

# Technical Document

## NiagaraAX Platform Guide

AX-3.8 and AX-3.7u1

December 18, 2013



# Niagara<sup>AX</sup> Platform Guide

Copyright © 2013 Tridium, Inc.

All rights reserved.

3951 Westerre Pkwy., Suite 350

Richmond

Virginia

23233

U.S.A.

## Confidentiality Notice

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, Niagara<sup>AX</sup> Framework, and Sedona Framework are registered trademarks, and Workbench, WorkPlace<sup>AX</sup>, and <sup>AX</sup>Supervisor, 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 © 2013 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

<b>Preface .....</b>	<b>ix</b>
<b>Document Change Log .....</b>	<b>ix</b>
<b>Niagara platform.....</b>	<b>1-1</b>
<b>Platform overview .....</b>	<b>1-2</b>
About a platform connection .....	1-3
Platform connection session info .....	1-3
Platform daemon (niagarad) .....	1-4
Platform daemon on a PC .....	1-4
Provisioning versus platform interface .....	1-5
Types of platform views .....	1-5
<b>About platform differences .....</b>	<b>1-6</b>
QNX-based .....	1-6
Sun Hotspot JVM or IBM J9 JVM .....	1-7
Backup Battery (or not) .....	1-7
Platform view differences, QNX-based vs. Windows-based .....	1-8
Platform Administration .....	1-8
Windows-based .....	1-9
Platform Administration .....	1-9
Win64-based Supervisor notes .....	1-10
Linux-based Supervisor .....	1-10
NiagaraAX platform rights on Linux Supervisor .....	1-10
Default Linux Supervisor platform administrator .....	1-11
Linux Supervisor platform views .....	1-11
Linux Supervisor port usage notes .....	1-11
Models of platforms .....	1-11
<b>Application Director .....</b>	<b>1-13</b>
Installed applications (stations) .....	1-14
Application selections .....	1-14
Application output .....	1-15
Standard output overview .....	1-15
Station log levels (spy/logSetup) .....	1-16
Station LogHistory (LogHistoryService) .....	1-16
Application and output controls .....	1-17
Start checkboxes .....	1-17
Application control buttons .....	1-18
Output control buttons .....	1-18
Output Settings .....	1-19
<b>Certificate Management .....</b>	<b>1-19</b>
Key Store .....	1-20
Trust Store .....	1-20
Allowed Hosts .....	1-20
<b>DDNS Configuration .....</b>	<b>1-21</b>
DDNS core configuration items .....	1-21
Provider .....	1-21
Mode .....	1-22
Adapter .....	1-22
About TZO .....	1-22
<b>Distribution File Installer .....</b>	<b>1-22</b>

