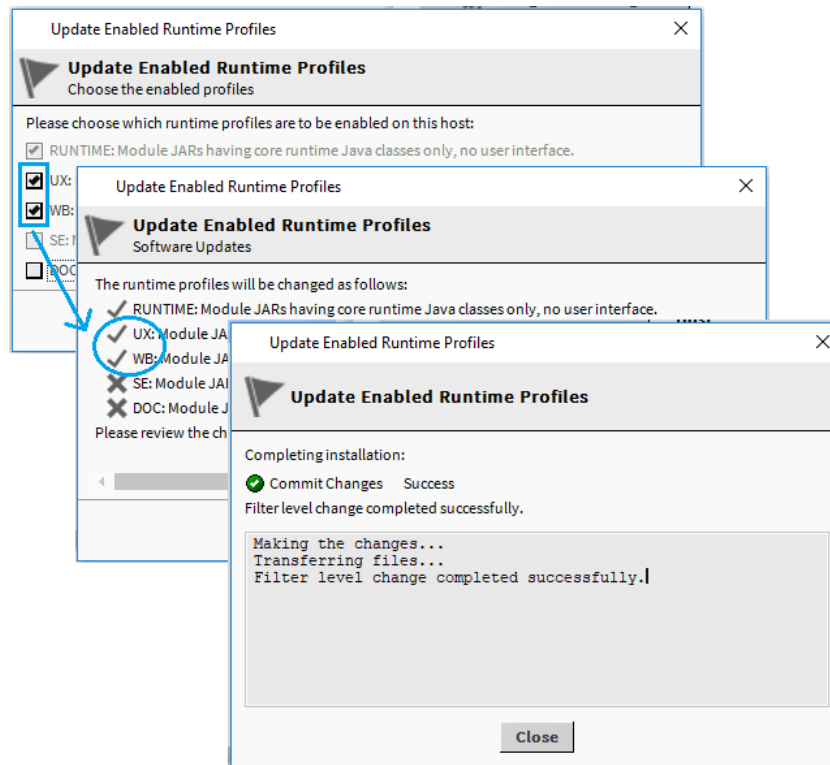
The **Update Enabled Runtime Profiles** wizard opens.



The figure above shows typical settings for controllers in the initial Niagara 4 release. For N4.0, the selection of UX automatically includes WB, and vice-versa. This is likely to change in a future release.

Step 4 Select the profiles to update and remove and click **Next**.

A confirmation window with a **Finish** button lists the added or removed module names.



- If you enabled additional profiles, say, a module goes from `rt` only to `rt`, `ux` and `wb`, additional modules will need to be installed in the controller.
- If you disabled currently enabled runtime profiles, say, from `rt`, `ux` and `wb` to just `rt`, some modules will need to be uninstalled, as they are no longer supported.

NOTE: Niagara 4 does not permit you to disable currently enabled runtime profiles to only `rt` and `ux`.

Step 5 To confirm the configuration, click **Finish**.

The wizard adds or removes the modules.

Step 6 To complete the process, click **Close**.

The wizard stops and restarts a running station.

Configuring memory to improve performance

You can manually configure a controller's NRE (Node Runtime Environment) memory pool settings to improve system performance. Depending on how the station is programmed, you may be able to adjust the allocations. However, there is a fine balance among these memory pool settings. Since there is a finite amount of memory available, increasing one allocation decreases another.

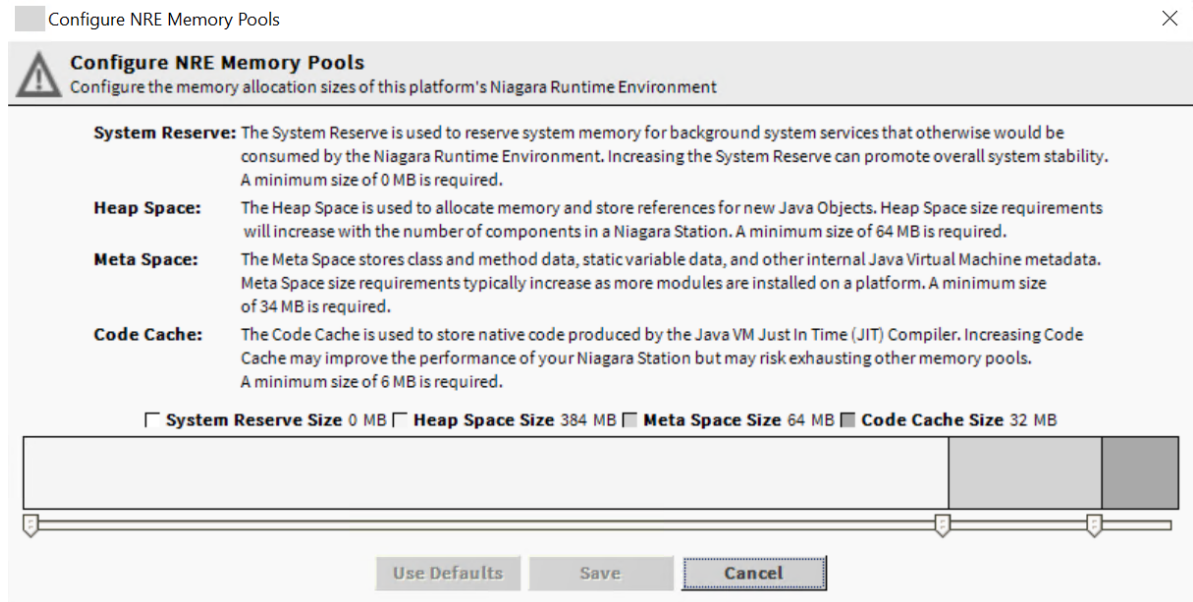
Prerequisites: You are using Workbench and are connected to a remote controller platform.

CAUTION: Configuring a controller with insufficient memory allocations could prevent the station from starting or could cause the station to fail and restart.

Step 1 Expand the **Platform** node in the Nav tree or double-click **Platform**.

The contents of the Nav Container view opens in the tree or in the main view.

- Step 2 Double-click  **Platform Administration**.
The **Platform Administration** view opens.
- Step 3 Click **Configure NRE Memory**.
The **Configure NRE Memory Pools** window opens.



The screen capture shows the default memory allocation values established for Niagara 4.11 and later. The file system writes the alarm and history data directly to the flash memory. This frees up 384MB of RAM to improve performance. The file system reallocates the available space to the Heap Space, Meta Space, and Code Cache memory pools. Any additional memory space, approximately 352 MB, is available as general free memory.

Backing up a station using Platform Administration

Commissioning a controller

Commissioning installs the software on a controller. This procedure works for commissioning a new controller or upgrading an existing controller.

Prerequisites: You are working in Workbench. You are connected to the controller platform.

- Step 1 Double-click the **Platform** node for the controller in the Nav tree.
The **Authentication** window opens.
- Step 2 Enter the credentials you just created when you connected to the platform for the first time, enable **Remember these credentials**, and click **OK**.
The **Nav Container View** opens.
- Step 3 Double-click **Platform Administration** row in the table.
The **Platform Administration** view opens.

Platform Administration

- View Details
- User Accounts
- System Passphrase
- Change HTTP Port
- Change TLS Settings
- Change Date/Time
- Advanced Options
- Change Output Settings
- View Daemon Output
- View System Log
- Configure Runtime Profiles
- Configure NRE Memory
- Commissioning**
- Reboot

Baja Version Tridium 4.5.74.4
Daemon Version 4.5.74.4
System Home /opt/niagara
User Home /home/niagara
Host 172.31.66.10 (J8_Unit10_Fips)
Daemon HTTP Port 3011
Daemon HTTPS Port 5011
Host ID Qnx-TITAN-ATA1-E827-6C6D-843B
Model TITAN
Product JACE-8000
Local Date 14-May-18
Local Time 9:35 Eastern Daylight Time
Local Time Zone America/New_York (-5/-4)
Operating System qnx-jace-n4-titan-am335x-hs (4.5.74.6)
Niagara Runtime nre-core-qnx-armle-v7 (4.5.74.4)
Architecture armle-v7
Enabled Runtime Profiles rt,ux,vwb
Java Virtual Machine oracle-jre-compact3-qnx-arm (Oracle Corporation 1.8.0.141.4.0)
Niagara Stations Enabled enabled
Number of CPUs 1
Current CPU Usage 4%
Overall CPU Usage 7%

Filesystem	Total	Free	Files	Max Files
/	3,492,848 KB	3,294,376 KB	374	109152
/mnt/aram0	393,215 KB	393,017 KB	0	0
/mnt/ram0	8,192 KB	8,131 KB	0	0

Physical RAM

Total	Free
1,048,576 KB	78,984 KB

Step 4 Click the **Commissioning** button.
 The **Commissioning** wizard opens.

Commissioning

This wizard combines steps for configuring a host to run stations. Please check below for each type of configuration change you wish to make:

- ☒ Request or install software licenses
- ☒ Set enabled runtime profiles
- ☒ Install a station from the local computer
- ☐ Install lexicons to support additional languages
- ☒ Install/upgrade modules
- ☒ Install/upgrade core software from distribution files
- ☐ Sync with my local system date and time
- ☒ Configure TCP/IP network settings
- ☐ Configure system passphrase
- ☒ Configure additional platform daemon users

Clear All Check All

Back Next Finish Cancel

This wizard is intended for a remote controller only. This button is not available when connected to any Windows platform. For commissioning details, refer to the appropriate installation and startup guide for your particular controller.

NOTE: If the Workbench FIPS property **Show FIPS Options** is set to `true` certain FIPS options become visible in this window. If selected, the framework enforces FIPS-strength password requirements.

Step 5 Follow the wizard, clicking **Next** until you configure all options.

Step 6 After reviewing all changes, click **Finish**.

Commissioning begins.

CAUTION: Do not interrupt the commissioning process. If you interrupt, you may not be able to restore the station.

After the commissioning process is complete, the controller boots.

Step 7 When prompted, click **Close**.

The wizard installs the software and reboots the controller.

Configuring TCP/IP settings

The TCP/IP Configuration step assigns an IP address to the controller. The TCP/IP object in the **Nav Container View** allows you to review and adjust the platform's TCP/IP settings.

Prerequisites: You are running the Commissioning Wizard or updating the TCP/IP properties after commissioning.

Step 1 At the TCP/IP Configuration step, the **TCP/IP Configuration** view opens.

The screenshot shows the 'TCP/IP Configuration' window. It has a sidebar on the left with 'Interfaces' selected. The main area is divided into two sections: 'General' and 'Interfaces'. The 'General' section includes fields for 'Host Name' (localhost), 'Hosts File' (dropdown), 'Use IPv6' (checkbox, Yes), 'DNS Domain' (text), 'IPv4 Gateway' (172.31.64.1), 'DNSv4 Servers' (list), 'IPv6 Gateway' (text), and 'DNSv6 Servers' (list). The 'Interfaces' section shows 'Interface 1' selected, with details for 'en0' (Onboard Ethernet Adapter en0, Physical Address 50:72:24:F4:6B:00, Adapter Enabled checked). Below this are tabs for 'IPv4 Settings' and 'IPv6 Settings'. The 'IPv4 Settings' tab is active, showing 'DHCPv4' (checkbox, Enabled), 'IPv4 Address' (172.31.66.17), 'IPv4 Subnet Mask' (255.255.252.0), and fields for 'DHCPv4 Server', 'DHCPv4 Lease Granted', and 'DHCPv4 Lease Expires'. At the bottom, there are buttons for 'Undo Changes', 'Back', 'Next', 'Finish', and 'Cancel'.

Step 2 Review the **Interface 1** settings on the **IPv4 Settings** tab, which include the temporary factory-shipped IP address.

Step 3 Assign the controller a unique network IPv4 address.

No other device on this network should use this same IP address. Include the appropriate subnet mask used by the network.

If you are enabling more than one LAN port (applicable to LAN1, LAN2, and WiFi), the IP address for each must be configured on different subnets, otherwise the ports will not function correctly. For example, with a typical Class C subnet mask of 255.255.255.0: setting Interface 1=192.168.1.99 and Interface 2=192.168.1.188 are invalid as both addresses are on the same subnet.

Alternatively, if the network supports DHCP, you can enable it (click **DHCP Enabled**). In this case, the IP Address and Subnet Mask properties become read-only.

In general, for stability, static IP addressing is recommended over DHCP. If DHCP is preferred, an IP Address Reservation should be entered for the controller in the DHCP Server. The controller IP address should not change.

CAUTION: Do not enable DHCP unless you are certain that the network has DHCP servers! Otherwise, the controller may become unreachable over the network.

If the JACE-8000 platform is to be used to wirelessly connect to the enterprise network, do not enable DHCP here. The WiFi adapter in Client Mode requires use of the controller's DHCP feature. For more information, refer to the *JACE-8000 WiFi Guide*

Step 4 Review, and, if needed, adjust other TCP/IP settings, which, in usual order of importance, include:

- **IPv4 Gateway** defines the IP address for the device that forwards packets to other networks or subnets.

The JACE only supports one gateway for all adapters. This includes the JACE-8000 WiFi Adapter in Client mode.

- **DNS Domain Name** provides the name of the network domain, or, if not applicable, leave it blank.
- **DNSv4 Servers** enters the IPv4 address of one or more DNS servers. Click the add button (+) to open the property.
- **Hostname** defaults maybe to "localhost," or you may enter another name to use for this host. If you enter a **Hostname**, typically the name is unique for the domain.
In some installations, changing **Hostname** may result in unintended impacts on the network, depending on how the DHCP or DNS servers are configured. If in doubt, leave **Hostname** at its default setting.
- **Hosts File** opens a TCP/IP hosts file (click the edit property). The format for this file is a standard TCP/IP hosts file, where each line associates a particular IP address with a known host name. Each entry should be on an individual line. Place the IP address in the first column, followed by the corresponding host name. The IP address and the host name should be separated by at least one space.

The **Undo Changes** button resets all settings (all Interfaces) back to the original pre-step values.

Step 5 To add a line, click at the end of the last line and press **Enter**, then type in the required data on the new line.

Step 6 To return to see all TCP/IP settings when you are done, click the control.

This collapses the edit property.

Step 7 To continue, click the **Next** button.

JACE-8000 and JACE-9000 controllers have two Ethernet ports, where **Interface 2** is available for configuring the secondary (LAN2) Ethernet port. By default, this port is disabled, that is, it is without a default address. Its intended usage is for the secondary LAN port as follows:

- Isolating a driver's Ethernet traffic from the primary (LAN1) interface, or
- Creating a private network by daisy chaining multiple IP devices off of the controllers secondary LAN port.

This scenario requires that you configure the LAN2 port as a DHCP server and set up the Access Point WiFi mode.

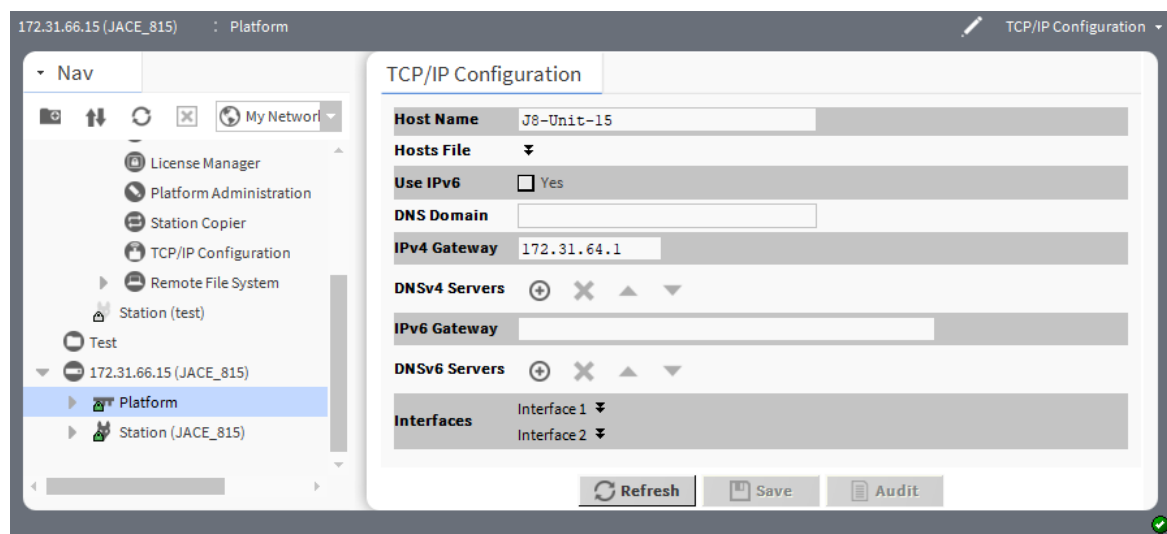
- In some cases, LAN2 may be set up with a standard, fixed, IP address that is used only by a company's service technician, when on site. This allows access to the JACE without disconnecting it from the customer's network, or without connecting the technician's service PC to the customer's network (which might go against local IT security policies).

In any case, only one LAN port can be a DHCP server. If you are enabling LAN2, you must specify another (network) static IP address and the appropriate subnet mask, that is, a different subnet mask for each enabled LAN port IP address.

- The JACE does not provide IP routing or a bridging operation among different Interfaces (LAN ports or WiFi).

Step 8 Once the controller is commissioned, you access the TCP/IP properties by expanding **Platform** and double-clicking TCP/IP Configuration

The **TCP/IP Configuration** view opens.



Upgrading a controller

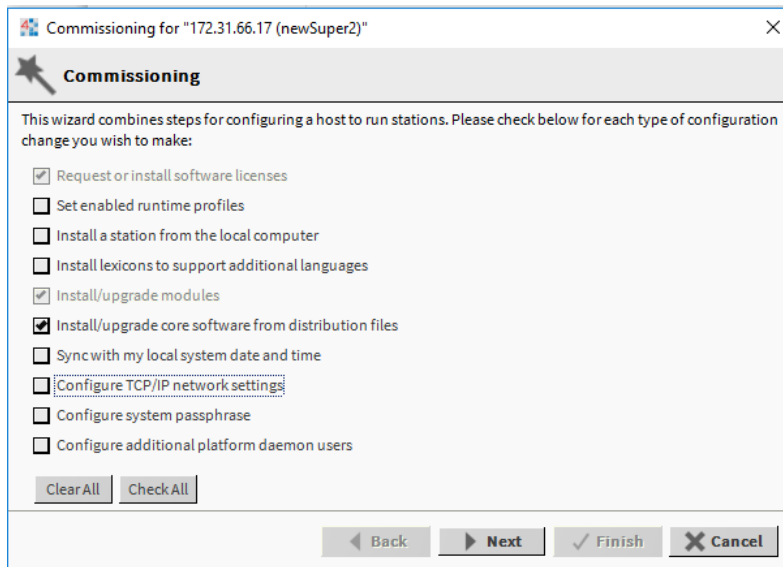
You use the **Commissioning Wizard** to upgrade the software in a controller. This means either an update upgrade from one software build to the next version of that build, or a full minor release upgrade from one build to another build.

Prerequisites: You have upgraded your Supervisor. You purchased a license upgrade in preparation for this upgrade. You are working in Workbench.

Step 1 Make a connection to the controller.

Step 2 To launch, right-click on that **Platform** node and click **Commissioning Wizard**.

The **Commissioning** view opens.



Step 3 Since this is a controller upgrade, in the wizard's opening selection of steps, deselect most items that were previously run at the controller's initial commissioning, for example to set enabled runtime profiles, set date and time, configure TCP/IP settings, and so on.

- Request or install software licenses defaults to being selected.
- If the installation of a station requires commissioning the controller, select `Install station from the local computer`.
- Install/upgrade modules defaults to being selected.
- Install/upgrade core software from distribution files must be selected.

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

The wizard automatically finds and selects all core distributions needed for the controller. Then, in the pre-selected `Install/Upgrade modules` step, the wizard provides the option to also upgrade all out-of-date software modules.

Step 5 At the `Install/Upgrade modules` step select the option to upgrade all out-of-date software modules.

A final summary step allows you to review the upgrade before the wizard executes and performs its operations. For further details, refer to the *JACE Niagara 4 Install and Startup Guide*

Rebooting a controller station

In Niagara 4, reboot is available only for remote platforms. Reboot is always unavailable when connected to any Windows platform, either local or remote.

Prerequisites: You are working in Workbench and are connected to a remote controller. The controller's station is configured for auto-restart.

Step 1 Expand the **Platform** node in the Nav tree or double-click **Platform**.

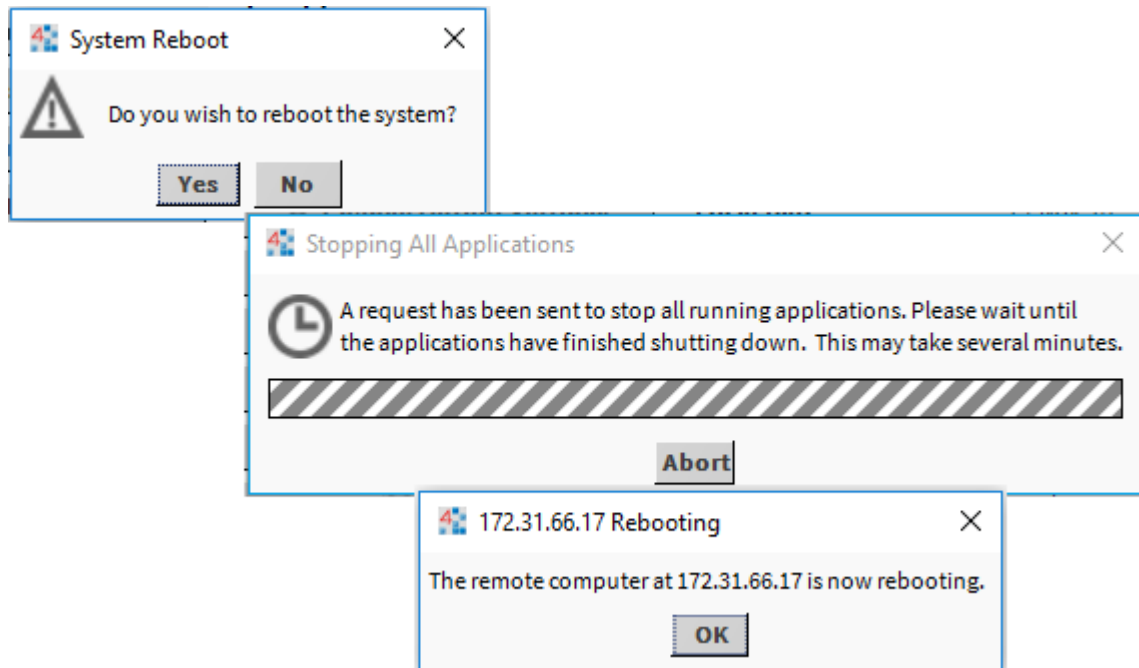
The contents of the Nav Container View opens in the tree or in the main view.

Step 2 Double-click the  **Platform Administration**.

The **Platform Administration** view opens.

Step 3 Click the **Reboot** button.

A message confirms the action.



The daemon drops your Workbench platform connection and attempts to stop any running station before issuing the reboot. A reboot restarts the QNX OS, Java VM, platform daemon, and, finally, the station.

Step 4 Connect to the rebooted station.

Depending on the platform type, it may take from several seconds to a couple of minutes before you can connect to the station again.

Chapter 9 Software Manager

Topics covered in this chapter

- ◆ Viewing the PC's software database
- ◆ Setting up the software database in the PC
- ◆ Installing modules in a remote platform
- ◆ Upgrading out-of-date modules
- ◆ Removing modules from a remote platform

The **Software Manager** keeps track of the modules in the PC and on the remote platform. It facilitates installing, uninstalling and reviewing all software modules installed in a remote platform. This information about the **Software Manager** is summarized here to assist if you are already familiar with previous Workbench versions.

The software module files have separate runtime profiles.

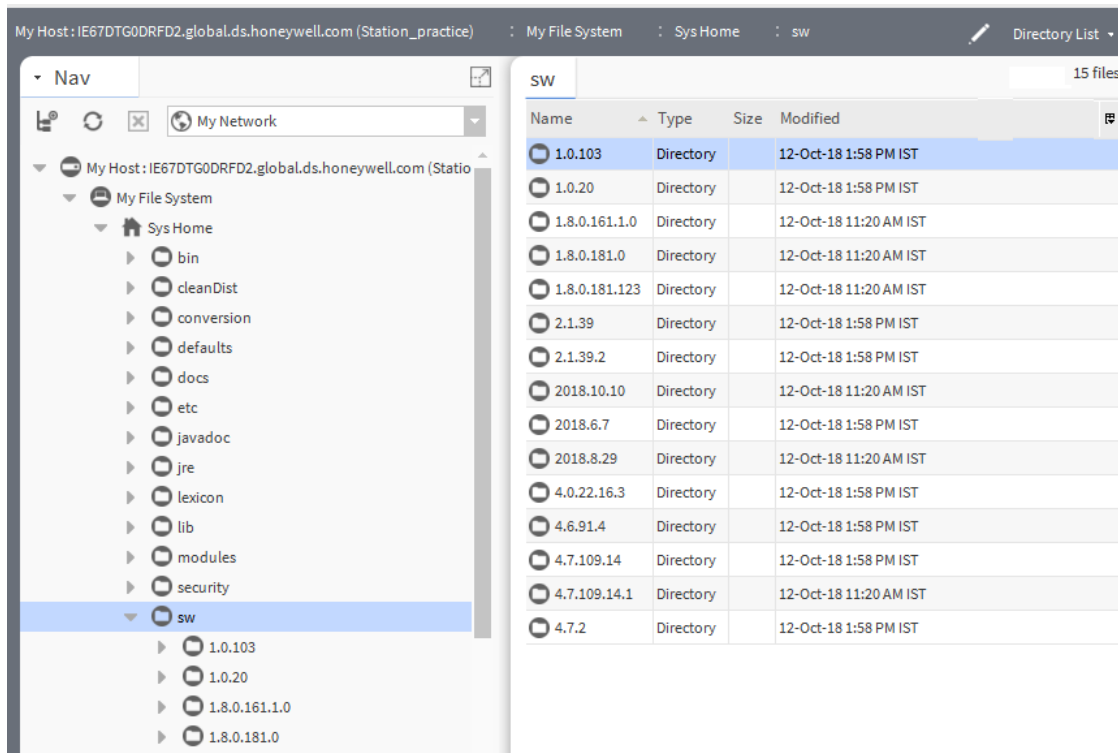
- The **Software Manager** shows only software modules, versus all installable parts including .dist files, etc. The standard lexicons are distributed in Niagara 4 builds as modules, named (by convention) as `niagaraLexiconLc-rt.jar` (where `Lc` is a two-character language code). For details, refer to the *Niagara Lexicon Guide*.
- Module statuses of `Out of Date` and `Not Installed` can include `(Requires Commissioning)`. You cannot install such modules without first commissioning (upgrading) the controller, using the **Commissioning Wizard**.
- In some cases, you can install a new module or modules without rebooting the controller, with its station kept running. This does not apply if you are upgrading (or downgrading) an existing module on the controller.
- If needed, you can install an earlier Niagara 4 version of a module, versus its latest `Available` version—provided the earlier version is in your Workbench's software database.

Viewing the PC's software database

The software database for Workbench is located under the Supervisor PC's **Sys Home** `sw` subdirectory. If Workbench was installed using the `use as an installation tool` option, this directory contains several subdirectories for various distribution (.dist) files, with each subdirectory named by its software version number. While you can see your `sw` subdirectory structure using Windows Explorer, this procedure views the database from within Workbench

Prerequisites: You are working in Workbench and are connected to your Supervisor PC or workstation. The modules to install are available on the PC.

- Step 1 In the Nav tree, expand **My File System**→**Sys Home** and double-click **sw**.
The **Directory List** for your system home opens.



The screen capture is an example of the version number names in an **sw** subdirectory. Your names will vary.

NOTE: To ensure proper operation, do not manually create or rename these subdirectories. Instead, use the **Software Manager** to automatically administer this database.

In the Niagara 4.0 installation (4.0.11.0), shown above, the software database has several versioned subdirectories as follows:

- **1.8.0.0.8** reflects the version of .dist files for the Oracle Java 8 compact3 JRE for controllers: two files, one for PPC processor controllers and one for the ARM processor JACE-8000 or JACE-9000.
- **1.8.0.31.0** reflects the version of .dist files for the Oracle Java 8 Standard Edition JRE for Windows platforms: two files one for 64-bit Windows and one for 32-bit Windows (earlier versions of Niagara).
- **4.0.11.0** reflects the current Niagara release, by build number. It contains numerous Niagara nre config and core .dist files, installed by the Workbench installation option.
- **4.0.25.0** reflects the version of .dist files for the QNX operating system for controllers, with four different .dist files.
- **5.0.1** indicates the version of a few prototype Workbench help modules.
- **inbox** provides a means for you to copy any installable file here, and have the **Software Manager** automatically create a proper versioned subdirectory for it. Or, if the correct subdirectory already exists, the **Software Manager** copies the inbox file(s) there.

Step 2 To view folder details or the details for an individual module, double-click the folder or module name in the Nav tree.

A **Distribution View** of the folder or module opens.

Setting up the software database in the PC

While software modules are usually part of a Niagara installation .zip file, you may receive additional modules separately.

Prerequisites: You are working in Workbench and are connected to your Supervisor PC or workstation. The modules to import are available in another location on the PC.

When receiving updated or new module .jar files, you have two basic options when copying them to your Workbench PC:

- You may copy the modules directly into your `! /modules` folder. This makes the module(s) available to your Workbench environment, and also available to install in other remote platforms (when the installer runs, it copies the module(s) into your software database, making them available for remote platform installation). This is the typical choice.
- You may update the software database by copying the modules or .dist files into your computer's `! /sw/inbox` folder. In this case, your Workbench environment does not use the module(s) themselves, but makes them available in the software database for installation in remote platforms. This would be the choice where you want to keep using a newer (or older) version of the received module(s) in your Workbench environment. A scenario that fits here is if you received older versions of modules, perhaps needed to restore an older backup .dist file in a certain remote platform.

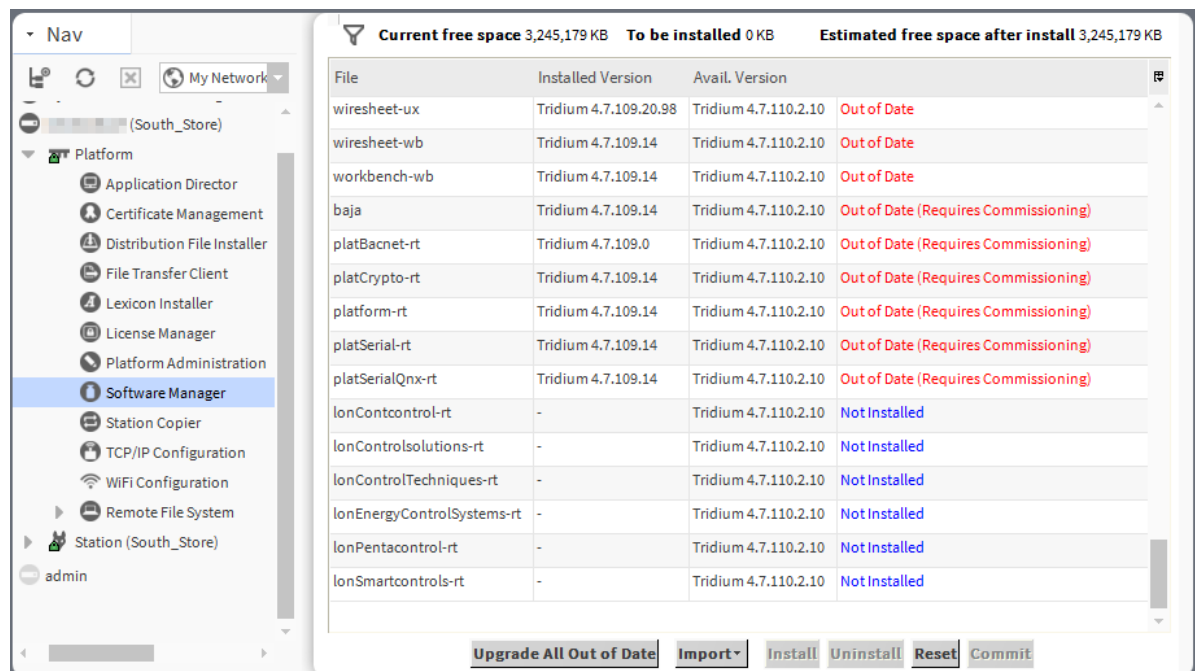
Copying the modules to the computer's `! /sw/inbox` is equivalent to using the **Import** button in the **Software Manager**, which this procedure documents.

Step 1 Expand the **Platform** node in the Nav tree or double-click **Platform**.

The contents of the Nav Container View opens in the tree or in the main view.

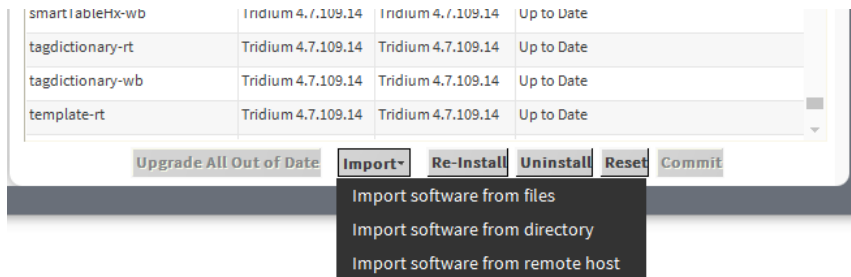
Step 2 Double-click **Software Manager**.

The **Software Manager** view opens.



Step 3 Expand the drop-down list of the **Import** button.

The list provides three options.



- **Import software from files** opens the standard **File Chooser** from which you can navigate to the module location and select one or more software files for import.
- **Import software from directory** the standard **Directory Chooser** with which to navigate to and select a directory for inclusion of any contained software files. For example, you might do this for an earlier installed software build, selecting its **sw** folder or a portion thereof.
- **Import software from remote host** opens the standard **Import from Platform** window in which you can navigate to the location and select one or more software files for import.

Step 4 Select one of the options.

Upon import, the **Software Manager** builds or re-builds the software list. Popup windows open while software files are being copied. Afterwards, any modules that are newer-versioned, or that did not previously exist, are represented in the software table.

If imported modules are earlier versions, they are also available for installation.

When you add different-versioned installable files, the number of different subdirectories under your **sw** directory increase. By default, the **Software Manager** displays only the most recent version of any module as the **Avail. Version**.

Older software files (modules, .dists) are also useful in your software database when restoring a backup .dist for the controller, if the backup was made using a previous software release. You use the platform **Distribution File Installer** to restore a backup.

Installing modules in a remote platform

You may upgrade or downgrade the software in a remote controller.

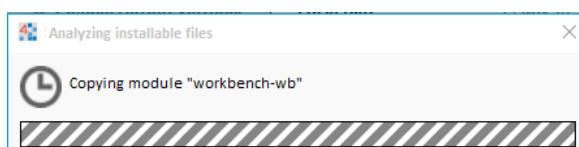
Prerequisites: You are working in Workbench and are connected to a remote platform. The modules to install are available on the PC. After installing the modules, you will be prompted to reboot the controller. Any controlled equipment that might be adversely affected by the controller's station stopping and then host re-booting (from software changes) is put in a manually controlled state.

Step 1 Expand the **Platform** node in the Nav tree or double-click **Platform**.

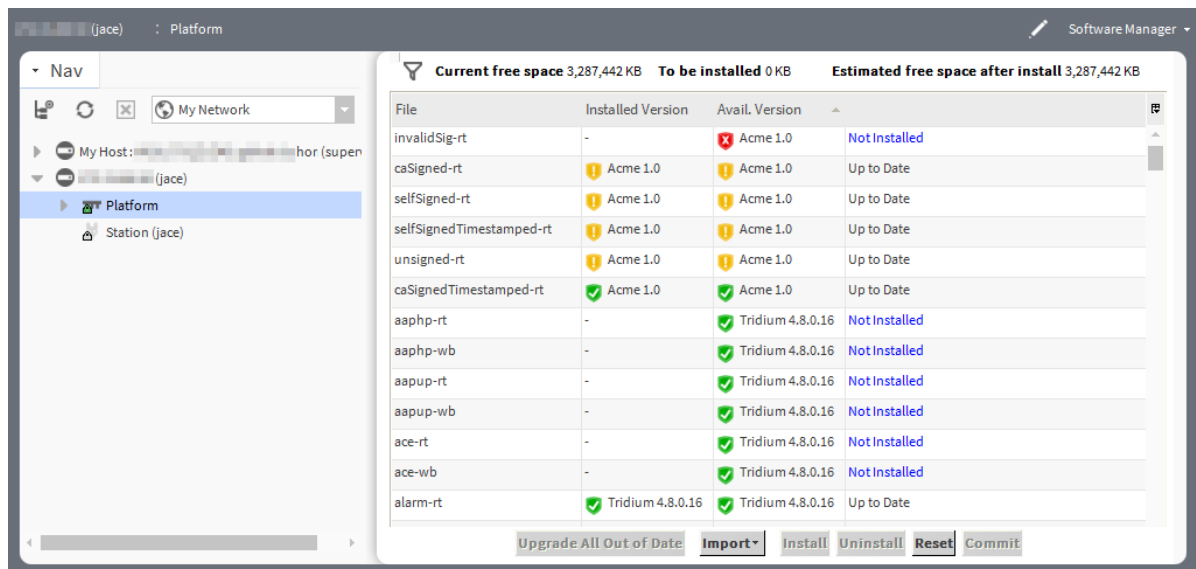
The contents of the Nav Container View opens in the tree or in the main view.

Step 2 Double-click **Software Manager**.

If this is the first time you have accessed the **Software Manager**, it copies modules from your **Sys Home** !/modules folder into a build-named subfolder in your software database (!/sw), for example !/sw/4.13.11.0.



Copying also occurs when you import software into your local software database. Then every time you access the **Software Manager** it rebuilds the modules list, reflecting the latest revision of your available modules, as well modules currently installed in the opened platform.



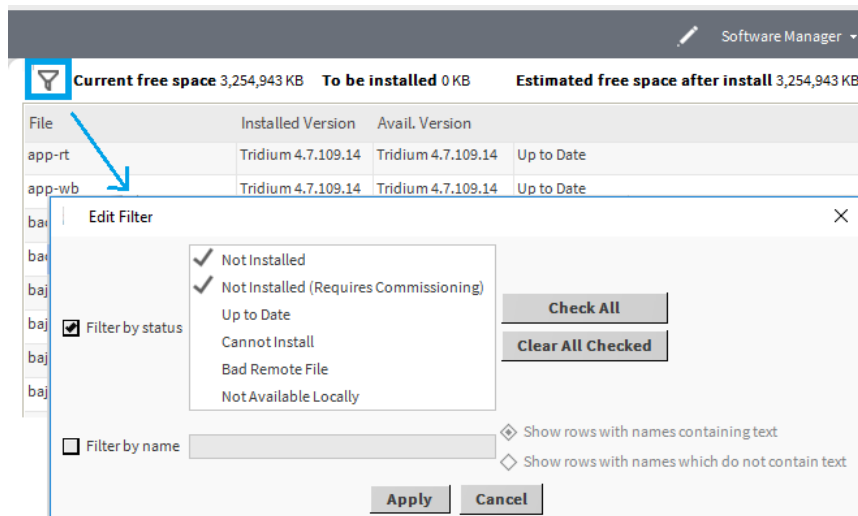
The **Software Manager** lists all the remote platform's out-of-date modules at the top of the table, then lists the uninstalled modules, and lastly the up-to-date modules (sorted alphabetically).

NOTE: The **Software Manager** view and **Commissioning Wizard's Software Installation** step include signature status icons in the Installed Version and Available Version columns indicating the signature status of the installed and available modules. Attempting to install modules with signature warnings (indicated by a yellow  icon) opens a signature warning window, and attempting to install modules with signature errors (indicated by a red ) causes the installation to fail. For details refer to, the *Niagara Third Party Module Signing* guide.

Step 3 To sort the module list alphabetically, click a column title.

Step 4 To reduce the number of modules displayed by the table, click the filter icon () in the upper left corner of the view.

The **Edit Filter** window opens.

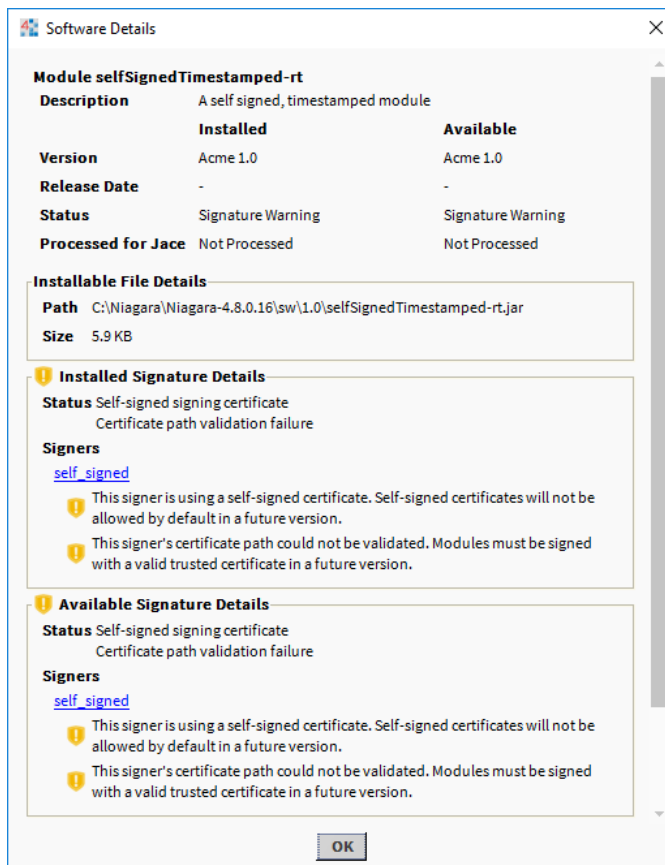


You can use either **Filter by status** or **Filter by name**, or a combination of the two.

Step 5 Select the filter option(s) and click **Apply**.

Step 6 To view details about a specific module, double-click its row in the table.

The **Software Details** window opens.

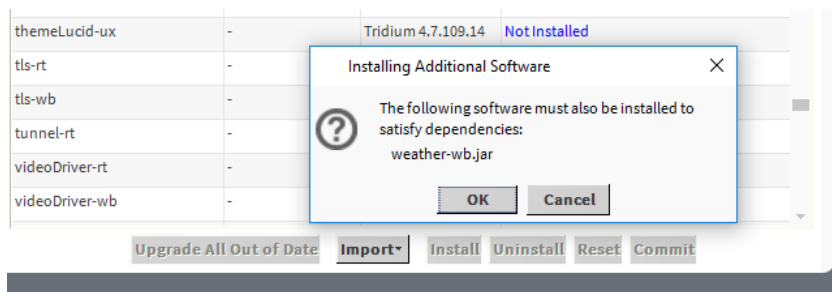


This window includes details about the module's signature status. Also provided is a link to view the certificate that the module is signed with. For more details on signature statuses, refer to the *Niagara Third Party Module Signing* guide.

Step 7 Select one or more modules whose status is **Not Installed**.

The status of the selected modules changes to **Install <version>**. If you select the modules again, the button changes to **Cancel Install**.

If a selected module is dependent on modules that are not already installed (or flagged to install), the Installing Additional Software window opens.

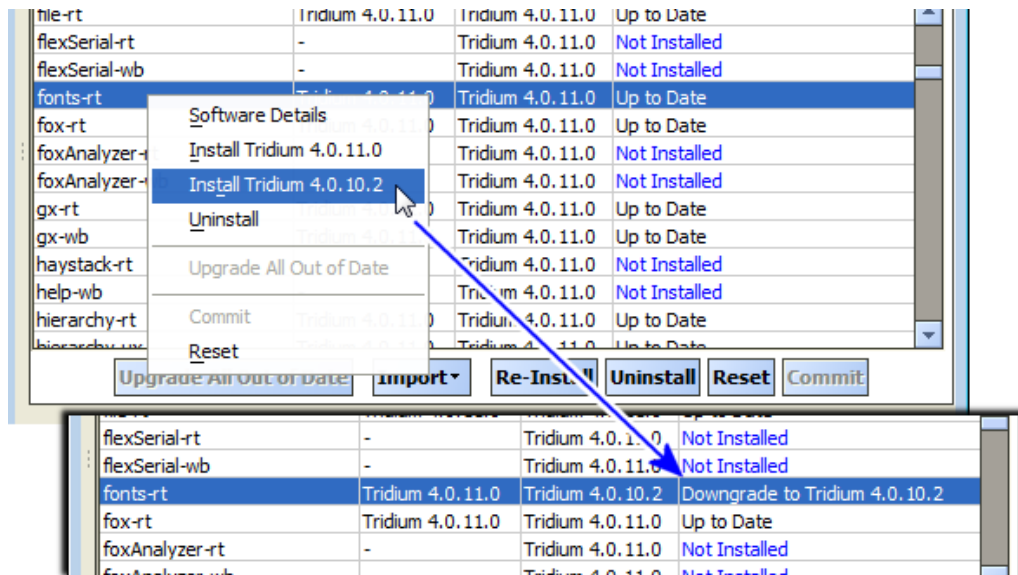


This window explains that additional software is needed.

Step 8 To continue, click **OK**.

The **Software Manager** flags the additional modules and changes the status of all affected modules to **Install <version>**.

- Step 9** To install an earlier version of a specific module, right-click the module and click the earlier version. The earlier version must be available in the Supervisor PC's software database.



- Step 10** After all modules to install or replace are selected, click **Commit**.

When you **Commit**, one of these two things happens after the **Software Manager** copies the module files from the Supervisor PC's software database to the remote controller:

- If you are upgrading or downgrading modules, a confirmation window opens. This window advises you that the station must be stopped and the host rebooted. After the software operation completes, the host reboots.
- In many cases, if you are only installing new module(s), meaning modules not previously installed, the software is immediately installed and the station continues running on the platform.

Upgrading out-of-date modules

Whenever one or more local modules are newer than in the modules in an opened platform, the **Software Manager** enables an **Upgrade All Out of Date** button. This allows you to flag all out-of-date modules to be upgraded. Unlike other action buttons, specific item(s) do not need selection first.

The platform is configured and running.

- Step 1** Open a secure platform connection to a target controller.
- Step 2** Expand the **Platform** container in the Nav tree and double-click the **Software Manager** container.
- The **Software Manager** compares the modules on the Supervisor platform with the modules in each open platform. If a module on the Supervisor side is newer than its equivalent on the controller side, the **Software Manager** enables the **Upgrade All Out of Date** button.

- Step 3** Click the **Upgrade All Out of Date** button.

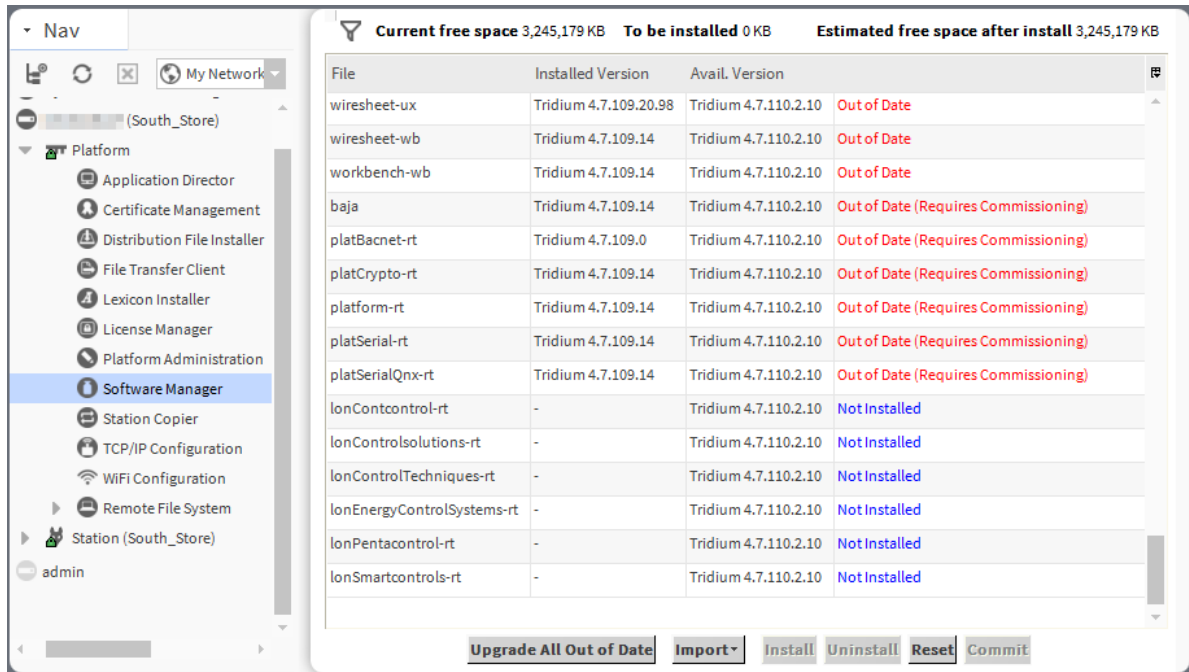
The status of all out-of-date modules changes to **Upgrade to version**, where **version** is the latest version available.

Removing modules from a remote platform

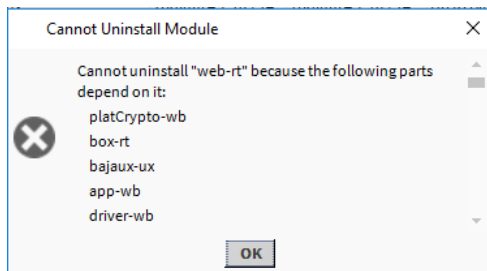
Modules you remove from a remote platform may be up-to-date or out-of-date, but no other modules can be dependent upon them.

Prerequisites: You are working in Workbench and are connected to a remote platform.

- Step 1** Expand the **Platform** node in the Nav tree or double-click **Platform**.
The contents of the Nav Container View opens in the tree or in the main view.
- Step 2** Double-click **Software Manager**.
The **Software Manager** view opens.



- Step 3** Scroll down and select the module(s) to delete.
Module status may be either Up to Date or Out of Date.
If other installed modules are dependent on one or more of the modules you selected, a **Cannot Uninstall Module** window opens.



- Step 4** You can decide to reflag another uninstall, selecting also all modules that are dependent.

Chapter 10 Station Copier

Topics covered in this chapter

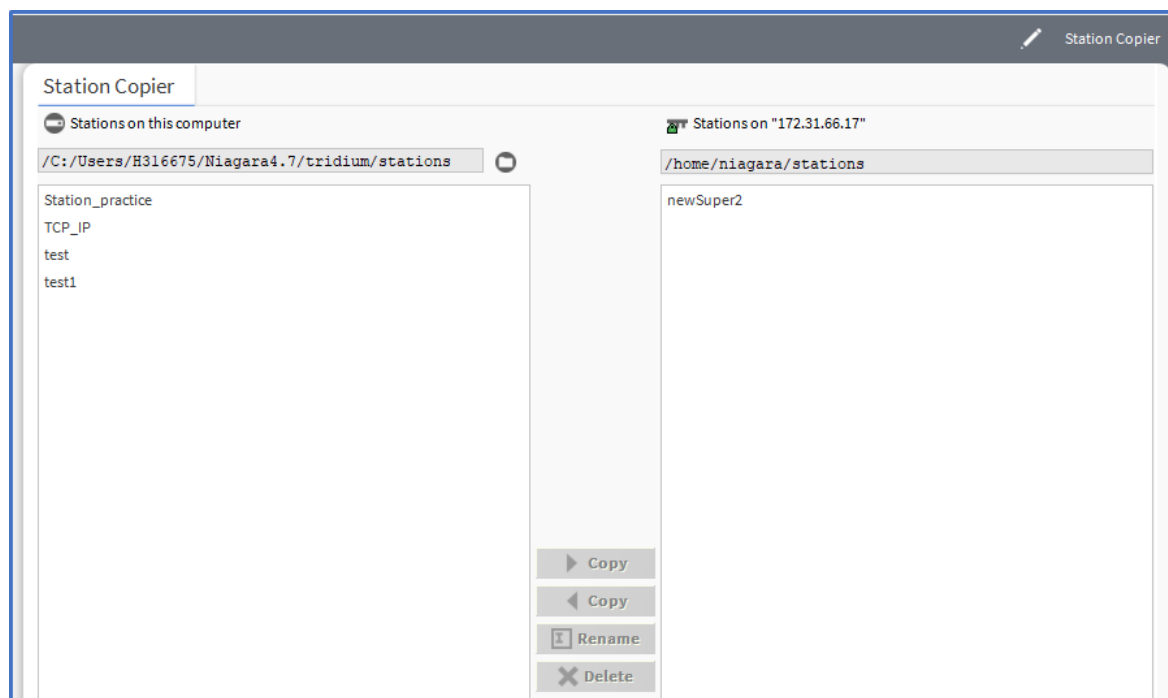
- ◆ Copying a station
- ◆ Station Copier dependencies check
- ◆ Station Transfer Wizard
- ◆ Renaming a station
- ◆ Deleting a station
- ◆ Station installation troubleshooting

The **Station Copier** is one of several platform views. You use it to install a station in any Niagara 4 platform (remote or local), as well as make a copy in your user home of any running station (remote or local). You can also use **Station Copier** to rename and delete stations, either locally or remotely.

NOTE: In Niagara, there is added support for verifying third-party module signatures. When installing a station that requires additional dependencies to be installed using the **Station Copier**, any module signature warnings for the dependencies are displayed in a **Signature Warning** window, and any module signature errors cause the station copy to fail. For more details, see *Niagara Third Party Module Signing*.

You see the **Station Copier** view even when opening a local platform connection at your Supervisor computer as well as when opening a remote Niagara host. The following figure shows the **Station Copier** in a platform connection to a controller.

Figure 20 Example Station Copier view for remote platform



As shown above, the **Station Copier** view is split into two main areas:

- Stations on your Workbench PC, typically your User Home (left)
- Station in the daemon User Home of the opened platform (right)

By default, contents of your User Home stations folder is shown on the left side. If you have station folders located elsewhere, click the folder icon for a **Change Directory** window, and point the **Station Copier** there. That changed location is used the next time you access the **Station Copier**.

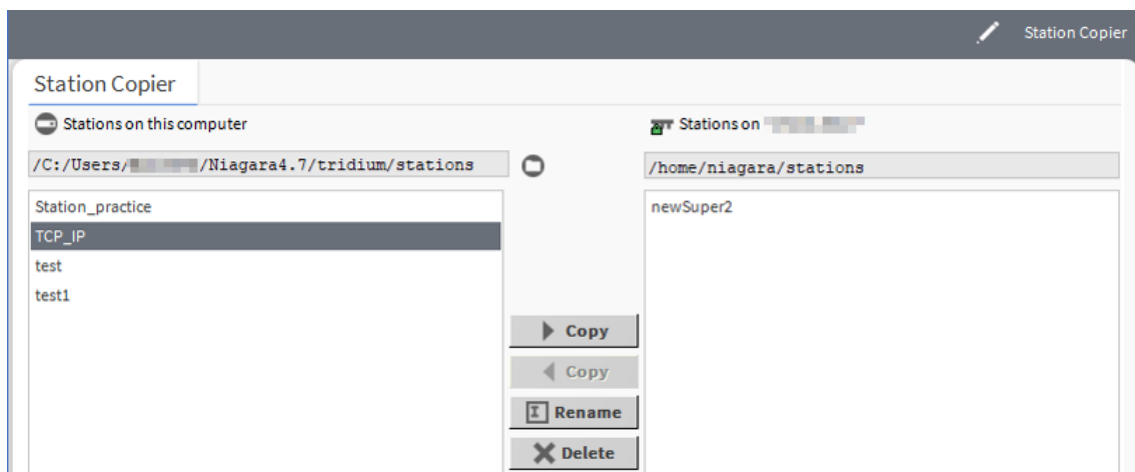
Copying a station

Station installation usually involves copying an existing station from the Workbench user home to a target controller platform. The procedure uses the **Station Transfer Wizard**.

