

Technical Document

NiagaraAX Platform Guide

September 2, 2016



NiagaraAX 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, NiagaraAX Framework, and Sedona Framework are registered trademarks, and Workbench, WorkPlaceAX, and AXSupervisor, are trademarks of Tridium Inc. All other product names and services mentioned in this publication that is 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 © 2016 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	ix
Document change log	ix
Related documents	x
 Chapter 1 Niagara platform.....	1
Platform overview	1
About a platform connection	3
Platform connection session info	3
Platform daemon (niagarad)	4
Platform daemon port	4
Platform credentials.....	4
Platform access without a platform connection	5
Platform daemon on a PC.....	5
Provisioning versus platform interface.....	5
Types of platform views	5
About platform differences.....	7
QNX-based.....	7
Sun Hotspot JVM or IBM J9 JVM	8
Backup Battery (or not)	8
Battery-less JACE.....	9
Platform view differences, JACE controller vs. Windows-based host.....	9
Platform Administration	9
Windows-based	11
Platform Administration	11
Win64-based Supervisor notes	12
Known Limitations.....	13
Installation and interface differences.....	13
Linux-based Supervisor	13
NiagaraAX platform rights on Linux Supervisor.....	13
Default Linux Supervisor platform administrator.....	14
Linux Supervisor platform views	14
Linux Supervisor port usage notes	14
Platform models.....	14
Application Director	16
Installed applications (stations)	18
Application Director (station management).....	19
Application output	20
Standard output overview	20
Station log levels (spy:/logSetup).....	21
Station LogHistory (LogHistoryService).....	22
Start-up options.....	22
Start checkboxes	26
Application control buttons.....	27
Output control buttons.....	28
Output Settings	29
Certificate Management.....	29
Key Store	30
Trust Store	31
Allowed Hosts.....	31
DDNS Configuration	31
DDNS core configuration items	31
Provider.....	32
Mode.....	32

Adapter	33
About TZO	33
Distribution File Installer.....	33
Security update 1 changes to backup dist usage.....	34
AX-3.8 changes to backup dist usage	34
Operation of the Distribution File Installer	35
Dist file selection.....	35
Distribution file install process	36
About backup dist files.....	37
Restoring a station backup	37
Restoring a backup dist	37
Wiping clean a JACE (clean dist)	39
Clean dist installation preparation	40
Running clean dist on a controller	41
Upgrading a controller.....	42
File Transfer Client.....	43
system.properties notes	44
Editing system.properties	45
GPRS Modem Configuration	46
GPRS modem configuration sections	47
Status and Runtime Data area	48
Lexicon Installer	49
License Manager	50
License operations	52
Import using License Manager	53
Export using License Manager.....	54
License Import results.....	55
About the licensing server.....	56
Platform Administration	57
Types of Platform Administration functions.....	59
View Details.....	60
Update Authentication	61
Digest platform authentication	61
Improvements to digest authentication.....	62
Basic platform authentication	63
System Passphrase	66
Update the system passphrase.....	67
System passphrase usage in backups and station copies.....	67
Editing BOG files offline.....	68
System passphrase usage in JACE-8000.....	68
Change HTTP Port	69
Change TLS Settings.....	69
Change TLS Settings window.....	71
Change Date/Time	73
Set System Date/Time window.....	74
Advanced Options.....	74
Change Output Settings.....	76
View Daemon Output	77
Set Module Filters	78
Results from a change in module content level.....	78
Backup	79
Commissioning.....	81
Reboot.....	81
Software Manager	82
Software Manager notes	82
About your software database	83

Default module listing and layout.....	84
Software Manager table columns	85
Software Details.....	86
Filtering displayed software	87
Filter by status	88
Filter by name.....	88
Software Import.....	88
Import vs. copy into modules.....	89
Software actions	89
Upgrade All Out of Date.....	90
Install	90
Uninstall	90
Re-Install, Upgrade, Downgrade	91
Commit and Reset.....	91
Right-click option to install earlier version.....	91
Station Copier.....	92
Security update 1 changes to Station Copier usage.....	92
When config.bog edits are not needed.....	92
When config.bog edits are needed	93
Station copy direction.....	93
Station Copier dependencies check	94
Station Transfer Wizard	94
Name step.....	95
Delete step.....	95
Content step	96
Disposition step	96
Station settings step	97
Details step	98
Modules step.....	98
Station can be installed with most current modules.....	99
Station can be installed with “out of date” modules	99
Stop station step	100
Review step	100
Transferring station	100
Renaming stations.....	102
Deleting stations	102
TCP/IP Configuration	103
Configuring TCP/IP	103
Configure TCP/IP network settings.....	104
TCP/IP Host fields	107
TCP/IP DNS fields	107
TCP/IP Interface fields	108
IPv4 Settings.....	109
IPv6 Settings.....	111
User Manager	111
WiFi Configuration	115
WiFi Certificate Manager	117
Remote File System	118
Chapter 2 Platform Services	121
About Platform Services	121
Component differences for platform services	122
PlatformServiceContainer parameters	122
PlatformServiceContainer status values	123
PlatformServiceContainer configuration parameters.....	124

Model-specific PlatformServiceContainer properties.....	127
PlatformServiceContainer actions	127
SystemService (under PlatformServices).....	128
Platform service types.....	129
Using platform services in a station.....	130
Power monitoring	130
Battery monitoring disabled	131
PlatformServices binding and link caveats	131
About the NtpPlatformService.....	132
About the Ntp Platform Service Editor.....	133
About the Ntp Platform Service Editor Qnx	133
Ntp Platform Service Editor Qnx settings	134
Ntp Platform Service Editor Qnx time servers	134
Sync Now action	135
About the Ntp Platform Service Editor Win32	135
NTP port/firewall considerations.....	136
Chapter 3 Platform Component Guides.....	137
Components in platCrypto	137
platCrypto-CertManagerService.....	138
platform-DaemonSecureSession.....	138
Components in platDataRecovery module.....	138
platDataRecovery-DataRecoveryService.....	138
Components in platGprs module.....	138
platGprs-GprsPlatformService	138
platGprs-GprsHostSettings.....	139
platGprs-GprsRuntimeData	139
Components in platform module.....	139
platform-DefaultDaemonFileSpace	139
platform-DaemonSession	140
platform-LicenseDatabaseTool	142
platform-LicensePlatformService	142
platform-NtpPlatformServiceQnx.....	142
platform-NtpPlatformServiceWin32.....	142
platform-PlatformAlarmSupport	142
platform-PlatformServiceContainer	142
platform-SystemPlatformServiceQnxJavelina.....	143
platform-SystemPlatformServiceQnxNpm6xx	143
platform-SystemPlatformServiceWin32.....	143
platform-TcpIpPlatformService	143
Components in platHwScan.....	143
platHwScan-HardwareScanService	143
Components in platIEEE8021X	144
platIEEE8021X-IEEE8021XAdapterSettings.....	144
platIEEE8021X-IEEE8021XHostSettings.....	144
platIEEE8021X-IEEE8021XPlatformService	144
Components in platPower module	144
platPower-ExternalSlaBattery	144
platPower-JavelinaBatteryPlatformService	144
platPower-NimhBattery.....	145
platPower-Npm2NimhBattery	145
platPower-NpmDualBatteryPlatformService	145
platPower-NpmExternalSlaBattery	145
platPower-PowerMonitorPlatformServiceQnx.....	146

Components in platPowerNxs module	146
platPowerNxs-PowerMonitorPlatformServiceNxsWin32.....	146
Components in platSerialQnx module	146
platSerialQnx-SerialPortPlatformServiceQnx	146
platSerialQnx-SerialPortQnx.....	146
Components in platSerialWin32 module.....	147
platSerialWin32-SerialPortPlatformServiceWin32.....	147
platSerialWin32-SerialPortWin32.....	147
Components in platSerialWin64 module	147
platSerialWin64-SerialPortPlatformServiceWin64.....	147
platSerialWin64-SerialPortWin64.....	147
Components in platSysmonNx module.....	147
platSysmonNx-HardwareMonitorNxPlatformServiceWin32	148
Components in platSysmonNxs module	148
platSysmonNxs-HardwareMonitorNxsPlatformServiceWin32.....	148
Components in platSysmonNxt module	148
platSysmonNxt-HardwareMonitorNxtPlatformServiceWin32	148
Components in platUsbmon module	148
platUsbmon-UsbMonitorPlatformServiceQnx	148
Components in platWifi module.....	148
platWifi-WifiPlatformService	148
Chapter 4 Platform Plugin Guides.....	151
Plugins in platCrypto	151
platCrypto-CertManagerView.....	151
Plugins in platDaemon module	152
platDaemon-ApplicationDirector.....	152
platDaemon-DistInstaller.....	154
platDaemon-DistributionView.....	154
platDaemon-FileTransferClient.....	154
platDaemon-LexiconInstaller.....	154
platDaemon-LicenseManager.....	155
platDaemon-PlatformSessionListView	155
platDaemon-SoftwareManager.....	155
platDaemon-SoftwareView	155
platDaemon-PlatformAdministration.....	155
platDaemon-R2PlatformTool.....	155
platDaemon-StationCopier	156
platDaemon-StationTextSummaryEditor.....	156
platDaemon-TcplpConfiguration	156
platDaemon-UserManager	156
Plugin in platDataRecovery.....	156
platDataRecovery-DataRecoveryServiceEditor	156
Plugins in platform module	156
platform-LicensePlatformServicePlugin	156
platform-NtpPlatformServiceEditorLinux	157
platform-NtpPlatformServiceEditorQnx	157
platform-NtpPlatformServiceEditorWin32	157
platform-PlatformServiceContainerPlugin	157
platform-PlatformServiceProperties	157
platform-SystemDateTimeEditor.....	157
platform-SystemPlatformServicePlugin.....	157
platform-SystemPlatformServiceQnxPlugin	157
platform-TcplpPlatformServicePlugin	157

platform-WorkbenchLicenseManager.....	158
Plugins in platGprs	158
platGprs-GprsConfiguration	158
platGprs-GprsPlatformServicePlugin.....	158
Plugins in platHwScan.....	158
platHwScan-HardwareScanServiceView	158
Plugins in platIEEE8021X	158
platIEEE8021X-IEEE8021XAdapterSettingsEditor	158
platIEEE8021X-IEEE8021XDaemonSessionPlugin	159
platIEEE8021X-IEEE8021XPlatformServicePlugin.....	159
Plugins in platPower.....	159
platPower-JavelinaBatteryPlatformServicePlugin.....	159
platPower-PowerMonitorPlatformServicePlugin.....	159
Plugins in platWifi module.....	159
platWifi-WifiConfiguration	159
platWifi-WifiPlatformServicePlugin.....	159
platWifi-WifiSecurityManager	160
Chapter 5 License Tools and Files	161
Workbench License Manager	162
Import File using Workbench License Manager.....	163
Export File.....	163
Delete.....	164
Sync Online	165
Request License	165
About the local license database	166
Local license database rationale	167
Local license inbox	167
About license archive (.lar) files	168
About license files	168
Items common to all license files	169
license	169
vendor	169
expiration	169
version	169
hostId.....	169
serialNumber	169
generated.....	169
brand	169
accept.station.in.....	170
accept.station.out.....	170
accept.wb.in	170
accept.wb.out	170
brandId	170
about	170
project.....	170
owner.....	170
signature.....	170
Controller hardware features	170
dataRecovery.....	171
ibmj9j2me.....	171
maxheap.....	171
mstp.....	171
port.limit	171
ndio	171

nrio	172
serial	172
sunj2se	172
Driver attributes	172
name	173
expiration	173
device.limit	173
history.limit	173
point.limit	173
schedule.limit	173
parts	173
Driver types	173
aaphp	174
aapup	174
bacnet	174
export	174
bacnetAws	174
bacnetOws	174
dust	174
fileDriver	174
jen6lp	174
jennic	175
lonworks	175
modbusAsync	175
modbusCore	175
modbusSlave	175
modbusTcp	175
modbusTcpSlave	175
obixDriver	175
export	175
opc	176
niagaraDriver	176
rdbDb2	176
rdbOracle	176
rdbSqlServer	176
sedonanet	176
snmp	176
videoDriver	177
zwave	177
Applications	177
station	177
resource.limit	177
guestEnabled	177
web	177
ui	177
ui.wb	178
ui.wb.admin	178
workbench	178
box	178
crypto	178
eas	179
allCostReports	179
allE2Reports	179
costMeter.limit	179
dataPoint.limit	179
Messaging features	179
fips140-2	179
genericAppliance	179
ieee8021x	179

ldapv3	180
mobile	180
provisioning	180
sedonaProvisioning	180
tenantBilling	180
tenant.limit	180
Chapter 6 Time Zones	181
Time zones and terminology	181
UTC	181
DST	181
Selecting a time zone	182
About the historical time zone database	182
Updating a historical time zone database	183
Chapter 7 Platform Tunneling	185
Platform tunneling overview	185
Platform tunneling requirements	185
Supervisor configuration to support platform tunneling	186
WebService settings	186
FoxService settings	187
Platform tunneling usage	188
Example: Opening Platform dialog if tunneling	188
Connected (via tunneling)	189
Notes on platform tunneling	190
SSL considerations for platform tunneling	190

PREFACE

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. In order to make the most of the information in this book, readers should have some training or previous experience with Niagara 4 or NiagaraAX software.

Document content

This document provides information about Niagara platform services, components and plugins, license tools and other topics related to NiagaraAX-3.8U1, AX-3.8, and AX-3.7U1 hosts.

Document change log

Changes to this document are listed below.

- Updated September 2, 2016
Includes many minor changes throughout related to reverting document content to AX. Restored several component and plugin sections, previously removed.
- Updated July 21, 2016
Many changes throughout to revert Niagara 4 content inadvertently added to this AX version of the guide in a prior revision.
- Updated June 6, 2016:
 - This update is related to NiagaraAX-3.8U1. Added a note to the end of the “Commissioning” topic explaining that non-portable password encoding in AX-3.6 and AX-3.7 stations prevents upgrading those stations to AX-3.8 without first converting the passwords to a portable encoding format. This can be done via the “plat makeportable” command available in AX-3.8U1.
- NiagaraAX-3.8U1 update: April 22, 2016
Includes many minor changes throughout, in addition to the following:
 - In the topic “Niagara Platform”, added bullet item for new features in AX-3.8U1 (Java Web Start and JACE-8000-AX).
 - In the topics “About Platform differences”, under QNX-based platforms, added a section on platform differences for JACE-8000 controllers running AX-3.8U1.
 - In the topic “Models of platforms”, added info about JACE-8000 controllers running AX-3.8U1.
 - In the topic “Distribution File Installer”, added reference to conversion dist used to downgrade JACE-8000 controllers from N4 to AX-3.8U1.
 - Inserted topics on system passphrase functionality which apply to JACE-8000 controllers running AX-3.8U1.
 - In the topic “Station Copier dependencies check”, a paragraph mentions the need to commission JACE-8000 controllers that are converted from N4 to AX prior to copying a station to it.

- In the topics “WiFi Configuration” and “WiFi Certificate Manager”, added a note explaining that although JACE-8000 controllers have WiFi capability, WiFi is not supported when the controller is running AX.
- Added an explanatory note regarding the presence of some Niagara 4 content in this version of the Platform Guide to the following chapters: “Platform Services”, “Platform Components Guides” and “Platform Plugins Guides”, “License Tools and Files”, and “Time Zones and Niagara”.
- Added a cautionary note regarding upgrading AX-3.6U4 stations with CryptoService to AX-3.8U1 to the “Commissioning” topic.
- Update: April 30, 2014, minor changes
- Update: December 18, 2013, minor changes
- AX-3.8 update: November 5, 2013
- NiagaraAX-3.7 “Update 1” revision, May 30, 2013
- NiagaraAX-3.7 revision, August 30, 2012

Related documents

Additional information about Workbench operation, and AX-3.8U1 features (JACE-8000-AX, Java Web Start) is available in the following documents.

- *NiagaraAX User Guide*
- *JACE-8000 Install and Startup Guide*
- *Niagara Web Start Guide*

CHAPTER 1 NIAGARA PLATFORM

TOPICS COVERED IN THIS CHAPTER

Platform overview
About platform differences
Application Director
Certificate Management
DDNS Configuration
Distribution File Installer
Upgrading a controller
File Transfer Client
GPRS Modem Configuration
Lexicon Installer
License Manager
Platform Administration
Software Manager
Station Copier
TCP/IP Configuration
User Manager
WiFi Configuration
WiFi Certificate Manager
Remote File System

Platform is the name for everything that is installed on a Niagara host that is not part of a Niagara station. The platform interface provides a way to address all the support tasks that allow you to setup and support and troubleshoot a Niagara host.

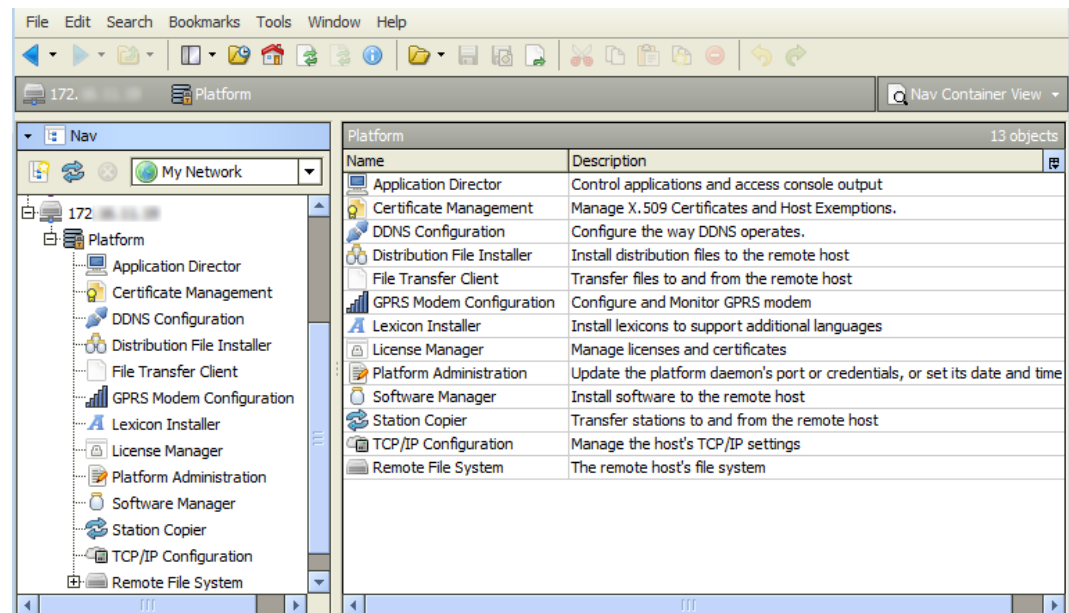
In update release, AX-3.8U1, two new features were added: Java Web Start and JACE-8000-AX.

- Java Web Start offers an alternative to running the WbApplet in a browser by providing an application which can be launched by Java Web Start. Useful, since many late version browsers have disabled support for the NPAPI Java plugin which prevents the Java WbApplet from running in a browser.
- JACE-8000-AX, when added to a N4 JACE-8000 license, enables the platform to run AX-3.8U1. This allows customers to extend their existing AX system with a JACE-8000 (running AX), take advantage of new features such as Analytics, and have the capability to upgrade the unit to Niagara 4.x at a later time.

In AX-3.7, dialup modem support ended (no longer any **Dialup Configuration** view). Also, in some cases a **Sedona Environment Manager** view may appear-see the related note in the section “Platform overview”.

Platform overview

In Workbench, when you open a platform connection to a Niagara host (whether controller or Supervisor), that host’s available platform functions are listed in the platform’s Nav Container View, as shown below.

Figure 1. Platform functions listed in Nav Container View

Each platform function has its own Workbench view (plugin); you access it by simply double-clicking. Most of the same platform views exist whether a platform connection to a controller or a Supervisor, with these exceptions:

- If you open a local platform connection at your computer, note that some platform views appear to be missing, for example the **Distribution File Installer**, **File Transfer Client**, and **Software Manager** are not in the list. These views have no application when working at your computer—instead, you simply use Windows Explorer.
- For any Windows-based platform, a **User Manager** view is available. This view is not available if the platform is a QNX-based JACE or a Linux-based Supervisor.
- For any QNX-based JACE platform, a **GPRS Modem Configuration** view is available. This view is not available if the platform is a Windows-based JACE or a any Supervisor.
- Starting in AX-3.7, some platforms may have a **Certificate Management** view. This view appears only for a host licensed for SSL. This view is unavailable of the older QNX-based JACE platforms that run the IBM J9 Java VM (JACE-2, JACE-4, JACE-5 series).
- In AX-3.8, some QNX-based JACE platforms may have an **IEEE 802.1X Configuration** view. This view appears only if the host is licensed for IEEE 802.1X, with the **plati-IEEE8021X** module installed.
- A JACE with an installed WiFi option has two related platform views: **WiFi Configuration** and **WiFi Certificate Manager**. Currently, a JACE-700 is the only applicable platform.
- Starting in AX-3.6 (3.6.44 or later), platform support was added for Niagara R2 configuration and upkeep of retrofit board R2JACE-403 and JACE-545 controllers (also known as JACE-603 and JACE-645 controllers). A related **R2 Platform Tool** view is available only in a platform connection to these two models (if configured for Niagara R2).

Also, a few of the platform views differ depending on platform type.

The following sections provide additional background on Niagara platform access:

- About a platform connection
- Provisioning versus platform interface
- Types of platform views

- About platform differences

NOTE: If your 3.7 or later Niagara Workbench has been enabled for Sedona Framework TXS 1.2 (via the **Sedona Installer** tool), note any platform connection to a remote AX host provides yet another platform view: the **Sedona Environment Manager**. Note this platform view appears regardless if the remote host is configured with Sedona-related modules (**nsedona**, **sedonanet**, etc.). Refer to the NiagaraAX Sedona Framework TXS-1.2 Installer Guide for details on the Sedona Installer tools in Workbench, and (if enabled for Sedona TXS 1.2) the NiagaraAX Sedona Framework TXS 1.2 Networks Guide for complete details on the **Sedona Environment Manager**.

About a platform connection



A platform connection is different than a station connection. When connected to a Niagara platform, Workbench communicates (as a client) to the host's *platform daemon* (also known as "niagarad" for Niagara daemon), a server process.

Unlike a station connection that uses the Fox protocol, a client platform connection ordinarily requires full Workbench, meaning it is unavailable using a standard Web browser (that is, using the "Web Workbench" applet).

NOTE: Browser access of a Supervisor station can provide platform connectivity, albeit indirectly, through its **ProvisioningService**.

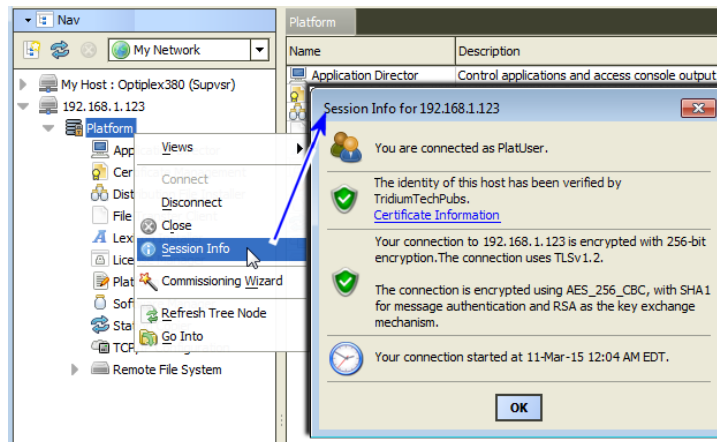
Platform connection session info

It is possible to open a *secure* (encrypted, SSL or TLS) platform connection to most types of AX hosts, providing the host is properly configured. The platform-connection session icon appears in the Nav tree with a small padlock to indicate this connection type, that is:

either  for secure "platformtls" using TLS (Transport Layer Security) encryption, or  for regular (unencrypted).

NOTE: For best security, always use TLS. In Workbench, default **Open Platform** and **Open Station** (Foxs) assume a secure connection, where to connect in a regular (unencrypted) fashion you must change the connection **Type** first.

Once the platform is connected, the available platform functions are identical—regardless of connection method. Workbench provides a right-click **Session Info** action on any platform connection, as well as any station (Foxs) connection.

Figure 2. Example right-click Session Info dialog for a secure (TLS) platform connection

The figure above shows an example of this client session info from a secure (TLS) platform connection. In this example, the identity of the (server) has been verified by a signed certificate, and all data on this connection is being encrypted.

For complete details on SSL configuration, refer to the *NiagaraAX SSL Connectivity Guide*.

Platform daemon (niagarad)

The platform daemon is an executable that runs independently from Niagara core runtime, and is pre-installed on every JACE controller as factory-shipped, and runs whenever the JACE 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.

Note that older JACE platforms (JACE-2, JACE-4, JACE-5 series) still use a platform daemon written in "native code", which runs without a Java VM. A single J9 Java VM is used for the station process.

Platform daemon port

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.

Platform credentials

Finally, as a platform client, you sign on using "host level" credentials for authentication. This means a user account and password separate from any station user account. Consider it the *highest level access to that 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. During the start-up commissioning process, you should always change platform credentials from defaults to something known only to your company and/or customers. In AX-3.8, measures were added to alert you (and other platform users) to any JACE running with default platform credentials. For related details, see Update Authentication.

Platform access without a platform 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**.

Platform daemon on a PC

When you install Niagara on your PC, one of the last “Would you like to?” install options is:

Install and Start Platform Daemon

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

Once installed and started on a PC, you can see the platform daemon listed as a *Niagara service* from the Windows Control Panel, by selecting **Administrative Tools > Services**.

NOTE: Alternatively, after installing Niagara on your PC, you can install and start the platform daemon at any time, if needed. From the Windows Start menu, click **All Programs > Niagara 3.8 nnn > Install Platform Daemon** (shortcut for “`plat.exe installdaemon`”).

In summary, your PC’s local platform daemon is not necessary for making client platform connections to other Niagara hosts, only to provide the ability to run a station locally on your PC.

Provisioning versus platform interface

The focus in this document is about the Niagara platform user interface, meaning the different platform views and functions available when you (a Workbench user) open a direct platform connection to a Niagara host. These same views and functions are available when you open a “tunneled” platform connection to a host, through an opened Supervisor station.

However, be aware that a Supervisor station can perform “provisioning”, which can automate some platform tasks. Provisioning typically applies to its subordinate JACEs, which are represented in the Supervisor station as Niagara Stations (devices) under its Niagara Network.

For more details, see the “Niagara Provisioning overview” in the *Niagara Provisioning Guide for Niagara Networks* document.

NOTE: Some of the provisioning views provided by a Supervisor are nearly identical to platform views described in this document, including the **Software Manager** and **Application Director (Station Director)**, and work in the same fashion. However, if new to Niagara, it is recommended that you become familiar with “direct” platform views described in this document, before using provisioning in a Supervisor.

Types of platform views

The Workbench platform connection to a Niagara host, either JACE or Supervisor, provides various functional views.

NOTE: In addition to the platform views listed below, a **Commissioning Wizard** is available as a right-click platform option. This wizard provides a “step-by-step” method to perform a sequence of platform tasks used for Niagara commissioning of a new JACE controller, or when upgrading Niagara in controller. For more details, see “About the Commissioning Wizard” in the *JACE NiagaraAX Install & Startup Guide*.

The following sections summarize the various Niagara platform functions and views, including typical usage:

- **Application Director**
To start, stop, restart, or kill a station on the platform. Output from the station displays in the view pane, useful for monitoring and troubleshooting. You also configure a station's **Auto-Start** and **Restart on Failure** settings from this view.
- **Certificate Management**
To import signed PKI certificates into the platform's key store and trust store for SSL/TLS secure connections, and to perform related functions. See *NiagaraAX SSL Connectivity Guide*.
- **DDNS Configuration**
Allows for DNS IP addresses to be dynamically updated (DDNS), an option sometimes used for hosts – although infrequently.
- **Distribution File Installer**
To restore a backup .dist file to the target controller, or to install a clean dist file to wipe the file system of a controller to a near-factory minimum state. Note, do not attempt to use this to upgrade a JACE—instead, you must use the Commissioning Wizard.
- **File Transfer Client**
To copy files between your Workbench PC and the 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 controller.
- **GPRS Modem Configuration**
(When connected to a QNX-based JACE) To configure the wireless GPRS modem option card (General Packet Radio Service) that may be installed in a JACE-2,-3,-6,-7 series controller.
- **IEEE 802.1X Configuration**
(Applies only if an AX-3.8 "Hotspot" JACE that is licensed for IEEE 802.1X, with platIEEE8021X module). To configure settings for the JACE to allow it to join an IEEE 802.1X wired-authentication network. Refer to the Engineering Notes 3.8 document NiagaraAX IEEE 802.1X Configuration.
- **Lexicon Installer**
To install file-based Niagara lexicon sets from your Workbench PC to the remote platform, to provide non-English language support, or to customize English display of selected items. Starting in AX-3.7, lexicons are distributed as software modules (.jar files), which are installed using the platform Software Manager view instead.
- **License Manager**
To review, install, save, or delete licenses and (license) certificates on the remote Niagara platform.
- **Platform Administration**
To perform configuration, status, and troubleshooting of the Niagara platform daemon. Included are commands to change time/date, backup all remote configuration, and reboot the host platform. Also included are functions to modify platform users, specify the TCP port monitored by the platform daemon, and various settings for a secure (TLS) platform connection.
- **Software Manager**

To review, install, update, or uninstall Niagara modules (.jars) on the remote Niagara platform. The **Software Manager** compares modules installed on the connected platform against those available (locally) in **Sys Home** on your Workbench PC.

- Station Copier

To *install* (copy) a station from your Workbench **User Home** to a remote platform (or if a Supervisor, to the local PC's daemon **User Home**). Also to *backup* (copy) a station to your Workbench **User Home**, or to *delete* a remote station. You can also *rename* stations.

NOTE: Starting in 2013 update releases (e.g AX-3.7u1), password storage methods changed. This can affect usage of the Station Copier, especially if installing a saved station to a different host than the original (saved) host. For complete details, refer to the document *NiagaraAX 2013 Security Updates*.

- TCP/IP Configuration

To review and configure the TCP/IP settings for the network adapter(s) of the Niagara platform.

- User Manager

Applies to remote Win32 platforms (e.g. JACE-NXT). To access host Windows OS user and group accounts, including ability to add or delete users/groups, change passwords and group members.

- WiFi Certificate Manager, WiFi Configuration

Applies to remote JACE with installed WiFi option (currently, a JACE-700 only). Used to configure the 802.11b/g wireless interface provided by the WiFi option.

- Remote File System

For read-only access to folders and files on the remote platform, including all those under its Niagara system home (**Sys Home**) and daemon **User Home**.

In some cases, you may also have a Sedona Environment Manager platform view listed too. See the related Note in the section Platform overview.

About platform differences

Depending on the platform type opened, some platform views differ. There are three main categories of platforms, by OS (operating system) used. In order of frequency, these include:

- QNX-based (JACE controllers)
- Windows-based hosts (Supervisor hosts)
- Linux-based Supervisor

There are various controller host *models*, each with a "model" string descriptor. For a list of host models that support NiagaraAX, current with this document.

QNX-based

Sometimes called "embedded" JACE controllers, these include all models shipped with the QNX operating system. All these devices use onboard flash memory for file storage and provide wired Ethernet connectivity.

The "embedded" JACE controllers include the following:

- JACE-8000
- JACE-2,-3,-6,-7 series models
- JACE-4,-5 series models

The JACE-3,-6,-7 series offer an option for an onboard wireless (GPRS) modem, and the JACE-700 offers a wireless 802.11b/g (WiFi) option.

The JACE-8000 platform, introduced with the release of Niagara 4, provides a number of exclusive features, such as integral WiFi (802.11b/g) support, backup and restore to and from a removable USB drive, and easy communications expansion using attachable modules.

JACE-8000 controllers initially supported Niagara 4 only—however update build AX-3.8U1 provides JACE-8000 support with some feature limitations (specifically WiFi and USB backup/restore functionality is not supported when running AX). Whereas the JACE-3,-6,-7 series controllers were originally released with NiagaraAX, and may be migrated to Niagara 4 if already running AX-3.8, or else configured from scratch using Niagara 4.

NOTE: Starting in AX-3.7, support ended for dialup modem operation in Niagara (dialup modem option card for a JACE controller, and/or external dialup modem for a controller or Supervisor).

Sun Hotspot JVM or IBM J9 JVM

In earlier (pre-AX-3.6) releases, all QNX-based JACE controllers used the IBM J9 JVM (Java Virtual Machine) to host the Niagara Runtime Environment (NRE) for running a station. Starting in AX-3.6, more recent controllers, along with the newest (JACE-3E, JACE-6E, JACE-603, JACE-645), now use Oracle's Sun Hotspot Java VM—the same VM type used in Windows-based NiagaraAX platforms.

For any JACE-6 or JACE-7 series controller upgraded from an earlier (pre-AX-3.6) release, the core software distribution automatically replaces the J9 JVM with the Hotspot JVM. The associated license upgrade includes the required "sunj2se" feature, needed to allow the JACE to operate.

The Hotspot JVM provides a significant performance improvement. Plus, the Hotspot JVM provides J2SE support—useful for developers and system integrators skilled in creating program components or custom applications (written in Java). This allows many of the newer Java APIs, which have never been supported by the J2ME version in the IBM J9 JVM.

NOTE: Due to resource limits, the JACE-2 series (all NPM2-based) controllers and previous (JACE-4, JACE-5) continue to use the IBM J9 JVM, regardless of NiagaraAX release level. For the same reason in AX-3.7, these controllers also continue to use a Niagara platform daemon (niagarad) written in "native code", rather than a Java-based platform daemon (see "Platform daemon (niagarad)" for related details).

For the most part, these differences in Java VM and platform daemon are typically "transparent" to the normal configuration of the controller's hosted station or platform.

However, there are now notable advantages for a controller using the Hotspot JVM, as follows:

- Supports IPv6 in its TCP/IP platform configuration. See "TCP/IP changes in AX-3.6" for related details.
- When running AX-3.7 or later, capable of supporting secure encrypted (SSL) connections. See platform connection.

For reasons like these, the two subgroups of QNX-based controllers are sometimes referred to as either "Hotspot JACE" or "J9 JACE" in this document.

Backup Battery (or not)

JACE-6 and JACE-7 controller models use an onboard NiMH backup battery (nickel metal hydride), used to preserve runtime data, and also allow continuous operation during brief power outages. The JACE-3E and JACE-6E controllers use integral SRAM to backup runtime data, but can also *optionally* use an onboard NiMH battery for continuous power event operation.

The JACE-7 and JACE-603/JACE-645 models also support an additional external 12V SLA (sealed lead acid) battery for backup usage.

For any of the above controller models, the JACE station provides a “power monitoring” component to track its AC power and backup battery level, with a configurable delay for orderly shutdown of the JACE upon AC power failures. Access power monitoring of a JACE in the **PowerMonitorService** in a running *station*. See “*About Platform Services*”.

Battery-less JACE

Starting at the initial AX-3.6 release, an “SRAM option card” became available for any JACE-2,-6,-7 series controller. If installed, the controller can operate without any backup battery, on-board NiMH or otherwise. SRAM support works via a station platform service, the “DataRecoveryService.”

The DataRecoveryService continuously records all database changes in SRAM, and upon reboot from a power event, restores (plays back) these changes. In the initial AX-3.6 release, the “DataRecoveryService” automatically replaced the “PowerMonitorService” in the JACE station's PlatformServices.

Now, the latest JACE-3E and JACE-6E controllers include integral onboard SRAM, making the SRAM option card unnecessary. Although these controllers ship without a NiMH backup battery, you can field install a NiMH backup battery as an option.

Starting in AX-3.6.44, NiagaraAX support for SRAM and backup battery changed, to allow the usage of both backup methods. Now, any SRAM-equipped JACE controller can also have a backup battery, and be configured to use either SRAM or backup battery, or both. By default, its station's PlatformServices contains both the PowerMonitorService and the DataRecoveryService.

For details, refer to the *Engineering Notes II* document “JACE Data Recovery Service (SRAM support)”.

Platform view differences, JACE controller vs. Windows-based host

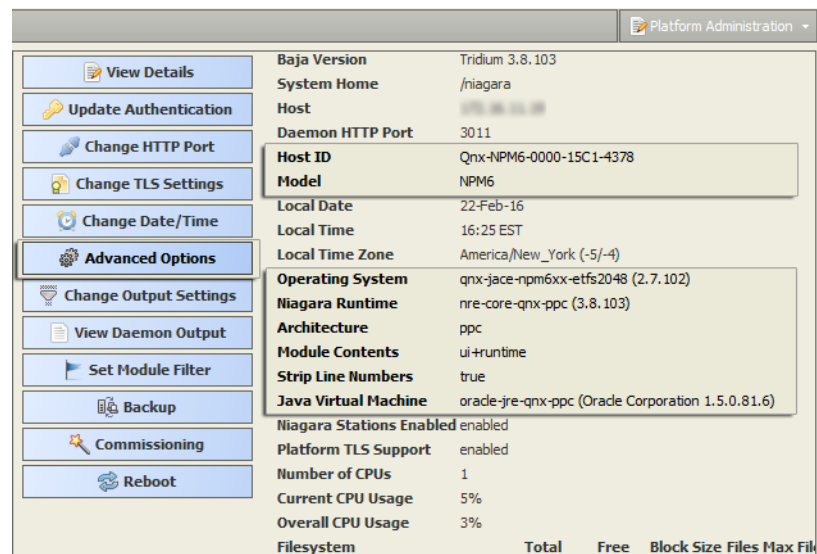
For any JACE controller platform, the following platform views differ from Windows-based platforms.

- Platform Administration

Also, in the **Application Director**, you cannot **Start** a station after manually stopping it. You must **Reboot** the controller instead. See “Application and output controls.”

Platform Administration

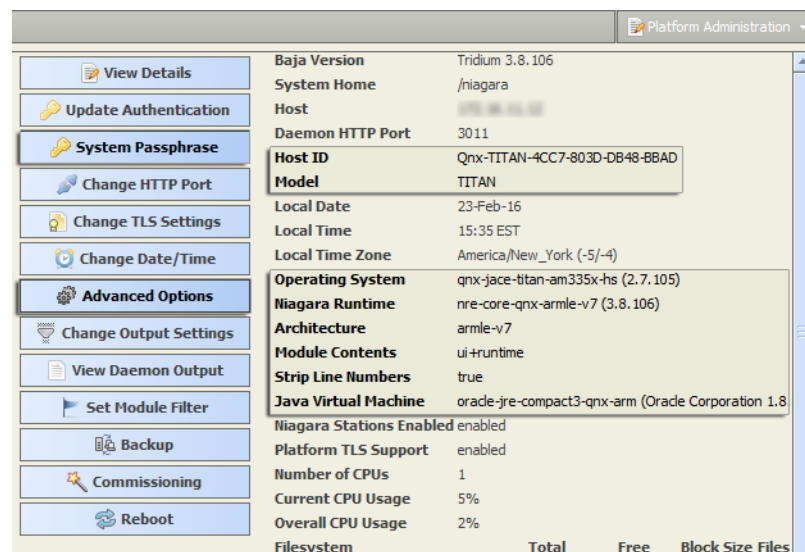
Platform Administration for a JACE platform differs as shown below:

Figure 3. Platform Administration for JACE controller

- An **Advanced options** button is available for enabling/disabling FTP/Telnet, and Daemon Debug functions. For details, see “Advanced Options”.
- Various data in the view (repeated in “View Details”) differ greatly from that for Windows hosts.

Platform differences for JACE-8000 running AX

In AX-3.8U1, adding the JACE-8000-AX license feature to a JACE-8000 enables the platform to run AX. Platform Administration differs as shown below:

Figure 4. Platform Administration for JACE-8000 controller

- The **System Passphrase** button is available for changing the passphrase used to encrypt sensitive information on the platform’s file system.
- The **Advanced options** button is available for enabling/disabling SFTP/SSH, and Deamon Debug functions. For details, see “Advanced Options”.

- Various data in the view (repeated in “View Details”) differ greatly from that for Windows hosts.

Windows-based

Windows-based platforms include Win-32-based JACE hosts like the JACE-NXT (and previous JACE-NXS and JACE-NX models), and most Windows-based PC hosts and SoftJACE hosts. File storage is typically a hard drive, and the operating system is typically either Windows 7 Professional, Windows Vista Business, or Windows XP Professional. Alternatively, a Supervisor may be running Windows Server 2003 or Windows Server 2008.

NOTE:

- AX-3.8 also supports Windows 8 Professional and Windows Server 2012, in addition to the other Windows operating systems mentioned.
- NiagaraAX Supervisor support was added for Windows 64-bit OS (Win64-based), including Win64 editions of Windows Vista, Windows 7, and Windows Server 2003 and 2008 (and if AX-3.8, also Windows 8 Professional and Windows Server 2012). Although a Win64 Supervisor uses a 64-bit JVM (Java Virtual Machine) and different NRE core binaries, its NiagaraAX platform interface is nearly identical to any Win32-based Supervisor. Therefore, you can equate a Win64-based Supervisor as a “Windows-based” host in various discussions in this document, unless particularly noted. For further details, see Win64-based Supervisor notes.
- The JACE-NXT, like the preceding JACE-NXS model, is a Win32-based platform, is available in both a CompactFlash-memory based model and a hard-drive based model. In either case, “Windows XP Embedded” is the operating system.

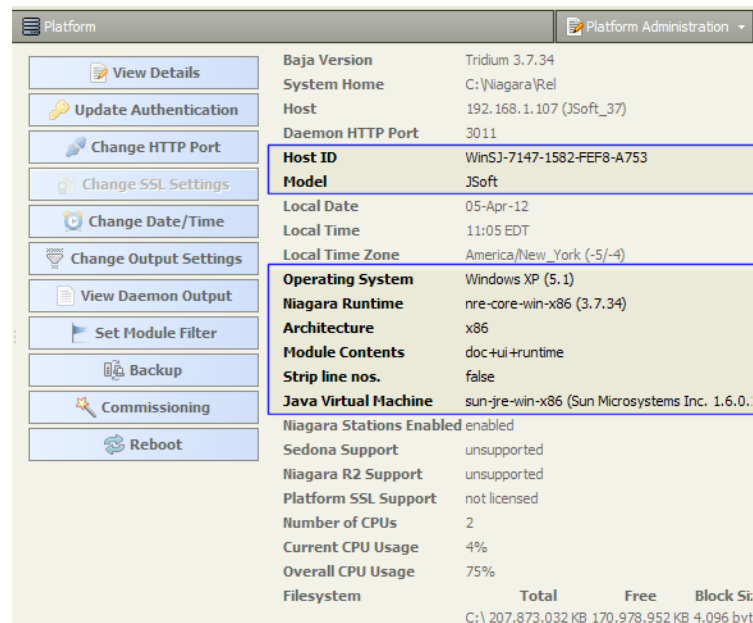
For any Windows-based platform, the following platform views differ from JACE controller platforms:

- Platform Administration

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

Platform Administration

Platform Administration for a Windows-based platform differs as follows:

Figure 5. Platform Administration for Windows-based platform

- No **Advanced Options** button is available (FTP/Telnet configuration can be done directly using Windows).
- Choices available from the **Update Authentication** function are more involved.

NOTE: If you have your local PC platform open, such as a Supervisor, buttons **Set Module Filter**, **Commissioning**, and **Reboot** are unavailable.

- Setting the Module Filter is intended only for initial configuration in a remote JACE. For more details, see “Set Module Filter.”
- The Commissioning Wizard is intended only for initial Niagara installation and startup in a remote JACE, or whenever upgrading a JACE. For more details, see “Commissioning.”
- Reboot is intended only for remote JACE platforms (see Reboot). To locally reboot a Supervisor, you should stop its local station, exit Workbench, then restart the operating system.

See “Platform Administration” for more details.

Win64-based Supervisor notes

Supervisor support for installations on PCs running a 64-bit Windows operating system is typical, for example Windows Server 2012 or Windows 7 or 8 Professional 64-bit. The primary application for a 64-bit installation is for a Supervisor station with a large NiagaraNetwork (a job with a large numbers of controllers, each with many proxy points), and thus, a large station database.

In particular, the 64-bit Java VM (Virtual Machine) does not have a 2GB memory limit, unlike the Java VM on a Win32-based Supervisor. Typically, any PC with 64-bit Windows also has 4GB or more of RAM installed, and, unlike a 32-bit Windows PC, the 64-bit OS can effectively utilize all of it. Therefore, a 64-bit Windows host may be the solution for the largest enterprise level Supervisor.

Known Limitations

At the time of this document update, there are several known limitations for a Supervisor running on a 64-bit Windows operating system. Although most of these do not apply to a typical Supervisor, they should be understood before installation time.

These 64-bit Windows platform limitations include the following:

- 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*.

- Lonworks FTT-10 is not fully supported on a 64-bit Windows platform—although there are Echelon 64-bit drivers, most are 32-bit drivers in a “64-bit wrapper”, and are likely unsuitable. Further, a Supervisor is not typically licensed for Lonworks. However “LonIP” is supported.

Installation and interface differences

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 platform services under its **PlatformServices** as with a station running on a Win32 host.

Linux-based Supervisor

Supervisor software is available targeted for a specific Linux-based platform: an Intel-based PC platform running the OS of Red Hat Enterprise Linux 5. NiagaraAX installation on this platform is done as user “root” using the supplied “Bash” install script. This results in a “niagarad” user and group added, where almost all of the installed software files use niagarad as both owner/group.

During the install script process, existing users of the Linux host platform can be added as Workbench users. This includes menu options to start Workbench and/or the Niagara Console application at the Supervisor PC.

Refer to the *Linux AX Supervisor Notes* document for further installation details.

NiagaraAX platform rights on Linux Supervisor

During the install script process for the Supervisor Linux platform, a choice is presented as to whether NiagaraAX users should be allowed to perform certain “root-privileged” tasks. These include tasks such as specifying the host’s date and time, time zone, TCP/IP settings, and NTP settings, as made available in various platform views. Note that in addition to the single NiagaraAX platform administrator, these items may be available to Supervisor station users too, via views of the different Platform service types (for users with admin-level permissions on PlatformServices).

The default install choice for this is “no,” such that related items in the platform views appear as read-only. However, if this is changed to “yes” at installation time, NiagaraAX is installed such that the Niagara platform administrator user will have the ability to modify these settings, as well as any Supervisor station users with admin-write permissions on the station’s PlatformServices.

Default Linux Supervisor platform administrator

Following installation, the (single) default NiagaraAX platform administrator has these credentials:

- Username: tridium
- Password: niagara

On any real job, these credentials should always be immediately changed, by opening a platform connection and using the **Update Authentication** option in the **Platform Administration** view. Note this is particularly important if the "root-privileged" tasks were enabled at installation time.

Linux Supervisor platform views

A platform connection to a Linux-based Supervisor provides the same collection of platform views as to Windows-based Supervisor, except the following views are not present:

- DDNS Configuration
- Dialup Configuration
- User Manager (always specific to Windows-based hosts only)

Platform service types in the Supervisor's station also include fewer types than in other host platforms, currently limited to the TcpIpService, LicenseService, and NtpPlatformServiceLinux.

Linux Supervisor port usage notes

Note that the station running on a Linux Supervisor is "owned" by a specially created user/group niagarad:niagarad, and therefore cannot bind to Linux "root owned" software ports 1-1024. This is not an issue for the conventional port (3011) used for a platform connection, but does affect the standard port used by the station's WebService (Http Port), which cannot be used at the default port (80) setting. In addition, other software ports potentially used by various drivers must be adjusted above port 1024.

Platform models

Among the two groups of JACE controllers (embedded controllers and Windows-based), there are different models, each of which has a host model text descriptor. You see this descriptor 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 that host.

The following table lists various controller models starting with the host model text descriptor.

A few models listed *are not compatible* with Niagara 4, and are so noted. However, it is possible that some may exist on a job site where the Supervisor was migrated to N4.0, along with some number of controllers that *are* compatible. For related background details, refer to the *AX to N4 Migration Guide*. In addition, several models discontinued more than 10 years are not listed.

Table 1.1. Host models of JACE platforms

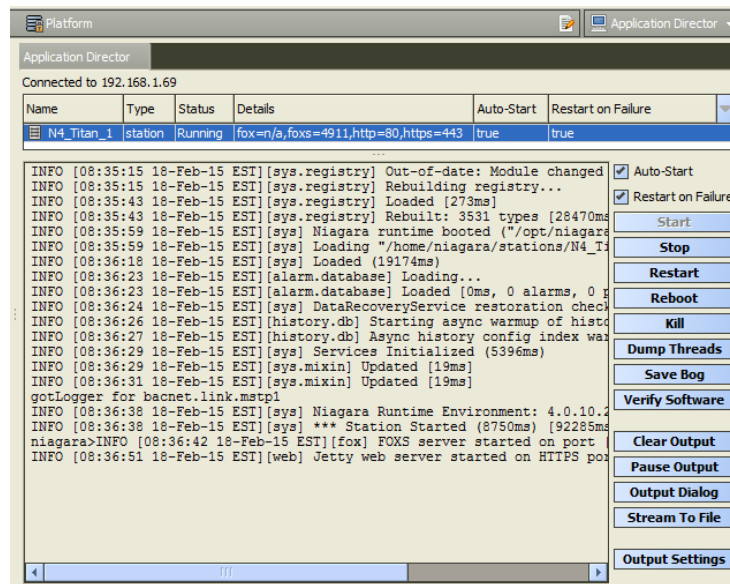
Model desc.	Actual Model	Notes	Compatible with Niagara 4?
JNXS	JACE-NXS	Discontinued Win32-based (Windows XP Embedded), model before JACE-NXT series.	No
JNXT	JACE-NXT series	Discontinued Win32-based controller (Windows XP Embedded).	No
Jsoft	SoftJACE installed on user-supplied PC	Windows-based. This is different than a Supervisor for example, which appears instead as "Workstation".	Yes
JVLN	JACE-7 series (JACE-700)	Discontinued model, with more processing power than JACE-2/6 series.	No, the WiFi option for this controller is not supported
NPM2	JACE-2 series	Discontinued QNX- based controller. Uses the IBM J9 JVM (Java Virtual Machine).	No
NPM3	JACE-3E series, introduced in mid 2013.	QNX-based controller, JACE-3E is between the JACE-2 series and the JACE-6E series in performance, and includes onboard SRAM for battery-less operation (if desired).	Yes
NPM6	JACE-6 series	Discontinued QNX- based, with more processing power than the JACE-2 series. NOTE: Security controllers are not compatible.	Yes, except for security controller
NPM6E	JACE-6E, as well as "retrofit board" controllers JACE-603 and JACE-645.	QNX-based controllers. NPM6E based controllers include onboard SRAM for battery-less operation (if desired). The JACE-603 and JACE-645 are retrofitted R2 JACE-403 and JACE-545 controllers.	Yes

Model desc.	Actual Model	Notes	Compatible with Niagara 4?
TITAN	JACE-8000 series	<p>QNX-based controller, with the most processing power and resource capacity of any controller. Includes onboard SRAM for operation without a battery, integral WiFi, and a USB port for backup/restore usage using a USB flash drive. Plug-in option modules provide additional communications ports. Supports Niagara 4, and AX-3.8U1 (with the JACE-8000-AX license feature). Refer to the <i>JACE-8000 Install and Startup Guide</i> for commissioning details.</p> <hr/> <p>NOTE: The USB Backup/Restore and WiFi functionality are not supported on the JACE-8000 platform running AX-3.8U1.</p>	Yes, except when running older software
Workstation	User-supplied PC, for example, a Supervisor or engineering workstation.	Windows-based customer supplied PC that runs Workbench, minimally.	Yes

Some platform views differ depending on the type of controller.