Security update 1 changes to backup dist usage .....	1-23
AX-3.8 changes to backup dist usage .....	1-23
Operation of the Distribution File Installer .....	1-23
<i>Dist file selection</i> .....	1-24
<i>Distribution file install process</i> .....	1-25
Restoring a backup dist .....	1-25
Downgrading a JACE (Clean Dist) .....	1-26
<b>Upgrading a JACE .....</b>	<b>1-29</b>
<b>File Transfer Client .....</b>	<b>1-30</b>
system.properties notes .....	1-31
<i>Station restart needed</i> .....	1-31
<b>GPRS Modem Configuration .....</b>	<b>1-32</b>
GPRS modem configuration sections .....	1-32
<i>Modem Configuration</i> .....	1-32
<i>Provider Configuration</i> .....	1-33
<i>SMS Configuration</i> .....	1-33
Status and Runtime Data area .....	1-33
<i>GPRS Status</i> .....	1-33
<i>GPRS Runtime Data</i> .....	1-34
<b>Lexicon Installer .....</b>	<b>1-34</b>
<b>License Manager .....</b>	<b>1-35</b>
License operations .....	1-37
<i>Import</i> .....	1-37
<i>Export</i> .....	1-37
<i>License Import results</i> .....	1-38
About the licensing server .....	1-39
<b>Platform Administration .....</b>	<b>1-40</b>
Types of Platform Administration functions .....	1-41
View Details .....	1-41
Update Authentication .....	1-42
<i>Digest platform authentication</i> .....	1-42
<i>Improvements to AX-3.8 digest authentication</i> .....	1-43
<i>Basic platform authentication</i> .....	1-44
Change HTTP Port .....	1-46
Change SSL Settings .....	1-47
<i>Save notes on SSL platform settings</i> .....	1-48
Change Date/Time .....	1-48
<i>Use Local</i> .....	1-49
FTP/Telnet .....	1-49
Change Output Settings .....	1-49
<i>Log filter settings</i> .....	1-50
View Daemon Output .....	1-50
Set Module Filter .....	1-52
<i>Operations from a change in module content level</i> .....	1-52
Backup .....	1-53
Commissioning .....	1-54
Reboot .....	1-55
<b>Software Manager .....</b>	<b>1-55</b>
Software Manager notes .....	1-56
About your software database .....	1-56
Default module listing and layout .....	1-57
<i>Software Manager table columns</i> .....	1-58
<i>Software Details</i> .....	1-59
Filtering displayed software .....	1-59
<i>Filter by status</i> .....	1-60
<i>Filter by name</i> .....	1-60
Software Import .....	1-60
<i>Import vs. copy into modules</i> .....	1-60
Software actions .....	1-61
<i>Upgrade All Out of Date</i> .....	1-61
<i>Install</i> .....	1-61

Uninstall .....	1-62
Re-Install, Upgrade, Downgrade .....	1-62
Commit and Reset .....	1-62
Right-click option to install earlier version .....	1-62
<b>Station Copier .....</b>	<b>1-63</b>
Security update 1 changes to Station Copier usage .....	1-63
When config.bog edits are not needed .....	1-64
When config.bog edits are needed .....	1-64
Station copy direction .....	1-65
Station Copier dependencies check .....	1-65
Station Transfer Wizard .....	1-66
Name step .....	1-66
Delete step .....	1-66
Content step .....	1-67
Disposition step .....	1-67
Station settings step .....	1-67
Details step .....	1-68
Modules step .....	1-68
Stop station step .....	1-69
Review step .....	1-70
Transferring station .....	1-70
Renaming stations .....	1-71
Deleting stations .....	1-71
<b>TCP/IP Configuration .....</b>	<b>1-72</b>
TCP/IP changes in AX-3.6 .....	1-73
IPv6 support .....	1-73
JACE WiFi .....	1-73
TCP/IP Host fields .....	1-73
TCP/IP DNS fields .....	1-74
TCP/IP Interface fields .....	1-74
IPv4 Settings .....	1-75
IPv6 Settings .....	1-76
<b>User Manager .....</b>	<b>1-77</b>
Users management .....	1-77
Review user .....	1-78
Add user .....	1-78
Delete user .....	1-78
Change user password .....	1-78
Drag and drop .....	1-79
Groups management .....	1-79
Review group .....	1-79
Add group .....	1-79
Delete group .....	1-80
Add member .....	1-80
Remove member .....	1-80
<b>WiFi Configuration .....</b>	<b>1-81</b>
<b>WiFi Certificate Manager .....</b>	<b>1-82</b>
<b>Remote File System .....</b>	<b>1-82</b>

## Platform Services ..... 2-1

<b>About Platform Services .....</b>	<b>2-2</b>
Component differences for platform services .....	2-2
<b>PlatformServiceContainer parameters .....</b>	<b>2-3</b>
PlatformServiceContainer status values .....	2-3
PlatformServiceContainer configuration parameters .....	2-4
Additional PlatformServiceContainer properties .....	2-6
Model-specific PlatformServiceContainer properties .....	2-7
PlatformServiceContainer actions .....	2-7
<b>SystemService (under PlatformServices) .....</b>	<b>2-8</b>
<b>Platform service types .....</b>	<b>2-9</b>