Prerequisites: All hardware (PC or controller) has been installed and connected. The controller has been commissioned and network communication configured. The station you wish to install exists in the Workbench user home.

This topic is part of configuring a station for the first time or upgrading a station from a previous version of Niagara.

- Step 1 Open a secure platform connection to the target controller, the one on which you wish to install the station.
- Step 2 Expand the **Platform** container in the Nav tree and double-click the **System Copier**.



The copier works in either direction.

- Step 3 If the station is located elsewhere, click the folder icon (📁) for a **Change Directory** window, and point the **Station Copier** there. The **Station Copier** uses this location the next time you access it.
- Step 4 Click to select a station on one side (to copy to the other side) and click the **Copy** button that points to the other side of the view.

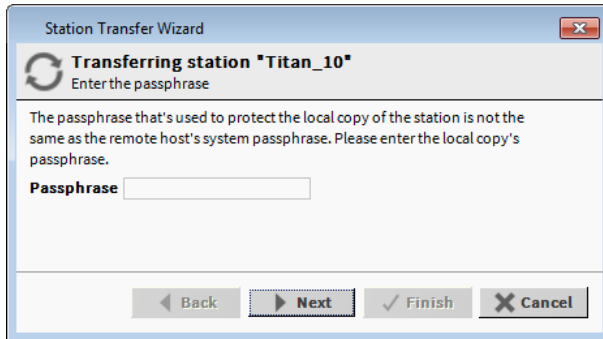
When you click a station, the station is selected (highlighted) and the appropriate **Copy** button by direction becomes available. This clarifies the source and target locations.

Clicking in right side for a copy from the daemon User Home to the Workbench User Home makes a local copy of a station, saved to your Workbench computer. This is described as a backup.

The Station Copier displays "Loading module information" and, if all needed modules are available, launches the **Station Transfer Wizard**.

If any module needed by the station has a dependency that requires the controller (platform) to be commissioned (commissioning upgrades core software or the operating system), the station installation stops immediately, displays a message, and provides the option to start the **Commissioning Wizard** instead. The need to commission a controller arises if you are installing a brand new controller or upgrading a controller from an old version to new version and forgot to run the **Commissioning Wizard**.

When you click **Copy**, the **Station Transfer Wizard** attempts to validate the BOG file's passphrase with the target host system's passphrase. If they are the same, the process continues without prompting for a passphrase. If they are different, the wizard prompts for the file passphrase.



If a BOG file is protected with an unknown passphrase, you can use the Workbench toolbar icon to unlock (force-remove) the passphrase, making the file unprotected, or “force-change” the passphrase to enter a new value. When you choose either of these options, any sensitive data in the file are cleared.

CAUTION: Be aware that unlocking (force-remove) and changing (force-change) the passphrase on a BOG file results in the loss of sensitive data in the file.

Step 5 Follow the wizard prompts clicking **Next** or **Back** as necessary.

When copying from Workbench to a remote platform, not all types of files can be transferred. For example, the copier does not allow alarm and history data to be transferred. This is true even if you choose the option to `Copy every file in the station directory and its subdirectories`. And, if you choose the option to `Copy files from selected directories`, it does not allow you to select any alarm or history folder or subfolder. To include such data, perform a backup/restore operation instead, that is back up the station to a *.dist file making sure to edit the default backup settings to not exclude (include) alarms and history data.

If the station is running, the wizard informs that it will stop and restart the station.

Step 6 Review all the changes you selected and click **Finish**.

The wizard displays installation progress in the **Transferring station** window.

Step 7 To complete the operation, click **Close**.

The Station Copier prompts you to open the Application Director now.

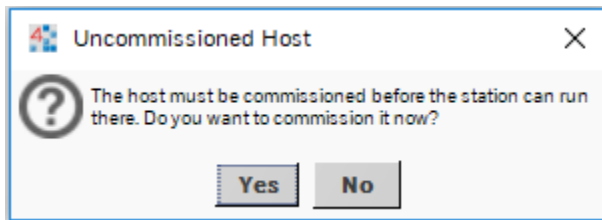
Step 8 To view the station log, click **Yes**.

Station Copier dependencies check

The **Station Copier** checks, whenever installing a station, to determine if the target controller platform does not already have all modules installed that are required by that station. Such dependencies may prevent the installation of a selected station. Changes are summarized as follows:

If any module needed by the station has a dependency that requires the controller to be commissioned (upgrade core Niagara software or QNX OS), the station install immediately stops, upon station selection. Steps in the **Station Transfer Wizard** do not appear. A dialog explains the controller needs commissioning, and provides the option to start the **Commissioning Wizard**, as shown here.

Figure 21 Uncommissioned Host window



Click **Yes** to start the **Commissioning Wizard**, or **No** to simply return to the Station Copier.

This may occur if are trying to install a station in a new, uncommissioned controller. Despite documentation to first commission any new controller using the platform **Commissioning Wizard**, this continues to occasionally come up. For complete details, see the *JACE Niagara 4 Install and Startup Guide*.

If all modules needed by the station are found on your PC, the **Station Transfer Wizard** starts normally. However, upon reaching the “Modules step”, in some cases you may see a caution.

Station Transfer Wizard

This wizard assists with any station copy (installing or backing up) by presenting a number of steps. The exact steps vary by the direction of copy, as well your selections in wizard step dialogs.

The wizard buttons control progress:

- **Next** advances to the next step.
- **Back** returns to a previous step.
- **Cancel** exits from the wizard without copying the station. If the source station config.bog file is locked, the wizard opens in a state where you must cancel. No other option is available. A station is locked if:
 - The source config.bog contains unsaved changes, that is, it is being edited elsewhere in Workbench. After saving the changes, you can try to copy again.
 - The station source config.bog is currently in the process of being saved. Try the copy again later.
- **Finish** is enabled only in the final step. When you click **Finish**, the copy begins and you see progress updates in the **Transferring Station** window.
- **Close** exits the wizard when the copy is complete.

Transferring a station to a controller

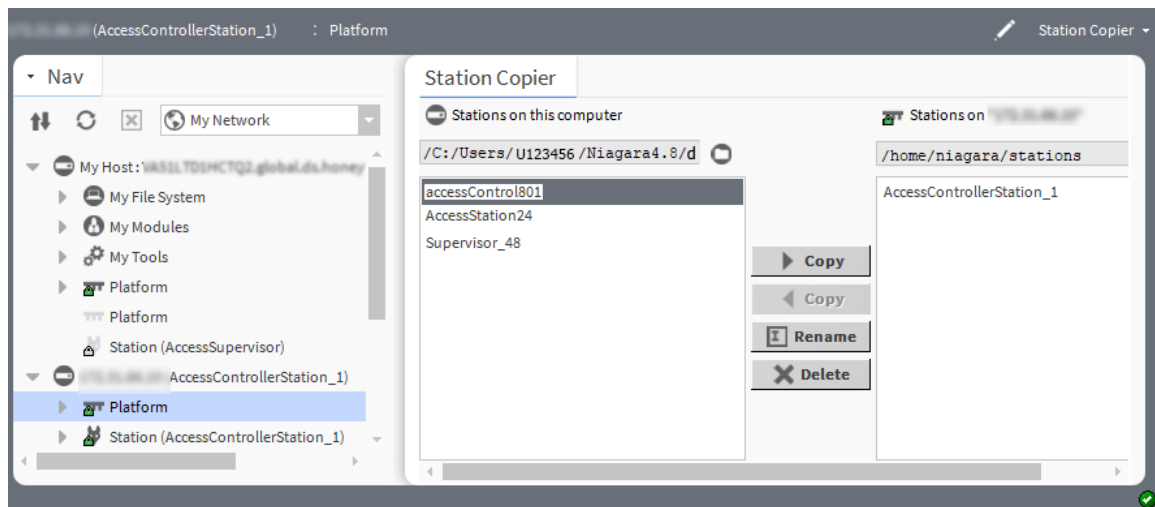
The Station Transfer Wizard assists with any station copy (installing or backing up) by presenting a number of steps. The exact steps vary by the direction of copy, as well your selections in wizard windows. In each step, click **Next** to advance to the next step. As needed, click **Back** to return to a previous step and make changes, or click **Cancel** to exit from the wizard (no station copy performed).

Prerequisites: You are working in Workbench running on a PC and are connected to the remote station.

Step 1 Stop the remote station.

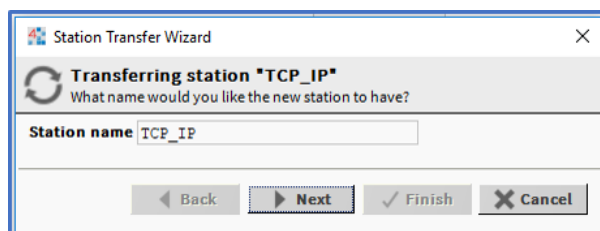
Step 2 In the PC, expand **Platform** and double-click **Station Copier**.

The **Station Copier** view opens.



Step 3 To copy the source station BOG file from the computer to the target remote controller, select the station to copy in the left pane and click Copy ().

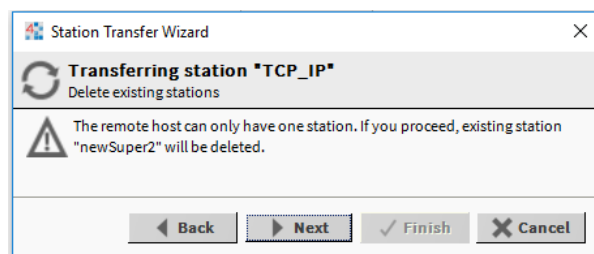
The **Station Transfer Wizard** opens with a prompt to enter or confirm the station name.



Default name is the station directory being copied. If you rename the station, it will be identical to the source (copied) station in every way except for the name of the station directory.

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

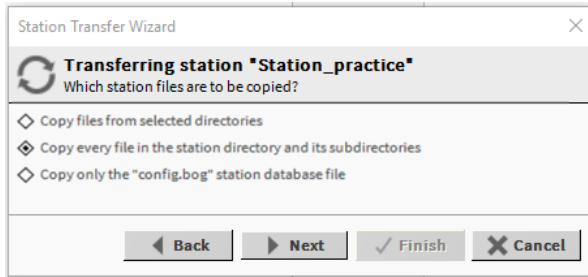
If the target controller already has a station installed whose name is different from the name of the source station, the wizard prompts you with a message.



The wizard skips this message if no station exists on the target controller or the names of the source and target stations are the same. Otherwise, it deletes the entire remote station directory (all subdirectories and files) when the station installation starts. If you are unsure, it may be best to **Cancel**, then backup the remote controller's station first before copying the new station to the controller.

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

If the source station consists of more than a config.bog file, the wizard prompts you to select the station files to copy.



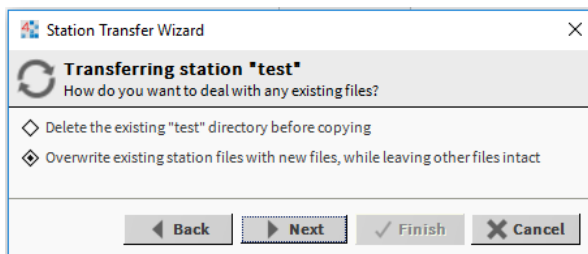
This selection defaults to copy all files and folders under the station directory.

When copying from Workbench to a remote platform, the copier transfers only some files. For example, the copier does not allow alarm and history data to be transferred. This is true even if you choose **Copy every file in the station directory and its subdirectories**. And if you choose **Copy files from selected directories**, the wizard does not allow you to select any alarm or history folder or subfolder. To include such data, perform a backup/restore operation instead, that is, back up the station to a *.dist file making sure to edit default backup settings to not exclude alarms/history data.

- **Copy files from selected directories** is not shown if the source station has no subdirectories.
- **Copy every file in the station directory and its subdirectories** copies all files.
- **Copy only the "config.bog" station database file** limits the copy to only the config.bog file.

Step 6 Make your choice and click **Next**.

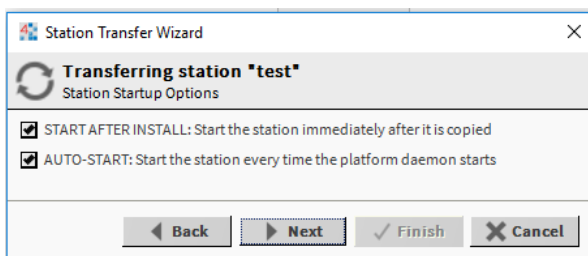
If an identically-named target station already exists, the wizard prompts you to choose what to do with it.



If you previously selected to copy everything, this step defaults to **Delete existing station directory before copying**. Otherwise, this step defaults to the overwrite option.

Step 7 Make your choice and click **Next**.

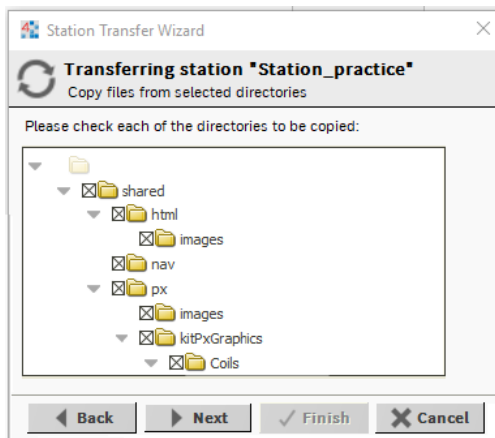
The **Station Startup Options** window opens.



Auto-Start is one of two settings for any station as specified in the **Application Director** view. Typically, you enable both settings.

Step 8 To continue, click **Next**.

If you selected to copy selected directories, the copy files window opens.

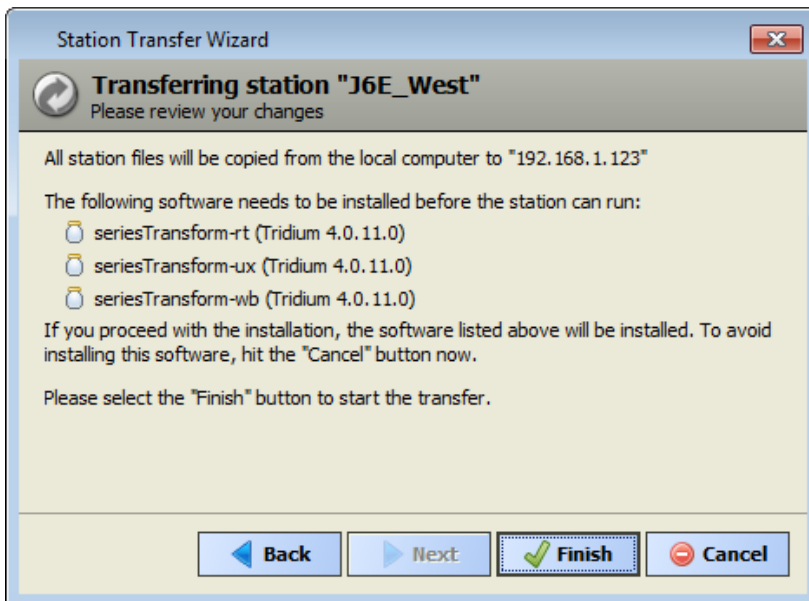


This step provides a tree to select station subdirectories (folders) to include in the copy. By default, all selectable folders are both expanded and selected, while unselectable folders are not. If present, you cannot select a station's `alarm` and `history` folders.

Step 9 Click to deselect any folder and click **Next**.

The wizard skips this step if all required modules are already in the controller.

If the target platform is missing one or more of the modules required by the station being copied (installed), the wizard lists the missing modules and versions to be installed during the station copy operation.



The Station Copier compares any missing modules against the software that is already installed in the target platform and looks in your User Home software database for versions of the missing modules that can be installed without re-commissioning the target platform.

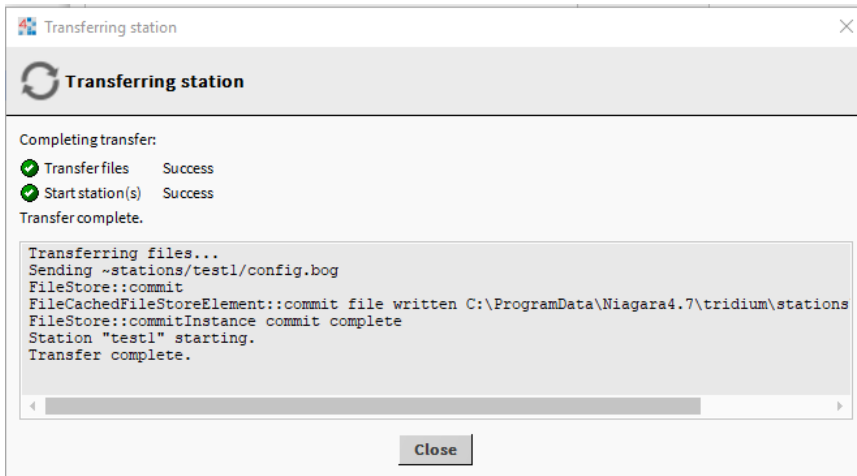
There are two possible results when the wizard reaches this step:

- Station can be installed with most current modules. If all missing modules can be installed using the most current versions, they list without any warning.
- Station can be installed with out-of-date modules. If any module to be installed is not the most current version, you have the option to cancel the copy.

Step 10 Do one of the following:

- To continue click **Finish**.
- To terminate the copy, click **Cancel**.

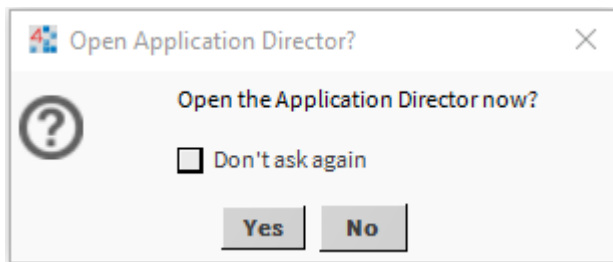
If you clicked **Finish**, the local-to-remote copy starts including installation of the station and listed modules. The **Transferring Station** window reports progress.



If you clicked **Cancel**, The **Station Transfer Wizard** closes. Then, either select another station to install, or, if upgrading the controller is possible and you have purchased an upgrade license for it, run the **Commissioning Wizard**. This will also install a station on the controller.

Step 11 To complete the process, click **Close**.

The wizard asks if you wish to switch to the **Application Director**.



It is a good idea to observe a station's output upon first startup.

Step 12 To automatically switch to the **Application Director** after installing a station, click the check box to **Don't ask again**, then click **Yes**.

If you selected Auto-Start in the wizard, the station starts automatically and you can watch its output.

Backing up a station using Station Copier

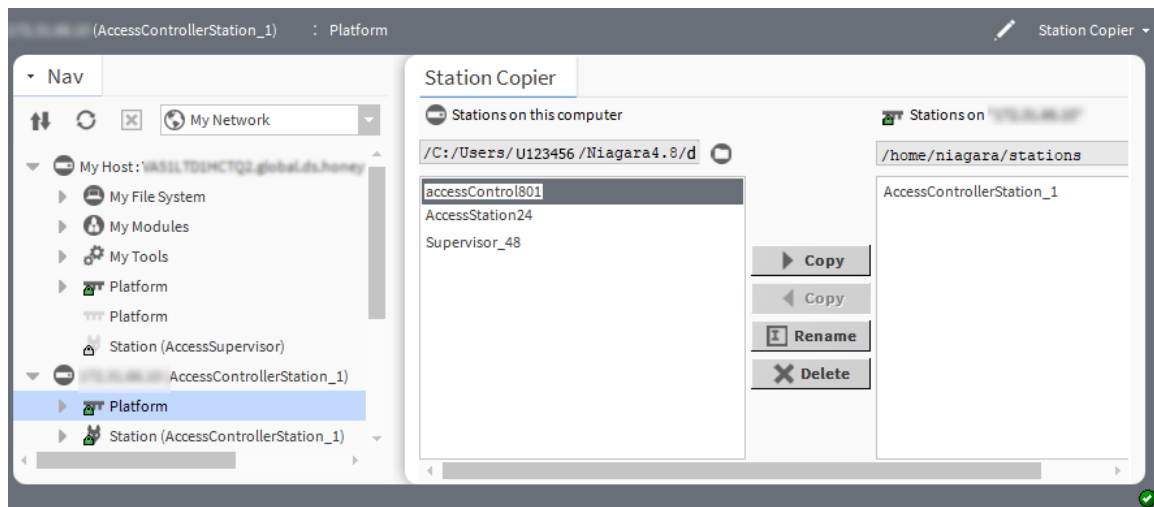
A distribution backup (.dist) depends on platform files, which makes it inappropriate when upgrading to a different model host. The Station Copier can back up all the files and subdirectories in a station so that you can restore the backed-up station to a different model host platform. In addition, Station Copier can back up just the config.bog, or just a single folder. Station copies do not contain encryption keys or software dependency information. This procedure works for both Supervisor and remote controller stations.

Prerequisites: You are using Workbench running on a PC that is connected to the network.

For Niagara Enterprise Security customers, the web UI does not support the Station Copier.

- Step 1 Open a connection to the remote platform.
- Step 2 Expand the **Platform** node in the Nav tree and double-click **Station Copier** or double-click the **Platform** node, and double-click **Station Copier**.

The **Station Copier** view opens.



- Step 3 Click the File icon (📁) on the **Stations on this computer** pane (left pane) and select the folder in which to store the station copy.

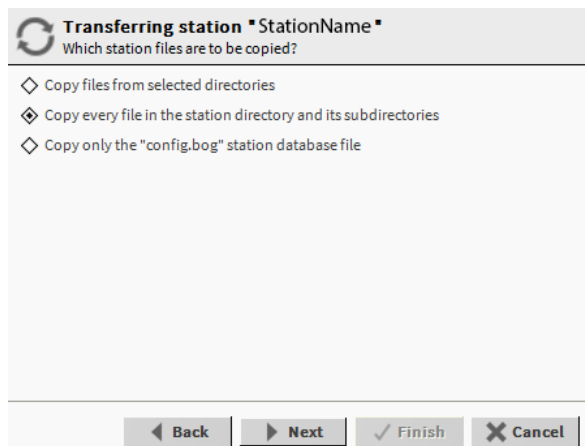
CAUTION: If you are copying only a `security` folder from the remote to your PC, do not overwrite the `security` folder in your PC. This folder contains encryption keys, which you will need in the new controller to decrypt user passwords.

- Step 4 Select the station or folder and click **Copy**.

The **Loading Module Information** window opens followed by the **Station Transfer Wizard**.

- Step 5 Click **Next**.

The wizard prompts you to select what to copy.



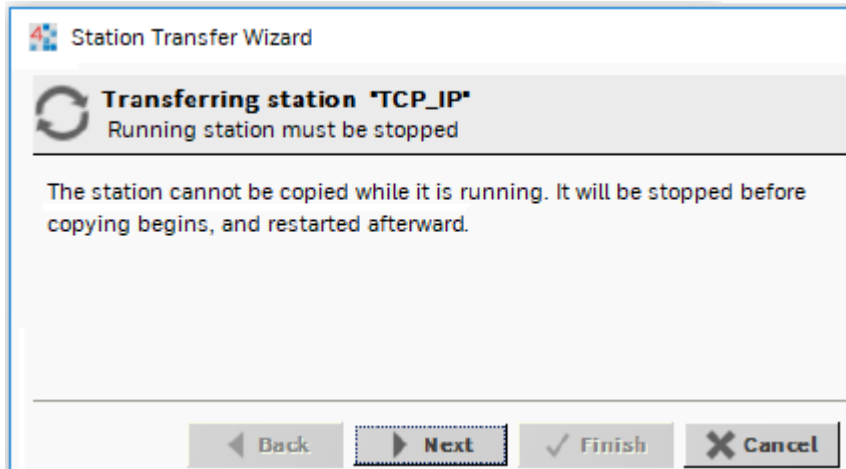
This option defaults to `Copy every file in the station directory and its subdirectories`. While Station Copier can back up just the `config.bog`, or just a single folder, it is recommended that you always back up every file in the station directory and its subdirectories.

- Step 6 Make a selection, and click **Next**.

If a station with the same name exists in the target location, the wizard prompts you to delete or overwrite the existing station.

Step 7 Accept the default (delete), and click **Next**.

The wizard reminds you that the station must be stopped before it can be copied.



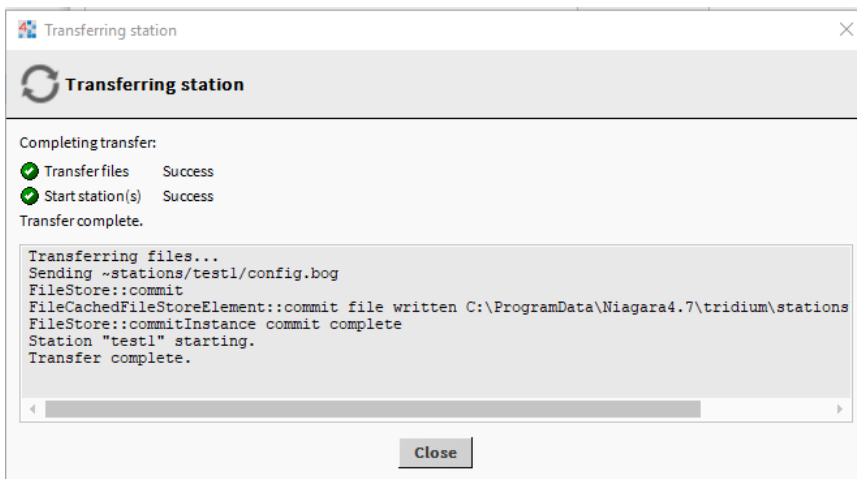
Step 8 To continue, click **Next**.

The wizard asks you to review the copy configuration (changes).

If you selected only specific station subdirectories to copy, they are listed.

Step 9 If needed, click **Back** and make changes, or, to complete the wizard, click **Finish**.

The Station Copier saves the station, if the remote station is currently running, begins the copy process and reports transfer status in the **Transferring Station** window.



The Station Copier saves the station if the remote station is currently running and

Step 10 When the save completes, click **Close**.

The date for all copied files reflects when the files were copied.

Renaming a station

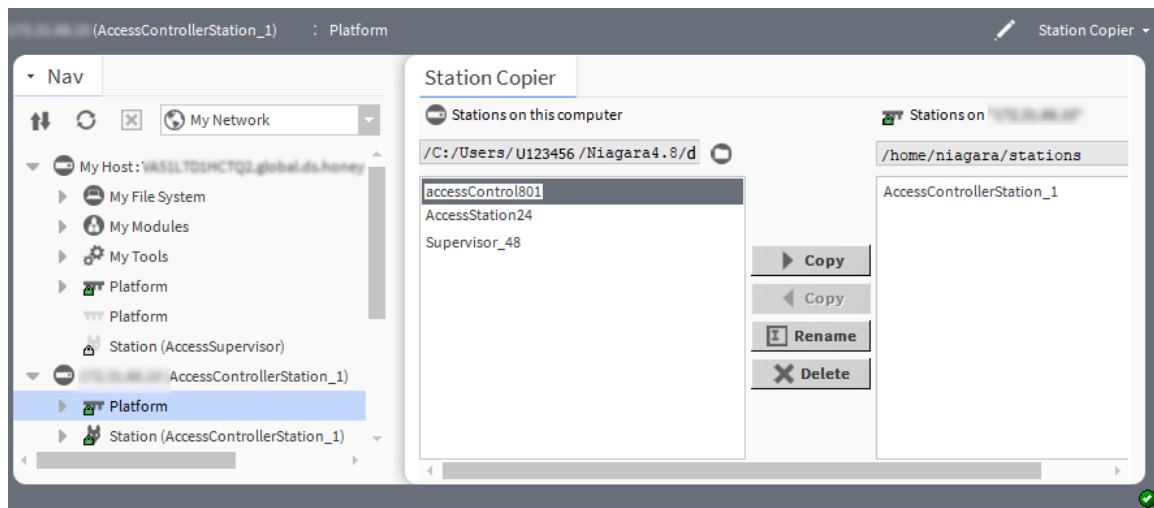
The **Station Copier** can change the name of any station, either in your **User Home** (left side) or in the opened platform's daemon **User Home** (right side).

Prerequisites: You are working in Workbench running on a PC and are connected to the remote station.

If the renamed running station is already included in the **NiagaraNetwork**, its corresponding **NiagaraStation** component remains down until renamed to match the new name. Thus, all child components (Niagara proxy points and so on) will also be down until this is done. In addition, other unforeseen consequences may result from changing the name of a station that has already been integrated into other stations. Therefore, station renames are best done on your **User Home** (left side) stations or when initially configuring a job site network, such as when first installing (copying) a station.

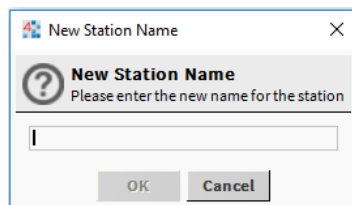
Step 1 In the PC, expand **Platform** and double-click **Station Copier**.

The **Station Copier** view opens.



Step 2 Select the station name in either the left or right pane and click **Rename**.

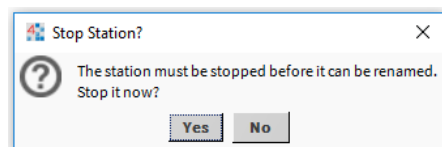
The **New Station Name** window opens.



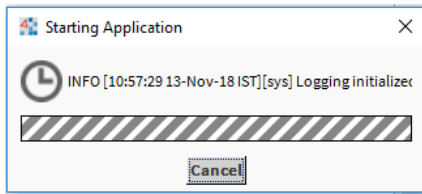
Be careful when renaming stations, as there is no undo.

Step 3 Enter a new name and click **OK**.

A confirmation window informs that the station must be stopped.



After the station stops, the Station Copier renames it and automatically restarts it. A series of other windows open, each showing a station startup message.



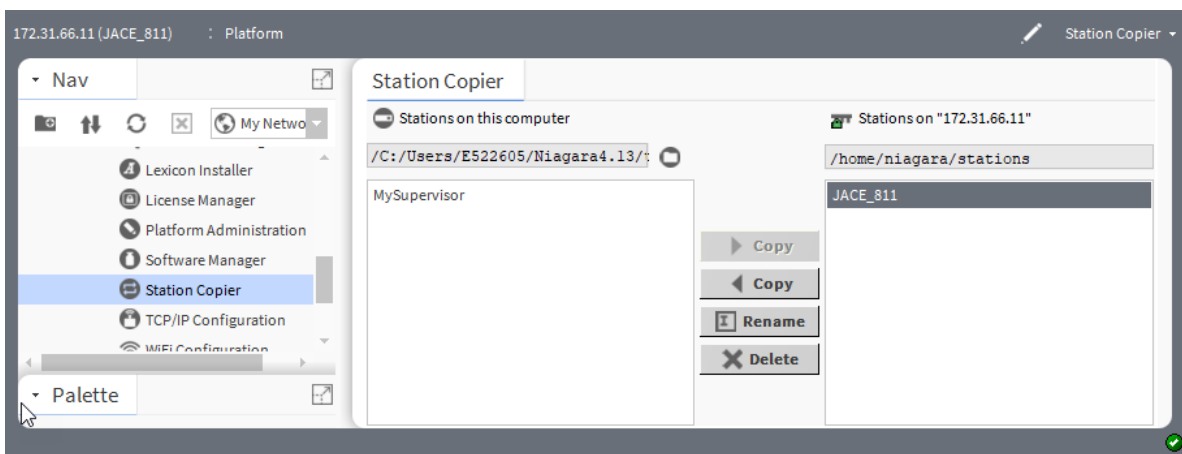
Deleting a station

The **Station Copier** lets you delete any station, either in your **User Home** (left side) or in the opened platform's daemon **User Home** (right side). Be careful when deleting stations, as there is no undo.

Prerequisites: You are working in Workbench running on a PC and are connected to the remote station.

Step 1 In the PC, expand **Platform** and double-click **Station Copier**.

The **Station Copier** view opens.

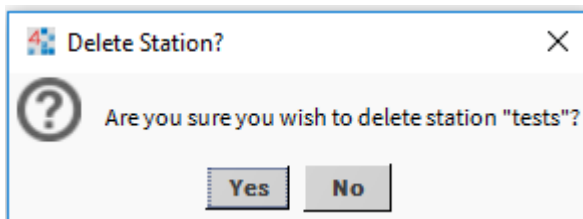


Step 2 Before deleting a running station, make a backup copy first.

If desired, when backing up, you can rename the station using some a temporary name to flag it for later housekeeping.

Step 3 Select the station to delete and click **Delete**.

A confirmation window opens.



Step 4 To confirm the deletion, click **Yes**.

The Station Copier deletes the entire selected station directory including all subdirectories and file contents.

Special notification does not occur if you choose to delete a running station (you may briefly see a stop-station popup, with the opportunity to **Abort**).

Station installation troubleshooting

These troubleshooting tips concern the **Station Copier**.

I started the Station Copier, selected the station and clicked Copy, and got a message prompting me to enter a file passphrase.

The bog file's passphrase is not the same as the target host's system passphrase. You must enter the correct file passphrase to proceed with the station copy. An alternative is to edit the bog file offline to either unlock the file, making it unprotected, or to change the passphrase value. However, either of these choices will clear any sensitive information in the file.

I started the Station Copier, entered the station name and got a message indicating that the controller requires commissioning.

The controller may be new or you may be upgrading from NiagaraAX to Niagara 4 and you forgot to commission the platform first. Run the **Commissioning Wizard** and come back to the Station Copier later.

I started the Station Copier, selected the station to copy and clicked Next, but the only option available is to Cancel and exit the wizard.

The station database (config.bog) is locked. This happens if:

- The config.bog has been edited elsewhere in Workbench and contains unsaved changes. After saving changes, try the copy again.
- The system is in the process of saving the config.bog using the BackupService. Wait a while and try the copy again.

Chapter 11 File Transfer Client

Topics covered in this chapter

- ◆ Transferring files to and from a remote host

The **File Transfer Client** allows you to copy files and/or folders in both directions between your Workbench PC and a remote platform. You can also use it to delete files and folders.

The **File Transfer Client** is useful to copy graphics images to a controller, or to copy a text file from a **User Home** folder on a remote controller (say, `~etc/system.properties`) to your local PC, to allow editing. Then use the **File Transfer Client** to copy the edited version back to the controller's `~etc` folder.

CAUTION: Be careful when using the **File Transfer Client**, especially when copying files to a target platform, or when using the delete (X) control. In either direction, when transferring a file and an identically-named file already exists or if deleting a file, a popup window confirms the action. After confirmation there is no Undo.

Do not use the **File Transfer Client** to copy modules to a controller, as runtime profile types are not applied, nor are module dependencies. Incorrect or missing modules may result. Always use the platform **Software Manager** to install (or uninstall) software modules on a controller.

Transferring files to and from a remote host

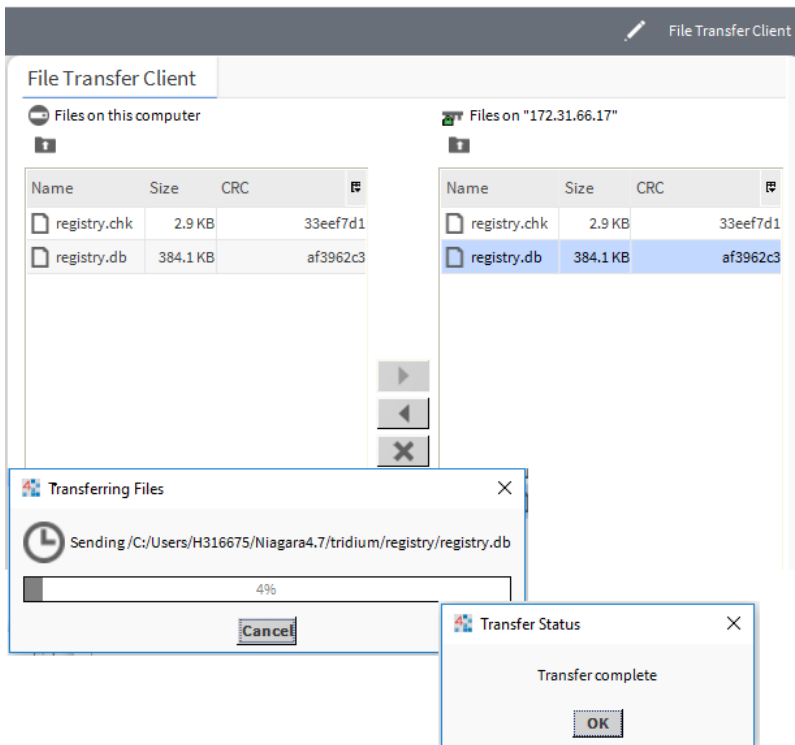
This procedure provides general steps for using the **File Transfer Client** to copy files between a Supervisor PC and a remote controller host. Transferring files between hosts changes the `system.properties` file.

Prerequisites: You are connected working in Workbench running on a PC and are connected to a remote controller platform.

CAUTION: Editing, and especially activating `system.properties` entries is an operation for advanced users, with the possibility of undesirable results. Read all entries in this file carefully, and consult your support channel before making a change! Always save a backup copy of this file before editing it, and test the system after implementing a change.

Step 1 Connect to the platform and click **Platform→File Transfer Client**.

The **File Transfer Client** window opens.



The **File Transfer Client** provides a two-pane view.

- The left pane provides access to local files on your working PC.
- The right pane provides access to files on the remote platform.

Step 2 Click the navigation controls at the top of each pane to go to the appropriate location for source and target.

Step 3 Select one or more items on one side (as source) to copy to the other side (target), and click the appropriate transfer arrow.

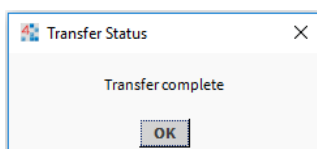
For local-to-remote transfer of a file containing encrypted, sensitive data, the **File Transfer Client** does not prompt you to enter a passphrase.

The transfer completes successfully if the file is protected with a passphrase that matches the system passphrase.

Transfer fails if:

- The file is protected with a passphrase that differs from the system passphrase
- You include more than one protected file in the same transfer

When finished, the system displays:



A station must be restarted before changes to `system.properties` take effect.

Step 4 Stop the station using the platform **Application Director**, and wait for the station to stop completely, ensuring that it saves its database.

- Step 5 From the platform **Platform Administration** view, select **Reboot**. Allow sufficient time for the controller to reboot and station to start.
- Step 6 Reconnect to the station to verify operation.

Chapter 12 Platform tools

Topics covered in this chapter

- ◆ Creating a new station
- ◆ Copying a new station to the daemon user home
- ◆ Importing a license to the local license database
- ◆ Requesting a license from the license server
- ◆ Exporting a license file using a tool
- ◆ Deleting a license file using a tool
- ◆ Using a tool to synchronize licenses

Unlike platform views (which require a platform connection), or equivalent **PlatformServices** plugin views (requiring a station connection), the tools are available whenever running full Workbench.

You find tools on the Workbench **Tools** menu.

Creating a new station

Use the **New Station** tool from the **Tools** menu to create a new controller or Supervisor station. The new station is automatically configured with appropriate services.

Prerequisites: You are connected to the platform that will host the station.

If the platform that you are running is licensed for FIPS, the **New Station** tool creates a FIPS-compliant station.

Step 1 In Workbench, select **Tools**→**New Station**.

The **New Station Wizard** opens.

Name	Vendor	Version	Description
NewControllerStation.ntpl	Tridium	1.5	
NewJACEProvisioningStation.ntpl	Tridium	1.3	
NewSupervisorStationLinux.ntpl	Tridium	1.7	
NewSupervisorStationWindows.ntpl	Tridium	1.7	

Step 2 Enter the **Station Name**.

The **Station Name** property is case-sensitive and must begin with a letter. Best practice is to keep station names short and use a station display name if a longer name with spaces or other characters is required.

To create a directory of the same name, the system automatically appends the **Station Name** text you enter to the read-only **Station Directory** property.

If you enter a duplicate station name, the system prompts you to delete the existing station, as shown below.

Either delete the existing station or enter a different station name.

Step 3 Select a **Station Template** type and click **Next**.

The **Station Templates** table contains the default new station templates provided in Workbench as well as any user-defined templates.

A second window opens to set the admin user password and prompts you choose an action to take once the station creation is complete.

Step 4 To enter the station's password, click **Set Password**.

The **Set Password** window opens.

Step 5 Enter a password for the station and click **OK**.

Your password must contain at least 10 characters, one digit, one lowercase character and one uppercase character.

If the new station template selected earlier contains any exposed properties, such as the ports, etc., this window presents those properties for you to configure.

Step 6 Select an option for the preferred action-on-completion and click **Finish**.

The **New Station Wizard** closes. If the default option, *open it in user home* is selected, a **Property Sheet** view of the new station `config.bog` file opens.

Copying a new station to the daemon user home

In Niagara 4, the **New Station Wizard** finishes with an option to copy the station from the station home (the location for each new station) under your Workbench **User Home** to the **User Home** of the local platform daemon.

Prerequisites: The new station exists in the station home (under User Home).

Step 1 When the **New Station Wizard** prompts you with the option to *Copy station*, select the option and click **Finish**.

Step 2 Make a local platform connection and log on.

The **Station Copier** transfers the station and prompts you with the options to start the station after copying and enabling auto-start.

Step 3 Select the option to start the station.

The **Application Director** opens with the new station present in the daemon **User Home**.

The new station now exists in two locations on your local host: the original location in your Workbench **User Home**, and also in the platform daemon **User Home**.

Once the station is running in the daemon **User Home**, you can make a backup of the running station, where the backup .dist file goes in the `backups` folder of your Workbench **User Home**. Or, you can use the platform **Station Copier** to copy the station back to the `stations` folder of your Workbench **User Home**.

NOTE: Using the **Station Copier** to copy the station back to your Workbench **User Home** is highly recommended if you made any changes to the station. This is essential if you are installing it (copying it) to any remote platform. Remember, the copy of the station in your Workbench **User Home** is immediately obsolete as soon as you make changes to the copy of the station running in the daemon **User Home**.

Importing a license to the local license database

The local license database is available under the platform's **My File System**. You can use this procedure to add and update licenses in a license archive or the equivalent import command from the platform **License Manager** (or similar **License Platform Service Plugin**).

Prerequisites: You are working in Workbench and are connected to a Supervisor platform.

Step 1 Expand **My File System**→**Sys Home**→**security** and double-click **licenses**.

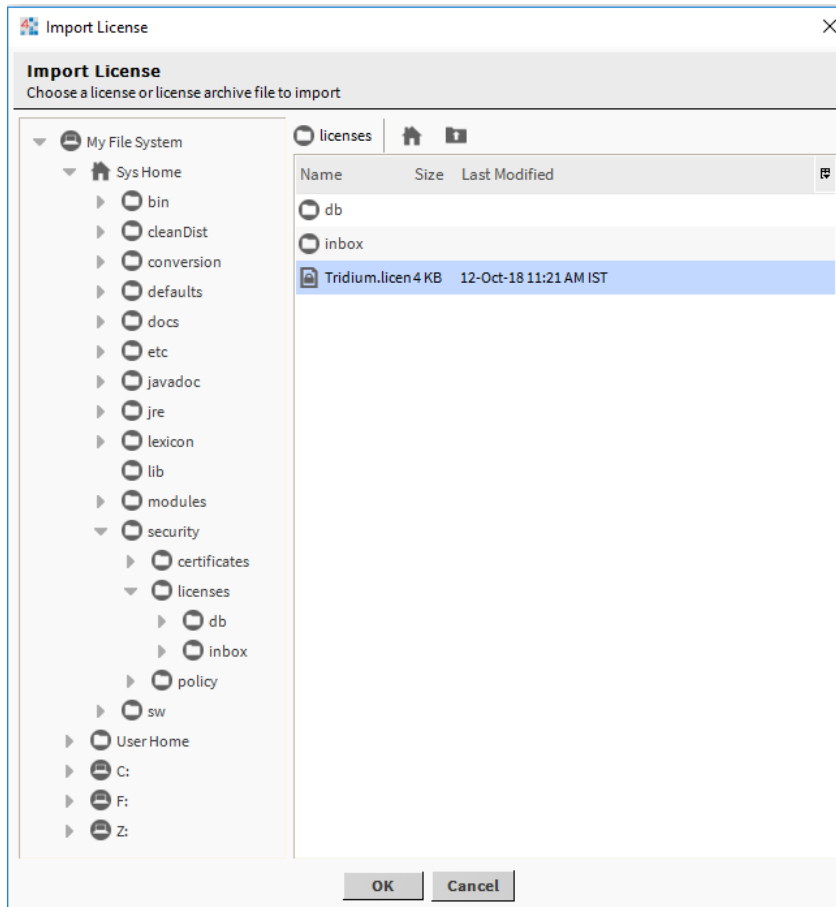
The **Directory List** displays the licenses in the local license database.

Step 2 To import a license from this database, click **Tools**→**Local License Database**

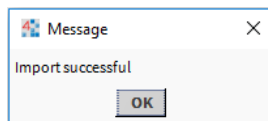
The **Workbench License Manager** opens.

Step 3 Click **Import File**

This button in the **Workbench License Manager** is always enabled, and opens the **Import License** window for you to navigate to a source file (.license or .lar). Only two types of files appear for selection.



- Step 4** To add to (or update in) your local license database, select a license file and click **OK**.
A popup window confirms success, and the license(s) are added or updated in your database.



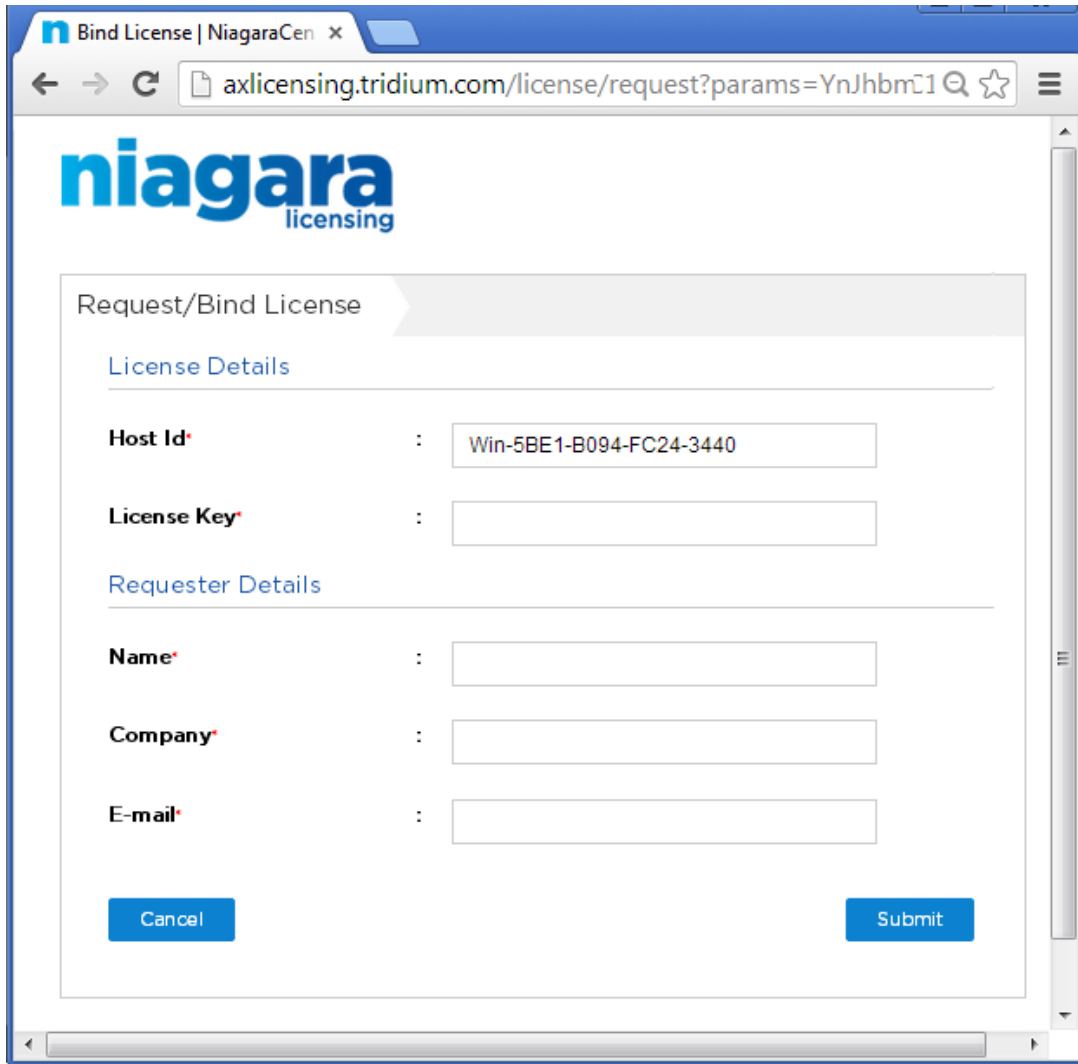
If any of the license(s) you select to import are older than the ones currently in your local database, meaning that the generated attribute timestamp is earlier, newer license(s) in your local license database are not overwritten. However, the same import successful message popup appears for such file import operations.

Requesting a license from the license server

You do not need a platform or station connection to request a license from the online license server.

Prerequisites: You are using Workbench running on a PC that is licensed.

- Step 1** Select **Tools**→**Request License**,
The **Request/Bind License** form opens.



The screenshot shows a web browser window with the title 'Bind License | NiagaraCen'. The address bar shows the URL 'axlicensing.tridium.com/license/request?params=YnJhbm1'. The page features the 'niagara licensing' logo. Below the logo is a section titled 'Request/Bind License' with a sub-section 'License Details'. This section contains two input fields: 'Host Id*' with the value 'Win-5BE1-B094-FC24-3440' and 'License Key*'. Below this is a sub-section 'Requester Details' with three input fields: 'Name*', 'Company*', and 'E-mail*'. At the bottom of the form are two buttons: 'Cancel' and 'Submit'.

Step 2 Enter the **Host Id** for another PC or for a remote controller and click **Submit**.

If you are requesting a license for another PC on which you have installed Niagara, enter the PC's host ID along with the other pertinent information.

Exporting a license file using a tool

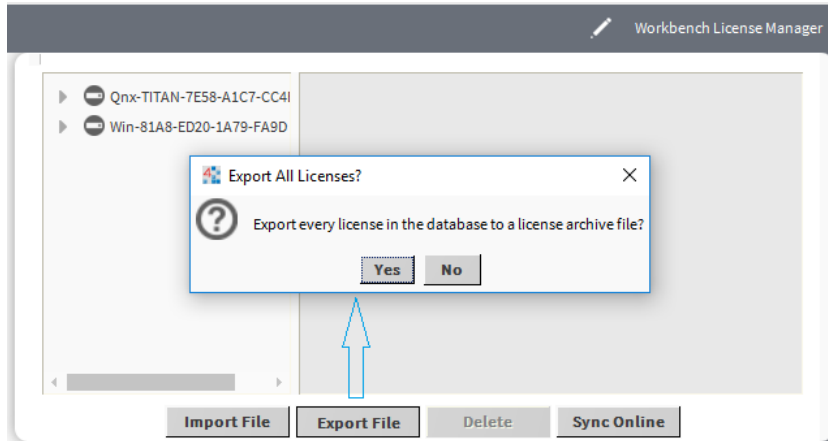
You may export a license using a platform tool without first making a platform connection. You may save any number (or all) licenses in your local license database locally on your Workbench PC, as a license archive (.lar) file.

Prerequisites: You are using Workbench running on a PC that is licensed.

The license archive format allows you to easily share saved .lar files (however they are named) among multiple PCs without overwriting a license file for a different host platform.

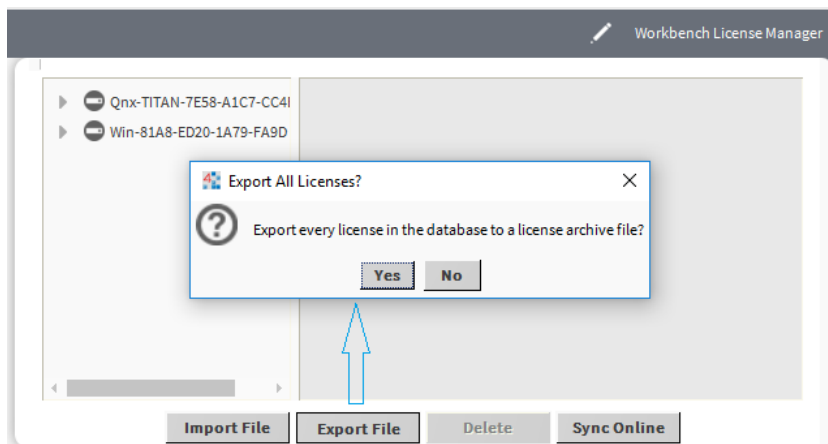
Step 1 Select **Tools→Local License Database**.

The **Workbench License Manager** opens.



Step 2 To export all licenses from your local license database, click **Export File** without first selecting a license.

The system prompts you to confirm that you intended to export all licenses.

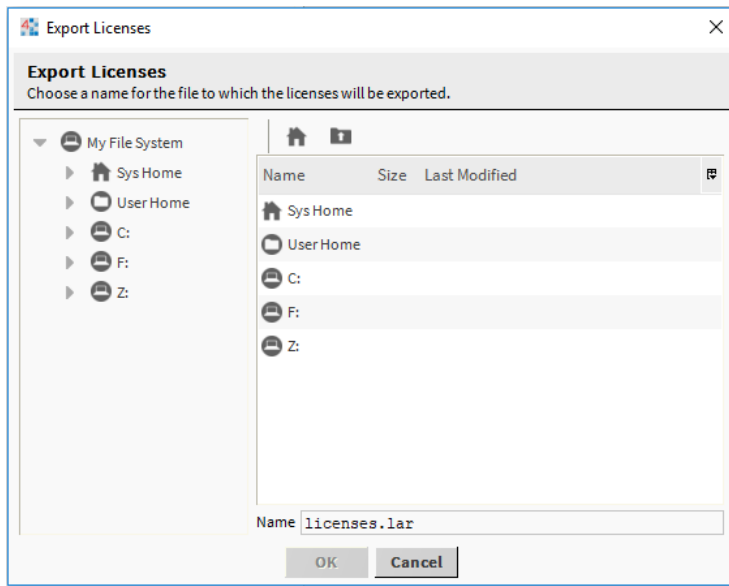


Step 3 To export one or more specific licenses, select them in the left pane (host IDs or license files).

Step 4 Do one of the following:

- If you are exporting all licenses, click **Yes**.
- If you are exporting selected licenses, click **Export File**.

An **Export Licenses** file chooser opens.



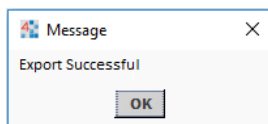
Step 5 Navigate to the spot to save the .lar file.

Step 6 To rename the license archive file change the name.

For example, instead of: licenses.lar, you could rename it MyController.lar.

Step 7 To continue, click **OK**.