Application Director

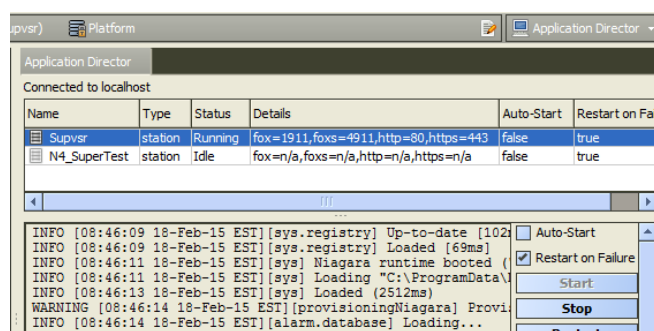
The Application Director is one of several platform views. You use it to *start* or *stop* a *station* in any Niagara host (whether a remote JACE or a local or remote Supervisor PC), as well as see *station output* for troubleshooting purposes. From the Application Director, you also define a station's "restart" settings, plus have access to other station actions.

Figure 6. Application Director view, looking at Niagara station

As shown above, the **Application Director** is split into three main areas:

- Installed application (stations) — at top
- Application output — main area
 - Related are log levels defined for the station. See the Station log levels (DebugService).
- Application and output controls — right-side checkboxes and buttons

NOTE: In the Application Director for any JACE, 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 7. Application Director for Supervisor host showing multiple stations

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, in order 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.

Installed applications (stations)

The top area of the **Application Director** shows a table of installed applications (stations), as shown below.

Figure 8. Application Director installed applications

Name	Type	Status	Details	Auto-Start	Restart on Failure
J8000AX_demo	station	Running	fox=1911,foxs=4911,http=80,https=443	false	true

Every 1.5 seconds, the platform daemon fetches data about the station(s) and updates this in the following columns:

- **Name**
The name of the station directory.
- **Type**
This is always “station” for a Niagara station.
- **Status**
One of the following, as applied to a station:
 - Idle — Station is not running, but can be started without a reboot.
 - Running — Station is running.
 - Starting — Platform daemon has started the station, but the station has not reported back its status back to the daemon.
 - Stopping — Daemon has ordered the station to stop, but its process has not yet terminated.
 - Halted — Station is not currently running, and cannot be restarted without a reboot.
 - Failed — Station terminated with a failure exit code.
- **Details**
For any station, shows four items:
 - `fox`= TCP/IP port monitored for regular (unencrypted) Fox connections to Workbench and other Niagara stations. Shows “n/a” if station is not running, or if Fox Enabled is set to false.
 - `foxs`= TCP/IP port monitored for secure Fox connections to Workbench and other Niagara stations, if so configured. Shows “n/a” 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 false.
 - `http`= HTTP port that the station’s WebService monitors for regular (unencrypted) browser connections to the station. Shows “n/a” if station is not running, or if it does not have a running WebService, or if Http Enabled is set to false.
 - `https`= HTTP port that the station’s WebService monitors for secure browser connections to the station, if so configured. Shows “n/a” 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 false, or if the station does not have a running WebService.
- **Auto-Start**
Either true or false. If true, the station starts whenever the platform daemon starts. Configured with a right-side checkbox.
- **Restart on Failure**

Either true or false. If true, the daemon automatically restarts the station after it terminates with a failure exit code. Configure with a right-side checkbox.

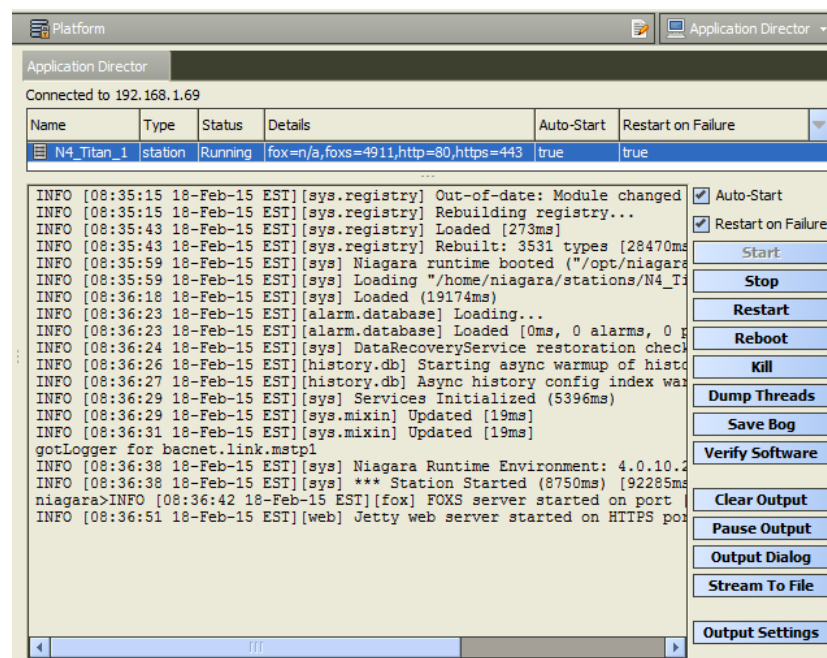
Apart from the data shown in the table, application selections are possible.

Application Director (station management)

The **Application Director** (📁) is the platform view that allows you to start and stop a station running in any host (whether a remote JACE, a local, or a remote Supervisor PC) that is connected to a Niagara platform.

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 9. Application Director view, looking at a station



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

- To select a station, click the row in the table.

This action highlights the station. When a station is selected, its standard output appears, and all enabled right-side buttons that apply to it.

- 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. For details on included menu commands.

- To open a Workbench (Fox) connection to a running station in the current tab, double-click the station row in the table.

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

- To open a Workbench (Fox) connection to a running station 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.

Application output

The largest area in the **Application Director** view shows the “standard output / standard error” output text for the selected station, as shown below.

Figure 10. Station output in Application Director’s application output area

Connected to 00000000					
Name	Type	Status	Details	Auto-Start	Restart on Failure
J8000AX_demo	station	Running	fox=1911,foxs=4911,http=80,https=443	false	true


```

MESSAGE [10:02:52 01-Mar-16 EST][sys.registry] Loaded [389ms]
MESSAGE [10:03:07 01-Mar-16 EST][sys] Baja runtime booted ("/ffs0/niagara") on Qnx-III
MESSAGE [10:03:07 01-Mar-16 EST][sys] Loading "/ffs0/niagara/stations/J8000AX_demo/con
MESSAGE [10:03:24 01-Mar-16 EST][sys] Loaded [17025ms]
MESSAGE [10:03:30 01-Mar-16 EST][alarm.database] Loading...
MESSAGE [10:03:30 01-Mar-16 EST][alarm.database] Loaded [0ms, 0 alarms, 0 pages]
WARNING [10:03:30 01-Mar-16 EST][platDataRecovery.service] Recovery data detected, rep
MESSAGE [10:03:32 01-Mar-16 EST][sys] DataRecoveryService restoration check complete (
MESSAGE [10:03:33 01-Mar-16 EST][sys] Services Initialized [1489ms]
MESSAGE [10:03:33 01-Mar-16 EST][sys.mixin] Updated [69ms]
MESSAGE [10:03:34 01-Mar-16 EST][sys.mixin] Updated [47ms]
MESSAGE [10:03:35 01-Mar-16 EST][history.db] Starting async warmup of history config i
MESSAGE [10:03:35 01-Mar-16 EST][history.db] Async history config index warmup complet
MESSAGE [10:03:36 01-Mar-16 EST][web.server] HTTP server started on port [80]
MESSAGE [10:03:36 01-Mar-16 EST][fox] FOX server started on port [1911]
MESSAGE [10:03:37 01-Mar-16 EST][sys] Niagara Runtime Environment: 3.8.106
MESSAGE [10:03:37 01-Mar-16 EST][sys] *** Station Started (3684ms) [48965ms total] ***
niagara>MESSAGE [10:03:38 01-Mar-16 EST][sys] Saving station...
MESSAGE [10:03:43 01-Mar-16 EST][history.db] Saved history archive (699ms)
MESSAGE [10:03:44 01-Mar-16 EST][sys] Saved /ffs0/niagara/stations/J8000AX_demo/config

```

Depending on the status of the station selected, the standard output text is one of the following:

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

NOTE: You can use the Windows copy shortcut (Ctrl + C) to copy output text to the clipboard. As needed, use scroll bars to view all text, and use the right-side output control buttons. One of these lets you stream station output to a file. For more details, see [“Output control buttons”](#), page 28.

The following sections provide more details related to a station’s standard output:

- Standard output overview
- Station log levels (spy:/logSetup)
- Station LogHistory (LogHistoryService)

Standard output overview

Station output log messages can include errors and warnings that let you 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—see [“Station log levels”](#).

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

- MESSAGE

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.

- **WARNING**

Informs you of a potential problem, such as inability to open a specific port. Typically, warnings do not keep a station from starting.

- **ERROR**

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.

- **TRACE**

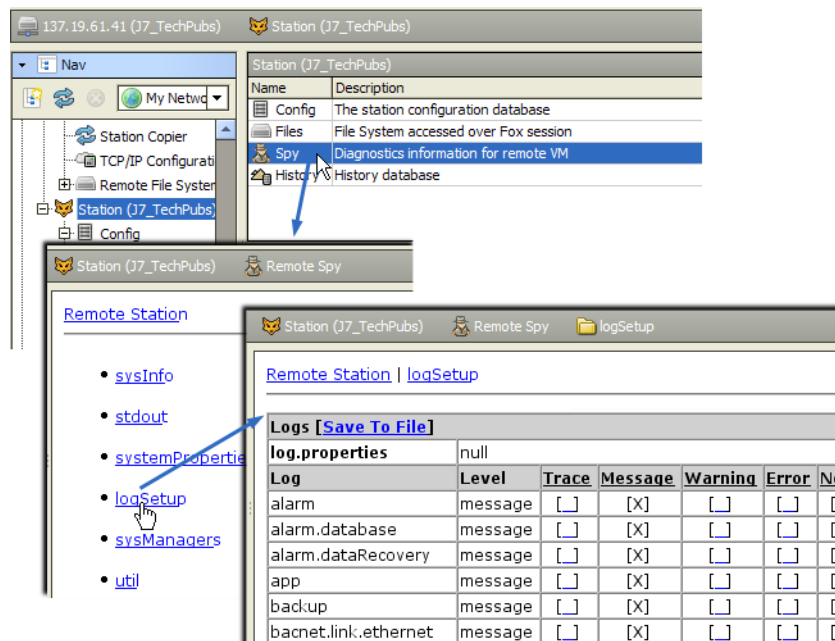
A verbose debug-level message that may be generated upon every process transaction. Typically this is useful only in advanced debugging mode. You see these for station processes only if you have set the log level at "Trace".

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

Station log levels (spy:/logSetup)

A running station is a combination of many ongoing processes. Using the station's spy "logSetup", you can change the "log level" of the station processes of interest in order to "tune" station output. To access the station's logSetup page in 8, double-click the running station in the Nav tree for its **Station Summary** view. From there, double-click **Spy**, then click **logSetup**.

Figure 11. Station spy logSetup (from Station Summary)



By default, all station processes have a "Message" log level (level selection denoted by [X]). To change the level of any listed process, click in the desired level column.

Level selection columns are ordered left-to-right in *increasing* order of message volume.

- **Trace** — Returns all message activity (verbose). This includes all transactional messages, which may result in too many messages to be useful. Be cautious about using Trace!

- Message (Default) — Returns informational "MESSAGE"s, plus all "ERROR" and "WARNING" types.
- Warning — Returns only "ERROR" and "WARNING" type messages (no informational "MESSAGE"s).
- Error — Returns only "ERROR" type messages (no "WARNING" or informational "MESSAGE"s).
- None — No messages are returned to the station's output.

CAUTION: Increasing station output by assigning trace levels consumes extra station resources and exacts a performance penalty! After troubleshooting, return log levels to default values.

Station LogHistory (LogHistoryService)

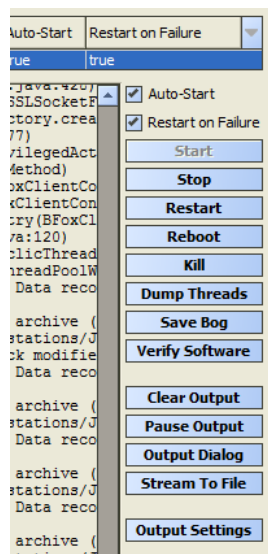
If a station is configured with the LogHistoryService (under its Services container), it maintains a buffered history ("LogHistory") of *some* of the messages seen in the station's standard output. In the LogHistoryService's configuration, you specify its log level, meaning the minimum message type (from station output) to log. By default, the log level (property "Minimum Severity") is `Error`. You may wish to change this to `Warning`.

This is mentioned because when looking at a station's output, you are usually troubleshooting. As part of troubleshooting, you should always check the station's histories for the LogHistory. It should contain recently recorded station errors and (if configured) warnings. This information may help when evaluating "live" output from the station.

Start-up options

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 12. Application Director start-up options and buttons



Option or button	Value	Description
Auto-start	check box	<p>Specifies whether the station starts following platform daemon startup. A station restart occurs:</p> <ul style="list-style-type: none"> • 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) <p>A station restart may or may not follow the installation of new modules using the Software Manager—say, for a new driver. If a station restart is required for a module to become effective, a reboot is prompted.</p>
Restart on Failure	check box	<p>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).</p> <p>In Niagara 4, controllers can have a station restart <i>without</i> a reboot. Therefore, if this option is enabled, and the station fails (terminates with error), the station is restarted.</p> <p>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.</p>
Start	Button enabled if the selected station has an <code>Idle</code>	When pressed, the host's platform daemon immediately starts the station, clears

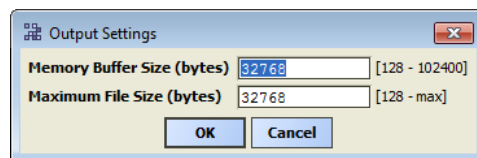
Option or button	Value	Description
	or Failed status in the installed applications area.	the text in the station output, and displays messages for the new station.
Stop	Button enabled if the selected station has a Running status in the installed applications area.	When pressed, opens a pop-up confirmation window. If you confirm, the host's platform daemon shuts the station down <i>gracefully</i> (saving configuration to its config.bog file, and potentially saving history data).
Restart	Button enabled if the selected station has a Running status in the installed applications area.	When pressed, opens a pop-up confirmation window. If you confirm, the host's platform daemon shuts the station down gracefully, then restarts it.
Reboot	Button always enabled.	When pressed, opens a pop-up confirmation window. If you confirm, reboots the selected host. This is considered a drastic action.
Kill	Button enabled only if the selected station has a status of Starting, Stopping, or Running the installed applications area.	When pressed, opens a pop-up confirmation window. If you confirm, the host's platform daemon terminates the station process immediately. Always use Stop instead of Kill , unless unavailable (stuck for a long time as either Starting or Stopping). Unlike a station stop, a station kill does not cause the station to save its database (config.bog), histories or alarms, nor does it update the station output area.
Dump Threads	Button enabled only if the station has a Running status in the installed applications area.	When pressed, the host's platform daemon has the station send a VM thread dump to its station output.
Save Bog	Button enabled only if the station has a Running status in the installed application area.	When pressed, the host's platform daemon has the station locally save its configuration to config.bog.
Verify Software	Button enabled regardless of station status.	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

Option or button	Value	Description
		<p>any other software upon which they depend) are installed. If available in your Workbench installation, any missing software is listed in a popup window, the window offers to install the missing software into the remote host.</p> <p>Only modules (or versions of modules) needed by the station are installed that do not require commissioning. If the station needs modules that require commissioning, meaning an upgrade of core software, those modules are not copied.</p>
Clear Output	Button enabled only if the station has a Running status in the installed application area.	<p>When pressed, the button toggles to Load Output, and the next press toggles back to Pause Output (and so on).</p> <ul style="list-style-type: none"> During a paused output, text remains frozen in the standard output area. This is useful when the station is rapidly writing output. When you press Load Output, text in the station output area is reloaded with the station's buffered output, and output remains updating in real time.
Output Dialog	Button enabled regardless of station status.	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.

Option or button	Value	Description
		NOTE: You may find this compact version of a station's standard output easier to work with than in the main area of the Application Director . Also, if needed you can open multiple output dialogs for comparison purposes.
Stream To File	Button	Opens a window for assigning a file name. Once open, the system saves all application output to this file.
Output Settings	additional properties	Opens a window for specifying how the platform daemon buffers the output from the station.

Output Settings window

Figure 13. Output Settings dialog



NOTE: Changes to either output setting may clear the output buffer's contents.

Property	Value	Description
Memory Buffer Size	number	Defines the size of the memory buffer for the station output. If the station creates more output than the size of the memory buffer, the oldest output is lost.
Maximum File Size	number	When a station stops, its output buffer is written to a console.txt file. This setting defines the maximum size of that file.

Start checkboxes

For the currently selected station in the **Application Director**, you can enable (check) or disable (clear) two start settings using checkboxes, as shown above. Typically, for any JACE station you *enable both* checkboxes. In certain troubleshooting scenarios, you may clear **Restart on Failure** in order keep the station from constantly restarting after successive failures.

NOTE: Changes reflect in the corresponding column of the Application Director’s “installed applications” area.

The two start settings for a station are as follows:

- Auto-start

Specifies whether the station starts following platform daemon startup. This means following a host reboot, perhaps as a result of a power cycle, but possibly from a **Reboot** command.

NOTE: For any JACE controller, a reboot also occurs following any installed dist file(s), as well any TCP/IP-related changes. However, installing new modules from the Software Manager—say, for a new driver, does not always result in a reboot (yet in a few cases, in order for a module to become effective, a reboot may be required—and so is prompted following the module install). At the same time, note that *changing* any existing module (upgrading or downgrading) *always* results in a reboot.

- Restart on Failure

Specifies whether the platform daemon restarts the station if its process exits with a non-zero return code (e.g., engine watchdog had killed the station because of a “deadlock” condition).

NOTE: JACE controllers cannot have a station restart without a reboot. Therefore, if this setting is enabled on such a JACE, if the station fails (terminates with error), the controller reboots. If the controller continues to have 3 “automatic reboots” like this within an hour (or however many specified in the station’s PlatformService “Failure Reboot Limit” property), it remains in a “Failed” state, regardless of the setting above. For related details, see configuration.

Application control buttons

For the selected station in the **Application Director**, application control buttons apply as follows:

NOTE: Be careful about using station controls, and understand the difference between them before using them.

- Start

Enabled if the selected station has an “Idle” or “Failed” status in the “installed applications” area. When pressed, that host’s platform daemon immediately starts that station. Text in the “station output” area is cleared, and output messages begin with the new startup of that station.

NOTE: If you manually stop a station on a AX JACE (using Stop button), it has a status of “Halted.” In this case, the Start button will not be available. You must Reboot the platform to restart the station. This differs from a manually stopped station on a Windows-based host or Linux-based Supervisor, which then shows a status “Idle.”

- Stop

Enabled if the selected station has a “Running” status in the “installed applications” area. When pressed, a popup confirmation appears. If you confirm, the host’s platform daemon

shuts the station down gracefully (saving configuration to its config.bog file, and potentially saving history data).

- Restart

Enabled if the selected station has a “Running” status in the “installed applications” area. When pressed, a popup confirmation dialog appears. If you confirm, the host’s platform daemon shuts the station down gracefully, then restarts it.

- Reboot

Always enabled. When pressed, a popup confirmation dialog appears. If you confirm, the selected host is rebooted. This is considered a drastic action. For details, see [“Reboot”](#).

- Kill

Enabled only if the selected station has a status of “Starting”, “Stopping”, or “Running” the “installed applications” area. When pressed, a popup confirmation dialog appears. If you confirm, the host’s platform daemon terminates the station process immediately.

Always use **Stop** instead of **Kill**, unless unavailable (stuck for a long time as either “Starting” or “Stopping”). Unlike a station stop, a station kill does not cause the station to save its database (config.bog), histories or alarms, nor does it update the “station output” area.

- Dump Threads

Enabled only if the station has a “Running” status in the “installed applications” area. When pressed, the host’s platform daemon has the station send a VM thread dump to its station output.

- Save Bog

Enabled only if the station has a “Running” status in the “installed applications” area. When pressed, the host’s platform daemon has the station locally save its configuration to config.bog.

- Verify Software

Enabled regardless of station status. When pressed, Workbench parses the station’s config.bog file and the host’s platform.bog file, looking for module references. Workbench then checks to see if those modules (and any other software upon which they depend) are installed. Any missing software is listed in a popup dialog, and if available in your Workbench installation, the dialog offers to install the missing software into the remote host.

Only modules (or versions of modules) needed by the station are installed that do not require commissioning. If the station needs modules that require commissioning, meaning an upgrade of core Niagara software, those modules are not copied.

Output control buttons

For a selected station in the **Application Director**, output control buttons are as follows:

- Clear Output

Enabled regardless of station status. When pressed, all text in the “standard output” area is cleared.

NOTE: Data in standard output area is fetched from a memory buffer in the platform daemon. Clearing the output does not actually clear the daemon’s buffer. Therefore, if you change the selection away from, and then back to the station, it re-fetches all buffered data.

- Pause Output

Enabled if the selected station has a “Running” status. When pressed, the button toggles to **Load Output**, and the next press back toggles back to **Pause Output** (and so on).

- During a paused output, text remains frozen in the “standard output” area. This is useful when the station is rapidly writing output.
- When you press **Load Output**, text in the “station output” area is reloaded with the station’s buffered output, and output remains updating in real time.
- 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.

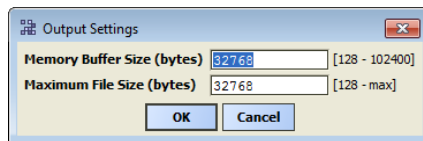
NOTE: You may find this compact version of a station’s standard output easier to work with than in the main area of the **Application Director**. Also, if needed you can open multiple output dialogs for comparison purposes.

- Stream To File
- Opens a dialog for assigning a file name. Once this file is opened, the system saves all application output to this file.

Output Settings

For the selected station in the **Application Director**, the **Output Settings** button produces a dialog in which you specify how the platform daemon buffers the output from that station.

Figure 14. Output Settings dialog



The two available settings are in bytes, and are:

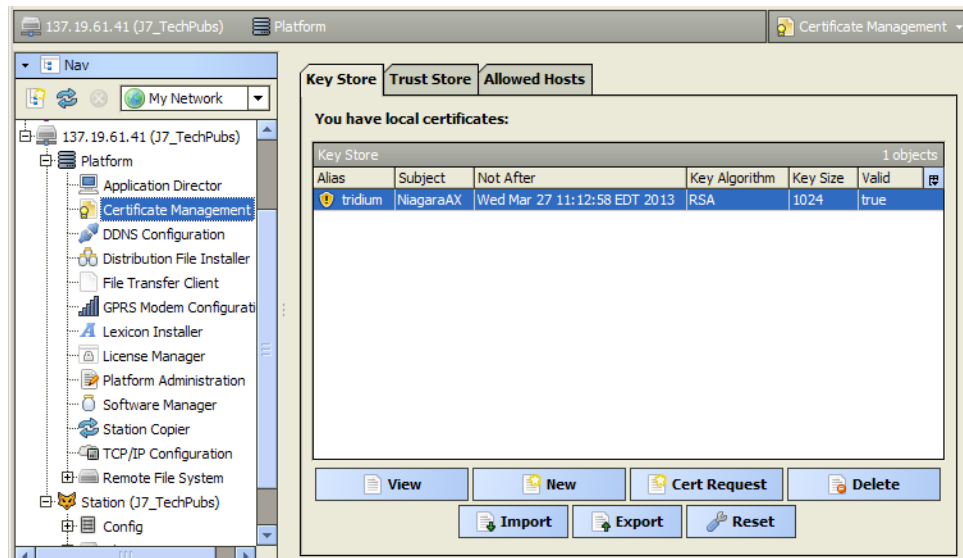
- Memory Buffer Size
- Size of the memory buffer for the station output. If the station creates more output than the size of the memory buffer, the oldest output is lost.
- Maximum File Size
- When the station stops, its output buffer is written to a console.txt file. This setting defines the maximum size of that file.

NOTE: Changes to either output setting may result in the output buffer’s contents to be cleared.

Certificate Management

Starting in AX-3.7, Certificate Management is one of several available platform views. This view appears only if the platform-connected host is licensed for SSL/TLS (feature “crypto”), and has the necessary modules installed, including platCrypto.

Refer to the document *NiagaraAX SSL Connectivity Guide* for complete details. The rest of this section provides overview level information.

Figure 15. Certificate Management platform view

You use this view to manage PKI (Public Key Infrastructure) digital certificates or “self-signed” digital certificates on the platform. Certificates are used in any secure (TLS) connections to this host.

Following the certificate management portion of configuration, secure connections can be enabled (and/or *required*) for any of the following connection types:

- Platform connection from Workbench (client) to the platform’s (JACE or Supervisor) Niagara platform daemon (server), also known as “niagarad”. A secure platform connection is sometimes referred to as “platformssl”. You enable this in the Platform Administration view of the platform. See [“Platform Administration”](#).
- Fox (station) connection from a Workbench client or via Web Workbench. You enable this in properties of the Fox Service in the station. The **FoxService** is typically located as a child container slot under the station’s NiagaraNetwork (Config > Drivers > NiagaraNetwork). For related details see “About the Fox Service” in the *NiagaraAX Drivers Guide*.
- Any browser (HTTP) connection to the station’s web server. You enable this in properties of the station’s **WebService**, also found in its Config > Services container.
- Client connections to the station’s email server (**EmailService**), if applicable. You enable this in properties of the station’s EmailService, typically found in the station’s Config > Services container.

The following sections provide a few basic details about the different tabs in the **Certificate Management** platform view:

- [User Key Store](#)
- [System Trust Store](#)
- [User Trust Store](#)
- [Allowed Hosts](#)

Key Store

The Key Store tab in the **Certificate Management** view lists all installed CA certificates and/or self-signed certificates which are unique to the currently opened platform.

For details, refer to “About the Key Store tab” in the *NiagaraAX SSL Connectivity Guide*.

Trust Store

The Trust Store tab in the **Certificate Management** view lists installed signed certificates by trusted entities that you have imported (your own certificates).

For more details, refer to “Trust Store tab” in the *NiagaraAX SSL Connectivity Guide*.

Allowed Hosts

The User Trust Store tab in the **Certificate Management** view lists all approved hosts (with connection port) and associated self-signed certificates in the currently opened platform.

For details, refer to “About the Allowed Hosts tab” in the *NiagaraAX SSL Connectivity Guide*.

DDNS Configuration

In late 2012, the company Dyn acquired TZO (Tzolkin Corporation), the sole DDNS provider that this NiagaraAX DDNS client was developed against. At the time of this document update, Dyn is not signing up any new accounts for the TZO service, although TZO servers may be working for existing accounts. Unless (or until) other DDNS provider options are available when using this platform configuration view, JACE controllers that require Internet connectivity using DDNS may be best served by installing on a LAN with a router capable of DDNS, and working through some other DDNS provider. The next update to this document will mention if related change have been made to this platform view.

NOTE: The DDNS Configuration view is not available in a JACE-8000 platform running AX-3.8U1.

DDNS Configuration is one of several platform views. DDNS (Dynamic Domain Name System) allows for DNS IP addresses to be dynamically updated. Typically, these are DHCP (Domain Host Configuration Protocol) addresses (Internet or Intranet).

Figure 16. DDNS Configuration platform view

Refer to the Engineering Notes document "NiagaraAX-3.1 DDNS" for more details and examples.

DDNS core configuration items

The platform DDNS Configuration view has the following configuration sections.

Provider

The top property in the DDNS Configuration view is to select from a list of supported DDNS providers. Currently, the only supported provider is Tzo (see About TZO), although future builds may support other providers.

Figure 17. DDNS Provider selection and Provider-supplied configuration properties

The screenshot shows a web-based configuration interface for DDNS. At the top, there are three dropdown menus: 'Select a DDNS Provider' (set to 'Tzo'), 'Select the DDNS Mode' (set to 'Tzo'), and 'Select the Adapter Interface' (set to 'Onboard Ethernet Adapter en0'). Below these is a section titled 'Provider (Ddns Provider)' which contains three input fields: 'Key', 'Email', and 'Domain'. Each field has a radio button next to it, and the 'Key' radio button is selected. At the bottom of the form is a section titled 'Status (Ddns Status)' with a radio button next to it.

NOTE: A DDNS account with a provider is typically a fee-based subscription, in which you must first register and pay before your account is active.

Once you select the DDNS Provider, you require information from that provider about your DDNS account, in order to populate the following properties in the Provider section.

- Key
Furnished by provider (TZO) after account creation.
- Email
Email address associated with the provider (TZO) account.
- Domain
Domain name associated with the DDNS account.

Mode

This property in the DDNS Configuration view selects the operational mode of DDNS.

DDNS mode choices include:

- Disabled
DDNS functionality is completely disabled.
- Internet
Uses the IP address assigned to the adapter specified in the Adapter property.
- Intranet
Uses the IP address as seen when connected to the DDNS provider (note that not all providers will support this).

NOTE: Support for dialup modem connectivity (Mode=Dialup) was dropped starting in AX-3.7.

Adapter

This property in the DDNS Configuration view applies if DDNS Mode is Internet, and selects the adapter to interrogate for an IP address.

About TZO

Formerly, you could create a DDNS account with TZO (Tzolkin Corporation) at <http://www.tzo.com>. However, this company has been acquired by Dyn.

NOTE: The original development and testing of NiagaraAX DDNS Configuration focused primarily with TZO accounts.

Distribution File Installer

The Distribution File Installer is one of several platform views used to install dist (distribution) files. A dist file is a zip that contains other files and a manifest that describes the contents of the distribution. For technical details, see the “Distributions” section in the online *Niagara Developer Guide*.

Use this view for either of these two tasks:

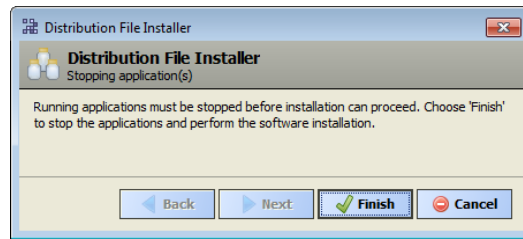
- To *restore* a locally available backup .dist file to a remote controller. Such a restore can be initiated using the **Backup** command from the platform’s Platform Administration view, or more commonly from the **BackupService** in the station running on that host. For details, see [“Restoring a backup dist”](#).

NOTE: In AX-3.7U1, backup .dist file restoration to a host other than the original host may first require manual editing. See Security update 1 changes to backup dist usage. However, in AX-3.8 this sort of editing is unnecessary. For further details, refer to the *NiagaraAX 2013 Security Updates* engineering notes document.

- To install a clean dist file (or conversion dist file). This downgrades a controller to an older release level, or restores it to a known empty state. For example, using a conversion dist file you can downgrade a JACE-8000 controller from Niagara 4 to AX-3.8U1. Following a clean (or conversion) dist install, you must *recommission* the controller, as this wipes the file system (almost all Niagara software, as well as all station files), leaving the controller in an empty near-factory state. For details, see “Wiping clean a JACE (clean dist)”.

NOTE: Do not use this view to upgrade a controller. Instead, use the **Commissioning Wizard** to upgrade Niagara in a controller. The **Commissioning Wizard** is a right-click option on a platform when opened in Workbench. See [“Upgrading a JACE”](#).

CAUTION: If a station is already running on the remote host, any dist file install requires that all applications be stopped, after which *all are invariably overwritten*. After selecting a dist file, the Installer provides a confirmation dialog for this, as shown here. When you finalize the install (click **Finish**), the Installer automatically stops the station, then continues with the distribution file install process. *Before* finalizing any dist installation, make sure that any controlled equipment, which might be adversely affected by the station stopping (and the removal of software) is put in a manually controlled state.

Figure 18. Stop station dialog in Distribution File Installer

The following sections provide more details on the **Distribution File Installer**:

- [Operation of the Distribution File Installer, page 35](#)
- [Restoring a backup dist](#)
- [Wiping clean a JACE \(clean dist\)](#)

Security update 1 changes to backup dist usage

In the NiagaraAX 2013 security update releases (e.g. AX-3.7U1), station password storage changed to become much more secure than in "pre-update" releases. Now, in some cases after making a station backup dist file in AX-3.7U1 (typically using a station's BackupService), you should subsequently edit that dist before restoring it with the Distribution File Installer.

This is especially true if you need to install the same backup dist in multiple hosts, say as a "system image" for a replicated AX-3.8 station. Otherwise (without a dist file edit), system security on those hosts will be compromised. For details, refer to the NiagaraAX 2013 Security Updates engineering notes document.

NOTE: In AX-3.8, improvements in station backups and restorations were made that simplify usage, such that ".dist file edits" are typically unnecessary. See AX-3.8 changes to backup dist usage in this document.

Note that in any case where you want to restore a backup dist to the identical host (JACE) from which you made the backup, no edits are needed beforehand. In this scenario, operation remains as before, and security remains uncompromised.

AX-3.8 changes to backup dist usage

AX-3.8 includes all station security improvements in earlier 2013 security update releases (e.g. AX-3.7U1), along with additional platform security improvements. For related details, see Improvements to AX-3.8 digest authentication.

Improvements were also made when restoring AX-3.8 station backups with the platform Distribution File Installer, or when using the Station Copier with copied AX-3.8 stations. Unlike in the prior update releases, AX-3.8 station backup .dist files and config.bog (station copy) files are "more portable", such that editing .dist files is typically unnecessary. And you can use the Station Copier to install a station copied from one AX-3.8 host to another without any "client password" re-entry or other editing.

However, note the following change in a station backup .dist made from a AX-3.8 host:

- Platform credentials are no longer included in a station backup .dist of the AX-3.8 host (unlike in a backup .dist file for any earlier release).

In some cases this may result in confusion when restoring a station backup made from the AX-3.8 host, as this varies from restoring a station backup made from a host running AX-3.7U1 (or earlier). Note that when restoring a station backup made from an AX-3.8 host:

- If the host is running AX-3.8 prior to installing the backup .dist file, it will reboot from the restore with the same platform credentials that it had before (no change in platform credentials).
- If the host is running any earlier release prior to installing the backup .dist file, it will reboot from the restore with factory default platform credentials.

Note this always occurs if you install a "clean dist" file in the host (JACE) before installing an AX-3.8 station backup—even if you assign interim "non-default" platform credentials to the JACE first.

Operation of the Distribution File Installer

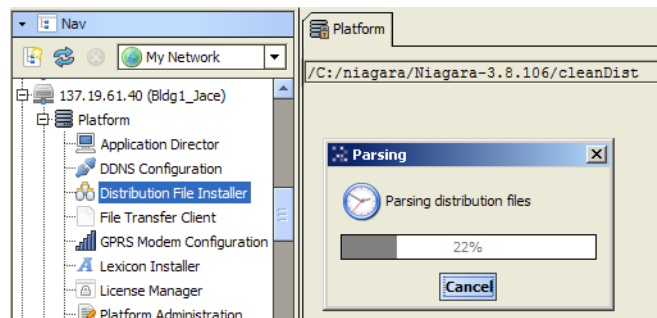
The following subsections explain more about dist file selection and the install process:

- [Dist file selection](#)
- [Distribution file install process](#)

Dist file selection

When you select the Distribution File Installer, it “parses” through the dist files on your local PC, using the last source folder selected. See the figure below.

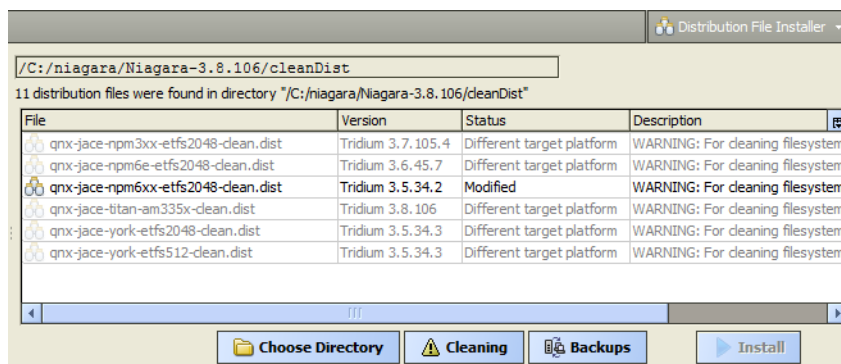
Figure 19. Parsing dialog in Distribution File Installer



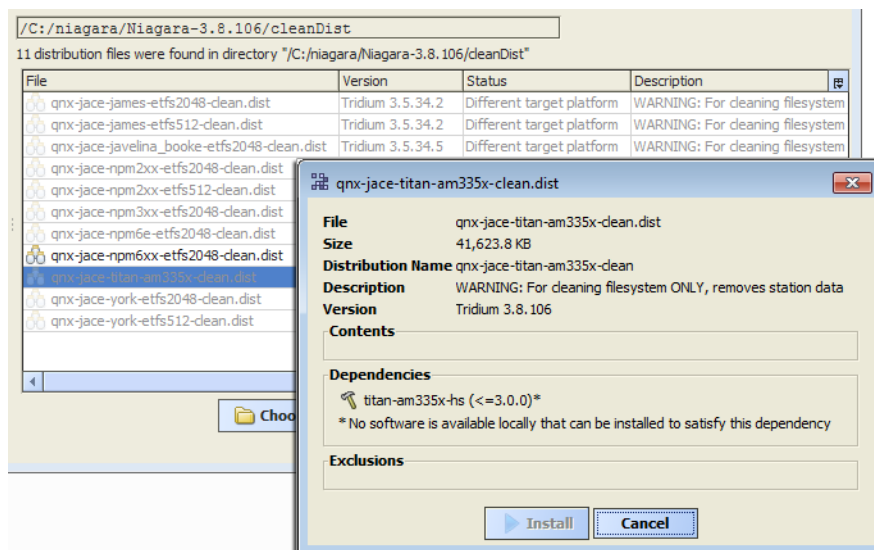
By default, the first time you use the Installer, the “backup” folder under the Niagara software location (!\backups) is parsed. If that folder does not exist yet (no backups have been made), then the “cleanDist” folder (!\cleanDist) is parsed instead.

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 for a **Change Directory** dialog, and point the Installer to that location.

When parsing completes, a table of the found dist files appears, with the appropriate dist files available for selection in the table, as shown below (using default software location).

Figure 20. Available dist files in Distribution File Installer

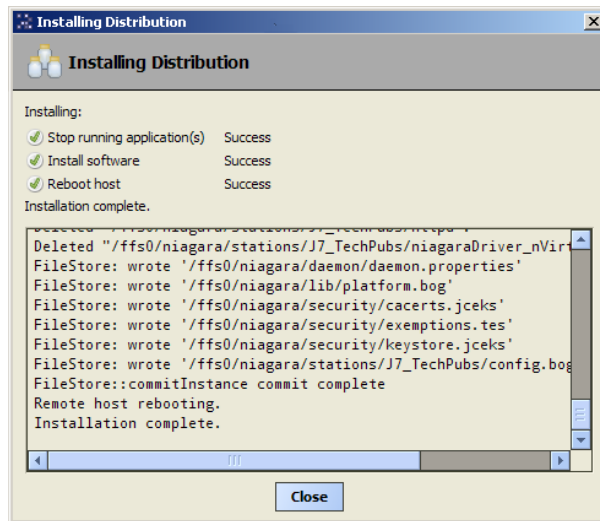
Dist files that are inappropriate, for example that are for a different target platform or have unmet dependencies, are *dimmed*—the **Install** button does not become active if you select such a file. For details on any dist file, double-click it for a popup, including a list of its dependencies, as shown below.

Figure 21. Example details dialog for a dist file, showing dependencies.

To be able to restore a backup dist file, your Workbench installation requires the same versions of software, including modules, to be available in its "software database." Therefore, it is recommended that you make and keep frequent backups as you upgrade a JACE.

Distribution file install process

After proceeding with **Finish**, the dist installation process appears in a dialog that tracks its progress as it continues, as shown below.

Figure 22. Distribution File Installer progress dialog

After the distribution file (and modules, if selected) are installed on the JACE platform, the JACE is rebooted, and the progress dialog indicates complete. You must click the **Close** button to continue. You can then reopen a platform connection, perhaps to view output in the **Application Director**.

About backup dist files

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

Typically, station backups are done from Workbench *station connections* (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, station backup dist files are saved in the “backup” folder under the Niagara software location (!\backups).

Restoring a station backup

The following procedure describes restoring a backup for a JACE controller.


NOTE: To be able to restore a backup dist file, your Workbench installation requires the same versions of software, including modules, to be available in its “software database.” Therefore, it is recommended that you make and keep frequent backups as you upgrade controllers. Also for this reason, you may need to import the software database from prior revisions of Niagara into your current Workbench installation. For related details, see [“About your software database”](#)

Restoring a backup dist

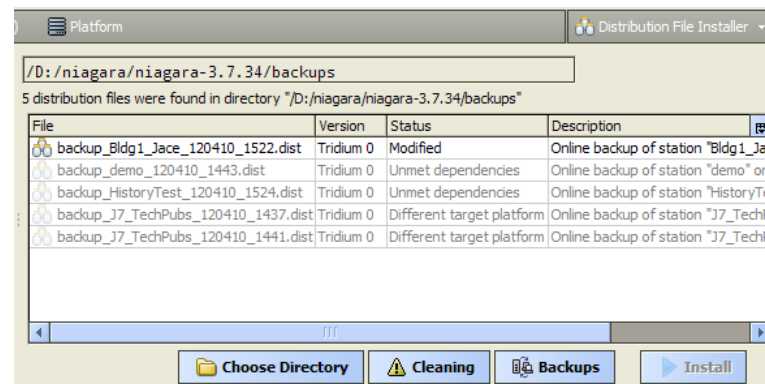
Prerequisites:

- Station backup dist file for the target JACE controller.
- The software database of your Niagara installation must include matching versions of all software modules used by the station when the station backup was made, or else the backup restore will fail.

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

- Step 2 In the **Distribution File Installer** click the  **Backups** button for the !/backups folder. or if needed, click the **Choose Directory** button to point to another backup dist file location.

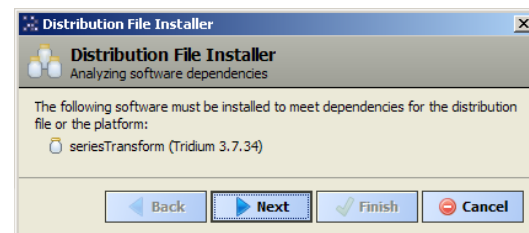
The Installer parses through the dist files, and makes selectable only those files that are compatible with the opened JACE platform. When done parsing, available backup dists appear listed, as shown below.



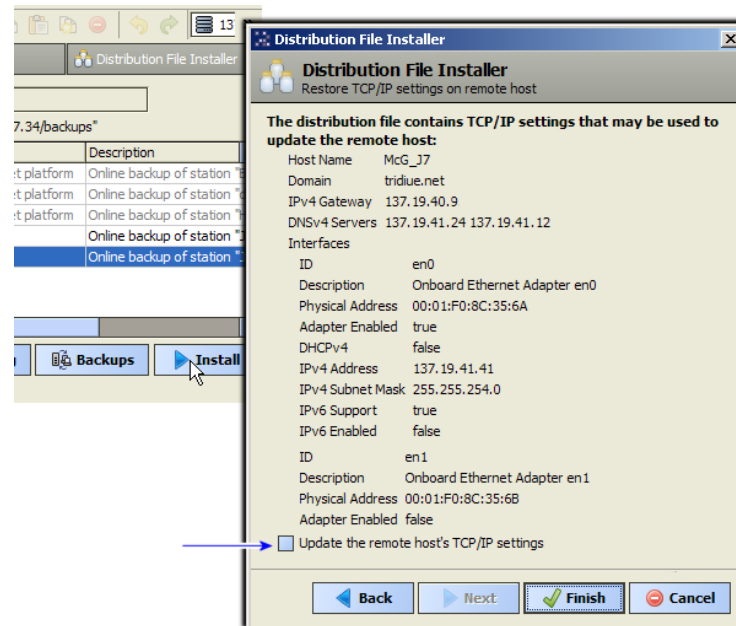
- Step 3 If needed, you can double-click any .dist for more details in a popup dialog.
- Step 4 To restore any selected backup, click **Install**.

Proceeding with the Install, if the host is already running a station, a dialog appears telling you that it must be stopped.

- Step 5 If the station backup .dist file contains software modules different from (or in addition to) those already installed in the remote host, another dialog appears to inform you, as shown below.



- Step 6 Click **Next** to continue.
- Step 7 As shown below, when restoring a backup, another dialog always asks if you wish to re-store the TCP/IP settings stored in the dist file (as displayed) into the remote host.



TCP/IP settings contained in the dist file are listed, and by default, the checkbox “Update the remote host’s TCP/IP settings” is unselected.

- Step 8 In the **Update the remote host’s TCP/IP settings** checkbox, do one of the following:
- Leave this checkbox unselected to retain the current TCP/IP settings in the remote host.
 - Click the checkbox to replace the TCP/IP settings in the remote host with those stored in the dist file—meaning the same ones shown in this dialog.

Step 9 Click **Finish** to begin installation.

See [“Distribution file install process”](#).

Wiping clean a JACE (clean dist)

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

Wiping clean a JACE is typically unnecessary if upgrading an operational controller to a later Niagara software build—simply using the **Commissioning Wizard** should be all that is necessary. However, if *downgrading* a JACE to an earlier release build, 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.

NOTE: A conversion (clean) dist file is available to downgrade N4 JACE-8000 controllers to run AX-3.8U1. A requirement for this is adding the JACE-8000-AX license feature to the JACE-8000 license.

See the following for more details on JACE clean dist files:

- [“Clean dist installation preparation”](#)
- [“Installing a clean dist file”](#)

Clean dist installation preparation

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

- TCP/IP settings
- license files
- brand.properties
- most secure communication (SSL/TLS) configuration

All other data is deleted from the file system, including station bog files, Px files, modules, etc. Note that the unit's SSL/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).

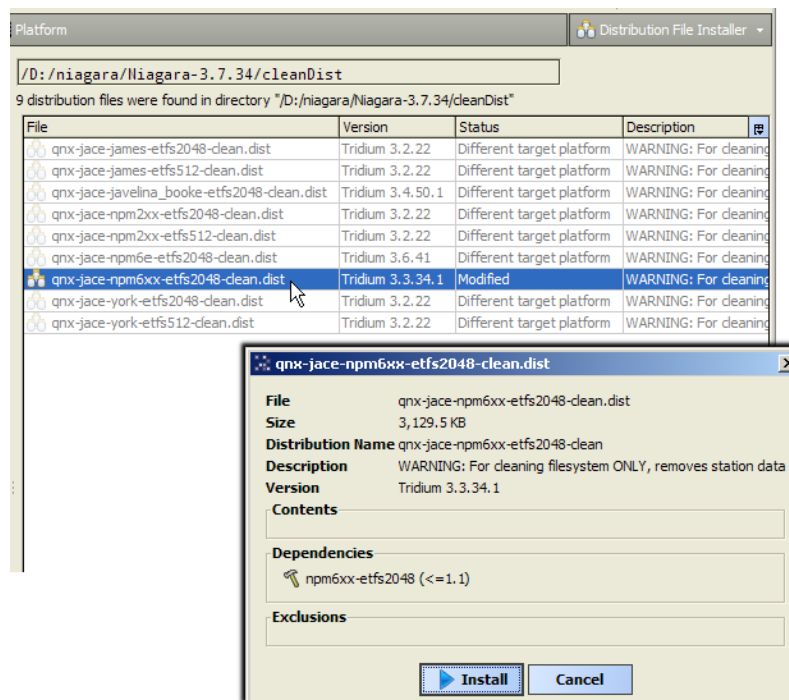
For any JACE configured for platform "Ssl Only", you should change this State to "Disabled" before installing a clean dist file. Otherwise, after the JACE reboots from the clean dist installation, a platform connection fails. See .

NOTE: Cleaning also deletes the !security folder, which in 2013 update releases (e.g. AX-3.7u1) contains files used for encryption of "client passwords" in any hosted station. If cleaned, any previous station save (config.bog) for that AX-3.7u1 host will have non-working client passwords when re-installed. This happens because an AX-3.7u1 host will generate new (and different) key files when it first initializes. In this case, you will need to re-enter all such passwords (e.g. the Password in the ClientConnection container of a NiagaraStation, or the Password in an OutgoingAccount under the EmailService), to have it work again. For related details in this document, see [Security update 1 changes to backup dist usage, page 34](#)

Therefore before installing a clean dist file, make sure to backup station files plus any other items on the controller you wish to keep. You should always backup (export) certificate keys of any SSL/TLS-configured unit, such that if the controller needed to be replaced (hardware swap-out), you could re-import those keys.

Note that *after* installing a clean dist, you must always recommission the controller for Niagara, using the **Commissioning Wizard**. For related details, see "About the Commissioning Wizard" in the *JACE NiagaraAX Install & Startup Guide*.

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

Figure 23. Clean dist file shown selected in the Distribution File Installer

Clean dist files appear listed with a “WARNING” in the Description, as shown in the one selected in the figure above. Only the appropriate one for the currently opened platform will be selectable.