<b>Using platform services in a station</b> .....	<b>2-10</b>
JACE power monitoring .....	2-10
<i>Battery monitoring disabled</i> .....	2-10
JACE-NXT (and JACE-NXS) power monitoring .....	2-11
JACE-NXT hardware monitoring .....	2-11
JACE-NXS hardware monitoring .....	2-11
JACE-NX hardware monitoring .....	2-11
PlatformServices binding and link caveats .....	2-12
<b>About the NtpPlatformService</b> .....	<b>2-13</b>
Interaction with station's TimeSyncService .....	2-13
NTP port/firewall considerations .....	2-13
About the Ntp Platform Service Editor .....	2-13
About the Ntp Platform Service Editor Win32 .....	2-14
<i>About Ntp Platform Service Editor Win32 settings</i> .....	2-14
<i>About Ntp Platform Service Editor Win32 time servers</i> .....	2-15
About the Ntp Platform Service Editor Qnx .....	2-15
<i>About Ntp Platform Service Editor Qnx settings</i> .....	2-15
<i>About Ntp Platform Service Editor Qnx time servers</i> .....	2-16
<i>Sync Now action</i> .....	2-16
About the Ntp Platform Service Editor Linux .....	2-17
<i>About Ntp Platform Service Editor Linux settings</i> .....	2-17
<i>About Ntp Platform Service Editor Linux time servers</i> .....	2-18

## Platform Component Guides ..... 3-1

<b>Component Reference Summary</b> .....	<b>3-1</b>
<b>Components in platCrypto</b> .....	<b>3-1</b>
platCrypto-CertManagerService .....	3-1
platform-DaemonSecureSession .....	3-1
<b>Components in platDataRecovery module</b> .....	<b>3-2</b>
platDataRecovery-DataRecoveryService .....	3-2
<b>Components in platGprs module</b> .....	<b>3-2</b>
platGprs-GprsPlatformService .....	3-2
platGprs-GprsHostSettings .....	3-2
platGprs-GprsRuntimeData .....	3-2
<b>Components in platform module</b> .....	<b>3-3</b>
platform-DefaultDaemonFileSpace .....	3-3
platform-DaemonSession .....	3-3
platform-LicenseDatabaseTool .....	3-3
platform-LicensePlatformService .....	3-3
platform-NtpPlatformServiceLinux .....	3-3
platform-NtpPlatformServiceQnx .....	3-3
platform-NtpPlatformServiceWin32 .....	3-4
platform-PlatformAlarmSupport .....	3-4
platform-PlatformServiceContainer .....	3-4
platform-SystemPlatformServiceQnxJavelina .....	3-4
platform-SystemPlatformServiceQnxNpm2xx .....	3-4
platform-SystemPlatformServiceQnxNpm6xx .....	3-4
platform-SystemPlatformServiceWin32 .....	3-4
<i>Reboot action</i> .....	3-4
platform-TcplpPlatformService .....	3-5
<b>Components in platHwScan</b> .....	<b>3-5</b>
platHwScan-HardwareScanService .....	3-5
<b>Components in platIEEE8021X</b> .....	<b>3-5</b>
platIEEE8021X-IEEE8021XAdapterSettings .....	3-5
platIEEE8021X-IEEE8021XHostSettings .....	3-5
platIEEE8021X-IEEE8021XPlatformService .....	3-5
<b>Components in platPower module</b> .....	<b>3-6</b>
platPower-ExternalSlaBattery .....	3-6
platPower-JaceSlaBattery .....	3-6