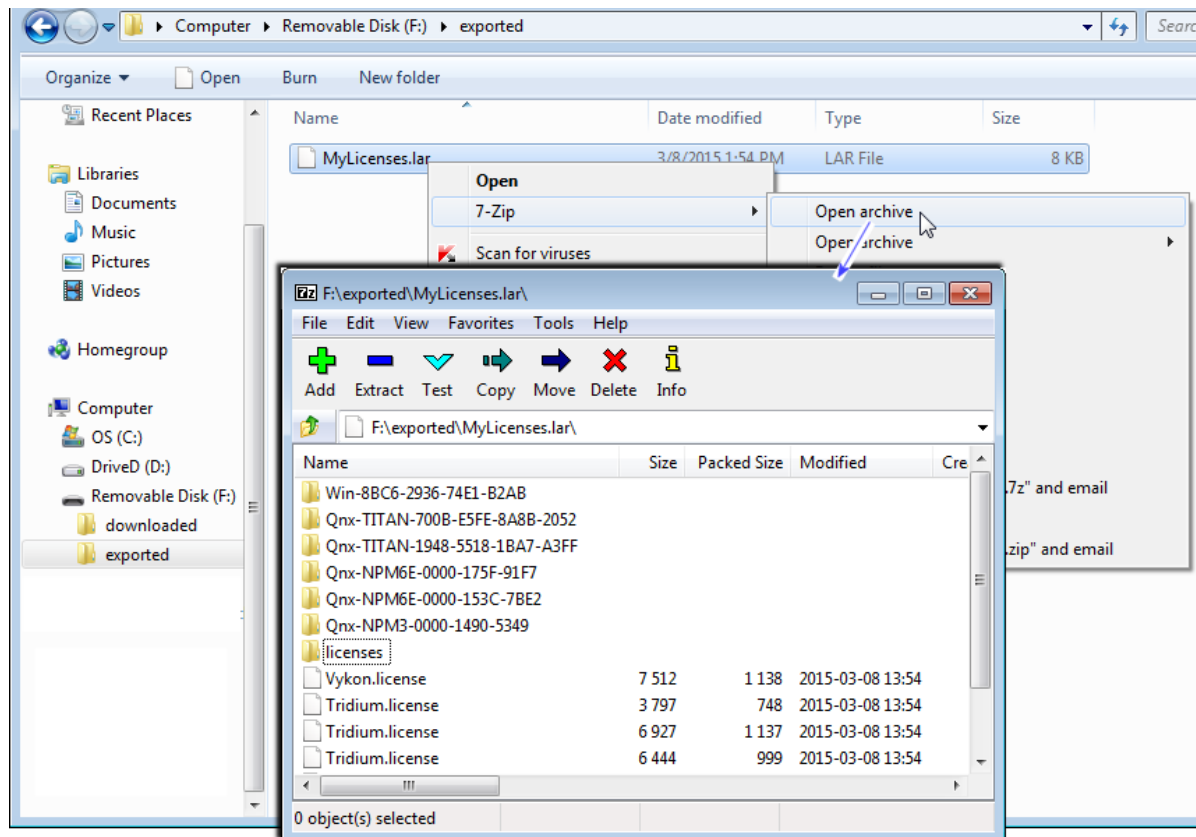
A popup window confirms export success.



The system saves the license in a compressed (zip-compatible) format known as a license archive. This is a file with a .lar file extension. It includes the complete `licenses/hostID` folder (subdirectory) structure (relative to sys home) for any included licenses.

Step 8 To view the license zip file, use Windows Explorer to navigate to the folder that contains the file, right-click the .lar file and open it with a utility, such as 7-zip.

The .lar file contains one or more licenses.



The shows a .lar file in Windows Explorer, opened using 7-Zip, and its subsequent contents. In this case, where the archive contains multiple licenses, it was created by an export performed using the **Workbench License Manager** tool.

Deleting a license file using a tool

You may delete a license using a platform tool without first making a platform connection.

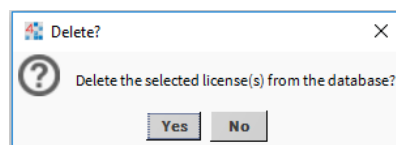
Prerequisites: You are using Workbench running on a PC that is licensed.

Step 1 Select **Tools**→**Local License Database**.

The **Workbench License Manager** opens.

Step 2 Select a license to delete and click **Delete**.

A **Delete?** window prompts you to confirm this action.



Step 3 To delete the selected license(s), click **Yes**.

If the selected host ID folder contained only a .license file, the system removes the entire folder. If the folder contained other files (or subfolders), the system removes only the selected .license file, which no longer appears in the left pane.

Step 4 You may need to click refresh (🔄) to update the left pane contents.

Using a tool to synchronize licenses

Using the **Workbench License Manager** you may update any number (or all) licenses in your local license database with the most current license available online from the licensing server. This feature requires Internet connectivity from your Workbench PC.

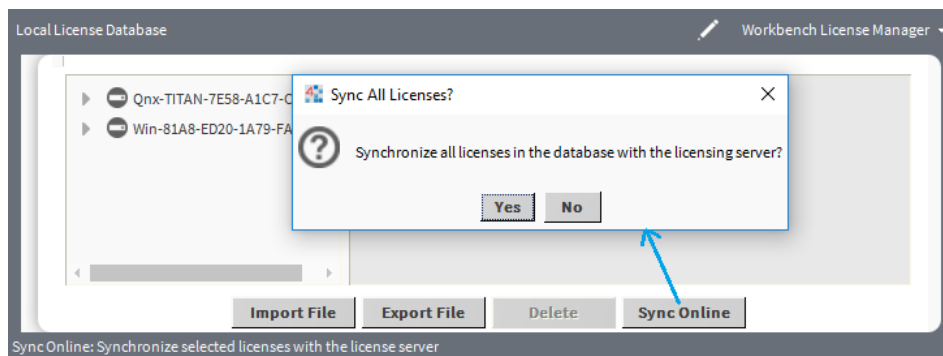
Prerequisites: You are using Workbench running on a PC and you have Internet access.

Step 1 Select **Tools**→**Local License Database**.

The **Workbench License Manager** opens.

Step 2 To include every license in your license database, click **Sync Online** without first selecting one or more licenses.

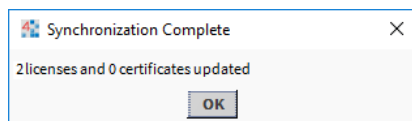
The system prompts you to confirm.



Step 3 Do one of the following:

- To synchronize all licenses, click **Yes**.
- To synchronize selected licenses, click **Sync Online**.

The system sends an immediate request to the licensing server. Intermediate popup windows may briefly appear while the sync request is handled. The operation concludes with a **Synchronization Complete** prompt, which summarizes the number of licenses and certificate files that were updated in your local license database.



If all licenses (and certificates) were already up-to-date, this window reports 0 licenses and 0 certificates updated.

Chapter 13 Platform Services

Topics covered in this chapter

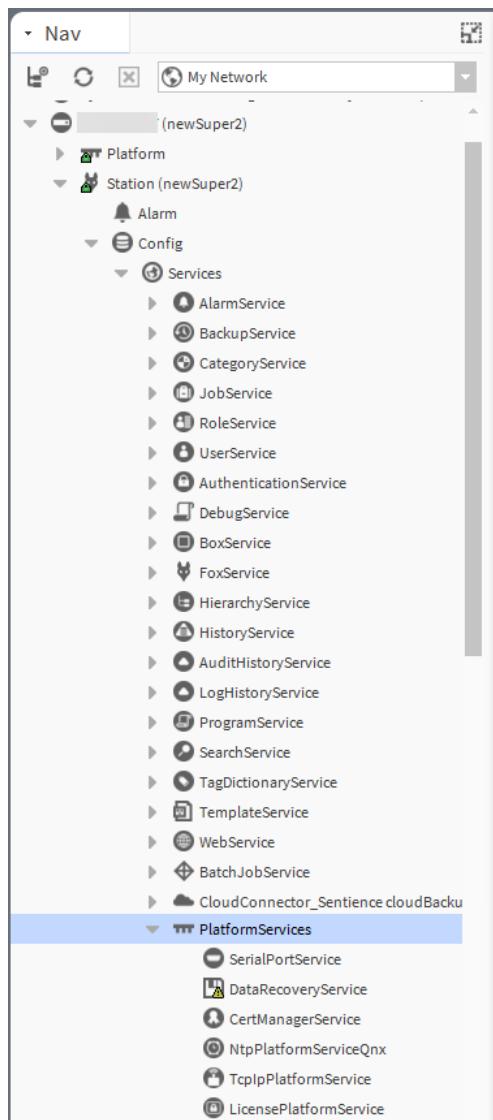
- ◆ Monitoring power to a controller
- ◆ About the NtpPlatformService
- ◆ Reverting to the legacy CPU usage for BACnet networks
- ◆ Global capacity licensing

PlatformServices provide the station's perspective on its host platform. They act as the station interface to specifics about the host platform (whether a controller or a PC). Unlike the various platform views, a platform connection is not needed to access **PlatformServices**. Instead, you need only a standard station (Fox) connection.

A station user with admin-level permissions on the **Services** container (in the component **Config** space) of a running station also has access to a special subset of platform functions, via **Platform Services**.

Under **Config, Services**, every running station has a **PlatformServices** container, which any station user, with admin-level permissions to this component, can access. **PlatformServices** are built dynamically at station runtime—you do not see **PlatformServices** in an offline station.

Figure 22 Example of a station's PlatformServices



PlatformServices and all child components are unique from all other station components. In a running station they provide two main types of functionality:

- A subset of the platform views that are available in a platform connection. These services do not provide the full set of functions available in a platform connection. For example, you cannot install or upgrade software, or transfer stations and files. However, a number of platform configuration views are available.

Apart from configuration usage, some **PlatformServices** provide status values that you can further incorporate including built-in alarm features. Usage is typical for power monitoring.

- Certain platform configuration settings are accessible only through **PlatformServices** and are not available in a client platform connection.

Changes you make to **PlatformServices** and all child services are not stored in the station database. Instead, changes are stored in other files on the platform, such as its platform.bog file, or within the platform's operating system. The changes are independent from the running station. Do not attempt to edit the platform.bog directly; always use **PlatformServices'** views.

If you install another station, **PlatformServices** are dynamically recreated again when the new station starts, based upon the last settings.

Some **PlatformServices** changes may require you to reboot the host to become effective. Examples include: TCP/IP changes and some NTP-related changes in a controller. A **Reboot Now?** window opens upon saving such a change.

NOTE: When you design station security, be careful about assigning user permissions to **PlatformServices** and its child service components. In general, you should regard this portion of the station as most critical, as it allows access to items such as host licenses and TCP/IP settings. Furthermore, right-click actions on the **PlatformServices** include **Restart Station**.

Table 2 Platform Services

Service	Platform	Description
CertManagerService	both PC and controller	Manages PKI certificate stores and/or allowed host exceptions, used in certificate-based TLS connections between the station/platform and other hosts.
TcpIpPlatformService	both PC and controller	Provides access to the same configuration using the platform's TCP/IP Configuration view.
LicensePlatformService	both PC and controller	Provides access to the same configuration using the platform's License Manager view.
SerialPortService	controller only	Allows review of available serial ports on the host platform.
NtpPlatformService	controller only	Provides the Niagara 4 interface to the NTP (Network Time Protocol) service or daemon of a controller's OS (QNX), including several configuration properties and a list specifying one or more NTP time servers.
DataRecoveryService	controller only	Monitors the service that automatically creates and manages static RAM buffers in the controller, allowing battery-less operation (if so configured), or usage of the SRAM along with an installed backup battery (if applicable).
HardwareScanService	controller only	Provides a graphical diagram of communication ports and other features on the hosting platform, including callouts to a table that explains the location, description (such as COM2), port type, and status/usage of each item. This optional feature requires the installation of the modules <code>platHwScan</code> and a corresponding <code>platHwScan<type></code> where <code><type></code> is a controller model.
SyslogPlatformService	both PC and controller	Provides a standard protocol for message logging. It allows messages that are generated by Niagara to be stored and analyzed on a remote server.

Monitoring power to a controller

Using the **PowerMonitorService**, a controller provides status monitoring of its AC power and battery level.

Prerequisites: You are running Workbench and are connected to a running remote station.

Step 1 Expand **Config→Services→PlatformServices** and double-click **PowerMonitorService**.

The **Power Monitor** view opens.

Step 2 To configure this service, change the Shutdown Delay Time and or alarm source configuration for both types of alarms: low battery level and primary power lost.

The software provides status monitoring of the following items, via Boolean slots:

- AC power is monitored by the Primary Power Present slot. This slot reports `true` when AC power is currently supplied to the controller.
- Battery level is monitored by the Battery Good slot. This slot reports `true` if the last test of the NiMH battery was good.

Step 3 If needed, make a Px binding or link to the these slots

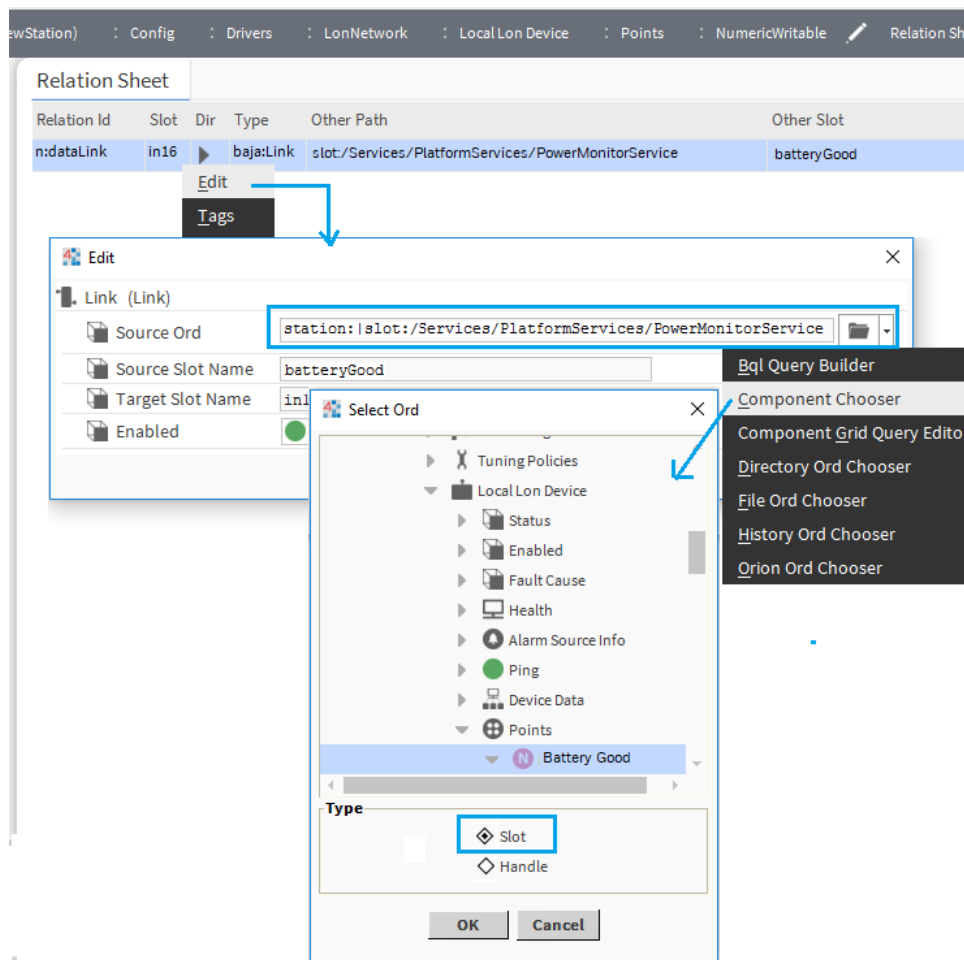
Because any station's **PlatformServices** are dynamically built upon startup, if binding its slots to Px widgets (or linking to other station components), be aware of the following limitations and guidelines:

- Subscription behavior is unique to a station's **PlatformServices** slots in that property values initially load, but do not automatically update. To explicitly refresh such properties, you must invoke the poll action on the container for those properties.

For example, if on a Px page you bind a BoundLabel to the **PowerMonitorService's Battery Good** slot, it displays `true` or `false`, however, this value does not update until you right-click and execute the Poll action, which forces a fresh read.

- Links from **PlatformServices** (and child slots) to other station components must use a source ORD's slot path rather than its handle. After a station restart or host reboot, handle-sourced links may be lost.

Consider this update limitation before linking **PlatformServices** slots into other components that provide control logic. Linked slot values may well be outdated shortly after station startup, yet still subscribed and not marked as stale.



The screen capture shows a link from the RelationSheet of a target component and how to edit the link to use the slot path for source ORD.

In addition to the read-only status slots, the **PowerMonitorService** provides related configuration slots, which you typically review at commissioning time. For more details refer to the *JACE Niagara 4 Install and Startup Guide*.

About the NtpPlatformService

PlatformServices, in a QNX hosted station, contains a child **NtpPlatformServicesQnx** component, which provides an interface to the RFC 1305-compliant NTP (Network Time Protocol) service or daemon running on that host platform. NTP is the currently recommended time synchronization protocol to use between inter-networked devices, offering more accuracy than the older RFC 868 Time Protocol.

By default, this platform service is disabled.

- If left disabled, this platform service does nothing.
- If enabled, this platform uses NTP as a client to sync its clock with time values retrieved from one or more NTP time servers, according to other configuration properties.

An enabled **NtpPlatformService** will not allow client synchronization with time servers using RFC 868, even if the station also has a **TimeSyncService** under its **Config→Services** folder.

NOTE: Support for **Windows** and **Linux** Supervisor **NTP Platform Service** has been discontinued in Niagara 4.9 and later. Previous versions of the **NTP Platform Service** in these environments was readonly. Support for this service continues for embedded environments. The **NTP Platform Service** is now removed from the platform.bog file in Niagara 4.9 and later for affected environments.

Verifying access to an NTP server

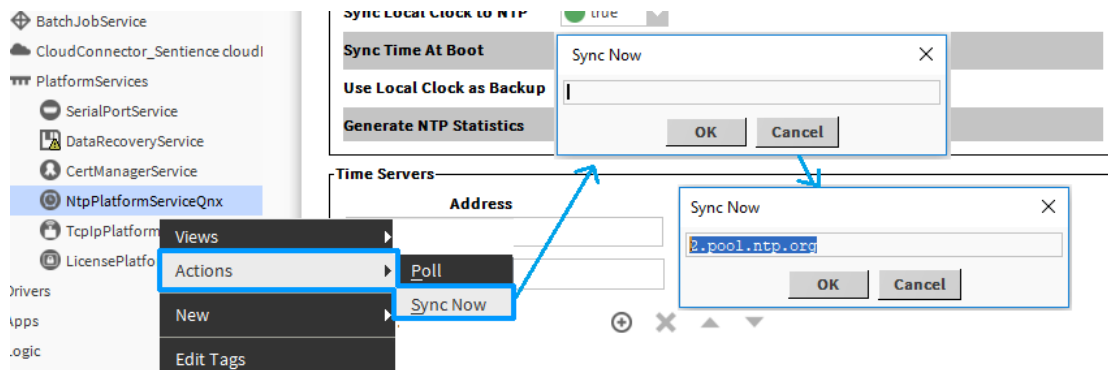
This procedure verifies that a provided NTP server is reachable and responding. It cannot be used while NTP is enabled on the controller.

Prerequisites: You are using Workbench and are connected to the remote controller. You know the domain name or IP address of the public NTP server the controller is attempting to connect to.

Step 1 Disable NTP for the controller.

Step 2 Expand **Config→Services→PlatformServices**, right-click **NtpPlatformServiceQnx** and click **Actions→Sync Now**.

The **Sync Now** window opens.



Step 3 Type in the fully qualified domain name of a public NTP server or the IP address of any accessible NTP server and click **OK**.

Step 4 To verify, do one of the following:

- Right-click the station in the Nav tree, select **Spy→platform diagnostics→log**.
- Click **File→Open ord (Ctrl + L)** and enter: `ip:<controller_IP_address>|fox:|spy:/platform diagnostics/log`, where the variable name `<controller_IP_address>` is the controller's IP address, and press **enter**.

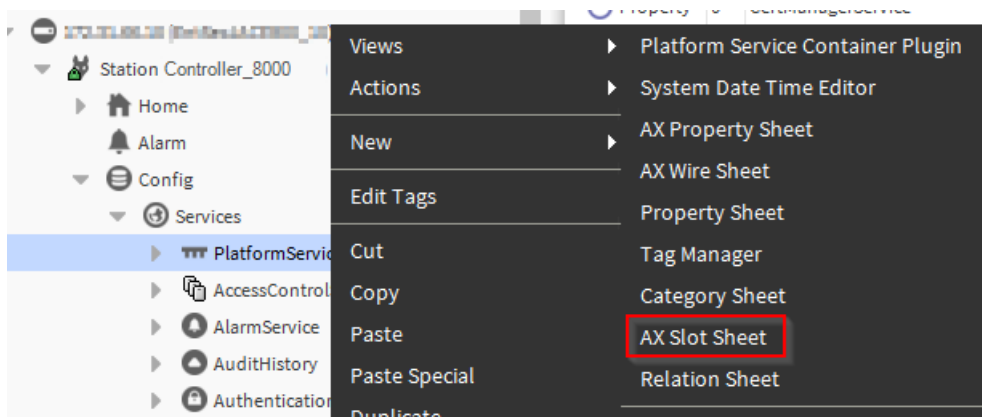
Reverting to the legacy CPU usage for BACnet networks

For controllers running Niagara 4.12 and later, the MS/TP engine is located, by default, on the controller's RS-485 co-processor. This task describes how to configure the `.platMstp-BacnetMstpPlatformServiceQnx` module so that the MS/TP engine runs on the controller CPU instead.

Prerequisites: You are working in Workbench and are connected to a platform and running station on a properly configured BACnet Network.

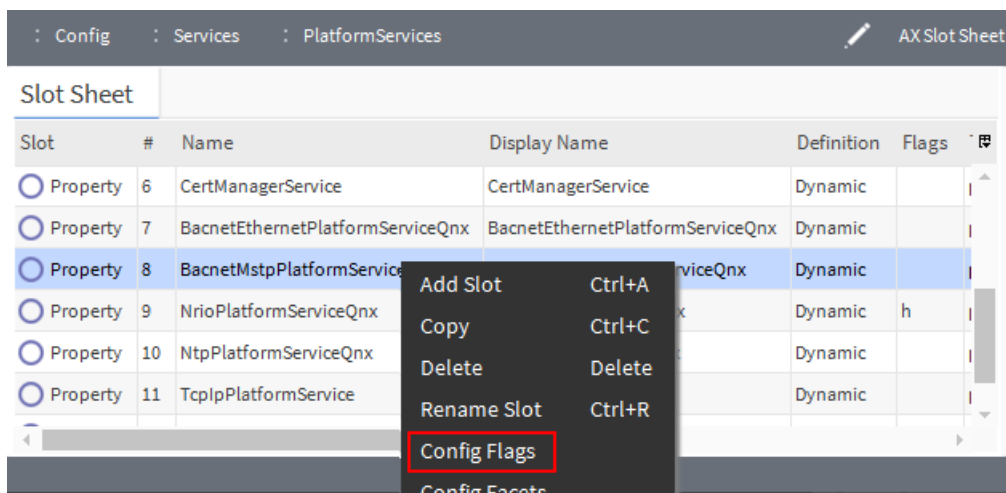
The **BacnetMstpPlatformService** is hidden by default. This procedure explains how to make it visible under **PlatformServices**, and how to disable the coprocessor.

- Step 1 Expand the controller station's **Station**→**Config**→**Services**, right-click the **PlatformServices** and select **AX Slot Sheet** from the popup menu.

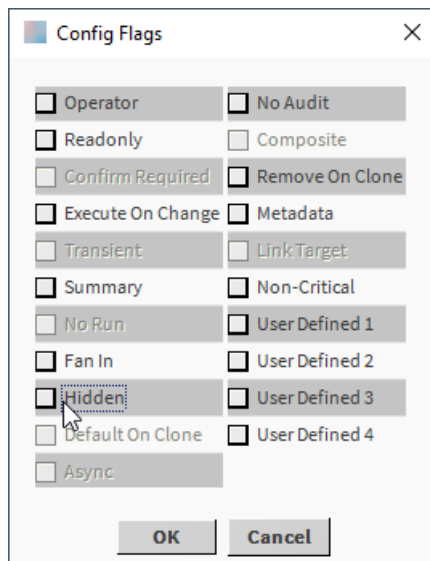


The **PlatformServices AX Slot Sheet** view opens.

- Step 2 Right-click the **BacnetMstpPlatformServiceQnx** and select the **Config Flags** menu item.



The **Config Flags** window opens.

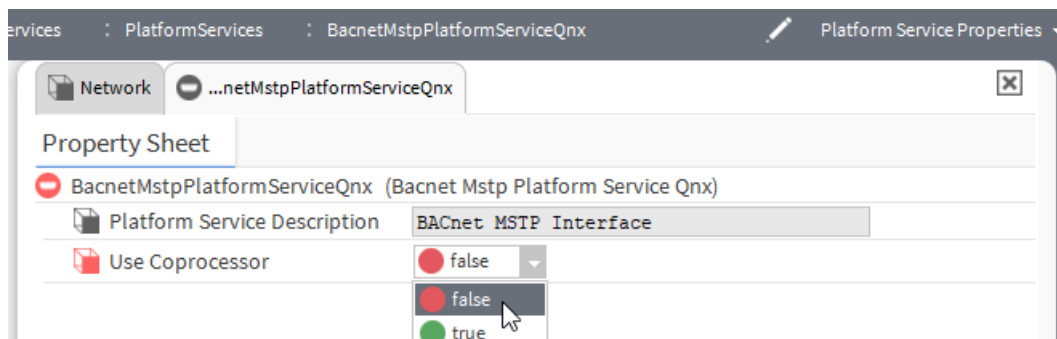


Step 3 Clear the **Hidden** property check box and click **OK**.

The **BacnetMstpPlatformService** component displays under **PlatformServices** in the Nav tree.

Step 4 To display the **Use Coprocessor** property, right-click **BacnetMstpPlatformServiceQnx**, choose its **AX Slot Sheet**, right-click **useCoprocesor**, select **Config Flags** from the popup menu, clear the **Hidden** property check box and click **OK**.

The **useCoprocesor** property is now visible in the **Platform Service Properties** view.



Step 5 Double-click the **BacnetMstpplatformServiceQnx** node in the Nav tree, set **Use Coprocessor** to **false** and click the **Save**.

Step 6 Restart the station to finalize the change in processing location.

Global capacity licensing

This licensing model is available for some platforms, among them the JACE-8000 and JACE-9000. Global capacity licensing tracks specific resources in the station using global counters. This provides more flexibility in how resources are allocated.

The station keeps a global count of all networks, devices, proxy points, links, histories and schedules. When one of these resources goes over a licensed limit, the resource either goes into fatal fault or becomes inactive. The content of any particular global capacity license depends on the feature purchased for the controller. For example:

```
<feature name="globalCapacity" expiration="2016-04-01"
point.limit="1250" device.limit="50"
excludedDevices="ndio;nrio;niagaraDriver"
```

```
excludedPoints="ndio;nrio;niagaraDriver"/>
```

The example above sets global limits on points (1250) and devices (50), but no limits on networks, links, histories, or schedules. The `excludedDevices` and `excludedPoints` attributes mean that there is no limit on the number of devices and points from these modules: `ndio`, `nrio` or `niagaraDriver`.

A more restrictive global capacity example feature could look like below.

```
<feature name="globalCapacity" expiration="never" network.limit="3"
device.limit="25" point.limit="500" link.limit="400" history.limit="125"
schedule.limit="10" excludedNetworks="nrio" excludedDevices="nrio"
excludedPoints="nrio"/>
```

The example indicates:

- A limit of three networks of any kind
- A limit of 25 devices of any kind,
- A limit of 500 points of any kind
- A limit of 400 links of any kind.
- A limit of 124 histories and 10 schedules of any kind.

Attributes allow for `excludedNetworks`, `excludedDevices`, and `excludedPoints`, each as a comma-separated list of modules. This means that there is no limit for `nrio` networks, devices and ports. Any modules in these attributes are excluded from the respective global capacity limit. However all links, histories, and schedules from these modules are not excluded from any other global capacity counts. All stations include an `AuditHistory` and `LogHistory`, which are included in the history limit.

Example globalCapacity feature entry

```
<feature name="globalCapacity" expiration="2016-04-01"
point.limit="1250" device.limit="50"
excludedDevices="ndio;nrio;niagaraDriver"
excludedPoints="ndio;nrio;niagaraDriver"/>
```

The example above sets global limits on points (1250) and devices (50), but no limits on networks, links, histories, or schedules. The `excludedDevices` and `excludedPoints` attributes mean that there is no limit on the number of devices and points from these modules: `ndio`, `nrio` or `niagaraDriver`.

A more restrictive global capacity example feature could look like below.

```
<feature name="globalCapacity" expiration="never" network.limit="3"
device.limit="25" point.limit="500" link.limit="400" history.limit="125"
schedule.limit="10" excludedNetworks="nrio" excludedDevices="nrio"
excludedPoints="nrio"/>
```

The example indicates:

- A limit of three networks of any kind
- A limit of 25 devices of any kind,
- A limit of 500 points of any kind
- A limit of 400 links of any kind.
- A limit of 124 histories and 10 schedules of any kind.

Attributes allow for `excludedNetworks`, `excludedDevices`, and `excludedPoints`, each as a comma-separated list of modules. This means that there is no limit for `nrio` networks, Devices and ports. Any modules in these attributes are excluded from the respective global capacity limit. However all links, histories, and schedules from these modules are not excluded from any other global capacity counts.

NOTE: All stations include an `AuditHistory` and `LogHistory`, which are included in the history limit.

Checking capacity licensing status

Every station that runs on a platform with global capacity licensing keeps a running tally on all corresponding resources in the **Resource Manager** view.

Prerequisites: You are using Workbench and are connected to a station with global capacity licensing.

Step 1 Right-click the opened **Station** and select **Views→Resource Manager**.

The **Resource Manager** opens.

Name	Value
capacityLicensing.recountLastFail	never
capacityLicensing.recountLastFailReason	
capacityLicensing.recountLastRun	09-Mar-15 4:16 PM IST
engine.queue.longTimers	13 (Peak 13)
globalCapacity.devices	46 (Limit: 50)
globalCapacity.histories	158 (Limit: none)
globalCapacity.links	400 (Limit: none)
globalCapacity.networks	5 (Limit: none)
globalCapacity.points	1,208 (Limit: 1,250)
globalCapacity.schedules	5 (Limit: none)
heap.free	128 MB

Any station with global capacity licensing has specific entries in the lower Resource Information table of this view.

In this example, three `capacityLicensing.recount` statistics include a timestamp that indicates when the global capacity recount last ran.

Another group of `globalCapacity.resource` statuses show the current counts along with respective license limits (if any). The station view above reflects the first `globalCapacity` example given is:

```
<feature name="globalCapacity" expiration="2016-04-01" point.limit="1250"
device.limit="50" excludedDevices="ndio;nrio;niagaraDriver"
excludedPoints="ndio;nrio;niagaraDriver"/>
```

where global limits exist only on devices (Limit: 50) and proxy points (Limit: 1,250) with `ndio`, `nrio` and `niagaraDriver` devices and points being excluded from any limits.

Step 2 To refresh the count, click **Update**.

Often you delete as well as add resources in the process of engineering a station. Capacity licensing never decrements any resource counter. This could result in inaccurate counts over a long period of time. Therefore, any resource created with an over-capacity error will never get out of fault. You must delete it.

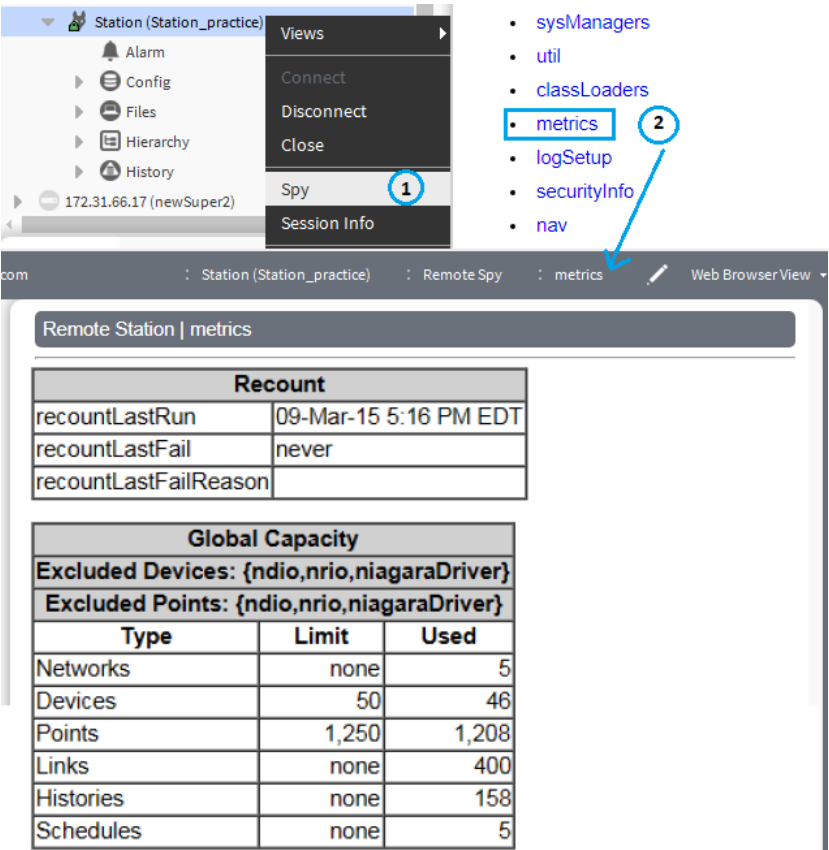
A station restart corrects global capacity counts. However, since this is often inconvenient, the system performs a global capacity recount approximately every 10 minutes to correct any inflated

counts. You can always see the recount status, along with the current global licensing counts and limits in the station’s **Resource Manager** view (or spy view).

Step 3 If you are a super user, you can get this same information by right-clicking the spy page on a station, at the following location: **Spy→metrics**.

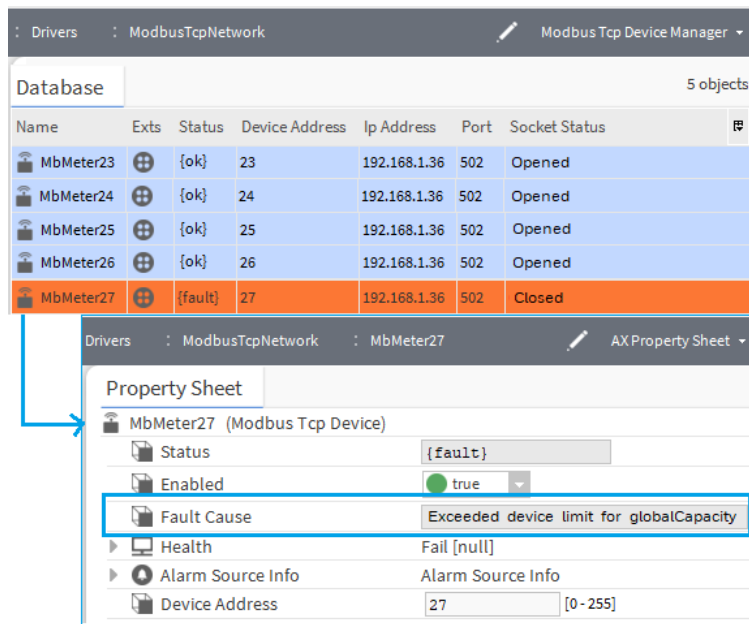
Special permission is required to view spy pages.

The **Remote Station | metrics** view opens.



This information matches the **Resource Manager** view example with the exception of the **recountLastRun** timestamp shown (one hour later).

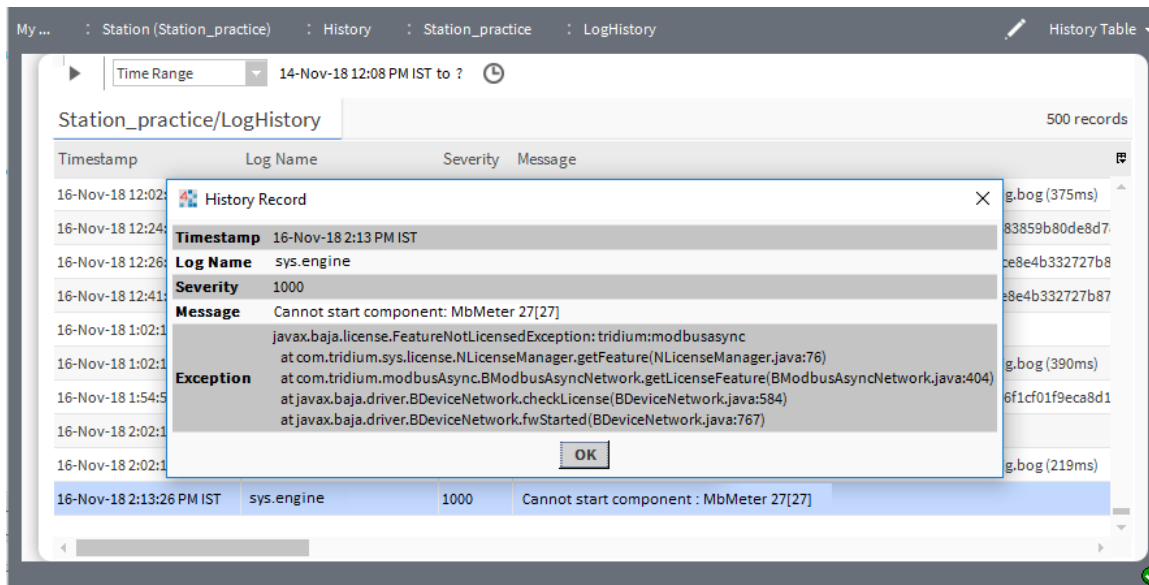
Added components that exceed global capacity limits provide a Fault Cause explaining the reason. In this example, where there are 46 global existing devices, if five (5) new ModbusTcpDevices are added this results in a fault, as 51 devices is one over the 50 device limit.



As shown above, the property sheet for the device shows this reason in **Fault Cause**.

Step 4 Delete this (or any) component with a similar fault cause.

If you do not delete the component, it remains in fault. Corresponding events are also entered in the station's LogHistory.

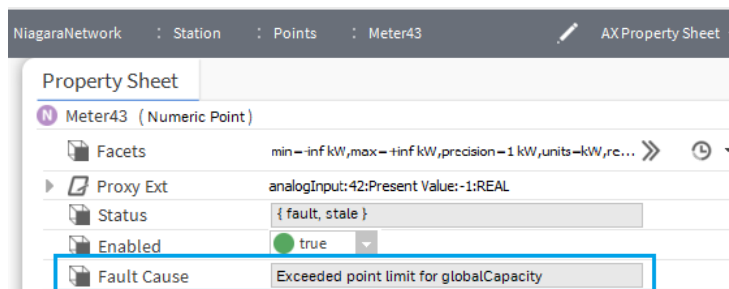


The globalCapacity count for any exceeded resource appears in the corresponding entry in the station's **Resource Manager**.

Resource Information		34 objects
Name	Value	
component.count	418	
globalCapacity.devices	51 (Limit: 50)	
globalCapacity.histories	158 (Limit: none)	
globalCapacity.links	400 (Limit : none)	

The necessary deletion of an over-limit device (51) does not decrement the count back at that time. Instead, a periodic recount (about every 10 minutes) or station restart is needed to update the count.

Similar component faults and error logs apply to networks, points, links, and schedules.



As shown above, the ProxyExt for a proxy point shows a **Fault Cause** reason. The property sheet of a network in global capacity fault is similar `Exceeded network limit for globalCapacity`, and this applies also to a schedule in global capacity fault: `Exceeded schedule limit for globalCapacity`.

Exceeding the globalCapacity link limit produces a popup **Capacity Licensing** window on the wire sheet or active view that reports: "Exceeded Link Limit," and the link is not functional.

Capacity licensing and histories

Histories that exceed the globalCapacity limits manifest in different ways. If the number of histories in the station is greater than the globalCapacity limit, the resource counter in the station prevents more histories from being created.

In the default **History Ext Manager** view on the **HistoryService**, you see a table of all history extensions along with the state of each (enabled, disabled, fault). Included is a total count of all history extensions. However, there is no easy, intuitive way to get a count of those in a particular state or set of states. An extension in a fault state with a **Fault Cause** of `Exceeded history limit` indicates a related problem. Simply disabling such an extension does not fix the issue. You must delete the history in the history database, which cannot be done from the **HistoryService**.

You delete histories from the **Database Maintenance** view of the **History** space; but note there are no counts available there. If you delete one or more histories to reduce history count, you must wait until the periodic recount (approximately every 10 minutes) calculates the reduced count. However, you cannot delete the history extension that created the history from there. So it is likely that the history is going to be re-created in the future, unless you delete or modify the history extension that created the history database table.

In addition to history extensions in the same station, other items can create histories, including:

- History imports in the same station
- History exports in another station
- AuditHistory service
- LogHistory service

These activities add to the count. This is the reason why a station quickly exceeds its history limit.

Chapter 14 Supervisor components

Topics covered in this chapter

◆ Workbench License Manager (platform-LicenseDatabaseTool)

Components include services, folders and other model building blocks associated with a module. These components are available only in a Supervisor PC platform.

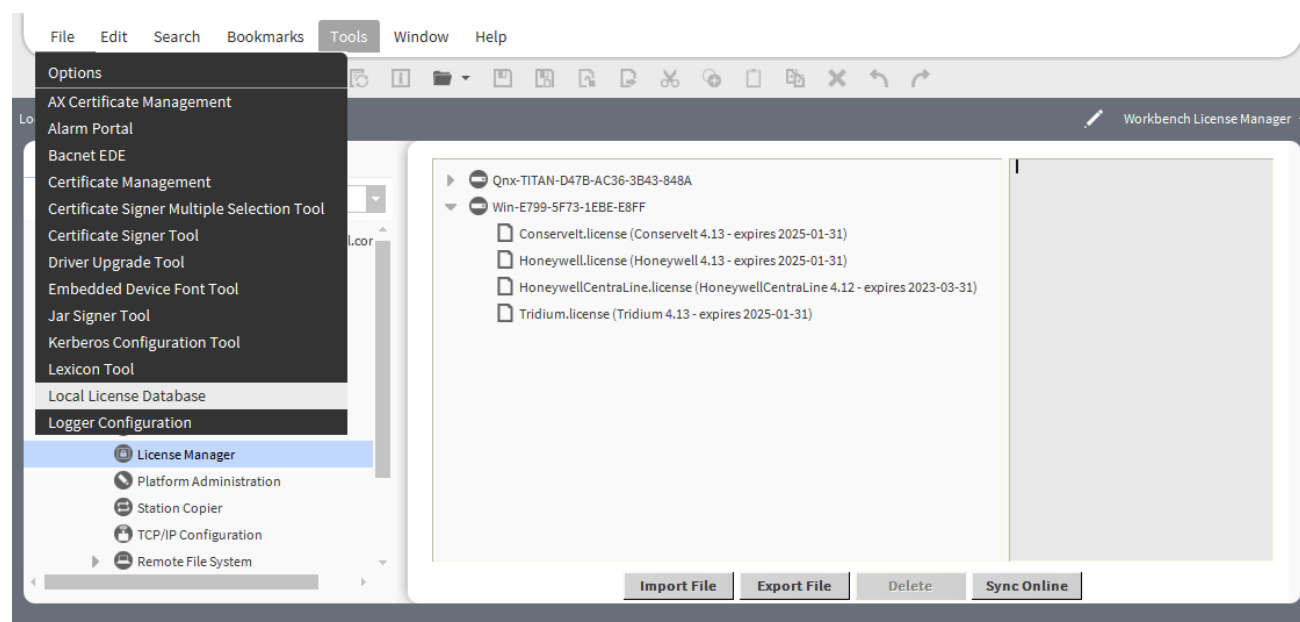
Property descriptions included in the following topics appear as context-sensitive help topics when accessed by:

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

Workbench License Manager (platform-LicenseDatabaseTool)

This component represents your Workbench PC's local license database. This allows you to manage locally-stored licenses.

Figure 23 Workbench License Manager



To access, click **Tools→Local License Database**.

This view provides a two-pane window into all the license files and parent “host ID” folders:

- The left pane provides tree navigation where you can expand folders and click (to select) license files.
- The right pane shows the text contents of any selected license file.

Buttons

Buttons at the bottom of this view provide a way to manage the contents of your local license database, and are described as follows:

- **Import File** always available, this button adds license file(s) from a local license file or license archive (.lar) file.

- **Export File** always available, this button saves all licenses (or any selected licenses) locally, as a license archive file.
- **Delete** deletes selected licenses from the local license database.
- **Sync Online** typically available if you have Internet connectivity, this button updates all licenses (or any selected licenses) in your local license database with the most current versions of the license(s) from the online licensing server.

Chapter 15 Controller components

Topics covered in this chapter

- ◆ Data Recovery Service Editor (`platDataRecovery-DataRecoveryService`)
- ◆ Model-specific `PlatformServiceContainer` properties
- ◆ NTP Platform Service (`platform-NtpPlatformServiceNpsdk`)
- ◆ Platform Service Properties (`platform-NtpPlatformServiceQnx`)
- ◆ Platform Alarm Support (`platform-PlatformAlarmSupport`)
- ◆ System Platform Service (`platform-SystemPlatformServiceQnxJavelina`)
- ◆ System Platform Service (`platform-SystemPlatformServiceQnxNpm6xx`)
- ◆ Tcp/Ip Configuration (`platDaemon-TcpIpConfiguration`)
- ◆ Hardware Scan Service View (`platHwScan-HardwareScanService`)
- ◆ Platform Service Properties (`platMstp-BacnetMstpPlatformServiceQnx`)
- ◆ External SLA Battery (`platPower-ExternalSlaBattery`)
- ◆ Javelina Battery Platform Service (`platPower-JavelinaBatteryPlatformService`)
- ◆ NIMH Battery (`platPower-NimhBattery`)
- ◆ NPM2 NIMH Battery (`platPower-Npm2NimhBattery`)
- ◆ NPM External SLA Battery (`platPower-NpmExternalSlaBattery`)
- ◆ Npm Dual Battery Platform Service (`platPower-NpmDualBatteryPlatformService`)
- ◆ Power Monitor Platform Service (`platPower-PowerMonitorPlatformServiceQnx`)
- ◆ Serial Port Platform Service (`platSerialQnx-SerialPortPlatformServiceQnx`)
- ◆ Serial Port Service (`platSerialQnx-SerialPortQnx`)

Components include services, folders and other model building blocks associated with a module. These components are available only in a remote controller platform.

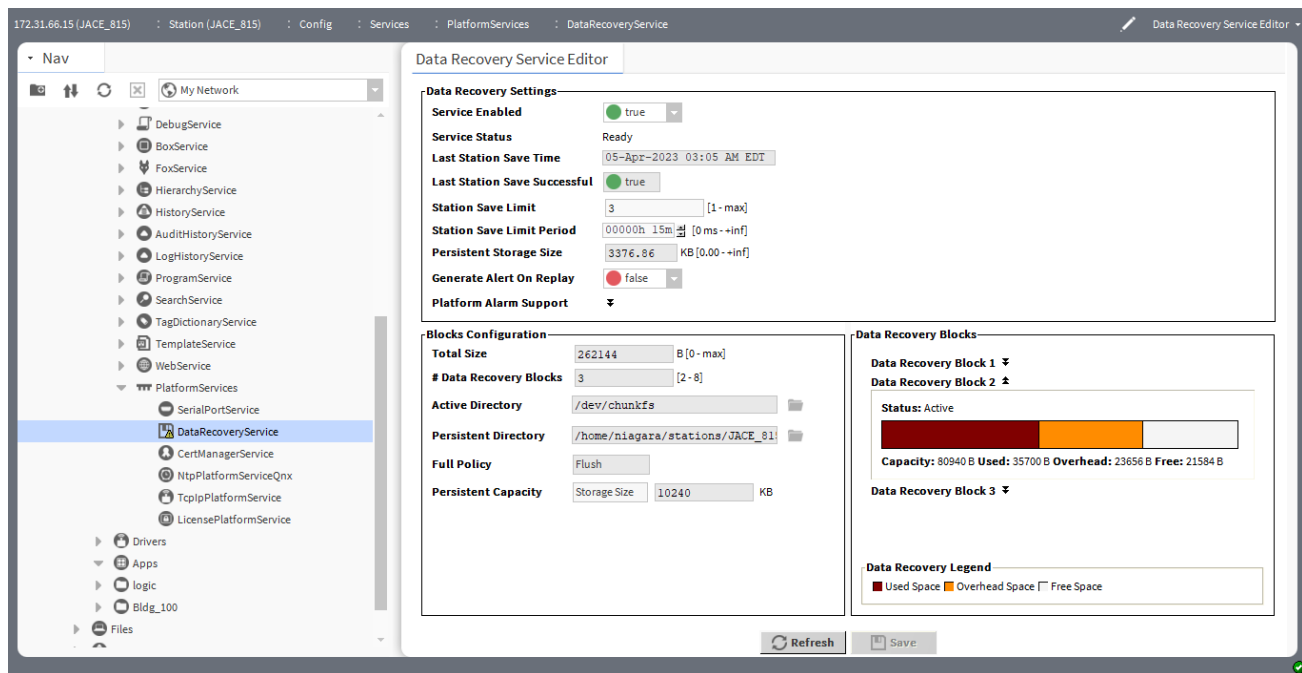
Property descriptions included in the following topics appear as context-sensitive help topics when accessed by:

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

Data Recovery Service Editor (`platDataRecovery-DataRecoveryService`)

This component in the `platDataRecovery` module automatically creates and manages buffers in a controller's available RAM, allowing a controller to function without a battery or to use RAM along with a installed backup battery.

Figure 24 Data Recovery Service Editor view



To access, expand **Config**→**Services**→**Platform Service** and double-click **DataRecoveryService** of JACE.

The controllers with integral RAM include: JACE-9000, JACE-8000, and the following legacy controllers; JACE-3E, JACE-6E, JACE-603, JACE-645) as well as the JACE-6 and JACE-7 with an installed SRAM option card.

For details, see the *Niagara Data Recovery Service Guide*.

Model-specific PlatformServiceContainer properties

Some JACE controller models may have yet more **PlatformServices** properties, specific to special hardware features. This is in addition to the standard and additional properties. Typically, these are configured at commissioning time.

For more details, see the controller-specific **PlatformServices** properties in the *JACE Niagara 4 Install and Startup Guide*.

NTP Platform Service (platform-NtpPlatformServiceNpsdk)

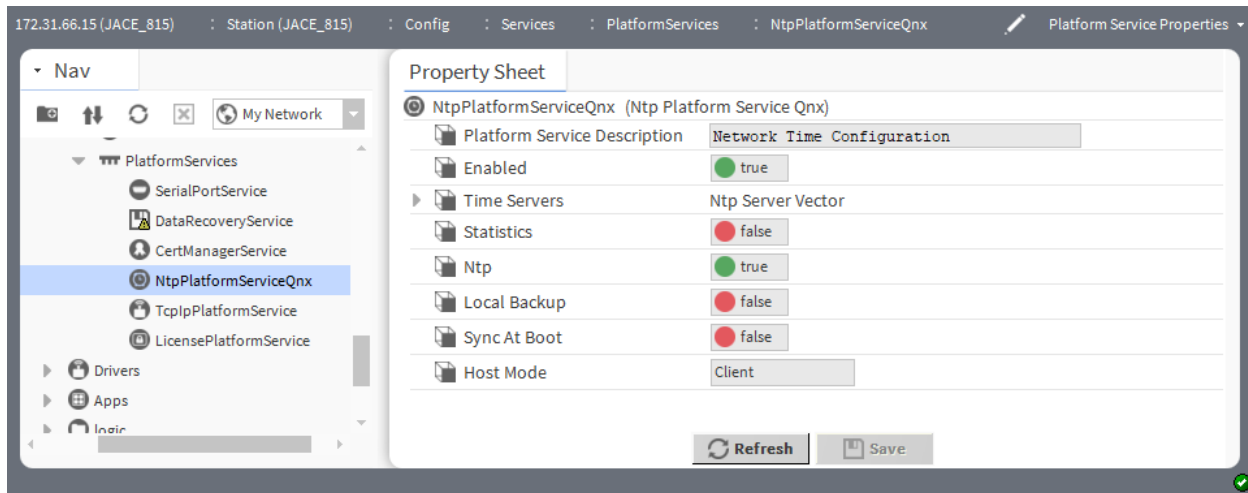
This controller component is the Niagara interface to the NTP (Network Time Protocol) daemon of the Niagara Portability software development kit (Npsdk) running on a controller. If enabled, it provides client and server support for NTP.

Platform Service Properties (platform-NtpPlatformServiceQnx)

This controller component is the Niagara interface to the NTP (Network Time Protocol) daemon of the QNX OS running on a controller. If enabled, it provides client and server support for NTP.

The default view of this platform service is the **Platform Service Properties** view, in which you can adjust a few settings, as well as specify time servers.

Figure 25 Platform Service Properties



To access this view expand **Config**→**Services**→**Platform Services** and right-click **NtpPlatformServiceQnx**→**Views**→**Platform Service Properties**

Property	Value	Description
Platform Service Description	read-only	Reports the type of service.
Enabled	true (default) or false	Enables and disables the use of the Network Time Protocol in the controller. true causes the host to use NTP to sync its clock with time values retrieved from other servers.
Time Servers	additional properties.	Additional properties.
Statistics	read only (defaults to false)	Indicates if statistics are available. true reports information about NTP operation.
Ntp	read only (defaults to true)	Reports if NTP (Network Time Protocol) is enabled.
Local Backup	read only (defaults to false)	Reports how server polling is handled. true indicates that if the specified NTP server(s) become unavailable during a poll, the system clock provides the time used. This prevents the timing of the polling algorithm in the ntpd (which is executed at specified/changing intervals) from being reset. A true value does not change the NTP daemon's polling interval (frequency). In fact, by using the local system clock, the NTP-calculated polling time would remain the same and thus not result in more polling.

Property	Value	Description
Sync At Boot	read only (defaults to <i>false</i>)	Reports if the local system time should be updated at platform boot. <i>true</i> executes the <code>ntpd</code> command to update system local time when the controller boots. This happens before the station starts or the <code>ntpd</code> starts.
Host Mode	read-only	Indicates that this host acts as an NTP client only. The NTP data retrieved by this host from configured servers is not available to local network devices.

Platform Alarm Support (platform-PlatformAlarmSupport)

This controller component is a container slot that appears for each alarmable value under a **Platform Service**, such as the **PowerMonitorService** for many controllers.

For the controller platform, example **PlatformAlarmSupport** components include:

- **Battery Alarm Support** configures how low battery level alarms are handled in the station.
- **Power Alarm Support** configures how AC power loss alarms are handled in the station.

Properties under each **Platform Alarm Support** container are used to designate the station's Alarm Class to be used, and also to populate the alarm record when the specific alarm occurs. These properties work in the same fashion as those in an alarm extension for any control point.

System Platform Service (platform-SystemPlatformServiceQnxJavelina)

This controller component is the QNX implementation of **SystemPlatformService** in a station running on a JVLN-based (JACE-700) controller.

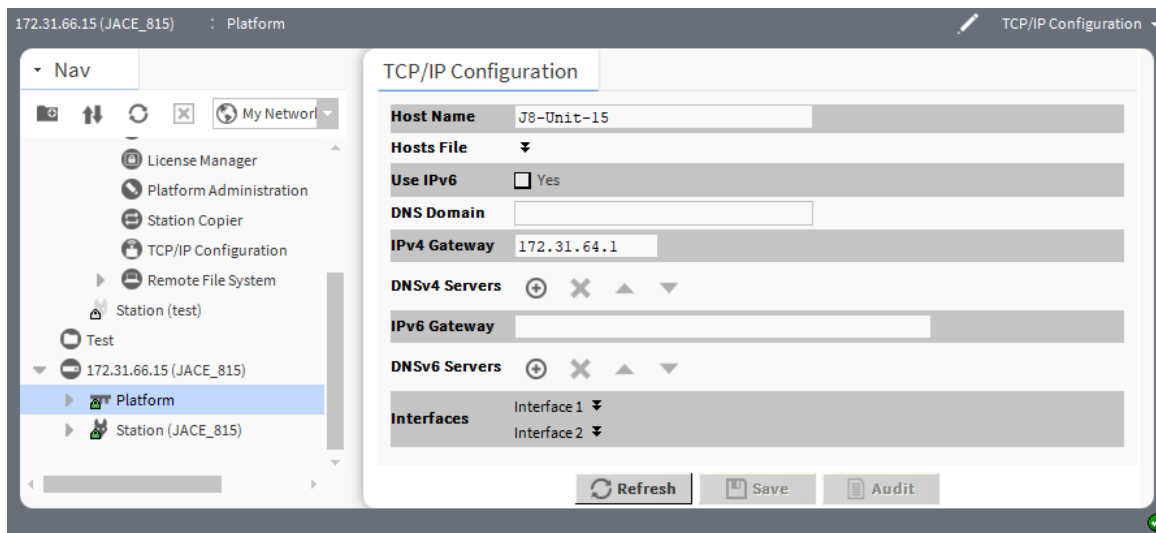
System Platform Service (platform-SystemPlatformServiceQnxNpm6xx)

This controller component is the QNX implementation of **SystemPlatformService** in a station running on the JACE controller.