Running clean dist on a controller

Prerequisites:

- The JACE controller is a Niagara unit.
- You have backed up any station files as well as any other files needed later, for example digital certificate keys. See [“Clean dist installation preparation”](#).

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

Step 2 Open the **Distribution File Installer** and click the **Cleaning** button to access the !cleanDist directory.

Step 3 Select the appropriate clean dist file for the platform and install.

The file system clean will take a few minutes, then the controller will automatically reboot. Wait for the reboot to complete.

NOTE: After reboot from a clean dist install, the controller is using default platform credentials and port (3011).

To re-install the software versions to the controller:

- Using a version of Workbench that runs the same software versions that you want on the controller, use the platform **Commissioning Wizard** to install the desired software build. For details, refer to the section “About the Commissioning Wizard” in the *JACE NiagaraAX Install & Startup Guide*.

- b. If you have a backup dist file for the controller that was made when it had the desired prior software versions, use the **Distribution File Installer** to install it. See the previous section, “[Restoring a backup dist](#)”.

Upgrading a controller

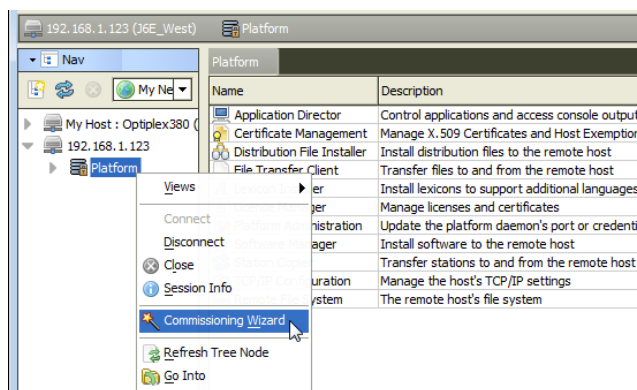
You must use the **Commissioning Wizard** to upgrade Niagara software in a JACE. This means either an “update” upgrade (say from build 3.7.44 to 3.7.106), or a full “minor” release upgrade, for example build 3.7.106 to build 3.8.107.

NOTE: When updating a multi-station system for the first time to an update release, it is recommended to upgrade a Supervisor before its subordinate JACEs.

Any JACE to be upgraded from one minor version to another, say from 3.7.nn to 3.8.nn, requires a *license upgrade*, purchased *before* starting the upgrade. Otherwise, the Commissioning Wizard in Workbench will not perform the upgrade. This prevents the scenario where an upgraded JACE cannot start its station, due to a licensing error.

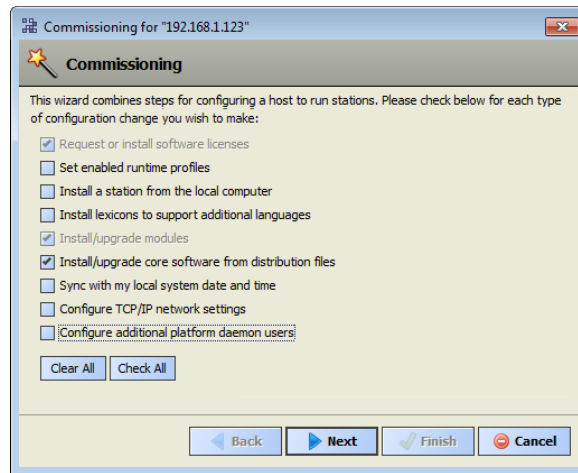
With a platform connection to any Niagara JACE, access the Commissioning Wizard by simply right-clicking on that platform and selecting it from the menu, as shown here.

Figure 24. *Commissioning Wizard is right-click option of opened platform*



If a controller upgrade, in the wizard's opening selection of steps you typically *deselect* most items that were previously run at the controller's initial commissioning time—for example to set enabled runtime profiles, set date and time, configure TCP/IP settings, and so on. See the figure below.

Figure 25. Typical upgrade selections for existing Niagara JACE (already running a station)



To upgrade a Niagara JACE, you *do select*:

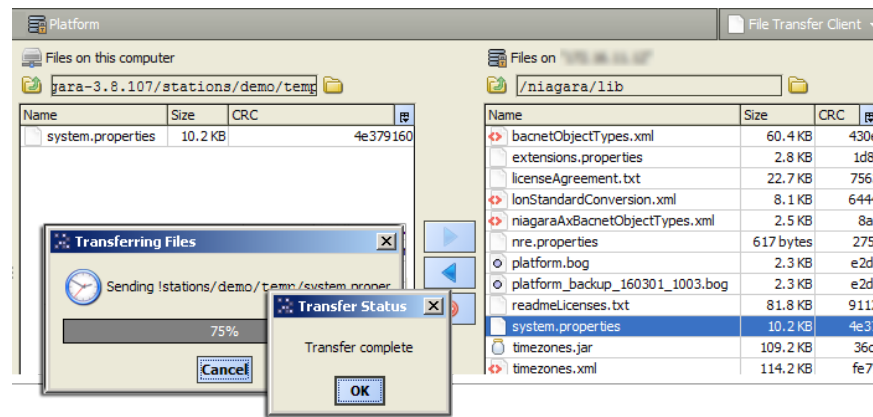
- In the case where the upgrade requires an updated license installed:
“Request or install software licenses” (this may already be pre-selected).
- In the case where a station install also requires commissioning the JACE (i.e. upgrade):
“Install station from the local computer”
- And always:
“Install/upgrade core software from distribution files”

When you proceed in this manner, the wizard automatically finds and selects all core distributions needed for the JACE. Then, in the pre-selected “Install/Upgrade modules” step, the wizard provides the option to also upgrade all out-of-date software modules (always do that).

A final summary step allows you to review the upgrade before the wizard executes and performs its operations. For further details, refer to the section “About the Commissioning Wizard” in the *JACE NiagaraAX Install & Startup Guide*.

File Transfer Client

The File Transfer Client is one of several platform views. It allows you to copy files and/or folders between your Workbench PC and the remote Niagara platform. You can also use it to delete files and folders.

Figure 26. File Transfer Client

This may be useful if you wish to copy graphics images to a controller, as one example. Or, use it to copy a text file *from* a folder on a remote controller (say, `!lib/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 `!lib` folder. An example of this is shown above. Also see “[system.properties notes](#)”.

However, *do not use* the File Transfer Client to copy modules to a JACE, as “runtime profile types” are not applied, nor are module dependencies. As a result, incorrect or missing modules may result. Always use the platform **Software Manager** to install (or uninstall) software modules on a JACE controller.

CAUTION: Be careful when using the File Transfer Client, especially when copying files to a target JACE platform, or whenever using the delete (X) control. Note that in either direction, when transferring a file and an identically-named file already exists, a confirmation dialog appears before the copy. A confirmation dialog also appears before any delete. However, after confirmation there is no “Undo.”

The File Transfer Client provides a two-pane view, as shown.

The left pane provides access to *local* (Workbench PC) files, and the right pane provides access to files on the *remote* platform.

Use is straightforward, you simply click navigation controls at the top of each pane to go to the appropriate location for source and target. Then you click one or more items on one side (as source) to select for copying to the other side (target). Then, you click the appropriate transfer arrow.

A dialog appears when all files are transferred, as shown above.

system.properties notes

Occasionally there may be a need to edit the `system.properties` file used by any station running on a host, which for any NiagaraAX platform is located in its System Home `!lib` folder.

- On a remote JACE (QNX OS), this folder is at `/niagara/lib`.
- On a local Supervisor (Windows OS), this folder is at `C:\niagara\Niagara-3.8.xxx\lib`.

Note the `system.properties` file in a Workbench PC or Supervisor platform is similar to, but different from this same file on a remote JACE platform.

You cannot directly edit in place the `system.properties` file on the JACE. Instead, you must copy it to your PC first (using the platform **File Transfer Client**) to edit.

Only a few entries in a `system.properties` file are typically processed. Most lines in this file are comments, which start with a `#` and are not processed. Comments “inactivate” many entries in this file—and typically these entries should remain inactive. To activate such an entry, you must delete the leading `#` character on that line of code.

CAUTION: Editing (and activating) `system.properties` entries is an operation for advanced users, with the possibility of undesirable results. Read all comment lines carefully, and consult your support channel before making a change! Always save a backup copy of this file, and test after implementing a change.

Editing `system.properties`

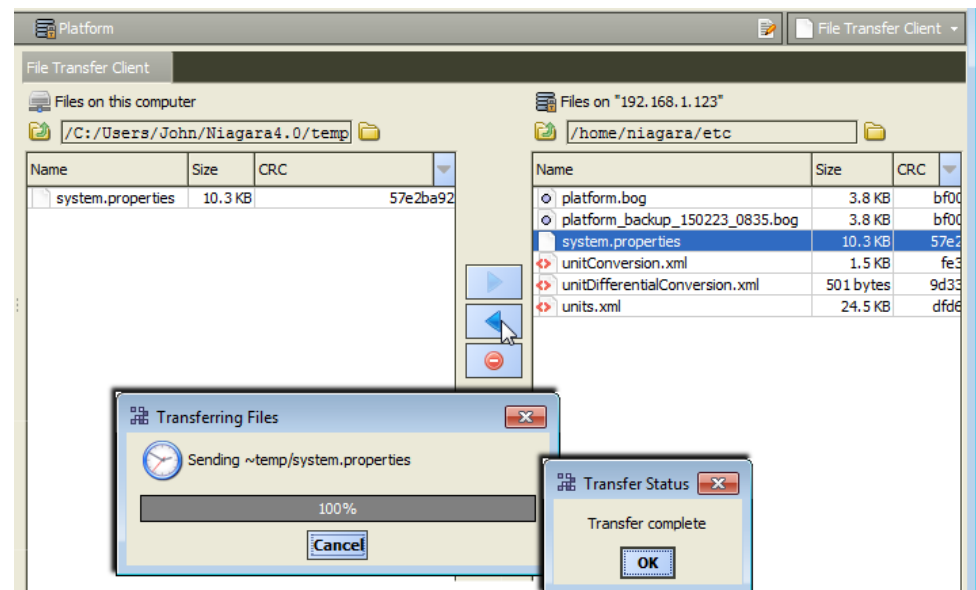
This procedure provides general steps for using the **File Transfer Client** to copy files between a Supervisor PC and a remote host.

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.

Perform the following steps:

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* (Workbench PC) files.
- 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.

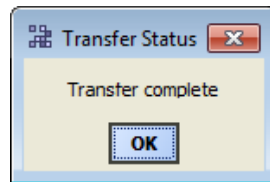
Step 3 Select one or more items on one side (as source) to copy to the other side (target), and click the appropriate transfer arrow.

For local-to-remote transfer of a file containing encrypted, sensitive data, the **File Transfer Client** does not prompt you to enter a passphrase. Instead, the results of the transfer are one of the following:

This message displays when all files are transferred:

- 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 become effective.

- 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 with Workbench to verify operation.

GPRS Modem Configuration

The **GPRS Modem Configuration** view (shown here) is one of several NiagaraAX platform views for a QNX-based JACE. It is used to configure the (wireless) modem option card that may be installed in the host JACE controller.

An equivalent view, the GprsPlatformService, is added under the PlatformServices in the station running on the JACE. Note, this "Gprs Platform Service Plugin" (and GprsPlatformService) appears only if the controller actually has the GPRS modem option card installed—unlike the platform view. Everything in this section about the **GPRS Modem Configuration** view also applies to the **Gprs Platform Service Plugin** view.

NOTE: Although the **GPRS Modem Configuration** view appears if platform-connected to a JACE-4 or JACE-5 series controller, it does not apply to these platforms—in this case, you can safely ignore it.

Figure 27. GPRS Modem Configuration view

NOTE: Refer to the *Engineering Notes* document "GPRS modem option" for complete details, including a reference section covering all properties and fields in this platform **GPRS Modem Configuration** view (this also applies to the equivalent **Gprs Platform Service Plugin** view).

GPRS modem configuration sections

The **GPRS Modem Configuration view** (or the **Gprs Platform Service Plugin** view) has the following configuration sections.

Also, the **Status and Runtime Data** area near the bottom of the view shows data served up from modem.

NOTE: Refer to the section "Required platform GPRS setup" in the *Engineering Notes* document "GPRS modem option" for details on the most important configuration properties in the sections below.

Modem Configuration

This section of the platform GPRS Modem Configuration view includes properties to enable/disable the modem, set debug level and monitor cycle time, plus other properties.

Provider Configuration

This section of the platform GPRS Modem Configuration view is to specify access properties to the wireless provider (corresponding to the SIM card installed in the GPRS modem option),

along with numerous properties related to the PPP (point-to-point) protocol used for GPRS connections.

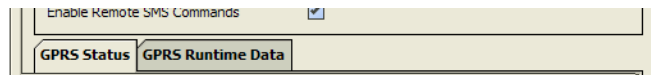
SMS Configuration

This section of the GPRS Modem Configuration view defines the behavior of the SMS (Short Message Service) handling portion of the GPRS modem driver. Properties include whether to delete SMS messages or allow remote commands.

Status and Runtime Data area

The bottom area of the GPRS Modem Configuration view contains debug-level data from the low-level GPRS modem driver ("GPRSD") on the JACE.

Figure 28. Tabs near the bottom of the GPRS Modem Configuration view

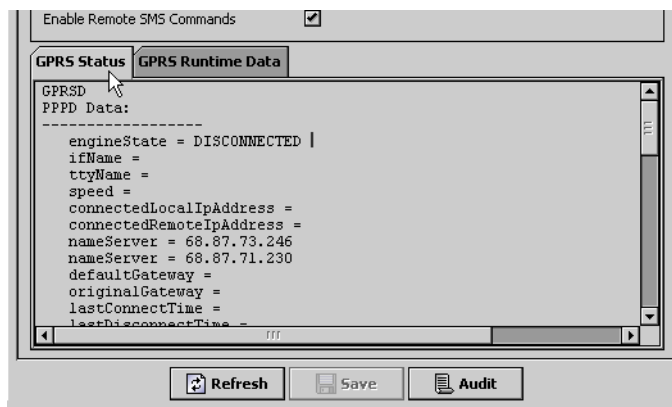


This data is found on two separate tabs: GPRS Status and GPRS Runtime Data.

- Information on these tabs updates only when you load or refresh this view.
- Refer to the section "GPRS Status and Runtime Data tabs" in the *Engineering Notes* document "GPRS modem option" for reference details on the various data sections below.

GPRS Status

Figure 29. GPRS Status tab in GPRS Modem Configuration view (or Gprs Platform Service Plugin)



The **GPRS Status** tab in the platform **GPRS Modem Configuration** view (or **Gprs Platform Service Plugin**) shows data separated into the following categories:

- **PPPD Data** - This section in the GPRS Status tab shows "ppp" (point to point protocol) related data.
- **Modem Data** - This section shows data about the installed GPRS modem option card.
- **SMS Data** - This section shows data about failed SMS messages.
- **Monitor Data** - This section in the GPRS Status tab shows various data, using various terms including ME (mobile equipment), MS (mobile station), and PLMN (public land mobile network).

GPRS Runtime Data

Figure 30. GPRS Runtime Data tab in GPRS Modem Configuration view (or Gprs Platform Service Plugin)

Field	Value	Status
GPRS Status	{ok}	ok
RSSI	2.00	ok
Roaming	false	ok
On Demand PPP	false	ok
IP Address	{ok}	ok
IMSI	310260361595856	ok
IMEI	355633003152351	ok
SIM	8901260360015958568	ok

Buttons: Refresh, Save, Audit

The GPRS Runtime Data tab in the platform GPRS Modem Configuration view (or Gprs Platform Service Plugin) shows status data for the modem, including its RSSI (Received Signal Strength Indicator, from 0 to 5), whether Roaming is active, and its unique IMSI, IMEI and SIM values.

Lexicon Installer

The Lexicon Installer is one of several Niagara platform views. Currently, this view lets you install *file-based* Niagara “lexicon sets” from your Workbench PC to a remote JACE platform, as needed.

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

```
niagaraLexiconLc-rt.jar
```

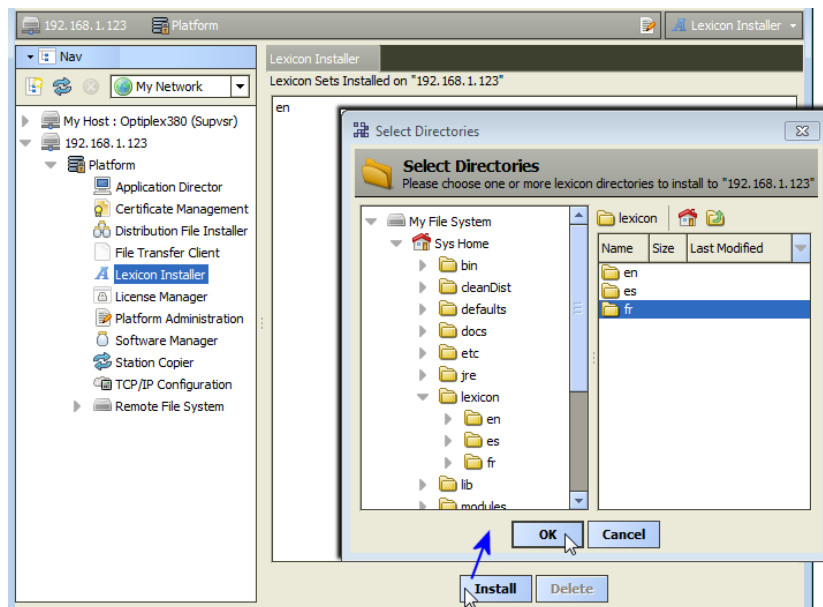
where Lc is the two-character “language code”, such as Fr for French or Es for Spanish. Workbench provides a **Lexicon Tool** with a special “Lexicon Module Maker” view that you can use to modify or make new lexicon modules, from edited text-based lexicon files. For complete details, refer to the *Niagara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

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

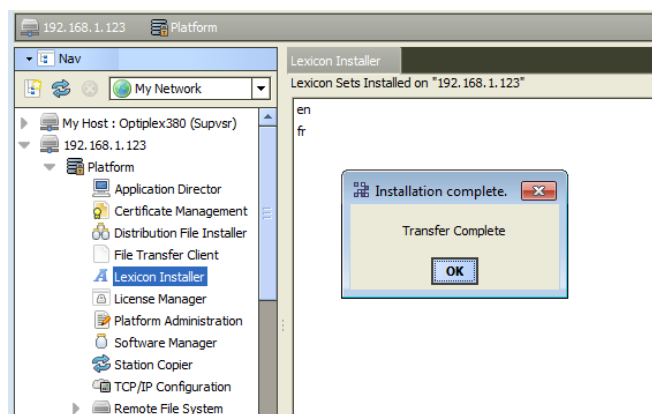
- International locations: For non-English language support
- Domestic (U.S.) locations: where you have modified the English (en) lexicon in order to change the wording used in default labels.

Beforehand, you typically 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.

When you select Lexicon Installer, any existing file-based lexicon sets (already installed in that platform) are listed in the view pane. When you click **Install**, a “Select Directories” dialog appears for you to select lexicon sets (in the lexicon folder under your Workbench **Sys Home**) to install in the remote platform, as shown below.

Figure 31. Lexicon Installer, selecting lexicon

When you click **OK**, the selected lexicon directory or directories are installed in the remote JACE platform. An “Installation Complete” dialog appears when all files are transferred, as shown below.

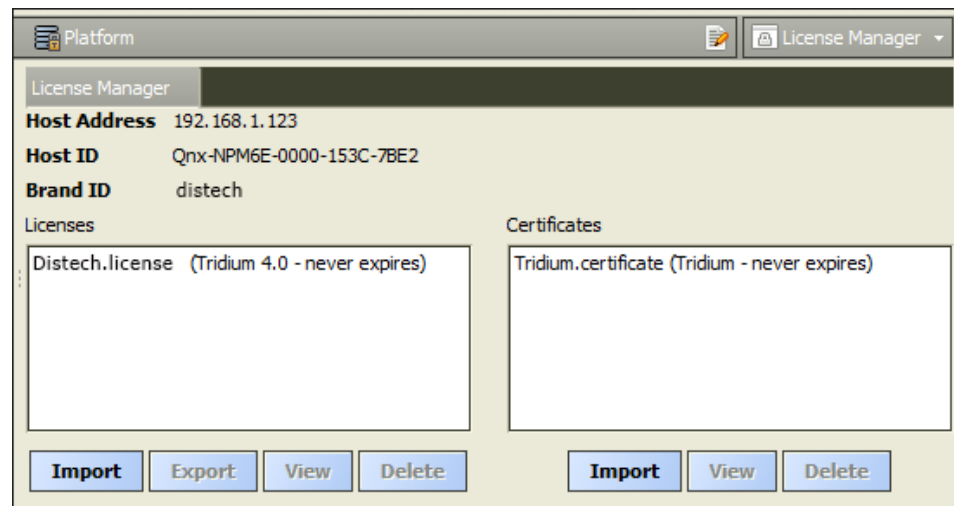
Figure 32. Lexicon Installer, lexicon installed

License Manager

The License Manager lets you install (import) 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 and utilization of a license archive file format. In addition, a Workbench **License Manager** tool is available, which does not require a platform (or station) connection to use. These features are explained in this document, , along with details about the contents (features) of license files.

Figure 33. License Manager lists existing licenses and certificates



The **License Manager** lists any existing licenses and certificates (already installed in the platform). Below the left-hand *license side* of the **License Manager**, are these buttons (commands):

- **Import** installs a *new* license or certificate file. Typically, you import license files from either the online licensing server or from your local license database.
- **Export** saves a license file as a license archive (.lar) file.
- **View** opens an *existing* license file (clicking this button is the same as double-clicking an item).
- **Delete** removes an *existing* license file.

Click a license or certificate to select it, or *double-click* to view it in a window, as shown below.

Figure 34. Viewing a license in License Manager

```

klicense vendor="Tridium" expiration="2015-07-28" hostId="Qnx-NPM6E-0000-153C-7BE2" ser
<feature name="brand" accept.station.in="*" accept.station.out="*" accept.wb.out="*" b
<feature name="about" project="Tridium-Training" owner="Tridium, Inc."/>
<feature name="appFramework" expiration="2015-07-28" app.limit="none"/>
<feature name="axvelocity" expiration="2015-07-28"/>
<feature name="bacnet" expiration="2015-07-28" schedule.limit="none" export="true" poi
<feature name="box" expiration="2015-07-28" session.limit="none"/>
<feature name="crypto" expiration="2015-07-28" ssl="true"/>
<feature name="dataRecovery" expiration="2015-07-28"/>
<feature name="email" expiration="2015-07-28"/>
<feature name="eventService" expiration="2015-07-28"/>
<feature name="jre8qnx" expiration="2015-07-28"/>
<feature name="ldapv3" expiration="2015-07-28" kerberos="true"/>
<feature name="lonworks" expiration="2015-07-28" schedule.limit="none" point.limit="nc
<feature name="mobile" expiration="2015-07-28" history="true" schedule="true" alarm="t
<feature name="modbusAsync" expiration="2015-07-28" schedule.limit="none" point.limit=
<feature name="modbusSlave" expiration="2015-07-28" schedule.limit="none" point.limit=
<feature name="modbusTcp" expiration="2015-07-28" schedule.limit="none" point.limit="r
<feature name="modbusTcpSlave" expiration="2015-07-28" schedule.limit="none" point.lim
<feature name="mstp" expiration="2015-07-28" port.limit="5"/>
<feature name="ndio" expiration="2015-07-28" schedule.limit="none" point.limit="none"
<feature name="niagaraDriver" expiration="2015-07-28" virtual="true" schedule.limit="r
<feature name="nre" expiration="2015-07-28"/>
<feature name="obixDriver" expiration="2015-07-28" schedule.limit="none" export="true"
<feature name="search" expiration="2015-07-28" local="true"/>
<feature name="serial" expiration="2015-07-28"/>
<feature name="station" expiration="2015-07-28" station.limit="500" resource.limit="nc
<feature name="sunj2se" expiration="2015-07-28" rev="8"/>
<feature name="tls" expiration="2015-07-28" schedule.limit="none" point.limit="none" b
<feature name="web" expiration="2015-07-28" ui="true" ui.wb="true" ui.wb.admin="true"/
<feature name="workbench" expiration="2015-07-28" admin="true"/>
<feature name="zwave" expiration="2015-07-28" schedule.limit="none" point.limit="none"
<signature>MCwCFBontln8OWKTKQza0L6TsaMMeJLT/AhQ97ygsr6nfHMqTFbDVkgMbQ8dqAA==</signature>
</license>

```

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. It is used for verifying the authenticity of license files. All hosts require a Tridium 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 specific reason, as you will likely render the controller inoperable until a proper license or certificate is reinstalled!

License operations

Below the left-hand *license side* of the **License Manager**, these two buttons (commands) are displayed in addition to **View** and **Delete**:

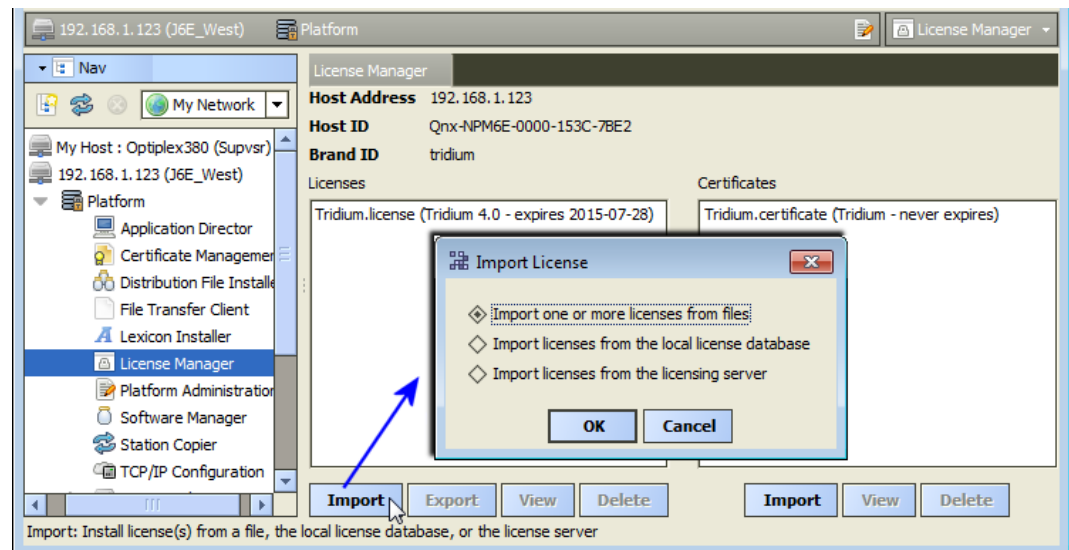
- [Import, page 53](#) — Always available, this provides various options for installing a license file from local files, from the licensing server, or from your “local license database.”
- [Export, page 54](#) — Available if you have a license selected, to save locally as a “license archive file.”

Import using License Manager

The ability to import a license using the **License Manager** is always available, and provides various options for installing a license file from local files, from the licensing server, or from your local license database.

If you choose **Import** from the **License Manager**, the **Import License** window asks you to select the location of the source license.

Figure 35. Import window from License Manager



Select *one* of the following options (depending on scenarios, some may be unavailable, as noted):

- **Import one or more licenses from files**

Always an available option, this 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, an attempt is made to install the license in the host platform.

- **Import licenses from the local license database**

This option 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**

Typically, this option is available if your Workbench PC has Internet connectivity. When you select this option, the system silently searches the licensing server and installs the license.

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

- **Licensing Complete**

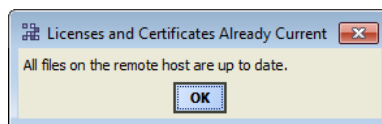
The license was successfully added, as shown below.

Figure 36. Licensing Complete dialog

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.

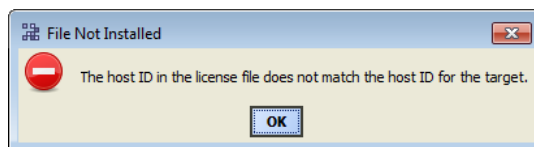
- **Licenses and Certificates Already Current**

The license currently installed on the host already matches the source license (whether specifying any of the license import options). A window opens as shown below.

Figure 37. License and Certificates Already Current

- **File Not Installed**

No appropriate license (by host ID) was found in either the license file or the license archive specified when importing by file, noted with a window similar to below.

Figure 38. File Not Installed

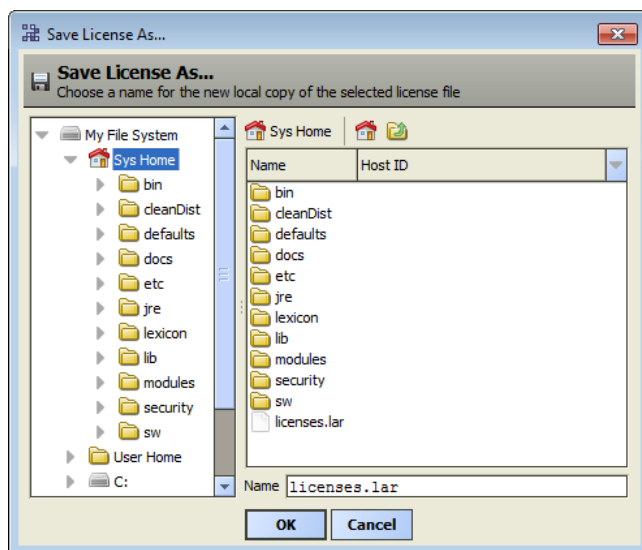
- **(License Request Form, in browser)**

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 for this host.

Export using License Manager

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.

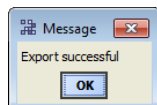
With a license selected in the **License Manager**, the **Export** button opens a **Save License As...** window to save that license file locally on your Workbench PC, as a *license archive* (.lar) file, as shown below.

Figure 39. Save License As dialog

NOTE: You can use the **License Manager's Import** command to install any exported license archive, or the equivalent **Import File** command in the **License Manager** view.

By default, the system saves a license archive file in the root of your Niagara release directory. If needed, use the window's navigation controls to specify another target folder or drive. 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`.

After exporting a license, a notification window opens, as shown below.

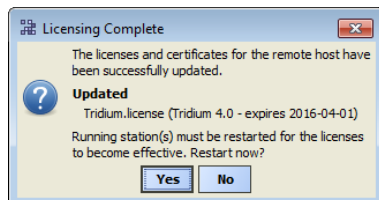
Figure 40. Exported license archive notification dialog

License Import results

Depending on the **Import** option chosen in the **License Manager** (shown below) and the success of the import attempt, after you click **OK**, one of several dialogs may appear to signal completion, as follows:

- Licensing Complete

The license was successfully added, as shown below.

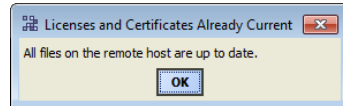
Figure 41. Licensing Complete dialog

NOTE: If a station is running on the host platform, this dialog 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

The license currently installed on the host already matches the source license (whether specifying any of the license import options). A dialog appears as shown below.

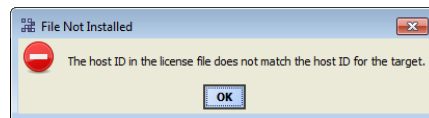
Figure 42. License and Certificates Already Current



- File Not Installed

No appropriate license (by host ID) was found in either the license file or the license archive specified when importing by file, noted with a dialog similar to below.

Figure 43. File Not Installed



- (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 for this host.

About the licensing server

The licensing server is an online 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.

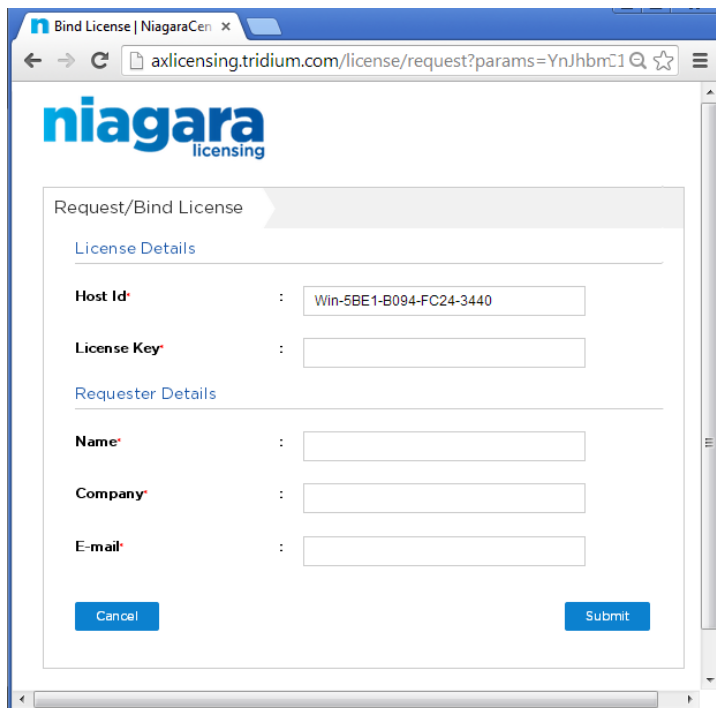
In addition to using the **License Manager** to access licenses via the licensing server, other Workbench views use the licensing server to confirm that a feature is licensed. Examples include the Workbench **Local License Database** tool and the **Network License Summary** view of the Licenses slot of the NiagaraNetwork's **ProvisioningNwExt**

Providing that your PC *currently has Internet connectivity* 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 selecting the license server option. As a side benefit, the system updates your local license database.

NOTE: If sourcing from the license server while platform-connected to a host that has not yet been assigned a license by the server (or has a pending license), a license request form opens in your computer's default browser, similar to that shown below.

Figure 44. License request form in browser (from Workbench, Tools > Request License)

The image shows a web browser window displaying the 'Request/Bind License' form from Niagara licensing. The browser's address bar shows the URL 'axlicensing.tridium.com/license/request?params=YnJhbmC1'. The form has a header with the 'niagara licensing' logo. Below the header, there are two main sections: 'License Details' and 'Requester Details'. In the 'License Details' section, the 'Host Id' field is pre-filled with 'Win-5BE1-B094-FC24-3440', and the 'License Key' field is empty. In the 'Requester Details' section, the 'Name', 'Company', and 'E-mail' fields are all empty. At the bottom of the form, there are two buttons: 'Cancel' and 'Submit'.

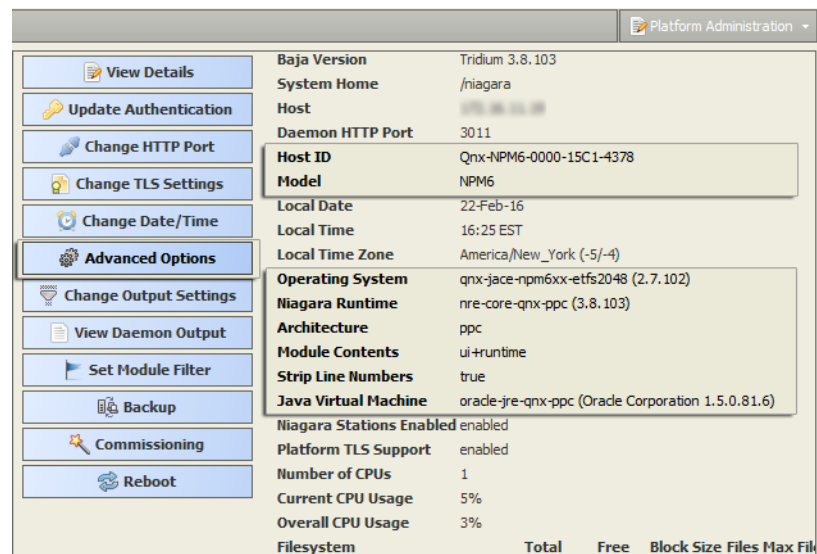
This lets you submit a license request to the licensing server that includes the platform's **Host ID**. In this window, be sure to enter your name, and email address.

If you already received a License Key, a pending unbound license already exists on the licensing server. In this case, you can enter the license key along with the part number to activate that license, and make it immediately available.

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

Platform Administration

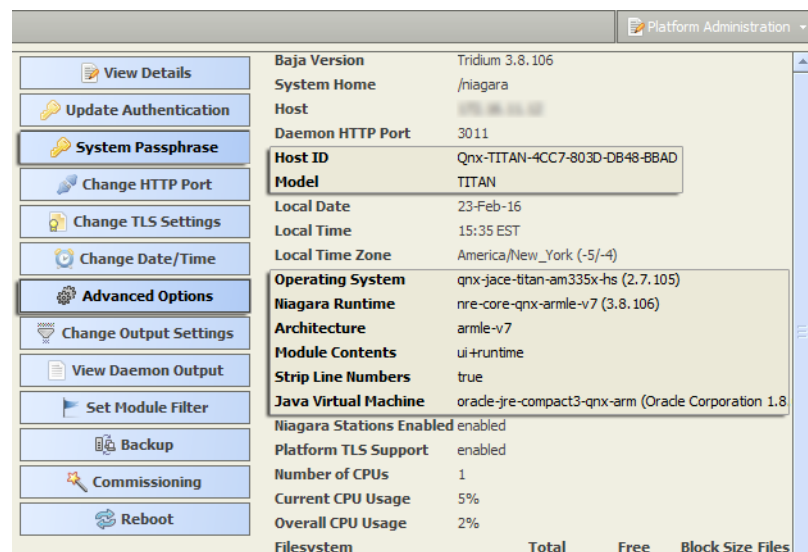
Platform Administration for a JACE platform differs as shown below:

Figure 45. Platform Administration for JACE controller

- An **Advanced options** button is available for enabling/disabling FTP/Telnet, and Daemon Debug functions. For details, see “Advanced Options”.
- Various data in the view (repeated in “View Details”) differ greatly from that for Windows hosts.

Platform differences for JACE-8000 running AX

In AX-3.8U1, adding the JACE-8000-AX license feature to a JACE-8000 enables the platform to run AX. Platform Administration differs as shown below:

Figure 46. Platform Administration for JACE-8000 controller

- The **System Passphrase** button is available for changing the passphrase used to encrypt sensitive information on the platform’s file system.
- The **Advanced options** button is available for enabling/disabling SFTP/SSH, and Deamon Debug functions. For details, see “Advanced Options”.

- Various data in the view (repeated in “View Details”) differ greatly from that for Windows hosts.

Types of Platform Administration functions

The following list summarizes platform administration functions, by button in the view:

- **View Details**

Provides platform summary data, available to the Windows clipboard. Includes all summary information shown in main **Platform Administration** view, plus installed modules, and so on.

- **User Accounts** (NiagaraAX JACE platform) *or* **Update Authentication** (Windows platform)

For dialogs to change *platform login access* (user name and password). In QNX-based platforms this is simple, as there is only one platform administrator. Windows-based platforms offer a choice of a single (digest) platform account, or use of Windows OS user accounts (basic authentication).

- **System Passphrase**

On Niagara 4 JACE platforms only, a dialog to change the password used to encrypt sensitive data such as client passwords in station database files (config.bog) and station backup .dist files. On a JACE-8000 platform, this is also used to encrypt data on its removable microSD flash drive, and when creating backup images to a USB flash drive.

- **Change HTTP Port**

For a dialog to change the HTTP port for the host’s platform daemon from (default) port 3011 to some other port.

- **Change TLS Settings**

For a dialog to enable/disable secure TLS connections to the host’s platform daemon, specify the TCP port used, plus other settings.

- **Change Date/Time**

For a dialog to change the hosts’s current date, time, and time zone, as used by that host’s OS.

- **Advanced Options**

(JACE controllers only) On NiagaraAX JACE platforms, a dialog to enable/disable FTP, Telnet, and Daemon Debug functions. On Niagara 4 JACE platforms, a dialog to enable/disable both SFTP and SSH access to the controller, change the default port number shared by these protocols, or to enable/disable Daemon Debug.

NOTE: Enabling SFTP and SSH poses security risks. We strongly recommend you *keep this access disabled*, unless otherwise directed by Systems Engineering.

- **Change Output Settings**

Provides a dialog to change the log level of different processes that can appear in the platform daemon output.

- **View Daemon Output**

Provides a window in which you can observe debug messages from the platform daemon in real time, including the ability to pause.

- **Set Module Filter**

Provides a dialog to change the enabled runtime profile of the host. Used very infrequently.

- **Backup**

Make a complete backup of all configuration on the connected host platform, including all station files as well as other Niagara configuration.

- **Commissioning**

A way to launch the Commissioning Wizard (alternative to right-click on Platform in the Nav tree).

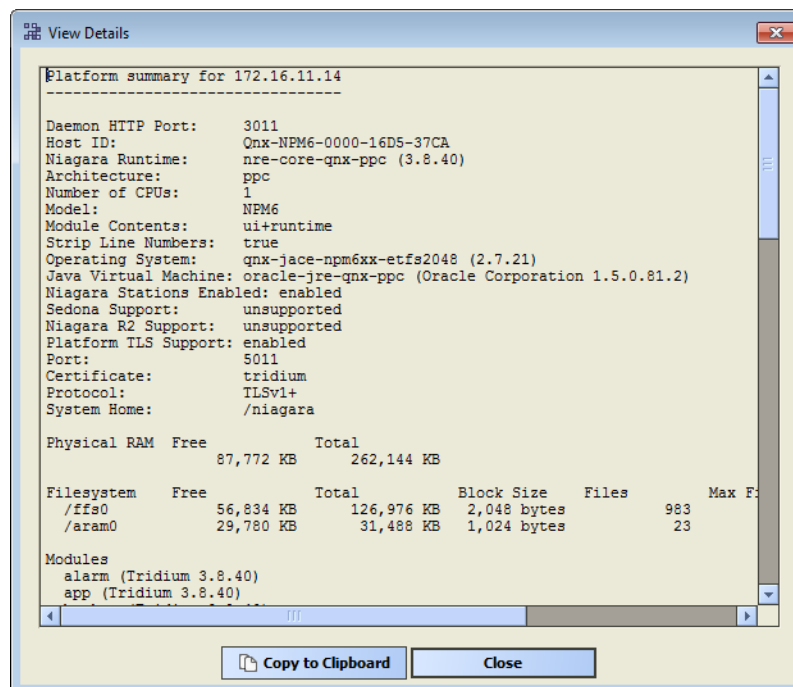
- **Reboot**

Provides a method to reboot a JACE platform, which restarts all software including the OS and JVM, then the platform daemon, then (if so configured in the **Application Director**) the installed station. If you click this, a confirmation dialog appears. If you answer yes, the JACE is rebooted and the platform connection drops.

View Details

This selection from the main **Platform Administration** view lists more platform information than shown in the main view.

Figure 47. View Details dialog in Platform Administration



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. Generally, information in this view is helpful when troubleshooting or asking for technical support. Buttons include:

- **Copy to Clipboard**

Puts all details in the dialog on your PC's Windows clipboard.

- **Close**

Exits the dialog, same as Windows close control (contents copied remain on clipboard).

Update Authentication

This selection from the main **Platform Administration** view lets you change that platform's authentication. This affects the login used to access the host's platform daemon. Depending on the type of platform currently opened (QNX-based or Windows-based), update authentication provides different dialogs, as follows:

- QNX-based platforms:
 - Digest platform authentication

NOTE: Also see “Improvements to AX-3.8 digest authentication”

- Windows-based platforms, either:
 - Basic platform authentication
 - Digest platform authentication

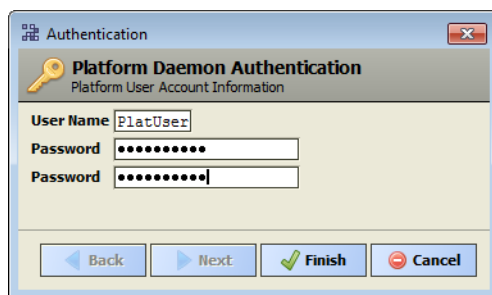
Digest platform authentication

Digest platform authentication is the only method for a QNX-based host or Linux-based Supervisor, and is an alternative for a Windows-based host. The associated Authentication dialog lets you change the single platform account credentials (user name and password), as shown here.

Digest platform authentication is the only method for a QNX-based host or Linux-based Supervisor, and is an alternative for a Windows-based host. The associated Authentication dialog lets you change the single platform account credentials (user name and password), as shown here.

NOTE: Credentials are case-sensitive. For example, PlatUser and Platuser are not the same.

Figure 48. Platform authentication dialog for digest authentication



CAUTION: When commissioning a new JACE, always change platform credentials from the defaults! A JACE installed with default platform credentials is extremely susceptible to unauthorized intrusion! Further, in AX-3.8 there are related "warnings". See “Improvements to digest authentication”.

In digest authentication, platform user name can be as follows:

User Name

- If QNX-based host, 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 (_).

- If Windows-based host, any number of alphanumeric characters, including hyphens and underscores.

Password

- In digest authentication, platform password for both QNX-based and Windows-based hosts can be any combination of alphanumeric characters, including common punctuation (! @ # \$ %). This permits a strong password.

NOTE: A "strong password" is highly recommended. Some basic guidelines on strong passwords:

- Use both upper and lower case.
- Include numeric digits.
- Include special characters.
- Don't use dictionary words.
- Don't use company name.
- Don't make the same as the user name.
- Don't use common numbers like telephone, address, birthday, and so on.

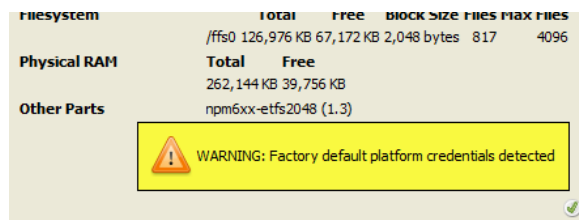
Usage Notes

In digest authentication, when changing credentials (user name or password, or both), your new credentials become immediately effective when you click **Finish**. If you previously had "Remember these credentials," selected in the Authentication login dialog, the cached credentials are automatically updated.

Improvements to digest authentication

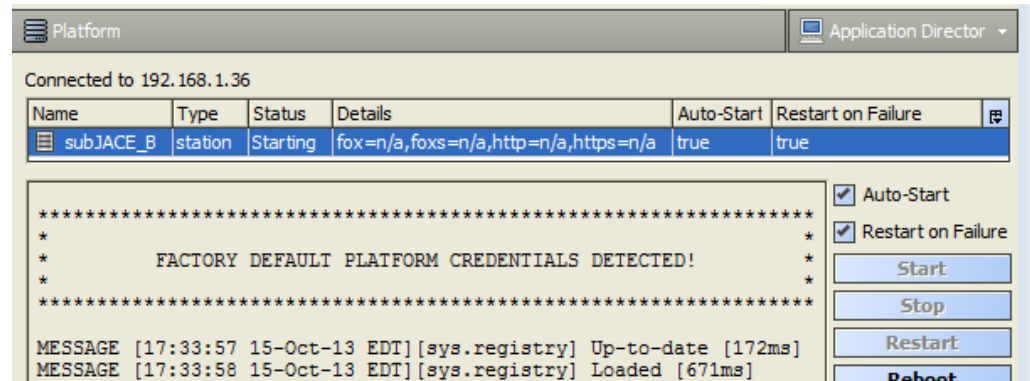
In AX-3.8, improvements were made in platform digest authentication and JACE security, as follows:

- Platform digest credentials now use a strong, two-way AES-256 encryption technique, utilizing the unique keyring and key material file of the host (JACE).
- Platform digest credentials were relocated to a more secure location in the registry of the host platform (e.g. JACE).
- Any AX-3.8 JACE controller operating with factory default platform credentials issues warnings seen with an AX-3.8 Workbench platform connection to it, in these areas:
 - In the **Platform Administration** view, a yellow warning box remains in the bottom right area of this view: Factory default platform credentials detected



This warning remains until you change the credentials to non-defaults, using Update Authentication from this same **Platform Administration** view.

- In the **Application Director** view, upon station startup a text warning is seen in station output before any other messaging.



Note that related to this change in AX-3.8 digest platform credentials, that station backups no longer store platform credentials which can affect backup restoration behavior. For related details, see AX-3.8 changes to backup dist usage.

Basic platform authentication

In a Windows-based platform can use either digest or basic (native Windows OS user based) authentication for Niagara platform access.

- Digest platform authentication provides good protection against password eavesdropping. However, there is only one level of platform login access, using a single platform user account.
- Basic platform authentication provides integration with existing Windows installations, and provides two levels of platform access. However, it does not protect against password eavesdropping.

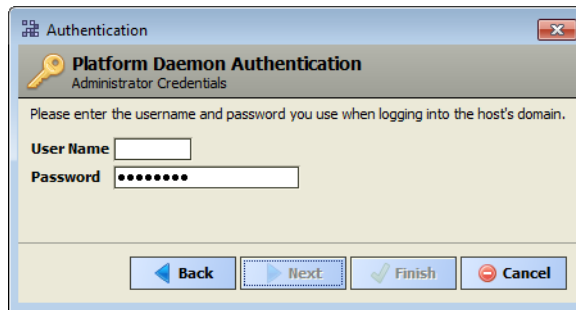
For any Windows-based host, when you update platform authentication a dialog asks you to select one of the two methods, as shown.

Figure 49. Authentication dialog for Windows Niagara host

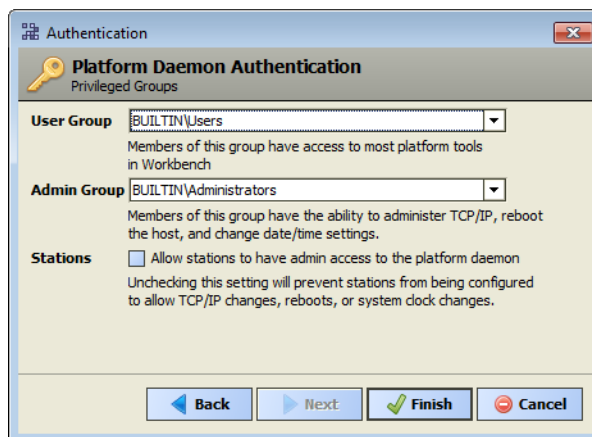


- If you select digest authentication, upon clicking Next you go to the authentication dialog to set the single platform login account. There is no linkage between Windows OS user accounts and the platform administrator.
- If you select basic authentication, a dialog opens where you can assign one existing Windows user group to each of the two possible levels of platform access.

NOTE: If the host platform is currently configured for digest authentication, and you change to basic authentication, you first see a login dialog, as shown here. If already configured for basic authentication, you go directly to the basic authentication dialog.