platPower-JavelinaBatteryPlatformService .....	3-6
platPower-NimhBattery .....	3-6
platPower-Npm2NimhBattery .....	3-6
platPower-NpmDualBatteryPlatformService .....	3-7
platPower-NpmExternalSlaBattery .....	3-7
platPower-PowerMonitorPlatformServiceQnx .....	3-7
<b>Components in platPowerNxs module .....</b>	<b>3-7</b>
platPowerNxs-PowerMonitorPlatformServiceNxsWin32 .....	3-7
<b>Components in platSerialQnx module .....</b>	<b>3-7</b>
platSerialQnx-SerialPortPlatformServiceQnx .....	3-7
platSerialQnx-SerialPortPlatformServiceQnx403 .....	3-7
platSerialQnx-SerialPortPlatformServiceQnx404 .....	3-8
platSerialQnx-SerialPortQnx .....	3-8
<b>Components in platSerialWin32 module .....</b>	<b>3-8</b>
platSerialWin32-SerialPortPlatformServiceWin32 .....	3-8
platSerialWin32-SerialPortWin32 .....	3-8
<b>Components in platSerialWin64 module .....</b>	<b>3-8</b>
platSerialWin64-SerialPortPlatformServiceWin64 .....	3-8
platSerialWin64-SerialPortWin64 .....	3-8
<b>Components in platSysmonNx module .....</b>	<b>3-8</b>
platSysmonNx-HardwareMonitorNxPlatformServiceWin32 .....	3-8
<b>Components in platSysmonNxs module .....</b>	<b>3-9</b>
platSysmonNxs-HardwareMonitorNxsPlatformServiceWin32 .....	3-9
<b>Components in platSysmonNxt module .....</b>	<b>3-9</b>
platSysmonNxt-HardwareMonitorNxtPlatformServiceWin32 .....	3-9
<b>Components in platUsbmon module .....</b>	<b>3-9</b>
platUsbmon-UsbMonitorPlatformServiceQnx .....	3-9
<b>Components in platWifi module .....</b>	<b>3-9</b>
platWifi-WifiPlatformService .....	3-9

## Platform Plugin Guides ..... 4-1

<b>Plugin Reference Summary .....</b>	<b>4-1</b>
<b>Plugins in platCrypto .....</b>	<b>4-1</b>
platCrypto-CertManagerView .....	4-1
<b>Plugins in platDaemon module .....</b>	<b>4-1</b>
platDaemon-ApplicationDirector .....	4-2
platDaemon-DistInstaller .....	4-2
platDaemon-DistributionView .....	4-2
platDaemon-FileTransferClient .....	4-2
platDaemon-LexiconInstaller .....	4-2
platDaemon-LicenseManager .....	4-2
platDaemon-PlatformSessionListView .....	4-2
platDaemon-SoftwareManager .....	4-3
platDaemon-SoftwareView .....	4-3
platDaemon-PlatformAdministration .....	4-3
platDaemon-R2PlatformTool .....	4-3
platDaemon-StationCopier .....	4-3
platDaemon-StationTextSummaryEditor .....	4-3
platDaemon-TcplpConfiguration .....	4-3
platDaemon-UserManager .....	4-3
<b>Plugin in platDataRecovery .....</b>	<b>4-4</b>
platDataRecovery-DataRecoveryServiceEditor .....	4-4
<b>Plugin in platDDNS .....</b>	<b>4-4</b>
platDdns-DdnsConfigurationView .....	4-4
<b>Plugins in platform module .....</b>	<b>4-4</b>
platform-LicensePlatformServicePlugin .....	4-4