Tcp/Ip Configuration (platDaemon-TcpIpConfiguration)

TCP/IP Configuration is the platform view you use to configure a remote host's TCP/IP settings. Typically, you make initial settings when you first commission the controller, where this view is one step in the platform's Commissioning Wizard. For Windows platforms, this view is read-only in Niagara 4. For more details, see TCP/IP Configuration.

Figure 26 Tcp/Ip Configuration view



To access this view of the controller expand **Platform** and double-click **TCP/IP Configuration**.

Property	Value	Description
Host name	text	Identifies the name (Id) of the host platform. For a Supervisor PC this is localhost.
Host File	text	Displays the details of host file used.
Use IPv6	check box	If checked uses IPv6.
DNS Domain	text	Defines the name of the network domain, or if not applicable, leave it blank.
IPv4 Gateway	IP address	IPv4 Gateway is the IP address for the device that forwards packets to other networks or subnets. The controller only supports one gateway for all adapters. This includes the JACE-8000 WiFi Adapter in Client mode.
DNSv4 Servers	IP address	Defines the IP addresses for any DNS servers. Separate each with a comma.
IPv6Gateway	IP address	Defines the IPv6 address for the router that forwards packets to other IPv6 networks or subnets.
DNSv6 server	IP addresses	Defines one or more DNSv6 server(s).
Interfaces	drop-down arrow	Displays the details of the interfaces.

Hardware Scan Service View (platHwScan-HardwareScanService)

This optional platform service component is available on the controller station, provided that the platform has the `platHwScan` module installed. This service provides a graphical diagram of communication ports and other features on the hosting platform, including callouts to a table that explains the location, description (such as COM2), port type, and status/usage of each item.

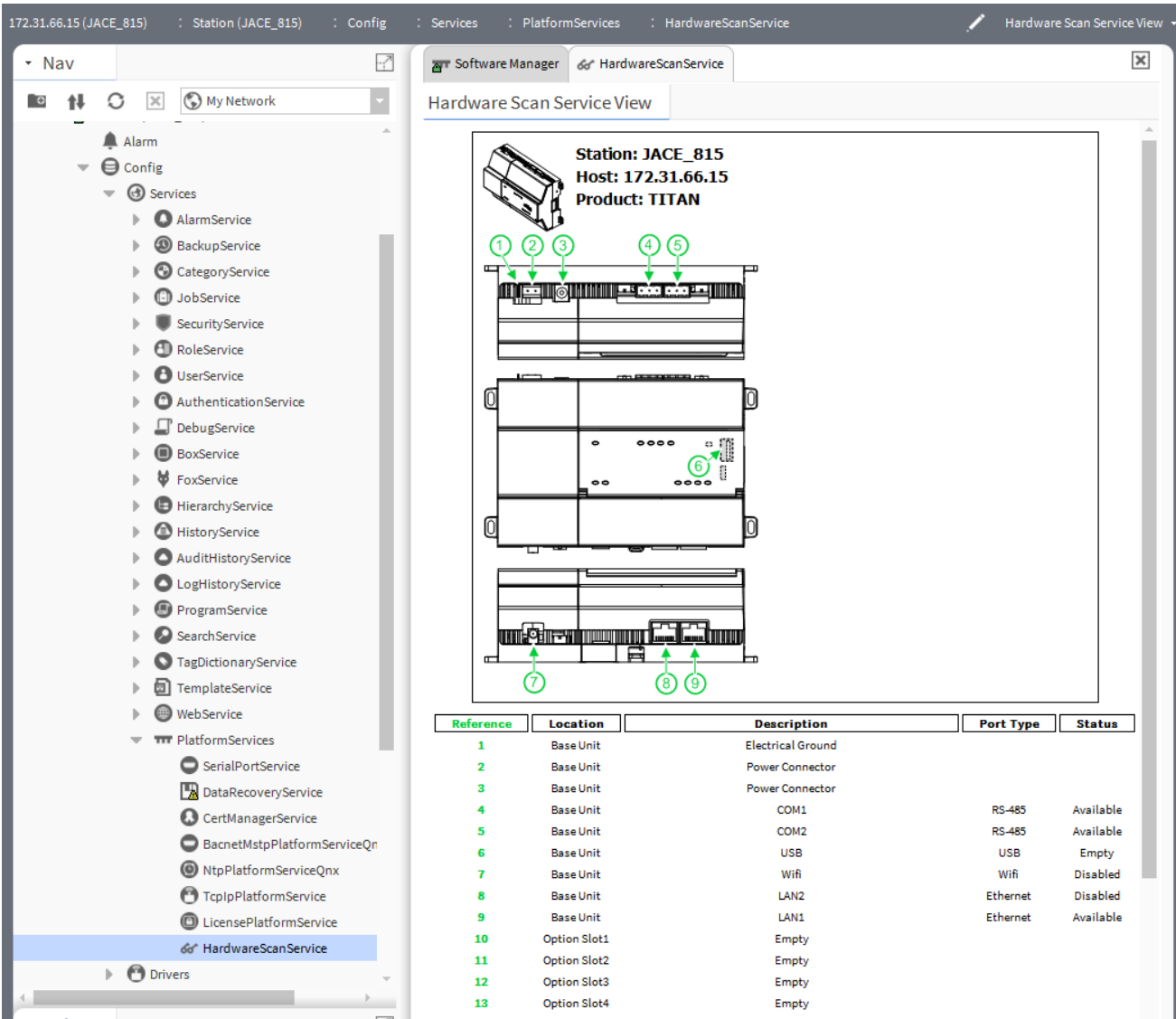
To function correctly, the appropriate `platHwScan<Type>` module needs to be installed on the controller. Otherwise, the default **Hardware Scan Service View** displays:

Jar file `platHwScanType` is required to support this platform

where the appropriate `platHwScan<Type>` is as follows:

Controller Series	platHwScanType module
JACE-6E, JACE-6, JACE-3E	platHwScanNpm
JACE-7	platHwScanJvln
JACE-603 (JACE-403 with retrofit board)	platHwScanJ603
JACE-645 (JACE-545 with retrofit board)	platHwScanJ645
JACE-602 Express (J-602-XPR or M2M)	platHwScanXpr
JACE-8000	platHwScanTitan

Figure 27 Hardware Scan view



To open this view, expand **Config→Services→PlatformServices** and double-click **HardwareScanService**.

This diagram of the controller shows its communication ports and other features including, if applicable, installed communication options, such as modules or cards. The callouts refer to a table that explains each item's location, description (such as COM2), port type, usage, and status.

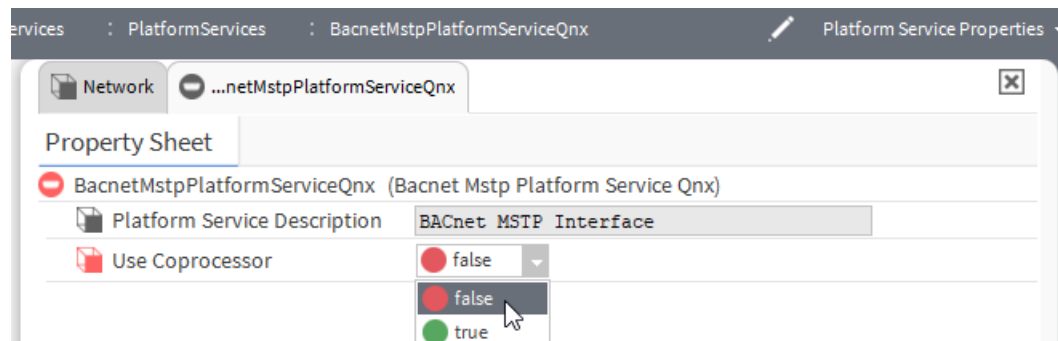
For more information, refer to the *Niagara Engineering Notes*.

Platform Service Properties (platMstp-BacnetMstpPlatformServiceQnx)

This component enhances CPU processing. If you are using multiple MS/TP trunks and large numbers of MS/TP points, your controller CPU usage may be very high, contributing to slow performance. Beginning with Niagara 4.12, QNX-based controllers that are licensed with the `mstp` license use enhanced MS/TP (emstp) processing provided by the CPU's RS-485 co-processor.

Prior to this, processing for MS/TP communication was provided on the controller CPU. With this update, the enhanced MS/TP processing mode is the default condition when using this protocol, however, you can revert to the legacy mode by configuring the **BacnetMstpPlatformServiceQnx** properties. Enhanced MS/TP provides a significant benefit by taking a substantial load off of the CPU, providing overall controller performance improvements. Depending on your configuration, improvements may include increased communications reliability and more responsive graphic views.

Figure 28 Bacnet Mstp Platform Service properties with Use Coprocessor property exposed



This is a hidden component. In most situations, its default settings should be sufficient and preferable. However, to revert to the legacy (CPU-based) processing mode, you can make this component's properties visible in the station's **PlatformServices**. Two other properties actions are available to be exposed.

With some additional configuration, it is possible to capture diagnostic data. To access the statistics related to an MS/TP network port that may be helpful in diagnosing problems, refer to the *Niagara Engineering Notes*.

External SLA Battery (platPower-ExternalSlaBattery)

ExternalSlaBattery () is one of two battery slots in the **JavelinaBatteryPlatformService** in the JACE-700 (JACE-7 series) controller station's **PlatformServices** container. This slot indicates the host JACE platform can use an optional, sealed-lead acid (SLA) battery, in addition to the onboard NiMH backup battery.

Javelina Battery Platform Service (platPower-JavelinaBatteryPlatformService)

This component applies to a station running in the JACE-700 (JACE-7 series) controller. It can monitor primary power status and backup battery levels in both the onboard 12V NiMH battery and an optional 12V sealed-lead acid (SLA) battery.

It can monitor alarm contacts of an external, customer-supplied UPS— if enabled and wired to the two corresponding onboard contact inputs (CIs) of the controller. The JACE-7 controller has three onboard CIs, with the intended use for UPS AC power lost, UPS low battery, and (door) tamper switch.

The tamper switch CI on the JACE-7 controller is enabled and monitored by two properties in the **PowerMonitorService**'s parent **PlatformServices** container).

Configuration properties in this **PowerMonitorService** allow changing the shutdown delay time, and also specifying whether external equipment is connected (12V SLA battery, UPS). Separate alarm source configuration properties are available for all five types of alarms (low NiMH battery level, low SLA battery level, primary power lost, UPS AC power lost, UPS low battery).

The support is enabled and configured at JACE commissioning time. For related details, see JACE power monitoring configuration in the latest *JACE Niagara 4 Install and Startup Guide*.

NIMH Battery (platPower-NimhBattery)

NimhBattery (🔋) is a battery container slot under the **PowerMonitorService** in the JACE-700 (JACE-7 series) station's **PlatformServices** container. This slot indicates the host JACE platform uses a nickel-metal hydride (NiMH) battery. Included are two status properties that show the current State (Idle, Charging, Discharging, Unknown) and Charge Time Left (in hours and minutes, if state is charging).

NPM2 NIMH Battery (platPower-Npm2NimhBattery)

This slot (🔋) indicates that the host controller platform uses a nickel-metal hydride (NiMH) battery. Included are two status properties that show the current State (Idle, Charging, Discharging, Unknown) and Charge Time Left (in hours and minutes, if state is charging). This slot is located under the **PowerMonitorService** or **PlatformServices** container depending on controller type.

This slot also appears in the **NpmDualBatteryPlatformService** (dual battery **PowerMonitorService**) of the controller that is capable and enabled for dual battery support.

NPM External SLA Battery (platPower-NpmExternalSlaBattery)

NpmExternalSlaBattery (🔋) is one of two battery slots under the **NpmDualBatteryPlatformService** in a dual battery enabled JACE's station's **PlatformServices** container. This slot indicates that the host JACE platform can use an optional, sealed-lead acid (SLA) battery, in addition to the onboard NiMH backup battery.

Npm Dual Battery Platform Service (platPower-NpmDualBatteryPlatformService)

■ **NpmDualBatteryPlatformService** (**PowerMonitorService**) applies to a station running in a remote host platform that is capable and enabled for dual battery support. It is used to monitor primary power status and backup battery levels in both the onboard NiMH battery as well as the optional sealed-lead acid (SLA) battery. A few configuration parameters allow changing the shutdown delay time, as well as alarm source configuration for all three types of alarms (low NiMH battery level, low SLA battery level, primary power lost).

Typically, support is enabled and configured at JACE commissioning time. For related details, see "JACE power monitoring configuration in the latest *Install and Startup Guide*.

Power Monitor Platform Service (platPower-PowerMonitorPlatformServiceQnx)

■ This component monitors the primary power status and backup battery level in many controllers. It applies to legacy platforms and not JACE-8000 or JACE-9000 controllers.

This **PowerMonitorService** is found under the **PlatformServices** container in a station running on many controllers except for those models that are capable and/or enabled for dual battery support. A few configuration properties allow you to change the shutdown delay time, as well as alarm source configuration for both types of alarms (low battery level, primary power lost). Typically, support is enabled and configured at controller commissioning time.

An SRAM-equipped controller can be configured for battery-less operation (the `platDataRecovery` module must be installed, and the controller licensed for the dataRecovery feature). The **PowerMonitorService** continues to monitor for an (optional) backup battery, and upon loss of AC power allows continuous operation on battery power until the **Shutdown Delay** time is reached. This happens unless you set the **Battery Present** property (of controller's **PlatformServiceContainer**) from `true` (the default) to `false`. This disables backup battery support and, when there is no backup battery, prevents ongoing battery bad nuisance alarms.

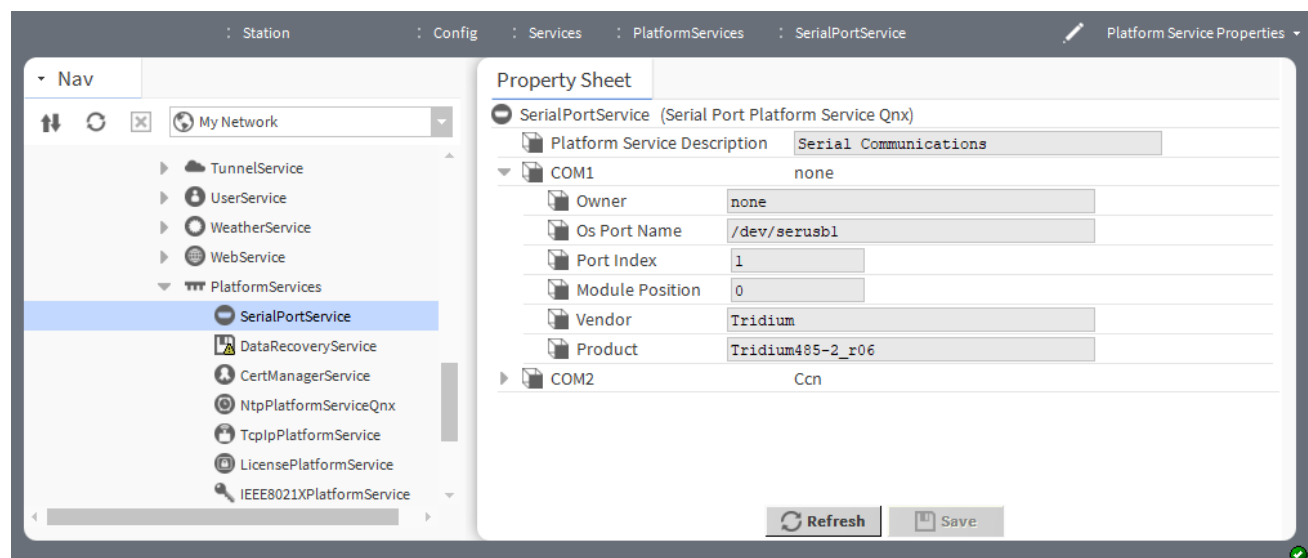
Property	Value	Description
Primary Power Present	<code>true</code> or <code>false</code>	Reports if the controller is being powered by AC power (<code>true</code>) or not (<code>false</code>).
Battery Good	<code>true</code> or <code>false</code>	Reports if the last controller test of the NiMH backup-battery was good (<code>true</code>) or not (<code>false</code>).
Time of Last Test	timestamp	Reports when the battery was tested last.

For related details, refer to the latest applicable *Install and Startup Guide*.

Serial Port Platform Service (platSerialQnx-SerialPortPlatformServiceQnx)

This component is the remote platform's interface to its serial port configuration. This service is found under the running station's **PlatformServices** container as the **SerialPortService**. It allows for the review of available serial ports on the host platform.

Figure 29 Platform Service Properties



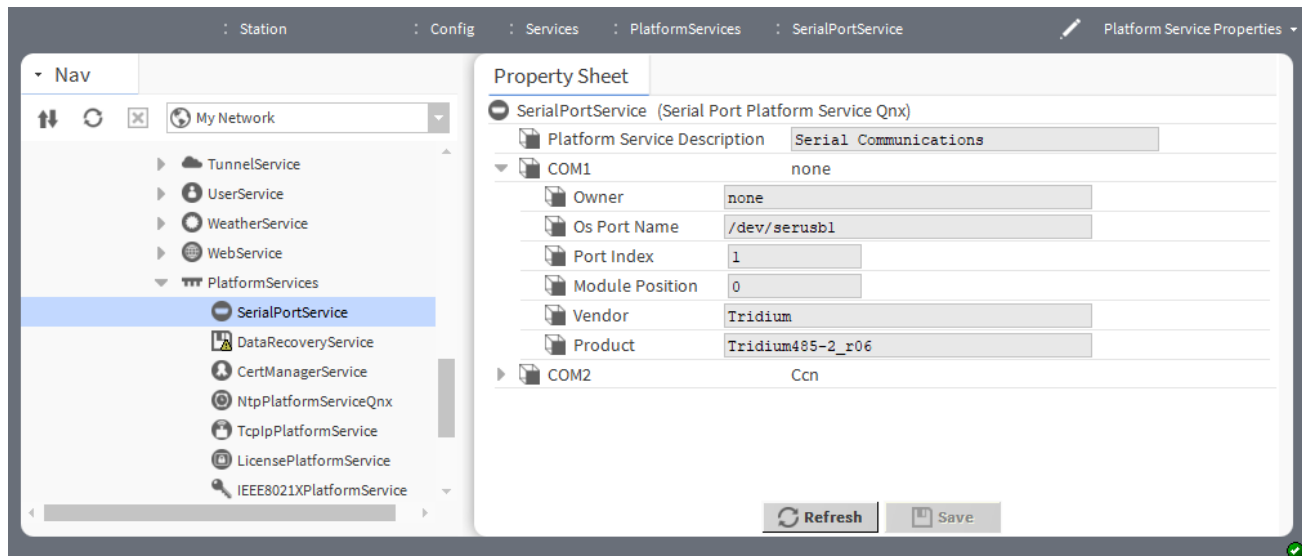
To access this view expand the running station, **Config**→**Services**→**PlatformServices** and double-click **SerialPortService**.

Properties	Value	Description
Platform Service Description	read-only	Reports the type of communication.
Owner	read-only	Reports the driver network or function currently associated with that COM port, for example, <code>NrioNetwork</code> , <code>dialup</code> , <code>none</code> , <code>ModbusAsyncNetwork</code> , or <code>dbgjmp</code> (latter indicated for COM1 when the serial shell jumper is installed on the controller).
Os Port Name	read-only	Reports how the port is known to the QNX OS and associated low-level drivers.
Port Index	read-only	Configures the unique serial port index number, starting with 1 for COM1.
Module Position	read-only	Reports a number that indicates the position of the module in relationship to the controller.
Vendor	read-only	Always displays Tridium.
Product	read-only	Reports product details.
Enable Port Reset on Error	<code>true</code> (default) or <code>false</code>	Controls when to reset the port. <code>true</code> resets the port after any error.

Serial Port Service (platSerialQnx-SerialPortQnx)

This component contains properties that describe how a serial port (RS-232 or RS-485) on the controller is being used by the software as COMn.

Figure 30 Serial Port Service properties



To access this view expand the running station, **Config→Services→PlatformServices** and double-click **SerialPortService**.

Property	Value	Description
Owner	read-only	Reports the driver network or function currently associated with that COM port, for example, <code>NrioNetwork</code> , <code>dialup</code> , <code>none</code> , <code>ModbusAsyncNetwork</code> , or <code>dbgjmp</code> (latter indicated for COM1 when the serial shell jumper is installed on the controller).
Os Port Name	read-only	Reports how the port is known to the QNX OS and associated low-level drivers.
Port Index	read-only	Configures the unique serial port index number, starting with 1 for COM1.
Module Position	read-only	Reports a number that indicates the position of the module in relationship to the controller.
Vendor	read-only	Always displays Tridium.
Product	read-only	Reports product details.
Enable Port Reset On Error	<code>true</code> (default) or <code>false</code>	Controls when to reset the port. <code>true</code> resets the port after any error.

Chapter 16 Shared components

Topics covered in this chapter

- ◆ Nav Container View (platCrypto-DaemonSecureSession)
- ◆ Station Text Summary Editor view (platDaemon-StationTextSummaryEditor)
- ◆ Remote File System (platform-DefaultDaemonFileSpace)
- ◆ Daemon Session (platform-DaemonSession)
- ◆ SyslogPlatformService (platform-PlatformServiceProperties)
- ◆ SystemMonitorService -(systemMonitor-SystemMonitorService)

Components include services, folders and other model building blocks associated with a module. These components are available in both a Supervisor PC and remote controller platform.

Property descriptions included in the following topics appear as context-sensitive help topics when accessed by:

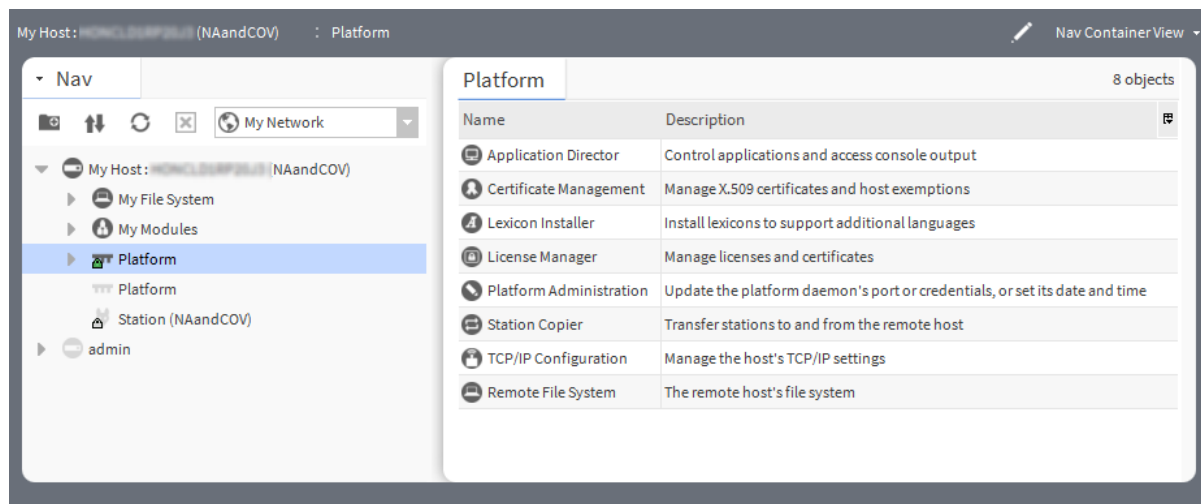
- Right-clicking on the object and selecting **Views→Guide Help**
- Clicking **Help→Guide On Target**

Nav Container View (platCrypto-DaemonSecureSession)

This component represents a Workbench secure platform connection to a host.

As in a regular (un-encrypted) platform connection, the default view is the **Nav Container View**, which provides a table of all the various platform views. This view contains additional functions when connected to a remote platform.

Figure 31 Nav Container View



To access this view make a platform connection to a host and double-click on **Platform**.

The platform session icon () is labeled **Platform**, shows a small padlock, and is directly under the host for the platform session that is in progress. To support such connections, the host must have its **Platform TLS Settings** enabled (accessed in its **Platform Administration** view).

Platform objects

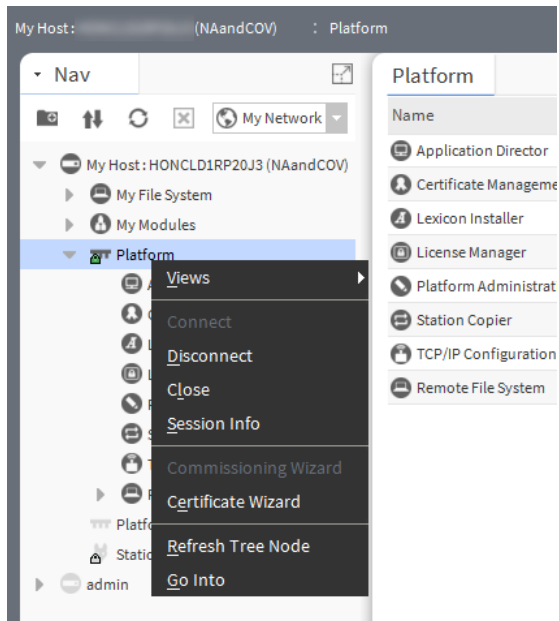
These objects provide platform functions:

- **Application Director** starts, stops, restarts, kills and configures auto-start and restart on failure for a station that is running on a platform. The output from the station that displays in the view pane is useful for monitoring and troubleshooting.
- **Certificate Management** manages signed PKI certificates in the platform's key and trust stores for TLS secure communication. Refer to *Niagara Station Security Guide*.
- **Distribution File Installer** restores a backup distribution (.dist) file to the target platform, or installs a clean .dist file to wipe the file system of a controller to a near-factory minimum state. This view is available when connected to a remote host.
- **File Transfer Client** copies files between your Workbench PC and a remote platform (in either direction). For example, you use this platform view when editing a controller's `system.properties` file—once to copy it from the controller to your Workbench PC (for local editing), then afterwards to copy it back to the remote controller. This view is available when connected to a remote host.
- **Lexicon Installer** installs file-based lexicon set from your Workbench PC to a remote platform, to provide non-English language support, or to customize the English display of selected items. In Niagara 4, usage of this view and file-based lexicons may be atypical.
- **License Manager** reviews, installs, saves, and deletes licenses and (license) certificates on the remote platform.
- **Platform Administration** configures, provides status, and enables the troubleshooting of the platform daemon. Included are commands to change the time and date, back up all remote configurations, reboot the host platform, modify platform users, specify the TCP port monitored by the platform daemon, and change various settings for a secure (TLS) platform connection.
- **Software Manager** reviews, installs, updates, and uninstalls Niagara modules (.jars) on the remote platform. It compares modules installed on the connected platform against those available (locally) in **Sys Home** of your Workbench PC. This view is available when connected to a remote host.
- **Station Copier** installs (copies) a station from your Workbench User Home to a remote platform (or if a Supervisor, to the local PC's daemon User Home), backs up (copies) a station to your Workbench User Home, renames and deletes a remote station.
- **TCP/IP Configuration** reviews and configures the TCP/IP settings for the network adapter(s) of the platform.
- **Remote File System** provides read-only access to folders and files on the remote platform, including all those under its system home (Sys Home) and daemon User Home.

Platform actions

Right-clicking the **Platform** node in the Nav tree opens a list of platform actions.

Figure 32 Platform actions



- **Views** opens a list of the platform functions.
- **Connect** opens the credentials pages for connecting to the selected platform.
- **Disconnect** removes the connected to the selected platform.
- **Close** terminates the current platform session.
- **Session Info** displays information about the current session.
- **Commissioning Wizard** opens the wizard used to commission the new remote controller platform.
- **Certificate Wizard** opens the wizard used to create a new root CA certificate.
- **Refresh Tree Node** renews the Nav tree.
- **Go Into** closes the Nav tree.

Station Text Summary Editor view (platDaemon-StationTextSummaryEditor)

This component enables the export of daemon and station console output and thread dumps for the purpose of troubleshooting.

Figure 33 Station Text Summary Editor view

```

temp4530621332776891267.txt - Notepad
File Edit Format View Help

Console output for station test
-----
at org.eclipse.paho.client.mqttv3.MqttAsyncClient.createNetworkModules(MqttAsyncClient.java:642)
at org.eclipse.paho.client.mqttv3.MqttAsyncClient.connect(MqttAsyncClient.java:748)
at org.eclipse.paho.client.mqttv3.MqttAsyncClient.connect(MqttAsyncClient.java:708)
at com.tridium.mqttClientDriver.clients.jwt.GcpMqttClient.connect(GcpMqttClient.java:133)
at com.tridium.mqttClientDriver.authenticator.gcp.BGcpAuthenticator.connect(BGcpAuthenticator.java:
at com.tridium.mqttClientDriver.BAbstractMqttDevice.doConnect(BAbstractMqttDevice.java:545)
at auto.com.tridium_mqttClientDriver_BAbstractMqttDevice.invoke(AutoGenerated)
at com.tridium.sys.schema.ComponentSlotMap.invoke(ComponentSlotMap.java:1909)
at com.tridium.sys.engine.EngineUtil.doInvoke(EngineUtil.java:62)
at javax.baja.sys.BComponent.doInvoke(BComponent.java:1268)
at javax.baja.util.Invocation.run(Invocation.java:47)
at javax.baja.util.Worker.process(Worker.java:168)
at javax.baja.util.Worker$Processor.run(Worker.java:141)
at java.lang.Thread.run(Thread.java:750)

WARNING [11:23:15 03-May-23 IST][abstractMqttDriver.GcpAuth] Client: did not connect.
WARNING [11:23:15 03-May-23 IST][abstractMqttDriver.device] MqttException
INFO [11:23:31 03-May-23 IST][fox] Opened: cb4cc806e411113883fb0bf72cf3107b01b7c4b9bffe9088f4eedc26
WARNING [11:23:44 03-May-23 IST][abstractMqttDriver.GcpAuth] The RSA keys has expired, hence cannot
WARNING [11:23:44 03-May-23 IST][abstractMqttDriver.device] RSA keys for authentication has expired
WARNING [11:23:45 03-May-23 IST][NetworkModuleService] null: URI=ssl://:1883
java.security.AccessControlException: access denied ("java.lang.reflect.ReflectPermission" "suppress
at java.security.AccessControlContext.checkPermission(AccessControlContext.java:472)
at java.security.AccessController.checkPermission(AccessController.java:886)
at java.lang.SecurityManager.checkPermission(SecurityManager.java:549)
at java.lang.reflect.AccessibleObject.setAccessible(AccessibleObject.java:128)
at org.eclipse.paho.client.mqttv3.internal.NetworkModuleService.setURIField(NetworkModuleService.ja
at org.eclipse.paho.client.mqttv3.internal.NetworkModuleService.applyRFC3986AuthorityPatch(NetworkM
at org.eclipse.paho.client.mqttv3.internal.NetworkModuleService.createInstance(NetworkModuleService

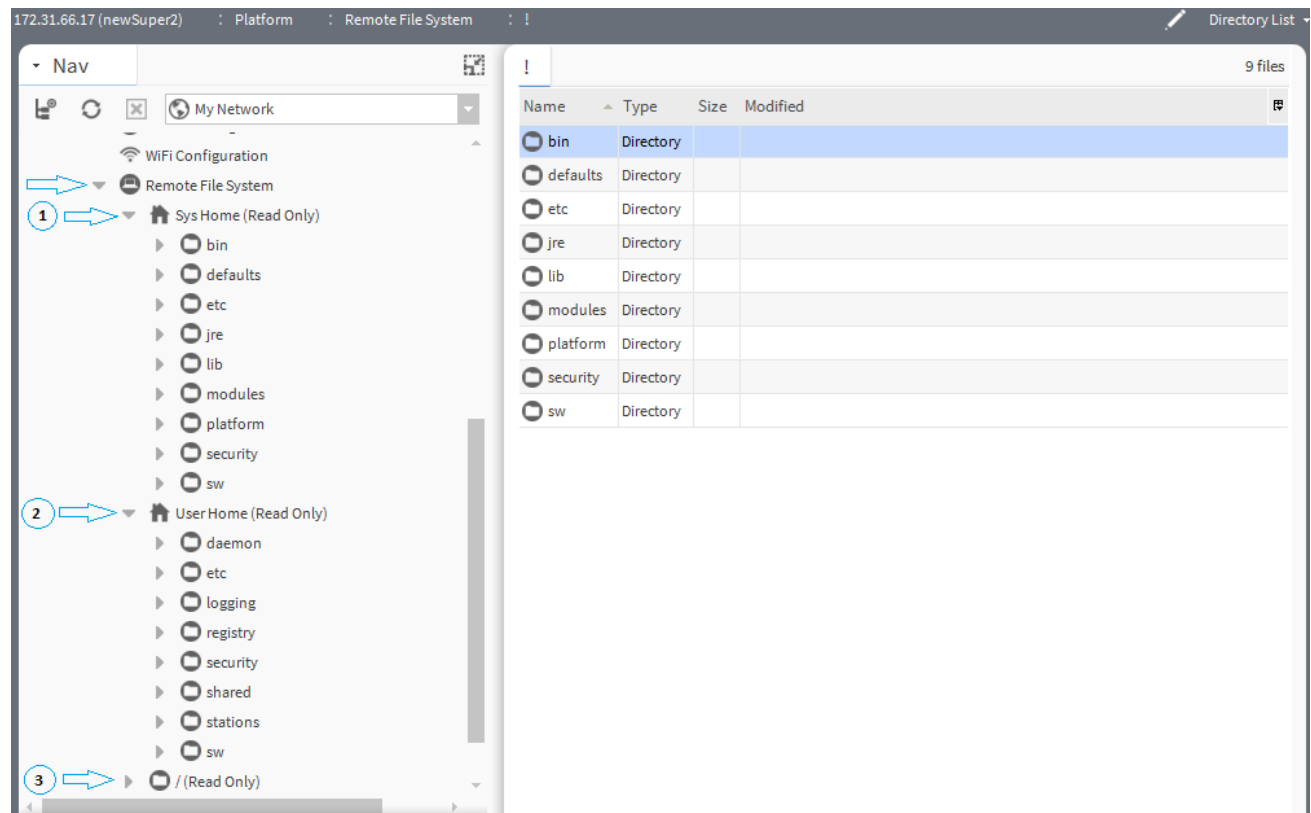
```

To export these data, expand **Platform**→**Application Director**, click the **Export** tool bar icon () , select **Setup** tab and click **OK**.

This provides the platform's daemon and station details. The screen capture shows the exported data as they appear in Notepad.

Remote File System (platform-DefaultDaemonFileSpace)

This PC and controller component is in the **program** module. The **Remote File System** view is one of several platform views. It provides a read-only view of the remote platform's file system.

Figure 34 Remote File System for a controller platform

To open this view, connect to the platform, expand **Platform** and double-click **Remote File System**.

The **Remote File System (DefaultDaemonFileSpace)** represents the files that are accessible for read-only access when a platform-is connected to a remote host. As needed, you can expand folders and examine and/or copy files to your local computer. Included in the Nav tree under the **Remote File System** are main nodes for:

- The system home (**Sys Home**) root folder, under which all installation/runtime files are installed.
- The user home (**User Home**) root folder for the platform daemon, under which all configuration files are stored.
- (controllers only) The root folder for the entire file system, with browse capability.

To edit or write files on the remote Niagara platform, you use the platform's **File Transfer Client**.

Daemon Session (platform-DaemonSession)

This controller component represents a platform connection made in Workbench to a controller host that is not secure.

To access this component, expand **My Host** and double-click **Platform**, then double-click any of the platform objects. Each opens this component.

In the Nav tree view, the daemon session icon () is labeled **Platform**, and is directly under the host for which the platform session is in progress.

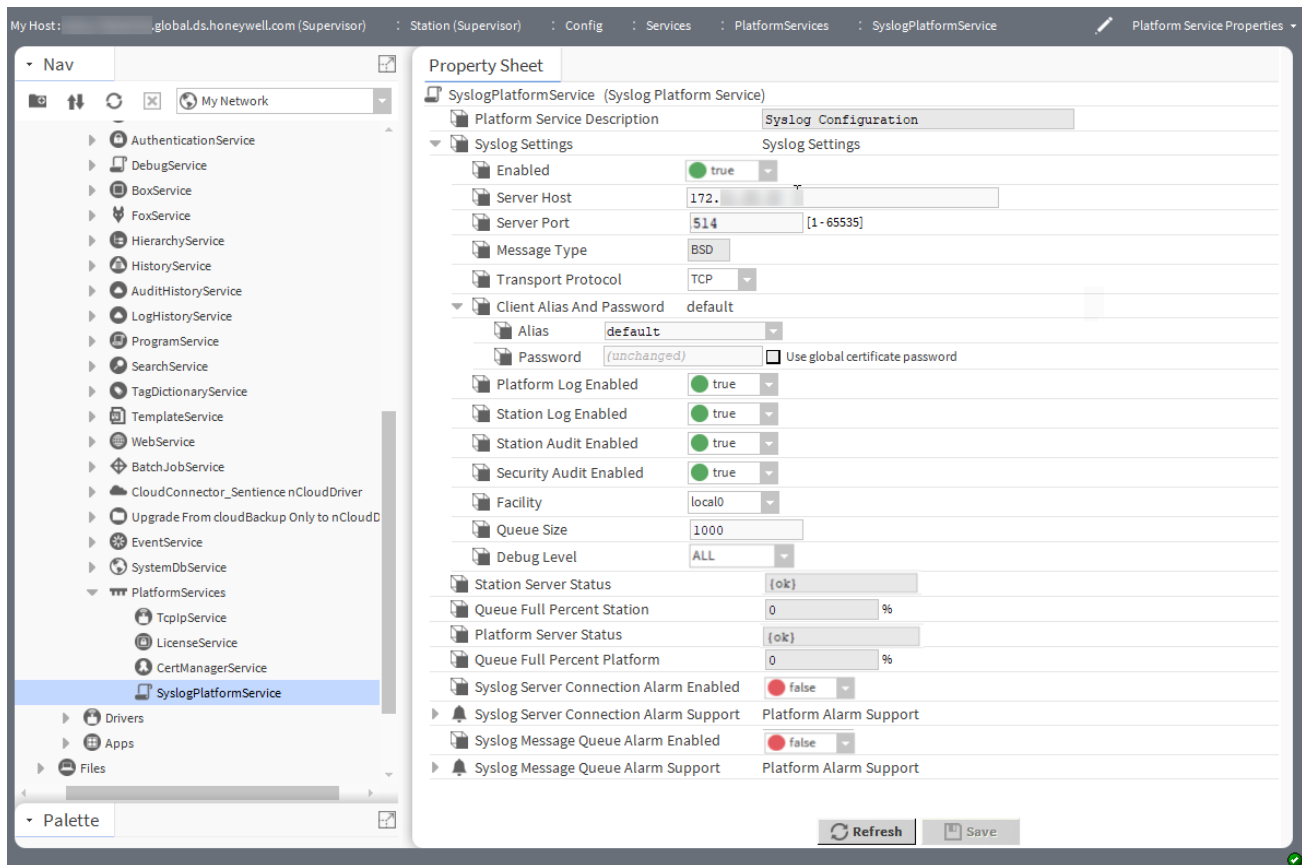
The default view is the **Nav Container View**, which provides a list of all the platform object views.

SyslogPlatformService (platform-PlatformServiceProperties)

As of Niagara 4.13 **Syslog Configuration** is available in **PlatformServices**. Syslog is a standard protocol for message logging, which allows messages generated by Niagara to be stored and analyzed on a remote server.

Niagara can send its platform and station log messages, audit, and security audit messages to a syslog server. Using the BSD message format, messages can be sent over UDP, TCP, or TLS.

Figure 35 Syslog Configuration window



To access this view, expand **Config**→**Services**→**PlatformServices** and double-click **SyslogPlatformService**.

Property	Value	Description
Platform Service Description	read-only	Displays the name of the service.
Syslog Settings	additional properties	Configures additional parameters. Refer to Syslog Settings section below.
Station Server Status	read-only (disabled)	Displays the status of last station log message sent to the syslog server.
Queue Full Percent Station	read-only (defaults to 0%)	If connection is down, percentage of queue of station message used.
Platform Server Status	read-only (disabled)	Displays the status of last platform log message sent to the syslog server.

Property	Value	Description
Queue Full Percent Platform	read-only to 0%)	If connection is down, percentage of queue of platform message used.
Syslog Server Connection Alarm Enabled	true (default) or false	If set to true, it generates alarms on connection failure (only for TCP and TLS).
Syslog Server Connection Alarm Support	additional properties	Configures additional parameters to generate alarms.
Syslog Message Queue Alarm Enabled	true (default) or false	If set to true, it generates alarms on queue full.
Syslog Message Queue Alarm Support	additional properties	Configures additional parameters to generate alarms.

Syslog Settings

Type	Value	Description
Enabled	false (default) or true	Disables (false) or enables (true) the system log service.
Server Host	IP address	Specifies the IP address or hostname of the Syslog Server.
Server Port	number (defaults to 1514)	Specifies the port for communication.
Message Type	read-only	Specifies the type of message supported. Currently only the BSD type is supported.
Transport Protocol	drop-down list	Specifies the transport protocol used for communicating messages to the server.
Client Alias And Password	text	This is only required if the syslog server requires mutual TLS (mTLS) protocol. This property defines the client certificate in the User Key Store to use. Refer in Niagara Station Security Guide to "Creating a Client Certificate for Syslog configuration" for more information on generating Client Certificates.
Platform Log Enabled	true (default) or false	Enables (true) or disables (false) the platform logs sent to the server.
Station Log Enabled	true (default) or false	Enables (true) or disables (false) the station logs sent to the server.
Log Level Filter	Off, Severe, Warning, Info, Config, Fine, Finer, Finest, All (defaults to Info)	Sets the minimum level of platform and station logs that will be sent to the syslog server.
Station Audit Enabled	true (default) or false	Enables (true) or disables (false) the station audit records sent to the server.

Type	Value	Description
Security Audit Enabled	true (default) or false	Enables (true) or disables (false) the security audit records sent to the server.
Facility	drop-down list (defaults to local0)	Specifies the facility (or process) which generated the syslog messages.
Queue Size	number (defaults to 1000)	Specifies the queue size to hold the messages until they are sent.

SystemMonitorService -(systemMonitor-SystemMonitorService)

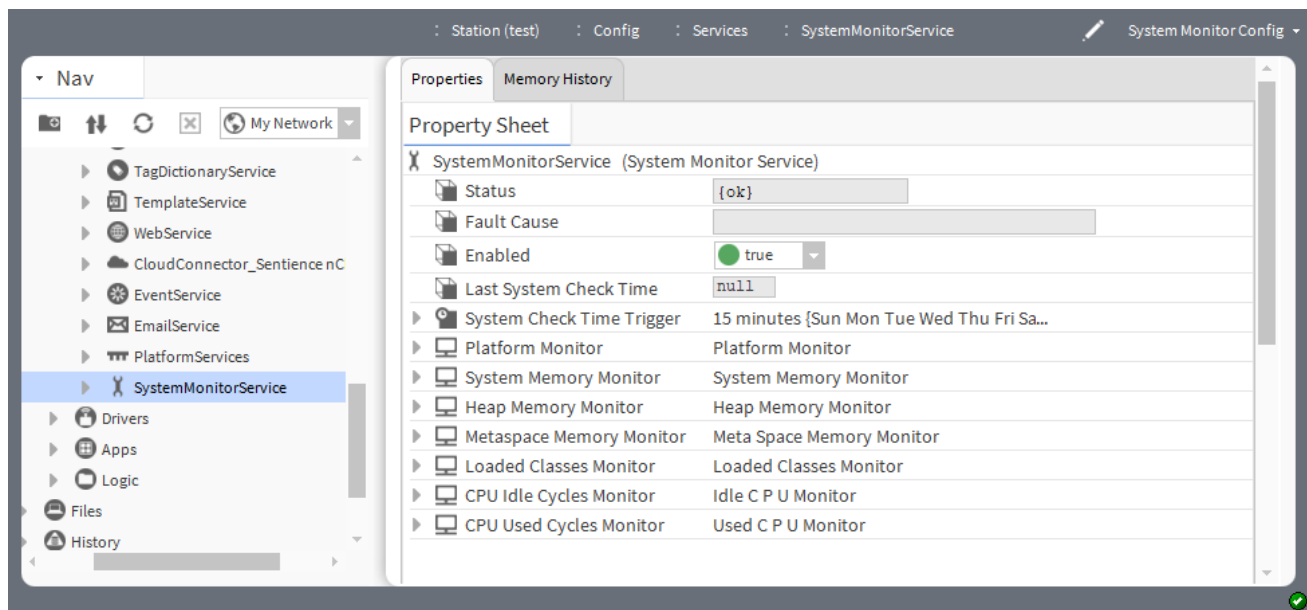
This service monitors current memory and CPU usage in your **NiagaraStation** to detect memory leaks, determine resource high water marks and track daemon availability.

You can use the service when developing applications to make sure that there are sufficient resources to maintain your application over time. The information obtained from this service can, in some cases, be translated to histories for tracking over time and can cause the generation of alarms to monitor specific error conditions or limits.

Using the System Monitor Service is not recommended in every production environment since it consumes resources (in the form of histories and heap memory). Instead, it is best used on a single device in a population as a sample. It can be useful for confirming memory leaks on a platform where you suspect one might exist, or to determine high water marks for the NRE Configuration.

NOTE: The service is best used in moderation, when you already expect that there might be a memory problem. Once you have properly tuned the system you should remove the service from the station.

Figure 36 System Monitor Service properties



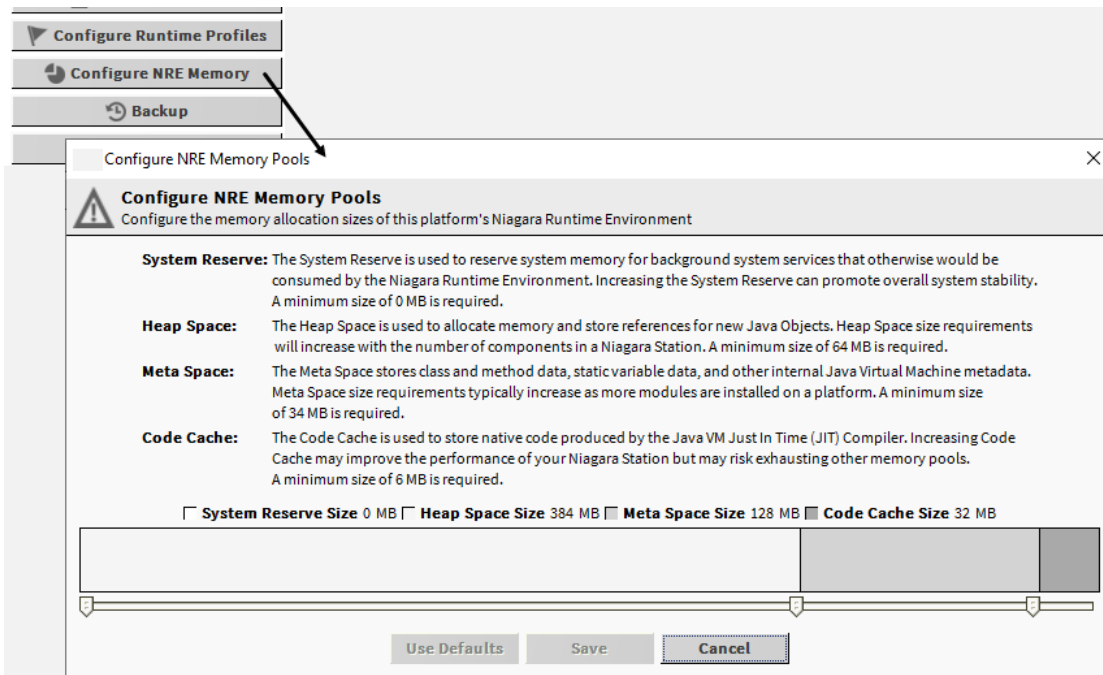
To access, expand **Config→Services** and double-click **SystemMonitorService**.

This service and the monitors it provides are mostly implemented in Java and can be used on both Supervisor and controller platforms, both Niagara products and Niagara Portability Software Development Kit (NPSDK) products. However, the **Socket State Monitor** is used specifically with Tridium JACE devices. The **RAM Disk Monitor** is only for embedded devices that uses a RAM disk for alarm and history records. The **Socket State Monitor** is further specific to the Tridium JACE since it is tied to the netstat, an application that is only available through the Tridium spy.

While these monitors are valid for Supervisors, they are probably of lesser value in those environments since resources on Supervisor platforms are expected to exist in abundance. For example, it might be of little value to monitor something like the System Memory on a Supervisor platform because it may have multiple gigabytes of space. However, you can elect to monitor this if you choose.

The **System Monitor Service** may be of particular value in any embedded controller, including third-party controllers, that elect to use the **NRE Configuration View**, to control the size of various memory pools.

Figure 37 Platform Administration tool, the NRE Configuration View



When dealing with the pools defined here, space not actually used is space wasted. If you can use the associated monitor (that is., **Heap Memory Monitor**, **Meta Space Memory Monitor**, **Code Cache Memory Monitor** or **RAM Disk Monitor**) to determine that you are not actually using the amount of space you reserved, then you can confidently lower that pool's size as a means of providing memory for another pool. For instance, if the **Meta Space Memory Monitor** shows the allotted memory is only half used in a long running station, then you can safely reduce the size of that pool to free up memory for the Heap Space or System Reserve.

For more details, see documentation that is in the NPSDK.

- **Java Heap Space** is used to allocate memory and store references for new Java Objects. Heap Space size requirements will increase with the number of components in a station.
- **Java Meta Space** stores class and method data, static variable data, and other internal Java Virtual Machine metadata. Meta Space size requirements typically increase as more modules are installed on a platform.
- **Java Code Cache** is used to store native code produced by the Java VM Just In Time (JIT) Compiler. Increasing Code Cache may improve the performance of your station but may risk exhausting other memory pools.
- **RAM Disk** stores station alarm and history data. RAM Disk size requirements increase with the number of history and alarm records in a station.
- **System Reserve** is used to reserve system memory for background system services that otherwise would be consumed by the Niagara Runtime Environment. Increasing the System Reserve can promote overall system stability.

Usage notes

When added to station Services, the System Monitor Service is enabled by default. For any of the monitors provided by the service, there is no enabled state. The monitors are either present and collecting or absent. If you do not wish to monitor a particular pool, delete the corresponding monitor from the service.

In general, configurable properties for any monitor are context specific to the monitor and self-explanatory. Exceptions are the **Generate Alarm** and **Log Memory to History** properties, which are described in the following sections.

Generate alarm

For any monitor that supports limit properties like **Minimum System Memory Limit** (System Memory Monitor) or **Minimum Heap Memory Limit** (Heap Memory Monitor), the **Generate Alarm** boolean indicates that an alarm should generate when this limit is crossed. For example, if Free Heap Memory were to drop below 512K (or other specified number) when polled, an alarm should be generated to indicate that the event occurred.

The generated alarm uses the relevant alarm information (Alarm Source Info, etc.) associated with any particular monitor.

The **Generate Alarm** action for the different monitors works as follows:

- Platform Monitor – causes an alarm if the 3011 port is no longer responding to traffic (that is, the Niagara daemon has exited or is frozen)
- System Memory Monitor – causes an alarm to fire if the Free System Memory drops below the specified limit.
- Heap Memory Monitor – causes an alarm to fire if the Free Heap Memory drops below the specified limit.
- Metaspace Memory Monitor – is not used
- Loaded Classes Monitor – is not used
- CPU Idle Cycles Monitor – is not used
- CPU Used Cycles Monitor – is not used
- Code Cache Memory Monitor – is not used
- RAM Disk Monitor – causes an alarm to fire if the RAM disk's freespace falls below 5%.
- Socket State Monitor – causes an alarm to fire if the monitor detects any socket from the `spy:/platform/diagnostics/netstat -A` to be in the `CLOSE_WAIT` state. This is only valid for Tridium JACEs.

Log Memory to History

The history for any particular monitor follows the rules as defined in the monitor's **History Config**, allowing you to configure retention policies, capacity, tags, etc., as with any history extension. The intervals of these configs should be kept at the default so that the history is always generated as a consequence of the **Check System** action.

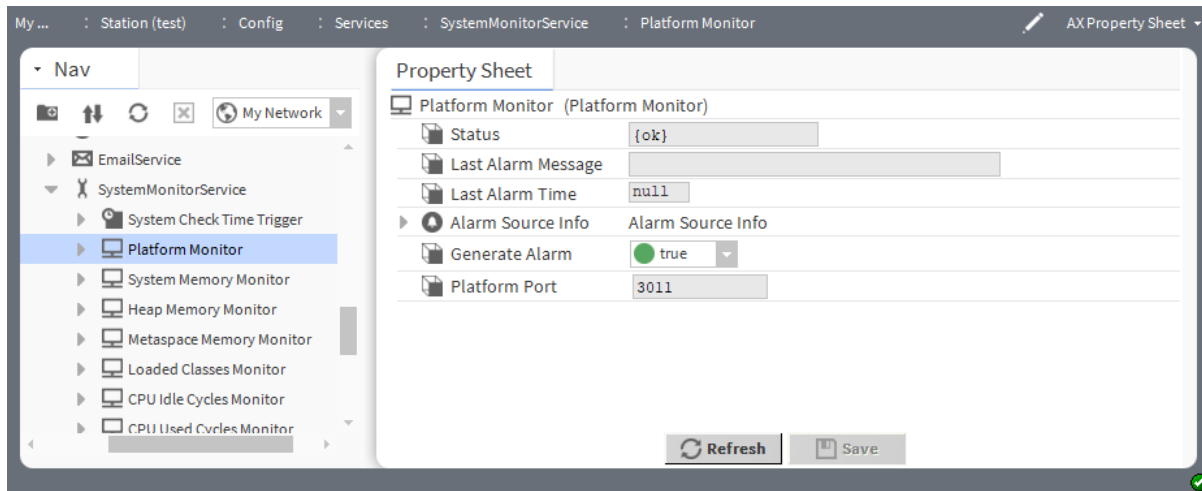
The **Log Memory to History** action for the different monitors works as follows:

- Platform Monitor – is not implemented
- System Memory Monitor – trends Free System Memory for each fire of the **System Check Time Trigger**.
- Heap Memory Monitor – trends Free Heap Memory.
- Metaspace Memory Monitor – trends Used Metaspace Memory.
- Loaded Classes Monitor – trends Java Classes loaded.
- CPU Idle Cycles Monitor – trends Idle CPU Cycles over the last collection period.
- CPU Used Cycles Monitor – trends Used CPU Cycles over the last collection period.
- Code Cache Memory Monitor – trends Used Code Cache Memory.
- RAM Disk Monitor – trends Used RAM Disk Space.
- Socket State Monitor – is not implemented

Platform Monitor (systemMonitor-PlatformMonitor)

This component monitors current utilization of memory and CPU in your platform.