Figure 50. Login dialog when changing from digest to basic authentication

Use your standard Windows login credentials — if the host is on a Windows domain, login using the credentials you use when logging into 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 will be selectable in the next dialog for Basic Platform Authentication, shown below.

Figure 51. Basic platform authentication dialog, group selection

This basic authentication dialog lets you select one Windows group for each of the two levels of platform access. In addition, the "Stations" checkbox determines certain platform writes from a station.

Stations access

A "Stations" checkbox in the basic authentication dialog allows you to disable any station user from changing TCP/IP settings, system time, or rebooting the host by accessing the station's PlatformServices.

NOTE: In general, if a Windows-based JACE, you should leave the Stations checkbox enabled. However, if a Supervisor (PC) platform, you may wish to clear this checkbox, particularly if the local IT department has host access concerns.

Levels of platform access

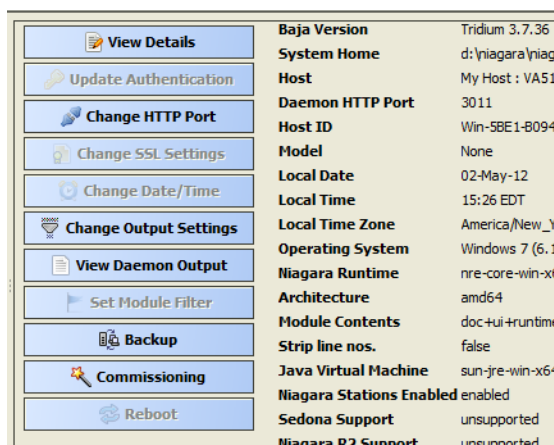
Basic platform authentication provides two levels of platform access, which are determined by a user's group membership(s). The levels of platform access are:

- **User** - Platform access at this level allows full use of most Workbench platform views. This includes the ability to change platform daemon HTTP port, install or delete licenses and stations (including the one running), also to install, re-install, or upgrade the platform dist file and/or modules, and to start, re-start, or stop a station.

- **Admin** - Full access. This includes all user-level platform operations, plus the ability to configure host TCP/IP settings and dialup configuration, change platform authentication, change host date/time settings, use the File Transfer Client, and reboot the host.

NOTE: When platform-connected at the user level (vs. admin), some platform views are read only. This includes views for **TCP/IP Configuration** and **User Manager**. In addition, some **Platform Administration** view buttons are unavailable, as shown.

Figure 52. Platform Administration view if user-level platform login

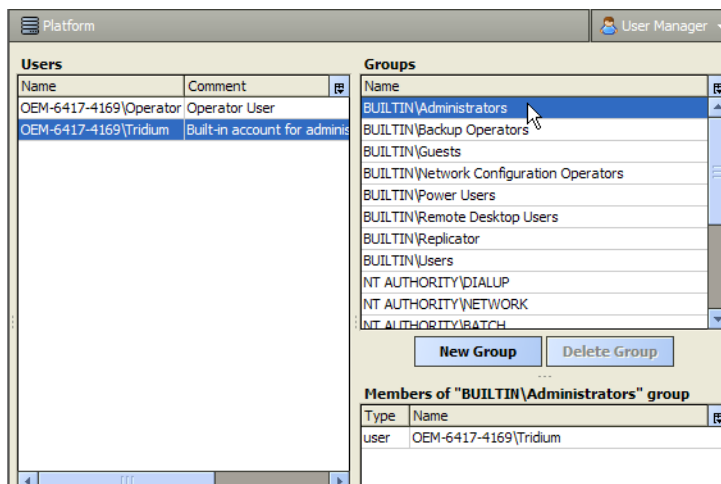


Platform access to a remote Windows-based host also provides a **User Manager** view in which you can manage Windows users and groups local to that host.

Privileged group selections

For platform admin level access, you can select from a list of user groups known to Windows on that host, as shown here.

Figure 53. Group selections include Windows built-in user groups



Groups include Windows "built-in" user groups (include "BUILTIN" or "NT AUTHORITY" prefix), as well as any locally-defined user groups. If the remote host has been added to a Windows domain, groups defined in that domain are also listed and available.

NOTE: Domain groups are limited to only those in which the login user is a member.

If a user has membership in both assigned Windows user groups, upon successful platform login they have admin-level platform access.

NOTE: Default group selections for a Niagara Windows installation (either Workbench/Supervisor installation or a factory-shipped JACE-NXS) are as follows:

- User Group – BUILTIN/Users
 - Admin Group – BUILTIN/Administrators
-

System Passphrase

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. The passphrase increases security for the files that contain critical information. In various Workbench operations, you are prompted to enter the passphrase, such as when copying stations or restoring station backups in remote platforms.

NOTE: This system passphrase functionality applies to a JACE-8000 controller, even if you downgrade the unit from N4 to AX. In this situation, configuring sensitive data must be accomplished via the Workbench.

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

- Provisioning
- Distribution File Installer
- File Transfer Client
- Station Copier
- 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. Files located under a Workbench 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* or *portable* locations, when transferring files between the daemon User Home and another Workbench 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.

- **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 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.

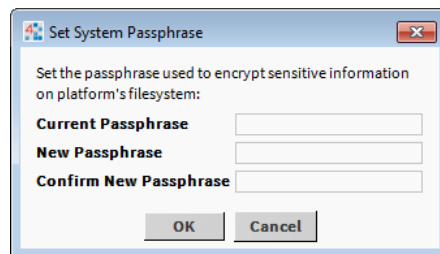
CAUTION: It is important to remember the system password and keep it safe. If you lose the system passphrase, you will lose access to encrypted data.

Update the system passphrase

To change the system passphrase on a Niagara 4 platform use either the **Commissioning Wizard** or the **Platform Administration** view as described here.

In the **Platform Administration** view, when you click **System Password** for any Niagara 4 platform (including a JACE-8000 downgraded to run AX-3.8U1) the window below opens.

Figure 54. Set System Password dialog



A strong password is required (must *match* in both password fields). The characters you enter are obscured.

Password rules are the same as for platform users, using a minimum of 10 characters, with:

- *At least one UPPER CASE character.*
- *At least one lower case character.*
- *At least one digit (numeral).*

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

NOTE: 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.

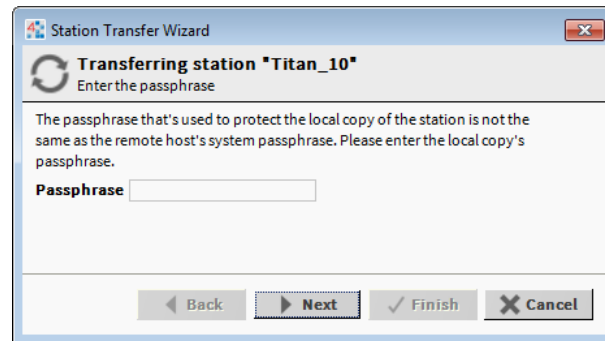
System passphrase usage in backups and station copies

The system passphrase is required when using either the **Distribution File Installer** to restore a backup .dist file, or the **Station Copier** to transfer a local file

Note the following:

- If the file passphrase and system passphrase are the same, the station copy proceeds without prompting for a passphrase.
- If the passphrase for the bog file is not the same as the passphrase for the target host platform then you are prompted to enter the bog file's passphrase, as shown below.

Figure 55. Station Transfer Wizard prompt for bog file passphrase



The above dialog is prompting you to enter the bog file's passphrase.

NOTE:

- If you do not know the passphrase for a BOG file, you can edit it offline.
 - If you do not know the passphrase for a .dist file you cannot install it.
-

Editing BOG files offline

Files created in Workbench initially have no passphrase at all since the files do not yet contain sensitive data. You can add passphrase protection to offline BOG files by clicking the **Bog File Protection** icon in the toolbar.

If you change or add a passphrase value, attempts to **Save** prompt you to enter the file passphrase. You can **Save** only if you enter the correct passphrase or add a new one.

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. Choosing either of these options clears any sensitive data in the file.

CAUTION: When you Unlock (force-remove) or Change (force-change) the passphrase on a Bog file it results in the loss of the sensitive data in the file.

System passphrase usage in JACE-8000

The JACE-8000 makes additional use of its system passphrase, to encrypt sensitive information on its removable microSD flash drive, as well as when writing backup images to a USB flash drive. The passphrase is assigned as the file passphrase for portable copies of backups and station copies.

NOTE: The system passphrase default value is the same as the default platform password for controllers that you have just converted from NiagaraAX, and controllers on which you have just installed a clean dist. On the first commissioning of such controllers you are prompted to change the passphrase from the default value.

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

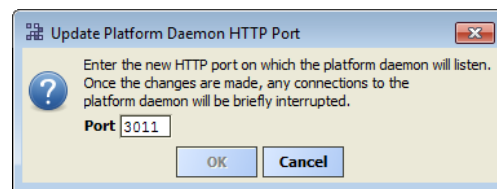
- If the replacement unit is preconfigured with the same system passphrase, the unit starts.
- If the replacement unit has a different system passphrase, the unit will not boot, and the status LED flashes every second. To resolve, you must make a serial connection and, when prompted, select either: **Update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

Change HTTP Port

This function on the **Platform Administration** view lets you change the HTTP port monitored by the host's platform daemon for regular platform client connections (connections that are not secure). By default, port 3011 is monitored for such connections. This differs from any port used for station (Foxs) connections that are 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.

Figure 56. Update Platform Daemon HTTP Port dialog



If needed, you can change the daemon monitored port to another HTTP port. You may choose to do this for specific firewall reasons, or perhaps for additional security. As shown in the figure above, you can type in the new port number in the **Port** field, which enables the **OK** button.

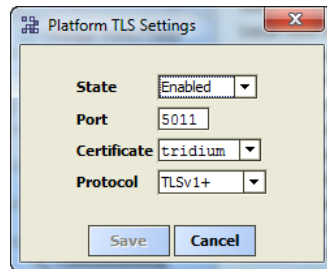
When you click **OK**, the platform daemon restarts, and your platform connection reopens (this does not affect the operation of any running station). If previously connected on the port without security, the platform icon shows in the Nav tree with the new HTTP port number (:n) in parenthesis.

NOTE: Before closing the host, which removes it from the Nav tree, *carefully note* the new (non-default) port number you entered. You must specify the port number the next time you open a platform using a connection that is not secure. To check this port number in a station running on the host, open **Config > Services > PlatformServices** property sheet.

Change TLS Settings

This selection from the **Platform Administration** view lets you configure for secure (TLS) platform connections, as well as change related secure platform connection (platformtls) parameters.

The figure below shows the dialog with default values.

Figure 57. Platform TLS Settings with default values (enabled)

Fields in this dialog are as follows:

- State

Either Disabled, Enabled, or Tls Only, to specify how Workbench clients can connect to this host's platform daemon.

- Disabled — Secure platform connections not possible (only regular platform connections).
- Enabled — Secure platform connections permitted, *as well as* regular platform connections.
- 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 "Tls Only" setting provides the best security. However, for any AX JACE in which you are about to install a "clean dist" file, you should first change State to "Disabled". Otherwise after the unit reboots from the clean dist installation, you will be unable to make a default (non-TLS) platform connection. In this case, a direct serial shell session to the JACE is required, with intervention during the boot process. Note that in Niagara 4, all platforms support secure (TLS) platform connections, even if a freshly "clean disted" controller.

- Port

Software port monitored by the platform daemon for a secure platform connection, where port 5011 is the default. Note this is different than the default HTTP port (3011) for a regular platform connection that is not secure.

CAUTION: Before changing this port, if there is a firewall on the host make sure that it will allow traffic to the new port.

- Certificate

The "alias" for the server certificate in the platform's "key store" to use for any platformtls connection. The default is the `tridium` self-signed certificate, which is automatically created when Niagara is first loaded. If another certificate has been imported in the platform's key store, you can use the drop-down control to select it instead.

Certificates on the platform are managed via the platform **Certificate Management** view. For general information in this document, see *NiagaraAX SSL Connectivity Guide*.

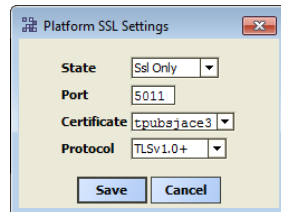
- Protocol

The minimum TLS protocol (Transport Layer Security) that the platform daemon's secure server will accept to negotiate with a client for a secure platform connection. During the handshake, the server and client agree on which protocol to use.

- TLSv1.0+ — (default) Includes TLS versions 1.0, 1.1, and 1.2, providing most flexibility.
- TLSv1.1+ — Only TLS versions 1.1 or 1.2 are accepted.
- TLSv1.2 — Only TLS versions 1.2 is accepted.

The figure below shows an example dialog for a controller enabled for platform TLS (only).

Figure 58. Example settings for a controller enabled for TLS, with a signed certificate

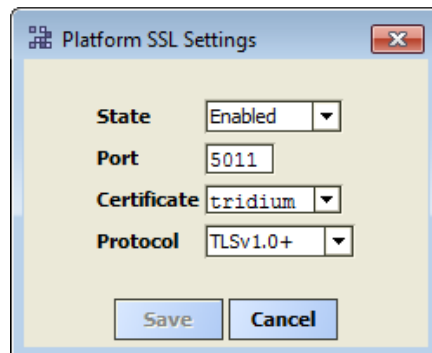


In this example, the controller uses a signed certificate with alias `tpubsjace3` (previously imported), with the port and protocol settings left at defaults.

Change TLS Settings window

This window provides access to the primary TLS settings.

Figure 59. Platform TLS Settings with default values (enabled)



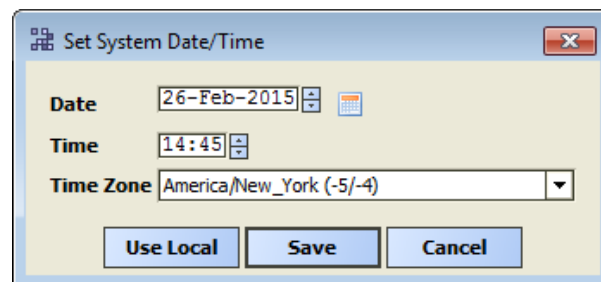
Properties	Value	Description
State	Disabled, Enabled, or Tls Only	<p>Specifies how Workbench clients connect to this host's platform daemon.</p> <ul style="list-style-type: none"> Disabled — Secure platform connections not possible (only regular platform connections). Enabled — Secure platform connections permitted, <i>as well as</i> regular platform connections. Tls Only — Only secure platform connections are allowed. Any attempt to connect without security goes unresolved (errors out). <p>This state is reflected among the properties listed on the main Platform Administration view, as “Platform TLS Support” state.</p> <hr/> <p>NOTE: The Tls Only option provides the best security. In Niagara 4, all platforms support secure (TLS) platform connections, even if a freshly “clean disted” controller.</p>
Port	four-digit number (default is 5011)	<p>Identifies the software port monitored by the platform daemon for a <i>secure</i> platform connection. This is different than the default HTTP port (3011) for a regular platform connection that is not secure.</p> <hr/> <p>CAUTION: If there is a firewall on the host (or its network), <i>before changing</i> this port make sure that the firewall will allow traffic to the new port.</p>

Properties	Value	Description
Certificate	text (default is the <code>tridium</code> self-signed certificate)	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. 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 in this document, see <i>Station Security Guide</i> .
Protocol	TLSv1.0+ — (default) 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 — Only TLS version 1.2 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.

Change Date/Time

This selection from the main **Platform Administration** view lets you change the date and time in the platform, as well as specify its time zone.

Figure 60. Set System Date/Time window



Typically, if your Workbench PC's current date/time setting are accurate, you click the **Use Local** button to synchronize the remote host's date, time, and time zone with your Workbench PC. Upon **Save**, the remote host will have the identical settings.

NOTE: To keep time synchronized across multiple platforms, configure the **NtpPlatformService** in the **PlatformServices** of the *station* running on each platform, as appropriate.

The **Save** button becomes available after you change one or more fields in the window, or when you click **Use Local**. Upon **Save**, any change is processed by the host's operating system.

- To set the date, click in a day-month-year position to select, then click up/down controls, or click and type in numerals directly, or click the calendar icon for a popup dialog to select the date from a calendar.
- To set the time, click in a hour or minute position to select, then click up/down controls, or click and type in numerals directly.
- Select the time zone from the drop-down list.

Set System Date/Time window

This window configures the remote platform's date and time.

Buttons

Button	Value	Description
Use Local	button	
Save		

Properties

Property	Value	Description
Date	three fields	Defines a day-month-year.
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).

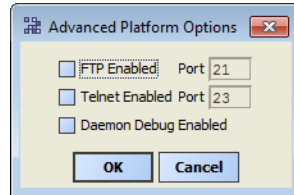
Advanced Options

This selection from the **Platform Administration** view differs slightly depending on which type of JACE platform is accessed.

Advanced Options for NiagaraAX JACE platforms

For NiagaraAX JACE platforms, this dialog appears to enable/disable FTP, Telnet, and Daemon Debug functions, as shown. For Windows-based hosts, you typically use Windows “Remote Desktop Connections” instead.

Figure 61. Advanced Platform Options dialog



As factory-shipped, a QNX-based NiagaraAX JACE, has the FTP and Telnet service 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 one or both services, perhaps to facilitate debugging.

CAUTION: FTP and Telnet pose security risks. We strongly recommend you keep each one disabled, unless otherwise directed by Systems Engineering.

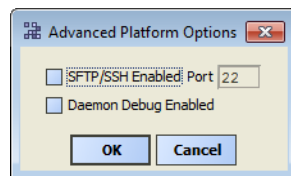
Note that Telnet access to a QNX-based JACE controller provides “system shell” access, providing (after login using platform credentials) the same menu as “serial shell access” to its RS-232 port. For related details, see the “System shell” section in the JACE NiagaraAX Install & Startup Guide.

You can also change the TCP/IP port used by each service from the “well-known” port to some other port. However, be sure that any firewalls being used on your network will allow traffic to that port.

Advanced Options for Niagara 4 platforms

For Niagara 4 JACE platforms, such as the JACE-8000, this option is available to enable, disable, or configure SFTP (Secure File Transfer Protocol) or SSH (Secure Shell Protocol) access. For Windows-based hosts, you typically use Windows “Remote Desktop Connections” instead.

Figure 62. Advanced Platform Options dialog



NOTE: This replaces an “FTP/Telnet” selection available for QNX-based JACE platforms running NiagaraAX, which are both inherently less secure services.

As factory-shipped, a Niagara 4 JACE controller has the SFTP and SSH service disabled — which protects against platform exposure 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 for platform SSL only, and *keep this function disabled*, unless otherwise directed by Systems Engineering.

Note that SSH access to a JACE controller provides “system shell” access, providing (after login using platform credentials) the same menu as “serial shell access” to its RS-232 port. For related details, see the “System shell” section in the *JACE Niagara 4 Install & Startup Guide*.

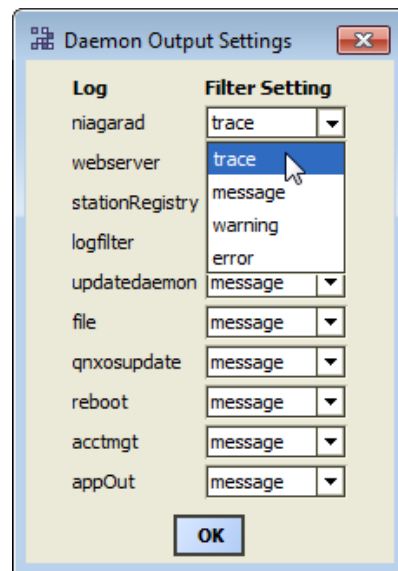
You can also change the TCP/IP port shared by these services from the “well-known” port to some other port. However, be sure that any firewalls being used on your network will allow traffic to that port.

Change Output Settings

This function from the main **Platform Administration** view lets you adjust (tune) the amount and content of the platform daemon output.

You do this by changing the log filter settings of the various daemon processes.

Figure 63. Daemon Output Settings window for a JACE controller



Logs

By default, all daemon processes have a message log filter level, and include the following:

- **niagarad** — Log for the platform daemon (niagarad) process, with high level entries like `niagarad starting, baja home = . . . , niagarad stopping`.
- **webserver** — Log for HTTP server for incoming platform client connections. Entries are often generic, before the daemon hands off to the appropriate platform servlet.
- **stationregistry** — Log for platform daemon management of stations, including startup, shutdown, and watchdog actions.
- **logfilter** — Logs changes to daemon log states, meaning it tracks the changes made in this window.
- **updatedaemon** — Log for handling Workbench requests for current platform daemon configuration, used mainly by the **Platform Administration** view.
- **file** — Logs requests made to the platform daemon’s file servlet, used in platform views like the **File Transfer Client**, Commissioning Wizard, **Software Manager**, **Station Copier**, and so on. Many different things can print on this log, such as `request for file xxx`, and `wrote file xxx`.
- **qnxosupdate** — Log for the OS upgrade servlet created by the platform daemon. Workbench uses this servlet to upgrade the QNX OS in the host JACE when using the

Commissioning Wizard or Distribution File Installer. Entries here can reflect a problem when updating the QNX OS, such as `os crc isn't right`, and `waitpid` when launching `osupdate` command failed.

- **reboot** — Log for the reboot servlet, one of the servlets the platform daemon manages.
- **appOut** — Log for the thread managing buffers associated with station output, making that output visible in the **Application Director** view. Entries may reflect buffer size changes (available in Application Director interface), or if a problem occurs streaming the output to Workbench.

Filter Settings

For any item, use the **Filter Setting** drop-down to select one of the following:

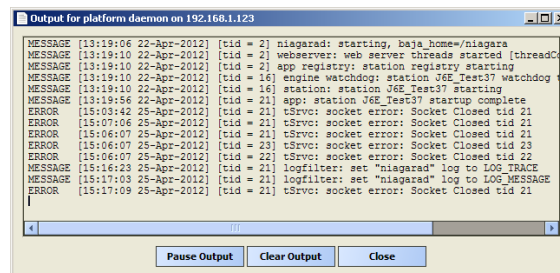
- **Trace**
Returns all message activity (*verbose*). This includes all transactional messages, which may result in too many messages to be useful. Be careful using Trace!
- **Message**
(Default) Returns informational “MESSAGE”s, plus all “ERROR” and “WARNING” types.
- **Warning**
Returns only “ERROR” and “WARNING” type messages (no informational “MESSAGE”s).
- **Error**
Returns only “ERROR” type messages (no “WARNING” or informational “MESSAGE”s).

View Daemon Output

This selection from the main **Platform Administration** view lets you examine standard output from the host’s platform daemon in real time. It is available for troubleshooting purposes.

NOTE: Output is different from the output of a running station, as seen in the **Application Director**.

Figure 64. Example Output for platform daemon



Depending on the log filter settings set in platform administration’s **Daemon Output Settings** dialog, the activity level in the output window will vary. Output is “non-modal,” meaning that you can leave this window open and still do other Workbench operations (including change output settings).

As needed, use the scroll bars to navigate through messages, which will have headings “TRACE,” “MESSAGE,” “WARNING,” or “ERROR,” depending on message type. Each message includes a timestamp and a thread id number.

Use the Windows copy shortcut (CTRL + C) to copy text of interest to the Windows clipboard.

Click **Pause Output** to freeze the output from updating further (no longer in real time). When you do that, note that 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.

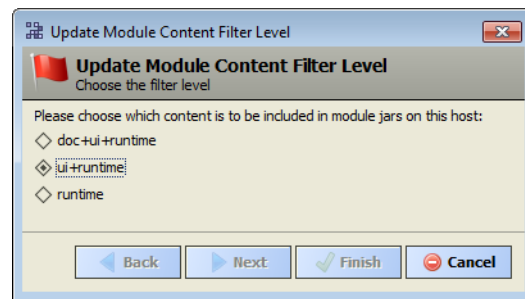
Click **Clear Output** to clear all collected messages from the current daemon output window. This not a “destructive clear,” as another (or new) daemon output window retains daemon messages.

Set Module Filters

This selection from the **Platform Administration** view lets you globally change the “runtime profile” types for software modules on the connected platform. This affects how much file space consumed by installed Niagara modules.

NOTE: Typically, enabled runtime profiles you set this *once* during initial JACE commissioning, then never change it.

Figure 65. Update Enabled Runtime Profiles dialog



- For any Windows-based host (providing it has hard drive for file storage), you typically want all runtime profiles enabled for the fullest possible content level, meaning including all documentation (doc+ui+runtime).
- For a JACE controller, with more limited flash-based file storage, in certain scenarios you may wish to change the enabled runtime profiles. Selection produces the dialog above.

Module content level is one of the following, from largest to smallest:

- `doc+ui+runtime`—Typically appropriate only for Windows-based platforms.
- `ui+runtime`—Appropriate for any JACE that is to run the Web Service. This is typical for any “standalone” JACE, as well as any JACE that serves PxPages directly to browser clients.
- `runtime`—Typically only for a JACE controller not running Web Service (all PxPages served instead by a Supervisor).

Results from a change in module content level

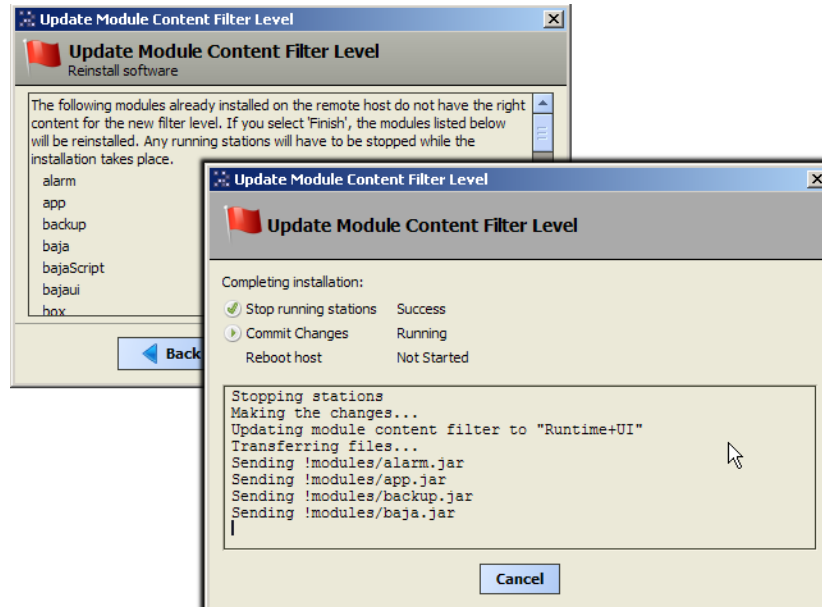
Depending on how you change the module content filter level, operations on the platform vary:

- If you decrease the content level (say, go from “ui+runtime” to “runtime”), modules already installed are not automatically re-installed (to reduce storage). You simply click the **Finish** button to close the dialog, and platform/station operation is otherwise unaffected. However, if you later re-install existing modules, or install new modules, the new content filter level is applied—typically with resulting savings in storage space.

So, if “freeing” storage space is the goal when restricting module content, after changing the content level, you should re-install existing modules. Do this using the Software Manager.

- If you increase the content level (say, from "runtime" to "ui+runtime"), this typically requires modules to be re-installed in that platform. In this case, the dialog provides a **Next** button and explains that this automatically occurs, with the station first stopped as a result, as shown. Note that if a JACE platform, this also automatically results in a reboot of that host platform.

Figure 66. Dialog example after increasing module content level



Backup

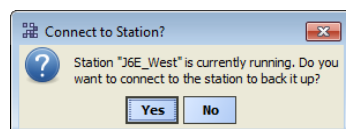
This selection from the **Platform Administration** view performs a complete backup of the connected JACE, saved as a .dist file on your PC. The backup dist contains the entire station folder plus the specific NRE config used by that JACE platform, including license(s) and certificate(s). The dist also contains pointers to the appropriate NRE core, Java VM, modules, and OS. If ever needed, you restore a backup dist using the platform **Distribution File Installer** view.

NOTE: The backup dist file also contains the TCP/IP configuration of the host when it is backed up. When restoring the backup, you can select to restore these settings, or retain the TCP/IP settings currently in use by the target host. See [“Restoring a backup dist”](#).

You can perform a backup with a station running on the target host, or when no station is running.

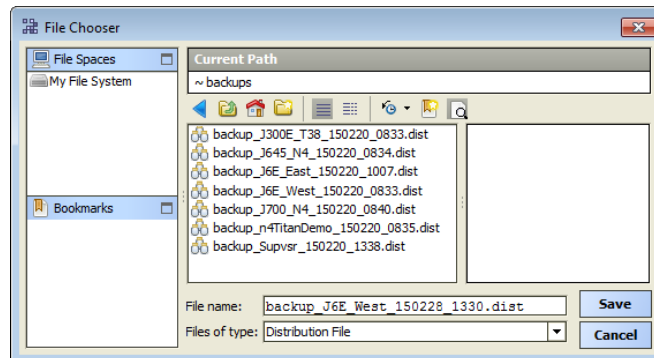
- If the JACE is running station, a confirmation dialog appears to connect to it, as shown below. This routine uses that station’s **BackupService** to perform an “online backup.” (If the station is not already open in Workbench, you must then logon as a station user.)
- If no station is running on the JACE, the platform daemon performs its own “offline backup.”

Figure 67. Backup with station running, station connection



After station login and connection to the station (or if no station is running), the **File Chooser** appears, as shown below. Navigate to a target location to save the backup file, and to rename if desired.

Figure 68. File Chooser to select target folder and dist file name



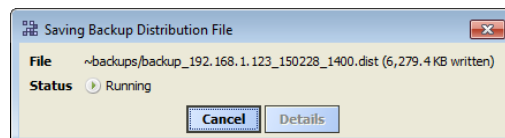
By default, the Backup function automatically creates (if not already present) a backups subdirectory under your Niagara build directory. The default name for a backup file uses a format of: `backup_stationName_YYMMDD_HHMM.dist`

For example, “backup_J6E_West_150228_1330.dist” for a backup made of station “J6E_West” on February 28, 2015 at 1:30 pm.

After you click **Save** the backup starts.

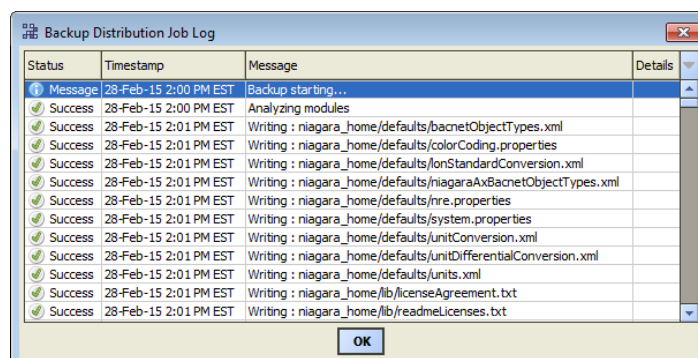
- If the station is running, a Fox Backup job is performed. A notification popup appears in the lower right of your display when the backup is done. This job is recorded in the station’s **BackupService** and visible in that component’s **BackupManager** view. Details are also available by accessing the job in the station’s **Job Service Manager**.
- If doing an “offline backup” (no station running), the platform daemon provides another progress dialog during the backup to a dist file, as shown below.

Figure 69. Backup from Platform Administration, no station running



Upon completion, you can click **Close** to return to the **Platform Administration** view, or click **Details** to see another popup with a log of actions performed in the backup, as shown below.

Figure 70. Available Details from backup using platform daemon (no station running)



Commissioning

This selection from the **Platform Administration** view launches the **Commissioning Wizard**, an ordered sequence of various platform steps.

NOTE: The **Commissioning Wizard** is intended for a remote controller only. This button is *unavailable* whenever you are connected to any Windows platform.

Typically, you use the **Commissioning Wizard** for the following:

- The *initial* installation and startup of a controller.
- To *upgrade* a controller.

CAUTION: For any AX-3.6U4 station with CryptoService that you attempt to upgrade to AX-3.8U1, once you commission the controller the station will fail to start after the "successful" upgrade. The same is true if you attempt to move an AX-3.6U4 supervisor to an AX-3.8U1 station and start it. As a preparatory step, manually remove CryptoService from the station's Services directory before attempting to commission it.

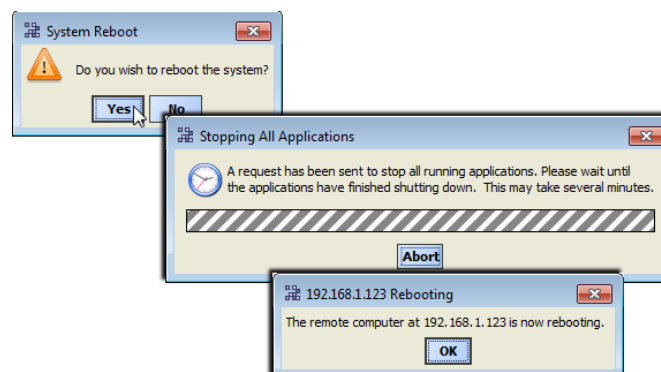
NOTE: Non-portable password encoding in AX-3.6 and AX-3.7 stations prevents upgrading those stations to AX-3.8 installations without first converting the passwords to a portable encoding format. Update release AX-3.8U1 provides the plat makeportable tool which converts such passwords to a portable format. For details on how to use the tool, see "Running the plat makeportable command" in the *JACE NiagaraAX Install & Startup Guide*.

Reboot

This selection from the **Platform Administration** view reboots the host of a connected platform.

CAUTION: For any Windows-based host, never use reboot in place of restart station (from Application Director), unless there is a specific need for it! Reboot is a drastic action to take on any Niagara host.

Figure 71. Reboot performs operating system reboot



As shown above, a confirmation dialog appears, after which the daemon attempts to stop any running station before issuing the final reboot. A reboot restarts the host OS, Java VM, platform daemon, and finally the Niagara station (providing that it is configured to "Auto-restart," see Application Director).).

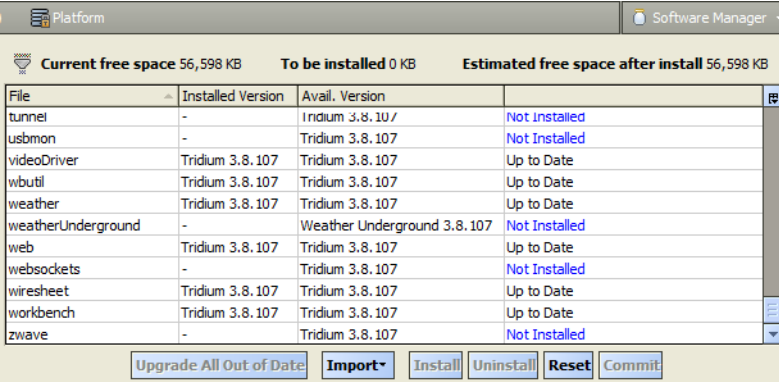
When the platform reboots, your Workbench platform connection to it is dropped. Depending on the platform type, it may take from several seconds to a couple of minutes before you can connect again.

NOTE: Reboot is intended for a remote JACE only. Please note that this button is unavailable whenever you are connected to your "localhost" (Supervisor) platform.

Software Manager

As shown in the figure below, the **Software Manager** is one of several platform views. This view lets you install, uninstall, or simply review all software modules installed in a remote JACE platform. By default, this view compares the platform's modules against your "locally available" modules, meaning the most current modules in the software database on your Workbench PC.

Figure 72. Software Manager compares remotely installed modules to locally available



File	Installed Version	Avail. Version	
tunnel	-	Iridium 3.8.10 /	Not Installed
usbmon	-	Tridium 3.8.107	Not Installed
videoDriver	Tridium 3.8.107	Tridium 3.8.107	Up to Date
wbutil	Tridium 3.8.107	Tridium 3.8.107	Up to Date
weather	Tridium 3.8.107	Tridium 3.8.107	Up to Date
weatherUnderground	-	Weather Underground 3.8.107	Not Installed
web	Tridium 3.8.107	Tridium 3.8.107	Up to Date
websockets	-	Tridium 3.8.107	Not Installed
wiresheet	Tridium 3.8.107	Tridium 3.8.107	Up to Date
workbench	Tridium 3.8.107	Tridium 3.8.107	Up to Date
zwave	-	Tridium 3.8.107	Not Installed

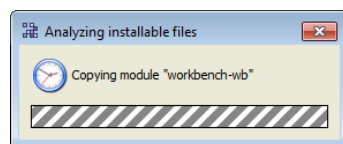
Current free space 56,598 KB To be installed 0 KB Estimated free space after install 56,598 KB

Buttons: Upgrade All Out of Date, Import, Install, Uninstall, Reset, Commit

The *first time* you run 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/3.8.107.

Note this can take several seconds, with a popup similar to the one below.

Figure 73. Copying modules into your software database



NOTE: Copying also occurs whenever you "import" software into the Workbench 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.

Software Manager notes

The following changes are described and noted in other sections of this document, and are summarized here only to assist if you are already familiar with previous Workbench versions.

- Only software modules are shown, versus all "installable parts" including dist files, etc. However, starting in AX-3.8, note that "standard lexicons" are distributed in NiagaraAX

builds as *modules*, named (by convention) as `niagaraLexiconLc-rt.jar` (where `Lc` is a two-character language code). This differs from the previous “lexicon sets” of directories, each with a set of text-based lexicon files. You can still edit and install text-based lexicons (using the platform Lexicon Installer view). However, Lexicon Tools now allow you to make your own lexicon modules, which you install using the **Software Manager** view. For details, see the *NiagaraAX Lexicon Guide*.

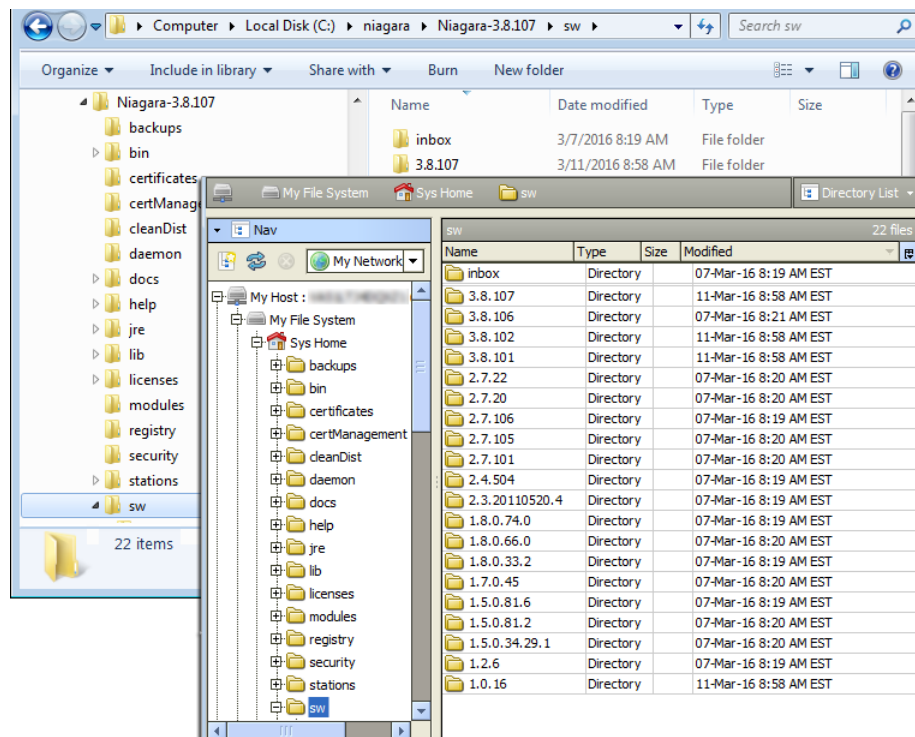
- Module statuses of “Out of Date” and “Not Installed” can now include “Requires Commissioning” too, for example “Out of Date (Requires Commissioning)”. You cannot install such modules without first commissioning (upgrading) the JACE, using the **Commissioning Wizard**.
- In some cases you can install a new module or modules without rebooting the JACE, with its station kept running. This does not apply if upgrading (or downgrading) an existing module on the JACE.
- If needed, you can install an earlier version of a module, versus its latest “Available” version —providing earlier versions are in your Workbench’s software database. See [“Right-click option to install earlier version”](#).

About your software database

The software database for your Niagara Workbench is located under the `Sys Home/sw` sub-directory. 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 using version numbers.

You can see your `sw` subdirectory structure using either Windows Explorer, or in the Workbench Nav tree, `My File System/Sys Home/sw` as shown below.

Figure 74. Software database is everything under `sw`



NOTE: Numbers of subdirectories and version number names in your `sw` subdirectories will be different, this is only a simple example. Do not manually create or rename subdirectories in this area for proper operation—instead, let the **Software Manager** automatically administer this database.

Using the above example of an AX-3.8 installation (3.8.107), this `sw` software database has several versioned subdirectories, which are described in this example as follows:

- `1.8.0.33.2` — Reflects the version of dist files for the Oracle Java 8 “compact3” JRE for JACE controllers (2 files, one for PPC processor JACE controllers, one for ARM processor JACE-8000).
- `1.8.0.74.0` — Reflects the version of dist files for the Oracle Java 8 “Standard Edition” JRE for Windows platforms (2 files, one for 64-bit Windows, one for 32-bit Windows).
- `3.8.107` — Reflects the current *Niagara release, by build number*. Contains numerous Niagara nre “config” and “core” dist files, installed by the “installation tool” Workbench installation option. Also, after the **Software Manager** is first used, the contents of the build’s modules directory (module .jars) are automatically copied here too.
- `3.8.101` — Reflects version of dist files for QNX operating system for JACE controllers, with 4 different dist files.
- `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 will copy the inbox file(s) there.

As an equivalent to the inbox feature, you can use the **Import** button at the bottom of the **Software Manager** to add to your Workbench software database. For details, see [“Software Import”](#).

When you add different-versioned installable files, the number of different subdirectories under your `sw` directory will continue to increase. By default, the Software Manager displays only the most recent version of any module as the **Avail. Version**.

NOTE: You can select to install an older version of any module listed in the Software Manager, if available in your software database. See “Right-click option to install earlier version”. Note that older software files (modules, dists) are also useful in your software database when restoring a backup dist for a JACE, if the backup was made using a previous software release. You use the platform **Distribution File Installer** to restore a backup.

Default module listing and layout

By default, the **Software Manager** lists all of the out-of-date modules on the JACE at the top of the table, then uninstalled modules, and lastly up-to-date modules (sorted alphabetically); see the figure below.

Figure 75. Software Manager default listing out-of-date, then uninstalled modules

File	Installed Version	Avail. Version	Status
svg	Tridium 3.6.44	Tridium 3.7.36	Out of Date (Requires Commissioning)
wbutl	Tridium 3.6.44	Tridium 3.7.36	Out of Date (Requires Commissioning)
web	Tridium 3.6.44	Tridium 3.7.36	Out of Date (Requires Commissioning)
wiresheet	Tridium 3.6.44	Tridium 3.7.36	Out of Date (Requires Commissioning)
workbench	Tridium 3.6.44	Tridium 3.7.36	Out of Date (Requires Commissioning)
aapahp	-	Tridium 3.7.36	Not Installed
aapup	-	Tridium 3.7.36	Not Installed
ak255	-	Tridium 3.7.36	Not Installed
alarmOrion	-	Tridium 3.7.36	Not Installed
andoverAC256	-	Tridium 3.7.36	Not Installed
andoverInfinity	-	Tridium 3.7.36	Not Installed
axisVideo	-	Tridium 3.7.36	Not Installed
bacnetAws	-	Tridium 3.7.36	Not Installed
bacnetOws	-	Tridium 3.7.36	Not Installed
bacnetWts	-	Tridium 3.7.36	Not Installed

- **Out of Date** modules are older than what you have in your PC software database.
- **Not Installed** modules do not exist on the platform, but are in your PC software database.
- **Up to Date** modules are the same (or possibly newer) than that in your PC software database.

NOTE: Both “out of date” and “not installed” modules may also show a “Requires Commissioning” status. This indicates you must upgrade the JACE first, before installing that module version. For more details, see status descriptions for **Software Manager** table columns below.

As needed, you can scroll down the table or click on headers of table columns to resort alphabetically.

Software Manager table columns

The **Software Manager** lists modules using four columns, from left-to-right labeled as follows:

- **File** — File name of locally available module file, or blank if the module is on the remote host only.
- **Installed Version** — Version of the module installed in the remote host, or blank if not installed.
- **Avail. Version** — Latest version of locally available module, or blank if the software is on the remote host only.
- **<unlabeled>** — *Status* of the module in the remote JACE platform. For each module, status is one of the following:

- **Not Installed** — Module is not in remote platform, but is available locally.

Blue text is used for this status.

- **Not Installed (Requires Commissioning)**— Module is not in remote platform, but is available locally. Blue text is also used for this status.

Dependencies prevent you from installing it, unless you first upgrade the JACE, using the Commissioning Wizard.

- **Up to Date** — Module is installed in the remote platform, and is equal to (or higher) than locally available module version.

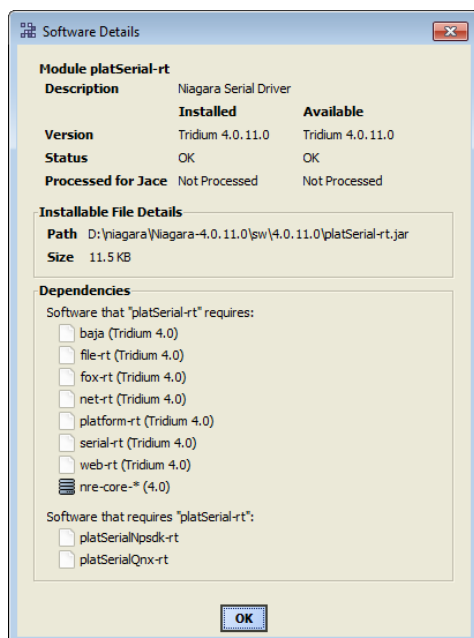
- Out of Date — Module is installed in remote platform, and is *older* than your local version.
Red text is used for this status.
- Out of Date (Requires Commissioning)— Module is installed in remote platform, and is *older* than your local version shown. Red text is also used for this status.
Dependencies prevent you from installing it, unless you first upgrade the JACE, using the Commissioning Wizard.
- Not Available Locally — Module installed in remote platform is not in your software database.
- Cannot Install — Local module is unreadable or has a bad manifest; you cannot install it.
- Bad Target — 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> — Remotely installed software is intended to be replaced with a module having a lower version.
- Install <version> — Module is intended to be installed; it does not currently exist on the remote platform.
- Re-Install <version> — Remotely installed module is intended to be replaced with a module having a the same version.
- Uninstall <version> — Remotely installed module is intended to be uninstalled.
- Upgrade to <version> — Remotely installed module is intended to be replaced with a module having 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.

You can also view *software details* about any item in the table. In addition, you can filter (reduce) the number of software items listed, based on text included in file name or the softwares’ status values. See “Filtering displayed software” for more details.

Software Details

From the **Software Manager**, double-click any module to see a popup dialog with details.

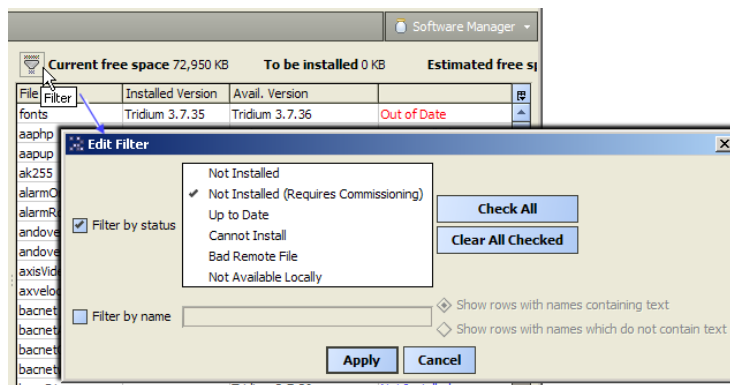
Figure 76. Software Details dialog from Software Manager

Details include a brief module description, comparisons between installed and available module, module file and size, 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.

NOTE: Essentially, 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.

Filtering displayed software

By default, the **Software Manager** lists all remotely installed and locally available modules, which can produce a very large table. A filter control provides an **Edit Filter** dialog, in which you select items for listing, thereby filtering undesired items. See the following figure.

Figure 77. Filter control and dialog to limit displayed modules

You can use either “Filter by status” or “Filter by name”, or a combination of the two.

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 filter by status, you can *check* other statuses *to include* (or clear to omit) the listing of associated items in the table, as follows:

- Not Installed — Modules on your PC that can be installed, but are not in the remote platform.
- Not Installed (Requires Commissioning) — Modules on your PC, but not in the remote platform. The remote JACE must be upgraded (using **Commissioning Wizard**) first.
- Up to Date — Modules on your PC *and* in the remote platform, where the software is not older.
- Cannot Install — Local module is unreadable or has bad manifest, you cannot install it.
- Bad File — Remote module is unreadable or has bad manifest.

NOTE: With status filtering enabled, you can also simply “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.
-

Filter by name

Name filtering lets you include *or* exclude items based on character string portion of module 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 — Only items with file name containing this string.
- Show rows with names which do not contain text — Only items with file name that does not contain this string.

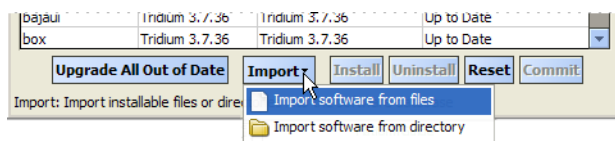
This feature can be useful to filter many modules with common name characters, for example “lon” or “doc” part-named modules.

Software Import

As shown below, an **Import** button at the bottom of the **Software Manager** provides two menu choices for you to add new (or earlier) installable software files (module .jars, .dists) in your software database.

NOTE: Also see, [“Import vs. copy into modules”, page 89](#).

Figure 78. Import choices to bring in file(s) or entire folders



The two import options are:

- Import software from files

This produces the standard **File Chooser** dialog, in which you navigate to the proper location and select one or more software files for import.

- Import software from directory

This produces the standard **Directory Chooser** dialog, in which you navigate to the proper location and select a directory, for inclusion of any contained software files. For example, you might do this for an earlier installed build of Niagara, selecting its “sw” folder, or a portion thereof.

Upon import, the software list is again rebuilt by the Software Manager (popup dialogs appear while software files are copied). Afterwards, any modules that are newer-versioned, or that did not previously exist, will now be represented by default in the software table.