platform-NtpPlatformServiceEditorLinux .....	4-4
platform-NtpPlatformServiceEditorQnx .....	4-4
platform-NtpPlatformServiceEditorWin32 .....	4-5
platform-PlatformServiceContainerPlugin .....	4-5
platform-PlatformServiceProperties .....	4-5
platform-SystemDateTimeEditor .....	4-5
platform-SystemPlatformServicePlugin .....	4-5
platform-SystemPlatformServiceQnxPlugin .....	4-5
platform-TcplpPlatformServicePlugin .....	4-5
platform-WorkbenchLicenseManager .....	4-5
<b>Plugins in platGprs .....</b>	<b>4-5</b>
platGprs-GprsConfiguration .....	4-5
platGprs-GprsPlatformServicePlugin .....	4-5
<b>Plugins in platHwScan .....</b>	<b>4-6</b>
platHwScan-HardwareScanServiceView .....	4-6
<b>Plugins in platIEEE8021X .....</b>	<b>4-6</b>
platIEEE8021X-IEEE8021XAdapterSettingsEditor .....	4-6
platIEEE8021X-IEEE8021XDaemonSessionPlugin .....	4-6
platIEEE8021X-IEEE8021XPlatformServicePlugin .....	4-6
<b>Plugins in platPower .....</b>	<b>4-6</b>
platPower-JavelinaBatteryPlatformServicePlugin .....	4-6
platPower-PowerMonitorPlatformServicePlugin .....	4-6
<b>Plugins in platWifi module .....</b>	<b>4-7</b>
platWifi-WifiConfiguration .....	4-7
platWifi-WifiPlatformServicePlugin .....	4-7
platWifi-WifiSecurityManager .....	4-7

## License Tools and Files.....A-1

<b>Workbench License Manager .....</b>	<b>A-2</b>
Import File .....	A-2
Export File .....	A-3
Delete .....	A-4
Sync Online .....	A-4
<b>Request License .....</b>	<b>A-5</b>
<b>About the local license database .....</b>	<b>A-6</b>
Local license database rationale .....	A-6
Local license inbox .....	A-6
<b>About license archive (.lar) files .....</b>	<b>A-7</b>
Licensing server use of license archives .....	A-7
<b>About NiagaraAX license files .....</b>	<b>A-8</b>
Items common to all license files .....	A-8
<i>license</i> .....	A-8
<i>about</i> .....	A-8
<i>brand</i> .....	A-8
<i>signature</i> .....	A-9
JACE hardware features .....	A-9
<i>dataRecovery</i> .....	A-9
<i>ibmj9j2me</i> .....	A-9
<i>maxheap</i> .....	A-9
<i>mstp</i> .....	A-9
<i>ndio</i> .....	A-10
<i>nrrio</i> .....	A-10
<i>serial</i> .....	A-10
<i>sunj2se</i> .....	A-10
Driver attributes .....	A-10
<i>name</i> .....	A-10
<i>expiration</i> .....	A-10
<i>device.limit</i> .....	A-11
<i>history.limit</i> .....	A-11



point.limit .....	A-11
schedule.limit .....	A-11
parts .....	A-11
Driver types .....	A-11
aaphp .....	A-11
aapup .....	A-11
bacnet .....	A-11
bacnetAws .....	A-11
bacnetOws .....	A-12
bacnetws .....	A-12
dust .....	A-12
fileDriver .....	A-12
jen6lp .....	A-12
jennic .....	A-12
lonworks .....	A-12
modbusAsync .....	A-12
modbusCore .....	A-12
modbusSlave .....	A-12
modbusTcp .....	A-12
modbusTcpSlave .....	A-12
obixDriver .....	A-13
opc .....	A-13
niagaraDriver .....	A-13
rdbDb2 .....	A-13
rdbOracle .....	A-13
rdbSqlServer .....	A-13
sedonanet .....	A-13
snmp .....	A-13
videoDriver .....	A-13
zwave .....	A-14
Applications .....	A-14
station .....	A-14
web .....	A-14
workbench .....	A-14
box .....	A-14
crypto .....	A-14
eas .....	A-15
email .....	A-15
fips140-2 .....	A-15
genericAppliance .....	A-15
ieee8021x .....	A-15
ldapv3 .....	A-15
mobile .....	A-16
provisioning .....	A-16
sedonaProvisioning .....	A-16
sox .....	A-16
tenantBilling .....	A-16

Time Zones and NiagaraAX .....