Figure 38 Platform Monitor properties



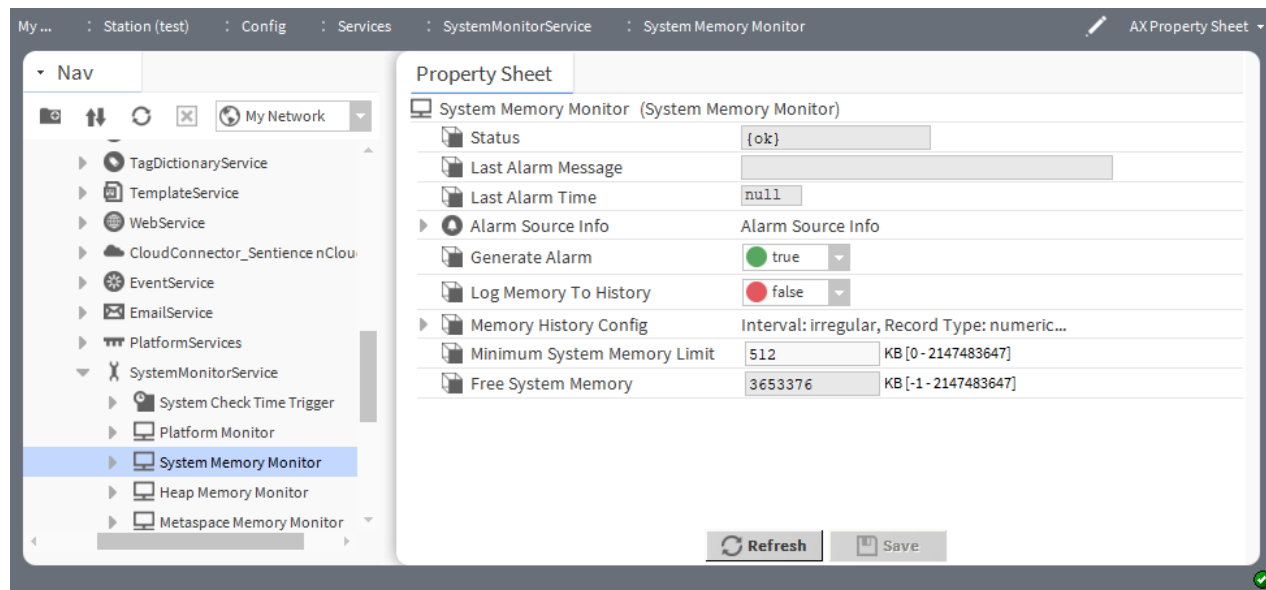
To access, click **Config**→**Services**→**SystemMonitorService** and double-click **Platform Monitor**.

Property	Value	Description
Status	text	Reports the current condition of the entity as of the last refresh: {alarm}, {disabled}, {down}, {fault}, {ok}, {stale}, {unackedAlarm}
Last Alarm Message	text	Displays the message that was triggered by the last alarm.
Last Alarm Time	read only	Displays when the system generated the last alarm assigned to this Alarm Class .
Alarm Source Info	additional properties	Contains a set of properties for configuring and routing alarms when this component is the alarm source. For property descriptions, refer to the <i>Niagara Alarms Guide</i>
Generate Alarm	true (default) or false	Controls if the software generates an alarm when a platform memory limit is crossed. true generates the alarm when the limit is crossed. false does not generate the alarm.
Platform Port	port number	Display the platform port number.

System Memory Monitor (systemMonitor-SystemMemoryMonitor)

This component monitors current utilization of memory in your system to detect memory leaks.

Figure 39 System Memory Monitor properties



To access, expand **Config**→**Services**→**SystemMonitorService** and double-click **System memory Monitor**.

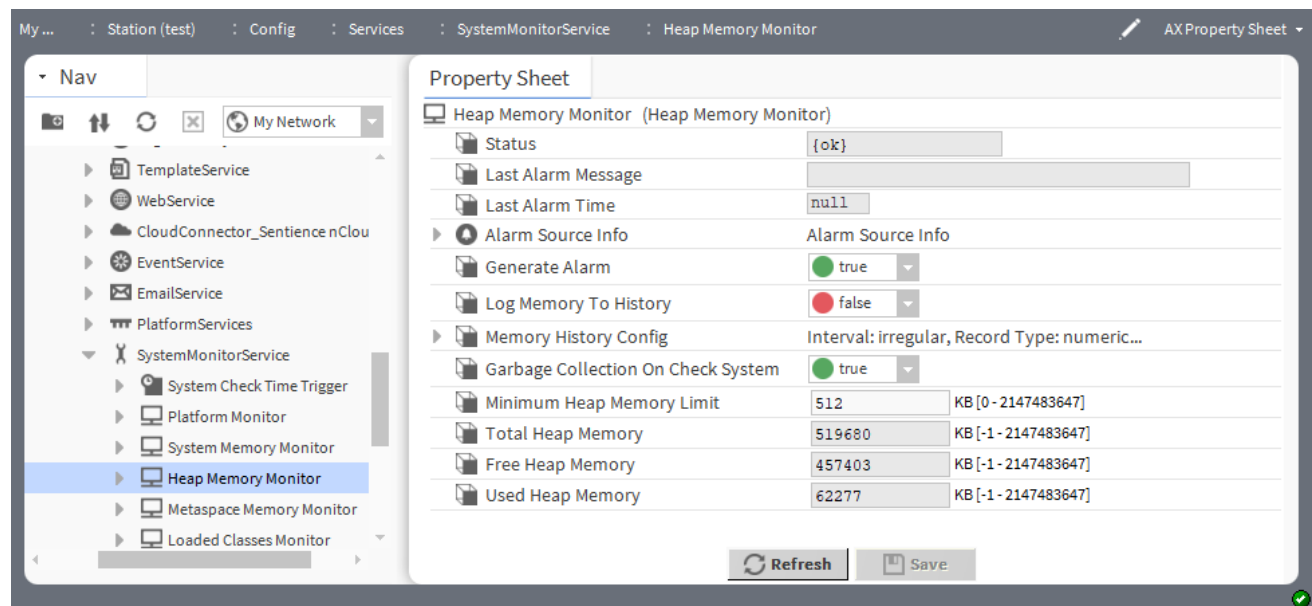
Property	Value	Description
Status	text	Reports the current condition of the entity as of the last refresh: {alarm}, {disabled}, {down}, {fault}, {ok}, {stale}, {unackedAlarm}
Last Alarm Message	read-only	Displays the message that was triggered by the last alarm.
Last Alarm time	text	Displays when the system generated the last alarm assigned to this Alarm Class .
Alarm Source Info	additional properties	Contains a set of properties for configuring and routing alarms when this component is the alarm source. For property descriptions, refer to the <i>Niagara Alarms Guide</i>
Generate Alarm	true (default) or false	Controls if the software generates an alarm when a platform memory limit is crossed. true generates the alarm when the limit is crossed. false does not generate the alarm.
Log Memory to History	true or false (default)	Controls when a trend for free system memory is created. true creates a trend each time the System Check Time Trigger fires. false ignores trend creation when the System Check Time Trigger fires.
Memory History Config	tab with additional properties	Provides a quick method of viewing the trends for Used Heap Memory (taken from the Heap Memory Monitor) and the Free System Memory (taken from the System Memory Monitor) when the Log Memory to History property is set to true.

Property	Value	Description
Minimum System output Memory	number	
Free System Memory	number	Displays the free memory available in the system.

Heap Memory Monitor (systemMonitor-HeapMemoryMonitor)

This component monitors the Heap Space allocation of memory and store references for new Java objects.

Figure 40 Heap Memory Monitor properties



To access, expand **Config→Services→SystemMonitorService** and double-click **Heap Memory Monitor**

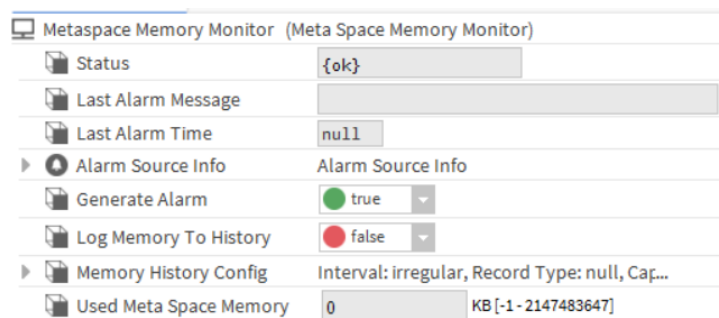
Property	Value	Description
Status	text	Reports the current condition of the entity as of the last refresh: {alarm}, {disabled}, {down}, {fault}, {ok}, {stale}, {unackedAlarm}
Last Alarm Message	text	Displays the message that was triggered by the last alarm.
Last Alarm time	text	Displays when the system generated the last alarm assigned to this Alarm Class .
Alarm Source Info	additional properties	Contains a set of properties for configuring and routing alarms when this component is the alarm source. For property descriptions, refer to the <i>Niagara Alarms Guide</i>
Generate Alarm	true (default) or false	Controls if the software generates an alarm when a platform memory limit is crossed. true generates the alarm when the limit is crossed. false does not generate the alarm.

Property	Value	Description
Log Memory to History	true or false (default)	Controls when a trend for free system memory is created. true creates a trend each time the System Check Time Trigger fires. false ignores trend creation when the System Check Time Trigger fires.
Memory History Config	tab with additional properties	Provides a quick method of viewing the trends for Used Heap Memory (taken from the Heap Memory Monitor) and the Free System Memory (taken from the System Memory Monitor) when the Log Memory to History property is set to true.
Garbage Collection on Check System	true (default) or false	If set to true release unused objects and making more memory available on the heap.
Minimum Heap Memory Limit	number	Indicates the minimum heap memory limit.
Total Heap Memory	number	Displays the total heap memory.
Free Heap Memory	number	Displays the free heap memory for usage.
Used Heap Memory	number	Displays the used heap memory.

Meta Space Memory Monitor (systemMonitor-MetaSpaceMemoryMonitor)

This component monitors the Meta Space memory utilization for stores of class and method data, static variable data, and other internal JVM metadata.

Figure 41 Meta Space Memory Monitor properties



To access, expand **Config→Services→SystemMonitorService** and double-click **Metaspaces Memory Monitor**

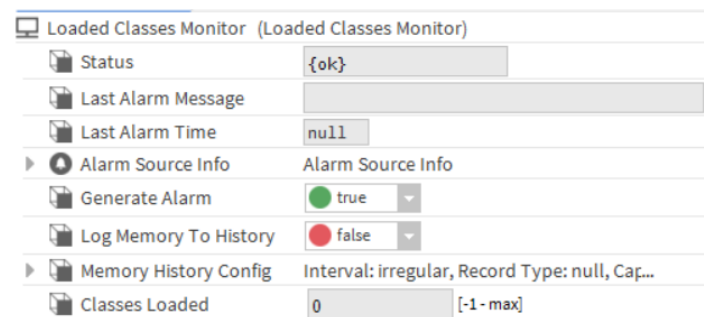
Property	Value	Description
Status	text	Reports the current condition of the entity as of the last refresh: {alarm}, {disabled}, {down}, {fault}, {ok}, {stale}, {unackedAlarm}
Last Alarm Message	text	Displays the message that was triggered by the last alarm.

Property	Value	Description
Last Alarm Time	text	Displays when the system generated the last alarm assigned to this Alarm Class .
Alarm Source Info	additional properties	Contains a set of properties for configuring and routing alarms when this component is the alarm source. For property descriptions, refer to the <i>Niagara Alarms Guide</i>
Generate Alarm	true (default) or false	Controls if the software generates an alarm when a platform memory limit is crossed. true generates the alarm when the limit is crossed. false does not generate the alarm.
Log Memory to History	true or false (default)	Controls when a trend for free system memory is created. true creates a trend each time the System Check Time Trigger fires. false ignores trend creation when the System Check Time Trigger fires.
Memory History Config	tab with additional properties	Provides a quick method of viewing the trends for Used Heap Memory (taken from the Heap Memory Monitor) and the Free System Memory (taken from the System Memory Monitor) when the Log Memory to History property is set to true.
Used Meta Space Memory	number	Displays the used Meta Space memory used by the system.

Loaded Classes Monitor (systemMonitor-LoadedClassesMonitor)

This component monitors Java Classes currently loaded on the system.

Figure 42 Loaded Classes Monitor properties



To access, expand **Config→Services→SystemMonitorService** and double-click **Loaded Classes Monitor**

Property	Value	Description
Status	text	Reports the current condition of the entity as of the last refresh: {alarm}, {disabled}, {down}, {fault}, {ok}, {stale}, {unackedAlarm}
Last Alarm Message	text	Displays the message that was triggered by the last alarm.

Property	Value	Description
Last Alarm time	text	Displays when the system generated the last alarm assigned to this Alarm Class .
Alarm Source Info	additional properties	Contains a set of properties for configuring and routing alarms when this component is the alarm source. For property descriptions, refer to the <i>Niagara Alarms Guide</i>
Generate Alarm	true (default) or false	Controls if the software generates an alarm when a platform memory limit is crossed. true generates the alarm when the limit is crossed. false does not generate the alarm.
Log Memory to History	true or false (default)	Controls when a trend for free system memory is created. true creates a trend each time the System Check Time Trigger fires. false ignores trend creation when the System Check Time Trigger fires.
Memory History Config	tab with additional properties	Provides a quick method of viewing the trends for Used Heap Memory (taken from the Heap Memory Monitor) and the Free System Memory (taken from the System Memory Monitor) when the Log Memory to History property is set to true.
Classes Loaded	number	Displays the Java classes currently loaded on the system.

CPU Idle Cycles Monitor (systemMonitor-IdleCPUMonitor)

This component monitors idle CPU Cycles over the last collection period.

Figure 43 Idle C P U Monitor properties

The screenshot displays the Niagara Platform configuration interface. On the left, a navigation pane shows a tree structure of components, with 'CPU Idle Cycles Monitor' selected under 'SystemMonitorService'. The main area shows the 'Property Sheet' for this component. The properties are organized into sections: Status, Alarm Source Info, and Cpu History Config. The 'Generate Alarm' property is set to 'true' (indicated by a green dot), and 'Log C P U To History' is set to 'false' (indicated by a red dot). The 'Cpu History Config' section shows various metrics including 'Last Interval' (749157 ms), 'Last Interval Percent' (13.88%), 'Last Minute' (385750 ms), 'Last Minute Percent' (80.36%), 'Last Hour' (2148750 ms), 'Last Hour Percent' (7.46%), and 'Last Day' (-1 ms).

Property	Value
Status	{ok}
Last Alarm Message	
Last Alarm Time	null
Alarm Source Info	Alarm Source Info
Generate Alarm	true
Log C P U To History	false
Cpu History Config	Interval: irregular, Record Type: numeric...
Target C P U	-1
Last Interval	749157 ms
Last Interval Percent	13.88 % [0.00 - 100.00]
Since Start Percent	24.03 % [0.00 - 100.00]
Last Minute	385750 ms
Last Minute Percent	80.36 % [0.00 - 100.00]
Last Hour	2148750 ms
Last Hour Percent	7.46 % [0.00 - 100.00]
Last Day	-1 ms
Last Day Percent	0.00 % [0.00 - 100.00]

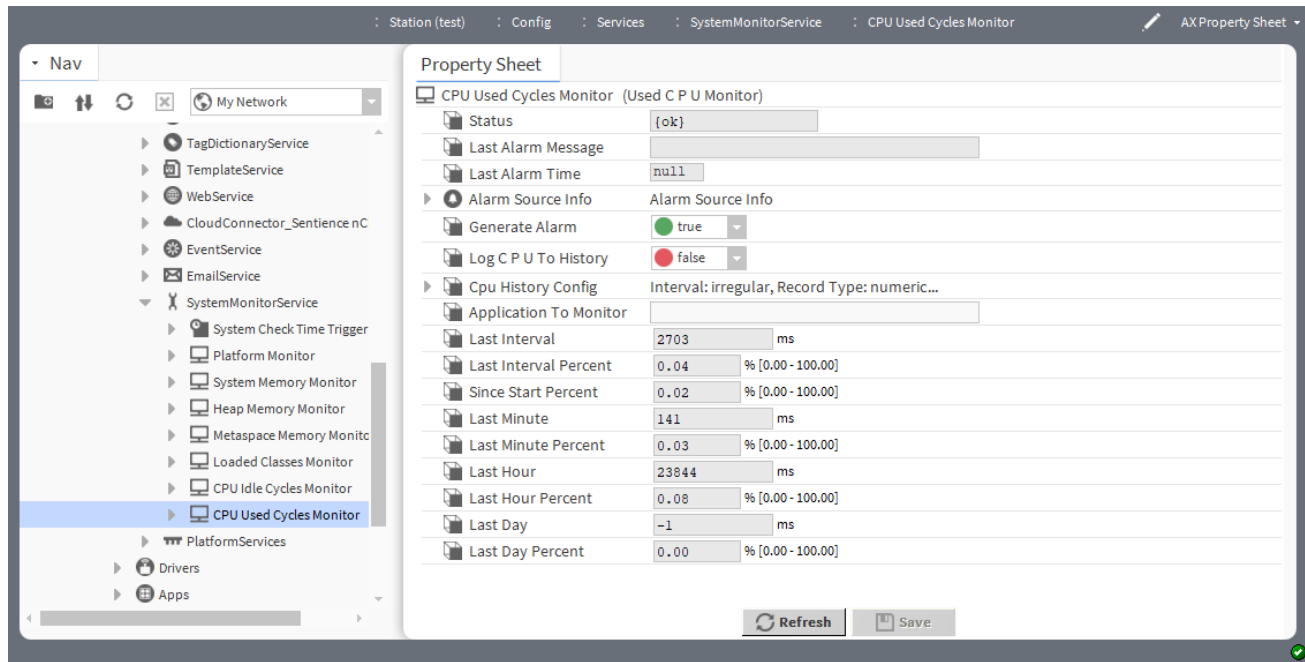
To access, expand **Config**→**Services**→**SystemMonitorService** and double-click **CPU Idle Cycles Monitor**.

Property	Value	Description
Status	text	Reports the current condition of the entity as of the last refresh: {alarm}, {disabled}, {down}, {fault}, {ok}, {stale}, {unackedAlarm}
Last Alarm Message	text	Displays the message that was triggered by the last alarm.
Last Alarm time	text	Displays when the system generated the last alarm assigned to this Alarm Class .
Alarm Source Info	additional properties	Contains a set of properties for configuring and routing alarms when this component is the alarm source. For property descriptions, refer to the <i>Niagara Alarms Guide</i>
Generate Alarm	true (default) or false	Controls if the software generates an alarm when a platform memory limit is crossed. true generates the alarm when the limit is crossed. false does not generate the alarm.
Log C P U To History	true or false (default)	Updates the history log with the number of idle CPU cycles over the last collection period.
Cpu History Config	additional properties	
Target C P U	time in milliseconds	
Last Interval	read-only	Reports idle cycles during the last interval.
Since Start Percent	read-only	Reports, as a percentage, the amount of idle cycles since the platform started.
Last Minute	read-only	Reports idle cycles during the last minute.
Last Minute Percent	read-only	Reports, as a percentage, the amount of idle cycles during the last minute.
Last Hour	read-only	Reports idle cycles during the last hour.
Last Hour Percent	read-only	Reports, as a percentage, the amount of idle cycles during the last hour.
Last Day	read-only	Reports idle cycles during the last day.
Last Day Percent	read-only	Reports, as a percentage, the amount of idle cycles during the last day.

CPU Used Cycles Monitor (systemMonitor-UsedCPUMonitor)

This component monitors used CPU Cycles over the last collection period.

Figure 44 Used C P U Monitor properties



To access, expand **Config**→**Services**→**SystemMonitorService** and double-click **CPU Used Cycles Monitor**

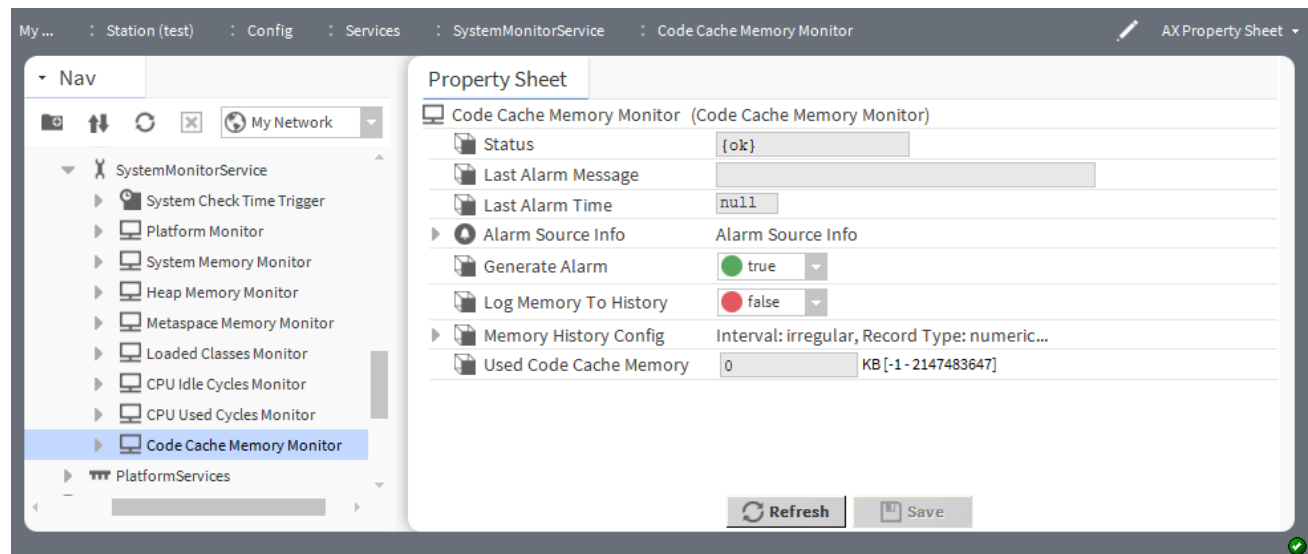
Property	Value	Description
Status	text	Reports the current condition of the entity as of the last refresh: {alarm}, {disabled}, {down}, {fault}, {ok}, {stale}, {unackedAlarm}
Last Alarm Message	text	Displays the message that was triggered by the last alarm.
Last Alarm time	text	Displays when the system generated the last alarm assigned to this Alarm Class .
Alarm Source Info	additional properties	Contains a set of properties for configuring and routing alarms when this component is the alarm source. For property descriptions, refer to the <i>Niagara Alarms Guide</i>
Generate Alarm	true (default) or false	Controls if the software generates an alarm when a platform memory limit is crossed. true generates the alarm when the limit is crossed. false does not generate the alarm.
Log C P U To History	true or false (default)	Updates the history log with the number of used CPU cycles over the last collection period.
Cpu History Config	additional properties	
Application To Monitor	time in milliseconds	Defines the application to monitor.
Last Interval	read-only	Reports used CPU cycles during the last interval.

Property	Value	Description
Since Start Percent	read-only	Reports, as a percentage, the amount of used CPU cycles since the platform started.
Last Minute	read-only	Reports used CPU cycles during the last minute.
Last Minute Percent	read-only	Reports, as a percentage, the amount of used CPU cycles during the last minute.
Last Hour	read-only	Reports used CPU during the last hour.
Last Hour percent	read-only	Reports, as a percentage, the amount of used CPU cycles during the last hour.
Last Day	read-only	Reports the used CPU during the last day.
Last Day Percent	read-only	Reports, as a percentage, the amount of used CPU cycles during the last day.

Code Cache Memory Monitor (systemMonitor-CodeCacheMemoryMonitor)

This component monitors the current stores of native code produced by the Java VM Just In Time (JIT) Compiler.

Figure 45 Code Cache Memory Monitor properties



To access, expand **Config**→**Services**→**SystemMonitorService** and double-click **Code Cache Memory Monitor**

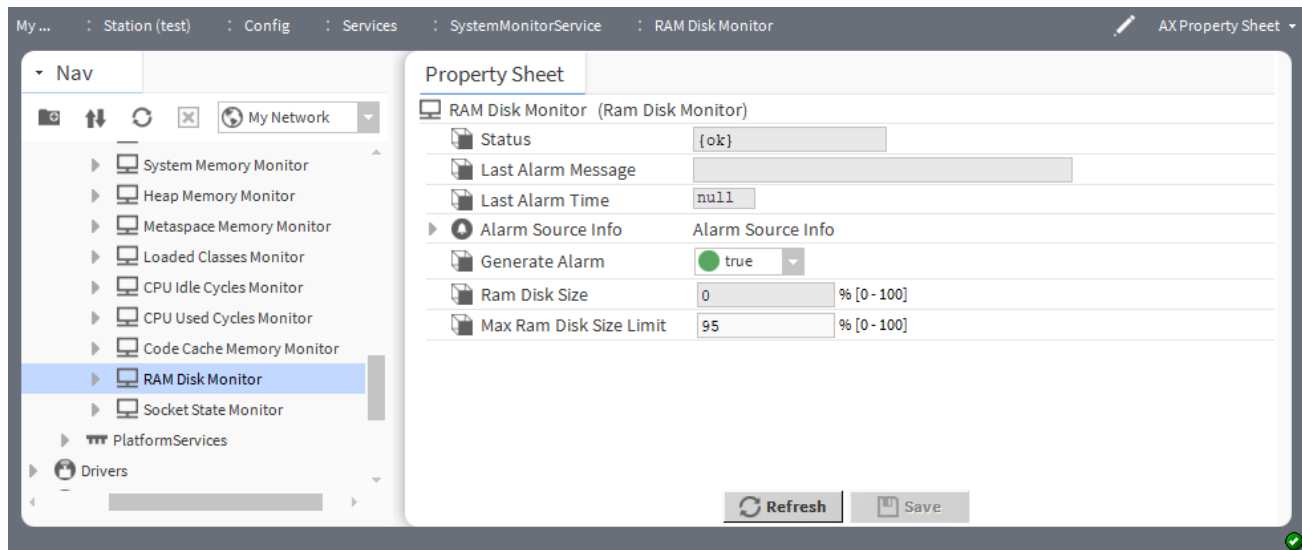
Property	Value	Description
Status	text	Reports the current condition of the entity as of the last refresh: {alarm}, {disabled}, {down}, {fault}, {ok}, {stale}, {unackedAlarm}
Last Alarm Message	text	Displays the message that was triggered by the last alarm.
Last Alarm time	text	Displays when the system generated the last alarm assigned to this Alarm Class .

Property	Value	Description
Alarm Source Info	additional properties	Contains a set of properties for configuring and routing alarms when this component is the alarm source. For property descriptions, refer to the <i>Niagara Alarms Guide</i>
Generate Alarm	true (default) or false	Controls if the software generates an alarm when a platform memory limit is crossed. true generates the alarm when the limit is crossed. false does not generate the alarm.
Log Memory to History	true or false (default)	Controls when a trend for free system memory is created. true creates a trend each time the System Check Time Trigger fires. false ignores trend creation when the System Check Time Trigger fires.
Memory History Config	tab with additional properties	Provides a quick method of viewing the trends for Used Heap Memory (taken from the Heap Memory Monitor) and the Free System Memory (taken from the System Memory Monitor) when the Log Memory to History property is set to true.
Used Code Cache Memory	number	Displays the memory utilized by code cache.

RAM Disk Monitor (systemMonitor-RamDiskMonitor)

This component monitors the current RAM disk freespace on Niagara platforms.

Figure 46 RamDiskMonitor properties



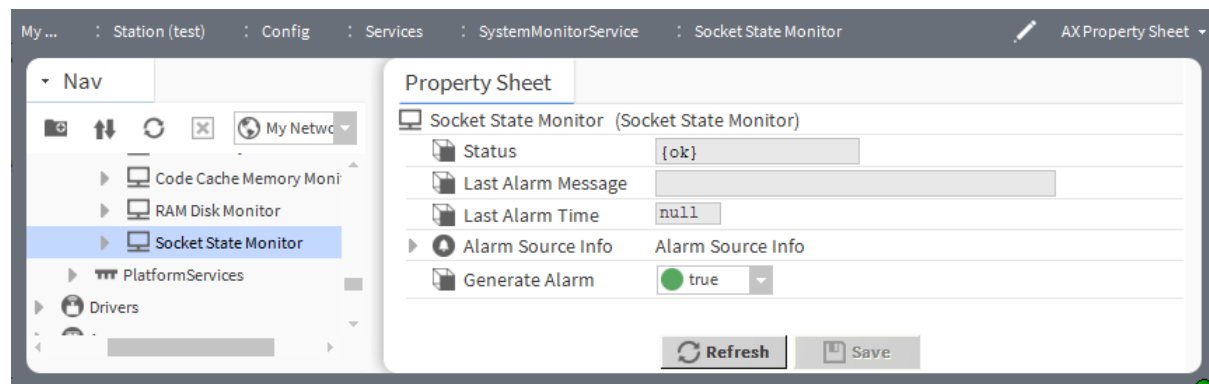
To access, expand **Config**→**Services**→**SystemMonitorService** and double-click **RAM Disk Monitor**

Property	Value	Description
Status	text	Reports the current condition of the entity as of the last refresh: {alarm}, {disabled}, {down}, {fault}, {ok}, {stale}, {unackedAlarm}
Last Alarm Message	text	Displays the message that was triggered by the last alarm.
Last Alarm Time	text	Displays when the system generated the last alarm assigned to this Alarm Class .
Alarm Source Info	additional properties	Contains a set of properties for configuring and routing alarms when this component is the alarm source. For property descriptions, refer to the <i>Niagara Alarms Guide</i>
Generate Alarm	true (default) or false	Controls if the software generates an alarm when a platform memory limit is crossed. true generates the alarm when the limit is crossed. false does not generate the alarm.
Ram Disk Size	number	Displays the disk space in percentage which is utilized.
Max Ram Disk Size Limit	number	Displays the maximum disk space in percentage that you can utilize.

Socket State Monitor (systemMonitor-SocketStateMonitor)

This component monitors the current state of any socket in the `spy:/platform diagnostics/netstat -A` on Niagara platforms.

Figure 47 Socket State Monitor properties



To access, expand **Config**→**Services**→**SystemMonitorService** and double-click **Socket State Monitor**

Property	Value	Description
Status	text	Reports the current condition of the entity as of the last refresh: {alarm}, {disabled}, {down}, {fault}, {ok}, {stale}, {unackedAlarm}
Last Alarm Message	text	Displays the message that was triggered by the last alarm.

Property	Value	Description
Last Alarm time	text	Displays when the system generated the last alarm assigned to this Alarm Class .
Alarm Source Info	additional properties	Contains a set of properties for configuring and routing alarms when this component is the alarm source. For property descriptions, refer to the <i>Niagara Alarms Guide</i>
Generate Alarm	true (default) or false	Controls if the software generates an alarm when a platform memory limit is crossed. true generates the alarm when the limit is crossed. false does not generate the alarm.

Chapter 17 Plugin guides

Topics covered in this chapter

- ◆ Application Director view (platDaemon-ApplicationDirector)
- ◆ HTML-5 Certificate Management View
- ◆ License Platform Service Plugin (platform-LicensePlatformService)
- ◆ Logger Configuration view (workbench-LoggerConfiguration)
- ◆ Data Recovery Service Editor view (platDataRecovery-DataRecoveryServiceEditor)
- ◆ Distribution File Installer (platDaemon-DistInstaller)
- ◆ File Transfer Client (platDaemon-FileTransferClient)
- ◆ Hardware Scan Service view (platHwScan-HardwareScanServiceView)
- ◆ Javelina Battery Platform Service Plugin (platPower-JavelinaBatteryPlatformServicePlugin)
- ◆ Lexicon Installer view (platDaemon-LexiconInstaller)
- ◆ License Manager view (platDaemon-LicenseManager)
- ◆ License Platform Service Plugin view (platform-LicensePlatformServicePlugin)
- ◆ Network License Summary view (provisioningNiagara-NetworkLicenseSummary)
- ◆ Ntp Platform Service Editor (platform-NtpPlatformServiceEditorNpsdk)
- ◆ Ntp Platform Service Editor Qnx view (platform-NtpPlatformServiceEditorQnx)
- ◆ Platform Administration view (platDaemon-PlatformAdministration)
- ◆ Platform Certificate Management (platCrypto-CertManagerService)
- ◆ Platform Service Container Plugin (platform-PlatformServiceContainer)
- ◆ Power Monitor Platform Service Plugin (platPower-PowerMonitorPlatformServicePlugin)
- ◆ Software Manager view (platDaemon-SoftwareManager)
- ◆ Station Copier view (platDaemon-StationCopier)
- ◆ Syslog Platform Service Plugin view (platform-SyslogPlatformServicePlugin)
- ◆ System Date Time Editor view (platform-SystemDateTimeEditor)
- ◆ System Monitor Config (systemMonitor-SystemMonitorConfig)
- ◆ System Platform Service Plugin (platform-SystemPlatformServicePlugin)
- ◆ System Platform Service Qnx Plugin (platform-SystemPlatformServiceQnxPlugin)
- ◆ Tcp Ip Platform Service Plugin (platform-TcplpPlatformService)
- ◆ Workbench Certificate Management (platCrypto-CertManagerTool)
- ◆ Workbench License Manager view (platform-WorkbenchLicenseManager)

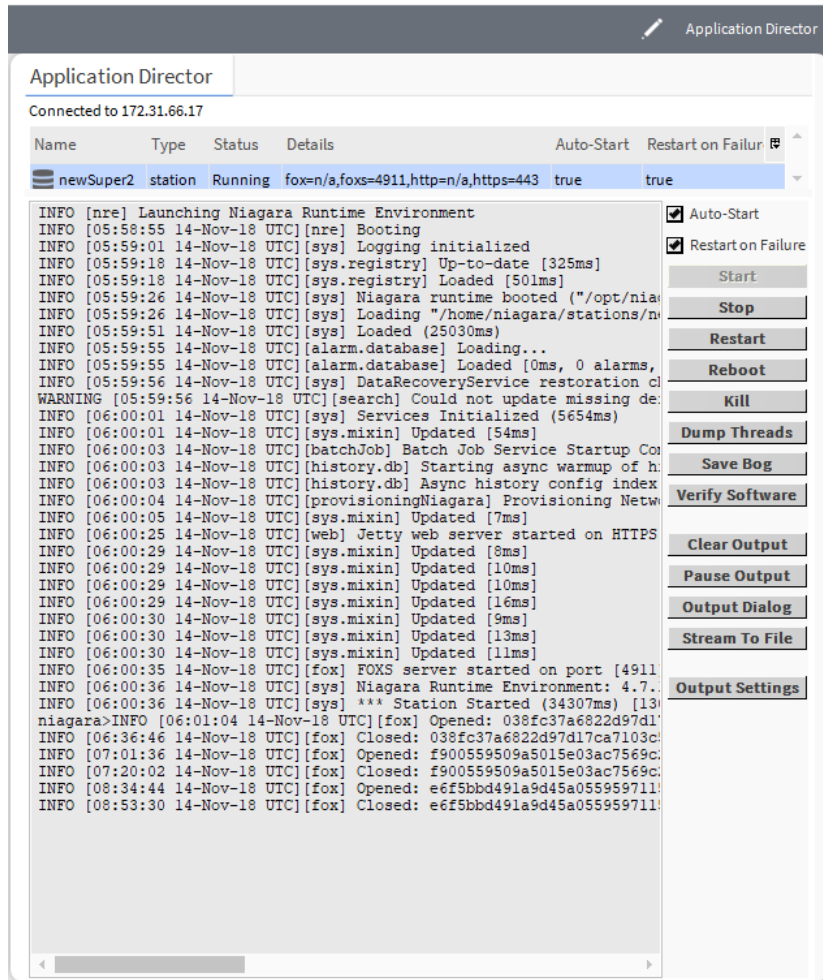
There are many ways to view plugins (views). One way is directly in the tree. In addition, you can right-click on an item and select one of its views. Plugins provide views of components.

In Workbench, access the following summary descriptions on any plugin by selecting **Help** → **On View (F1)** from the menu, or pressing **F1** while the view is open.

Application Director view (platDaemon-ApplicationDirector)

The **Application Director** view interfaces to each station whether it is running or not.

Figure 48 Application Director view for a controller

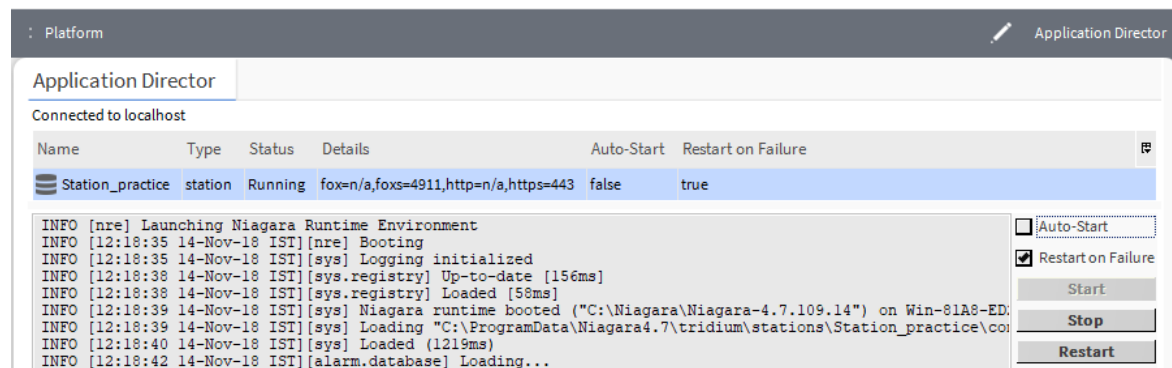


The **Application Director** is split into three main areas:

- Installed applications— at top
- Application output— main area
Related are log levels defined for the station.
- Application and output controls— right-side check boxes and buttons

In the **Application Director** for any controller, the installed applications area should show (at most) only one station, as shown above. However, the **Application Director** for a Windows platform (Supervisor, or engineering workstation) may show more than one station, as shown below.

Figure 49 Application Director for a Supervisor host



Even if a Windows platform is licensed for more than one station, running multiple stations at the same time requires you to use non-default ports for all but one of them to avoid port binding issues. For example, use a Fox and Foxs port other than 1911 or 3011 respectively, or Http and Https port other than 80 or 443 respectively.

Columns

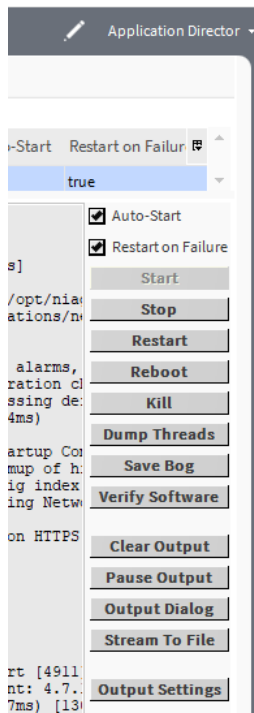
The top area of the **Application Director** shows a table of installed applications (stations).

Column	Description
Name	Displays the name of the station directory.
Type	Identifies the station as belonging to the Niagara family.
Status	Indicates the condition of the station at last polling.
Details	<code>fox=</code> TCP/IP port monitored for regular (unencrypted) Fox connections to Workbench and other stations. Shows <code>n/a</code> if station is not running, or if Fox Enabled is set to <code>false</code>. <code>foxs=</code> TCP/IP port monitored for secure Fox connections to Workbench and other stations, if so configured. Shows <code>n/a</code> if the host does not support (or is enabled) for a secure connection, or if the station is not running, or if Foxs Enabled is set to <code>false</code>. <code>http=</code> HTTP port that the station's WebService monitors for regular (unencrypted) browser connections to the station. Shows <code>"n/a"</code> if station is not running, or if it does not have a running WebService, or if Http Enabled is set to <code>false</code>. <code>https=</code> HTTP port that the station's WebService monitors for secure browser connections to the station, if so configured. Shows <code>"n/a"</code> if host does not support (or is enabled) for a secure connection, or if the station is not running, or if Https Enabled is set to <code>false</code>, or if the station does not have a running WebService.
Auto-Start	Displays the currently configured Auto-Start property for the station.
Restart on Failure	Displays the currently configured Restart on Failure property for the station.

Buttons

Unlike in most Workbench views, where changes are entered first and then applied with a **Save** button, in the **Application Director** when you click check boxes and buttons, changes are applied immediately to the selected station.

Figure 50 Application Director start-up options and buttons



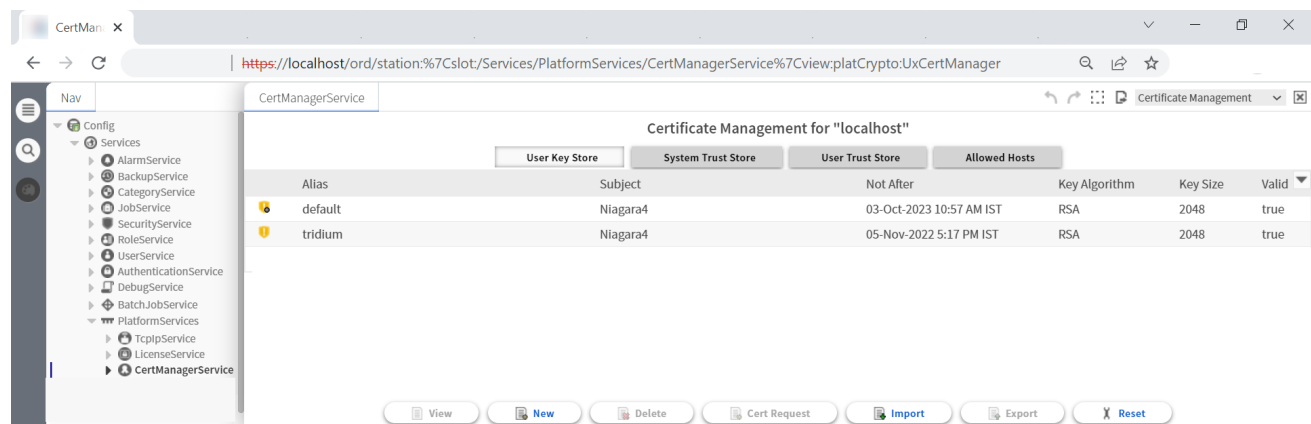
- **Auto-Start** Specifies whether the station starts following platform daemon startup. A station restart occurs:
 - Following a host reboot, such as after a power cycle
 - As the result of a **Reboot** command
 - Following the installation of any dist file(s)
 - Following any TCP/IP-related changes
 - When changing any existing module (upgrading or downgrading)
- **Restart on Failure** specifies whether the platform daemon restarts the station if its process exits with a non-zero return code (for example, the engine watchdog had killed the station because of a deadlock condition). In Niagara 4, controllers can have a station restart without a reboot. Therefore, if this option is enabled, and the station fails (terminates with error), the station is restarted. If a controller has three automatic restarts within 10 minutes (or however many specified in the station's PlatformService **Failure Reboot Limit** property, the station remains in a failed state, regardless of the setting above.
- **Start** when pressed, the host's platform daemon immediately starts the station, clears the text in the station output, and displays messages for the new station.
- **Stop** when pressed, opens a confirmation window. If you confirm, the host's platform daemon shuts the station down (saving configuration to its config.bog file, and potentially saving history data)
- **Restart** when pressed, opens a confirmation window. If you confirm, the host's platform daemon shuts the station down gracefully, then restarts it.
- **Reboot** when pressed, opens a confirmation window. If you confirm, reboots the selected host. This is considered a drastic action.
- **Kill** when pressed, opens a confirmation window. If you confirm, the host's platform daemon terminates the station process immediately.
- **Dump Threads** when pressed, the host's platform daemon has the station send a VM thread dump to its station output.

- **Save Bog** when pressed, the host's platform daemon has the station locally save its configuration to `config.bog`.
- **Verify Software** when pressed, Workbench parses the station's `config.bog` and the host's `platform.bog` files, looking for module references. It then checks to see if those modules (and any other software upon which they depend) are installed.
- **Clear Output** when pressed clears the output
- **Pause Output** when pressed, the output is freeze from updating further (no longer in real time). When you freeze the output, the button changes to **Load Output**.
- **Output Dialog** when pressed, it produces a separate non-modal output window displaying the same output text as in the **Application Director's** standard output area. Included are buttons to **Dump Threads**, **Pause Output**, **Clear Output**, and **Close** the window.
- **Stream To File** opens a window for assigning a file name. Once open, the system saves all application output to this file.
- **Output Settings** opens a window for specifying how the platform daemon buffers the output from the station.

HTML-5 Certificate Management View

In Niagara 4.13 and later, there is added support for **Certificate Management**, which is a browser implementation. Using this view, you can create digital certificates and certificate signing requests (CSRs) and import and export keys. The view always has a **default** certificate. This **default** certificate does not have a user-defined password, and cannot be deleted, signed, imported, or exported.

Figure 51 Certificate Management view



To access this view in web browser, expand **Config**→**Services**→**PlatformServices** and double-click **CertManagerService** or right-click **CertManagerService** and click **Views**→**Certificate Management**.

The browser view offers functional equivalents to the Workbench view when creating certificates. There are only a few additional functions.

EC Key Algorithm

The Elliptic Curve is a different sort of cryptographic formula used to produce the certificate keys, similar to the current RSA, which is the only Key Algorithm in the bajauri version. You select a `KeySpec` rather than a `KeySize`. The EC keys create digital signatures, generate pseudorandom numbers, and encrypt data.

Extensions

The **Subject Alternative Name** extension allows identities to be bound to the subject of the certificate.

The alternative extensions are:

- **Email:** The contact address for the certificate.
- **DNS Name:** Common name of a server.
- **Directory Name:** Directory Name must be an RDNSequence, similar to a Distinguished Name such as organization name and state etc.
- **Uniform Resource Identifier (URI):** A Universal Resource Identifier is the location of an Internet resource (for example, web-page, ftp service, and so on).
- **IP Address:** Defines the IP address of the target server.
- **Register ID:** This must be an OID with numbers separated by decimals like '1.2.3'.

Add Extensions

- **Extended Key Usage:** This extension indicates one or more purposes for which the certificate may be used, in addition to or in place of the basic purposes indicated in the key usage extension. In general, this extension will appear only in end entity certificates.
- **Basic Constraints:** The basic constraints extension identifies whether the subject of the certificate is a CA and how many certificates can follow this one in certification paths.
- **CRL distribution points:** The CRL distribution points extension identifies how CRL information is obtained. It consists of a sequence of Distribution Points, each of which consists of three optional fields: distribution point, CRL issuer, and reasons. Although all fields are optional, there must be at least a distribution point or a CRL issuer.

The view provides different type of certificate store tabs.

User key Store

The **User Key Stores** contain server certificates and self-signed certificates with their matching keys. Each certificate has a pair of unique private and public encryption keys for each platform. A **User Key Stores** supports the server side of the relationship by sending one of its signed server certificates in response to a client (Workbench, platform or station) request to connect.

Trust Stores

The trust stores (system and user) contain signed and trusted root CA certificates with their public keys. These stores contain no private keys. A trust store supports the client side of the relationship by using its root CA certificates to verify the signatures of the certificates it receives from each server. If a client cannot validate a server certificate's signature, an error message allows you to approve or reject a security exemption (on the **Allowed Hosts** tab).

The **System Trust Stores** contain installed signed certificates by trusted entities (CA authorities) recognized by the Java Runtime Engine (JRE) of the currently opened platform. A **User Trust Store** contains installed signed certificates by trusted entities that you have imported (your own certificates).

Only certificates with public keys are stored in the trust stores. The majority of certificates in the **System Trust Store** come from the JRE. You add your own certificates to a **User Trust Store** by importing them.

Feel free to pass out such root certificates to your team; share them with your customers; make sure that any client that needs to connect to one of your servers has the server's root certificate in its client trust store.

Allowed Hosts Tab

This tab lists self-signed certificates that have been manually approved for use to authenticate a server. As such, they have not been signed by a CA. They should not be approved unless you are certain that the communication they facilitate will be secure.

Columns

This table lists all columns in the stores.

Columns	Description
Alias	Identifies certificates by location or function.
Issued By	Identifies the entity that created the certificate.
Subject	Identifies the company that owns the certificate.
Not Before	Displays the date before which the certificate is not valid.
Not After	Displays the expiration date for the certificate.
Key Algorithm	Refers the cryptographic formula used to calculate the certificate keys. For the RSA select the key size in bits and for the EC selects the key specification.
Key Size	For RSA keys, the size of the keys in bits. Four key sizes are allowed: 1024 bits, 2048 bits (this is the default), 3072 bits, and 4096 bits. The bigger the key, the longer it takes to generate.
Key Spec	For EC keys the size of the key in bits. Seven key sizes are Brainpool P256 r1 bits, Brainpool P320 r1 bits, Brainpool P384 r1 bits, Brainpool P512 r1 bits, P-256 bits, P-521 bits, and P-384 bits.
Signature Algorithm	Names the mathematical formula used to sign the certificate.
Signature Size	Shows the size of the signature.
Valid	Displays the dates between which the certificate is valid.
Self Signed	Indicates that the certificate was signed with its own private key.
Host	Reports the server, usually an IP address.
Approval	Reports the servers within the network to which the a client may connect. If approval is no, the system does not allow the client to connect.
Created	Identifies the date the record was created.

Buttons

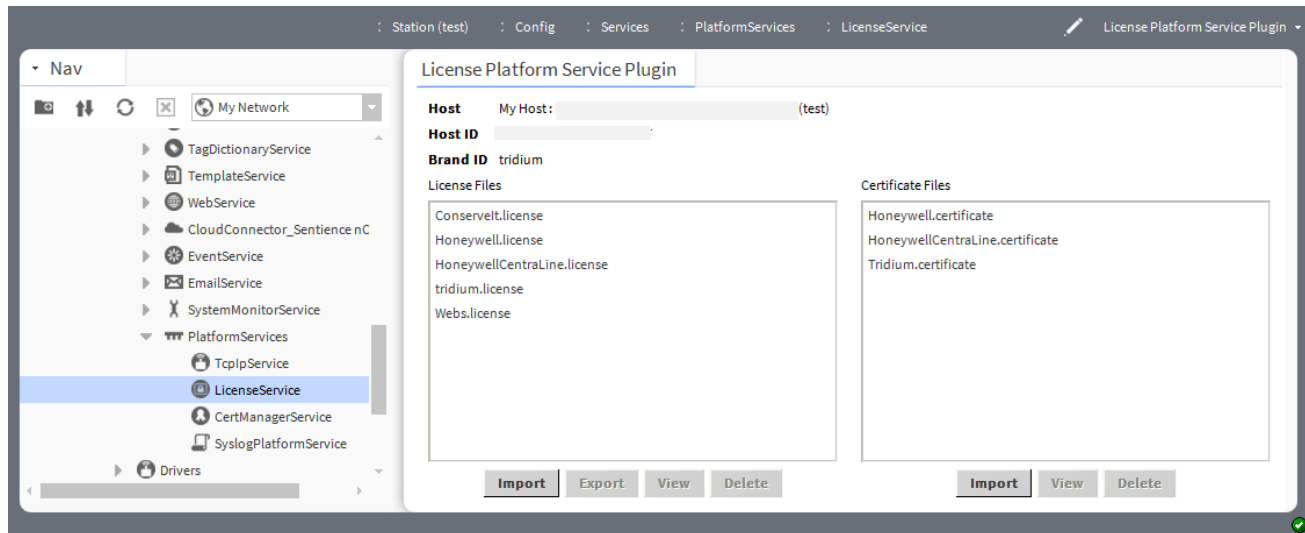
This list contains all the buttons available in the stores.

- **View** allows you to view the information of the selected certificate.
- **New** creates a new self-signed certificate.
- **Delete** deletes the selected certificate from the Keystore.
- **Cert Request** generates a certificate request and to exports it.
- **Import** adds a new certificate in the keystore.
- **Export** exports a selected certificate to a new directory.
- **Reset** resets the Keystore and generates a new self-signed certificate.
- **Approve** designates the server as an allowed host.
- **Unapprove** prohibits a connection to this server host. The system terminates any attempted communication.

License Platform Service Plugin (platform-LicensePlatformService)

This view provides station access to a PC platform's license(s) and certificate(s). This service is found under the running station's **PlatformServices** container. From the default plugin (view), you can perform the same operations as from the **License Manager** view using a platform connection.

Figure 52 License Platform Service Plugin



To access, expand **Config→Services→Platform Service** and double-click **LicenseService**.

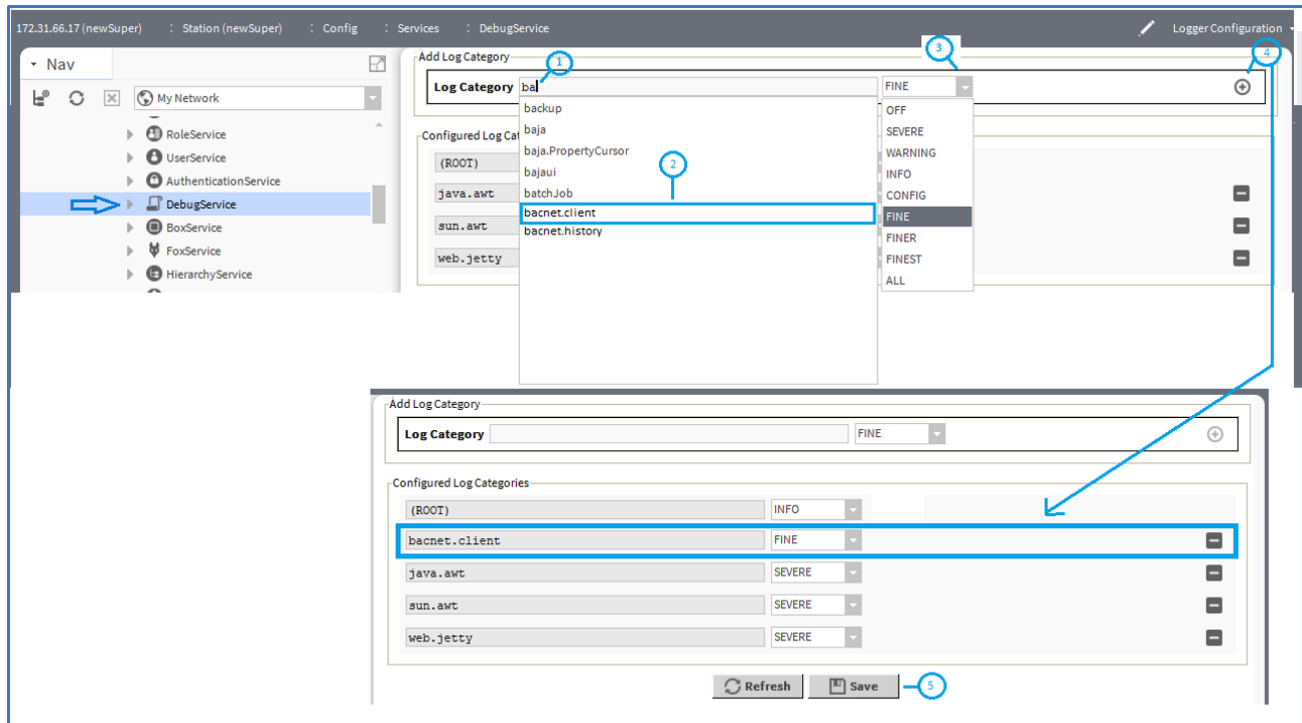
Buttons

- **Import** installs a new license or certificate file from a local license file or license archive (.lar) file. This button is always available. Typically, you import license files from either the online licensing server or from your local license database.
- **Export** saves all licenses (or any selected licenses) locally, as a license archive file. This button is always available.
- **View** opens the selected license file. Clicking this button is the same as double-clicking a license or certificate.
- **Delete** removes an existing license file.

Logger Configuration view (workbench-LoggerConfiguration)

This view configures the log level of station processes. This tunes the station output seen in the **Application Director**.