If imported modules are earlier versions, they are also available for installation in the Software Manager.

Import vs. copy into modules

When receiving updated or new module jar files, you have two basic options when copying them to your Workbench PC, as follows:

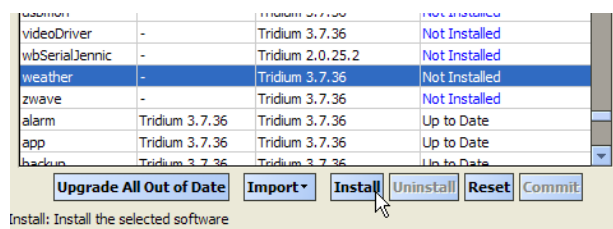
1. Copy directly into your `!/modules` directory. This makes the module(s) available in your Workbench environment, and also available to install in other remote platforms (when the installer runs, the module(s) are also copied into your software database, available for installation). This is the typical choice.
2. Copy into your `!/sw/inbox` directory (or, use the equivalent software **Import** option in the **Software Manager**). In this case, the module(s) are *not* used in your Workbench environment, but *are* available in your 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.

Software actions

As needed, from the **Software Manager** you can take actions on modules, such as install, uninstall, upgrade, downgrade, and re-install. You flag intended actions on software items using *action buttons* near the bottom of the manager’s view pane, as shown below. Action buttons become enabled when you have one or more items selected.

Figure 79. Software Manager action buttons



Included in action buttons are **Reset** and **Commit**. When you reset, all flagged module changes (since the last commit) are cleared. Commit is how you actually launch the flagged changes.

When you **Commit**, one of these two things happens:

- If upgrading (or downgrading) modules, a confirmation popup dialog appears, telling you the station must be stopped and the host rebooted. After the software operation completes, the host is rebooted.

CAUTION: *Before committing, make sure that controlled equipment that might be adversely affected by the JACE's station stopping and then host rebooting (from software changes) is put in a manually controlled state.*

- In many cases, if only installing new module(s), meaning modules not previously installed, the station continues running on that platform. The software is immediately installed.

Upgrade All Out of Date

Whenever one or more local modules are newer than in the opened JACE 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.

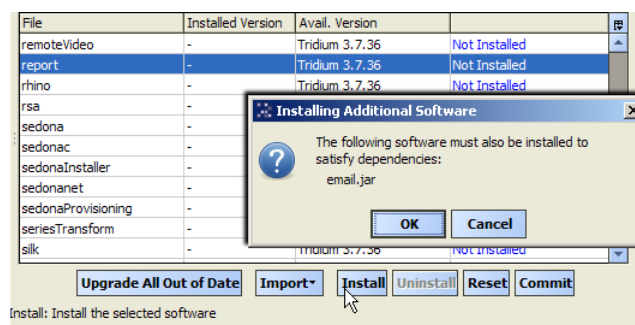
When you click it, the status of all out-of-date modules changes to “Upgrade to <version>,” and the button becomes unavailable. If needed, you can still make additional changes, such as choosing additional modules to install.

Install

This button is available in the **Software Manager** when you have one or more modules selected with a status of “Not Installed.” When you click it, the status of the selected modules changes to “Install <version>,” and if selected again, the button changes to **Cancel Install**.

NOTE: If a selected module has *dependencies* on modules not already installed (or also flagged to install), a dialog appears explaining additional software is needed, as shown below. After you click **OK** from this dialog, the additional modules are flagged, the status of all affected modules changes to “Install <version>”.

Figure 80. Installing Additional Software dialog

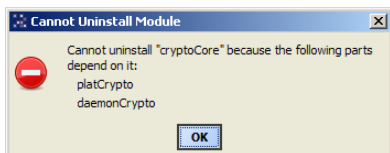


Uninstall

This button is available in the **Software Manager** when you have one or more *installed* modules selected (status of either “Up to Date” or “Out of Date”). If the selected module(s) are not dependencies of other installed modules, when you click Uninstall the module(s) status changes to “Uninstall <version>,” and the button changes to **Cancel Uninstall**.

NOTE: If other installed modules have *dependencies* on one or more modules you selected, a dialog appears explaining the uninstall cannot occur, as shown below. You can then decide if you want to reflag another uninstall, selecting also all modules that are dependent.

Figure 81. Cannot Uninstall dialog



Re-Install, Upgrade, Downgrade

In the **Software Manager**, when you have one or more *installed* software items selected, the “install” button changes to show one of these options.

- **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**.

Commit and Reset

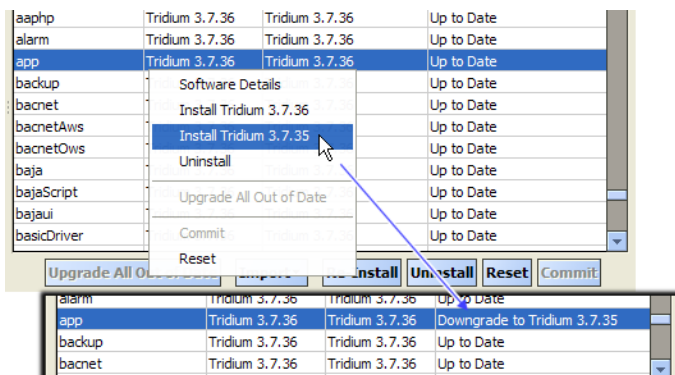
In the Software Manager, when you have one or more *pending* actions in place on software items, the **Commit** button is available. This is how you initiate the software action.

At any time before you commit, you can also click the **Reset** button. This removes all pending actions in place on software items, and makes the **Commit** button unavailable again.

Right-click option to install earlier version

In addition to button-based software actions in the Software Manager, you can also select an *earlier* version of a module to install, providing one is in your Workbench’s software database.

Figure 82. Right-click option to install earlier module version in Software Manager



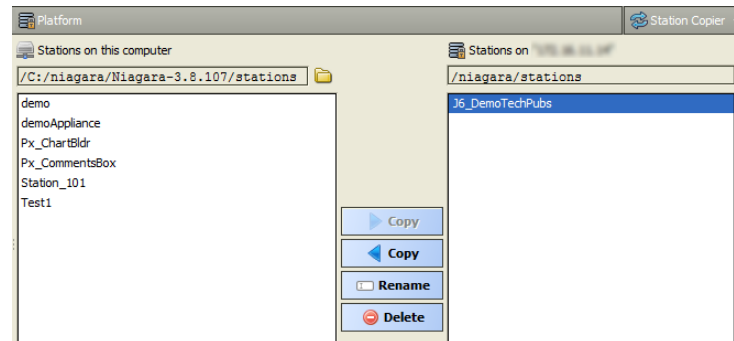
Simply right-click a module row, and from the shortcut menu select any “**Install > vendor > 3. > n > . > nn**” items as shown above. Note if a “downgrade”, a host reboot/station restart will result after you commit.

Station Copier

The **Station Copier** is one of several platform views. You use it to install a station in a remote NiagaraAX platform, as well as make a local backup copy of a remote station (copy its station database and files to your PC). You can also rename and delete stations, either locally or remotely.

You see this 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 83. Station Copier view for remote JACE platform



As shown above, the **Station Copier** view is split into two main areas:

- Stations on your Workbench PC (left side)
- Remote Station on the opened JACE platform (right side)

By default, contents of the !/stations folder is shown on the Workbench (left) side. If you have station folders located elsewhere, click the folder icon for a **Change Directory** dialog, and point the **Station Copier** to that location. That alternate location is remembered the next time you access the **Station Copier**.

Security update 1 changes to Station Copier usage

In the 2013 NiagaraAX update releases (e.g. AX-3.7U1), station password storage changed to become much more secure. Now, in some cases you may need to edit a saved station database (config.bog file) before installing it (copying it) to a remote platform using the **Station Copier**.

NOTE: For any station copied from a host running AX-3.8, config.bog file edits are unnecessary. Such files are saved in a more "portable format" than when using an 2013 update release. Therefore, the following sections apply to hosts running 2013 "update releases" only.

- "When config.bog edits are not needed"
 - "When config.bog edits are needed"
-

For more details on the changed password storage in the 2013 "update 1" releases (e.g. AX-3.7U1), as well as additional AX-3.8 changes, refer to the document *NiagaraAX 2013 Security Updates*. Included is information regarding system upgrades and usage of the platform **Distribution File Installer**, as well as details related to the platform **Station Copier**.

When config.bog edits are not needed

Edits to a config.bog file are not needed for any AX-3.8 station copy. For any AX-3.7U1 station copy, if you want to simply re-install a saved copy back to the same source host that you copied

it from (using the **Station Copier**), typically no edits to the station config.bog file are needed. This also applies to a previous station copy made using a pre-update NiagaraAX release (AX-3.7, AX-3.6, or AX-3.5), or after.

- In the first case, when the remote AX-3.7U1 or later host (typically JACE) starts up the newly copied station, it automatically converts all the passwords in the station to the newer storage formats, and immediately re-saves that station (config.bog file) in its file space.

Now, if you use the **Station Copier** to save that station back to your Workbench, that station database (config.bog file) has all passwords stored in the updated, more secure formats.

- In the second case, the saved AX-3.7U1 station database (config.bog file) already has passwords stored in the updated formats. Providing that you re-install it to the same JACE host that you saved it from, no edits to that config.bog file are necessary.

The two exceptions to this are as follows:

- If after you saved that station, a "clean dist" file was installed on that JACE, and you then installed (or re-installed) an update (e.g. AX-3.7U1) or later release. In this case, note all the "client passwords" in the saved config.bog (e.g., the Password property of the ClientConnection under each NiagaraStation, or the Password of the OutgoingAccount under the EmailService) are no longer valid, even if the JACE hardware is the same. In this case you can simply re-enter all the client passwords and re-save that station.
- If that JACE had since been "downgraded" to a pre-update release, e.g. AX-3.7 or AX-3.6. In this case, that station will not successfully start (as its software doesn't know how to handle the new password storage formats). Before this station is usable on any such host, you need to edit its config.bog file offline in Workbench, re-entering all password property values—both "client passwords" and all station User passwords (under UserService).

When config.bog edits are needed

Before the 2013 update releases such as AX-3.7U1, you could use the **Station Copier** to copy/save a station from one host, and install/copy it to another different (but similar) host, all without any issues. This same ability returned in AX-3.8, providing that the saved station was running AX-3.8.

However, in the AX-3.7U1 update release, because of the different station password storage methods, the following scenarios typically require you to perform some offline editing of the saved station file (config.bog) first, that is before using the **Station Copier** to install/copy it to other different platforms.

- When the saved station is to be replicated on multiple updated (e.g. AX-3.7U1) hosts.

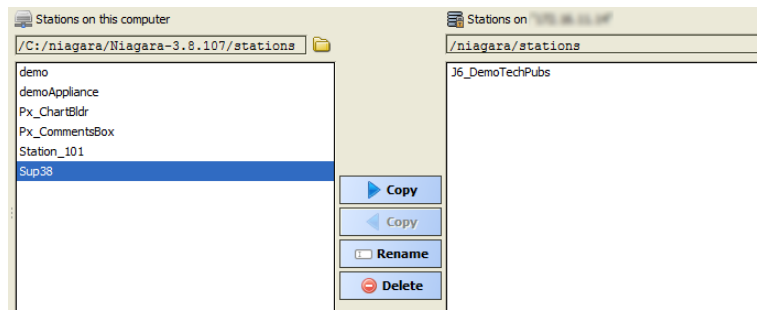
In this case, although all station User passwords (under UserService) will be working (they are considered "portable"), all the "client passwords" in the station will not work (unless installed back to the original host). Examples of these passwords are the Password property of the ClientConnection under each NiagaraStation, or the Password of the OutgoingAccount under the EmailService.

These passwords in the config.bog will not work because they are encrypted based on files in the platform's !security folder that are different (and unique) to each JACE controller.

For related details, refer to the *NiagaraAX 2013 Security Updates* engineering notes document, including section "Making modifications to archived station files".

Station copy direction

The copier works in either direction. In other words, click a station on one side (to copy to the other side). When you click a station, the station is selected (highlighted) and the appropriate **Copy** button, by direction, becomes enabled to clarify the source and target. See the figure below.

Figure 84. Copy direction by station side selection

To perform the following station operations, you:

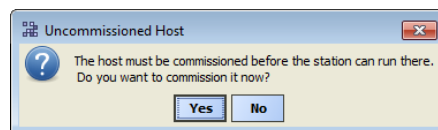
- Click in left side for a copy from local-to-remote.
Do this to install a station in a JACE. This is called “installing” in remaining document subsections.
- Click in right side for a copy from remote to local.
Do this to make a local backup copy of a station, saved to your Workbench computer. This is described as a “backup” in the following document subsections.

When you click **Copy**, the **Station Transfer Wizard** appears and guides you through the steps of the station transfer process.

Station Copier dependencies check

The **Station Copier** checks, whenever installing a station, to determine if the target JACE 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 JACE 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 JACE needs commissioning, and provides the option to start the **Commissioning Wizard**. See figure below.

Figure 85. Selected station cannot be installed without first commissioning the JACE.

Click **Yes** to start the **Commissioning Wizard**, or **No** to simply return to the Station Copier.

This may occur when trying to install a station in a new, uncommissioned, “out-of-the-box” JACE controller, or in a JACE that has been converted from AX to N4, but still not yet commissioned. Despite documentation to first commission any new JACE using the platform **Commissioning Wizard**, this continues to occasionally come up. For complete details, see the *JACE NiagaraAX Install & Startup Guide*.

If all modules needed by the station are found on your Workbench computer, the Station Transfer Wizard starts normally. However, upon reaching the “Modules step”, in some cases you may see a caution. For further details see “Modules step”.

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. 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).

NOTE: Use **Cancel** if you need to make a different selection to copy; this reruns the wizard.

The wizard's **Finish** button is enabled only in the final step. When you click **Finish**, the related operations begin, and you see progress updates in the **Transferring Station** dialog. When complete, click **Close** in the dialog to exit the wizard.

NOTE: In the unlikely case where the source station config.bog file is currently in use ("locked"), the wizard opens in a state where you must **Cancel** to exit (no other steps are given).

- If installing a station, the source config.bog is locked if it contains unsaved changes (it is being edited elsewhere in Workbench). After saving changes, you can try to copy again.
 - If backing up a station, the source config.bog is locked if currently in process of being saved. You can retry the copy later.
-

Name step

The first step in the **Station Transfer Wizard** is to confirm the name (or type a new name) for the copied station directory.

Figure 86. Station Transfer Wizard dialog, name step



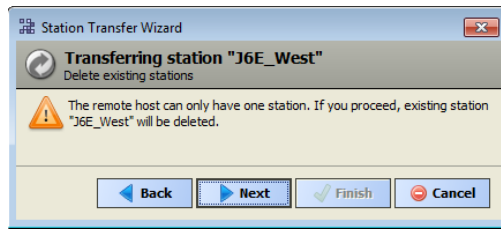
Default name is the station directory being copied. If you rename the station, it will be identical to the source (copied) station in every way *except* name of its station directory.

Delete step

NOTE: This step is skipped for any station backup, or if a station install in either of these cases:

- No existing station exists on the target.
 - The existing station is named the same as the one you are installing.
-

This step occurs because *all* JACE platforms have a support limit of one (1) installed station. The delete step simply cautions you that the existing station will be deleted, as shown below.

Figure 87. Station Transfer Wizard dialog, delete step

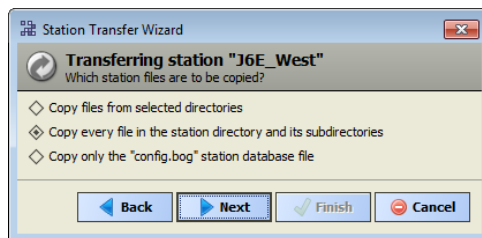
NOTE: The entire remote station directory (all subdirectories and files) is deleted when the station install starts. If unsure, it may be best to **Cancel**, then backup the remote JACE station first.

The next step is the “Content step”.

Content step

NOTE: This wizard step is skipped if the source station consists of *only* a config.bog file.

After the “name step” and possibly “delete step”, the wizard asks you to select what station files to copy, with the default selection being “all” files and folders under that station directory, as shown below.

Figure 88. Station Transfer Wizard dialog, content step

The three possible selections are:

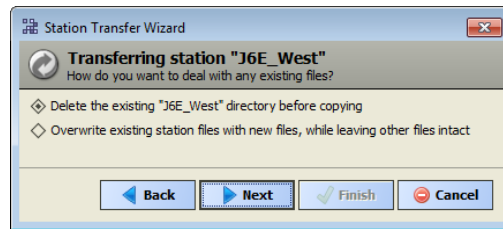
- Copy files from selected directories (not shown if source station has no subdirectories).
If you select this, a later “Details step” allows you to select the source subdirectories.
- Copy every file in the station directory and its subdirectories.
- Copy only the “config.bog” station database file.

The next step is the “Disposition step”.

Disposition step

NOTE: This wizard step occurs only when an identically-named target station already exists.

If the target station already exists, a disposition step asks what is to be done with it, as shown below.

Figure 89. Station Transfer Wizard dialog, disposition step

The two possible selections are:

- Delete existing station directory before copying.
- Overwrite existing station files with new files, while leaving other files intact.

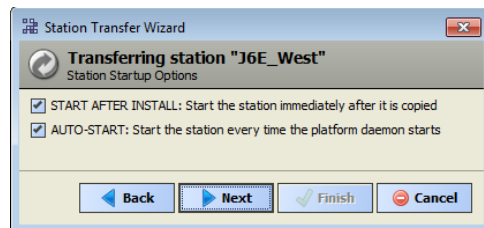
If you previously selected “copy everything” from the “Content step”, the default pre-selection is the first (delete existing station directory). Otherwise, the second selection (overwrite) is pre-selected.

The next step is the “Station settings step”.

Station settings step

NOTE: This wizard step is skipped for any station backup.

This step specifies the station’s Auto-Start setting.

Figure 90. Station Transfer Wizard dialog, station settings

Two items are listed:

- START AFTER INSTALL: Start the station immediately after it is copied.
- AUTO-START: Start the station every time the platform daemon starts.

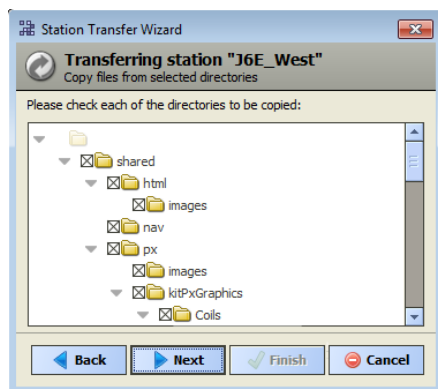
Auto-start is one of two station settings for any station, as specified in the **Application Director** view by using “start checkboxes”. See [“Application and output controls”, page 22](#).

Typically, you enable both settings and go to the next step, either the “Details step” or “Modules step”.

Details step

NOTE: This wizard step is skipped for any station backup, as well as for a station install—unless you selected “copy selected directories” in the “Content step”.

Figure 91. Station Transfer Wizard dialog, details step



As shown above, 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 (note that if present, a station’s `alarm` and `history` folders are unselectable).

For any selectable folder, click to toggle it as either selected (with X) or unselected (no X).

Modules step

This wizard step is skipped if a station backup, or if all modules required by the station to be installed are already in the JACE controller. In this case, you see either the “Stop station step” or “Review step” instead

This step occurs if the target platform is missing one or more of the modules required by the station being copied (installed). It lists the missing modules/versions that will be installed during the station copy operation. If included, this is the *final step* before the station copy process starts.

NOTE: Dependencies of the missing modules are compared against the software that is already installed in the target platform. The Station Copier looks for versions of those missing modules in your User Home software database that can be installed *without* re-commissioning the target platform, by default.

There are two possible results when the wizard reaches this step:

- Station can be installed with most current modules
- Station can be installed with “out of date” modules

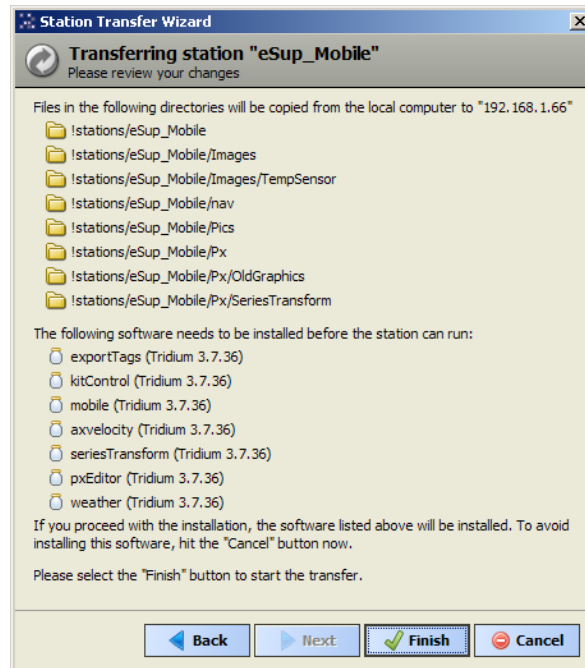
In either case, to continue you click either:

- **Finish** — Start the local-to-remote copy, including installation of the listed modules. Progress updates appear in a “Transferring station” dialog.
- **Cancel** — Exit from the Station Transfer Wizard, then either select another station to install, or if a JACE upgrade is possible (and you have purchased an upgrade license for it) run the **Commissioning Wizard** to upgrade the controller, including the installation of a station.

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, as shown below.

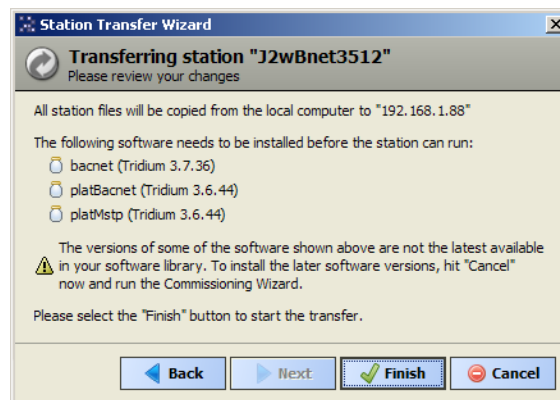
Figure 92. Station install example, all missing modules are most current versions



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 station install. A dialog explains that you can use the **Commissioning Wizard** to upgrade the JACE.

Figure 93. Station install with one or more missing modules not current versions



This may occur after a "point release" of Niagara, say AX-3.8, where you have previously imported the software database of an AX-3.7 installation.

For related details, see Software Manager topics ["About your software database"](#) and ["Software Import"](#). Also see ["Upgrading a JACE"](#).

Stop station step

You can see this wizard step in any of these scenarios:

- You are copying the station running in a remote platform to your local computer, and you selected either “copy files from selected directories” or “copy only the config.bog station database file” in the previous “Content step”. Note this step is skipped if you elect to “copy every file in the station directory and its subdirectories”. However, a station save occurs before the station copy transfer starts.
- If installing a “same-named” station.

This step reminds you that the station must be stopped while it is copied, as shown below.

Figure 94. Station Transfer Wizard dialog, stop station step



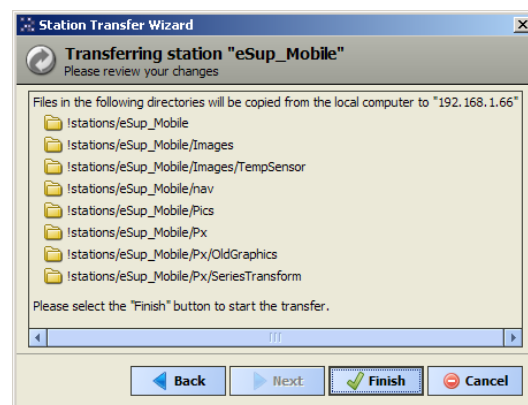
Click **Next** to go to the “Review step”.

Review step

NOTE: This wizard step is skipped when installing a station where additional modules are required. (Instead, the “Modules step” provides the **Finish** button.)

This step provides a summary of choices from previous steps, and a **Finish** button to begin the station copy process.

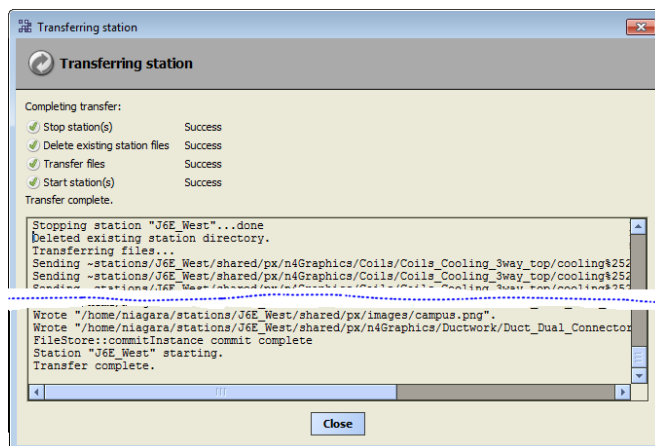
Figure 95. Station Transfer Wizard dialog, review step



As shown above, if you select only *specific* station subdirectories to copy (from the “Details step”), they are listed. If needed, click **Back** to make changes, or click **Finish** to begin the copy process and observe progress in the “Transferring station” dialog.

Transferring station

After clicking **Finish** in the “Modules step” or “Review step” of the Station Transfer Wizard, the station copy process begins and updates appear in a this dialog, as shown below.

Figure 96. Station Transfer Wizard, Transferring station (copy) process

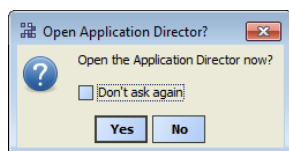
Depending on the type of copy, the following operations may be included in this process:

- If installing a station (copy from *User Home-to-daemon User Home*):
 - Stop all stations — whenever modules must be installed.
 - Stop one station — any JACE where same station is being reinstalled.
 - Delete station(s) — if you chose to delete station in the “Disposition step”, or if a station needs to be deleted to stay under maximum number of stations (only one for any JACE platform)
 - Transfer files — includes station and module files (actual copy portion).
 - Start station — if a station had to be stopped (module installation), or if you chose to start the station in the “Station settings step”.
- If backing up a station (copy from *daemon User Home-to-local User Home*):
 - Save station — whenever remote station is currently running.
 - Transfer files — includes station files (actual copy portion).

NOTE: A popup explaining that the existing station must be saved (if a backup) or stopped (if installing) may appear for a few seconds. Following, and during execution of the various operations, a **Cancel** button is available. If you click Cancel before all operations complete, the installation (or backup) is not valid.

After all operations are finished, a **Close** button is available and the last update in the dialog is “Transfer complete.” Click **Close** to exit the wizard.

By default, after *installing* a station, the wizard exits with a popup asking if you wish to switch to the **Application Director** platform view.

Figure 97. Switch to Application Director popup

Because it is a good idea to observe a station’s output upon first startup, you typically select **Yes**. To automatically switch to the **Application Director** after installing a station, click the checkbox to “Don’t ask again” before selecting **Yes**. Then, you do not see this popup again.

Renaming stations

The **Station Copier** lets you rename any station, either on your local PC (left side) or a remote platform (right side).

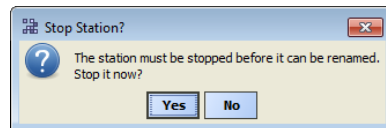
As shown, a Rename dialog appears when you select a station and click **Rename**.

Figure 98. Rename station dialog

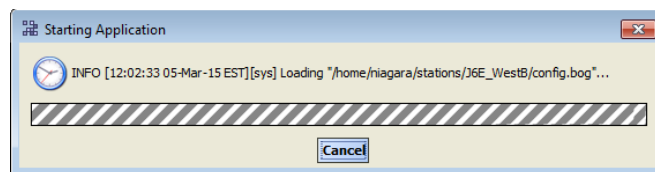


NOTE: Be careful when renaming stations, as there is *no undo*. Furthermore, please note the following:

- Any running station that is renamed must first be stopped—a confirmation popup dialog informs you of this after you enter the new station name and click OK.



After the station stops it becomes renamed, and then automatically restarts. A series of other popups appear, each showing a station startup message.



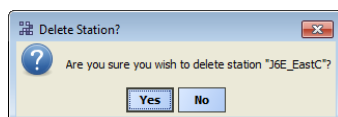
- If a renamed running station is already included in the NiagaraNetwork of other stations, its corresponding NiagaraStation component *will remain “down” until renamed* to match the new name. Thus, all child components (Niagara proxy points and so on) will also be down until this is done. In addition, 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 local (left side) stations, or when initially configuring a job site network, such as when installing (copying) a station.

Deleting stations

The **Station Copier** lets you delete *any* station, either on your local PC (left side) or a remote platform (right side).

Figure 99. Confirm delete station dialog



As shown, a confirmation dialog appears when you select a station and click **Delete**.

NOTE: Be careful when deleting stations, as there is *no undo*. Furthermore, note the following:

- The entire selected station directory gets deleted, 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 opportunity to **Abort**).
 - Also in general (as a precaution), *before* deleting a running station, it is generally recommended to make a backup copy first. If desired, when backing up you can rename it using some “temp” convention to flag it for later housekeeping.
-

TCP/IP Configuration

TCP/IP Configuration is one of several platform views. Typically, you use it to initially configure a remote controller’s TCP/IP settings.

NOTE: If connected to any Windows-based platform, all settings in this view are read-only. You typically use the Windows Control Panel for making these changes on a PC.

- [Configuring TCP/IP, page 103](#).
- [TCP/IP Host fields, page 107](#)
- [TCP/IP DNS fields, page 107](#) (host-level for JACE controller platforms only)
- [TCP/IP Interface fields](#)

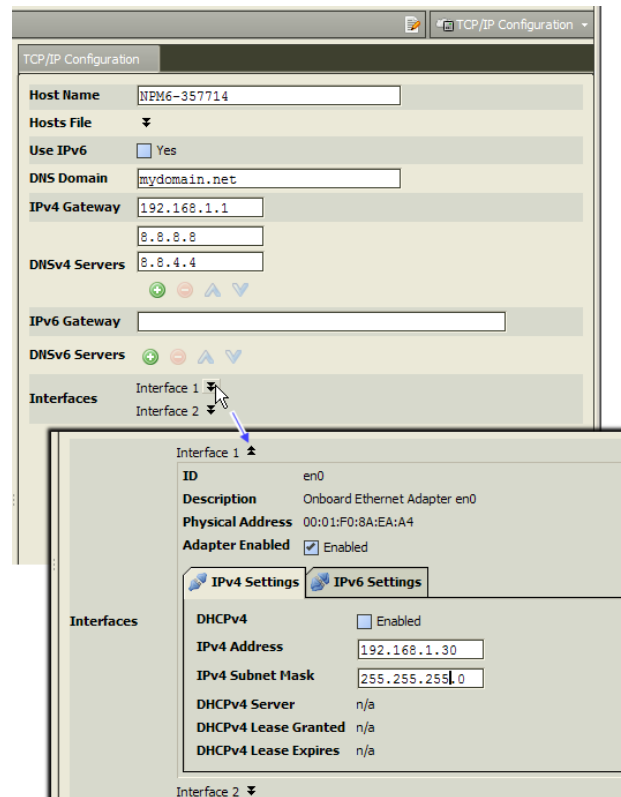
Configuring TCP/IP

Configuring TCP/IP communication settings is a task for the systems integrator when initially setting up a controller.

Prerequisites:

Perform the following steps:

- Step 1 Open a secure connection to the platform.
- Step 2 Expand the **Platform** container in the Nav tree and double-click the **TCP/IP Configuration** container.



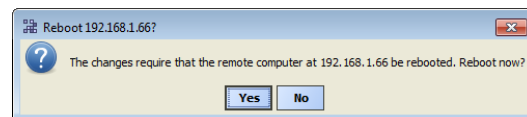
The system displays the main TCP/IP properties.

Step 3 Click the drop-down arrows to expand a group of properties.

To save yourself time when making multiple changes, enter *all* changes before you continue.

Step 4 When you finish the configuration, click **Save**.

The system displays:



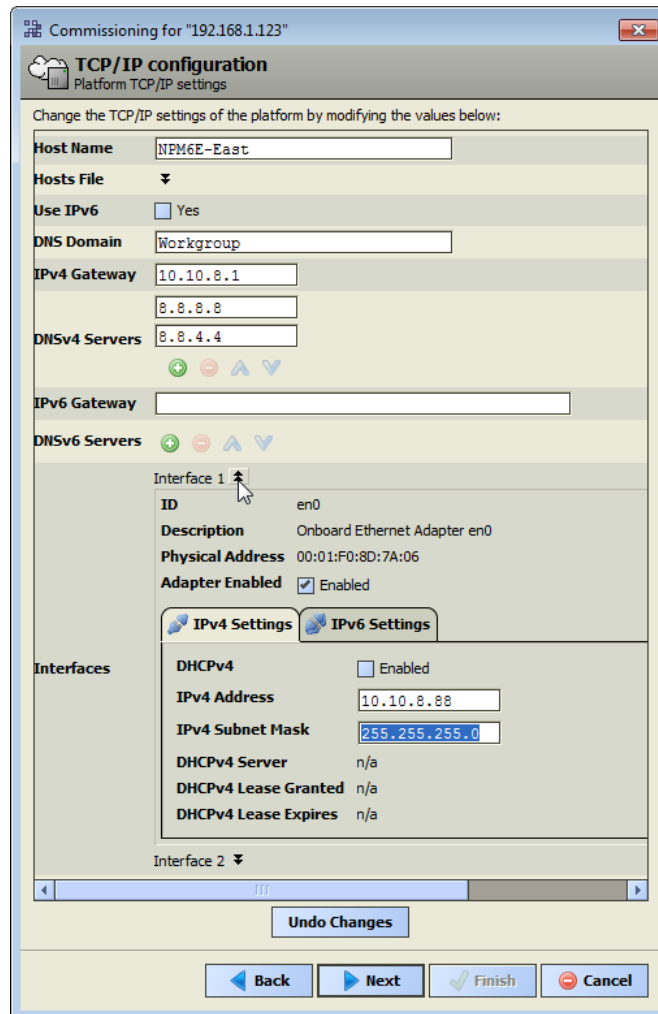
Step 5 Reboot the controller for the changes to take effect.

Configure TCP/IP network settings

This step sets up the properties required for client stations to connect to this server. IPv6 support is available, however this document focuses on IPv4 configuration.

Step 1 If you have not already done so, click **Next**.

The **TCP/IP configuration** window opens.




Step 2 Review, and if needed adjust other TCP/IP settings, which (in usual order of importance) include:

- **Hostname** — The default may be “localhost,” or enter another name you want to use for this host.

NOTE: 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 default.

- **Hosts File** — Click control to expand edit field. Format 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. The IP address should be placed 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.
- **Use IPv6** — Enable if using this feature.
- **DNS Domain** — Enter the name of network domain, or if not applicable, leave blank.
- **IPv4 Gateway** — The IP address for the device that forwards packets to other networks or subnets.

- **DNSv4 Servers** — Click the  add button for a field to enter the IPv4 address of one or more DNS servers.
- **IPv6 Gateway** — Use this if you enabled **Use IPv6**.

Step 3 To add a line, click at the end of the last line and press **Enter**.

Step 4 Type in the required data on the new line.

To return to see all TCP/IP settings, click the control to contract the edit field when done.

Step 5 Review the settings for **Interface 1** on the **IPv4 Settings** tab, which include the temporary factory-shipped IP address.

Step 6 Do one of the following:

- If the network supports DHCP, enable it (click **DHCP Enabled**). In this case, the **IPv4 Address** and **Subnet Mask** fields become read only.
- Otherwise, assign the host a unique **IPv4 Address** for the network you are installing it on. No other device on this network should use this same address. Include the appropriate **Subnet Mask** used by the network.

CAUTION: In general (for stability), static IP addressing is recommended over DHCP. *Do not enable DHCP unless you are certain that the network has DHCP servers!* Otherwise, the host may become *unreachable over the network*.

Step 7 To define a second interface, click the down arrows.

NOTE: JACE-3,-6,-7 controllers have two Ethernet ports, where **Interface 2** is available for configuring the LAN2 (secondary) Ethernet port. By default, this port is *disabled*, that is without a default address. Intended usage is for:

- Isolating a driver's Ethernet traffic from the primary (LAN1) interface, OR
- 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 host without disconnecting it from the 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 set as DHCP. If enabling LAN2, you typically specify another (network) static IP address and the appropriate subnet mask.

Also note the following:

- If enabling both LAN ports, the LAN1 IP address and LAN2 IP address must be 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 is an *invalid* configuration, as both addresses are on the *same subnet*.
- A host *does not* provide IP routing or bridging operations between different Interfaces (LAN ports, GPRS, dialup).
-

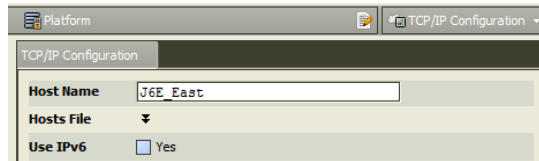
Step 8 To reset all settings (all interfaces) back to their original pre-step values, click **Undo Changes**.

Step 9 Click the **Next** button to go to the next step.

TCP/IP Host fields

The top of the **TCP/IP Configuration** view provides the platform's TCP/IP host settings.

Figure 100. Hosts fields on platform TCP/IP Configuration view



These available host fields are as follows:

- **Host Name**
Synonymous with “computer name,” this is a string that can be processed by a DNS server to resolve to an IP address. On Windows-based systems, this hostname is the computer's identification in its workgroup or domain. If using hostnames, each Niagara platform should have a *unique* hostname.
- **Hosts File**
The hosts file is a standard TCP/IP hosts file, where each line associates a specific IP address with a known hostname. To review, click the expand control to see all entries.
If the JACE controller, you can edit its host file.
 - To *add* an entry, click at the end of the last line and press Enter.
Then type the IP address, at least one space, then the known hostname.
 - To *delete* an entry, drag to highlight the entire line, then press Backspace.
Click the expand control again to collapse the Hosts File editor.
- **Use IPv6**
Default is No (unchecked). If set to Yes (checked), Niagara (platform daemon and station) respond to IPv6 requests, that is, creates IPv6 server sockets (daemon) and IPv6 fox multi-cast sockets.

TCP/IP DNS fields

If connected to a JACE controller, the DNS and gateway settings are also “host-level” parameters in the TCP/IP Configuration view, as shown below.

NOTE: If a Windows-based host, DNS and gateway settings are available under each Interface section.

Figure 101. Host-level fields for any JACE controller includes DNS and gateway

The screenshot shows the 'TCP/IP Configuration' window. The fields are as follows:

- Host Name:** J6E_North
- Hosts File:** (dropdown arrow)
- Use IPv6:** ☐ Yes
- DNS Domain:** mydomain.net
- IPv4 Gateway:** 172.12.69.113
- DNSv4 Servers:** 8.8.8.8, 8.8.4.4 (with add, delete, and move icons)
- IPv6 Gateway:** (empty field)
- DNSv6 Servers:** (empty field with add, delete, and move icons)
- Interfaces:** Interface 1, Interface 2 (dropdown arrows)

The available fields for JACE controllers are as follows:

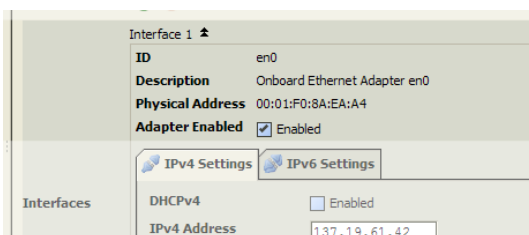
- **DNS Domain**
The TCP/IP Domain Name System (DNS) domain this host belongs to, if used.
- **IPv4 Gateway**
The IP address of the router that forwards packets to other IPv4 networks or subnets. A valid gateway address is required in multi-station (JACE) jobs to allow point discoveries under NiagaraNetworks.
- **DNSv4 Servers**
The IP address of one or more DNS servers (if available), where each can automate associations between hostnames and IPv4 addresses. Included are icon-buttons to Add (to enter IP address of server), Delete, and move Up/Down (to set the DNS search order).
- **IPv6 Gateway**
The IPv6 address for the router that forwards packets to other IPv6 networks or subnets.
- **DNSv6 Servers**
The IPv6 address for one or more IPv6 DNS servers (if available), where each can automate associations between hostnames and IPv6 addresses. Included are icon-buttons to Add (to enter IP address of server), Delete, and move Up/Down (to set the DNS search order).

TCP/IP Interface fields

For each Ethernet port on the connected platform, the **TCP/IP Configuration** platform view provides an expandable **Interface n** section.

All compatible JACE controllers have two Ethernet ports: LAN1 and LAN2. In the **TCP/IP Configuration** view, they are listed as **Interface 1** (en0) and **Interface 2** (en1).

NOTE: Some controllers can have an optional “WiFi” adapter installed. For JACE-700 controllers, it appears in the **TCP/IP Configuration** view as yet a third Interface 3 (bc0).

Figure 102. TCP/IP Interface fields, top properties

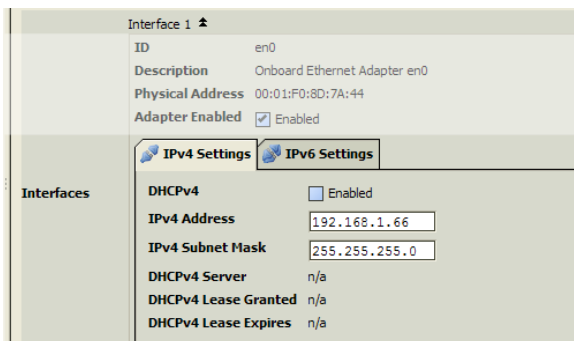
As shown above, each **Interface** has the following properties at the top:

- ID
A read-only OS identifier for the hardware interface, such as “en0” if a JACE controller, or if a Windows platform, either a 128-bit GUID (globally unique identifier) or a Windows network connection name, such as “Local Area Connection 2”.
- Description
A read-only text string such as “Onboard Ethernet Adapter en0” for a JACE controller, or “Intel(R) PRO/100 VE Network Connection” for a Win32-based host, describing a NIC model.
- Physical Address
The unique 48-bit MAC address of the Ethernet adapter, in six two-hexadecimal digits. For example, for the “en0” Interface 1 port of a JACE controller: 00:01:F0:80:13:E6
- Adapter Enabled
Checkbox to specify whether the Ethernet port is usable.

Below the above properties, each **Interface** has two separate tabs, as follows (each with properties):

- IPv4 Settings
- IPv6 Settings

IPv4 Settings

Figure 103. IPv4 tab for Interface of JACE controller, in platform TCP/IP Configuration view

The following properties are on the **IPv4 Settings** tab of the selected Interface:

- DHCPv4

NOTE: Only *ONE* adapter of any JACE controller may have DHCP enabled. A checkbox to specify DHCP (Dynamic Host Configuration Protocol) instead of static IP addressing. Successful use requires a DHCP server installed on your network. If enabled, other interface fields such as IP Address and Subnet Mask become read-only, as these are assigned by the DHCP server after the platform reboots. In general (for stability), static IP addressing is recommended over DHCP. If configuring for DHCP it is recommended that you reserve a specific, fixed IP address for this host in the network's DHCP server/router configuration, noting the MAC address of this adapter as shown above.

CAUTION: Do not enable DHCP unless sure that your network has one or more DHCP servers! Otherwise, the controller may become unreachable over the network.

- DNS Domain

(Windows hosts only) The TCP/IP Domain Name System (DNS) domain the host belongs to, if used.

- IPv4 Address

The “static” IP address for this host, unique on your network.

Be careful to understand the following:

NOTE:

- If enabling multiple ports, note that IP address must be 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 is an invalid configuration, as both addresses are on the same subnet.
 - A JACE controller *does not* provide IP routing or bridging operation between different Interfaces (LAN ports, GPRS, dialup, WiFi).
-

- IPv4 Gateway

(Windows hosts only) IP address for the device that forwards packets to other networks or subnets.

- IPv4 Subnet Mask

The “static” IP subnet mask used by this host.

- DHCPv4 Server

Applies only if DHCP is enabled. Shows read-only address of the DHCP server from which this host last obtained its IP address settings.

- DHCPv4 Lease Granted

Applies only if DHCP is enabled. Shows a read-only timestamp of when the DHCP lease started.

- DHCPv4 Lease Expires

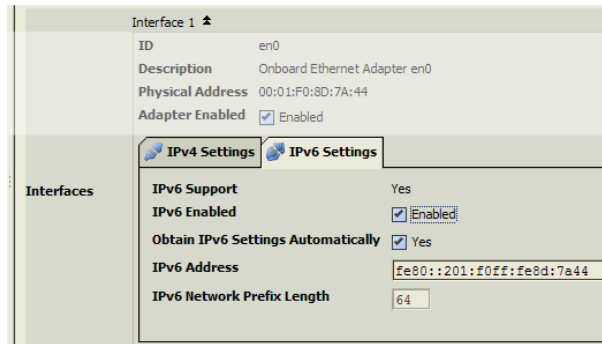
Applies only if DHCP is enabled. Shows a read-only timestamp of when the DHCP lease will expire, and will need renewal.

- DNSv4 Servers (DNS Servers)

(Windows hosts only) The IP address for one or more DNS servers, each of which can automate associations between hostnames and IP addresses. Included are icon-buttons to Add (to enter IP address of server), Delete, and move Up/Down (to set the DNS search order).

IPv6 Settings

Figure 104. IPv6 tab for Interface of JACE controller, in platform TCP/IP Configuration view



The following properties are on the **IPv6 Settings** tab of the selected Interface:

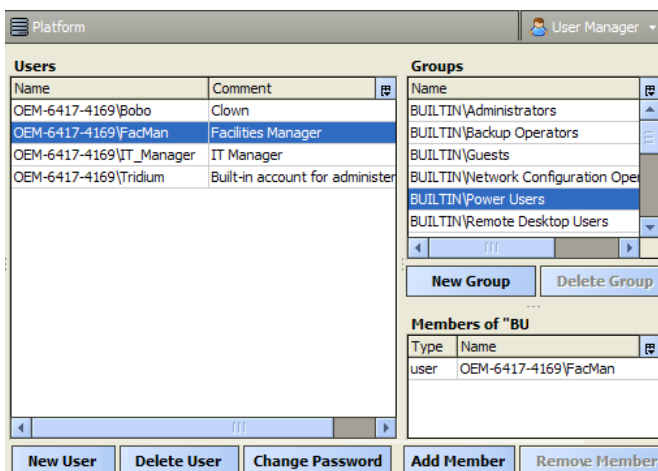
- **IPv6 Support**
Yes or No, as read-only. Indicates if host platform's OS supports IPv6.
- **IPv6 Enabled**
Checkbox for Enabled, where default is cleared (disabled). If a Windows host, this indicates if it is configured with the IPv6 protocol.
- **Obtain IPv6 Settings Automatically**
Checkbox for Enabled (default). Provides for "auto-configuration" of IPv6 address, if acceptable. If enabled on a JACE controller, the next two properties are read-only. If cleared, the two properties below must be entered manually.
- **IPv6 Address**
The host's IP address in IPv6 format, to be unique on its network.
- **IPv6 Network Prefix Length**
The number of left-most contiguous bits of the IPv6 address (in decimal) that compose the subnet prefix.
- **DNSv6 Servers**
(Windows hosts only, providing host's OS has IPv6 enabled) Read-only IPv6 address for one or more DNS servers, each of which can automate associations between hostnames and IPv6 addresses.

User Manager

The **User Manager** is one of several platform views, available only when connected to a Windows-based host. This view allows you to manage Windows OS user and group accounts local to that host (which otherwise would require accessing "Administrative Tools" in Windows on that host).

NOTE: You need "admin-level" platform access in order to change any user settings. When connected to the platform via a "user-level" login, you can review settings, but none of the buttons in this view are available, nor are drag-and-drop actions possible. See levels of platform access for related details.

Figure 105. User Manager for remote Win32-based host



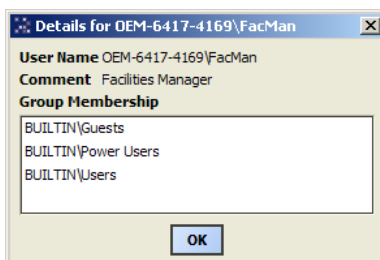
As shown here, the view has two main sides. Users are listed in a users table on the left side. Groups are listed in a groups table the right side. In addition, a lower "membership" table shows all members of any currently selected group. Buttons below each side provide launch dialogs in which you can add or delete a user or group, or change password for a selected user. The following sections provide more details:

- **Users management**

In the users side of the User Manager, click in the users table and buttons below to perform various Windows user management tasks. You can review, add, and delete users, and change passwords. You can also drag and drop users into groups.

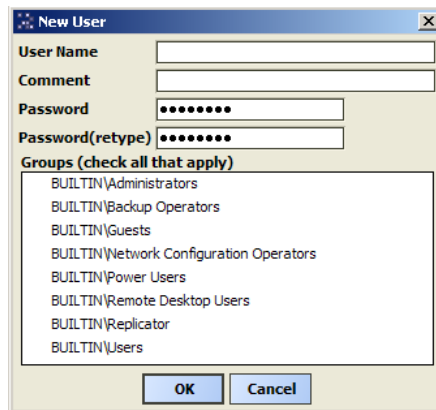
- **Review user** — Double-click any existing user for a **Details** dialog, as shown.

Figure 106. Details dialog for Windows user



This displays the user's account name, comment, and group memberships (including domain groups).

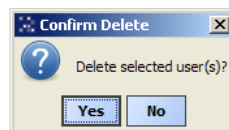
- **Add user** — Click the **New User** button for a New User dialog, as shown here.

Figure 107. New User dialog

 A Windows-style dialog box titled "New User". It contains four text input fields: "User Name", "Comment", "Password", and "Password(retype)". The "Password" and "Password(retype)" fields are masked with dots. Below these fields is a section titled "Groups (check all that apply)" containing a list of built-in Windows groups: BUILTIN\Administrators, BUILTIN\Backup Operators, BUILTIN\Guests, BUILTIN\Network Configuration Operators, BUILTIN\Power Users, BUILTIN\Remote Desktop Users, BUILTIN\Replicator, and BUILTIN\Users. At the bottom are "OK" and "Cancel" buttons.

In this dialog you must type a user name and password (text in both password fields must match). You can also type a comment, typically a full user name or description. Click in the groups checklist to designate which groups the new user should have membership.