Figure 53 Logger Configuration view



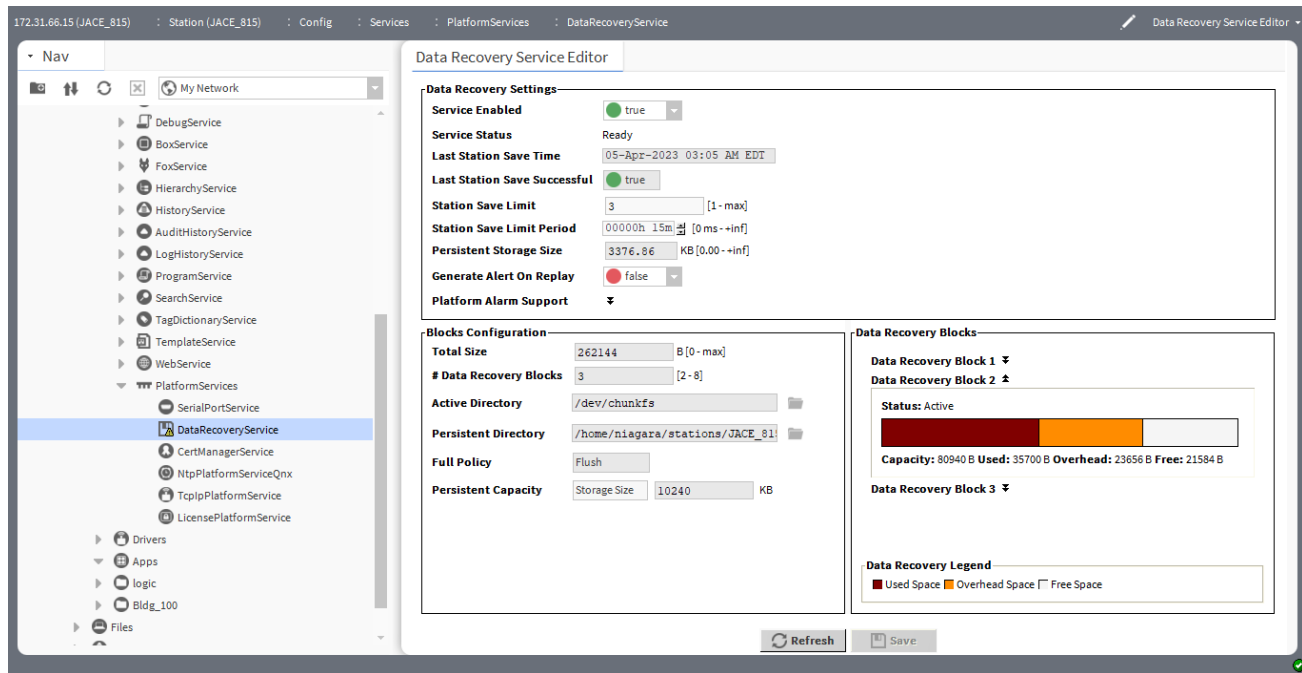
To access this view, expand **Station**→**Config**→**Services** and double-click **DebugService**.

In the example shown above, the **Log Category** (1) of the **bacnet.client** process (2) is being added with a log level of **FINE** (3). Clicking the add icon (4) adds the filter to the **Configured Log Categories** list and clicking **Save** (5) completes the task. Such an entry might be useful to troubleshoot errors about writes to BACnet proxy points.

Data Recovery Service Editor view (platDataRecovery-DataRecoveryServiceEditor)

The **Data Recovery Service Editor** is the default view on the **DataRecoveryService**, as found in the **PlatformServices** of JACE controllers with onboard static RAM (SRAM or FRAM), or an installed SRAM option card.

Figure 54 Data Recovery Service Editor view



To access this view expand the controllers **Config**→**Services**→**PlatformServices** and double-click **DataRecoveryService**.

This view allows monitoring of the battery-less support provided by this service. In a few cases, an SRAM-equipped JACE can additionally (and optionally) use a backup battery, such as an NiMH onboard battery pack, and (if applicable) and external 12V sealed lead-acid battery. In this case, both the **DataRecoveryService** and **PowerMonitorService** can exist in the station's **PlatformServices** container, operating independently or in unison, as configured.

For details, see the About the **DataRecoveryService** section in the document *Niagara Data Recovery Service Guide*.

Distribution File Installer (platDaemon-DistInstaller)

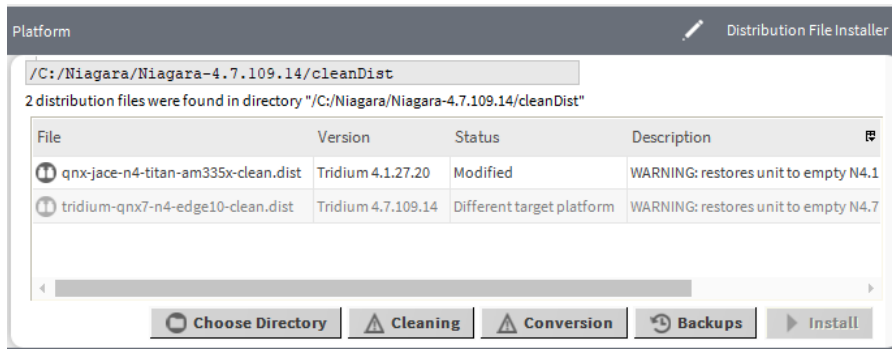
This view allows you to install distribution (dist) files from your Workbench PC to the remote host platform.

Typical use is for restoring backups, or for installing a clean distribution file to essentially erase the file system of a controller and start again with the near-factory defaults.

Dist file selection

By default, the first time you use the Installer, the system searches the backups folder under your Workbench **User Home** (~\backups). If that folder does not exist yet (no backups have been made), then it searches the cleanDist folder under your Niagara **Sys Home** (!\cleanDist) instead.

Figure 55 Available .dist files in Distribution File Installer

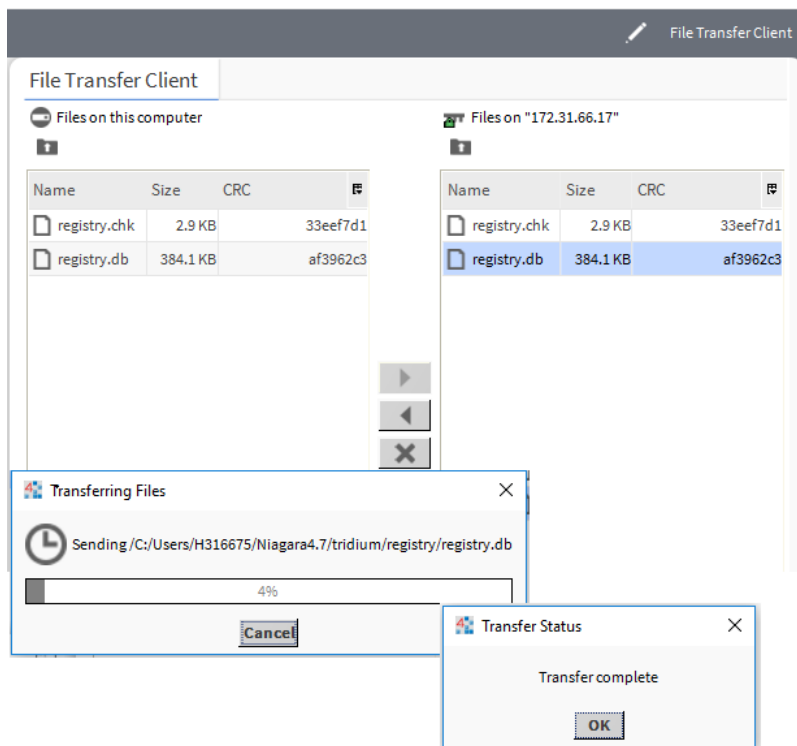


At the bottom of the view, the **Cleaning** and **Backups** buttons provide shortcuts to these two folder areas. If needed, you can also click the **Choose Directory** button to open a **Change Directory** window, and point the Installer to that location.

File Transfer Client (platDaemon-FileTransferClient)

The **File Transfer Client** is the platform view of a controller that allows you to copy files and/or folders between your Workbench PC and the remote platform.

Figure 56 File Transfer Client



To access this view in a controller expand **Platform**→**File Transfer Client**.

Hardware Scan Service view (platHwScan-HardwareScanServiceView)

The **Hardware Scan Service View** is the default view on the platform service **HardwareScanService** in a station, providing that the Edge 10 ACE Driver platform has the **platHwScan** module installed, along with

the appropriate `platHwScan Type` module. This view provides a graphical diagram of communication ports and other features on the hosting Edge 10 ACE Driver platform, including callouts to a table that explain the location, description (such as COM2), port type, and status.

Javelina Battery Platform Service Plugin (platPower-JavelinaBatteryPlatformServicePlugin)

The **Javelina Battery Platform Service Plugin** is the default view on the platform service **PowerMonitor-Service** in the JACE-7 (700) series controller. This view provides parameters for changing the shutdown delay time, as well as alarm source configuration settings.

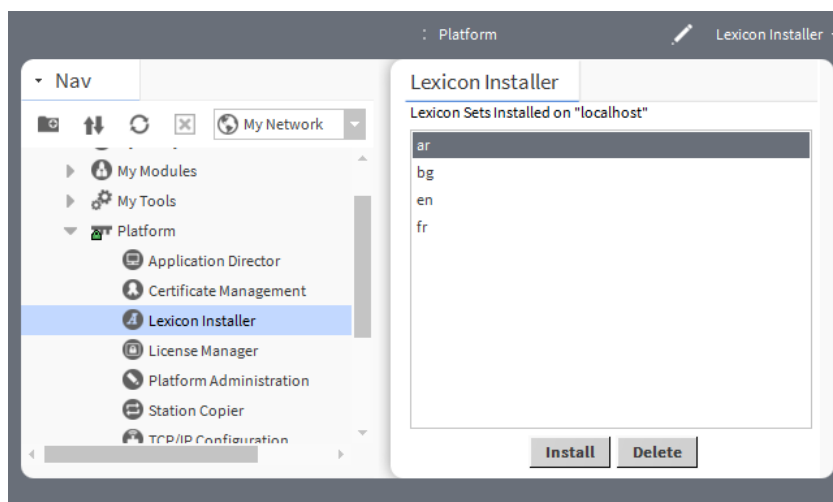
Typically, support is enabled and configured at commissioning time. For related details, see JACE power monitoring configuration in the latest *Niagara Data Recovery Service Guide*.

Lexicon Installer view (platDaemon-LexiconInstaller)

Lexicon Installer allows you to install text-based lexicon file sets (for localization) on a remote host.

Standard lexicons are distributed as modules, for example: `niagaraLexiconFr` as the French lexicon, or `niagaraLexiconDe` for German. The lexicon tools include a lexicon module maker, to make new or updated lexicon modules from lexicon files. You can still install lexicon files using the **Lexicon Installer**, but to install lexicon modules you must use the platform **Software Manager** view.

Figure 57 Lexicon Installer view



To access this view expand **Platform** and double-click **Lexicon Installer**.

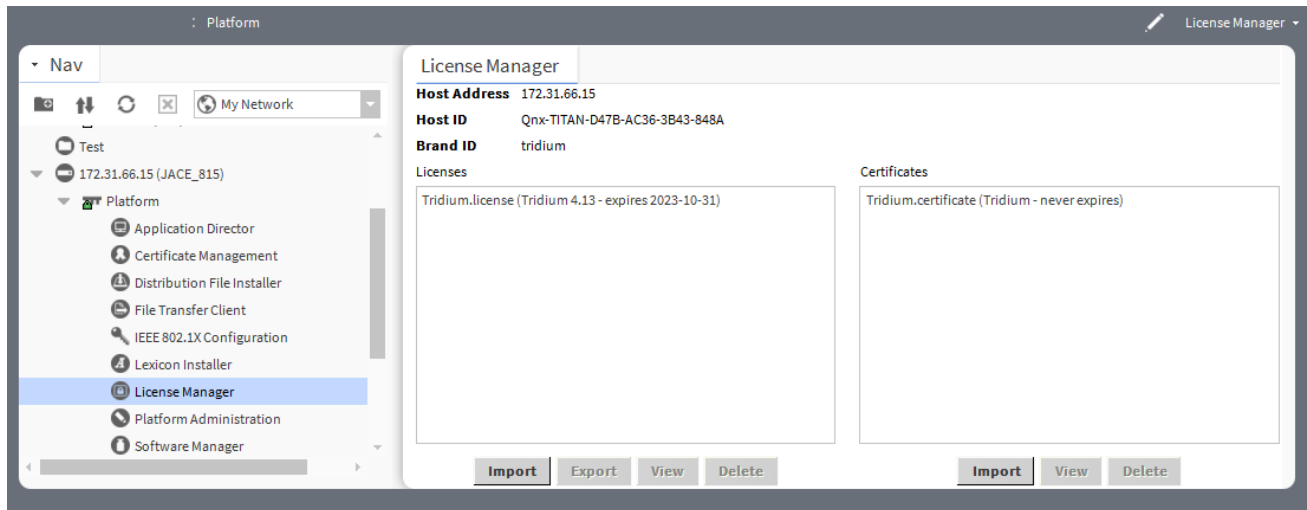
Buttons

- **Install** adds new lexicon from the directory.
- **Delete** deletes the selected lexicon module.

License Manager view (platDaemon-LicenseManager)

The **License Manager** allows you to view and install files required for Niagara licensing.

Figure 58 License Manager view



To access this view, expand **Platform** and double-click **License Manager**.

This view provides a two-pane window in which all the license files and certificates for the host are displayed:

- The left pane displays the license files associated with the Host Id.
- The right pane displays the certificate files associated with the Host Id.

Buttons

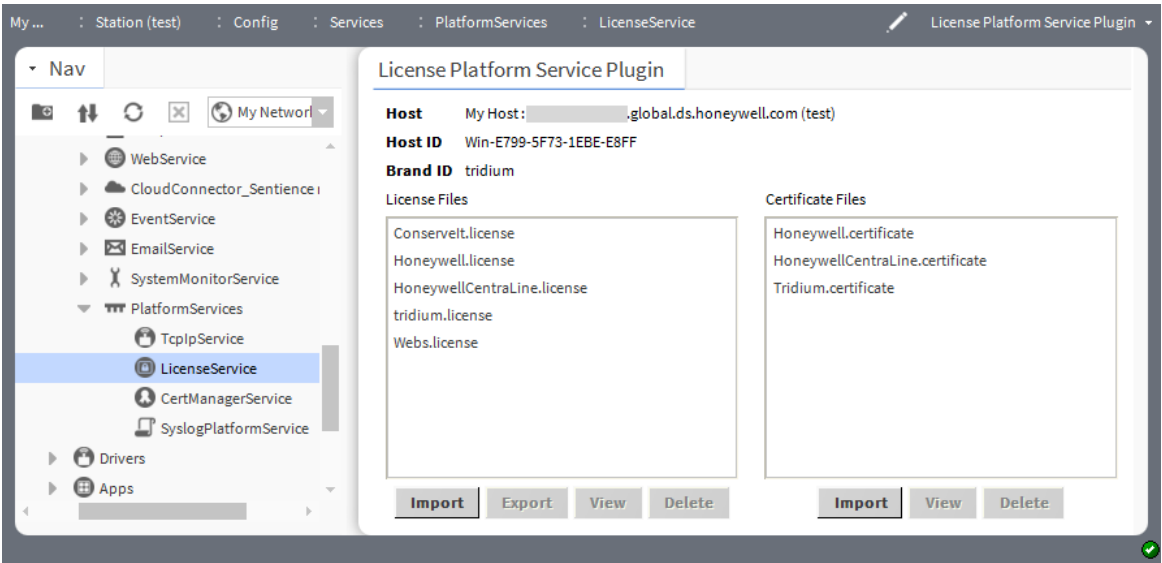
Buttons at the bottom of this view provide a way to manage the contents of your local license database, and are described as follows:

- **Import File** always available, this button adds license file(s) from a local license file or license archive (.lar) file.
- **Export File** always available, this button saves all licenses (or any selected licenses) locally, as a license archive file.
- **View** you can view the files.
- **Delete** deletes selected licenses from the local license database.
-

License Platform Service Plugin view (platform-LicensePlatformServicePlugin)

This plugin manages the controller host's licenses and certificates under a station's **PlatformServices** container. It provides the same interface as the **License Manager** view in a platform connection.

Figure 59 License Platform Service Plugin



To access this view expand **Config**→**Services**→**PlatformServices** and double-click **LicenseService**.

Buttons

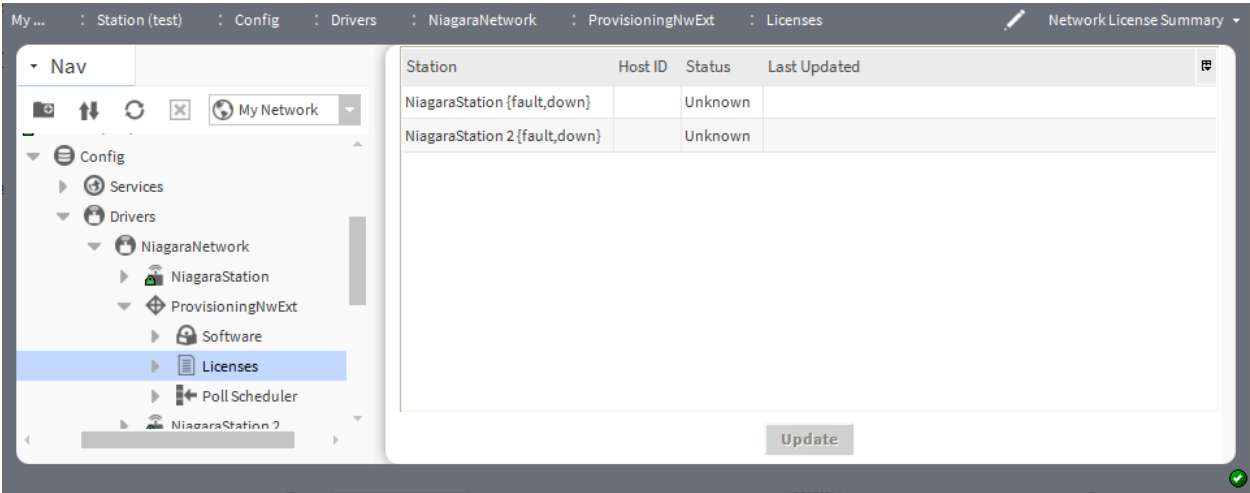
Buttons at the bottom of this view provide a way to manage the contents of your local license database, and are described as follows:

- **Import** always available, adds license file(s) from a local license file or license archive (.lar) file.
- **Export** always available, saves all licenses (or any selected licenses) locally, as a license archive file.
- **View** opens the selected license file.
- **Delete** deletes the selected license file.

Network License Summary view (provisioningNiagara-NetworkLicenseSummary)

This view provides lists the currently-known license information for each **NiagaraStation** in the network.

Figure 60 Network Licenses Summary view



Each row contains the license information for a host running a station.

To access this view expand **Config→Drivers→NiagaraNetwork→ProvisioningNwExt** and double-click **Licenses**.

Column	Value	Description
Station	text	Identifies the name of the station.
Host ID	text	Provides a 20-character identifier that uniquely identifies each host.
Status	Up-To-Date	A status of Up To Date means that the license on the remote host agrees with the license that the Supervisor has for it in its (own) local license database. It may be possible that a more recent license is available for it on the licensing server.
Last Updated	date and time	The timestamp when the station's license was last updated.

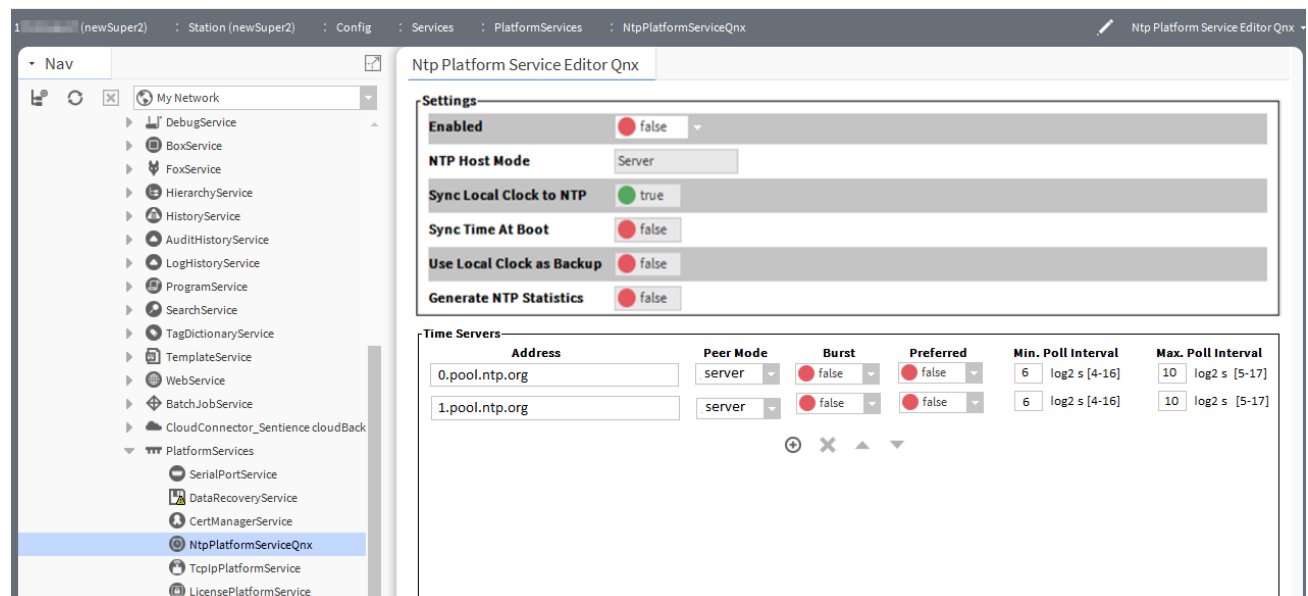
Ntp Platform Service Editor (platform-NtpPlatformServiceEditorNpsdk)

Ntp Platform Service Editor Npsdk is the default view of the station's **NtpPlatformServiceNpsdk**, which provides the platform interface to the NTP daemon (process) running on a controller. This view provides access to a few related settings, plus allows specifying one or more remote time servers.

Ntp Platform Service Editor Qnx view (platform-NtpPlatformServiceEditorQnx)

An example of an embedded **Ntp Platform Service Editor** for the JACE controller is shown below. This is the default view for the **NtpPlatformServiceQnx**.

Figure 61 Ntp Platform Service Editor Qnx view



This view provides access to some of the key settings of the NTP daemon (ntpd) of the QNX OS running on the host controller platform.

There are two main areas: **Settings** at top and **Time Servers** at bottom.

Ntp Platform Service Editor Qnx settings

This controller component settings in the **Ntp Platform Service Editor Qnx** include the following properties:

Property	Value	Description
Enabled	true or false (default)	If <code>true</code> , the host will use NTP to sync its clock with time values retrieved from other servers.
NTP Host Modes	Server (default)	Server (default): In addition to being an NTP client, this host will act as an NTP server. This configuration allows local network devices to use the host as a source of NTP data. Local network devices will also be able to query this host for additional NTP runtime information. Server (Time Only): In addition to being an NTP client, this host will act as an NTP server. This configuration allows local network devices to use the host as a source of NTP data. Unlike "Server", local network devices will be unable to query this host for additional NTP runtime information. Client: This host will act as an NTP client only. The NTP data retrieved by this host from configured servers will not be available to local network devices.
Sync Local Clock to NTP	true (default)	If true, the host will start the Network Time Protocol daemon. This boolean does not directly say that NTP will be used as a time source on this platform, rather, it enables that option. Additional properties below will inform the NTP daemon how time information now available will be used.
Sync Time At Boot	false (default)	If true, when the JACE boots, before the stations starts or the <code>ntpd</code> starts, it executes the <code>ntpdate</code> command. This updates the system local time.
Use Local Clock as Backup	false (default)	If true, should the specified NTP server(s) become unavailable at the time of a poll, the time used is provided by the system clock. This prevents the timing of the polling algorithm in the <code>ntpd</code> (which is executed at specified/changing intervals) from being reset. A true value does not result in any change to the NTP daemon's polling interval (frequency). In fact, by using the local system clock the NTP-calculated polling time would remain the same, and thus not result in more polling.
Generate NTP Statistics	false (default)	If true, the <code>NtpPlatformService</code> reports whatever information it can about its operation. To access these statistics with the station opened in Workbench, right-click the <code>NtpPlatformServiceQnx</code> and select Views → SpyRemote . Keep in mind that the <code>ntpd</code> is a QNX process; thus Niagara has no control over what it reports.

Ntp Platform Service Editor Qnx Time Servers

Each entry in the **Time Servers** list specifies a server to which the host's clock synchronizes when the service is **Enabled** (`true`), and **Sync Local Clock to NTP** is also `true`.

This also applies to any Niagara Portability Controller that implements NTP.

Controls below the list allow you to add  and delete  servers, as well as reorder up  or down  (to establish priority order, highest at top).

Property	Value	Description
Address	text	Fully qualified domain name, IP address, or host files alias for the NTP time server.
Peer Mode	drop-down list	Peer mode to use with the server, as either server or peer (symmetricActive).
Burst	true or false (default)	False by default. If true, when server is reachable, upon each poll a burst of eight packets are sent, instead of the usual one packet. Spacing between the first and second packets is about 16 seconds to allow a modem call to complete, while spacing between remaining packets is about two seconds.
Preferred	true or false (default)	If true, designates a server as preferred over others for synchronization. Note also that priority order (top highest, bottom lowest) is also evaluated if multiple servers are entered.
Min. Poll	number	Minimum poll interval for NTP messages, from 4 to 16. Units are in "log-base-two seconds," or 2 to the power of n seconds (NTP convention), meaning from 2 to the 4th (16 seconds) to 2 to the 16th (65,536 seconds).
Max. Poll	number	Maximum poll interval for NTP messages, from 10 to 17. Units are in "log-base-two seconds," or 2 to the power of n seconds (NTP convention), meaning from 2 to the 10th (1,024 seconds) to 2 to the 17th (131,072 seconds).

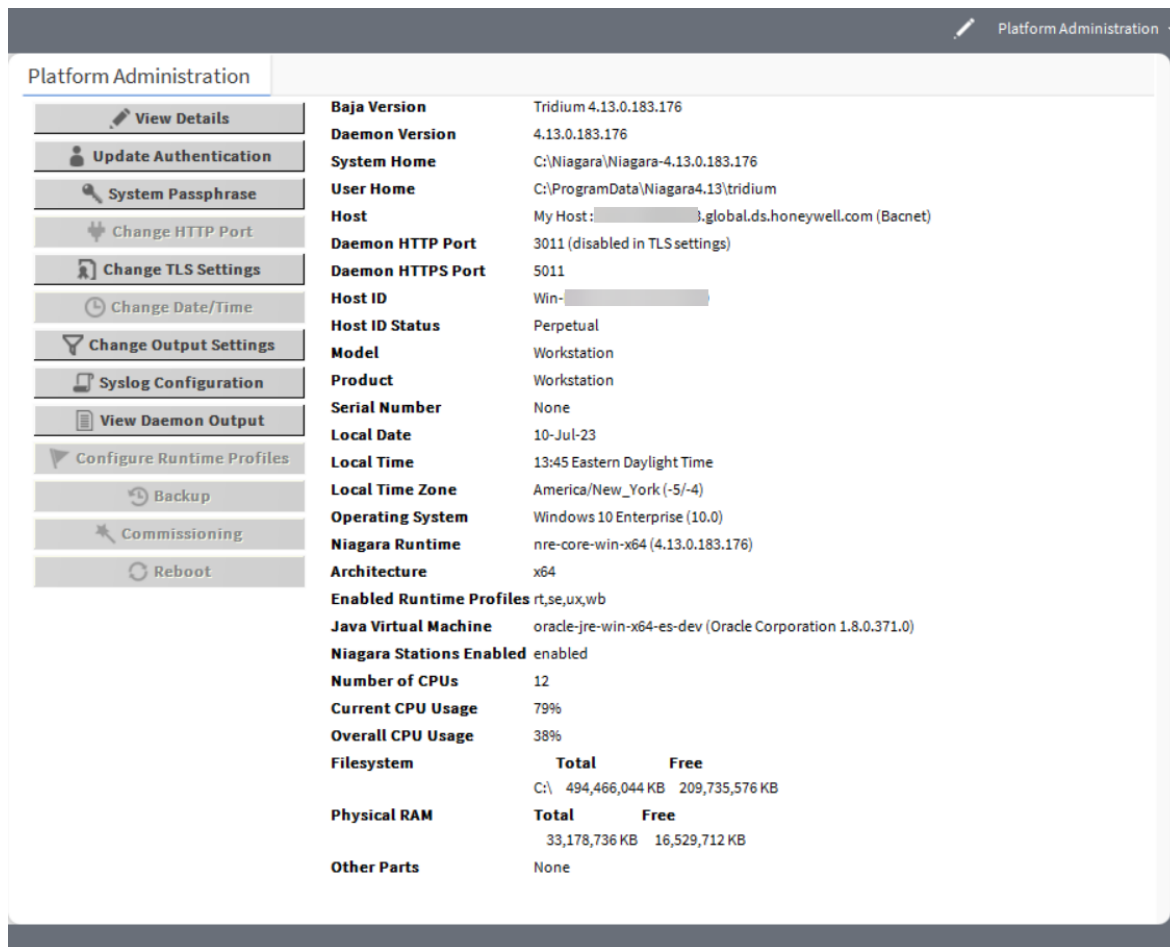
Actions

- **Sync Now** defines the fully-qualified domain name of a public NTP server or else the IP address of any accessible NTP server. Use this action only to verify that a provided NTP server is reachable and responding. You cannot use this action while NTP is enabled.

Platform Administration view (platDaemon-PlatformAdministration)

The **Platform Administration** view provides access to various platform daemon (and host) settings and summary information. Included are buttons to perform various platform operations.

Figure 62 Platform Administration



To access this view expand **Platform** and double-click **Platform Administration**.

Following are the options to access various platform daemon (and host) settings and summary information:

- **View Details** A platform summary that you can copy to the Windows clipboard.
- **User Accounts** A platform daemon authentication window to add, delete, or manage platform users (initially performed as a step in the Commissioning Wizard).
- **System Passphrase** A window to set or change the system passphrase used to encrypt sensitive information on the platform's filesystem.
- **Change HTTP Port** A window to change the HTTP port.
- **Change TLS Settings** A window to specify platform TLS settings, including port, certificate alias and its respective password, secure protocol to use, enable/disable extended master secret, and TLS Cipher Suite Group. For an overview, see Change TLS Settings in the *Niagara Platform Guide*.
- **Change Date / Time** A window to change the device's current date, time, and time zone (initially performed as a step in commissioning wizard).
- **Advanced Options** A window to enable or disable the following advanced platform options
 - **SFTP / SSH Enabled** A window to enable/disable SFTP and SSH access to the device, or change the default port number that these protocols use/share.
 - **Daemon Debug Enabled** Temporarily enable the browser based daemon debugging tools. This is automatically turned off the next time the system boots.

- **USB Backup Enabled** Enable or disable the USB Backup port on the device's enclosure.
- **Change Output Settings** A window to change the log level of different processes that can appear in the platform daemon output.
- **Syslog Configuration** A window to change Syslog configuration settings for message logging.
- **View Daemon Debug** A window in which you can observe debug messages from platform daemon processes in real time. Also includes ability to pause or load.
- **View System Log** A window for viewing system log(s) for the platform.
- **Configure Runtime Profiles** A window to change the types of runtime profiles for software modules installed on the device (initially performed in Commissioning Wizard).
- **Configure NRE Memory** A window to configure the memory allocation sizes of this platform's Niagara Runtime Environment.
- **Backup** Make a complete backup of all configuration on the connected host platform, including all station files, plus other Niagara configuration.
- **Commissioning** Another way to re-launch the Commissioning Wizard, as previously used in the initial commissioning of the device.
- **Reboot** A method to reboot the platform, which restarts all software including the OS and JVM, then the platform daemon, then if so configured in the Application Director (Station Director), the installed station. If you click this, a confirmation window appears.

If you reboot, your platform connection is lost, and it is typically a few minutes until you can reconnect to the device.

Platform Certificate Management (platCrypto-CertManagerService)

This view is the **Certificate Management** platform view on any Niagara host and the default view of the **CertManagerService** under a station's **PlatformServices**. Using this view you can create digital certificates and certificate signing requests (CSRs), and to import and export keys and certificates to and from the Workbench, platform and station stores.

Figure 63 Certificate Management view for "localhost"

Certificate Management for "localhost"

User Key Store System Trust Store User Trust Store Allowed Hosts

You have local certificates:

User Key Store 3 objects

Alias	Subject	Not After	Key Algorithm
supervisor-ncloud	N4:supervisor-nCloud:Tst-992F-13CF-3E1B-2348	Mon Jul 26 13:50:13 EDT 2027	EC
default	Niagara4	Fri Dec 22 10:38:44 EST 2023	RSA
newmansup0servercert	Sup0 Server Cert Claire's PC	Fri Sep 24 07:56:53 EDT 2021	RSA

Certificate Management for "localhost"

User Key Store System Trust Store User Trust Store Allowed Hosts

You have user certificates that identify these certificate authorities:

User Trust Store 1 objects

Alias	Subject	Not After	Key Algorithm	Key Size	Valid
ca root cert newman	CA Root Claire's PC	Fri Dec 22 10:38:44 EST 2023			true

To access this view for the localhost stores, connect to the platform, expand **Platform** and double-click **Certificate Management** or, in the station, expand **Config**→**Services**→**PlatformServices** and double-click **CertManagerService**.

If you are viewing this topic from a guide other than the *Niagara Station Security Guide*, refer to the *Niagara Station Security Guide* for more information.

User Key Store tab

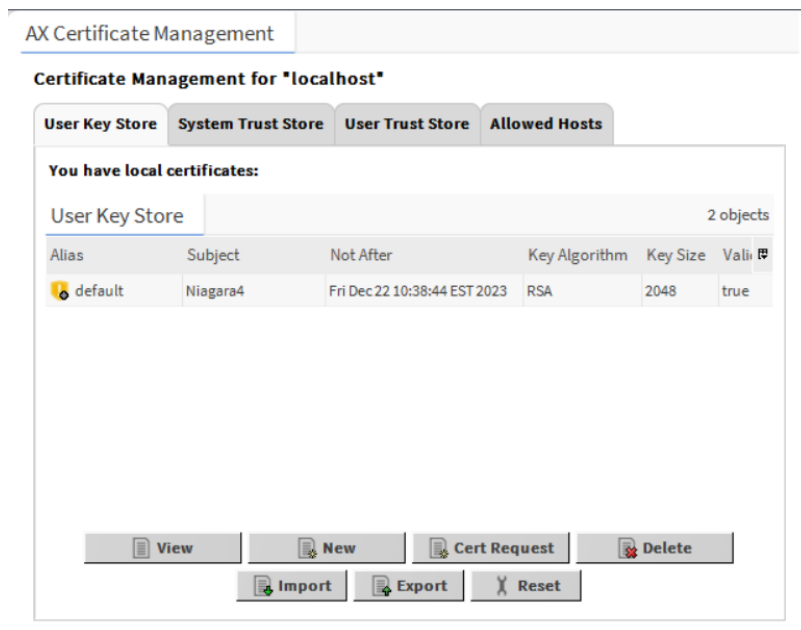
The **User Key Stores** contain server certificates and self-signed certificates with their matching keys. Each certificate has a pair of unique private and public encryption keys for each platform. A **User Key Store** supports the server side of the relationship by sending one of its signed server certificates in response to a client (Workbench, platform or station) request to connect.

If there are no certificates in a **User Key Store** when the server starts, such as when booting a new platform or station, the platform or station creates a default, self-signed certificate. This certificate must be approved as an allowed host. This is why you often see the certificate popup when opening a platform or station.

Default self-signed certificates have the same name in each **User Key Store** (default), however, each certificate is unique for each instance and is mainly used for recovery purposes. You cannot delete the default certificate.

Clicking the **New** and **Import** buttons also adds certificates to the **User Key Store**.

Figure 64 Example of a Key Store



Column	Description
Alias	Provides a short name used to distinguish certificates from one another in the Key Store . This property is required. It may identify the type of certificate (root, intermediate, server), location or function. This name does not have to match when comparing the server certificate with the CA certificate in the client's Trust Store.
Issued By	Identifies the entity that signed the certificate.
Subject	Specifies the Distinguished Name, the name of the company that owns the certificate.
Not Before	Specifies the date before which the certificate is not valid. This date on a server certificate should not be earlier than the Not Before date on the CA certificate used to sign it.
Not After	Specifies the expiration date for the certificate. This date on a server certificate should not be later than the Not After date on the CA certificate used to sign it.

Column	Description
	A period no longer than a year ensures regular certificate changes making it more likely that the certificate contains the latest cryptographic standards, and reducing the number of old, neglected certificates that can be stolen and re-used for phishing and drive-by malware attacks. Changing certificates more frequently is even better.
Key Algorithm	Refers to the cryptographic formula used to calculate the certificate keys.
Key Size	Specifies the size of the keys in bits. Four key sizes are allowed: 1024 bits, 2048 bits (this is the default), 3072 bits, and 4096 bits. Larger keys take longer to generate but offer greater security.
Signature Algorithm	Specifies the cryptographic formula used to sign the certificate.
Signature Size	Specifies the size of the signature.
Valid	Specifies certificate dates.
Self Signed	Indicates that the certificate was signed with its own private key.

User Key Store buttons

- **View** displays details for the selected item.
- **New** creates a new device record in the database.
- **Cert Request** opens a **Certificate Request** window, which is used to create a Certificate Signing Request (CSR).
- **Delete** removes the selected record from the database.
- **Import** adds an imported item to the database.
- **Export** saves a copy of the selected record to the hard disk.
For certificates, the file extension is .pem.
- **Reset** deletes all certificates in the **User Key Store** and creates a new default certificate. It does not matter which certificate is selected when you click **Reset**

CAUTION:

Do not reset without considering the consequences. The **Reset** button facilitates creating a new key pair (private and public keys) for the entity, but may disable connections if valid certificates are already in use. Export all certificates before you reset.

Trust Store tabs

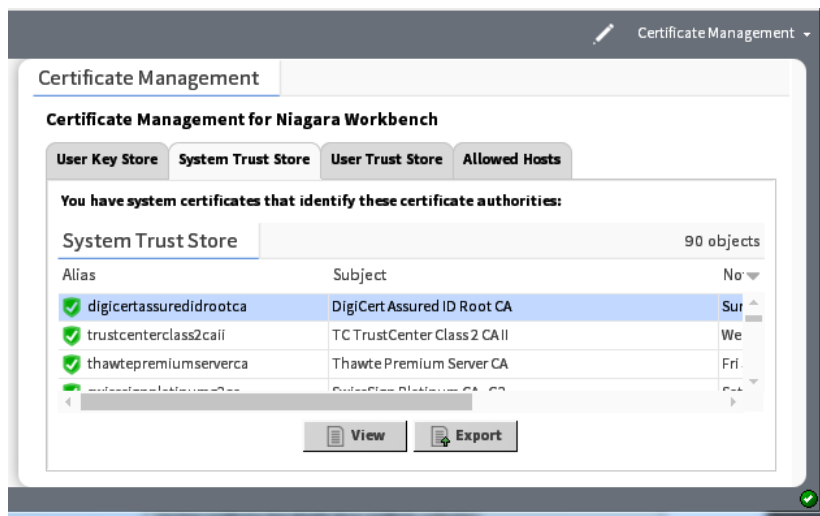
The **Trust Stores** contain signed and trusted root certificates with their public keys. These stores contain no private keys. A **Trust Store** supports the client side of the relationship by using its root CA certificates to verify the signatures of the certificates it receives from each server. If a client cannot validate a server certificate's signature, an error message allows you to approve or reject a security exemption (on the **Allowed Hosts** tab).

The **System Trust Stores** contain installed signed certificates by trusted entities (CA authorities) recognized by the Java Runtime Engine (JRE) of the currently opened platform. A **User Trust Store** contains installed signed certificates by trusted entities that you have imported (your own certificates).

Only certificates with public keys are stored in the **Trust Stores**. The majority of certificates in the **System Trust Store** come from the JRE. You add your own certificates to a **User Trust Store** by importing them.

Feel free to pass out such root certificates to your team; share them with your customers; make sure that any client that needs to connect to one of your servers has the server's root certificate in its client **Trust Store**.

Figure 65 System Trust StoreExample of a Trust Store



Trust Store columns

Column	Description
Alias	Provides a short name used to distinguish certificates from one another in the Key Store . This property is required. It may identify the type of certificate (root, intermediate, server), location or function. This name does not have to match when comparing the server certificate with the CA certificate in the client's Trust Store.
Issued By	Identifies the entity that signed the certificate.
Subject	Specifies the Distinguished Name, the name of the company that owns the certificate.
Not Before	Specifies the date before which the certificate is not valid. This date on a server certificate should not be earlier than the Not Before date on the CA certificate used to sign it.
Not After	Specifies the expiration date for the certificate. This date on a server certificate should not be later than the Not After date on the CA certificate used to sign it. A period no longer than a year ensures regular certificate changes making it more likely that the certificate contains the latest cryptographic standards, and reducing the number of old, neglected certificates that can be stolen and re-used for phishing and drive-by malware attacks. Changing certificates more frequently is even better.
Key Algorithm	Refers to the cryptographic formula used to calculate the certificate keys.
Key Size	Specifies the size of the keys in bits. Four key sizes are allowed: 1024 bits, 2048 bits (this is the default), 3072 bits, and 4096 bits. Larger keys take longer to generate but offer greater security.
Signature Algorithm	Specifies the cryptographic formula used to sign the certificate.
Signature Size	Specifies the size of the signature.
Valid	Specifies certificate dates.
Self Signed	Indicates that the certificate was signed with its own private key.

Trust Store buttons

The **Delete** and **Import** buttons are available only in a **User Trust Store**.

User Key Store buttons

- **View** displays details for the selected item.
- **Delete** removes the selected record from the database.

- **Import** adds an imported item to the database.
- **Export** saves a copy of the selected record to the hard disk.

For certificates, the file extension is .pem.

Allowed Hosts tab

The **Allowed Hosts** tab contains security exemptions for the currently open platform. These are the certificates (signed or self-signed) received by a client from a server (host) that could not be validated against a root CA certificate in a client's **Trust Store**. Whether you approve or reject the certificate, the system lists it in the **Allowed Hosts** list.

To be authentic, a root CA certificate in the client's **System** or **User Trust Store** must be able to validate the server certificate's signature, and the **Subject** of the root CA certificate must be the same as the **Issuer** of the server certificate.

Allowing exemptions makes it possible for a human operator to override the lack of trust between a server and client when the human user knows the server can be trusted.

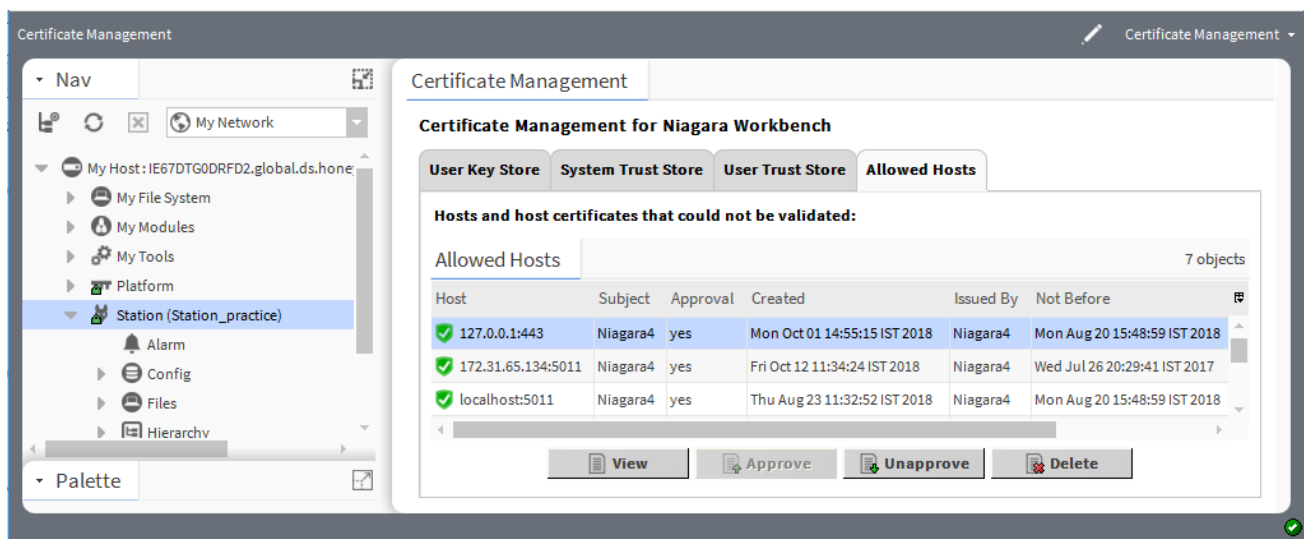
If this is a Workbench-to-station connection, the system prompts you to approve the host exemption. Workbench challenges server identity at connection time for unapproved hosts and, unless specific permission is granted, prohibits communication. Once permission is granted, future communication occurs automatically (you still have to log in). Both approved and unapproved hosts remain in this list until deleted.

If this is a station-to-station connection, and there is a problem with the certificates, the connection fails silently. There is no prompt to approve the host exemption. However, the last failure cause in the station reports the problem (expand the station **ClientConnection** under **NiagaraNetwork**).

The approved host exemption in the **Allowed Hosts** list is only valid when a client connects to the server using the IP address or domain name that was used when the system originally created the exemption. If you use a different IP address or domain name to connect to the server, you must approve an updated exemption. The same is true if a new self-signed certificate is generated on the host.

If you continue to use an approved self-signed certificate (rather than implement signed certificates, which are more secure), and the self-signed certificate's public key changes, the system negates the certificate, the green shield icon changes to a yellow shield icon with an exclamation mark (!), and the system returns an error. To approve this change, view the exemption (right-click the certificate row on the **Allowed Hosts** tab and click **View**) then approve the certificate by clicking **Accept**.

Figure 66 Example of an Allowed Hosts list



To open this view using Workbench, click **Tools→Certificate Management** and click the **Allowed Hosts** tab.

Allowed Hosts columns

Column	Description
Host	Reports the server, usually an IP address.
Subject	Specifies the Distinguished Name, the name of the company that owns the certificate.
Approval	Reports the servers within the network to which the a client may connect. If approval is <code>no</code> , the system does not allow the client to connect.
Created	Identifies the date the record was created.
Issued By	Identifies the entity that signed the certificate.
Not Before	Specifies the date before which the certificate is not valid. This date on a server certificate should not be earlier than the Not Before date on the CA certificate used to sign it.
Not After	Specifies the expiration date for the certificate. This date on a server certificate should not be later than the Not After date on the CA certificate used to sign it. A period no longer than a year ensures regular certificate changes making it more likely that the certificate contains the latest cryptographic standards, and reducing the number of old, neglected certificates that can be stolen and re-used for phishing and drive-by malware attacks. Changing certificates more frequently is even better.
Key Algorithm	Refers to the cryptographic formula used to calculate the certificate keys.
Key Size	Specifies the size of the keys in bits. Four key sizes are allowed: 1024 bits, 2048 bits (this is the default), 3072 bits, and 4096 bits. Larger keys take longer to generate but offer greater security.
Signature Algorithm	Specifies the cryptographic formula used to sign the certificate.
Signature Size	Specifies the size of the signature.
Valid	Specifies certificate dates.

Allowed Hosts buttons

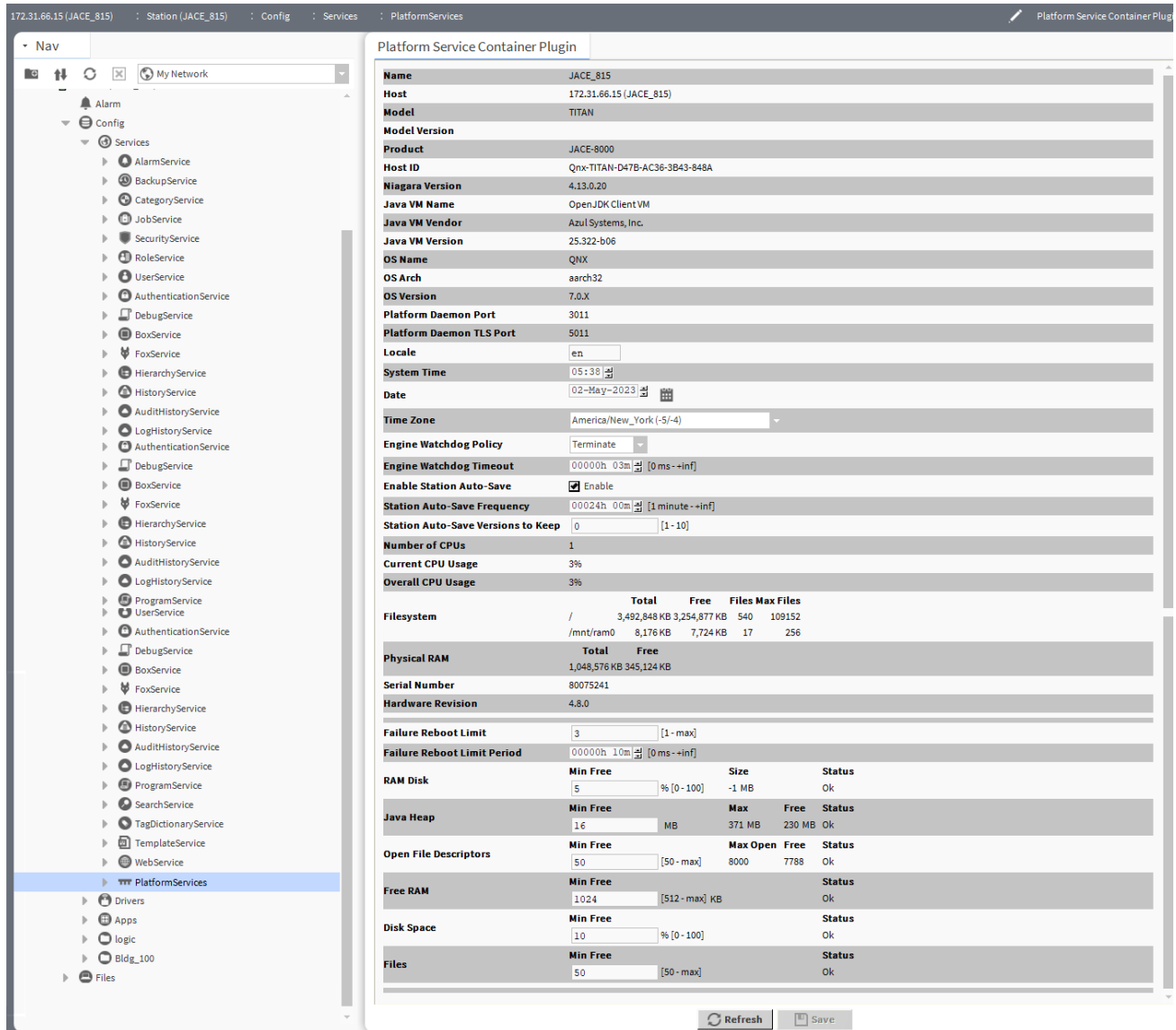
- **View** displays details for the selected item.
- **Approve** designates the server as an allowed host.
- **Unapprove** prohibits aconnection to this server host. The system terminates any attempted communication.

Platform Service Container Plugin (platform-PlatformServiceContainer)

The **Platform Service Container Plugin** allows you to view and edit platform properties on the host running an opened station. It is the default view for a station's **PlatformServices** container. It is also the container for other plugins, such as the **TcpIpService**, **LicenseService**, **CertManagerService**, and others depending on the host station operating system.

All plugins are also available as Hx pages.

Figure 67 Platform Service Container Plugin on a controller



To open this view expand **Config**→**Services** in a running station, scroll down and double-click **PlatformServices**.

The following properties are some that can be configured using this view.

NOTE: It is recommended that you leave engine-related properties and other advanced settings at default values, unless you have been directed otherwise by Systems Engineering.

Property	Value	Description
Name	read-only	Reports the name of the running station.
Host	read-only	Reports the IP address of the host platform.
Model	read-only	Reports the controller model name or Workstation.
Model Version	read-only	Reports the version number of the host model platform.
Product	read-only	Reports the product.

Property	Value	Description
Host ID	read-only	Reports the controller's ID, a string of unique alpha and numeric characters that identify this platform.
Niagara Version	read-only	Reports the version of the Niagara distribution running in the host platform.
Java VM Name	read-only	Reports the Java virtual machine used, for example, "Java HotSpot(TM) Embedded Client VM" for any N4 controller, or "Java HotSpot(TM) 64-Bit ServerVM" for a Supervisor on a Windows host.
Java VM Vendor	read-only	Reports the vendor for Java VM: Oracle Corporation.
Java Vm Version	read-only	Reports the version of Java VM, for example, "25.0-b 70" for the Java 8 compact3 VM on a controller, or "25.31-b07" for the Java 8 SE VM on a Windows host.
OS Name	read-only	Reports the operating system name, such as "QNX" or "Windows 10."
OS Arch	read-only	Machine architecture for OS, such as "arm" or "ppc" (controller hosts) or "amd64" (Windows hosts).
Os Version	read-only	Operating System version, such as "6.5.0" (QNX) or "10.0" (Windows 10).
Platform Daemon Port	read-only	Reports the port number on which the platform daemon that started the station is listening for its platform server (3011, or another port number). This can prove useful in case you changed the platform port, but then forgot what the new port is.
Platform Daemon TLS Port	read-only	Reports the port number on which the platform daemon is listening for its platform TLS server: 5011 or another port number provided that platform TLS is enabled. If platform TLS is disabled, this property reads <code>Unknown</code> . This can prove useful in case you changed the platform TLS port, and then forgot what the new port is.
Locale	string	Determines locale-specific behavior, such as date and time formatting, and also which lexicons are used. A string entered must use the form: <code>language["_" country["_" variant]]</code> . For example U.S. English is <code>"en_US"</code> and traditional Spanish would be <code>es_ES_Traditional</code> . See Oracle documentation at http://docs.oracle.com/javase/1.4.2/docs/api/java/util/Locale.html for related details.
System Time	read-only if a Windows host; hours minutes seconds if a controller host	Reports or configures the current local time in host.
Date	read-only if a Windows host; date if a controller host	Reports or configures the current local date in host.
Time Zone	read-only if a Windows host; drop-	Reports or configures the current local time zone for host.

Property	Value	Description
	down list if a controller host	
Engine Watchdog Policy	drop-down list (defaults to <code>Terminate</code>)	Defines the response taken by the platform daemon if it detects a station engine watchdog timeout. The engine watchdog is a platform daemon process, to which the station periodically reports its updated engine cycle count. The purpose of the watchdog is to detect and deal with a hung or stalled station, and is automatically enabled when the station starts. <code>Log Only</code> generates a stack dump and logs an error message in the system log. The station should ultimately be restarted if a watchdog timeout occurs with the <code>Log Only</code> setting. <code>Terminate</code> kills the VM process. If Restart on Failure in the Application Director, the station restarts. <code>Reboot</code> automatically restarts the host controller platform. If Auto-Start is enabled in the Application Director, the station restarts after the platform reboots.
Engine Watchdog Timeout	hours and minutes (defaults to 3 minutes)	If the station's engine cycle count stops changing and/or the station does not report a cycle count to the platform daemon, defines how long to wait before generating a stack dump for diagnostic purposes. The platform daemon causes the VM to generate the stack dump, then the daemon takes the action defined by the Engine Watchdog Policy .
Enable Station Auto-Save	check box (defaults to enabled)	Configures if Auto-Save should be enabled or disabled. Auto-Save creates a <code>config_backup_<YYMMDD>_<HHMM>.bog</code> file (where <code><YYMMDD></code> is the date and <code><HHMM></code> is the time when the automatic backup was created. Station Auto-Save Frequency defines when the save occurs. Auto-saved backup files are kept under that station's folder.
Station Auto-Save Frequency	hours and minutes (defaults to every 24 hours for controller platforms and every hour for a Windows host)	Defines how frequently to create an auto-save backup of a station's BOG file.
Station Auto-Save Versions to Keep	number (defaults to zero (0) on a controller platform and three (3) for a Windows host)	Configures the number of backups to save. Once this limit is reached, the framework replaces the oldest of the backups with the next manual or automatic save. Changing the default value for a controller from 0 to 1 provides a benefit in the case where a catastrophic (yet inadvertent) station change is made. If this happens a station kill reverts the station back to the backup copy. In Windows hosts, you can safely increase the default number to save more backups.
Battery Present	<code>true</code> (default) or <code>false</code>	(Applies only to a host other than the JACE-8000 or JACE-9000) Specifies if the controller has an integral backup battery, typically an onboard NiMH battery.