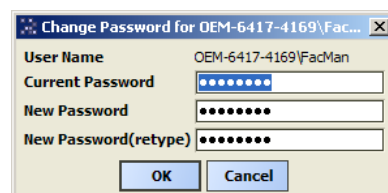
When you click **OK**, the new user is added and appears in the user table.

- **Delete user** — Click to select one or more users (press Ctrl and click to select multiples). Then click the **Delete User** button for a confirmation dialog, as shown.

Figure 108. Confirm Delete dialog

 A small Windows-style dialog box titled "Confirm Delete". It features a question mark icon and the text "Delete selected user(s)?". At the bottom are "Yes" and "No" buttons.

When you click **OK**, the selected user(s) is deleted and removed from the user table.

- **Change user password** — Click to select a user, then click the **Change Password** button for a popup dialog.

Figure 109. Change Password dialog

 A Windows-style dialog box titled "Change Password for OEM-6417-4169\FacMan". It contains four text input fields: "User Name" (pre-filled with "OEM-6417-4169\FacMan"), "Current Password", "New Password", and "New Password(retype)". The "Current Password", "New Password", and "New Password(retype)" fields are masked with dots. At the bottom are "OK" and "Cancel" buttons.

You must type the current user's password, then the new password twice (text in both new password fields must match). When you click **OK**, the password for that user is changed to your new password.

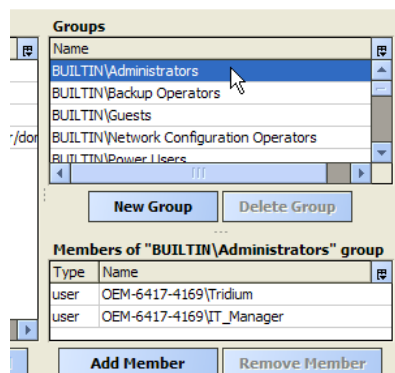
- **Drag and drop** — You can drag-and-drop rows from the users table on top of a row in the groups table. This adds the selected user(s) to the target group, without any popup dialog. Or, if a single group is already selected, you can drag-and-drop user rows into the lower membership table for that group. This adds the selected user(s) to that group, and updates the membership table.
- **Groups management**

In the groups side of the User Manager, click in the groups table, membership table, and buttons below to perform various Windows group management tasks. You can review, add, and delete groups, and in any group, you can add or remove members.

NOTE: For a Windows host, only domain groups are shown in which the current user is a member vs. all possible domain groups. Previously, it was found that on a large domain (e.g. a corporate domain with thousands of domain groups), platform daemon issues resulted that prevented proper loading of views such as the User Manager. This could also affect User Authentication dialogs launched from the Platform Administration view.

- **Review group** — Click any existing group in the User Manager to see user members in the table below, as shown.

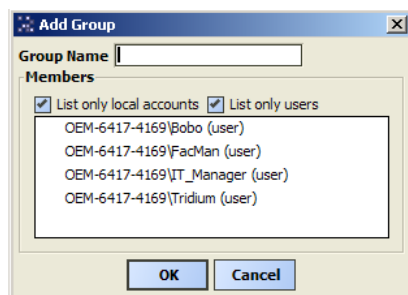
Figure 110. Select group to see membership



All users for the selected group are shown.

- **Add group** — Click the **New Group** button for a popup dialog, as shown

Figure 111.



Shows only those domain groups in which the current user is a member. In this dialog you must type a name for the new group. Click in the users checklist to designate which Windows users the new group should have as members.

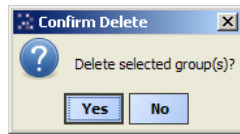
By default, the users checklist is "filtered" to reduce entries as follows:

- **List only local accounts** — Any domain users and groups do not appear.
- **List only users** — Groups do not appear.
-

As needed, click these checkboxes to add or remove these choices in the users checklist. When you click OK, the new group is added and appears in the groups table.

- **Delete group** — Click to select one or more groups (press Ctrl and click to select multiples). Then click the **Delete Group** button for a confirmation dialog, as shown.

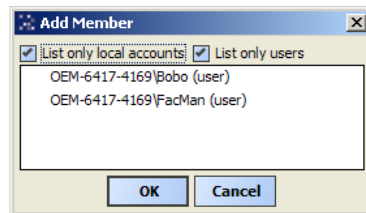
Figure 112. Confirm Delete dialog



NOTE: You cannot delete any Windows "Built-In" group.

- **Add member** — Click to select a group, then click the **Add Member** button.

Figure 113. Add Member dialog



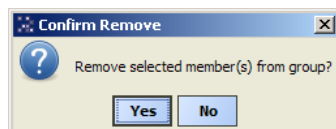
Only users not already members of this group are listed. Click in the users checklist to designate which Windows users the group should have as members.

By default, as in the **New Group** dialog, the users checklist is filtered to not list domain users and groups. If needed, click these checkboxes to add or remove these choices in the users checklist. When you click **OK**, the group's membership is updated with the member(s) you added.

NOTE: You can also drag and drop users (rows in users table) onto groups (rows in groups table).

- **Remove member** — Click to select a group, then click in the membership table, select one or more users. With the user(s) selected, click the **Remove Member** button for confirmation dialog.

Figure 114.



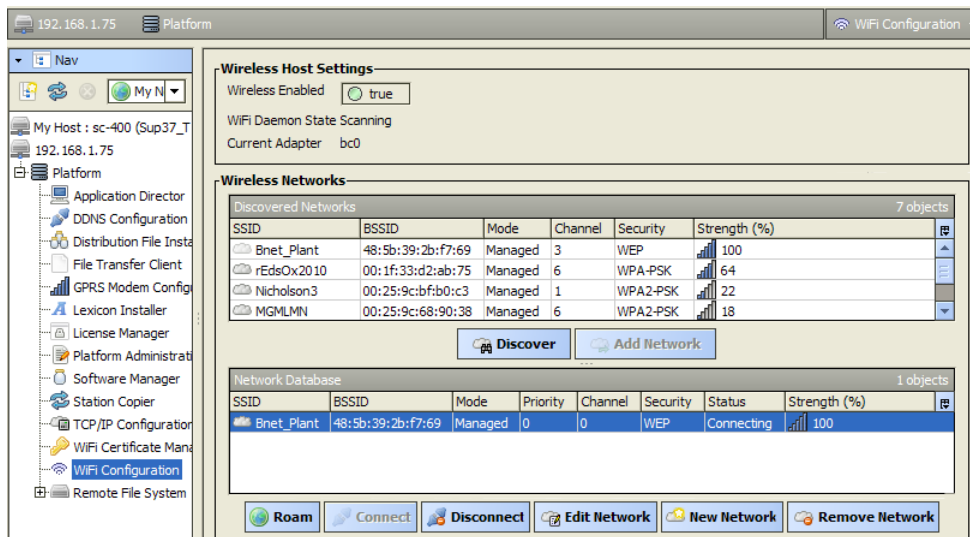
When you click **OK**, the selected user(s) is removed that group's membership.

WiFi Configuration

WiFi Configuration appears among platform views, along with a **WiFi Certificate Manager** view, only for a JACE host (AX-3.6 or later) that has an installed 802.11b/g wireless WiFi adapter. In NiagaraAX, this applies only to a JACE-700 with a Mini-PCI WiFi adapter card (T7-WIFI option).

NOTE: Although JACE-8000 controllers have WiFi capability, it is not supported when running AX.

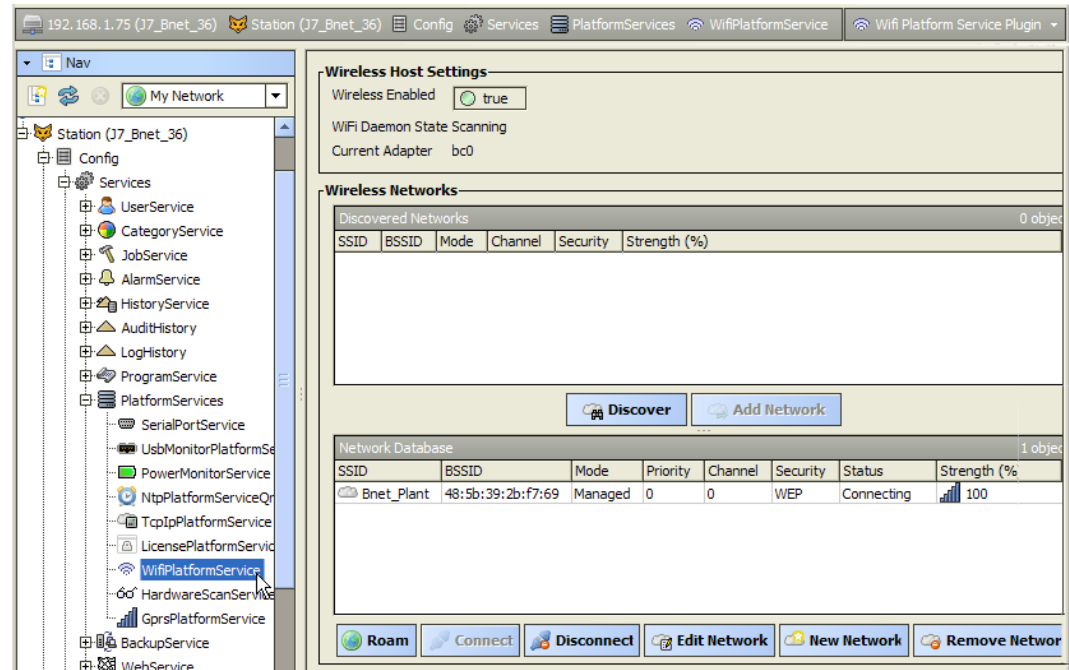
Figure 115. WiFi Configuration platform view



This view lets you discover 802.11b/g networks available to the JACE, and add one or more networks, as necessary.

NOTE: The running station on the JACE also has a "WifiPlatformService" among its platform services. Its default "Wifi Platform Service Pluginview is identical to the **WiFi Configuration** view.

Figure 116. Wifi Platform Service Plugin view



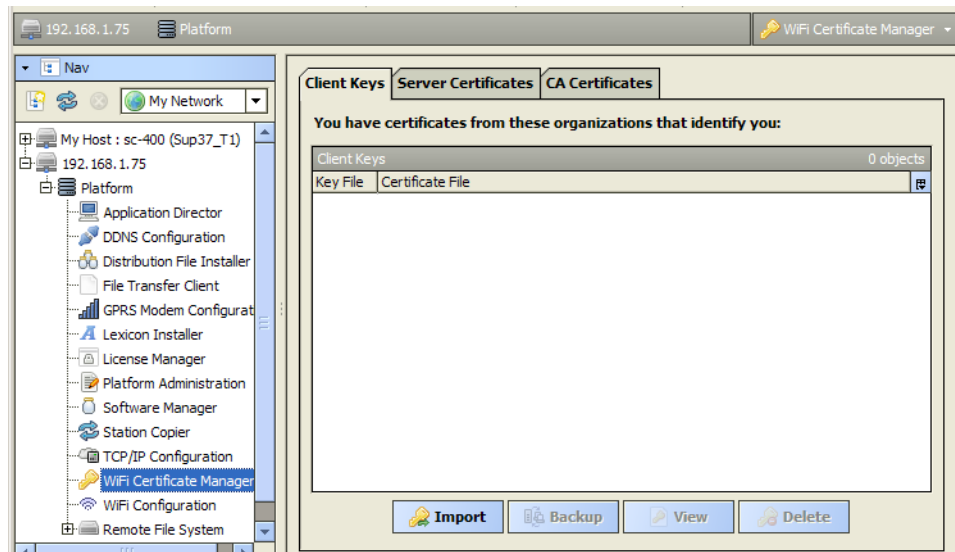
Refer to the *Engineering Notes II* document "NiagaraAX JACE WiFi option" for complete details, including usage scenario, configuring the JACE for WiFi, and further details on this view.

WiFi Certificate Manager

WiFi Certificate Manager appears among platform views, along with a **WiFi Configuration** view, only if a JACE host (AX-3.6 or later) that has an installed 802.11b/g wireless WiFi adapter. This applies only to a JACE-700 with a Mini-PCI WiFi adapter card (T7-WIFI option).

NOTE: Although JACE-8000 controllers have WiFi capability, it is not supported when running AX.

In AX-3.7, this view changed in appearance. It now resembles the (unrelated) platform Certificate Management view, used in SSL support in AX-3.7 and later. However, it uses a different key store, and applies only to WiFi security.

Figure 117. WiFi Certificate Manger platform view (AX-3.7 or later)

This view lets you import "CA certificate", "server certificate", and client "private key" files onto the JACE for use in WiFi security types WPA or WPA2. Usage of these security types (with such digital certificates) are uncommon except in an "enterprise level" network scenario.

NOTE: The running station on the JACE also has a "WifiPlatformService" among its platform services. However, an AX-3.7 or later station has no equivalent WiFi Certificate Manager view (under its PlatformServices). If WiFi certificate configuration is required, you must use this view in a platform connection.

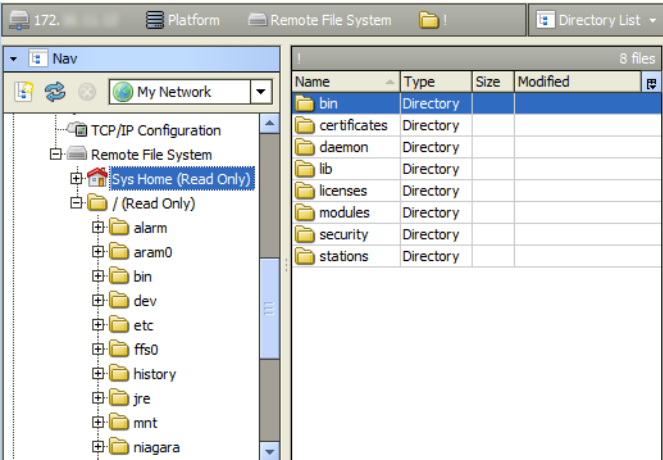
Refer to the *Engineering Notes II* document "NiagaraAX JACE WiFi option" for complete details, including further details on using this view.

Remote File System

The **Remote File System** view is one of several platform views. It provides a read-only view of the remote platform's file system. As needed, you can expand folders and examine and/or copy files to your local computer.

NOTE: To edit or write files on the remote Niagara platform, you must use the platform **File Transfer Client** view. See [“File Transfer Client”](#).

Figure 118. Remote File System for JACE controller platform



CHAPTER 2 PLATFORM SERVICES

TOPICS COVERED IN THIS CHAPTER

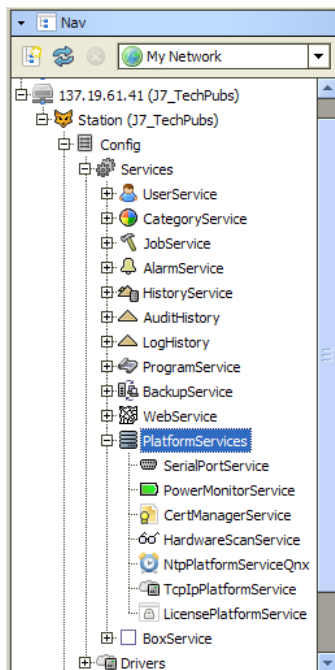
About Platform Services
PlatformServiceContainer parameters
SystemService (under PlatformServices)
Platform service types
Using platform services in a station
About the NtpPlatformService

This section explains the platform access available in a running station—in other words, the *station's perspective* on its host platform. Unlike the various platform views, a platform connection is not needed to access platform services. Instead, you need only a standard station (Fox) connection.

About 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.

Figure 119. Example JACE station's PlatformServices



Platform services in a running station provide two main types of functionality:

- A subset of platform views available in a platform connection. Platform services do *not* provide the full set of functions available in Workbench platform connection. For example, you cannot install or upgrade software, or transfer stations and files. However, a number of platform configuration views are available under a station's PlatformServices.
- Certain platform configuration settings accessible *only* through PlatformServices—that is, not available in a client platform connection.

NOTE: When engineering 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”. For details about station security, see “About Security” in the *NiagaraAX User Guide*.

PlatformServices and all child components are *unique from all other station components*. For details, see “Component differences for platform services”.

Component differences for platform services

PlatformServices is different from all other components in a station in the following ways:

- It acts as the station interface to specifics about the host platform (whether JACE or a PC).
- It is built dynamically at station runtime—you do not see PlatformServices in an offline station.
- Changes you make to PlatformServices and all child services are *not stored in the station database*. Instead, changes are stored in other files on that platform, such as its platform.bog file, or within the platform’s operating system.

NOTE: Do not attempt to edit platform.bog directly; always use PlatformServices’ views.

In summary, when you make changes under a station’s PlatformServices, those changes are independent of the running station. If you install another station, platform services are dynamically recreated again when the new station starts, based upon the last settings.

In addition, understand that some changes in platform services views may require the host to be rebooted to become effective. Examples include TCP/IP changes, or some NTP-related changes in a controller. A “Reboot Now?” popup dialog appears upon saving such a change.

PlatformServiceContainer parameters

In addition to being a container, the default **Platform Service Container Plugin** view provides various status and configuration entries for the host platform. In the Nav tree, double-click **PlatformServices** to access this view, as shown below.

Figure 120. PlatformServicesContainerPlugin view (many entries not shown)

Platform Service Container Plugin	
Name	J600E_T1
Host	192.168.1.123 (J600E_T1)
Model	NPM6E
Host ID	Qnx-NPM6E-0000-153C-7BE2
Niagara Version	3.8.31
Java VM Name	Java HotSpot(TM) Client VM
Java VM Vendor	Sun Microsystems Inc.
Java VM Version	1.5.0_34-b29
OS Name	QNX
OS Arch	ppc
OS Version	6.4.1
Platform Daemon Port	3011
Platform Daemon SSL Port	5011
Locale	en
System Time	18:40
Date	11-Dec-2013

Included are many read-only status values as well as configuration parameters. Each is described in separate sections as follows:

- [PlatformServiceContainer status values](#)
- [PlatformServiceContainer configuration parameters, page 124](#)

By default, any **PlatformServiceContainer** also provides three right-click actions. See [“PlatformServiceContainer actions”](#).

PlatformServiceContainer status values

Status values in a station’s **PlatformServices** container include the following:

- Name
Name of running station.
- Host
IP address of host platform.
- Model
Model of host platform type, such as "NMP6," "JVLN," or "Workstation." See “Models of platforms” for further details.
- 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) Client VM” or "J9" (QNX-based hosts) for any N4 controller, or “Java HotSpot(TM) Client VM” for a Windows-based host.
- Java VM Vendor
Vendor for Java VM: "Sun Microsystems Inc." (Java HotSpot) or "IBM Corporation" (J9).
- Java VM Version
Version of Java VM, for example, Version of Java VM, e.g. "1.5.0_34-b26" (Java Hotspot) or "2.3" (J9)
- OS Name
Operating System name, such as "QNX" or "Windows XP."
- OS Arch.
Machine architecture for OS, such as "ppc" (QNX-based hosts) or "x86" (Windows-based hosts)
Machine architecture for OS, such as “arm” or “ppc” (controller hosts) or “amd64” (Windows hosts).
- OS Version
Operating System version, such as "6.4.1" (QNX) or "5.1" (Windows XP)
Operating System version, such as “6.5.0” (QNX) or “6.1” (Windows 7).
- 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 (see *“Change HTTP 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 `Unknown`. This can prove useful in case you changed the platform TLS port (see “*Change SSL Settings*”), but then forgot what the new port is.

NOTE: In the container plugin, most of the remaining entries are *configuration* parameters. However a few status values are also mixed in, and are described below.

- 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.

- 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 a JACE-8000 host, it may look similar to:

	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

Current total and free RAM statistics for the host. For a JACE-8000, it may look similar to:

Total	Free
1,048,576 KB	113,424 KB

- Serial Number

(Appears only if a JACE host). The controller’s unique serial number.

- Hardware Revision

(Appears only if a JACE host). Hardware revision of the controller.

- Hardware Jumper Preset

(Applies only if a JACE host, except for a 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. See “System shell” in the *JACE NiagaraAX Install & Startup Guide*.

Also see the section “[PlatformServiceContainer configuration parameters](#)”, page 124.

PlatformServiceContainer configuration parameters

Configuration properties of a station’s **PlatformServices** Container are listed below. If needed, you can change any in the container plugin view (property sheet)—click **Save** to write to the host platform.

NOTE: It is recommended that you leave engine-related parameters and other advanced settings at *default* values, unless you have been directed otherwise by Systems Engineering.

- 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 JACE 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.

- Enable 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 Backups 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.

- Battery Present

(Applies only if a JACE host other than a JACE-8000) Applies to configuration of a JACE's backup battery. Used to specify whether the controller has an integral backup battery, typically an onboard NiMH battery. The default property value is true—which 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).

If set to false and saved, upon the next reboot the station's PowerMonitorService no longer monitors for a backup battery, with the underlying "power daemon" stopped. This prevents nuisance "battery bad" alarms. Station backup is dependent totally on SRAM and the station's DataRecoveryService (the JACE must have the platDataRecovery 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 (e.g. JACE-6E or JACE-3E, or else a JACE-6 or JACE-7 with an SRAM option card). The two other options are to use both backup battery and SRAM for backup, or to use backup battery only (and not SRAM). These other two options require that this Battery Present property is set to true.

For related details, refer to the document *Data Recovery Service Guide*.

- Failure Reboot Limit

(JACE platforms only) Limits the number of station restarts that can be triggered by station failures, within the Failure Reboot Limit Period, below (if the host is so configured using the **Application Director**, see "Start checkboxes"). Default value is 3.

- Failure Reboot Limit Period

(JACE platforms only) Specifies the repeating frequency of the Failure Reboot Limit period, with a default value at 10 minutes.

These two "Failure Reboot" settings are also adjustable (in any version of QNX-based host) within that JACE's !daemon/daemon.properties file, in the following two properties:

- failureRebootLimit=x (where x is integer, default is 3)
- failureRebootLimitPeriod=y (where y is long in milliseconds, default is 3600000)

- RAM Disk Size

Has one configurable field and one read-only field:

- Min Free — minimum allowable free size in %. If status is not Ok, a "Low RAM disk space" warning is overlaid in all Workbench views of the station.
- Size — Read-only in MB, where default is 32 for a JACE-3E or JACE-6 or JACE-6E series, or 48 for a JACE-7 series, or 394 for a JACE-8000 series. Specifies the size of RAM disk used to store history and alarm files.

- Java Heap

Has one configurable "Min Free" field, in MB. Specifies the *minimum* free Java heap size, in MB, against which the station compares (tests) for low memory conditions, that is excessive Java heap. The default varies according to JACE model. 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" appears 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

Has one configurable “Min Free” field, related to number of files (and/or open sockets). 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

Has one configurable “Min Free” field, in KB. Specifies the minimum RAM that can be left free during station operation. If status is not Ok, a “Low free RAM” warning is overlaid in all Workbench views of the station.

- Disk Space

Has one configurable “Min Free” field, in %. 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 is overlaid in all Workbench views of the station.

- Files

Has one configurable “Min Free” field, to specify the minimum number of free files available during station operation. Below this amount, a related platform warning appears. Note that the PlatformServiceContainer status property “Filesystem” includes both the current number of files and the maximum number of files for each partition on a JACE controller.

Also see the section “[Model-specific PlatformServiceContainer properties](#)”.

Model-specific PlatformServiceContainer properties

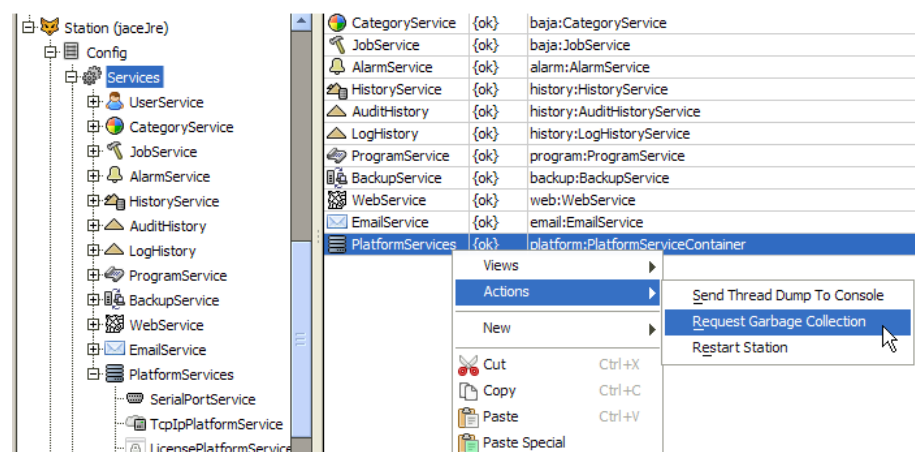
Some JACE controller models may have yet more **PlatformServices** properties, specific to special hardware features. This is in addition to the standard and additional properties described above. Typically, these are configured at JACE commissioning time.

For more details, see “Controller-specific PlatformServices properties” in the *JACE NiagaraAX Install & Startup Guide*.

PlatformServiceContainer actions

The **PlatformServices** Container also provides three (right-click) actions, as shown below.

Figure 121. PlatformServicesContainer actions.



These actions are described as follows:

- **Send Thread Dump to Console**

Causes that host's platform daemon to have the station send a VM thread dump to its standard output (console), equivalent to the "Dump Threads" command in the platform **Application Director** view. Typically used only during troubleshooting.

NOTE: Apart from Application Director (platform access) to view station output, you can also view a "snapshot" of station output in a browser. Do this via the "stdout" link in the **spy** utility, at URL `http://<hostIP>/ord?spy:/stdout`

- **Request Garbage Collection**

Causes the JVM running the station to perform garbage collection. This results in a "best effort" towards releasing unused objects and making more memory available on the "heap". Note that current heap and memory statistics for any running station are available on the **ResourceManager** view of the station component.

- **Restart Station**

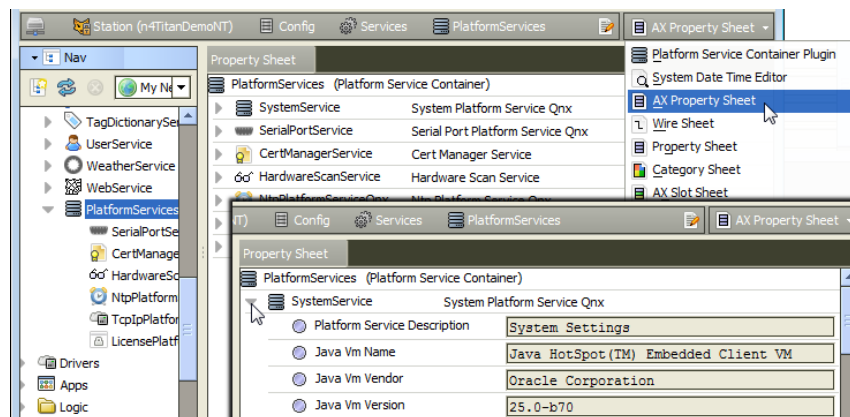
Produces a popup confirmation dialog. Applies directly to a station running in a Windows-based platform, where it is equivalent to issuing a "Restart" command from the **Application Director** (station is saved on its host, then restarted). If issued to a station running on a QNX-based platform, this results in a host reboot (station restart not available unless host is rebooted).

NOTE: Also, most *child* services under the **PlatformServices** Container have an available "Poll" action, which refreshes their property values. See "[Platform service types](#)", page 129 for a listing of possible child services.

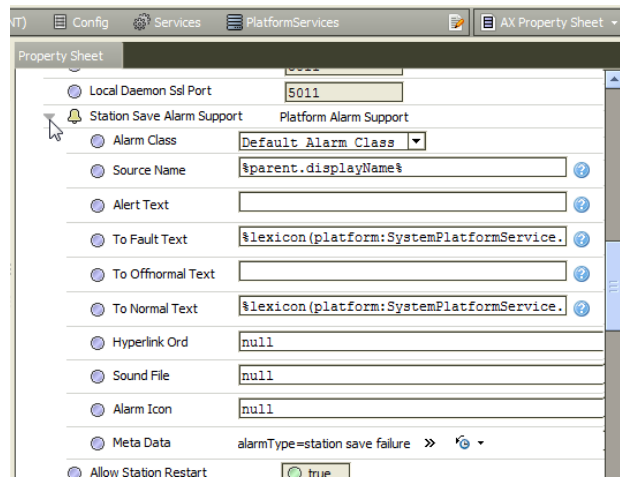
SystemService (under PlatformServices)

PlatformServices also contains a child "**SystemService**" container, accessible from its property sheet as shown below. Unlike other child services, **SystemService** does not appear in the Nav tree.

Figure 122. SystemService from property sheet of PlatformServices.



When you expand SystemService, you see most of the same properties available in the default Platform Service Container Plugin view (see "[PlatformServiceContainer parameters](#)"). In addition, as shown below, there is a container slot "Station Save Alarm Support".

Figure 123. 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”. For related details, see [“PlatformServices binding and link caveats”](#), page 131.

Platform service types

In addition to the **SystemService** found under its property sheet, the **PlatformServices** Container has various child services, of which different types are listed below.

NOTE: Some platform services are intended to support installations where *all* configuration must be done using only a browser connection (and not an Workbench platform connection to an JACE’s platform daemon). Examples include types **TcpIpService** and **LicenseService**.

The list of visible platform service types includes the following:

- CertManagerService

For management of PKI certificate stores and/or allowed host exceptions, used in certificate-based TLS connections between the station/platform and other hosts. For details, see the *Station Security Guide*.

- TcpIpPlatformService

Provides access to the same configuration using the platform’s **TCP/IP Configuration** view. See *“TCP/IP Configuration”*.

- LicensePlatformService

Provides access to the same configuration using the platform’s **License Manager** view. See *“License Manager”*.

- **SerialPortService**
(JACE platforms only) Allows review of available serial ports on the host platform.
- **PowerMonitorService**
(All platforms except for JACE-8000 series) Provides configuration and status of the controller's battery monitoring and AC power-fail shutdown routines. See [Power monitoring, page 130](#) for details.
- **NtpPlatformService**
Provides the Niagara 4 interface to the NTP (Network Time Protocol) service or daemon of the platform's OS (QNX or Windows), including several configuration parameters and a list specifying one or more NTP time servers. For details, see [“About the NtpPlatformService”, page 132](#).
- **DataRecoveryService**
(JACE platforms only) Allows monitoring 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). For details, refer to the document *Data Recovery Service Guide*.
- **HardwareScanService**
(JACE platforms only) Optional platform service that provides a graphical diagram of communication ports and other features on the hosting platform, including callouts to a table that explain the location, description (such as COM2), port type, and status/usage of each item. Requires installation of the modules `platHwScan` and a corresponding `platHwScanType`. Refer to the *Hardware Scan Service Engineering Notes* document.

Using platform services in a station

Apart from configuration usage, some platform services under the [PlatformServices](#) Container provide status values that you can further incorporate. Typically, each value also provides built-in alarm features. Usage is typical for the following:

- [Power monitoring, page 130](#)

Power monitoring

By default, through the **PowerMonitorService**, any JACE provides status monitoring of the following items, via “Boolean” type slots:

- **AC power**
 (“Primary Power Present” slot) — True whenever AC power is currently supplied to the JACE.
- **Battery level**
 (“Battery Good” slot) — True if last JACE test of NiMH backup-battery was good.
 Also included is a “Time of Last Test” slot that provides a timestamp for the last battery test.

If needed, you can make Px bindings or links to these slots (however, see [“PlatformServices binding and link caveats”, page 131](#)).

In addition to these read-only status slots, the **PowerMonitorService** provides related *configuration* slots, which you typically review at commissioning time. For more details and a related procedure, see “JACE power monitoring configuration” in the *JACE NiagaraAX Install & Start-up Guide*.

Battery monitoring disabled

(Does not apply to a JACE-8000 series controller) An SRAM-equipped JACE can be configured for “battery-less” operation (the `platDataRecovery` module must be installed, and JACE licensed with for the “dataRecovery” feature). The **PowerMonitorService** will continue 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—unless you set the “Battery Present” property (of its PlatformServiceContainer) from true (the default) to false. This disables backup battery support and prevents ongoing “battery bad” nuisance alarms—when there is no backup battery. For related details see [“PlatformServiceContainer configuration parameters”, page 124](#).

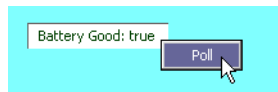
PlatformServices binding and link caveats

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/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 of the container for those properties.

For example, if on a Px page you bind a BoundLabel to the PowerMonitorService’s “Battery Good” slot, it will display text as “true” or “false.” However, this value does not update until the user right-clicks for the “Poll” action, which forces a fresh read.

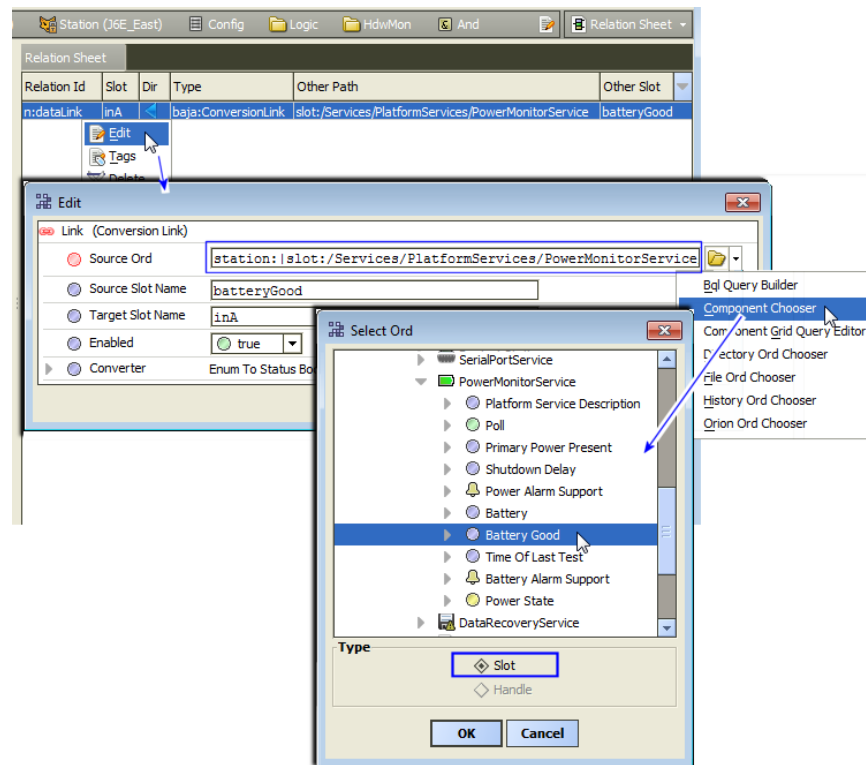
Figure 124. Poll action on bound PlatformServices property



- Links from PlatformServices (and child slots) to other station components must use a source or *“slot path”*, versus “handle”. Otherwise, after a station restart or host reboot, handle-sourced links may be lost. An example link being edited to use slot path is shown in the figure above.

NOTE: Consider the “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.”

Figure 125. From RelationSheet of target component, editing link to use slot path for source ord.



However, note that the station’s plugins (views) for the **PlatformServices** *do* provide updated property values, as they work in concert with the special polling used for platform-resident data.

About the NtpPlatformService

PlatformServices in any station contains a child **NtpPlatformServicesOS**, 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.

NOTE: 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. See the section *“Interaction with station’s TimeSyncService”* for related details.

See the following sections for more details:

- [About the Ntp Platform Service Editor](#)
- [NTP port/firewall considerations, page 136](#)

About the Ntp Platform Service Editor

For either platform OS type (Windows or QNX), the default view for any **NtpPlatformService** is an **Ntp PlatformService Editor** OS view, your typical interface. Double-click any **NtpPlatformService** to see this editor.

The **NtpPlatformService Editor** on any Windows platform is disabled and read-only.

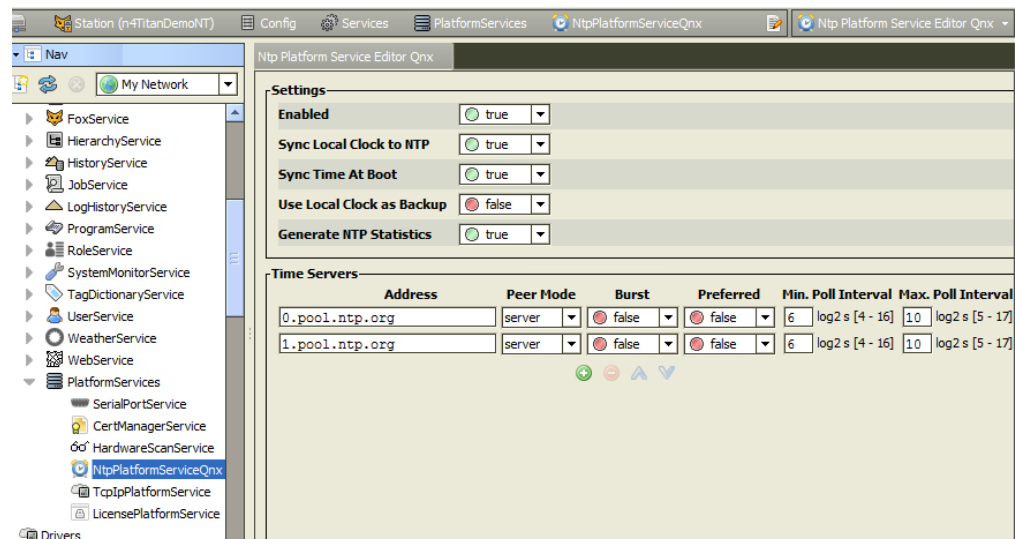
Although a few differences exist between the different NTP platform service editors, all three types use a similar division of "key properties" at top, and "specified time server(s)" at the bottom of the dialog. Specific details on each NTP platform service editor are in the following sections:

- [Ntp Platform Service Editor QNX, page 133](#)
- [Ntp Platform Service Editor Win32, page 135](#)

About the Ntp Platform Service Editor Qnx

An example **Ntp Platform Service Editor** for a JACE controller is shown below. This is the default view for the **NtpPlatformServiceQnx**.

Figure 126. Ntp Platform Service Editor Qnx



This dialog provides access to some of the key settings of the NTP daemon (ntpd) of the QNX OS running on the host JACE platform.

There are two main areas: **Settings** at top, **Time Servers** at bottom. The **NtpPlatformServiceQnx** also has an available “Sync Now” action. For more details, see [“Sync Now action”, page 135](#).

Ntp Platform Service Editor Qnx settings

Settings in the **Ntp Platform Service Editor Qnx** include the following properties:





- **Enabled**
If true, the host will use NTP to sync its clock with time values retrieved from other servers.
- **Sync Local Clock to NTP**
If true, this enables the host to adjust its local clock by means of NTP. If disabled (false), the local clock free-runs at its intrinsic time and frequency offset. This flag is useful in case the local clock is controlled by some other device or protocol and NTP is used only to provide synchronization (as server) to other clients. In this case, the local clock driver can be used to provide this function and also certain time variables for error estimates and leap-indicators.
- **Sync Time At Boot**
Default is false. If true, when the JACE boots, before the stations starts or the ntpd starts, it executes the `ntpdate` command. This updates the system local time.
- **Use Local Clock as Backup**
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 `ntpd` (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**
If true, the `NtpPlatformService` reports whatever information it can about its operation. To access these statistics with the station opened in Workbench, right-click the `NtpPlatformServiceQnx` and select **Views > SpyRemote**. Keep in mind that the `ntpd` 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 in the **Ntp Platform Service Editor Qnx** specifies a server to which the host's clock will be sync'ed when the service is Enabled (true), and "Sync Local Clock to NTP" is also true. These servers are *not* used if either of these properties are false.

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). Fields for each time server includes the following:

- **Address**
Fully qualified domain name, IP address, or host files alias for the NTP time server.
- **Peer Mode**
Peer mode to use with the server, as either server or peer (symmetricActive).
- **Burst**
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 2 seconds.
- **Preferred**

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**

Minimum poll interval for NTP messages, from 4 to 16. Note 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).

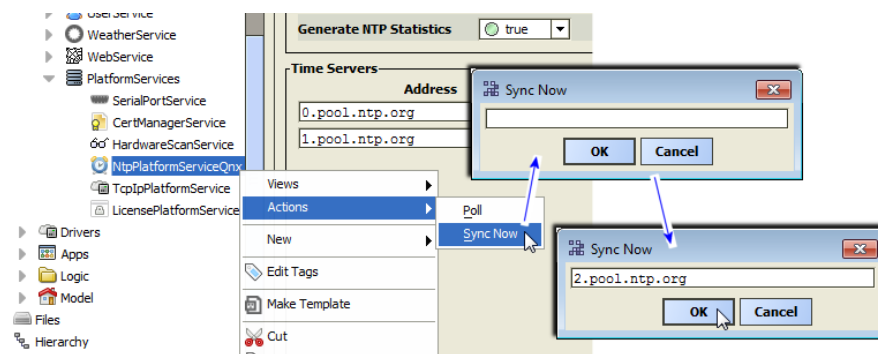
- **Max. Poll**

Maximum poll interval for NTP messages, from 10 to 17. Note 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).

Sync Now action

In addition to the “Poll” action present on any `NtpPlatformService`, the `NtpPlatformServiceQnx` component has an additional “Sync Now” action.

Figure 127. Sync Now action on NtpPlatformServiceQnx



As shown here, this action produces a popup **Sync Now** dialog, which is blank.

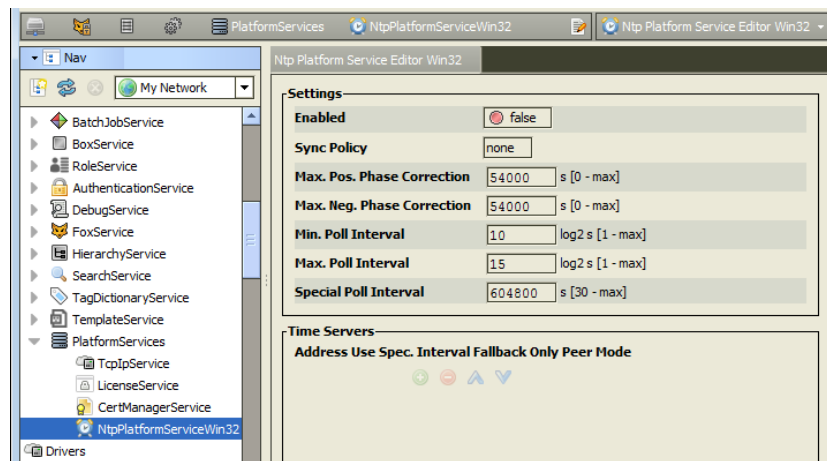
To use, type in the fully qualified domain name of a public NTP server (as shown above), or else the IP address of any accessible NTP server, and then click **OK**.

To verify, look for a related entry in the station’s spy “platform diagnostics” log. Do this in Workbench by right-clicking the station, then selecting **Spy > platform diagnostics > log** or from the **File** menu, **File > Open ord** (Ctrl + L) and enter:

```
ip:EC-BOS_IP_address|fox:|spy:/platform diagnostics/log
```

About the Ntp Platform Service Editor Win32

An example **Ntp Platform Service Editor Win 32** is shown below. This is the default view for the `NtpPlatformService` on a Windows-based (Win32 or Win64) host.

Figure 128. Ntp Platform Service Editor Win32

This dialog provides access to some of the key settings of the Windows Time service (W32Time) on the host platform.

NOTE: Settings are only a small subset of those possible to configure—for more fine-grained tuning of the time service, Windows registry settings can be set according to Microsoft’s latest instructions. Visit the Microsoft tech support site for more information on a particular Windows OS, for example using this search: <http://search.microsoft.com/en-us/SupportResults.aspx?q=ntp+time+service>

As in all Ntp Platform Service Editors, there are two main areas: **Settings** at top, **Time Servers** at bottom.

NTP port/firewall considerations

On any host, NTP requires the use of UDP port 123—this port is not configurable. On a JACE platform this is not an issue.

However, on a Windows host platform, in addition to configuring NTP using Windows native tools, typically you need to make the necessary firewall exception or “iptables” entry to allow UDP port 123 traffic. Otherwise, NTP time synchronization can fail because of firewall-blocked messages.

CHAPTER 3 PLATFORM COMPONENT GUIDES

TOPICS COVERED IN THIS CHAPTER

Components in platCrypto
Components in platDataRecovery module
Components in platGprs module
Components in platform module
Components in platHwScan
Components in platIEEE8021X
Components in platPower module
Components in platPowerNxs module
Components in platSerialQnx module
Components in platSerialWin32 module
Components in platSerialWin64 module
Components in platSysmonNx module
Components in platSysmonNxs module
Components in platSysmonNxt module
Components in platUsbmon module
Components in platWifi module

Component Guides provides summary information on on the following platform-related components.

- [platCrypto](#), page 137
- [platDataRecovery](#), page 138
- [platGprs](#), page 138
- [platform](#), page 139
- [platHwScan](#), page 143
- [platIEEE8021X](#), page 144 (AX-3.8)
- [platPower](#), page 144
- [platPowerNxs](#), page 146
- [platSerialQnx](#), page 146
- [platSerialWin32](#), page 147
- [platSerialWin64](#), page 147
- [platSysmonNx](#), page 147
- [platSysmonNxs](#), page 148
- [platSysmonNxt](#), page 148
- [platUsbmon](#), page 148
- [platWifi](#), page 148

Components in platCrypto

- CertManagerService
- DaemonSecureSession


platCrypto-CertManagerService

The component is a platCrypto platform service of any AX-3.7 or later station. It has few visible properties, but provides a default **Certificate Management** view, that is equivalent to that same-named platform view.

The **Certificate Management** view provides the means to import and export signed certificates (for TLS secure connections) into the platform's key and trust stores, and to perform other related functions. For complete details, refer to the document *Station Security Guide*.

platform-DaemonSecureSession

This platCrypto component represents a *secure* platform connection to a host made in Workbench. In the Nav tree view.


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).

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.

Components in platDataRecovery module

- DataRecoveryService

platDataRecovery-DataRecoveryService

This component () in the **platDataRecovery** module automatically creates and manages buffers in a controller's available SRAM (Static Random Access Memory), allowing a controller to function without a battery.

The controllers with integral SRAM include: JACE-8000, JACE-3E, JACE-6E, JACE-603, JACE-645) as well as a JACE-6 and JACE-7 with an installed SRAM option card.

Some SRAM-equipped controllers support a backup battery with the addition of an optional NiMH onboard battery pack, or an external 12V sealed lead-acid battery. For these controllers, both the **DataRecoveryService** and **PowerMonitorService** run in the station's **PlatformServices** container, operating independently or in unison, as configured.

For details, see the *Data Recovery Service Guide*.

Components in platGprs module

- [GprsPlatformService, page 138](#)
- [GprsHostSettings, page 139](#)
- [GprsRuntimeData, page 139](#)

platGprs-GprsPlatformService

Gprs Platform Service is the station's interface to the platform's GPRS daemon (gprsd). If the host QNX-based JACE has a GPRS modem option installed, along with the platGprs module, this service is found under the running station's PlatformServiceContainer.

NOTE: The GprsPlatformService has a default Gprs Platform Service Plugin view—identical to the platform view GPRS Modem Configuration. This provides station access to modem configuration properties, and also runtime data. For details, see [GPRS Modem Configuration, page 46](#). This GprsPlatformService also has many properties available, located under a child GprsHostSettings container with its own GprsRuntimeData child container. For complete GPRS modem option details, refer to the Engineering Notes document *GPRS modem option*.

platGprs-GprsHostSettings

GprsHostSettings (Settings) is a container child of the GprsPlatformService in a QNX-based controller that is equipped with a GPRS modem option. Access it under the service's property sheet view. It contains a number of read-only properties, most of which reflect configuration, as well as a GprsRuntimeData child container.

See the "GprsPlatformService" section in the Engineering Notes document *GPRS modem option*.

platGprs-GprsRuntimeData

GprsRuntimeData (Runtime Data) is a container child of the GprsHostSettings container under the GprsPlatformService in a QNX-based controller that is equipped with a GPRS modem option. Access it under service's property sheet view. It contains a number of read-only properties.

See the "GprsPlatformService" section in the Engineering Notes document *GPRS modem option*.

Components in platform module

platform-DefaultDaemonFileSpace


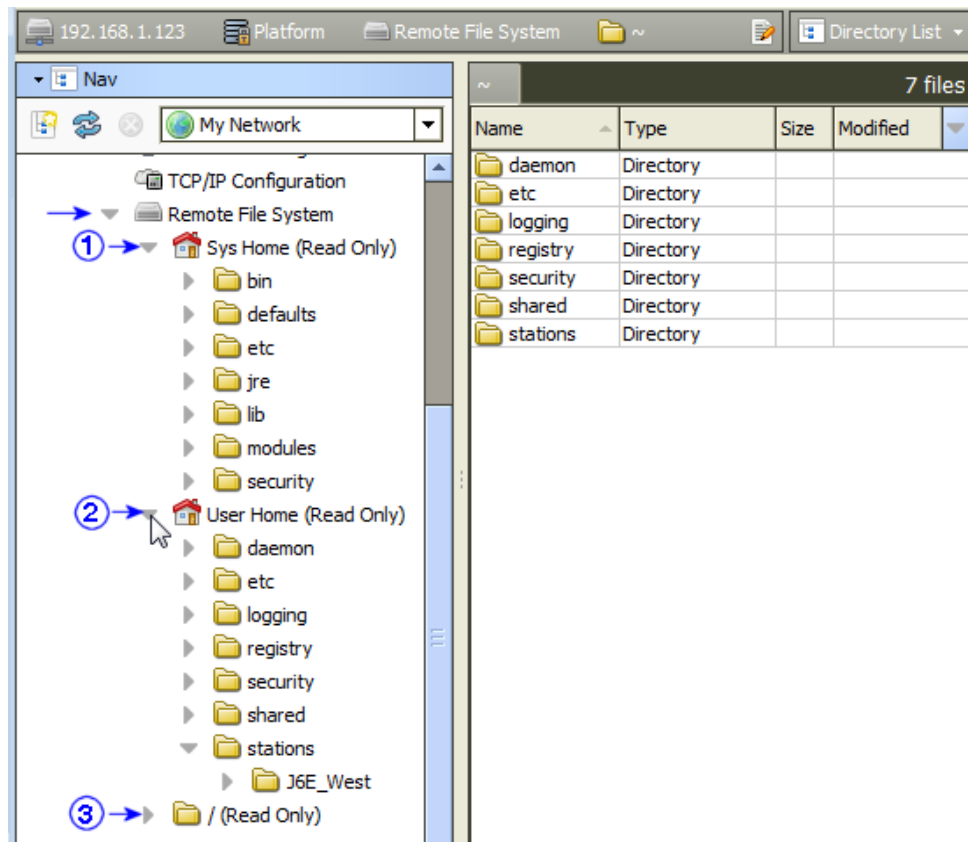
This component () is in the **program** module. The **Remote File System** view is one of several platform views. It provides a *read-only* view of the remote platform's file system.