Property	Value	Description																				
		<code>true</code> is recommended unless the controller is both SRAM-equipped and is without an attached backup battery (there is no way to detect the latter through software). <code>false</code> disables the PowerMonitorService at the next reboot. This prevents nuisance battery bad alarms. Station backup is dependent totally on SRAM and the station's DataRecoveryService (the controller must have the <code>platDataRecovery</code> module installed, and be licensed for DataRecovery). The configuration described above is only one of three possible backup options for an SRAM-equipped controller that can also have a backup battery installed (for example, JACE-6E and JACE-3E, or JACE-6 and JACE-7 with an SRAM option card). The two other options are to use both a backup battery and SRAM for backup, or to use only a backup battery and no SRAM. These options require that Battery Present property is set to <code>true</code>. For related details, refer to the document <i>Niagara Data Recovery Service Guide</i>.																				
Number of CPUs	read-only	Reports the number of CPUs used in the host platform . This is typically 1 if a controller, more if a Windows host.																				
Current CPU Usage	read-only	Reports the percentage of CPU utilization in the last second.																				
Overall CPU Usage	read-only	Reports the percentage of CPU utilization since the last reboot.																				
Filesystem	read-only	Reports file storage statistics for the host, including total file space, available (free) space, and file block size (minimum size for even the smallest file). For the JACE-8000 host, it may look similar to: <table><tr><td>Total</td><td>Free</td><td>Files</td><td>Max Files</td><td></td></tr><tr><td>/</td><td>3,476,464 KB</td><td>3,039,088 KB</td><td>602</td><td>108640</td></tr><tr><td>/mnt/aram0</td><td>393,215 KB</td><td>381,019 KB</td><td>0</td><td>0</td></tr><tr><td>/mnt/ram0</td><td>8,192 KB</td><td>8,192 KB</td><td>0</td><td>0</td></tr></table>	Total	Free	Files	Max Files		/	3,476,464 KB	3,039,088 KB	602	108640	/mnt/aram0	393,215 KB	381,019 KB	0	0	/mnt/ram0	8,192 KB	8,192 KB	0	0
Total	Free	Files	Max Files																			
/	3,476,464 KB	3,039,088 KB	602	108640																		
/mnt/aram0	393,215 KB	381,019 KB	0	0																		
/mnt/ram0	8,192 KB	8,192 KB	0	0																		
Physical RAM	read-only	Current total and free RAM statistics for the host. For the JACE-8000, it may look similar to: <table><tr><td>Total</td><td>Free</td></tr><tr><td>1,048,576 KB</td><td>113,424 KB</td></tr></table>	Total	Free	1,048,576 KB	113,424 KB																
Total	Free																					
1,048,576 KB	113,424 KB																					
Serial number	read-only	(Appears only for the JACE host). Reports the controller's unique serial number.																				
Hardware Revision	read-only	(Appears only for the JACE host). Reports the hardware revision of the controller.																				
Hardware Jumper Preset		(Applies only to the JACE host, except for the JACE-8000) Either true or false—indicates whether or not the mode jumper is installed for "serial shell mode" access. Read at boot time only. Refer to the <i>JACE Niagara 4 Install and Startup Guide</i>																				
Failure Reboot Limit	number (defaults to 3)	(Controller platforms only) Limits the number of station restarts that can be triggered by station failures, within the																				

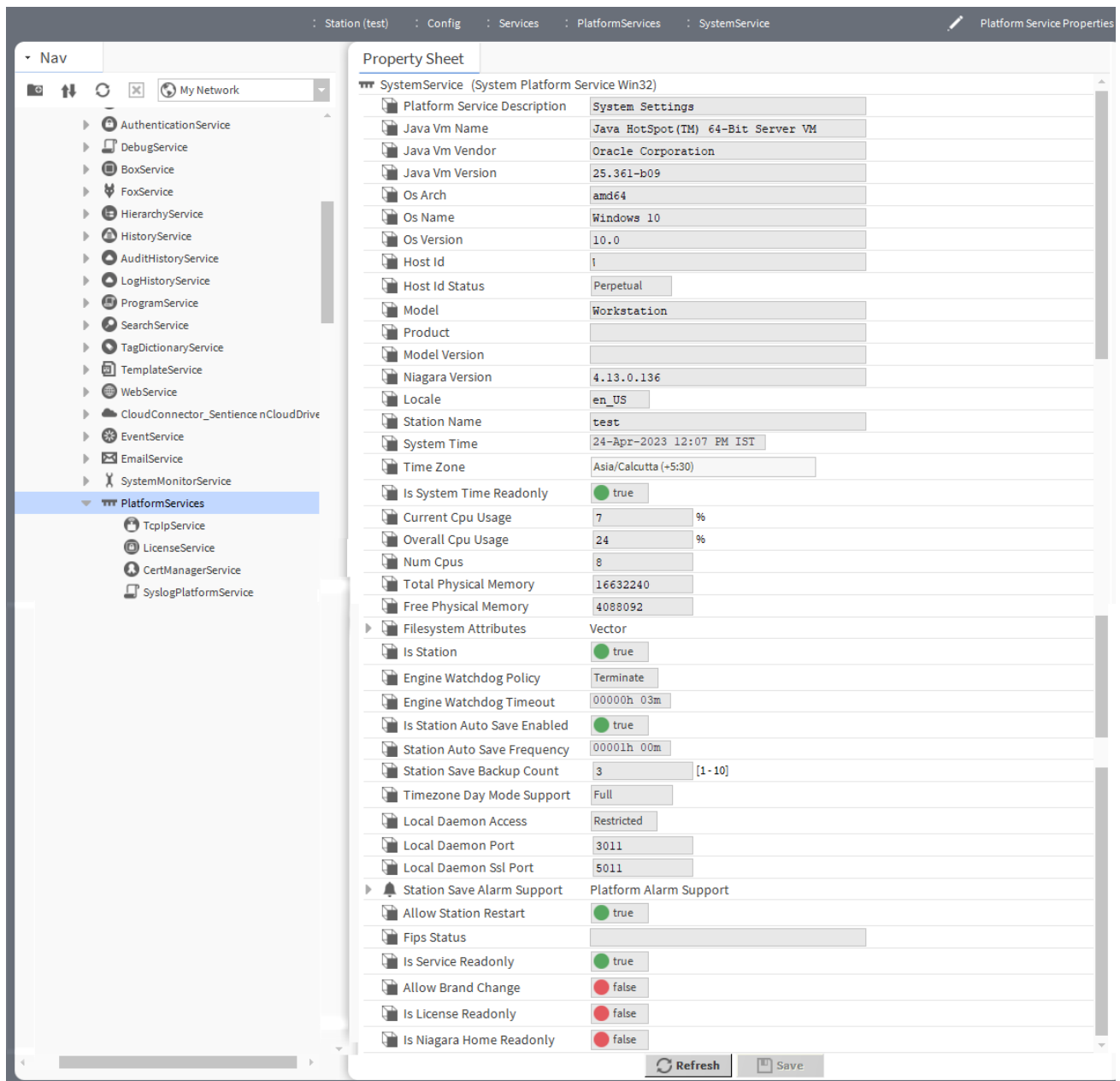
Property	Value	Description
		Failure Reboot Limit Period (if the host is so configured using the Application Director).
Failure Reboot Limit Period	hours and minutes (defaults to 10 minutes)	(Controller platforms only) Specifies the repeating frequency of the Failure Reboot Limit period. These two failure reboot settings are also adjustable (in any version of a QNX-based host) within that controller's <code>!daemon/daemon.properties</code> file, in the following two properties: <code>failureRebootLimit=x</code> where <code>x</code> is an integer, default is 3 <code>failureRebootLimitPeriod=y</code> (where <code>y</code> is long in milliseconds, default is 3600000)
RAM Disk	percentage (defaults to 5%)	Defines the amount of remaining RAM Disk that triggers a PlatformServices and Platform Administration warning. Size indicates the size of the RAM disk used to store history and alarm files. This value defaults to 64mb. Status reports the current status of RAM disk.
Java Heap, Min Free	MB (The default varies according to controller model.)	Specifies the minimum free Java heap size, in MB against which the station compares (tests) for low memory conditions, that is excessive Java heap. This test automatically runs once a minute. If the heap free byte count is less than the defined minimum free heap size, a low memory warning displays in all Workbench views of the station. The warning is a yellow message box overlaid on any new view accessed, or on any current view that is refreshed. This warning is removed when the heap free byte count rises above the defined minimum size—such as might occur if enough components are deleted from the station. All memory statistics, including those for heap, are accessible on a station opened in Workbench, via the Resource Manager view of the Station component.
Open File Descriptors, Min Free	number	Specifies the maximum amount of file descriptors that can be used. That is, the read-only Max Open number minus the Min Free amount. File descriptors are used for histories, modules, and Fox connections. If exceeded a "Station has too many open files or sockets" warning is overlaid in all Workbench views of the station.
Free RAM, Min Free	KB	Specifies the minimum RAM that can be left free during station operation. If status is not <code>Ok</code> , a "Low free RAM" warning overlays all Workbench views of the station.

Property	Value	Description
Disk Space, Min Free	percentage	Specifies the minimum percentage of disk storage that can be left free during station operation. Below this amount, a “Platform running low on disk space” warning overlays all Workbench views of the station.
Files, Min Free	number	Specifies the minimum number of free files available during station operation. Below this number, a related platform warning appears. The PlatformServiceContainer status property Filesystem includes both the current number of files and the maximum number of files for each JACE controller partition.

SystemService (under PlatformServices)

The **PlatformServices** child **SystemService** container is accessible from its property sheet. Unlike other child services, **SystemService** does not appear in the Nav tree.

Figure 68 System Service properties



To access, click **Config**→**Services** and right-click **PlatformServices**→**Ax Property Sheet** then expand **SystemService**.

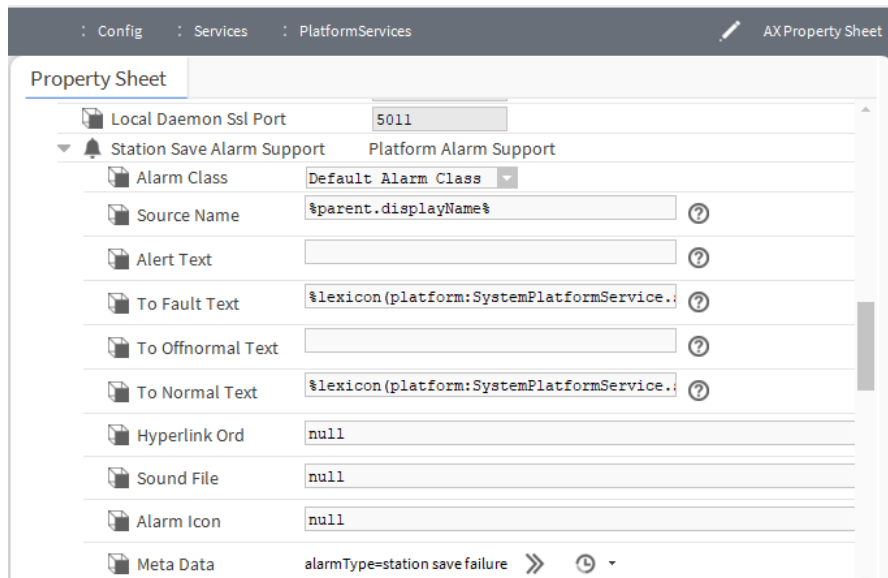
Property	Value	Description
Platform Service Description	read-only	Displays the service description for the Platform.
Java Vm Name	read-only	Displays the details of Java Vm.
Java Vm Vendor	read-only	Displays the vendor name of Java Vm.
Java Vm Version	read-only	Displays the version of Java Vm.

Property	Value	Description
Os Arch	read-only	Displays the operating system architecture details.
Os Name	read-only	Displays the operating system name.
Os Version	read-only	Displays the operating system versions.
Host Id	read-only	Displays the host ID is generated upon installation of the Niagara software, and typically begins with Win-, for example Win-5BE1-B094-FC24-3440.
Host Id Status	read-only	Displays the status of host (for example, perpetual).
Model	read-only	Displays model of Niagara.
Product	read-only	Displays the product details.
Model Version	read-only	Displays model version of Niagara.
Niagara Version	read-only	Display Niagara version in use.
Locale	read-only	Determines locale-specific behavior such as date and time formatting, and also which lexicons are used. A string entered must use the form: language ["_" country ["_" variant]]. For example, U.S. English is "en_US" and traditional Spanish would be "es_ES_Traditional".
Station Name	read-only	Current local time in host.
System Time	read-only	Current local date in host.
Time Zone	read-only	Current local time zone for host.
Is System Time Readonly	true (default)	If set to <code>true</code> the displayed time is read only.
Current Cpu Usage	read-only	Displays the current Cpu usage in percentage.
Overall Cpu usage	read-only	Displays the overall Cpu usage in percentage.
Num Cpus	read-only	Displays the number of logical processors of the system.
Total Physical Memory	read-only	Displays the physical memory is currently being utilized by the Server.
Free Physical Memory	read-only	Displays the physical memory is currently not being utilized by the Server.
File System Attributes	additional properties	Configures additional attributes for files.
Is Station	read-only	
Engine Watchdog Policy	read-only	The engine watchdog is a platform daemon process, to which the station periodically reports its updated engine cycle count. The watchdog purpose is to detect and deal with a "hung" or "stalled" station, and is automatically enabled when the station starts. The Engine Watchdog Policy defines the response taken by the platform daemon if it detects a station engine watchdog time-out. Watchdog policy selections include:

Property	Value	Description
		- Log Only — Generates stack dump and logs an error message in the system log. (The station should ultimately be restarted if a watchdog timeout occurs with the “Log Only” setting). - Terminate — (Default) Kills the VM process. If “restart on failure” is enabled for the station (typical), the station is restarted. - Reboot — Automatically reboots the host JACE platform. If “auto-start” is enabled for the station, the station is restarted after the system reboots.
Engine Watchdog Timeout	read-only	Default is 3 minute, and range is from 0 ms to infinity. If the station’s engine cycle count stops changing and/or the station does not report a cycle count to the platform daemon within this defined period, the platform daemon causes the VM to generate a stack dump for diagnostic purposes, then takes the action defined by the Engine Watchdog Policy.
Is Station Auto-Save Enabled	true(default)	If set to <code>true</code> saves the station data automatically.
Station Auto-Save Frequency	read-only	Default is every 24 hours for any JACE platform, or every (1) hour if a Windows host. Range is from 1 to many hours.
Station Save Backup Count	read-only	Displays the number how many times the station backup is stored.
Timezone Day Mode Support	read-only	
Local Daemon Access	read-only	Displays the type of access for local daemon.
Local Daemon Port	read-only	Displays the port number for local daemon.
Local Daemon Ssl Port	read-only	Displays the Ssl port number for local daemon.
Station SaveAlarm Support	additional properties	Configure additional parameters to get the alarm on station save.
Allow Station Restart	true (default)	If set to <code>true</code> allows the station to restart on failure.
Fips Status	read-only	
Is Service Readonly	read-only	
Allow Brand Change	read-only	
Is License Readonly	read-only	
Is Niagara Home Readonly	read-only	

When you expand **SystemService**, you see most of the same properties available in the default **Platform Service Container Plugin** view. In addition, there is a container slot **Station Save Alarm Support**.

Figure 69 Station Save Alarm Support expanded in property sheet of SystemService.



Properties under **Station Save Alarm Support** allow you to configure the alarm class and other parameters to use for **station save** alarms. Such an alarm may occur, for example, if there is insufficient disk space to complete the save.

Properties work the same as those in an alarm extension for a control point.

NOTE: Other platform warnings from defined limits, such as for low memory, low disk space, and so on are not really alarms—they simply generate a yellow overlay in the lower right corner when viewing the station in Workbench. If you need actual alarms, you can link from an appropriate boolean slot of the **SystemService** component (for example, LowHeap) into other persisted station logic in another area of the station.

If linking to **PlatformServices**, be aware that you should change the link type from handle to slot path.

Power Monitor Platform Service Plugin (platPower-PowerMonitorPlatformServicePlugin)

The **Power Monitor Platform Service Plugin** is the default view on the platform service **PowerMonitor-Service** in most JACE controller models. This view provides parameters for changing the shutdown delay time, as well as alarm source configuration settings.

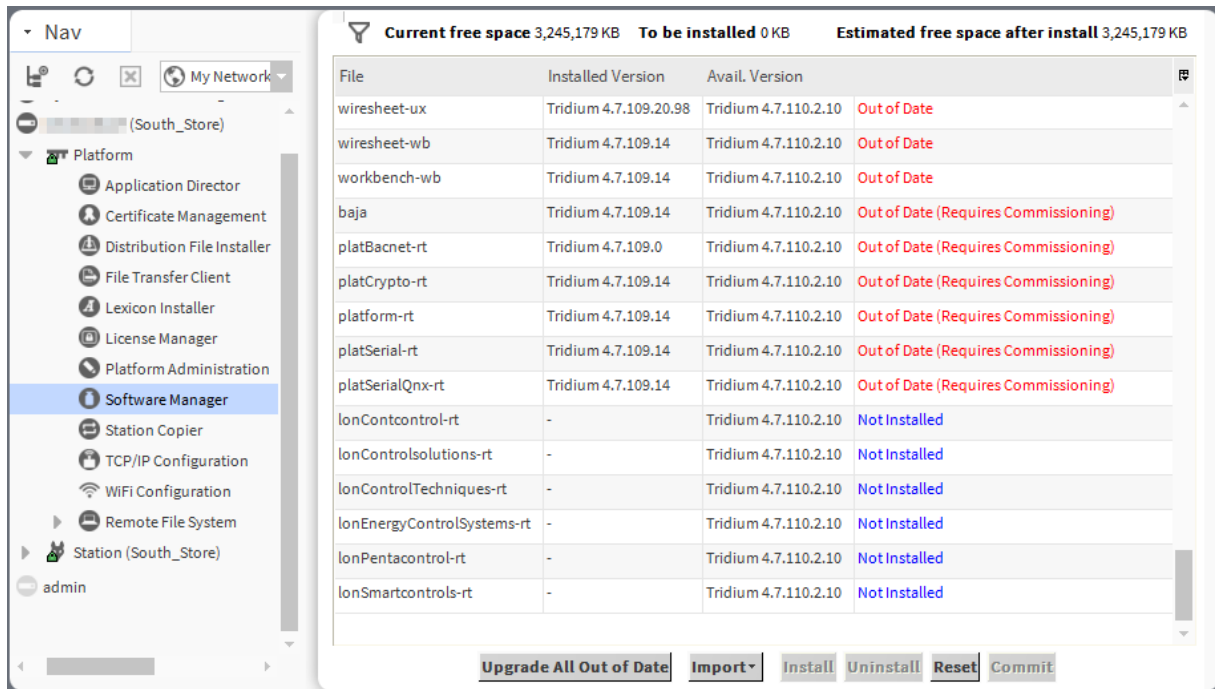
Typically, support is enabled and configured at JACE commissioning time. For related details, see JACE power monitoring configuration in the latest *Niagara Data Recovery Service Guide*.

Software Manager view (platDaemon-SoftwareManager)

The **Software Manager** is the platform view you use to install, upgrade, or remove modules in the connected Niagara platform.

By default, the **Software Manager** lists all the remote platform's out-of-date modules at the top of the table, then uninstalled modules, and lastly up-to-date modules (sorted alphabetically).

Figure 70 Software Manager view



To open this view, connect to a remote controller, expand the **Platform** node in the Nav tree and double-click **Software Manager**.

Above the table the manager provides data storage information: Current free space (KB), To be installed KB and Estimated free space after installation (KB).

The table provides these columns:

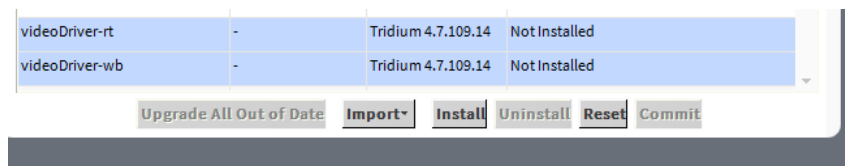
Column	Description
File	Displays the name of locally available module file or blank if the module is on the remote host only.
Installed Version	Displays the version of the module installed in the remote host or blank if it is not installed.

Column	Description
Avail. Version	Displays the latest version of the module that is locally available or blank if the software is on the remote host only.
unlabeled	Status of the module in the remote platform. For each module, status is one of the following: • Not Installed (blue text) indicates that the module is not in remote platform, but is available locally in your PC's software database. • Not Installed (Requires Commissioning) (blue text) indicates that the module is not in the remote platform, is available locally and requires you to first use the Commissioning Wizard to upgrade the remote platform. • Up to Date indicates that the module is installed in the remote platform and its version is equal to or higher than the locally available module version in your PC's software database. • Out of Date (red text) indicates that the module is installed in remote platform and its version is older than your local version in your PC's software database. • Out of Date (Requires Commissioning) (red text) indicates that the module is installed in remote platform, its version is older than the available local version and requires you to first use the Commissioning Wizard to upgrade the remote platform. • Not Available Locally indicates that the module installed in remote platform is not in your local software database. • Cannot Install indicates that the local module is unreadable or has a bad manifest. You cannot install it. • Bad Target indicates that a remotely installed module is unreadable or has a bad manifest and is therefore unusable by a station. Software in this state should probably be fixed since it could cause the station to not work correctly. • Downgrade to <version> indicates that the remotely installed software is intended to be replaced with a module having a lower version. • Install <version> indicates that the module is intended to be installed. It does not currently exist on the remote platform. • Re-Install <version> indicates that the remotely installed module is intended to be replaced with a module with the same version. • Uninstall <version> indicates that the remotely installed module is intended to be uninstalled. • Upgrade to <version> indicates that the remotely installed module is intended to be replaced with a module with a higher version. NOTE: Intended status values like Install <version> reflect un-committed actions made during your Software Manager session. Blue text is used to list these statuses.

Buttons

The **Software Manager** enables these buttons when you select one or more modules in the table.

Figure 71 Software Manager action buttons



- **Upgrade All Out of Date** replaces older version modules in the remote controller with current modules from the local software database.
- **Import** allows the import of software from files, directory or remote host.
- **Rebuild Module Signatures** rebuilds the module signatures when the station is not running.
- **Install/Re-Install/Upgrade/Downgrade** changes the button name based on what can be done with one or more installed modules that are selected.
 - **Re-Install** appears if the installed item is the same version as your locally available one.

- **Upgrade** appears if the installed item is an earlier version than your locally available one.
- **Downgrade** appears if the installed item is an newer version than your locally available one.

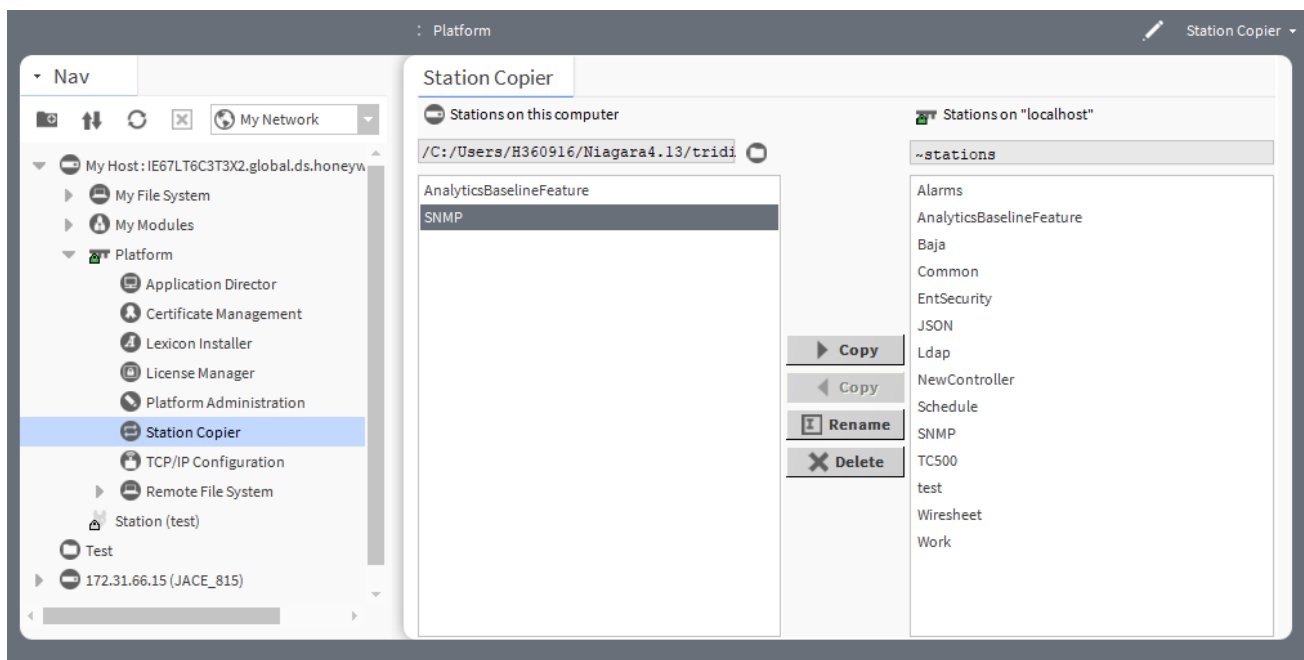
When you click this button, the software's status correspondingly changes to either **Re-Install <version>**, **Upgrade <version>**, or **"Downgrade <version>**, and the button changes to **Cancel <action>**, for example: **Cancel Re-Install**.

- **Uninstall** removes the selected module(s) from the remote platform assuming that no other modules are dependent on the selected module(s).
- **Reset** clears all flagged module changes since the last commit and disables the **Commit** button.. You cannot click **Reset** after clicking **Commit**.
- **Commit** initiates the software action when you have one or more pending actions in place on software items. This is how you launch or initiate the software action (flagged changes).

Station Copier view (platDaemon-StationCopier)

The **Station Copier** is the platform view used to install a station in either a remote or local Niagara platform, as well as make a copy on your local PC of a remote or a locally running station. You can also delete and rename stations using this view.

Figure 72 Station Copier view



To access this view expand **Platform** and double-click **Station Copier**.

As shown above, the **Station Copier** view is split into two main areas:

- Stations on your Workbench PC, typically your User Home, are shown on the left.
- Stations in the daemon User Home of the opened platform are on the right.

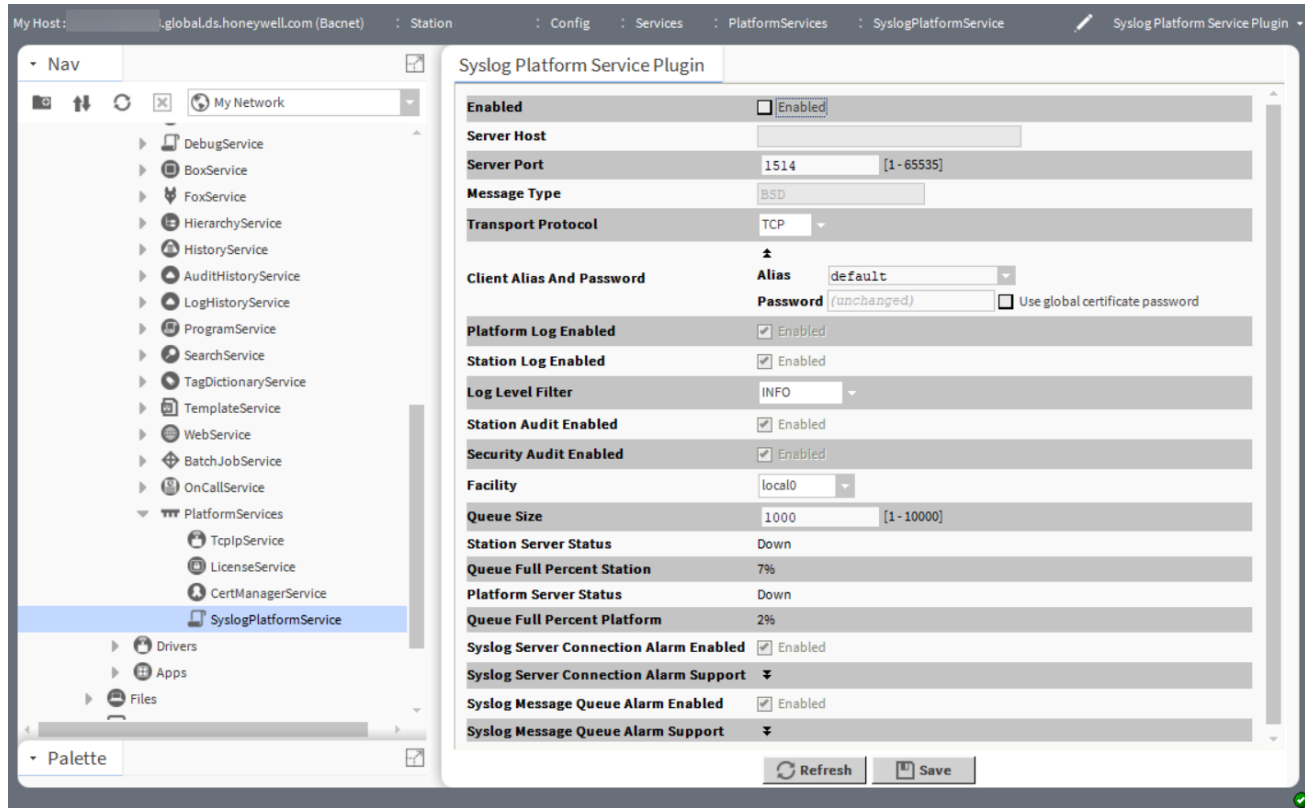
Buttons

- **Copy** copies the selected station to localhost or viceversa.
- **Rename** renames the selected station.
- **Delete** deletes the selected station.

Syslog Platform Service Plugin view (platform-SyslogPlatformServicePlugin)

As of Niagara 4.13, this plugin manages the syslog configuration settings for message logging, which allows messages that are generated by to be stored and analyzed on a remote server.

Figure 73 Syslog Platform Service Plugin view



To access the **Syslog Platform Service Plugin** view, expand **Config**→**Services**→**Platform Services** and double-click **SyslogPlatformService**. In addition, you can also access this plugin via browser connection to your platform.

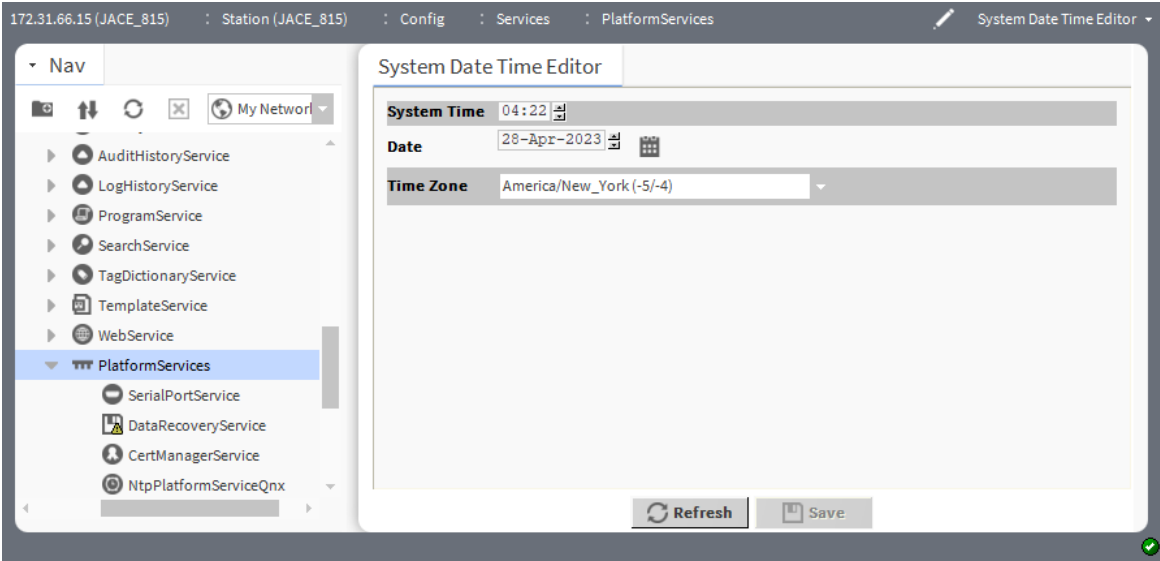
Type	Value	Description
Enabled	false (default) or true	Disables (false) or enables (true) the system log service.
Server Host	IP address	Specifies the IP address or hostname of the Syslog Server.
Server Port	number (defaults to 1514)	Specifies the port for communication.
Message Type	read-only (defaults to BSD)	Specifies the type of message supported. Currently only the BSD type is supported.
Transport Protocol	drop-down list	Specifies the transport protocol used for communicating messages to the server.
Client Alias And Password	text	This is only required if the syslog server requires mutual TLS (mTLS) protocol. This property defines the client certificate in the User Key Store to use. Refer in Niagara Station Security

Type	Value	Description
		Guide to “Creating a Client Certificate for Syslog configuration” for more information on generating Client Certificates.
Platform Log Enabled	true (default) or false	Enables (true) or disables (false) the platform logs sent to the server.
Station Log Enabled	true (default) or false	Enables (true) or disables (false) the station logs sent to the server.
Syslog Log Level Filter	Off, Severe, Warning, Info, Config, Fine, Finer, Finest, All (defaults to Info)	Sets the minimum level of platform and station logs that will be sent to the syslog server.
Station Audit	true (default) or false	Enables (true) or disables (false) the station audit records sent to the server.
Security Audit	true (default) or false	Enables (true) or disables (false) the security audit records sent to the server.
Facility	drop-down list (defaults to local0)	Specifies the facility (or process) which generated the syslog messages.
Queue Size	number (defaults to 1000)	Specifies the queue size to hold the messages until they are sent.
Station Server Status	read-only (disabled)	Displays the status of last station log message sent to the syslog server.
Queue Full Percent Station	read-only (defaults to 0%)	If connection is down, percentage of queue of station message used.
Platform Server Status	read-only (disabled)	Displays the status of last platform log message sent to the syslog server.
Queue Full Percent Platform	read-only to 0%)	If connection is down, percentage of queue of platform message used.
Syslog Server Connection Alarm Enabled	true (default) or false	If set to true, it generates alarms on connection failure (only for TCP and TLS).
Syslog Server Connection Alarm Support	additional properties	Configures additional parameters to generate alarms.
Syslog Message Queue Alarm Enabled	true (default) or false	If set to true, it generates alarms on queue full.
Syslog Message Queue Alarm Support	additional properties	Configures additional parameters to generate alarms.

System Date Time Editor view (platform-SystemDateTimeEditor)

As an available view on a station’s **PlatformServices** container, the **System Date Time Editor** allows you to set the date, time, and time zone for the platform running the station. If the station is running on a Windows platform, this view is read-only.

Figure 74 System Date Time Editor



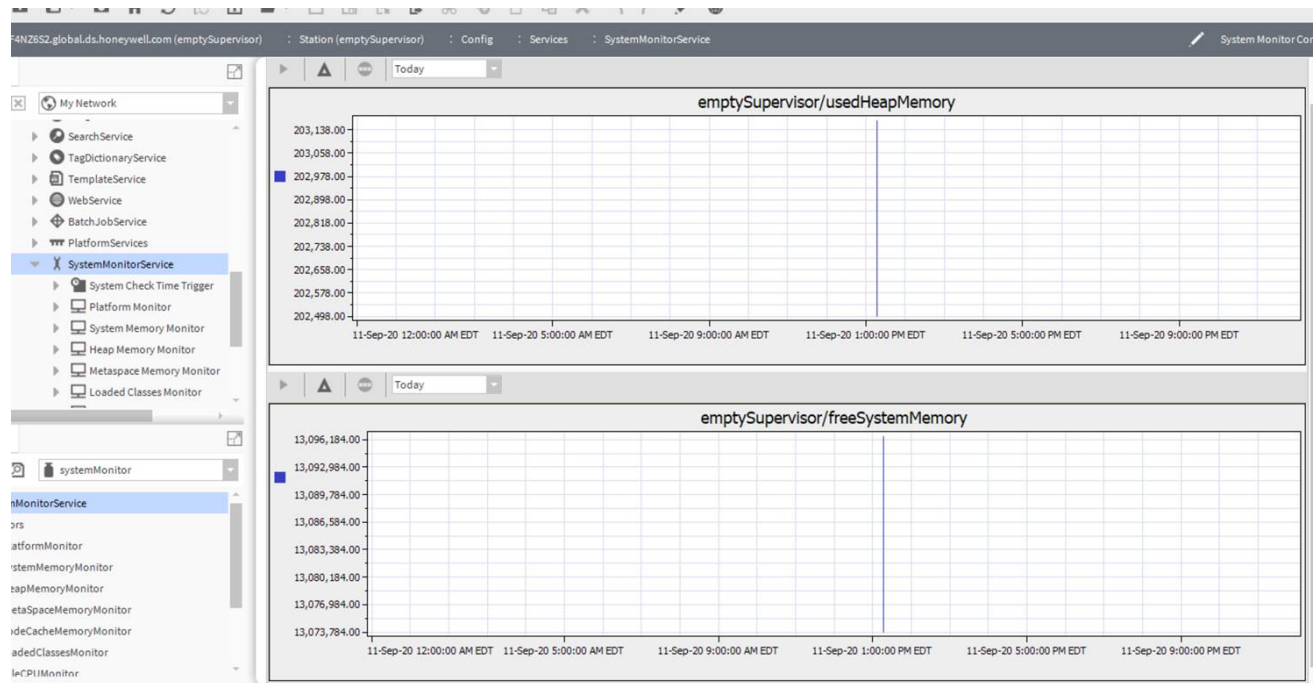
To access this view right click **PlatformServices**→**Views**→**System Date Time Editor**.

Property	Value	Description
System Time	hh:mm	Select the time hh:mm from the drop-down selector.
Date	Date Chooser	Click the date chooser to select the date.
TimeZone	Drop-down list	Select the timezone from the drop-down list for the system

System Monitor Config (systemMonitor-SystemMonitorConfig)

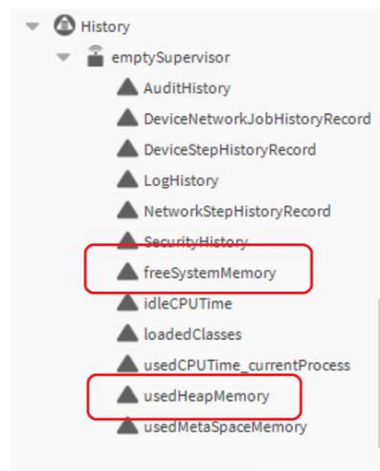
The **Memory History** tab in the **System Monitor Config** view provides a quick method of viewing the trends of Used Heap Memory (taken from the **Heap Memory Monitor**) and the Free System Memory (taken from the **System Memory Monitor**) when the **Log Memory to History** property is set to true.

Figure 75 Memory History tab in System Monitor Config view



Alternatively you could get this same information from the station's History nav tree (shown) adding the histories to a HTML5 web chart view.

Figure 76 Station's History space in the Nav Tree

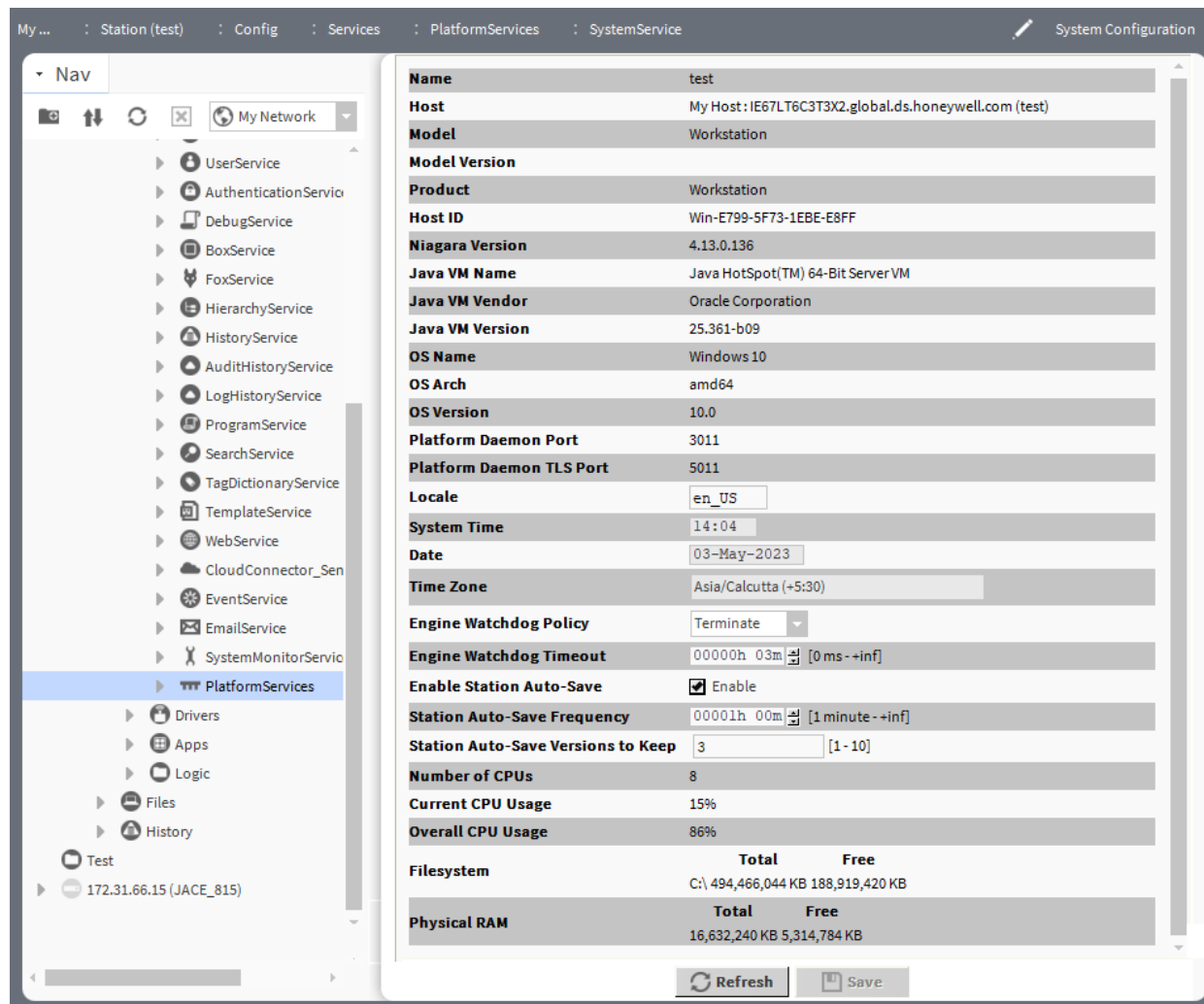


For details about the SystemMonitorService and related components, see "systemMonitor-SystemMonitor-Service" in the "Components" section of this guide.

System Platform Service Plugin (platform-SystemPlatformServicePlugin)

System Platform Service Plugin allows you to view and edit platform properties on a Windows-based host running the station, and is the default view on the station's **System Configuration** (SystemPlatformServiceWin32).

Figure 77 System Platform Service Plugin view



To access, click **Config→Services** and right-click **PlatformServices→Ax Property Sheet** then right-click **SystemService→Views→System Configuration**.

Column Name	Description
Name	Name of running station.
Host	IP address of host platform.
Model	Model of host platform type, such as NPM6, JACE-8000, or Workstation.
Model Version	Reports the version number of the host model.
Product	Defines the product.
Host ID	Niagara host identifier, a string unique to this one machine.
Niagara Version	Version and build number of the Niagara distribution running in the host platform.
Java VM Name	Java virtual machine used, for example, "Java HotSpot(TM) Embedded Client VM" for any N4 controller, or "Java HotSpot(TM) 64-Bit ServerVM" for a Supervisor on a Windows host.
Java VM Vendor	Vendor for Java VM: Oracle Corporation.

Column Name	Description
Java VM Version	Version of Java VM, for example, "25.0-b 70" for the Java 8 compact3 VM on a controller, or "25.31-b07" for the Java 8 SE VM on a Windows host.
OS Name	Operating System name, such as "QNX" or "Windows 10."
OS Arch.	Machine architecture for OS, such as "arm" or "ppc" (controller hosts) or "amd64" (Windows hosts).
OS Version	Operating System version, such as "6.5.0" (QNX) or "10.0" (Windows 10).
Platform Daemon Port	Port number on which the platform daemon that started the station is listening for its platform server (3011, or another port number). This can prove useful in case you changed the platform port, but then forgot what the new port is.
Platform Daemon TLS Port	Port number on which the platform daemon is listening for its platform TLS server (5011, or another port number, provided that platform TLS enabled). If platform TLS is disabled, it reads <code>Unknown</code> . This can prove useful in case you changed the platform TLS port, but then forgot what the new port is. In the container plugin, most of the remaining entries are configuration properties. However a few status values are also mixed in, and are described below.
Locale	Determines locale-specific behavior such as date and time formatting, and also which lexicons are used. A string entered must use the form: language ["_" country ["_" variant]]. For example, U.S. English is "en_US" and traditional Spanish would be "es_ES_Traditional".
System Time	Current local time in host (read-only if a Windows host).
Date	Current local date in host (read-only if a Windows host).
Time Zone	Current local time zone for host (read-only if a Windows host).
Engine Watchdog Policy	The engine watchdog is a platform daemon process, to which the station periodically reports its updated engine cycle count. The watchdog purpose is to detect and deal with a "hung" or "stalled" station, and is automatically enabled when the station starts. The Engine Watchdog Policy defines the response taken by the platform daemon if it detects a station engine watchdog timeout. Watchdog policy selections include: - Log Only — Generates stack dump and logs an error message in the system log. (The station should ultimately be restarted if a watchdog timeout occurs with the "Log Only" setting). - Terminate — (Default) Kills the VM process. If "restart on failure" is enabled for the station (typical), the station is restarted. - Reboot — Automatically reboots the host controller platform. If "auto-start" is enabled for the station, the station is restarted after the system reboots.
Engine Watchdog Timeout	Default is 1 minute, and range is from 0 ms to infinity. If the station's engine cycle count stops changing and/or the station does not report a cycle count to the platform daemon within this defined period, the platform daemon causes the VM to generate a stack dump for diagnostic purposes, then takes the action defined by the Engine Watchdog Policy.
Engine Station Auto-Save	Either Enable (default) or Disable. Allows for "auto save" of running station to "config_backup_<YYMMDD>_<HHMM>.bog" file at the frequency defined in next property. Auto-saved backup files are kept under that station's folder.
Station Auto-Save Frequency	Default is every 24 hours for any JACE platform, or every (1) hour if a Windows host. Range is from 1 to many hours.
Station Auto-Save Version to Keep	Oldest of kept backups is replaced upon next manual save or auto-save backup, once the specified limit is reached. The default value for JACE platform is 0 (none), and should be kept low. However, changing to 1 provides a benefit in the case where a catastrophic (yet inadvertent) station change is made, such that a station "kill" can be issued to revert back to the backup copy on the JACE. In Windows hosts, the default is 3, and typically can be safely adjusted up, if desired.
Number of CPUs	Number of CPUs used in the host platform (typically 1 if a controller, more if a Windows host).
Current CPU Usage	Percentage of CPU utilization in the last second.
Overall CPU Usage	Percentage of CPU utilization since the last reboot.

Column Name	Description																
Filesystem	File storage statistics for the host, including total file space, available (free) space, and file block size (minimum size for even the smallest file). For the JACE-8000 host, it may look similar to: <table><tr><td>Total</td><td>Free</td><td>Files</td><td>Max Files</td></tr><tr><td>/</td><td>3,476,464 KB</td><td>3,039,088 KB</td><td>602</td></tr><tr><td>/mnt/aram0</td><td>393,215 KB</td><td>381,019 KB</td><td>0</td></tr><tr><td>/mnt/ram0</td><td>8,192 KB</td><td>8,192 KB</td><td>0</td></tr></table>	Total	Free	Files	Max Files	/	3,476,464 KB	3,039,088 KB	602	/mnt/aram0	393,215 KB	381,019 KB	0	/mnt/ram0	8,192 KB	8,192 KB	0
Total	Free	Files	Max Files														
/	3,476,464 KB	3,039,088 KB	602														
/mnt/aram0	393,215 KB	381,019 KB	0														
/mnt/ram0	8,192 KB	8,192 KB	0														
Physical RAM	Current total and free RAM statistics for the host. For the JACE-8000, it may look similar to: <table><tr><td>Total</td><td>Free</td></tr><tr><td>1,048,576 KB</td><td>113,424 KB</td></tr></table>	Total	Free	1,048,576 KB	113,424 KB												
Total	Free																
1,048,576 KB	113,424 KB																

System Platform Service Qnx Plugin (platform-SystemPlatformServiceQnxPlugin)

System Platform Service Qnx Plugin allows you to view and edit platform parameters on the JACE platform running the station, and is the default view on the station's **SystemService** (SystemPlatformServiceQnx).

Figure 78 SystemPlatformServiceQnxPlugin view

The screenshot shows the 'System Platform Service Qnx Plugin' view. The left navigation pane lists various services, with 'PlatformServices' expanded. The main area displays configuration parameters for the JACE_815 station. The parameters are organized into sections: Host Information, System Settings, Resource Usage, and System Health. The 'Host Information' section includes fields for Name, Host, Model, Model Version, Product, Host ID, Niagara Version, Java VM Name, Java VM Vendor, and Java VM Version. The 'System Settings' section includes fields for OS Name, OS Arch, OS Version, Platform Daemon Port, Platform Daemon TLS Port, Locale, System Time, Date, Time Zone, Engine Watchdog Policy, Engine Watchdog Timeout, Enable Station Auto-Save, Station Auto-Save Frequency, Station Auto-Save Versions to Keep, Number of CPUs, Current CPU Usage, and Overall CPU Usage. The 'Resource Usage' section includes a table for Filesystem statistics and a table for Physical RAM statistics. The 'System Health' section includes fields for Failure Reboot Limit, Failure Reboot Limit Period, RAM Disk, Java Heap, Open File Descriptors, Free RAM, Disk Space, and Files.

Column Name	Description																
Filesystem	File storage statistics for the host, including total file space, available (free) space, and file block size (minimum size for even the smallest file). For the JACE-8000 host, it may look similar to: <table> <tr> <th>Total</th> <th>Free</th> <th>Files</th> <th>Max Files</th> </tr> <tr> <td>/</td> <td>3,476,464 KB</td> <td>3,039,088 KB</td> <td>602</td> </tr> <tr> <td>/mnt/aram0</td> <td>393,215 KB</td> <td>381,019 KB</td> <td>0</td> </tr> <tr> <td>/mnt/ram0</td> <td>8,192 KB</td> <td>8,192 KB</td> <td>0</td> </tr> </table>	Total	Free	Files	Max Files	/	3,476,464 KB	3,039,088 KB	602	/mnt/aram0	393,215 KB	381,019 KB	0	/mnt/ram0	8,192 KB	8,192 KB	0
Total	Free	Files	Max Files														
/	3,476,464 KB	3,039,088 KB	602														
/mnt/aram0	393,215 KB	381,019 KB	0														
/mnt/ram0	8,192 KB	8,192 KB	0														
Physical RAM	Current total and free RAM statistics for the host. For the JACE-8000, it may look similar to: <table> <tr> <th>Total</th> <th>Free</th> </tr> <tr> <td>1,048,576 KB</td> <td>113,424 KB</td> </tr> </table>	Total	Free	1,048,576 KB	113,424 KB												
Total	Free																
1,048,576 KB	113,424 KB																

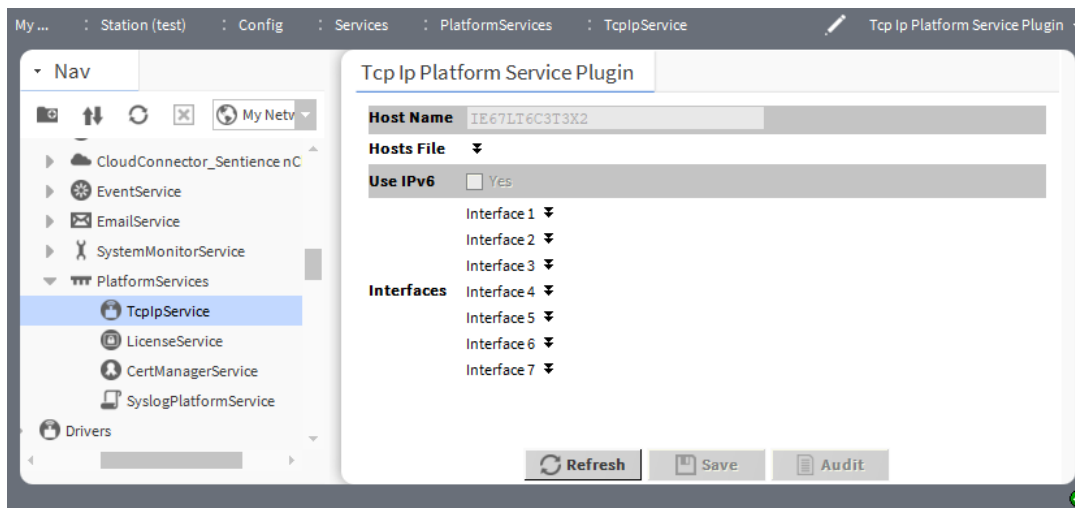
To access this view expand controllers **Config→Services** and right click **Platform Services→AX Property Sheet** and select **System Service**.

Tcp Ip Platform Service Plugin (platform-TcpIpPlatformService)

This plugin provides station access to the host platform's TCP/IP settings. This service is found under the running station's **PlatformServiceContainer**. From the default plugin (view), you can perform the same operation as from the **TCP/IP Configuration** view using a platform connection.

This platform service supports installations where all configuration is done using only a browser connection and not a platform connection from Workbench to a remote controller's platform daemon.

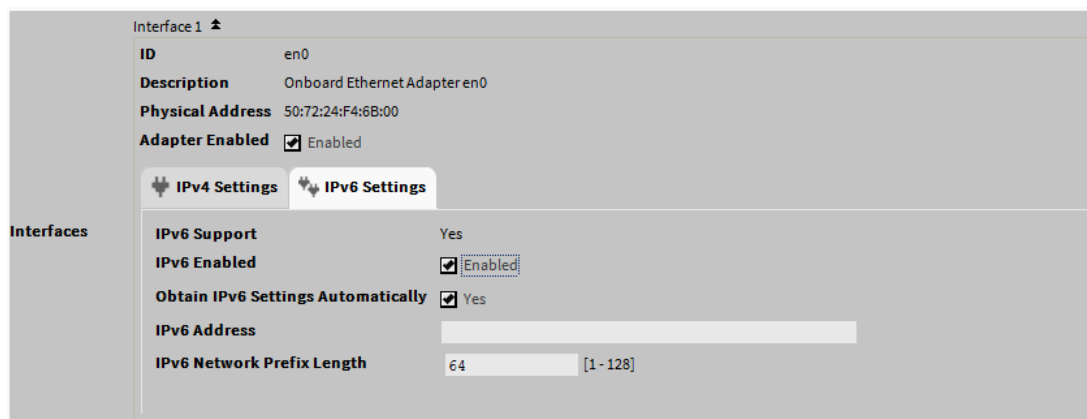
Figure 79 Tcp Ip Platform Service Plugin



To access, expand **Config→Services→PlatformServices** and double-click **TcpIpService**.

If the station is running on a Windows platform or if a Win32 host and the platform authentication setting labeled **Stations - allow stations to have admin access to platform daemon** is disabled, TCP/IP properties in this view are read-only.