Figure 129. Remote File System for JACE controller platform

The **Remote File System** (DefaultDaemonFileSpace) represents the files accessible for read-only access when platform-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.
- (JACE controllers only) The root folder for the entire file system, with browse capability.

To edit or write files on the remote Niagara platform, you must use the platform **File Transfer Client** view.

platform-DaemonSession

This component represents a platform connection to a host made in Workbench. To access this component, expand **My Host > Platform** and double-click **Platform Administration**.

Figure 130. Platform Administration

The screenshot shows the Niagara Workbench interface for Platform Administration. The left navigation pane lists various system components, with 'Platform' selected. The main pane displays the 'Platform Administration' settings, including system information, configuration options, and a file system table.

Platform Administration Settings:

- View Details** (icon)
- User Accounts** (icon)
- System Password** (icon)
- Change HTTP Port** (icon)
- Change SSL Settings** (icon)
- Change Date/Time** (icon)
- Advanced Options** (icon)
- Change Output Settings** (icon)
- View Daemon Output** (icon)
- Configure Runtime Profiles** (icon)
- Backup** (icon)
- Commissioning** (icon)
- Reboot** (icon)

System Information:

- Baja Version:** Tridium 4.0.12.3.515
- Daemon Version:** 4.0.12.3.515
- System Home:** /opt/niagara
- User Home:** /home/niagara
- Host:** 192.168.1.222 (n4TitanDemo)
- Daemon HTTP Port:** 3011
- Daemon HTTPS Port:** 5011
- Host ID:** Qnx-TITAN-7E58-A1C7-CC4B-EBF0
- Model:** TITAN
- Local Date:** 11-May-15
- Local Time:** 21:18 Coordinated Universal Time
- Local Time Zone:** UTC (+0)
- Operating System:** qnx-jace-n4-titan-am335x (4.0.26.1)
- Niagara Runtime:** nre-core-qnx-armle-v7 (4.0.12.3.515)
- Architecture:** armle-v7
- Enabled Runtime Profiles:** rt,ux,wb
- Java Virtual Machine:** oracle-jre-compact3-qnx-arm (Oracle Corporation 1.8.0.0.8)
- Niagara Stations Enabled:** enabled
- Number of CPUs:** 1
- Current CPU Usage:** 4%
- Overall CPU Usage:** 3%

Filesystem Table:

	Total	Free	Files	Max Files
/	3,492,848 KB	3,246,316 KB	857	109152
/mnt/aram0	393,215 KB	376,316 KB	0	0
/mnt/ram0	8,192 KB	8,131 KB	0	0

Physical RAM:


Total	Free
1,048,576 KB	117,748 KB

Other Parts: n4-titan-am335x (4.0.1)


In the Nav tree view, the DaemonSession 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 views.


platform-LicenseDatabaseTool

 The LicenseDatabaseTool (Local License Database) represents your Workbench PC's "local license database." The default view is the **Workbench License Manager**, which allows you to manage locally-stored licenses.


platform-LicensePlatformService

 The LicensePlatformService provides station access to the host 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.


platform-NtpPlatformServiceQnx

 The NtpPlatformServiceQNX 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 **Ntp Platform Service Editor Qnx** plugin, in which you can adjust a *few* settings, as well as specify time servers.

platform-NtpPlatformServiceWin32

 The NtpPlatformServiceWin32 is the Niagara interface to the Windows Time service (W32Time) on a Win32-based platform's Windows OS. This Windows service uses the SNTP (Simple Network Time Protocol) to synchronize to one or more designated time servers. The default view of this platform service is the **Ntp Platform Service Editor Win32** plugin, in which you can adjust a few settings of the Windows Time service, including identifying NTP time servers. For more details, see [About the NtpPlatformService, page 132](#).

platform-PlatformAlarmSupport


 PlatformAlarmSupport is a container slot that appears for each alarmable value under a Platform Service, such as the **PowerMonitorService** for many JACE controllers.

For a JACE platform, example PlatformAlarmSupport components include:

- Battery Alarm Support
To configure how "low battery level" alarms are handled in the station.
- Power Alarm Support
To configure how "AC power loss" alarms are handled in the station.

Properties under each PlatformAlarmSupport 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.

platform-PlatformServiceContainer

 PlatformServiceContainer (**PlatformServices**) provides a container for a station's Platform-Service instances. The **Platform Service Container Plugin** is its primary view. The Platform-ServiceContainer is available when online with any running station, under its Config, **Services** folder.

platform-SystemPlatformServiceQnxJavelina

SystemPlatformServiceQnxJavelina (**SystemService**) is the QNX implementation of SystemPlatformService in a station running on a JVLN-based (JACE-700) controller.

platform-SystemPlatformServiceQnxNpm6xx

SystemPlatformServiceQnx (**SystemService**) is the QNX implementation of SystemPlatformService in a station running on a JACE controller.

platform-SystemPlatformServiceWin32

SystemPlatformServiceWin32 (**SystemService**) is the Win32 implementation of SystemPlatformService.

platform-TcpIpPlatformService

TcpIpPlatformService 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. For more details see [TCP/IP Configuration, page 103](#). 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.

Components in platHwScan

platHwScan-HardwareScanService

The **Hardware Scan Service** is an available platform service on a JACE station, providing that the JACE platform has the platHwScan module installed.

To function correctly, the appropriate platHwScanType module also needs to be installed on the JACE. Otherwise, the default **Hardware Scan Service View** will simply display:

Jar file platHwScanType is required to support this platform

Where the appropriate platHwScanType is as follows:

Controller Series	platHwScanType module
JACE-6E, JACE-6, JACE-3E	platHwScanNpm
JACE-7 (700)	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

This default **Hardware Scan Service View** provides a diagram of the controller that shows its communication ports and other features (including, if applicable, installed communication

options such as modules or cards). The diagram has callouts to a table that explains each item's location, description (such as COM2), port type, usage, and status.

Components in platIEEE8021X

NOTE: IEEE 802.1X support is available only for QNX-based "Hotspot" controllers (JACE-3E, JACE-6,JACE-7 series) using AX-3.8.

- [platIEEE8021X-IEEE8021XAdapterSettings, page 144](#)
- [platIEEE8021X-IEEE8021XHostSettings, page 144](#)
- [platIEEE8021X-IEEE8021XPlatformService, page 144](#)

platIEEE8021X-IEEE8021XAdapterSettings

IEEE8021XHostAdapters (en0 or en1) is a container under a station's IEEE8021XPlatformService, accessible through the service's property sheet, under **Settings, Adapter Settings**. Each container holds various properties for a specific Ethernet (port) adapter, and has an available view, **IEEE 8021X Adapter Settings Editor**. For complete details, refer to the AX-3.8 document *NiagaraAX IEEE 802.1X Configuration — Engineering Note*.

platIEEE8021X-IEEE8021XHostSettings

IEEE8021XHostSettings (Settings) is a container of a station's IEEE8021XPlatformService, accessible through the property sheet of that platform service. It holds various properties for each of the available Ethernet (port) adapters, in separate **IEEE8021XAdapterSettings** sub-containers. For complete details, refer to the AX-3.8 document *NiagaraAX IEEE 802.1X Configuration — Engineering Note*.

platIEEE8021X-IEEE8021XPlatformService

In AX-3.8, the IEEE8021X Platform Service is an available platform service on a JACE station, providing that the host has the platIEEE8021X module installed and is licensed with the "ieee8021x" feature.

This service provides a station interface to configure the host platform (JACE) to be able to join a wired IEEE 802.1X-authenticated network. The service's default view, **IEEE 8021X Platform Service Plugin**, is identical to the platform view **IEEE 802.1X Configuration**, with separate tabs for each of the two Ethernet adapters on the controller (en0 for LAN1, en1 for LAN2).


A property sheet view also provides access to additional properties for the two adapters. For complete details, refer to the AX-3.8 document *NiagaraAX IEEE 802.1X Configuration — Engineering Note*.

Components in platPower module

platPower-ExternalSlaBattery

ExternalSlaBattery is one of two "Battery" slots in the **JavelinaBatteryPlatformService** in a 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.

platPower-JavelinaBatteryPlatformService

 JavelinaBatteryPlatformService (PowerMonitorService) applies to a station running in a 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. In addition, 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. Note the JACE-7 controller has three onboard CIs, with the intended use for UPS AC power lost, UPS low battery, and (door) tamper switch.

NOTE: The tamper switch CI on the JACE-7 controller is enabled/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).

Typically, support is enabled and configured at JACE *commissioning time*. For related details, see “JACE power monitoring configuration” in the latest *JACE NiagaraAX Install & Startup Guide*.

platPower-NimhBattery

- NimhBattery is a “Battery” container slot under the PowerMonitorService in a 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).

platPower-Npm2NimhBattery

- 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). 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 a JACE that is capable and enabled for dual battery support.

platPower-NpmDualBatteryPlatformService

- NpmDualBatteryPlatformService (PowerMonitorService) applies to a station running in a JACE 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 *JACE NiagaraAX Install & Startup Guide*.

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 simply indicates the host JACE platform can use an optional, sealed-lead acid (SLA) battery, *in addition to* the onboard NiMH backup battery.

platPower-PowerMonitorPlatformServiceQnx

PowerMonitorPlatformServiceQnx (**PowerMonitorService**) is used to monitor the primary power status and backup battery level in many JACE controllers. A few configuration parameters allow changing the shutdown delay time, as well as alarm source configuration for both types of alarms (low battery level, primary power lost).

This PowerMonitorService is found under the **PlatformServices** container in a station running on many JACE controllers *except* for those models that are capable and/or enabled for “dual battery” support. Typically, support is enabled and configured at JACE *commissioning time*. For related details, see “JACE power monitoring configuration” in the latest applicable *Install and Startup Guide*.

Components in platPowerNxs module

platPowerNxs-PowerMonitorPlatformServiceNxsWin32

PowerMonitorPlatformServiceNxsWin32 (PowerMonitorService) is used to monitor the status of primary power, UPS communications, and UPS battery condition for a JACE-NXT or JACE-NXS. Configuration parameters allow changing the shutdown delay time, as well as alarm source configuration for all three types of alarms (low battery level, primary power lost, UPS communications).

The PowerMonitorService is found under the PlatformServiceContainer in a station running on any JACE-NXT or JACE-NXS. See Using platform services in a station for related details. For specific details, see the section “Power monitoring configuration in JACE-NXT” or “Power monitoring configuration in JACE-NXS” in the appropriate JACE-NXT (or JACE-NXS) JACE NiagaraAX Install & Startup Guide.

NOTE: This service applies to any CompactFlash-based JACE-NXT or JACE-NXS, which includes the special “SITOP” DC UPS and UPS battery modules. However, if a hard drive-based unit (installed without this UPS option), you can safely ignore this service, and its contained slots.

Components in platSerialQnx module

- [SerialPortPlatformServiceQnx](#), page 146
- [SerialPortQnx](#), page 146

platSerialQnx-SerialPortPlatformServiceQnx

SerialPortPlatformServiceQnx is the station’s interface to the platform’s serial port configuration, such as used by a JACE-3,-6,-7 series host. This service is found under the running station’s **PlatformServices** container as the **SerialPortService**.

platSerialQnx-SerialPortQnx

SerialPortQnx contains properties that describe how a serial port (RS-232 or RS-485) on a JACE controller is being used in software as COMn. Each one is a child of that JACE’s SerialPortService (**SerialPortPlatformServiceQnx**). Properties are as follows:

- Owner — The driver network or function currently associated with that COM port, for example, “NrioNetwork”, “dialup”, “none”, “ModbusAsyncNetwork”, or “dbgjmp” (latter indicated for COM1 when “serial shell” jumper is installed on JACE).
- Os Port Name — How the port is known to the QNX OS and associated low-level drivers.

- Port Index — Unique serial port index number, starting with 1 for COM1.

Components in platSerialWin32 module

- [platSerialWin32-SerialPortPlatformServiceWin32, page 147](#)
- [platSerialWin32-SerialPortWin32, page 147](#)

platSerialWin32-SerialPortPlatformServiceWin32

SerialPortPlatformServiceWin32 is the station's interface to the platform's serial port configuration, used by any 32-bit Windows based host, such as a JACE-NXT or Supervisor PC. This service is found under the running station's PlatformServiceContainer as the SerialPortService.

platSerialWin32-SerialPortWin32

SerialPortWin32 contains properties that describe how a serial port (RS-232 or RS-485) on a Win32-based host is being used in software as COMn. Each one is a child of that host's SerialPortService (SerialPortPlatformServiceWin32). Properties are as follows:

- Owner – The driver network or function currently associated with that COM port, for example, "ModbusSlaveNetwork" or "none".
- Os Port Name – How the port is known to the Windows operating system, e.g. COM1 or COM3.
- Port Index – Unique serial port index number, starting with 0 for COM1.

Components in platSerialWin64 module

- [platSerialWin64-SerialPortPlatformServiceWin64, page 147](#)
- [platSerialWin64-SerialPortWin64, page 147](#)

platSerialWin64-SerialPortPlatformServiceWin64

SerialPortPlatformServiceWin64 is the station's interface to the platform's serial port configuration, used by any 64-bit Windows based host, typically a Supervisor PC. This service is found under the running station's PlatformServiceContainer as the SerialPortPlatformServiceWin64.

platSerialWin64-SerialPortWin64

SerialPortWin64 contains properties that describe how a serial port (RS-232 or RS-485) on a 64-bit Windows host is being used in software as COMn. Each one is a child of that host's SerialPortService (SerialPortPlatformServiceWin64). Properties are as follows:

- Owner — The driver network or function currently associated with that COM port, for example, "ModbusSlaveNetwork" or "none".
- Os Port Name — How the port is known to the Windows operating system, e.g. COM1 or COM3.
- Port Index — Unique serial port index number, starting with 0 for COM1.

Components in platSysmonNx module

- [platSysmonNx-HardwareMonitorNxPlatformServiceWin32, page 148](#)

platSysmonNx-HardwareMonitorNxPlatformServiceWin32

HardwareMonitorNxPlatformServiceWin32 (HardwareMonitorService) is the station's interface to internal environmental parameters in the host JACE-NXS/JACE-NXT, such as CPU temperature, fan speeds, and various voltages. This service appears under the running station's PlatformServiceContainer as the Hardware Monitor Service.

See the section “Using platform services in a station” for related details. For specific details, see the section on Hardware monitoring configuration in the JACE-NX NiagaraAX Install & Start-up Guides.

Components in platSysmonNxs module

- [platSysmonNxs-HardwareMonitorNxsPlatformServiceWin32, page 148](#)

platSysmonNxs-HardwareMonitorNxsPlatformServiceWin32

HardwareMonitorNxsPlatformServiceWin32 (HardwareMonitorService) is the station's interface to internal environmental parameters in any JACE-NXS host, namely the CPU temperature and board temperature. This service appears under the running station's PlatformServiceContainer as the Hardware Monitor Service.

For more details, see the section on Hardware monitoring configuration in the JACE-NXS NiagaraAX Install & Startup Guide.

Components in platSysmonNxt module

- [platSysmonNxt-HardwareMonitorNxtPlatformServiceWin32 , page 148](#)

platSysmonNxt-HardwareMonitorNxtPlatformServiceWin32

HardwareMonitorNxtPlatformServiceWin32 (HardwareMonitorService) is the station's interface to internal environmental parameters in any JACE-NXT host, namely the CPU temperature and board temperature. This service appears under the running station's PlatformServiceContainer as the Hardware Monitor Service.

For more details, see the section on Hardware monitoring configuration in the JACE-NXT NiagaraAX Install & Startup Guide.

Components in platUsbmon module

- [platUsbmon-UsbMonitorPlatformServiceQnx, page 148](#)

platUsbmon-UsbMonitorPlatformServiceQnx

UsbMonitorPlatformServiceQnx (Usb Monitor Platform Service) is the station's interface to low-level details from monitoring USB ports on the host JACE-7 (JVLN) platform. If client applications are installed that interface with this service, notifications may occur when USB devices are inserted or removed.

Components in platWifi module

- [platWifi-WifiPlatformService , page 148](#)

platWifi-WifiPlatformService

For AX-3.6 and later, WifiPlatformService is the station's interface to a WiFi-equipped JACE, providing views to discover and connect to a wireless 802.11 network, as well as a "secondary

view" to install CA (Certificate Authority) certificate files and client private key files on the controller. Note the latter view is typically not needed, unless installing the JACE on an "enterprise level" wireless network that uses either WPA or WPA2 security, based upon digital certificates.

The WifiPlatformService automatically appears in the station's PlatformServices if the controller has a WiFi adapter. At the time of this document, this means a JACE-7 series (JVLN) controller with a WiFi option. For general information, see "WiFi Configuration".

For complete details, refer to the Engineering Notes document NiagaraAX JACE WiFi option — Engineering Note.

CHAPTER 4 PLATFORM PLUGIN GUIDES

TOPICS COVERED IN THIS CHAPTER

Plugins in platCrypto
Plugins in platDaemon module
Plugin in platDataRecovery
Plugins in platform module
Plugins in platGprs
Plugins in platHwScan
Plugins in platIEEE8021X
Plugins in platPower
Plugins in platWifi module

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.

Access the following summary descriptions on any plugin by selecting **Help > On View** (F1) from the menu, or by pressing F1 while the view is open.

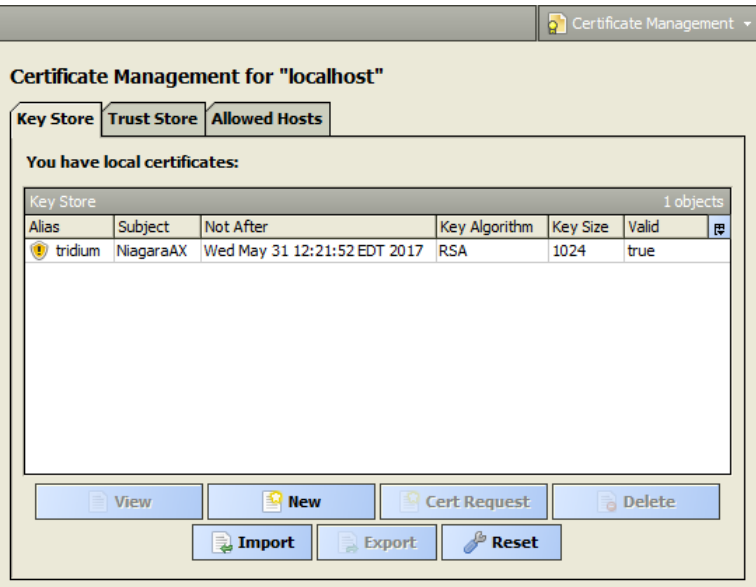
Plugins in platCrypto

- [Certificate Management](#)

platCrypto-CertManagerView

The **Certificate Management** view is a platform view on a host running AX-3.7 or later, provided it is licensed with the **crypto** feature, and has the necessary modules installed (**platCrypto**, possibly others depending on build). This view is also the default view of a station's CertManagerService under its PlatformServices, running on the host as described above.

Figure 131. Certificate Management view



The **Certificate Management** view provides the means to import and export signed certificates (for SSL or TLS secure connections) into the platform's key store and trust store, and to perform other related functions. It is also the default view of the CertManagerService under a station's PlatformServices.

The **Certificate Management** view provides the means to import and export signed certificates (for TLS secure connections) into the platform's key store and trust stores, and to perform other related functions. For a brief overview in this document, see the section "*Certificate Management*".

NOTE: Workbench also provides a similar view, via **Tools > > Certificate Management**. Also included is a related **Tools > > Certificate Signer Tool** view.

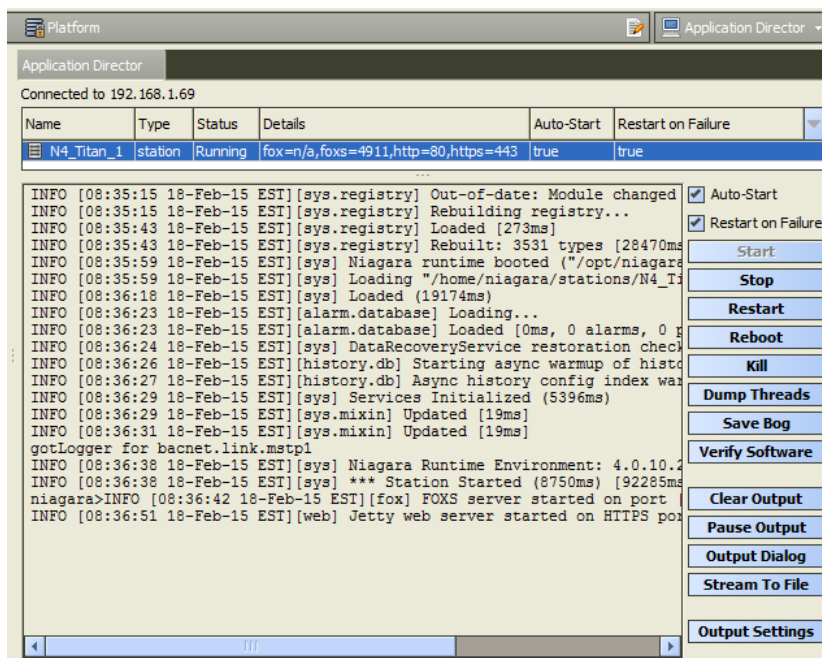
For complete details, refer to the document NiagaraAX SSL Connectivity Guide, including the section "About the Certificate Management dialog".

Plugins in platDaemon module

- [Application Director, page 152](#)
- [Certificate Management](#)
- [Distribution File Installer, page 154](#)
- [Distribution View, page 154](#)
- [File Transfer Client, page 154](#)
- [Lexicon Installer, page 154](#)
- [License Manager, page 155](#)
- [Software Manager, page 155](#)
- [Software View, page 155](#)
- [PlatformAdministration](#)
- [R2 Platform Tool, page 155](#)
- [Station Copier, page 156](#)
- [TCP/IP Configuration](#)
- [User Manager, page 156](#)

platDaemon-ApplicationDirector

The **Application Director** view interfaces to each station whether it is running or not.

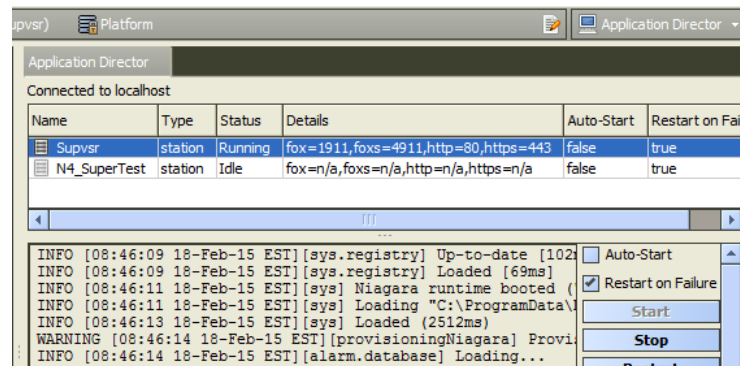
Figure 132. Application Director view, looking at Niagara station

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 checkboxes and buttons

NOTE: In the Application Director for any JACE, 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 133. Application Director for Supervisor host showing multiple stations



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.

platDaemon-DistInstaller

This view allows you to install distribution (dist) files from your local 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.

platDaemon-DistributionView



Distribution View is the dialog that appears when you double-click a distribution file listed in the platform's **Distribution File Installer** view. A number of details is provided about the selected distribution file, including all contents and any dependencies.

platDaemon-FileTransferClient




The **File Transfer Client** is the platform view that allows you to copy files and/or folders between your Workbench PC and the remote platform, as needed.

platDaemon-LexiconInstaller

A Lexicon Installer allows you to install text-based lexicon file sets (for localization) on a remote host.

NOTE: 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.

platDaemon-LicenseManager


 The **License Manager** allows you to view and install files required for Niagara licensing.

platDaemon-PlatformSessionListView


PlatformSessionListView is a tabular view of available platform views when platform connected to a host using a simplified "profiled" Workbench. Using the AX-3.6.44 Workbench or later, you can make platform connections to a Niagara R2-configured JACE-603 or JACE-645. In this case, this view is synonymous with the "main platform view".

For more details, refer to the *Retrofit Board Niagara R2 Install & Startup Guide*.

platDaemon-SoftwareManager

 The **Software Manager** is the platform view you use to install, upgrade, or remove modules in the connected Niagara platform.

platDaemon-SoftwareView

 Software View is the dialog 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.

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. For more details, see the section on "Platform Administration".

NOTE: In AX-3.6.44 or later Workbench, when platform-connected to a JACE-603 or JACE-645 controller (JACE-403 or JACE-545 controller that was upgraded with an NPM6E processor-based "retrofit board"), the Platform Administration view provides a means to configure the unit for either Niagara R2 operation or NiagaraAX operation. Refer to the *Retrofit Board Niagara R2 Install & Startup Guide* for complete details, including the section "Platform Administration".

platDaemon-R2PlatformTool

Starting in AX-3.6.44 or later of Workbench, when platform-connected to a JACE-603 or JACE-645 controller that has been configured for Niagara R2, the **R2 Platform Tool** is an available platform view. Note that JACE-603 or JACE-645 controllers are JACE-403 or JACE-545 controllers that have been upgraded with an NPM6E processor-based "retrofit board", which runs the QNX OS.

The **R2 Platform Tool** view provides a number of functions formerly done in the Niagara R2 "Admin Tool". In this case, some Workbench platform views vary, while other views and functions do not apply to the Niagara R2 host, including the usual "Commissioning Wizard". Refer to the Retrofit Board Niagara R2 Install & Startup Guide for complete details, including the section "R2 Platform Tool".

Alternatively, a retrofit board-upgraded JACE-603 or JACE-645 controller can be configured for AX-3.6 or later, in which case the **R2 Platform Tool** platform view does not apply. In this case you use the normal NiagaraAX commissioning process, the same as for any other QNX-based controller.

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 JACE station or a locally running station. You can also delete and rename stations using this view.

platDaemon-StationTextSummaryEditor

StationTextSummaryEditor is the dialog that appears when you click the export tool button when using the **Application Director** view. Setup in this dialog allows you to include/exclude the platform summary data, platform daemon console output, station console output, as well as limit both the daemon and station output.

platDaemon-TcplpConfiguration

TCP/IP Configuration is the platform view you use to configure a remote JACE host's TCP/IP settings. Typically, you make initial settings when you first commission a JACE, where this view is one step in the platform's Commissioning Wizard. For more details, see "TCP/IP Configuration".


platDaemon-UserManager

The **User Manager** is a platform view available only for Win32 based hosts (e.g. JACE-NXT). It allows you to manage Windows OS user and group accounts local to that host, which otherwise would require accessing "Administrative Tools" in Windows on that host.

For more details, see the section "User Manager".

Plugin in platDataRecovery

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.

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 *Data Recovery Service Guide*.

Plugins in platform module

platform-LicensePlatformServicePlugin

License Platform Service Plugin allows you to manage the 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.

platform-NtpPlatformServiceEditorLinux

Ntp Platform Service Editor Linux is the default view of a station's **NtpPlatformServiceLinux**, which provides the platform interface to the NTP daemon (process) running on a Linux-based host. This view provides access to a few related settings, plus allows specifying one or more remote time servers.

For more details, see the section on “Ntp Platform Service Editor Linux”.

platform-NtpPlatformServiceEditorQnx



Ntp Platform Service Editor Qnx is the default view of the station's **NtpPlatformServiceQnx**, which provides the platform interface to the NTP daemon (process) running on a JACE controller. This view provides access to a few related settings, plus allows specifying one or more remote time servers.

platform-NtpPlatformServiceEditorWin32



Ntp Platform Service Editor Win32 is the default view of a Windows platform station's **NtpPlatformServiceWin32**, which provides the platform interface to the Windows Time service (W32Time) on the host platform's Windows OS. For details, see “About the Ntp Platform Service Editor Win32”.

platform-PlatformServiceContainerPlugin

■ The **Platform Service Container Plugin** allows you to view and edit platform parameters on the host running the opened station. It is the default view for a station's **PlatformServices** container.

platform-PlatformServiceProperties

■ **PlatformServiceProperties** allows you to view and edit platform parameters on the host running the opened station, using a property sheet.

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 JACE platform running the station. If the station is running on a Windows platform, this view is read-only.

platform-SystemPlatformServicePlugin

■ **System Platform Service Plugin** allows you to view and edit platform parameters on a Windows-based host running the station, and is the default view on the station's **SystemService** (**SystemPlatformServiceWin32**).

platform-SystemPlatformServiceQnxPlugin


■ **System Platform Service Qnx Plugin** allows you to view and edit platform parameters on a JACE platform running the station, and is the default view on the station's **SystemService** (**SystemPlatformServiceQnx**).

platform-TcpIpPlatformServicePlugin

🔒 **Tcp Ip Platform Service Plugin** allows you to manage the host's TCP/IP settings under a station's **PlatformServices** container. It provides the same interface as the **TCP/IP**

Configuration view in a platform connection. If the station is running on a Windows platform, this view is read-only.

platform-WorkbenchLicenseManager

 **Workbench License Manager** allows you to browse and manage the contents of your Workbench PC's "local license database." For more details, see "*Workbench License Manager*".

Plugins in platGprs

platGprs-GprsConfiguration

Gprs Configuration (GPRS Modem Configuration) is the platform view used to configure the wireless GPRS modem option card that may be installed in the host platform. For general details, see "GPRS Modem Configuration".

NOTE: For complete details, refer to the Engineering Notes document *GPRS modem option*.


platGprs-GprsPlatformServicePlugin

The **Gprs Platform Service Plugin** is the default view on the GprsPlatformService in a station running on a JACE with a wireless GPRS modem option card installed (AX-3.6, or build AX-3.5.35 or later). It provides the identical interface as the platform view **GPRS Modem Configuration**. For general details, see "GPRS Modem Configuration".

For complete details, refer to the Engineering Notes document *GPRS modem option*.

Plugins in platHwScan

platHwScan-HardwareScanServiceView

 The **Hardware Scan Service View** is the default view on the platform service **HardwareScanService** in a station, providing that the JACE platform has the platHwScan module installed, along with the appropriate platHwScanType module. This view provides a graphical diagram of communication ports and other features on the hosting JACE platform, including callouts to a table that explain the location, description (such as COM2), port type, and status.

Plugins in platIEEE8021X

NOTE: IEEE 802.1X support is available only for QNX-based "Hotspot" JACEs (JACE-3, JACE-6, JACE-7 series) using AX-3.8.

platIEEE8021X-IEEE8021XAdapterSettingsEditor

The **IEEE 8021X Adapter Settings Editor** is the default view on an en0 or en1 container (**IEEE8021XAdapterSettings**) component, accessible through the property sheet of JACE station's platform service for IEEE 802.1X configuration (IEEE8021XPlatformService). Each view is identical to one of two separate tabs in this platform service's default view (or, one of two tabs in the identical **IEEE 802.1X Configuration** platform view). For complete details, refer to the 3.8 document *NiagaraAX IEEE 802.1X Configuration — Engineering Note (AX-3.8)*.

platIEEE8021X-IEEE8021XDaemonSessionPlugin

The **IEEE 802.1X Configuration** view is an available platform view on AX-3.8 JACE platforms, providing the platIEEE8021X module is installed and is licensed with the "ieee8021x" feature. This view is identical to the platform service **IEEE 802.1X Platform Service Plugin**, with separate tabs for each of the two Ethernet adapters on the JACE controller (en0 for LAN1, en1 for LAN2). For complete details, refer to the 3.8 document *NiagaraAX IEEE 802.1X Configuration — Engineering Note (AX-3.8)*.

platIEEE8021X-IEEE8021XPlatformServicePlugin

The **IEEE 8021X Platform Service Plugin** is the default view on a JACE station's platform service for IEEE 802.1X configuration (IEEE8021XPlatformService). This view is identical to the platform view **IEEE 802.1X Configuration**, with separate tabs for each of the two Ethernet adapters on the JACE controller (en0 for LAN1, en1 for LAN2). For complete details, refer to the document *NiagaraAX IEEE 802.1X Configuration — Engineering Note (AX-3.8)*.

Plugins in platPower

platPower-JavelinaBatteryPlatformServicePlugin

■ The **Javelina Battery Platform Service Plugin** is the default view on the platform service PowerMonitorService in a JACE-7 (700) series controller. This view provides parameters for changing the shutdown delay time, as well as alarm source configuration settings. For related details in this document, see JACE power monitoring.

Typically, support is enabled and configured at JACE commissioning time. For related details, see "JACE power monitoring configuration" in the latest JACE NiagaraAX Install & Startup Guide.

platPower-PowerMonitorPlatformServicePlugin

■ The **Power Monitor Platform Service Plugin** is the default view on the platform service PowerMonitorService 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 *Data Recovery Service Guide*.

Plugins in platWifi module

platWifi-WifiConfiguration

Wifi Configuration is the platform view available in a WiFi-equipped JACE to discover and connect to a wireless 802.11 network. This platform view appears only if the controller has a WiFi adapter, this means a JACE-7 series (JVLN) controller with a WiFi option. For general information, see "WiFi Configuration".

NOTE: For complete details, refer to the Engineering Notes II document *NiagaraAX JACE Wi-Fi option*.

platWifi-WifiPlatformServicePlugin

The **Wifi Platform Service Plugin** is the default view on a station's WifiPlatformService, and is identical to the platform **Wifi Configuration** view. The platform service is available in a Wi-Fi-equipped JACE, to discover and connect to a wireless 802.11 network. This means a JACE-7 series (JVLN) controller with a WiFi option. For general details, see "WiFi Configuration".

NOTE: Although the JACE-8000 controller includes a WiFi feature, WiFi is not supported when running AX-3.8U1.

For complete details, refer to the Engineering Notes II document *NiagaraAX JACE WiFi option*.

platWifi-WifiSecurityManager

Wifi Certificate Manager (WifiSecurityManager) is the platform view available in a WiFi-equipped JACE to import CA (Certificate Authority) certificates and client private key files. This can allow the JACE to access to an "enterprise level" wireless 802.11 network that uses either WPA or WPA2 security with digital certificates. For general information, see "Wifi Certificate Manager".

NOTE: Although the JACE-8000 controller includes a WiFi feature, WiFi is not supported when running AX-3.8U1.

This view is also a secondary view on a station's WifiPlatformService, with the primary view the WifiPlatformServicePlugin.

For complete details, refer to the Engineering Notes II document *"NiagaraAX JACE WiFi option"*.

CHAPTER 5 LICENSE TOOLS AND FILES

TOPICS COVERED IN THIS CHAPTER

Workbench License Manager

Request License

About the local license database

About license archive (.lar) files

About license files

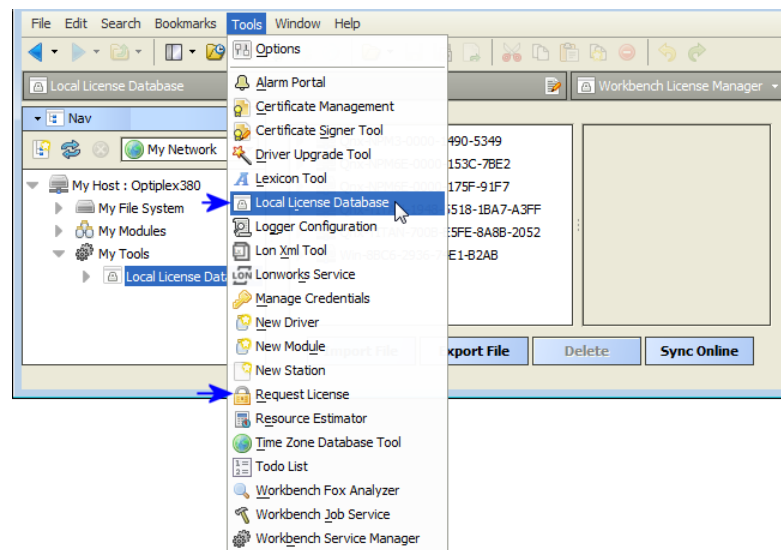
This appendix provides details about the Workbench tools related to Niagara license files, including license-management. Also included are details on the contents of license files.

The following subsections are included:

- License-related Workbench tools

Unlike platform views (which require a platform connection), or equivalent **PlatformServices** plugin views (requiring a station connection), Workbench tools are available whenever running full Workbench. Find Workbench tools on the **Tools** menu, as shown below.

Figure 134. Tools menu in Workbench



License-related tools include:

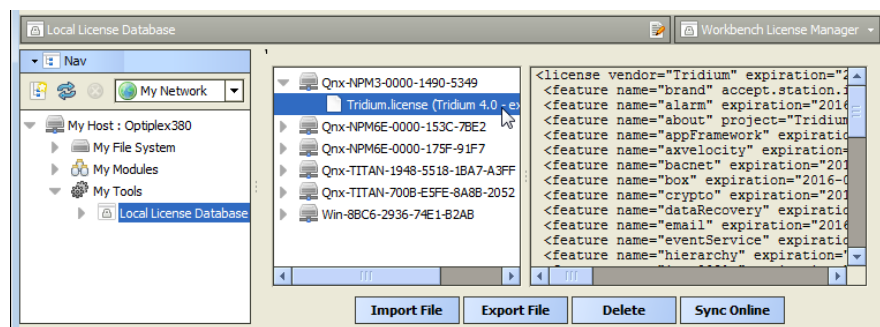
- **Workbench License Manager** tool
- **Request License** tool
- The following sections cover license management topics, in addition to Workbench License tools:
 - “About the local license database”
 - “About license archive (.lar) files”
- “About Niagara license files”
 - “Items common to all license files”
 - “Controller hardware features”
 - “Driver attributes”

- “Driver types”
- “Applications”
- “Global capacity licensing”
 - “Capacity licensing operation and recount”
 - “Checking capacity licensing status”
 - “Capacity licensing fault notifications”
 - “Capacity licensing notes about histories”

Workbench License Manager

The **Workbench License Manager** view is available via **Tools > Local License Database**.

Figure 135. Workbench License Manager



As shown above, this view lets you browse and manage the contents of your “local license database.”

NOTE: For details about the license database structure, see [“About the local license database”](#).

This view provides a two-pane window into all the license files and parent “host ID” folders, where

- Left pane provides tree navigation, where you can expand folders and click (to select) license files.
- Right pane shows the text contents of any selected license file.

Buttons at the bottom of this view provide a way to manage the contents of your local license database, and are described as follows:

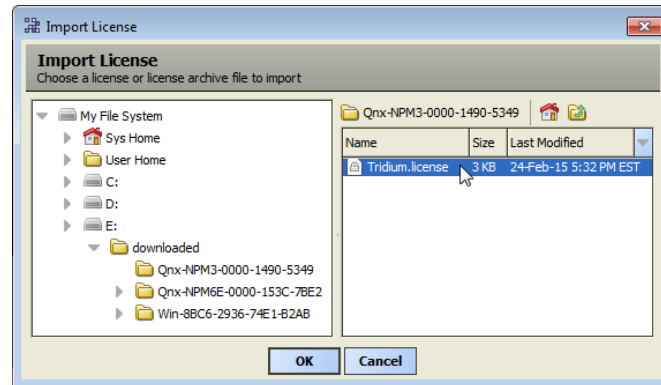
- [Import File, page 163](#) — Always available, this allows you to add license file(s) from a local license file or license archive (.lar) file.
- [Export File, page 163](#) — Always available, this allows you to save all licenses (or any selected licenses) locally, as a license archive file.
- [Delete, page 164](#) — This allows you to delete licenses from your Workbench local license database.
- [Sync Online, page 165](#) — Typically available if you have Internet connectivity. This lets you *update* all licenses (or any selected licenses) in your local license database with the *most current* versions, via the online licensing server.

Import File using Workbench License Manager

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).

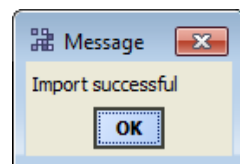
Only two types of files appear for selection.

Figure 136. 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 137. Import success



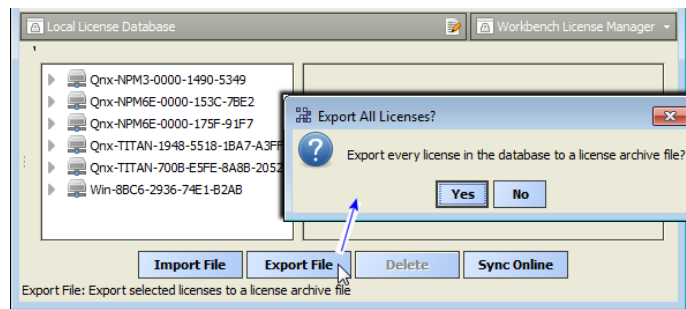
If any of the license(s) you select to import are older than the ones currently in your local database, meaning that the generated attribute timestamp is earlier, newer license(s) in your local license database are *not* overwritten. However, the same Import successful message popup appears for such file import operations.

Export File

The **Export File** button in the **Workbench License Manager** allows you to save any number (or all) licenses in your local license database locally on your Workbench PC, as a license archive (.lar) file.

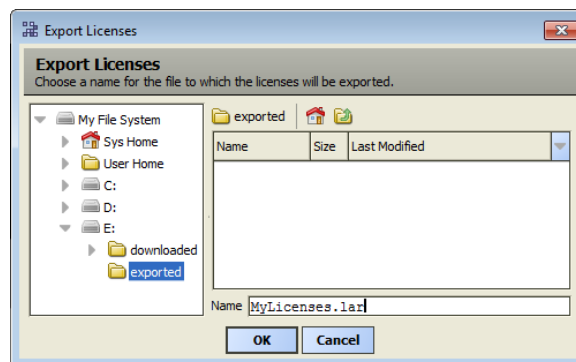
NOTE: The license archive format allows you to easily share saved .lar files (however named) among multiple PCs without overwriting a license file for a different host platform. You can use the “Import File” command in the **Workbench License Manager** to add/update licenses in a license archive, or the equivalent “Import” command from the platform **License Manager** (or similar License Platform Service Plugin). For more details, see [“About license archive \(.lar\) files”, page 168](#).

If you click **Export File** without first selecting any licenses (and/or) host IDs, every license in your local license database will be included in the archive, as noted in a confirmation dialog. See below.

Figure 138. Export All Licenses confirmation dialog

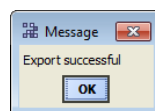
Or, you can select one or more entries in the left pane (host IDs or license files) to include only those selected (highlighted) licenses to be in the exported archive file.

When you click **Yes** (if all) or **Export File** for selected licenses, an **Export Licenses** dialog lets you navigate to the spot to save the .lar file, as shown below.

Figure 139. Export Licenses dialog

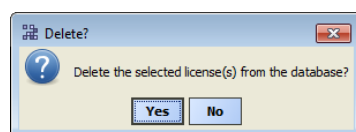
Use the dialog's navigation controls to specify another target folder or drive, as needed. Before saving, you can also *rename* the license archive file, to make it more identifiable. For example, instead of: licenses.lar, you could rename it MyJACE-6s.lar.

Upon export of license(s) to a license archive file, a popup dialog appears, as shown below.


Figure 140. Export file success

Delete

The **Delete** button in the **Workbench License Manager** is enabled when you have one or more host IDs and/or license files selected in the left pane, and produces a confirmation dialog to delete licenses from your local license database, as shown below.

Figure 141. Delete licenses confirmation

Click **Yes** to delete the license(s), or **No** to leave the local license database unchanged.

NOTE: Following a delete, you may need to click the  Refresh button in order to update the left pane contents. Note that if the selected “host ID” folder contained only a .license file, the entire folder is removed with a delete. However, if the folder contained other files (or subfolders), only the .license file is actually deleted, but it will no longer appear in the left pane.

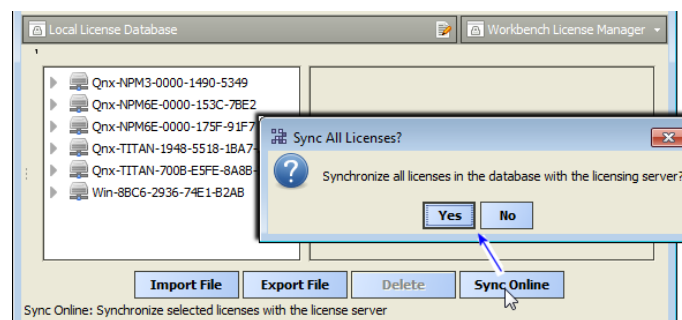
Sync Online

The Sync Online feature in the **Workbench License Manager** allows you to 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.

NOTE: For related details, see “About the licensing server”.

If you click **Sync Online** without first selecting any licenses (and/or) host IDs, every license in your license database will be included in the sync request, as noted in a confirmation dialog. See below.

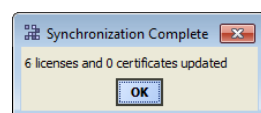
Figure 142. Sync All Licenses confirmation dialog



Or, you can select one or more entries in the left pane (host IDs or license files) to include only those selected (highlighted) licenses to be included in the sync request.

When you click **Yes** (if all) or **Sync Online** for selected licenses, an immediate request is sent to the licensing server. Intermediate popup dialogs may briefly appear while the sync request is handled. The operation concludes with a **Synchronization Complete** dialog, which summarizes the number of licenses and certificate files that were updated in your local license database. See below.

Figure 143. Synchronization Complete dialog

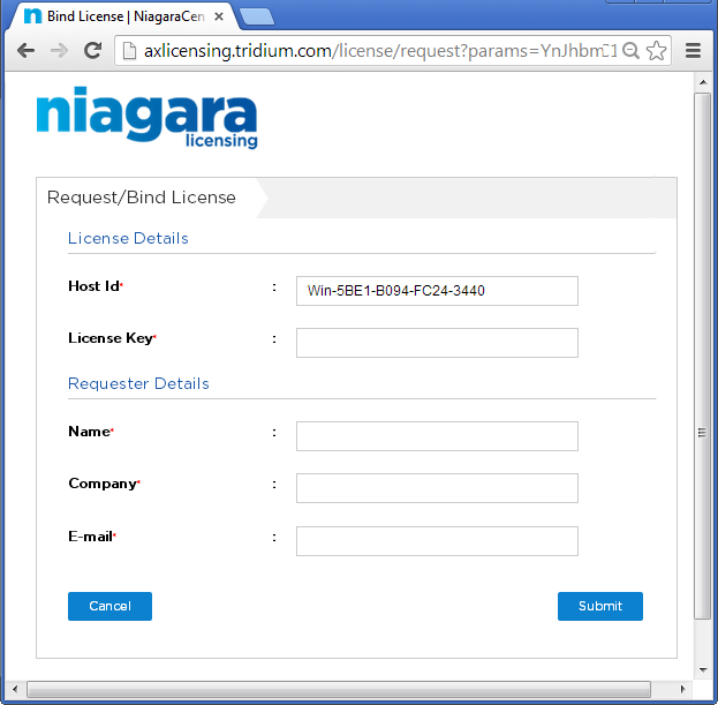


If all licenses (and certificates) were already up-to-date, this dialog will say “0 licenses and 0 certificates updated”.

Request License

In Workbench, selecting **Tools > Request License** opens a **Request/Bind License** form in your default browser. By default, the only pre-filled field in this form is the host ID of your PC. See below.

Figure 144. License request form opens in browser



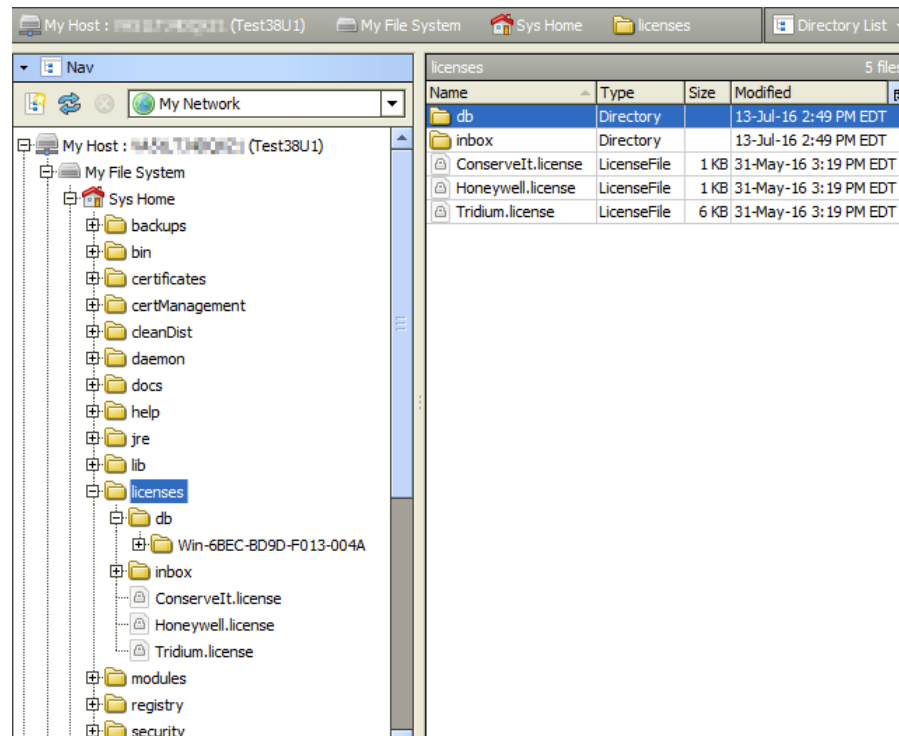
The screenshot shows a web browser window with the address bar displaying `axlicensing.tridium.com/license/request?params=YnJhbmC1`. The page features the Niagara licensing logo at the top. Below the logo, there is a tab labeled "Request/Bind License". Under this tab, the "License Details" section contains two fields: "Host Id" with the value "Win-5BE1-B094-FC24-3440" and "License Key" which is empty. The "Requester Details" section contains three fields: "Name", "Company", and "E-mail", all of which are empty. At the bottom of the form, there are two buttons: "Cancel" and "Submit".

Typically, your Workbench PC is already licensed. Otherwise, you would not be able to successfully start Workbench to request a license.

However, you could use this as a quick method to request a license for another PC on which you have installed Niagara. In that case, in this form you must enter the host ID for that other PC, along with the other pertinent information.