Figure 80 IPv6 tab for Interface of the JACE-8000 controller



To access these properties, click **Platform→TCP/IP Configuration**, expand any **Interface** and select **IPv6 Settings**.

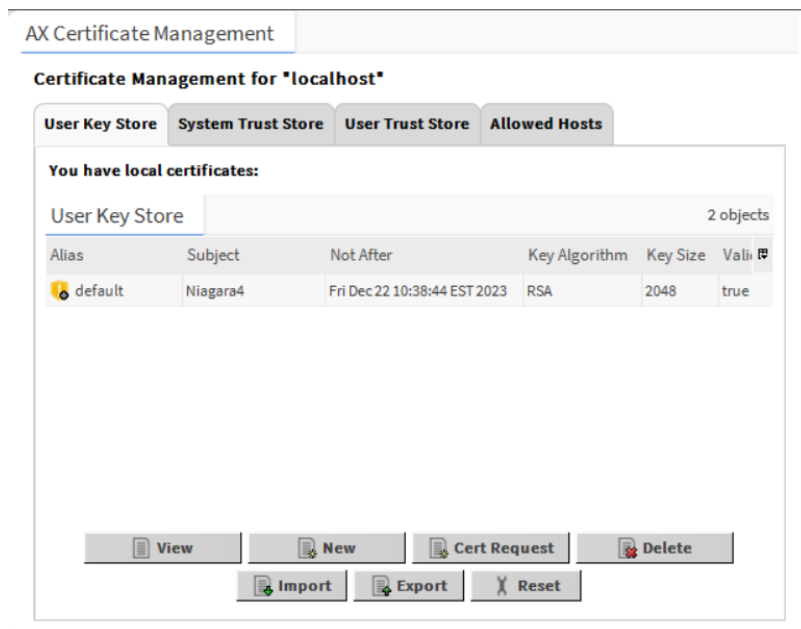
Property	Value	Description
IPv6 Support	read only	Indicates if host platform's OS supports IPv6.
IPv6 Enabled	check box for Enabled where default is cleared (disabled)	If a Windows host, this indicates if it is configured with the IPv6 protocol.
Obtain IPv6 Settings Automatically	Checkbox for Enabled (default)	Provides for "auto-configuration" of the IPv6 address, if acceptable. If enabled on the JACE-8000 controller, the next two properties are read-only. If cleared, the two properties below must be entered manually.
IPv6 Address	alphanumeric	The host's IP address in IPv6 format, to be unique on its network.
IPv6 Network Prefix Length	number	The number of left-most contiguous bits of the IPv6 address (in decimal) that compose the subnet prefix.
DNSv6 Servers	Read-only	IPv6 address for one or more DNS servers, each of which can automate associations between hostnames and IPv6 addresses. (Windows hosts only, providing host's OS has IPv6 enabled).

Workbench Certificate Management (platCrypto-CertManagerTool)

This view accesses the Workbench key stores. You use it to create digital certificates and Certificate Signing Requests (CSRs), and to import and export keys and certificates to and from the Workbench stores.

You use this view to manage PKI (Public Key Infrastructure) and self-signed digital certificates to secure communication within the **NiagaraNetwork**. Certificates secure TLS connections to this host.

Figure 81 Certificate Management view



To access this view, click **Tools→Certificate Management**. It defaults to the User Key Store

This view has four tabs:

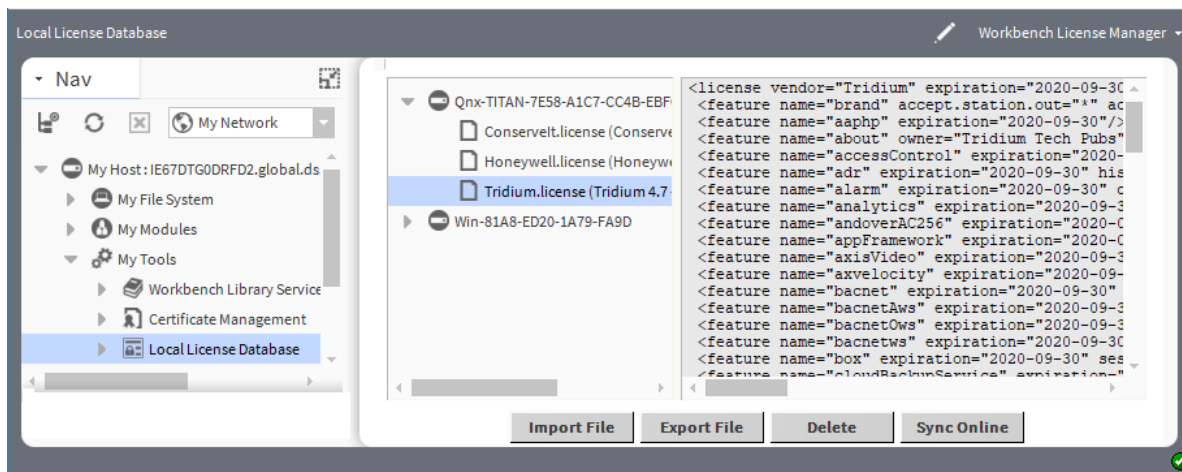
- **User Key Store** contains the root, server, client and intermediate certificates you create.
- **System Trust Store** contains the trusted, third-party, client certificates that commonly secure Internet servers.
- **User Trust Store** contains the trusted client certificates your company created to serve as its own Certificate Authority.
- **Allowed Hosts** contains approved self-signed certificates. These are certificates that you or someone else in your company knows to be secure certificates that can be used to encrypt data. These certificates cannot be used to authenticate a server because no root CA certificate in the **System Trust Store** or **User Trust Store** has signed them.

A separate topic documents each tab.

Workbench License Manager view (platform-WorkbenchLicenseManager)

This view manages the contents of your Workbench PC's local license database.

Figure 82 WorkbenchLicenseManager view



To access this view navigate to **Tools** and select **Local License Database**.

Buttons

- **Import File** imports the license file from the local drive.
- **Export File** exports the selected license file.
- **Delete** deletes the selected license file.
- **Sync Online** synchronize all licenses in database with the licensing server.

Chapter 18 Windows

Topics covered in this chapter

- ◆ Advanced Platform Options
- ◆ Change TLS Settings window
- ◆ Configure NRE Memory Pools window
- ◆ Daemon Output Settings window
- ◆ Edit Filter window
- ◆ Export windows
- ◆ Import License window
- ◆ User Accounts
- ◆ New User window
- ◆ Set System Date/Time window
- ◆ Software Details window
- ◆ Sync Now window
- ◆ Syslog Configuration window
- ◆ View Details window
- ◆ Software Details window

Windows create and edit database records or collect information when accessing a component. You access them by dragging a component from a palette into a station or by clicking a button.

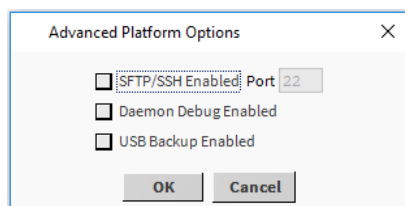
Windows do not support **On View (F1)** and **Guide on Target** help. To learn about the information each contains, search the help system for key words.

Advanced Platform Options

This **Platform Administration** window opens for controller platforms only, to enable, disable, or configure certain settings and properties.

For Windows-based hosts, you can use Windows "Remote Desktop Connections" for SFTP/SSH.

Figure 83 Advanced Platform Options



NOTE: This window replaces an **FTP/Telnet** selection available for QNX-based platforms running Niagara 4, which are both inherently less secure services.

This window contains these properties:

Property	Value	Description
SFTP/SSH Enabled Port	check box (defaults disabled: to no check mark)	Enables, disables, and configures SFTP (Secure File Transfer Protocol) or SSH (Secure Shell Protocol) access. For Windows-based hosts, you typically use Windows “Remote Desktop Connections” instead. As factory-shipped, a Niagara 4 controller’s SFTP and SSH services default to disabled. This may be best, especially if the platform is exposed to the public Internet. However, in some cases, you may wish to temporarily enable the single port shared by these services, perhaps to facilitate debugging. CAUTION: Even SFTP and SSH pose security risks. Before enabling, we strongly recommend you configure the platform TLS only, and keep this function disabled unless otherwise directed by Systems Engineering. After logging in with platform credentials, SSH access to a controller provides system shell access with the same menu as provided by serial shell access to its RS-232 port. You can change the TCP/IP port shared by these services from the well-known port to some other port. However, be sure that any firewalls used on your network allow traffic to the alternate port.
Daemon Debug Enabled	check box (defaults disabled: to no check mark)	Enables a QNX webserver to accept incoming browser connections on :3011 and :5011. Connecting to port 5011 using a browser (<a href="https://<ipAddress>:5011">https://<ipAddress>:5011) provides a list of links to additional information about the QNX controller.
USB Backup Enabled	check box (defaults disabled: to no check mark)	Enables and disables USB backup for platforms that have USB backup capability.

Change TLS Settings window

This window provides access to the primary TLS settings.

Figure 84 Platform TLS Settings with default values (enabled)

The screenshot shows the 'Platform TLS Settings' dialog box. The settings are as follows:

- State:** TLS Only
- Port:** 5011
- Certificate Alias:** default (dropdown menu)
- Certificate Password:** (unchanged) (text field)
- Use global certificate password:** ☐ (checkbox)
- Protocol:** TLSv1.2+ (dropdown menu)
- Use Extended Master Secret:** true (dropdown menu)
- TLS Cipher Suite Group:** Recommended (dropdown menu)

At the bottom are 'Save' and 'Cancel' buttons.

To access, expand **Platform**→**Platform Administration** and click **Change TLS Settings**.

Properties	Value	Description
State	Tls Only	Specifies how Workbench clients connect to this host's platform daemon. Tls Only — Only secure platform connections are allowed. Any attempt to connect without security goes unresolved (errors out). This state is reflected among the properties listed on the main Platform Administration view, as "Platform TLS Support" state. NOTE: The <code>Tls Only</code> option provides the best security. All platforms support secure (TLS) platform connections.
Port	four-digit number (default is 5011)	Identifies the software port monitored by the platform daemon for a secure platform connection. This is different than the default HTTP port (3011) for a regular platform connection that is not secure. CAUTION: If there is a firewall on the host (or its network), before changing this port make sure that the firewall will allow traffic to the new port.
Certificate Alias	text (defaults to the default self-signed certificate for Niagara 4.13 and later; defaults to <code>tridium</code> self-signed certificate for pre-Niagara 4.13 versions)	The alias for the server certificate in the platform's key store to use for any platformtls connection. The default is automatically created when Niagara is first loaded. NOTE: If the <code>tridium</code> certificate is already used on the station or the platform runs a pre-Niagara 4.13 version, the <code>tridium</code> certificate is used, but it will not serve as a recovery certificate. If another certificate has been imported in the platform's key store, use the drop-down control to select it instead. Certificates on the platform are managed via the platform Certificate Management view. For general information on this topic, see <i>Niagara Station Security Guide</i>.
Certificate Password	text and check box	As of Niagara 4.13, the certificate is password-protected by a unique password or the global certificate password. Prompts the user to provide the user-defined password or the global certificate password associated with the certificate.
Protocol	TLSv1.0+ — Includes TLS versions 1.0, 1.1, and 1.2, providing the most flexibility; TLSv1.1+ — Only TLS versions 1.1 or 1.2 are accepted; TLSv1.2+ — (default) Only TLS versions 1.2 or 1.3 are accepted; TLSv1.3 — Only TLS version 1.3 is accepted.	Defines the minimum TLS (Transport Layer Security) protocol version that the platform daemon's secure server accepts to negotiate with a client for a secure platform connection. During the handshake, the server and client agree on which protocol to use.

Properties	Value	Description
Use Extended Master Secret	true (default) or false	Turns on and off the “Extended Master Secret” on a server. When turned off (set to false) and the platform restarts, the CPU usage does not change significantly when connecting to the Platform Administration view from a FIPS-mode Workbench.
TLS Cipher Suite Group	drop-down list, recommended (default) or supported	Controls which cipher suites can be used during TLS negotiation. The default is more secure than the other option (Supported) and should be used unless it causes compatibility issues with the client.

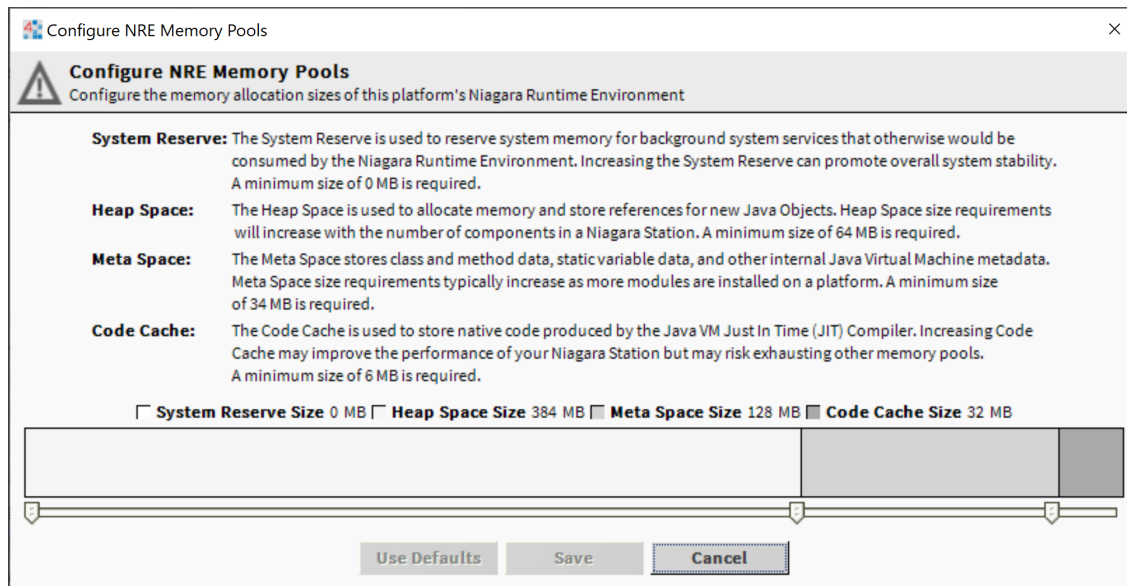
Configure NRE Memory Pools window

This window provides a mechanism to configure NRE (Node Runtime Environment) memory.

A fixed memory footprint serves all the memory pools. Adjusting one pool affects at least one other pool.

CAUTION: Configuring a controller with insufficient memory allocations could prevent the station from starting or could cause the station to fail and restart.

Figure 85 Configure NRE Memory Pools window



To access **Configure NRE Memory Pools** window, open the **Platform Administration** view and click **Configure NRE Memory**. The descriptions shown in the window include recommended allocation sizes. If you adjust the space allocation for any of the memories described above, you must reboot the controller for the new configuration to become effective.

Table 3 NRE memory default sizes in MB

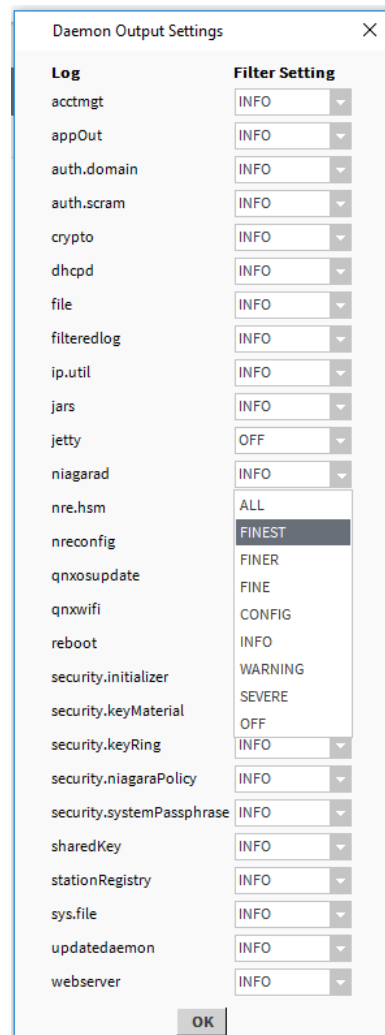
Type of memory	MB	Description
System Reserve Size	0	Allocates additional free operating system space. By default, this is set to 0MB. The reason you might reserve additional free operating system space is if additional system RAM is needed when a new thread is spawned by the station or niagarad (daemon) for native stack and overhead.
Heap Space Size	384	Configures where the station runs. As your program grows it requires more heap memory.
Meta Space Size	128	Holds all of the Java classes that are loaded from the modules. As you install more drivers, this increases the number of .jar files holding the classes.

Type of memory	MB	Description
Code Cache Size	32	Holds code that has already been compiled. Java functions with a JIT (Just-In-Time) compiler. As Java executes classes, it compiles code on the fly. Saving code that it has already compiled to the Code Cache eliminates the need to compile it again. Code that is used most often is cached. However, since there is a limited amount of memory available to cache code, it continues using the compiler as the station runs. If you adjust the memory allocation to allow a significant amount of space for code caching the station runs faster, but the risk is in taking away too much memory from the other memory spaces resulting in the station being unable to run at all.
Free Memory	~352	

Daemon Output Settings window

This window adjusts (tunes) the amount and content (output) provided by the **Application Director's** output and platform daemon output.

Figure 86 Daemon Output Settings window for the controller



To access this window, expand **Platform**, double-click **Platform Administration** and click **Change output Settings**.

Logs

By default, all daemon processes have a message log filter level. The logs include the following:

- **niagarad** logs information for the platform daemon (niagarad) process, with high level entries like `niagarad starting, baja home = ..., niagarad stopping`.
- **appOut** logs information for the thread that manages the buffers associated with station output, making that output visible in the **Application Director** view. Entries may reflect buffer size changes (available in the **Application Director** interface), or if a problem occurs streaming the output to Workbench.
- **file** logs requests made to the platform daemon's file servlet. Platform views, such as the **File Transfer Client**, **Commissioning Wizard**, **Software Manager**, and **Station Copier** generate these requests. Many different things can print on this log, such as `request for file xxx`, and `wrote file xxx`. where `xxx` is a file name.
- **qnxosupdate** logs information for the OS upgrade servlet created by the platform daemon. Workbench uses this servlet to upgrade the QNX OS in the host controller when using the **Commissioning Wizard** or **Distribution File Installer**. Output can reflect a problem when updating the QNX OS, such as `os crc isn't right`, and `waitpid when launching osupdate command failed`.
- **reboot** logs information for the reboot servlet, which is one of the servlets the platform daemon manages.
- **stationregistry** logs information for platform daemon management of stations, including startup, shutdown, and watchdog actions.
- **updatedaemon** logs information for handling Workbench requests for current platform daemon configuration. This process is used mainly by the **Platform Administration** view.
- **webserver** logs information for the HTTP server that manages incoming platform client connections. Entries are often generic, before the daemon hands off to the appropriate platform servlet.

Filter Setting

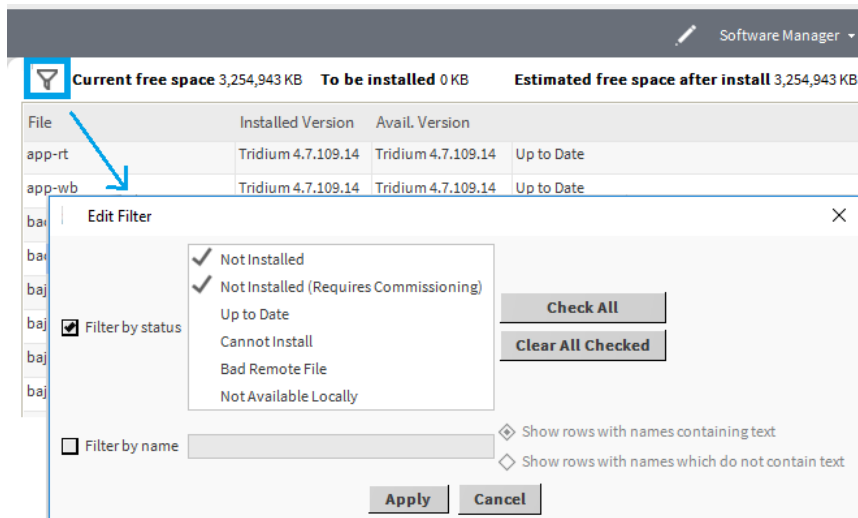
For any process, use the **Filter Setting** drop-down list to select one of the following:

- **FINE** returns all message activity (verbose). This includes all transactional messages, which may result in too many messages to be useful. Levels, such as **ALL**, **FINER**, **FINEST** are equivalent to the **FINE** level in the station log output.
- **CONFIG** returns configuration information.
- **INFO** returns informational **MESSAGES**, plus all **ERROR** and **WARNING** types.
- **WARNING** returns only **ERROR** and **WARNING** type messages (no informational **MESSAGES**).
- **SEVERE** returns only **ERROR** type messages (no **WARNING** or informational **MESSAGES**).
- **OFF** displays an off message.

Edit Filter window

This window selects items for listing, thereby filtering undesired items out of the **Software Manager** view.

Figure 87 Edit Filter window



To open this window, connect to the remote platform, expand **Platform**, double-click **Software Manager** and click the filter icon () in the upper left corner of the view.

Type of filter	Description
Filter by status	Modules with an Out of Date or Out of Date (Requires Commissioning) status always appear in the Software Manager. So do any with uncommitted (intended) status values, such as Install, Uninstall, and so on. When you enable the filter by status, you can check other statuses to include or clear to omit from the list of associated modules in the table, as follows: • Not Installed includes modules on your PC that can be installed, but are not in the remote platform. • Not Installed (Requires Commissioning) includes modules on your PC that are not on the remote platform. The remote controller must be upgraded using Commissioning Wizard first. • Up to Date includes modules on your PC and in the remote platform, where the software is not older. • Cannot Install includes local modules that are unreadable or have a bad manifest. The Software Manager cannot install these modules. • Bad File includes remote modules that are unreadable or that have a bad manifest.
Filter by name	Name filtering lets you include or exclude items based on a character string portion of module's file name. When enabled (checked), you can type in a string of characters, and then check one of the following: • Show rows with names containing text includes only items with a file name that contains the string. • Show rows with names which do not contain text includes only items with file names that do not contain the string. This feature can be useful to filter many modules with common name characters, for example "lon" or "doc" part-named modules.

Buttons

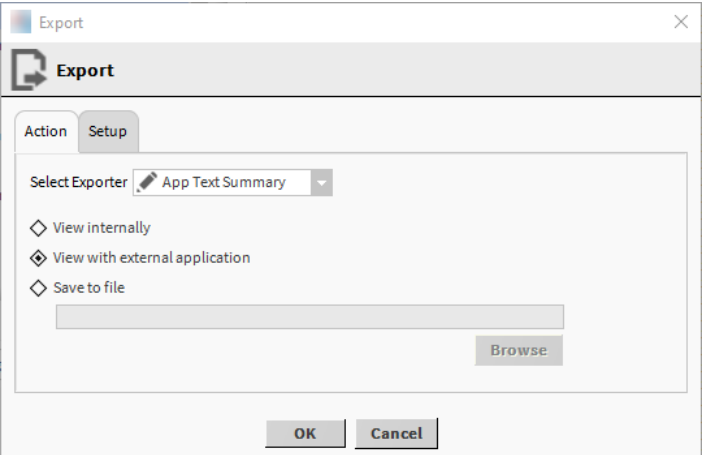
With status filtering enabled, you can also **Check All** and **Clear All Checked**.

- If all status items are cleared, only "Out of Date" and uncommitted status modules appear.
- If all status items are checked, the display is similar to disabled status filtering, except "non-module" items are not listed.

Export windows

These windows include, limit and exclude the platform summary data, platform daemon console output, and station console output to export from the daemon and station. The exported data may be used for trouble-shooting purposes.

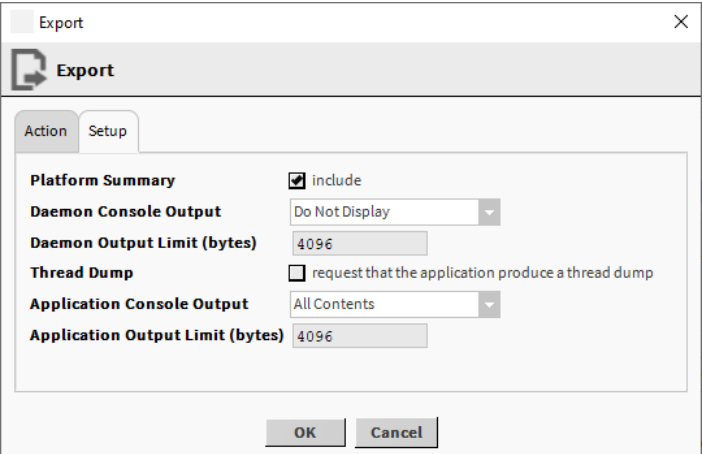
Figure 88 Export window Action tab



To access these windows, expand **Platform**→**Application Director** and click the **Export** tool bar icon (📄).

Property	Value	Description
Select Exporter	drop-down list (defaults to App Text Summary)	Selects how to configure the output for export. App Text Summary Object to Obix
radio button options	default to View with external application	Chooses how to view the exported data. View internally changes the daemon output to oBix XML. View with external application opens the output in an external application, such as Notepad. Save to file enables the Browse button with which to select an output location.

Figure 89 Export window Setup tab



To access this window, select **Setup** tab.

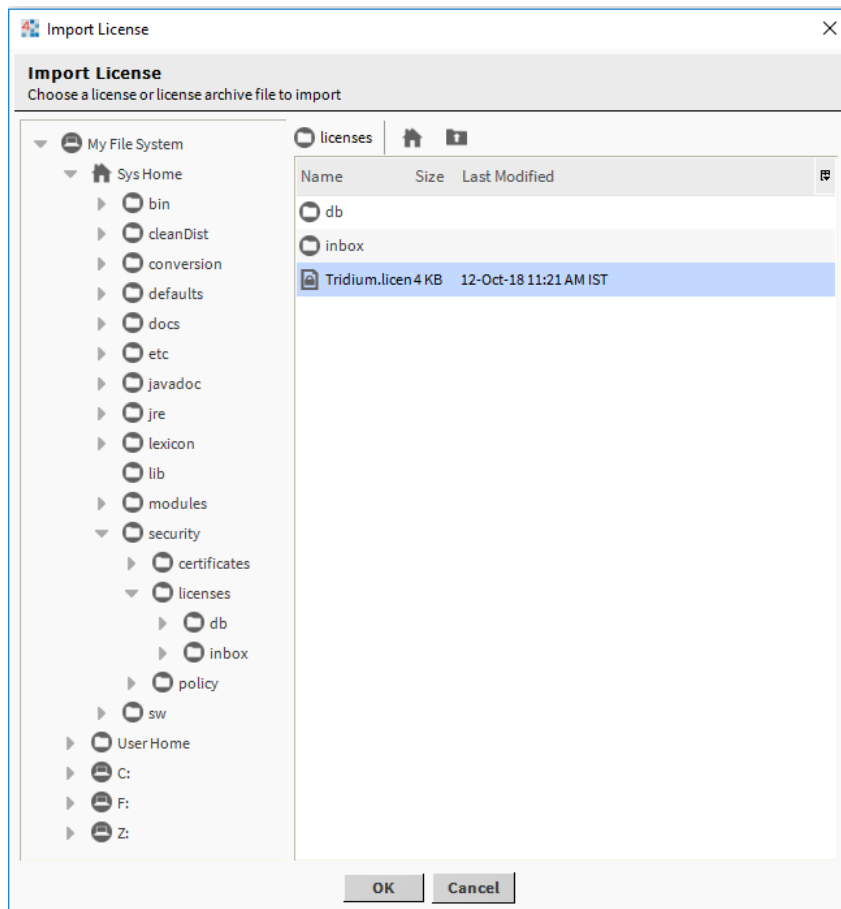
Property	Value	Description
Platform Summary	check box (defaults to enabled)	Determines if platform summary information should also be exported.
Daemon Console Output	drop-down list (defaults to Do Not Display)	Determines how much, if any, daemon console output to include in the export. Do Not Display excludes console output. All Contents includes all console output. Last N Bytes enables Daemon Output Limit (bytes) so you can select how many to export. From Last Thread Dump continues the output from the last export.
Datmon Output Limit (bytes)	read-only unless Daemon Console Output is set to Last N Bytes	Configures how many bytes of data to export.
Thread Dump	check box (defaults to disabled)	Enables a thread dump.
Application Console Output	drop-down list (defaults to All Contents)	Determines how much, if any, application console output to include in the export. Do Not Display excludes console output. All Contents includes all console output. Last N Bytes enables Application Output Limit (bytes) so you can select how many to export. From Last Thread Dump continues the output from the last export.

Import License window

The **Import File** button in the **Workbench License Manager** is always enabled, and opens the **Import License** window for you to navigate to a source file (.license or .lar).

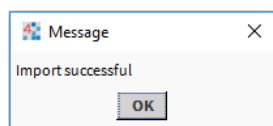
Only two types of files appear for selection.

Figure 90 Import License dialog to find local license file or license archive file



To add to (or update in) your local license database, select a license file and click **OK**. A popup window confirms success, and the license(s) are added or updated in your database.

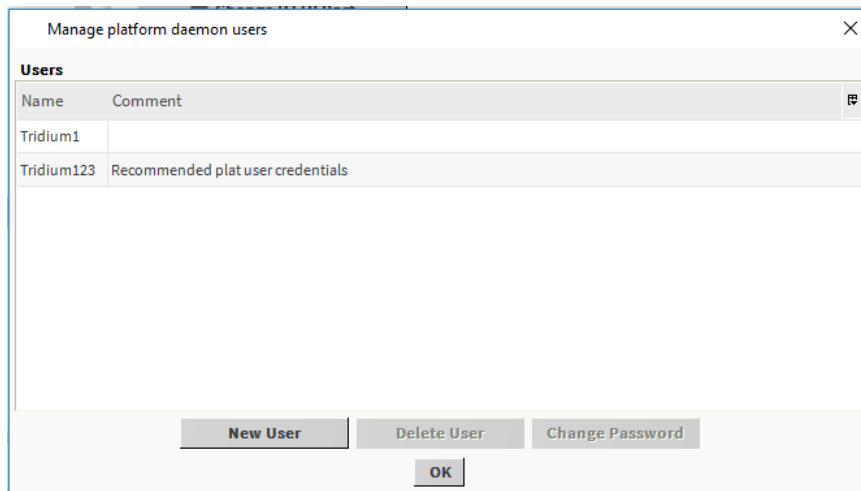
Figure 91 Import success



If any of the license(s) you select to import are older than the ones currently in your local database, meaning that the generated attribute timestamp is earlier, newer license(s) in your local license database are not overwritten. However, the same Import successful message popup appears for such file import operations.

User Accounts

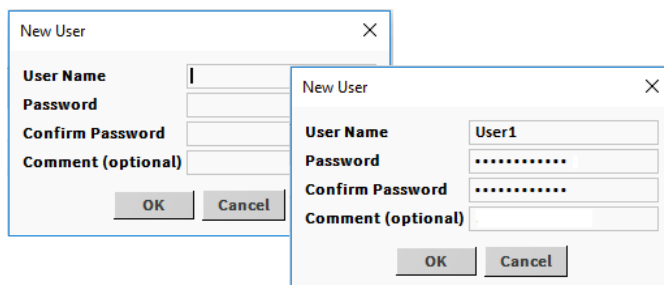
This selection from the main **Platform Administration** view is available on Niagara 4 controllers only. You can create multiple platform administrator users (up to 20 maximum). All have the same full administrator permissions, can create additional users, and can change the password of their own account.

Figure 92 Example where two platform admin accounts have been created

If you are commissioning a new unit, or a controller that has had a cleanDist file installed, only a well-known default platform admin account exists. Any unit with the default platform admin user is extremely susceptible to unauthorized intrusion. Therefore, before you can complete other commissioning tasks, the **N4 Commissioning Wizard** requires you to first replace the default platform user account in a wizard step.

New User window

Clicking **New User** opens the **New User** window.

Figure 93 Example New User dialog after typing user name, password, and comment

Follow these basic guidelines to create strong passwords:

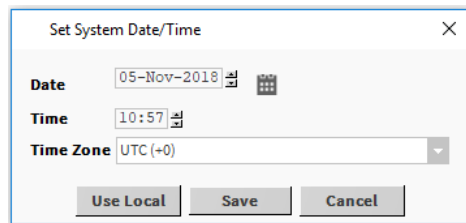
- Use both upper and lower case.
- Include numeric digits.
- Include special characters.
- Do not use dictionary words.
- Do not use your company name.
- Do not make your password the same as your user name.
- Do not use common numbers like telephone numbers, addresses, your birthday, and so on.

Property	Value	Description
User Name	text	A maximum of 14 alphanumeric characters (a - z, A - Z, 0 - 9), where the first character must be alphabetic, and following characters either alphanumeric or underscore (_).
Password	a minimum of 10 characters using: at least one UPPER CASE letter, at least one lower case letter, and at least one digit (numeral)	Creates and verifies a strong password using two fields. The password must <i>match</i> in both password fields. The characters you enter display as asterisks (*). An error popup reminds you if you attempt to enter a password that does not meet minimum rules.
Comment	optional text: at most 64 alphanumeric characters, with these also allowed: - = + () @ · _	NOTE: At the time of this document, comment text cannot be re-edited after adding it to a user account.

Set System Date/Time window

This window configures the remote platform's date and time.

Figure 94 Set System Date/Time window



To access, expand **My Platform**→**Platform Administration** and click **Change Date/Time**.

Properties

Property	Value	Description
Date	three fields	Defines a day, month and year using the up and down control buttons.
Time	24-hr. time	Always displays in 24-hour format.
Time Zone	drop-down list	Each time zone provides a text description, and in parenthesis the hour offset from UTC (and if daylight savings time is used) the offset plus daylight savings. For example: America/New_York (-5, -4).

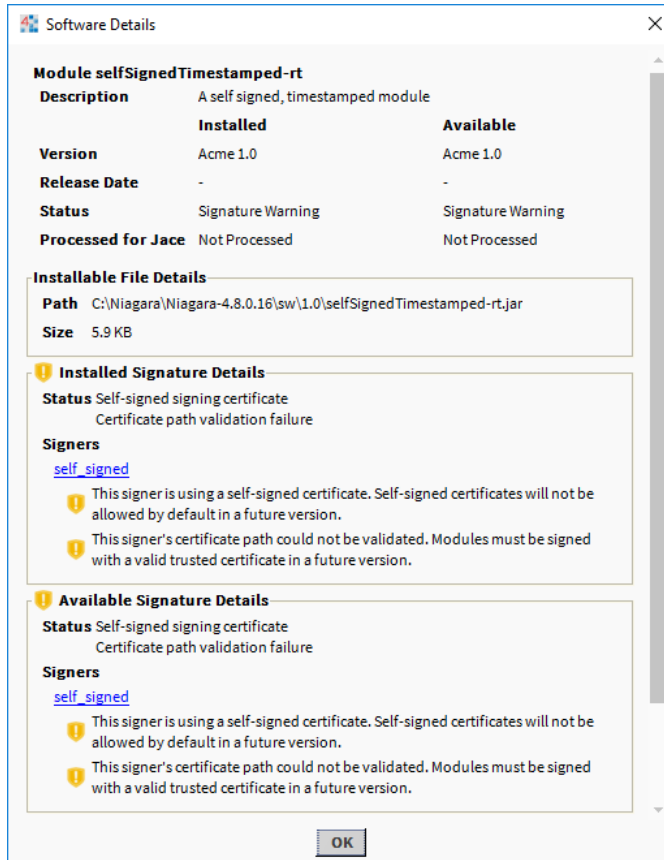
Buttons

- **Use local** synchronizes the remote host's date, time, and time zone with your Workbench PC.
- **Save** updates the date time and time zone in the platform and closes the window. This button becomes available after you change one or more properties in the window or when you click **Use Local**.
- **Cancel** abandons the update.

Software Details window

This window displays information about each software module that is available locally and installed on a remote controller platform.

Figure 95 Software Details dialog from Software Manager



To open this window, connect to the remote platform, expand **Platform**, double-click **Software Manager**, locate a module and double-click its row.

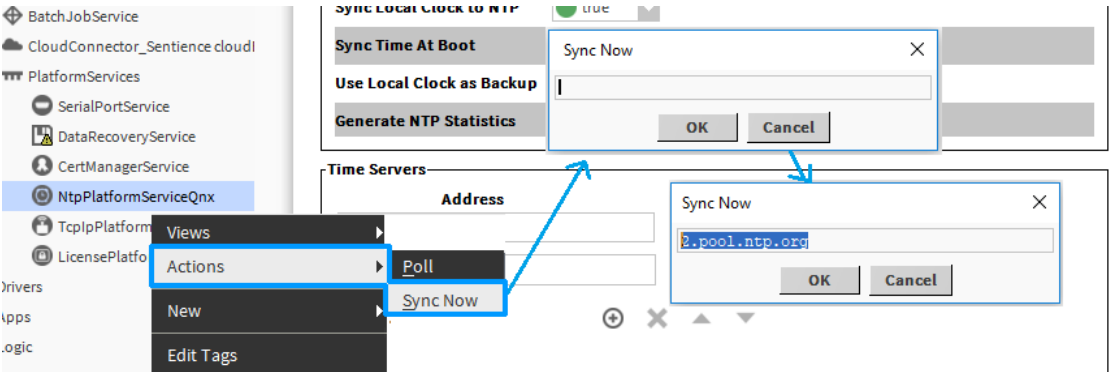
Details include a brief module description, comparisons between installed and available modules, module file and size, signature statuses, and whatever module dependencies exist, by part names. Dependencies are listed for both cases: what software is required by this module, plus software that is dependent on this module. The dependency details are for information only. When installing modules from the **Software Manager**, all dependent modules are automatically included when you select a module to install.

As well as displaying the signature statuses of the selected module, this window provides a link to view the certificate that the module is signed with. For more details on signature statuses, refer to the *Niagara Third Party Module Signing* guide.

Sync Now window

This window attempts to connect to a remote NTP server over the Internet. The purpose of this connection is to confirm that the controller can reach the NTP server to synchronize date and time.

Figure 96 Sync Now window

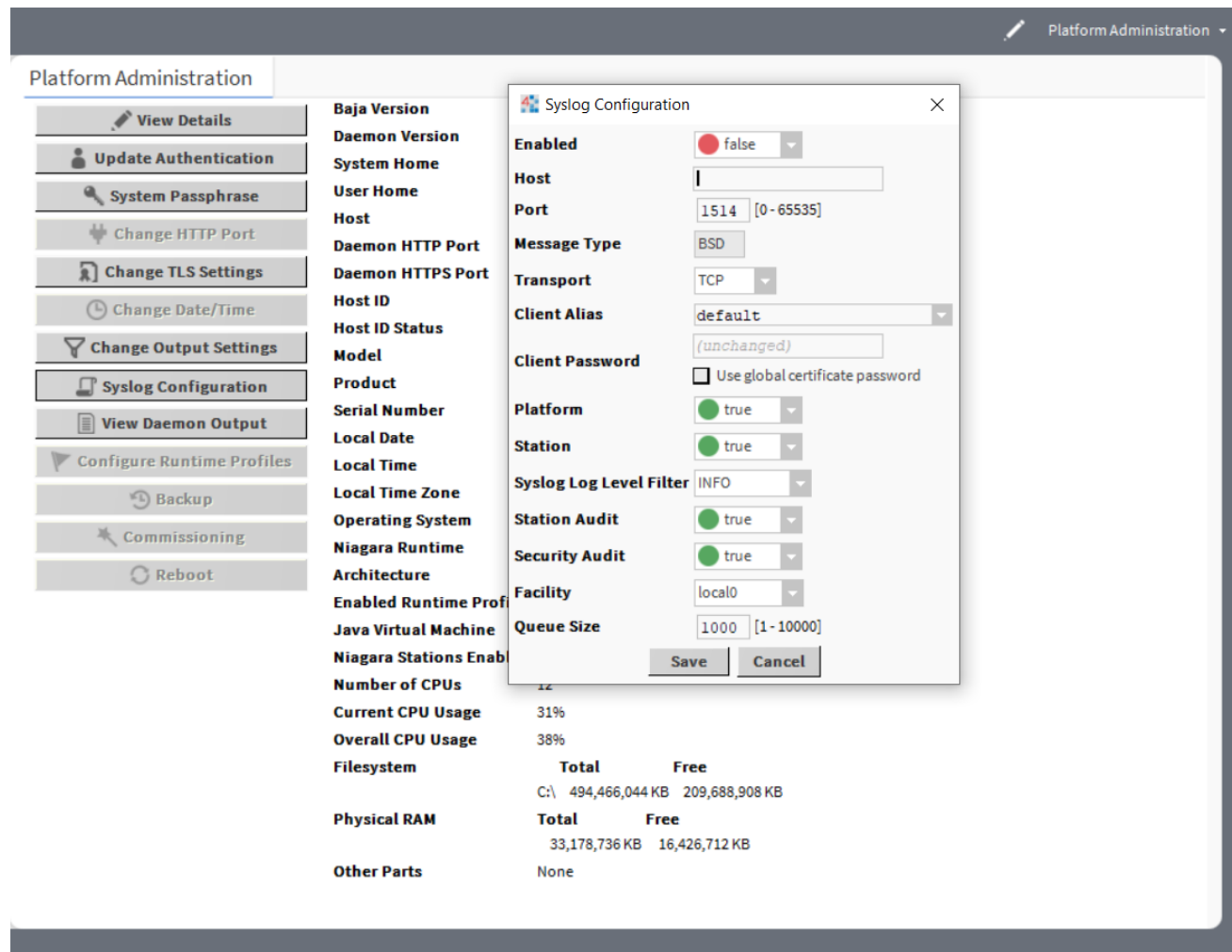


This window has only one property for the fully-qualified domain name of a public NTP server or the IP address of any accessible NTP server.

Syslog Configuration window

As of Niagara 4.13, you can access the **Syslog Configuration** window in the **Platform Administration** view. Syslog is a standard protocol for message logging, which allows messages generated by to be stored and analyzed on a remote server.

Figure 97 Syslog Configuration window



To access the **Syslog Configuration** window, open the **Platform Administration** view and click **Syslog Configuration**.

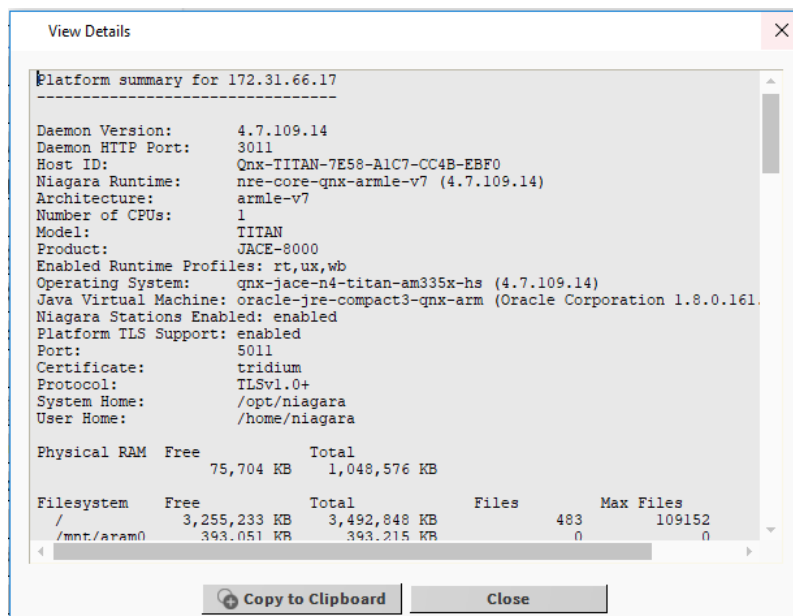
Type	Value	Description
Enabled	false (default) or true	Disables (false) or enables (true) the system log service.
Host	IP address	Specifies the IP address or hostname of the Syslog Server.
Port	number (defaults to 1514)	Specifies the port for communication.
Message Type	read-only (defaults to BSD)	Specifies the type of message supported. Currently only the BSD type is supported.
Transport	drop-down list	Specifies the transport protocol used for communicating messages to the server.
Client Password	text	This is only required if the syslog server requires mutual TLS (mTLS) protocol. This property defines the client certificate in the User Key Store to use. Refer in Niagara Station Security

Type	Value	Description
		Guide to “Creating a Client Certificate for Syslog configuration” for more information on generating Client Certificates.
Platform	true (default) or false	Enables (true) or disables (false) the platform logs sent to the server.
Station	true (default) or false	Enables (true) or disables (false) the station logs sent to the server.
Syslog Log Level Filter	Off, Severe, Warning, Info, Config, Fine, Finer, Finest, All (defaults to Info)	Sets the minimum level of platform and station logs that will be sent to the syslog server.
Station Audit	true (default) or false	Enables (true) or disables (false) the station audit records sent to the server.
Security Audit	true (default) or false	Enables (true) or disables (false) the security audit records sent to the server.
Facility	drop-down list (defaults to local0)	Specifies the facility (or process) which generated the syslog messages.
Queue Size	number (defaults to 1000)	Specifies the queue size to hold the messages until they are sent.

View Details window

This window summarizes all information related to the platform.

Figure 98 View Details window



Included in the **View Details** window is a listing of all installed modules, lexicons, licenses, and certificates. Included is a station line, listing configuration for autostart and autorestart, plus current status.

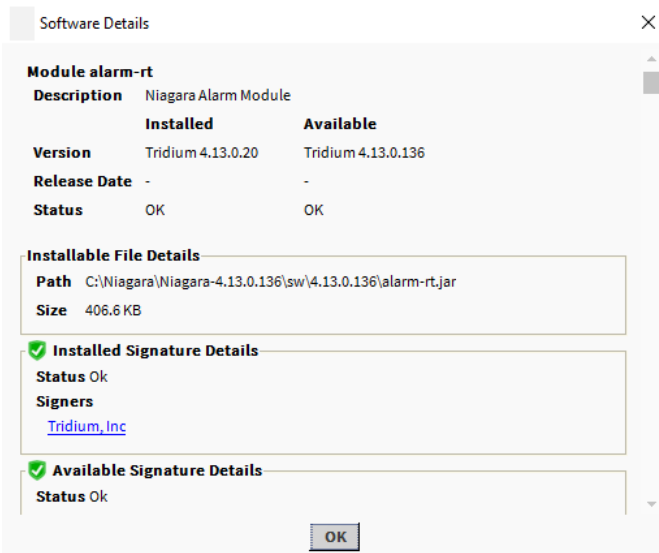
Buttons include:

- **Copy to Clipboard** puts all details in the window on your PC's Windows clipboard.
- **Close** exits the window, same as Windows close control (contents copied remain on clipboard).

Software Details window

Software Details is the window that appears when you double-click an item (for example, module) listed in the platform's **Software Manager** view. A number of details is provided about the selected item.

Figure 99 Software Details view



Provide the details for the selected module.

Glossary

engineering workstation	An installation of Niagara on a PC, which is used to commission controller hardware and perform application engineering on both offline and online stations. In some cases the this workstation may also be licensed to run a station to facilitate application development and testing.
NTP	Network Time Protocol. This protocol is used by computers and devices to synchronize clocks over the Internet.
UTC	Coordinated Universal Time (UTC) is the recognized atomic-clock standard of reference time, largely replacing GMT (Greenwich Mean Time) as the time to reference. Time zones are commonly expressed as negative or positive offsets from UTC time.

Index

64-bit serial operations 13

A

aaphp 59
 aapup 59
 Advanced Platform Options window 243
 Allowed Hosts tab 217
 Application Director 20, 31, 34
 applications 60

B

backup
 using station copier 126
 backup battery 13
 backups
 system passphrase 87
 bacnet 59
 bacnetAws 59
 BacnetMstpPlatformService 152
 bacnetOws 59
 BOG file
 adding passphrase protection 89
 boot
 remote controller 108
 box application 61

C

category mask 95
 certificate
 management 39
 wizard 39
 certificate management 213
 Certificate Management 20
 Certificate Management plugin 240
 certificate wizard
 generate CA Root certificate 40
 certificates
 verify imported 44
 CertManagerService 199, 213
 Change TLS Settings window 244
 co-processor 152
 Code Cache Memory Monitor 191
 components
 available in both a Supervisor and remote
 controller platform 173
 available in only a remote controller
 platform 161
 available in only a Supervisor platform 159
 controller
 commissioning 103

installing its license 65
 returning to factory defaults 51
 upgrading 107
 controller models 11
 CPU Idle Cycles Monitor 188
 CPU usage 95
 CPU Used Cycles Monitor 189

D

daemon
 availability 180
 export windows 250
 Export windows 250
 station
 export windows 250
 user home 25
 User Home 23
 daemon (niagarad) 14
 daemon output 97
 Daemon Output Settings window 247
 Daemon Session 177
 daemon user home
 copying a station to 138
 daemon User Home 26
 DaemonSecureSession 173
 Data Recovery Service 161
 Data Recovery Service Editor 203
 database 111
 date 92
 setting 94
 daylight saving time 93
 DebugService 34, 202
 distribution file 51
 restoring station from a backup 47
 Distribution File Installer 20, 47, 204
 distribution files 47
 importing into the software database 113
 driver features in the license file 58
 drivers 59

E

Edit Filter window 248
 email application 61
 encryption 87
 external SLA battery 167

F

factory defaults 51
 features
 in a license 58
 features to license 57

file sharing	30
file system	23
on a remote controller	176
File Transfer Client.....	20, 133, 205
fileDriver	59

G

global capacity licensing	153–154
and histories	158
counts.....	155
guestEnabled application.....	61

H

hardware features	
that can be licensed.....	57
Hardware Scan Service.....	165
Hardware Scan Service View	206
Heap Memory Monitor	185
histories	
and capacity licensing.....	158
host	
update license.....	71
host license	
updating	69
host models.....	11
HSQL database	
creating and updating the password	18
HTTP	
changing the platform port.....	90

I

Idle CPU Monitor.....	188
Import License window	251

J

JACE-8000	
passphrase.....	90
system passphrase	90
Javelina Battery Platform Service.....	167
Javelina Battery Platform Service Plugin	206

L

language	
changing.....	53
ldapv3 application	62
Lexicon Installer.....	20, 53
Lexicon Installer view	206
license	
archive file	67
by subscription.....	71
delete using a tool.....	144

exporting using the License Manager	67
files	55
global capacity licensing	153–154
importing using License Manager	63
installing using Workbench	65
local license inbox	68
register a subscription license	71
request from the online license server	140
submitting a license request	62
synchronize using a tool.....	145
synchronizing	71
updating the Supervisor database	69
using the Workbench License Manager	
tool.....	141
license key.....	62
License Manager	20, 55, 159
License Manager view.....	202, 206, 241
License Platform Service Plugin	207
license)	
database (local	139
LicenseDatabaseTool.....	159
licenses	
updating	69–70
limitations	13
Loaded Classes Monitor.....	187
local connection.....	15
log	33
history	35
levels for station output.....	34, 202
Logger Configuration	202
LogHistoryService	35
logs	
for controllers, viewing	99
lonworks	59

M

Manage platform daemon users window	252
memory	
configuring for performance	102
leaks.....	180
memory configuration.....	246
messages in the station log	33
Meta Space Memory Monitor	186
mobile application	62
modbusAsync.....	59
modbusCore	59
ModbusSlave.....	59
modbusTcp	59
modbusTcpSlave	59
models.....	11
module	
changing profiles.....	101
modules	111
importing into the software database.....	113
installing on a remote platform	114
uninstall	117
upgrading out-of-date	117

MS/TP	167
MS/TP:RS 485 co-processor	152

N

Nav Container View	19, 173
exporting	20
Network License Summary	70
Network Licenses Summary view	208
New Station Wizard	137
New User window	253
Niagara instance	
register using a command line	74
register using Workbench	72
niagarad	14–15
niagaraDriver	60
NIMH battery	168
NPM Dual Battery Platform Service	168
NPM external SLA battery	168
NPM2 NIMH battery	168
NRE command	74
NRE memory	246
Ntp Platform Service Editor	209
NTP server	255
verifying access	151
NtpPlatformService	151
NtpPlatformServiceNpsdk	162

O

obixDriver	60
onfigure NRE Memory Pools window	246
opc	60
out-of-date modules	
upgrading	117
output settings	
for the platform daemon output	97, 247

P

passphrase	87
changing	88
password	
changing for a platform	86
performance	
improving	167
platform	
connection. opening	15
daemon	14
daemon, installing and starting	14
licensing	55
management	11
platforms with backup batteries	13
supported platforms	11
platform admin user	253
Platform Administration	20, 79, 177
embedded controller	80

workstation	81
Platform Administration view	211
Platform Alarm Support	164
platform daemon	
user home	25
platform daemon output	
settings	247
Platform Daemon Session	177
platform details	82
Platform Monitor	183
platform security	39
Platform Service Container Plugin	218
Platform Service Properties	162
Platform Services	199
platform tools	137
platform user	
changing password	86
creating	85
deleting	87
PlatformServiceContainer	162
PlatformServices	147
plugins	
Certificate Management view	240
power	
monitoring for a controller	149
Power Monitor Platform Service	168
Power Monitor Platform Service Plugin	228
profile	
changing to conserve flash memory in a	
controller	100
protected station home	28
provisioning	19
provisioning application	62

Q

QNX Diagnostics Servlet	95
-------------------------------	----

R

Ram Disk Monitor	192
rdbOracle	60
rdbSqlServer	60
reboot	
remote controller	108
related documents	10
Remote File System	20, 176
resource usage	180
resource.limit application	61
runtime profile	100
changing	101

S

search application	62
security	39
Serial Port Platform Service	169

SerialPortService	170
ServiceQnxJavelina.....	164
ServiceQnxNpm6xx.....	164
session.....	177
Set System Date/Time window.....	254
shared folder	28
SMA expiration properties.....	76
SMA expiration reminder	76
snmp.....	60
Socket State Monitor	193
software database	111
Software Details window.....	255, 259
Software Manager	20, 111
Software Manager view	228
spy.....	36
station	
copying to the daemon user home	138
creating	137
deleting	130
installation troubleshooting	131
installing	120
log levels	34, 202
LogHistory	35
output	34
renaming	128
start and stop.....	31
starting and stopping	32
Station Transfer Wizard	120
station application	61
Station Copier	20, 119–120
dependencies	121
station copier backup	126
Station Copier view	231
station copies	
system passphrase	87
station home	28
station log	33
station spy.....	36
Station Text Summary Editor view.....	175
Station Transfer Wizard.....	122
stations folder	28
subscription license	
requirements	71
subscription licensing.....	71
Supervisor	
sharing files.....	30
Sync Now window	255
Sys Home.....	23
on a controller.....	25
on a Supervisor platform.....	23
syslog configuration.....	178
Syslog Configuration window	256
Syslog Platform Service Plugin view.....	232
System Date Time Editor view.....	234
system logs	
viewing for controllers	99
System Memory Monitor.....	183
System Monitor Config	234
System Monitor Service	180

system passphrase.....	87
System Platform Service Plugin	235
System Platform Service Qnx Plugin	238
system.properties.....	133
SystemService	224

T

Tcp Ip Platform Service Plugin	239
TCP/IP configuration.....	105, 164
TCP/IP Configuration	20
template application	62
time	92–93
setting	94
time zone	92
selecting	94
TLS level	
verify	44
TLS platform settings.....	91
tools	137
Tools Menu	137
troubleshooting.....	35, 97
station installation	131
Trust Store tab	215

U

ui application.....	61
ui.wb application	61
ui.wb.admin application	61
Used CPU Monitor.....	189
user account	
deleting	87
user accounts	
creating	85
User Accounts	252
user authentication	
on a workstation.....	83
user home	25, 30
User Home	23
on a controller.....	25
on a Supervisor platform.....	23
User Key Store	
tab.....	214

V

videoDriver	60
View Details	82
View Details window.....	258
views, See plugins	

W

web application	61
Web Browser View.....	11

windows.....243

workbench application.....61

Workbench user home 30

Workbench User Home 26