About the local license database

Any Workbench PC (including a Supervisor) has a “local license database”, a structured collection of subfolders and files under its Niagara installation (**Sys Home**) `! /licenses/db` directory. Each subdirectory has a unique Niagara “host ID” name, matching that for some remote host platform. The figure below shows an example license database structure, as viewed in the Nav tree.

Figure 145. Workbench local license database is everything under !/licenses/db

Your local license database is created and managed automatically by Workbench, and updated whenever you perform license operations from platform connections, **PlatformServices** plugins, or when using Workbench tools such as the **Workbench License Manager**. Note that you can see the same directory/file structure when looking at this location on your Workbench PC using Windows Explorer.

NOTE: The license required for your (local) Workbench PC operation is in the root of the licenses folder, named simply by your brand, for example Tridium.license.

See “Workbench License Manager” for more details.

Local license database rationale

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) also facilitates the exchange of licenses among different PCs, and is used in updating/synchronizing licenses to the online licensing server, as well as with provisioning features for Niagara Networks.

Local license inbox

In addition to the `!/licenses/db` folder, there is also a `!/licenses/inbox` folder. The inbox allows “drag and drop” importing 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.

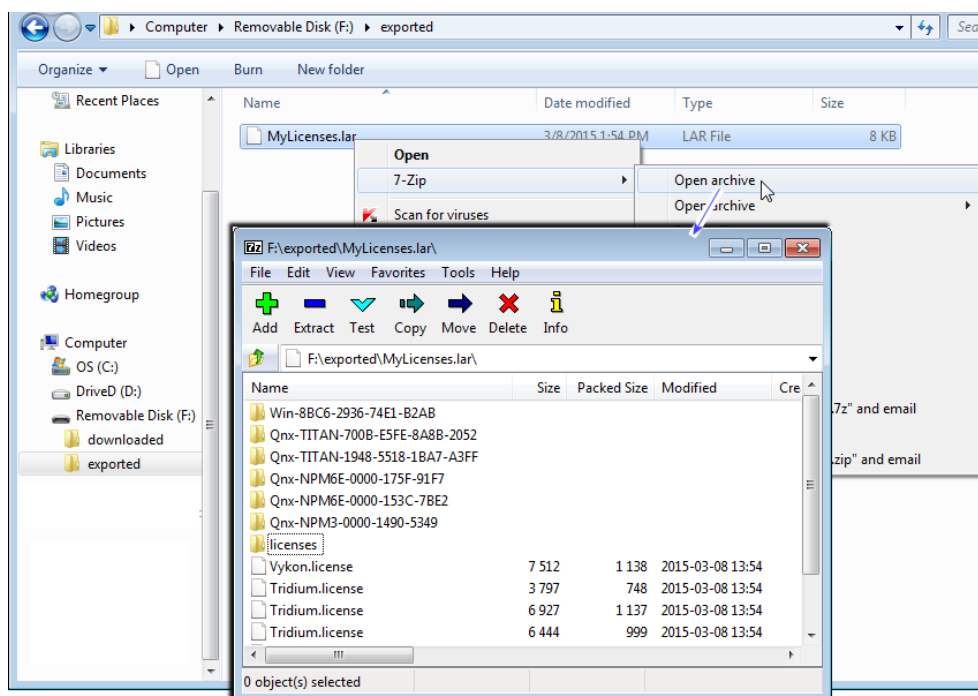
After copying license files and/or .lar files into your inbox subfolder, you need to close and restart Workbench. Then, the appropriate “host ID” named subfolders are automatically created in your local license database, each with the appropriate license file(s). Contents of the `inbox` folder are then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

NOTE: After you restart Workbench, your local license database will be correctly structured. In addition, now you can use the “Sync Online” feature of the **Workbench License Manager** to ensure you have the latest version of all your licenses. See [“Sync Online”](#), page 165.

About license archive (.lar) files

When you use the platform **License Manager** view or the **Workbench License Manager** view (under Workbench **Tools**) to export one or more license files, they are saved in a compressed (Zip compatible) format known as a license archive, that is a file with a “.lar” file extension. Any .lar file is simply a zip of the exported license file(s) that includes the complete “licenses/hostID” folder (subdirectory) structure for any included licenses.

Figure 146. License archive (.lar) is license file(s) in zip format, including folder paths relative to sys home.



The figure above 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. However, if you export a license in the **License Manager** when platform-connected to a remote host, the license archive file contains just the license(s) for that host.

About license files

A 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 (JACE, PC), matched by that host's unique host ID. License files are “digitally signed” by the vendor to prevent tampering.

The following sections provide more details on the contents of a NiagaraAX license file that validates against the Tridium certificate:

- Items common to all license files (license, about, brand, signature)

- Controller hardware features (e.g. dataRecovery, mstp, ndio, serial, others)
- Driver attributes (name, expiration, device.limit, history.limit, point.limit, schedule.limit, parts)
- Driver types (many types, including bacnet, lonworks, modbusTcp, obixDriver, niagaraDriver)
- Applications (email, provisioning, station, web, Workbench, others)

Items common to all license files

license

All license files require an opening `<license>` line, where the last line in the license file is the closing `</license>` tag, and all contents (lines) in between are `<feature>` elements, plus one signature element.

In the first `<license>` line, there are a number of common attributes, described below.

```
<license vendor="Tridium" expiration="never" version="3.8"
hostId="Qnx-NPM6E-0000-153C-6E44" serialNumber="4856"
generated="2015-01-27">
```

vendor

`vendor="Tridium"` - This is always Tridium.

expiration

`expiration="never"` - The expiration date of the license file. After the expiration date the Workbench software fails to start due to a license expired error. Typically, engineering copies of Workbench have expiration dates which expire on an annual basis. License files for actual projects are issued with non-expiring licenses, where this attribute value is "never".

version

`version="3.8"` - The highest release version of software which can be installed in the JACE. If a newer version of software is installed, the JACE will fail on startup with a license version error.

hostId

`hostId="Qnx-NPM6E-0000-153C-6E44"` - Alphanumeric code unique to the specific host. On a Windows-based platform, host ID is generated upon installation of the Niagara software, and typically begins with "Win-", for example "Win-5BE1-B094-FC24-3440".

JACE controllers are assigned a host ID at the factory. The first two segments are "Qnx-Model-" such as "Qnx-NPM6-" for a JACE-6 or "Qnx-TITAN-" for a JACE-8000 controller. The hostId in the license file must match the hostId of the JACE controller, otherwise the JACE cannot run a station.

serialNumber

`serialNumber="329696"` - Applies to a license for a JACE controller only. Designates its unique serial number assigned from the factory. The serial number in the license file must match the serial number of the JACE.

generated

`generated="2015-01-27"` - The date upon which the license file was generated.

brand

For any license with `vendor="Tridium"`, the NiCS (Niagara Compatibility Structure) provides a structure (or schema) that OEMs can use to define the various levels and types of Niagara interoperability that their products will support.

NiCS definitions are contained in this feature item, which is checked by a station or tool when it starts up. There are five attributes to the NiCS: BrandID, Station Compatibility In, Station

Compatibility Out, Tool Compatibility In, and Tool Compatibility Out. These elements can be combined in a variety of ways to achieve unlimited flexibility, and are described below.

```
<feature name=accept.station.in="*" accept.station.out="*"
accept.wb.out="*" "brand" brandId="Distech" accept.wb.in="*" />
```

accept.station.in

`accept.station.in="*"` - A list of brands that this local station will allow Niagara data to come in from. Simply stated from a JACE perspective, "this is the list of brands that I can accept data from". The "*" is a wildcard designation to allow all brands.

accept.station.out

`accept.station.out="*"` - A list of brands that this local station will allow Niagara data to be shared with. Simply stated, "This is the list of brands that I can share data with".

accept.wb.in

`accept.wb.in="*"` - A list of brands that this station will allow to be connected to it for engineering of its application. Simply stated, "This is the list of brands that can engineer me".

accept.wb.out

`accept.wb.out="*"` - A list of brands that this tool is allowed to connect to and engineer. Simply stated, "This is the list of brands that I can engineer".

brandId

`brandId="Distech"` - Every licensed station and tool has a Brand Identifier (BrandID). This field holds a text descriptor acts as the identifier for the product line. Each station or tool can have only one BrandID entry.

about

The "about" feature is used to designate 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:

```
<feature name="about" project="Testing"
owner="Tech Pubs" />
```

project

`project="Tech Pubs"` - Optional attribute to designate a project. This grouping should typically be assigned to all JACE controllers used for a particular project.

owner

`owner="Tech Pubs"` - Optional attribute to designate the name of a person or group responsible for the project, or possibly an end user.

signature

This ending element contains a digital signature which 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 will 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.

```
<signature>MCwCFFOdq4wJcYgVhTVtrf0oSyuCDCwjAhRj+
H9pNxQGStBnhEkIqK8rONB10g==</signature> </license>
```

Controller hardware features

Some license features are specific to JACE controller hardware capabilities.

Alphabetically, these include features `dataRecovery`, `jre8qnx`, `mstp`, `ndio`, `nrio`, and `serial`.

dataRecovery

This feature licenses a station's **DataRecoveryService**, sourced from the its platDataRecovery 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.

```
<feature name="dataRecovery" expiration="never" parts="NPB-SRAM">
```

ibmj9j2me

Starting in AX-3.5, for QNX-based JACE platforms, this feature licenses the IBM J9 Java virtual machine (VM) to be able to run on the controller, if applicable. The "rev" attribute reflects the version number of the J9 VM, as found in its version.xml file. Note starting in AX-3.6, some JACE controllers run the (Oracle) Sun Hotspot VM instead, and so require another sunj2se feature.

```
<feature name="ibmj9j2me" expiration="never"
rev="2.3" parts="VM-J9"/>
```

maxheap

This feature determines the maximum size of the Java heap for either JACE-2 or JACE-6 series controllers. In the absence of this feature, the maximum heap size is limited to 16 MB for JACE-2, and 48 MB for JACE-6. When this license feature is present, the maximum heap size is limited to 48 MB for JACE-2 and 96 MB for JACE-6. The feature may not be available on all JACE-2 controllers, earlier models were manufactured with 64 MB RAM chips. Verify the amount of physical memory in the controller before attempting to order the memory upgrade option.

```
<feature \name="maxHeap" expiration="never"
parts="NPM-128"/>
```

mstp

This feature determines how many of the available serial ports may be used for BACnet MS/TP communications. Note that features bacnet and serial must also exist in the license file.

```
<feature name="mstp" expiration="never" port.limit="5"
parts="DR-MSTP-AX"/>
```

port.limit

port.limit="5" - This specifies the number of serial ports which 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 then the port limit may be less than the number of physical ports (if the port activation has not been ordered as well).

ndio

This feature enables the NDIO (Niagara Direct Input Output) driver, required to configure and use a JACE controller's Ndio-type I/O modules. Not all JACE controllers support such I/O modules (which attach/chain directly to the controller, using 20-pin connectors); refer to specific JACE controller data sheets to confirm whether this is an available option. Note that in the ndio features line (below), a "device" equates to an "Ndio Board", and that history and schedule limits have no practical application.

```
<feature name="ndio" expiration="never" device.limit="none" history.
limit="none" point.limit="none" schedule.limit="none"
parts="DR-NDIO"/>
```

Refer to the *Niagara NDIO Guide* for related details.

nrio

This feature enables the NRIO (Niagara Remote Input Output) driver, required to configure and use a 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). Note that in the nrio features line (below), a “device” equates to an “Nrio16Module”, and that history and schedule limits have no practical application.

```
<feature name="nrio" expiration="never" device.limit="16"
history.limit="none" point.limit="none" schedule.limit="none"
parts="DR-NRIO"/>
```

Refer to the *Niagara Nrio Guide* for related details.

serial

This feature enables the use of JACE serial ports for various drivers, for example aapup or modbusAsync. Note that the JACE license needs this serial feature in addition to any specific driver feature. Only one serial feature line is needed, regardless of number of serial-based drivers. Note that in the case of a JACE used for BACnet MS/TP, it would require this serial feature and driver features bacnet and mstp.

```
<feature name="serial" expiration="never"/>
```

sunj2se

Starting in AX-3.6 for some QNX-based JACE platforms, this feature licenses the (Oracle) Sun Hotspot Java virtual machine (VM) to be able to run on the controller, if applicable. The “rev” attribute reflects the version number of the Hotspot VM, as found in its version.xml file. Note some JACE controllers must run the IBM J9 VM instead, and so require a different ibmj9j2me license feature.

```
<feature name="sunj2se" expiration="never"
rev="5" parts="VM-SUN"/>
```

Driver attributes

Each driver is enabled by a feature line (element) in the license file. Most of the drivers utilize the same attributes within that feature. The most common driver attributes are shown below.

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

The various “limit type” attribute values can be either “none” or a numerical (limit) value, for example `device.limit=32`. Note that 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, none of the Modbus-related drivers have any history or schedule import/export capability, due to the simplicity of the Modbus protocol. Thus, “`history.limit`” and “`schedule.limit`” values have no significance in the feature for a Modbus driver.

NOTE: The various “limit type” attribute values can be either “none” or a numerical (limit) value, for example `device.limit = 32`. Note that a limit value of none means unlimited, whereas a limit value of 0 means none allowed. Although most drivers include all the attributes shown in the feature line above, some attributes may not apply to a specific driver, with the exceptions of `point.limit` and `device.limit` attributes, which typically do apply to any driver. For example, none of the Modbus-related drivers have any history or schedule import/export capability, due to the simplicity of the Modbus protocol. Therefore, “`history`.”

name

Feature name of the driver, often the same as the actual module (.jar file) name, for example bacnet, lonworks, etc.

expiration

Each driver has an expiration date which 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 where the main license file is non-expiring.

device.limit

This attribute designates a license limit on the number of devices which may be added to this specific driver network in the station database. Above this limit, any added device component (and all its child components) will be 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 device.limit="none", this does not mean that you can exceed the normal limit of 64 devices per segment.

history.limit

This attribute limits the number of Niagara 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) will be in fault, and the associated import/export will not be successful.

point.limit

This attribute 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 will be in fault.

schedule.limit

This attribute limits the maximum number of Niagara 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) will be in fault, and the associated import/export will not be successful.

parts

This is an alphanumeric part code which 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 *name* attribute), and has common attributes.

NOTE: New Niagara drivers are continually developed and offered as products. This section includes some, but not all drivers available. It is included in this section to illustrate how driver features appear in licenses.

Alphabetically, driver types listed here include aaphp, aapup, bacnet, bacnetAws, bacnetOws, fileDriver, lonworks, modbusAsync, modbusCore, modbusSlave, modbusTcp, modbusTcpSlave, obixDriver, opc, niagaraDriver, rdbOracle, rdbSqlServer, snmp, videoDriver and zwave.

aaphp

Enables the American Auto-Matrix Public Host Protocol (PHP) driver. The serial feature is also required.

aapup

Enables the American Auto-Matrix Public Unitary Host (PUP) driver. The serial feature is also required.

bacnet

Enables functionality of the BACnet driver for BACnet/Ethernet and BACnet/IP. If a JACE controller, other features can be added to enable BACnet MS/TP communications over serial ports: mstp and serial.

```
<feature name="bacnet" expiration="never" device.limit="none"
export="true" history.limit="none" point.limit="none"
schedule.limit="none"/>
```

Refer to the *BACnet Guide* for details on all BACnet integration with Niagara.

export

export="true" - When set to "true" this field enables BACnet server operation. When the field is set to "false" only BACnet client operation is permitted.

NOTE: When BACnet export is enabled, any station histories and/or schedules that are exported to BACnet do not count towards any history.limit or schedule.limit values in the license (if any).

bacnetAws

Provides added functionality as *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). The bacnet feature is also required in the license. More details are available in an appendix in the *BACnet Guide*.

bacnetOws

Provides added functionality as *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 in the *BACnet Guide*.

dust

Enables the Dust Wireless Mesh driver. Details are available in the *NiagaraAX Dust User Guide*.

fileDriver

Enables the File driver, used to import comma or tab delimited text files and convert into histories. For more details, see the “file-FileNetwork” section in the *Niagara Drivers Guide*.

jen6lp

Applies to an AX-3.6 or later QNX-based JACE with an installed Sedona Jennic option card (with jennic license feature). Enables a network of wireless Jennic-based Sedona Framework devices (SedonaJen6lpNetwork), including device and point limits. Limits are independent of any in a sedonanet license feature (if present). Additionally, if the "export" attribute is set to true, Chopan server functionality is provided. Chopan facilitates operation of hibernating devices within the Sedona Jennic network.

```
<feature name="jen6lp" expiration="never"
export="true" device.limit="none" history.limit="none"
point.limit="500" schedule.limit="none" parts="DR-SOX-JEN-AX"/>
```

To engineer and support, Workbench requires Sedona Framework TXS software installed, using the Sedona Installer tool—see the NiagaraAX Sedona Installer Guide for related details.

jennic

Required for a QNX-based JACE controller to use a Sedona Jennic option card, where the jen6lp license feature is also required to enable a network of wireless Sedona Jennic devices (SedonaJen6lpNetwork). The jennic feature is also required by a Workbench host to use a USB wireless adapter (coordinator), as the tools New Jennic Wireless Adapter and Jennic Serial Port Tool do not function without it.

lonworks

Enables the Lonworks driver. Utilizing the driver also requires a LON interface on the JACE controller. Most JACE controller models require an optional Lonworks interface card to be installed. More details are available in the *Lonworks Guide*.

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 modbusCore and serial features are also required.

modbusCore

Required by a 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 *Modbus Guide*.

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 modbusCore and serial 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 modbusCore 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 modbusCore feature is also required

obixDriver

Enables the oBIX driver. The driver supports the oBIX protocol, which is M2M (Machine-to-Machine) communications via XML over TCP/IP. Refer to the *Obix Guide* for related details.

```
<feature name="obixDriver" expiration="never" device.limit="none"
export="true" history.limit="none" point.limit="none"
schedule.limit="none"/>
```

export

export="true" When set to "true" this field enables oBIX server operation. When the field is set to "false" only oBIX client operation is permitted.

opc

Enables the OPC client driver, and is only available on Windows-based platforms because of the protocol's dependency of Windows. Refer to the *OPC Guide* 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.

```
<feature name="niagaraDriver" expiration="never" virtual="true"
schedule.limit="none" point.limit="none" history.limit="none"
device.limit="none" parts="ENG-WORKSTATION"/>
```

For more details, refer to the Niagara Drivers Guide section “About the EC-Net Network”.

rdbDb2

Enables the Relational Database Driver using the IBM DB2 database format. This driver allows exporting of histories from the NiagaraAX station to an IBM DB2 database. The driver does not include the DB2 software, which must be purchased separately from a third party source.

```
<feature name="rdbDb2" expiration="never"
parts="ENG-WORKSTATION"/>
```

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.

```
<feature name="rdbOracle" expiration="never"
parts="ENG-WORKSTATION"/>
```

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.

```
<feature name="rdbSqlServer" expiration="never" history.limit="10"
historyImport="true" parts="ENG-WORKSTATION"/>
```

sedonanet

Enables the Sedona Framework Ethernet/Wi-Fi Network (SedonaNetwork) in a AX-3.6 or later JACE or Supervisor, including device and point limits. Device and point limits are independent from those in the jen6lp license feature on the host, if present.

```
<feature name="sedonanet" expiration="never"
export="false" device.limit="none" history.limit="none"
point.limit="500" schedule.limit="none" parts="DR-SOX-ETH-AX"/>
```

snmp

Enables the SNMP (Simple Network Management Protocol) driver, which allows sending and receiving SNMP messages. Refer to the *Snmp V1/V2 Driver Guide* for related details.

```
<feature name="snmp" expiration="never" device.limit="none"
history.limit="none" point.limit="500" schedule.limit="none"/>
```

videoDriver

Enables the Niagara Video Framework driver (modules `nvideo`, `videoDriver`, `nDriver`) that provide the foundation to integrate select commercial off-the-shelf video surveillance and recording systems into a Niagara station. Depending on the specific video hardware used, one or more vendor-specific license feature entries are also typically required. Refer to the *Video Framework Guide* for related details.

zwave

Applies to a JACE controller with an installed Z-Wave option card, or any host platform with a third-party, serially-connected, Z-wave gateway device. The serial feature is also required. Enables a network of wireless Z-Wave devices (`ZWaveNetwork`), including device and point limits. Refer to the *Niagara Z-Wave Driver Guide* for related details.

Applications

Alphabetically, application types listed here include `box`, `crypto`, `eas`, `email`, `fips140-2`, `genericAppliance`, `ieee8021x`, `ldapv3`, `mobile`, `provisioning`, `sedonaProvisioning`, `sox`, `station`, `tenantBilling`, `web`, and `workbench`. Applications `station`, `web`, and `workbench` have special importance, and are summarized first.

station

Enables a station to be run, and is present in any JACE platform, as well as a Supervisor.

```
<feature name="station" expiration="2015-04-01"
resource.limit="none" guestEnabled="true"/>
```

The station feature may not be present in a license for an engineering workstation (PC), unless specifically ordered with it. Optional attributes are listed below.

resource.limit

`resource.limit="none"` - If the `resource.limit` flag is specified (in kRUs), then the station displays a warning on startup if the actual resource units exceed the limit resource units. If the limit is exceeded by 110% then the station will not boot at all. This limit is normally only specified in (NiagaraAX) SoftJACE license files.

guestEnabled

`guestEnabled="true"` - Must be present and true, or else 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—thus is unavailable on any host with a "non-expiring" license.

web

The web feature must be present to start 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 appropriate `faultCause`.

NOTE: Full Workbench can connect to a station (via Fox connection) even if the web feature is missing or expired.

```
<feature name="web" expiration="never" ui="true"
ui.wb="true" ui.wb.admin="true"/>
```

ui

`ui="true"` - This flag allows browser access to users with an HTML5 Hx Profile.

Note: If `ui="false"`, users cannot access the browser UI with either HTML5 Hx or Wb web profiles. No browser access is allowed, except for Spy pages.

ui.wb

`ui.wb="true"` - This flag allows browser access to users with a Wb web profile.

Note: If `ui.wb="false"`, users with an HTML5 Hx web profile still have browser UI access, as long as `ui="true"`.

ui.wb.admin

`ui.wb.admin="true"` - This flag allows browser users with a Wb web profile access to admin-only views on components, providing 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. Or, if this flag is false, such views are also unavailable to Wb web profile users.

Note: If `ui.wb.admin="false"`, users still have access to the station with a browser, subject to the “ui” and “ui.wb” flags. Property sheet views are available on components. Slot sheets may be available too, providing a user has admin-level permissions on components.

workbench

The workbench feature must be present to start the full version of Workbench (for example, a copy of Distech Controls Niagara Workbench or an OEM-specific Workbench-based application). If the admin flag is false, then all views requiring admin access are unavailable. This feature is included for PC platforms only, with the sole exception of the (NiagaraAX) SoftJACE.

```
<feature name="workbench" expiration="never" admin="true"/>
```

box

This enables a host for Bajascript, a Javascript API (read and write) for Niagara data access from Javascript enabled environment like web browsers. Along with the mobile feature, this license feature is required for mobile application support.

```
<feature name="box" expiration="never" session.limit="none"
parts="ENG-WORKSTATION"/>
```

crypto

Enables SSL (Secure Socket Layer) operation.

```
<feature
name="crypto" expiration="never" ssl="true" parts="SP-SSL"/>
```

Although the crypto feature is used to license SSL connectivity in NiagaraAX releases through AX-3.6, note the SSL implementation changed completely in AX-3.7—for example, the crypto module is no longer used. See the sections below for more details.

SSL in AX-3.7 and later

Starting in AX-3.7, SSL and TLS (Secure Socket Layer and Transport Layer Security) architecture in NiagaraAX changed to a “platform based” foundation, with certificates and exemptions accessed and managed using platform based tools. SSL connections are now possible for all NiagaraAX connectivity, including platform connections, station (Fox) connections, and web browser (WebService) connections. For complete details, refer to the *NiagaraAX SSL Connectivity Guide*.

SSL for releases AX-3.6 and earlier

Prior to AX-3.7, SSL (Secure Socket Layer) connectivity in NiagaraAX used a “station based” model, with an associated “CryptoService” from the crypto module. Only web browser access

of a station using Hx profiles could effectively use SSL. For details on this outdated architecture, refer to the Engineering Notes document *NiagaraAX CryptoService (SSL)*.

eas

This feature enables the Energy Suite application and the associated reports, data points and meters.

```
<feature name="eas" expiration="never" allCostReports="true"
allE2Reports="true" brand=" MyBrand " costMeter.limit="none"
dataPoint.limit="none" parts="AX-DEMO"/>
```

allCostReports

`allCostReports="true"` - If set to "true" all Cost Profiler Reports are enabled. When set to "false" there are additional feature items for each of the specific Cost Profiler Reports that are enabled.

allE2Reports

`allE2Reports="true"` - If set to "true" all E2 Profiler Reports are enabled. When set to "false" there are additional feature items for each of the specific E2 Profiler Reports that are enabled.

costMeter.limit

`costMeter.limit="none"` - Designates the maximum number of Cost Profiler Meters that may be configured in the VES Application, where "none" means unlimited.

dataPoint.limit

`dataPoint.limit="none"` - Designates the maximum number of E2 Profiler Points that may be configured in the VES Application, where "none" means unlimited.

Messaging features

The devices monitored by the system and the services that do the monitoring can communicate status, alarms and reports as needed using direct email and SMS messaging.

The system supports these features:

- The email feature enables a station to communicate with an SMTP server:

```
<feature name="email" expiration="never"/>
```

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.

fips140-2

This feature enables an AX-3.8 Niagara Station to operate in "FIPS 140 Mode" (Federal Information Processing Standard 140) using cryptographic software fully compliant with FIPS 140-2. A special FIPS distribution must be installed on the host.

```
<feature name="fips140-2" expiration="2016-03-13"
lib="entrust" parts="PROTO-FIPS"/>
```

genericAppliance

This feature enables the NiagaraAX Generic Appliance Application.

```
<feature name="genericAppliance" expiration="never"
vendor.name="" parts="GA-GENERIC"/>
```

ieee8021x

ReferenceTopicID: aLicenseFileApplications_ieee8021x

This feature enables the AX-3.8 "Hotspot" QNX-based JACE (JACE-3,-6,-7) to be able to join a wired IEEE 802.1X wired-authentication network.

```
<feature name="ieee8021x" expiration="never"
parts="PROTO-8021X"/>
```

ldapv3

This feature enables AX-3.8 hosts to use the LdapV3UserService or LdapV3ADUserService in place of the standard baja UserService. These LDAPv3-compatible user services are available in the AX-3.8 ldap module, in addition to the former LDAPv2-compatible user services (which require no license feature).

If the kerberos attribute is "true", the AX-3.8 host is licensed for Kerberos authentication with LDAPv3.

```
<feature name="ldapv3" expiration="never"
kerberos="true" parts="ENG-WORKSTATION"/>
```

mobile

This enables the host to support the Mobile application framework, for station support of web browser access from mobile devices like cell phones or tablets. The host also requires to be licensed with the box feature for Bajascript support.

```
<feature name="mobile" expiration="never" history="true"
schedule="true" alarm="true" px="true" propsheet="true"
parts="ENG-WORKSTATION"/>
```

provisioning

Enables the operation of Niagara host provisioning, typically used to automate routine maintenance of Niagara system such as JACE software upgrades, file distribution and backups. It applies to an Supervisor platform only. Provisioning uses the BatchJobService and a "network extension model" (e.g. a "ProvisioningExt" under the NiagaraNetwork), sourced respectively from modules **batchJob** and **provisioningNiagara**.

```
<feature name="provisioning" expiration="never"/>
```

sedonaProvisioning

Required by a Workbench host to provision Sedona Framework devices, including operations Get App, Put App, Manage Kits, Backup App, and Restore App.

```
<feature name="sedonaProvisioning" expiration="never"
parts="AX-DEMO"/>
```

tenantBilling

Enables the NiagaraAX Tenant Billing Application.

```
<feature name="tenantBilling" expiration="never"
tenant.limit ="none" parts="S-TBS-AX"/>
```

tenant.limit

tenant.limit="none" - Designates the maximum number of tenants that may be configured in the station database, where "none" means unlimited.

CHAPTER 6 TIME ZONES

TOPICS COVERED IN THIS CHAPTER

Time zones and terminology

Selecting a time zone

Updating a historical time zone database

Platform configuration of a 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. This section provides details on time zone selection in Niagara, including the currently used “historical time zone database.”

The following main sections are included:

- [“Time zones and terminology”, page 181](#)
-

NOTE: Workbench provides a special “Time Zone Database Tool” that lets you explore the historical time zone database on the local host.

Time zones and terminology

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). See [“UTC”, page 181](#) and [“DST”, page 181](#)

UTC

Coordinated Universal Time (UTC) is the recognized atomic-clock standard of reference time, largely replacing GMT (Greenwich Mean Time) as reference time. Time zones are commonly expressed as negative or positive offsets from UTC time.

DST

Daylight Saving Time (DST) is used as a means of maximizing daylight hours during normal waking hours, and is used by many (but no means all) time zones. DST is a twice-yearly event acting upon local time, as follows:

- 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.”
- 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. These rules vary widely from zone to zone, since DST policies are set by national and regional governments. Also, DST policies are subject to change for this same reason—as in the recent 2007 change for all U.S. time zones that observe DST.

In the 2007 U.S. DST changes, the DST start time was changed to “first Sunday on or after the 8th in March” (from “first Sunday on or after the 1st in April” for 2006 and prior years). The DST end time was changed to “first Sunday on or after the 1st in November” (from “last Sunday in October” for 2006 and prior years).

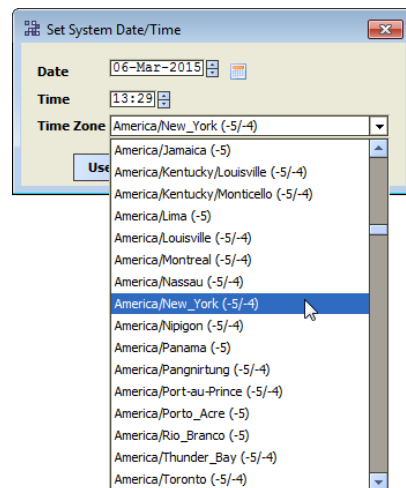
NOTE: A change in DST rules for a time zone can cause *issues* in Niagara when displaying historical data (histories and alarm records), particularly when applying new (current) DST rules to records collected using prior (old) DST rules. For more details, see [“About the historical time zone database”](#).

Selecting a time zone

Platform configuration of a Niagara host includes the setting of date and time, which includes specifying its local time zone.

Typically, you specify time zone in a JACE controller during its initial commissioning in a platform connection, when running the **Commissioning Wizard**. Or, you can do this at any time using the **Platform Administration** view (function “Change Date/Time”).

Figure 147. Selecting time zone from Change Date/Time selection in Platform Administration View



After a station is installed and running, you can also specify a JACE’s time zone using one of the station’s **PlatformService** views (“Platform Service Container Plugin” or “System Date and Time Editor”).

In any case, 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)
```

Note 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*”.

About the historical time zone database

AX-3.8 (and earlier) hosts use a time zone database with “historical perspective,” such that display of a station’s time-stamped data (histories and alarms) collected in time zones under “prior rules” (typically DST-related) will always display with the original (and correct) collected time.

In this database, a history of changes for applicable time zones are stored, such that multiple definitions for a time zone may exist, including past definitions as well as its current definition.

Time zones in this database are not user editable. However, this time zone database offers advantages for NiagaraAX jobs where accrued histories or alarms have spanned across different time zone definition eras.

The historical time zone database is sourced from the public "Olson Time Zone Database," and updated (synchronized to it) upon each AX build. It is implemented using a "timezones.jar" file that contains histories of changed rules for any so-affected time zones, along with changes to the NRE (Niagara Runtime Environment) that support this new method.

This allows display of a station's time-stamped data (histories and alarms) collected in time zones under "prior rules" (typically DST-related) to display with the original (and correct) collected time.

There are 2 associated files with historical time zones in an AX-3.8 or earlier host's `!lib` folder, described as follows:

- `timezones.jar`: The time zone database, in Java archive format. Contains a collection of binary files, one representing each time zone. Upon each build of AX, this `timezones.jar` file is updated (synchronized) to the Olson Time Zone Database to maintain historical accuracy.
- `system.properties`: The file responsible for loading various system settings at NRE (Niagara Runtime Environment) boot time. This file now contains 2 keys pertaining to historical time zones:
 - **`niagara.timezone.dbCache`**: The maximum number of zones to remain cached in memory when querying the database for a particular zone. This caching is done in an "LRU" fashion, vs. hitting the database repeatedly for the same zones. This method provides performance gains.
 - **`niagara.timezone.eraTolerance`**: The number of milliseconds to wait before loading a new historical time zone era. The higher the number, the better the performance (yet lower the accuracy). The reverse is true for higher numbers

Please note that time zones in this database are not user editable.

Updating a historical time zone database

Typically, you use the normal release/build upgrade process for a JACE to update its historical time zone database. However, in cases where an updated version of this database becomes available, and you want to maintain the current build in an AX-3.8 or earlier JACE, you can do the following.

Perform the following steps:

Step 1 Transfer the newer `timezones.jar` file to the host's `!lib` folder.

Step 2 Restart the station on that host.

The updated time zone database becomes effective on station restart.

NOTE: Host platforms running pre-AX-3.3 builds use a different method and different file to manage time zone data, the `!lib/timezones.xml` file. For details, see the *NiagaraAX-3.x Platform Guide* (dated September 2008).

CHAPTER 7 PLATFORM TUNNELING

TOPICS COVERED IN THIS CHAPTER

Platform tunneling overview

Platform tunneling requirements

Supervisor configuration to support platform tunneling

Platform tunneling usage

Notes on platform tunneling

SSL considerations for platform tunneling

Support was added for a "tunneled" Workbench platform connection to a remote NiagaraAX platform starting in AX-3.5, similar to the Fox and HTTP (browser) tunnel mechanisms introduced prior to AX-3.5. This chapter provides details about how to configure and establish tunneled platform connections.

For details about Fox and HTTP tunneling, see the Engineering Notes article "*Fox Tunneling and HTTP Tunneling*".

The following main sections are included.

Platform tunneling overview

Platform tunneling lets you make a NiagaraAX platform connection to a remote JACE platform by "tunneling" through the station running on another NiagaraAX proxy host, typically the Supervisor for the target JACE. Once the tunnel connection is made, you can use all the same platform views, and perform the same platform tasks, as if you had a platform connection directly to the target JACE.

This can be useful in cases where only the Supervisor has an exposed IP address, or if a firewall restricts access to Niagara hosts on a network through only a single port.

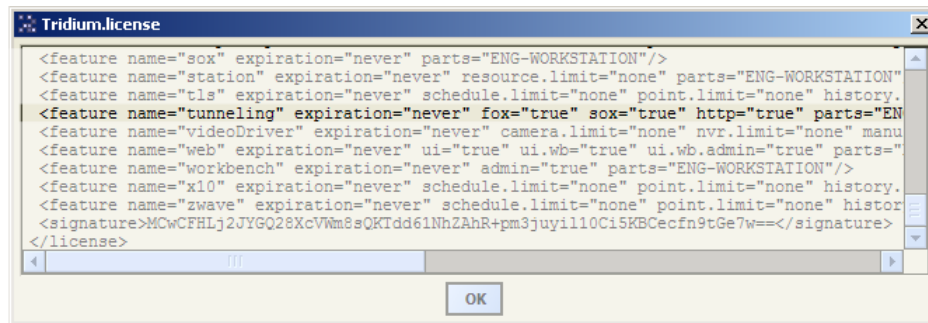
Key points to remember include:

- Usage applies to a full Workbench client only, and not a "Web Workbench" browser session for platform tunnel access (just as Web Workbench does not provide direct platform access).
- Tunneling is not a "daemon level" function. Platform tunneling relies on the Supervisor's running station, as its web server acts as the "tunnel proxy server", or tunnel entrance point. Tunneling is actually HTTP, to allow access to the platform daemon running on a JACE—the tunnel endpoint.

Platform tunneling requirements

- AX-3.5 or later is required by all platforms (Supervisor, JACEs) on ends of a platform tunnel.
- Platform tunneling requires a "tunneling" feature in the license of the Supervisor, (tunnel proxy server) with the "http" attribute set to true, which is standard, as shown here.

Figure 148. Supervisor (tunnel proxy server) requires "tunneling" feature to support platform tunneling



- The Supervisor station also requires a few configuration settings, as described in following sections.

Supervisor configuration to support platform tunneling

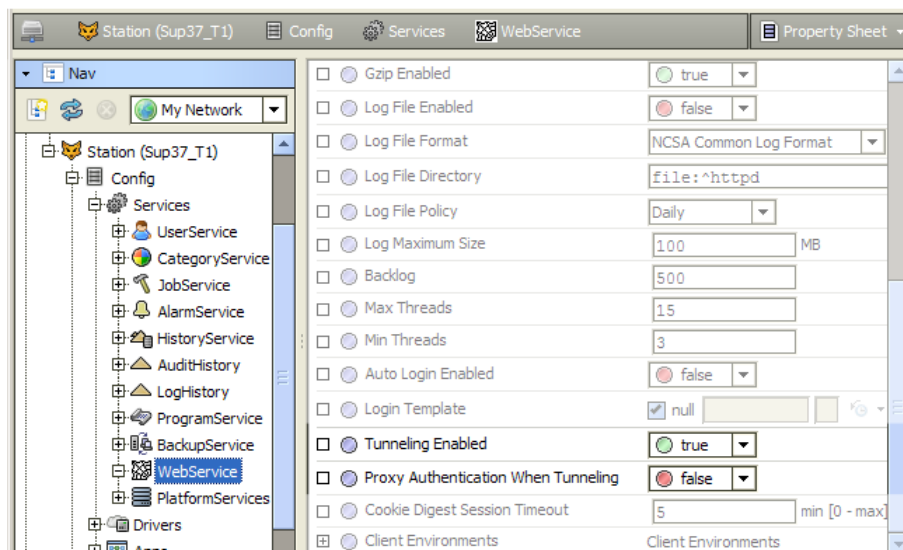
On the Supervisor station, these services are involved:

- WebService settings for platform tunneling
- FoxService settings for platform tunneling

WebService settings

In the Supervisor's WebService (under **Config > Services**), ensure that the two properties are configured as shown.

Figure 149. WebService in Supervisor to support platform tunneling



- **Tunneling Enabled**

Set to true. If Fox and HTTP tunneling are already enabled, this should already be enabled.

- **Proxy Authentication When Tunneling**

Set to `false`. In the initial AX-3.5 implementation of platform tunneling, proxy authentication is not supported. The only login authentication used in a tunneled platform connection are the tunnel endpoint's (remote controller's) platform credentials.

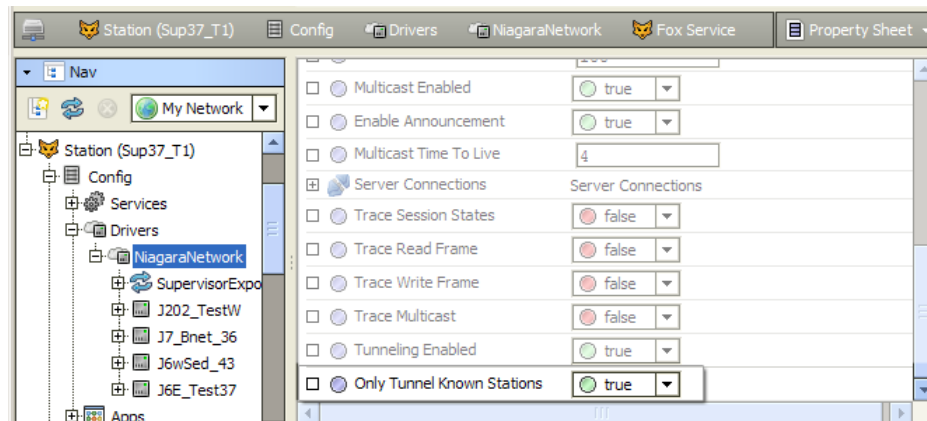
NOTE: This property was first introduced prior to AX-3.5, and for an existing job upgraded to AX-3.5 may be `true`. To permit platform tunneling, you must set it to `false`.

FoxService settings

Platform tunneling works independently from Fox tunneling. Typically, if a job uses tunneling, both Fox tunneling and HTTP tunneling are already enabled.

However, note that one FoxService property on the Supervisor (proxy tunnel server) affects platform tunneling. That property is shown here.

Figure 150.



On the Supervisor, expand its **Config > Drivers**, nodes to reveal its NiagaraNetwork, then right-click to select **Views > Property Sheet**. Expand the Fox Service container, and scroll near the bottom of its contained properties.

- **Tunneling Enabled**

Can be either `true` or `false`, it affects Fox tunneling only (not platform tunneling). If Fox tunneling is already enabled, this should already be `true`.

- **Only Tunnel Known Stations**

May be either `true` or `false`. Depending on its setting, this affects the ability to open a tunneled platform connection, including how you enter the target host in the **Open Platform** dialog:

- If `false`, you can tunnel a platform connection to any AX-3.5 or later JACE that is reachable from the Supervisor station, using the target IP address (or hostname).
- If `true`, you can tunnel a platform connection only to a JACE that is currently represented in the Supervisor's NiagaraNetwork. Here, you use its station name as the target destination—not the JACE's IP address or hostname.

Note that this property was originally for Fox tunneling, and works in the same manner. However, it is a bit more "intuitive" in Fox tunneling, where you equate Fox connections and stations.

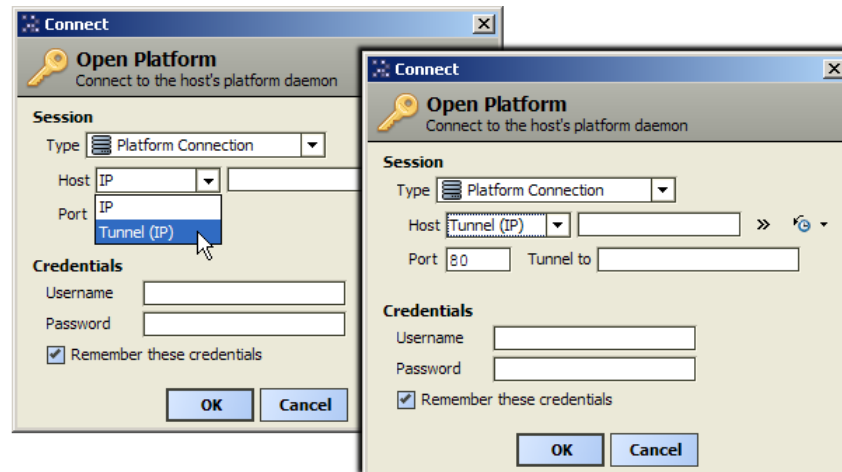
Platform tunneling usage

Using Workbench, open a tunneled platform connection through the Supervisor, whether remote or local at the Supervisor. If a standard connection is not already open, or you are working on the Supervisor itself, use the **Open Platform** dialog, available from the **File > Open** submenu.

Example: Opening Platform dialog if tunneling

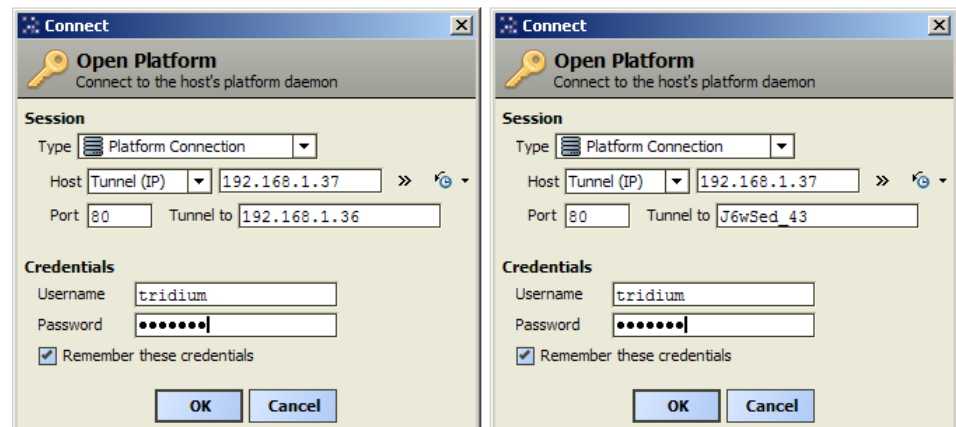
Perform the following steps:

- Step 1 In the **Open Platform** dialog, click the drop-down control in the **Host** field and select **Tunnel (IP)**, as shown here.



- Step 2 In the top **Host Tunnel (IP)** field, enter the IP address or hostname of the Supervisor (the tunnel proxy server).

In the example shown here, the Supervisor is at 192.168.1.37 IP address.



- Step 3 In the **Port** field, enter the configured Http Port property value in the Supervisor's WebService, where 80 is the standard (default) value.

- Step 4 In the **Tunnel to** field, enter either:

- IP address or hostname of the endpoint JACE platform, if "Only Tunnel Known Stations" is false in the FoxService of the NiagaraNetwork of the Supervisor (shown left in image above).
- Station name of the station associated with the endpoint JACE, if "Only Tunnel Known Stations" is true in the FoxService of the NiagaraNetwork of the Supervisor (shown right in image above).

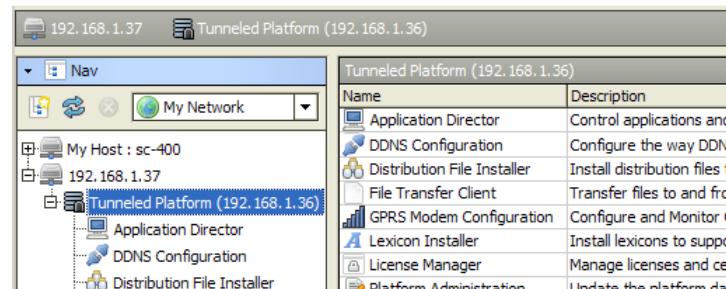
NOTE: If the endpoint JACE platform is using a "non-standard" HTTP port for the platform daemon, meaning not port 3011, append a colon (:) and that port number on the "Tunnel to" value. For example: 192.168.1.88:3012 (if the JACE is using port 3012 for the platform daemon)

- In the **Credentials** fields (Username and Password), enter the JACE's platform credentials.

Connected (via tunneling)

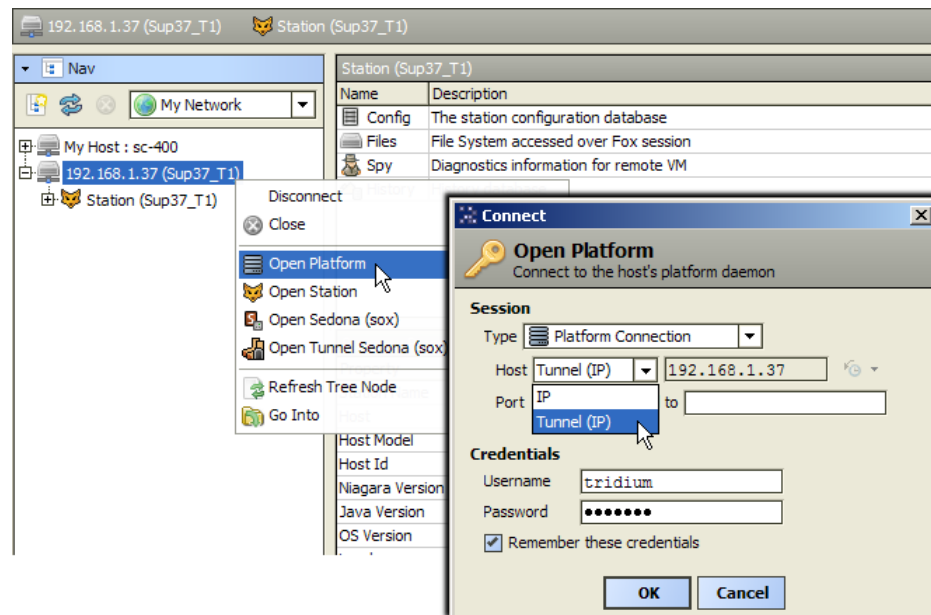
In Workbench, when connected via tunneling, the tunneled platform shows a different platform icon, along with either the target IP address or hostname (or station name if using "Only Tunnel Known Stations"), as shown here.

Figure 151. Tunnel-connected platform showing IP address of endpoint JACE



Once connected, you can perform all the same platform operations as if directly platform connected to the endpoint JACE, including running the Commissioning Wizard.

Note that if a Workbench connection to the remote Supervisor (either station or platform) is already open, you can right-click on the host in the Nav tree, and select Open Platform. Then choose "Tunnel (IP)" from the Host field drop-down control, as shown below.

Figure 152. Right-click remote Supervisor for Open Platform menu item

Enter values the same way as previously described (see Open Platform dialog if tunneling).

NOTE: Host menu choices "Open Tunnel Platform" and "Open Tunnel Station" were removed in Workbench 3.7 and later. Now you must specify the tunnel option in the Host field of the Connect dialog, as shown above.

Notes on platform tunneling

These additional notes apply to platform tunneling:

- If needed, you can have multiple tunneled platform connections through the Supervisor station. Traffic on each connection consumes some amount of resources, but typically capacity exists.
- Remember that tunneled platform connections are dependent on the running Supervisor station. Avoid any Supervisor station restarts (or stops) during critical tunneled platform operations, otherwise problems are likely to result.
- Although not recommended, currently there is nothing preventing a "loop back" platform connection to the Supervisor platform itself, using only the HTTP port as configured in its WebService. In this case the "Host" and "Tunnel to" fields would both have the same IP address.

This is a "workaround" solution if a firewall connection blocks all ports (for example, 3011) except for that one HTTP port.

SSL considerations for platform tunneling

Starting in AX-3.7, secure (SSL) connections to NiagaraAX hosts are possible for platform connections, station (fox) connections from Workbench or other stations, and web browser (http) connections to a station's WebService.

Currently, platform tunneling to a target host configured for platformssl only is not supported (Ssl Only from the **Change SSL Settings** function in its **Platform Administration** view.) To platform connect to such a host, you must make a direct platform connection from Workbench.

However, you can make a "regular" (unencrypted) tunneled platform connection to a target host that is merely "enabled" for platformssl. Additionally, the Supervisor (tunnel server) may be configured for SSL, say for all connection types (platform, fox, webserver)—and this may be more common than not.

For complete details on SSL configuration in AX-3.7 and later, refer to the *NiagaraAX SSL Connectivity Guide*.