B-1

Time zones and terminology .....	B-1
UTC .....	B-1
DST .....	B-1
Selecting a time zone in NiagaraAX .....	B-2
About the historical time zone database .....	B-2
Issue example and fix .....	B-2
Parts of the historical time zone database .....	B-3
Updating a historical time zone database .....	B-3
About timezones.xml .....	B-4
DST start and end syntax variations .....	B-4
Using Workbench to edit and transfer timezones.xml .....	B-4

Platform Tunneling.....

C-1

Platform tunneling overview .....	C-1
-----------------------------------	-----

Platform tunneling requirements .....	C-1
<b>Supervisor configuration to support platform tunneling .....</b>	<b>C-2</b>
WebService settings for platform tunneling .....	C-2
FoxService settings for platform tunneling .....	C-2
<b>Platform tunneling usage .....</b>	<b>C-3</b>
Open Platform dialog if tunneling .....	C-3
Connected in Workbench .....	C-4
<b>Notes on platform tunneling .....</b>	<b>C-5</b>
<b>SSL considerations for platform tunneling .....</b>	<b>C-5</b>

# CONTENTS

## Preface

---

- [Document Change Log](#)

## Document Change Log

Updates (changes/additions) to this *NiagaraAX Platform Guide* document are listed as follows.

- Minor changes, December 18, 2013  
Changes were few, and include the following:
  - The “[Platform Administration](#)” section “[Change SSL Settings](#)” on page 1-47 was revised to describe AX-3.8 and AX-3.7 differences when “State” is set to “Ssl Only”. The “Protocol” description was also revised to mention selection differences for AX-3.8 platforms that are also configured for FIPS-140.
  - In the section “[Platform Services](#)”, the subsection “[PlatformServiceContainer status values](#)” on page 2-3 was updated to include the “Local Daemon Ssl Port” property.
- NiagaraAX-3.8 update, November 5, 2013  
Various document updates for the initial release of NiagaraAX-3.8 (in this document denoted as “AX-3.8”), where this document also applies to the prior 2013 “update 1” release of AX-3.7 (denoted as “AX-3.7u1”). Changes related to AX-3.8 in this document are as follows:
  - The beginning “[Platform overview](#)” section was edited to mention a possible “IEEE 802.1X Configuration” view for some AX-3.8 QNX-based JACE hosts. The “[About platform differences](#)” subsection “[Windows-based](#)” on page 1-9 was edited to mention AX-3.8 support for Windows 8 Professional and Windows Server 2012.
  - The “[Distribution File Installer](#)” section was revised to cover restoring station backup files saved on hosts using the AX-3.8, including a new subsection “[AX-3.8 changes to backup dist usage](#)” on page 1-23. Similar mention was made about AX-3.8 changes in other subsections, for example “[Restoring a backup dist](#)” on page 1-25.
  - The section “[Station Copier](#)” on page 1-63 was revised to note that in AX-3.8, edits to `config.bog` files are unnecessary (it also references the standalone document *NiagaraAX 2013 Security Updates*).
  - The “[Platform Administration](#)” section “[Update Authentication](#)” on page 1-42 was revised to describe some AX-3.8 changes, including a new subsection “[Improvements to AX-3.8 digest authentication](#)” on page 1-43.
  - The “[Platform Component Guides](#)” section has a new section “[Components in platIEEE8021X](#)” on page 3-5, with summary entries for components in the AX-3.8 `platIEEE8021X` module.
  - The “[Platform Plugin Guides](#)” section has a new section “[Plugins in platIEEE8021X](#)” on page 4-6, with summary entries for plugins (views) in the AX-3.8 `platIEEE8021X` module.
  - In the “[License Tools and Files](#)” appendix, the subsection “[About NiagaraAX license files](#)” on page A-8 had several new AX-3.8-related entries in the section “[Applications](#)” on page A-14. The section “[Request License](#)” on page A-5 shows the current license request form.
- NiagaraAX-3.7 “Update 1” revision, May 30, 2013  
Various document updates concurrent with the “update 1” release for AX-3.7 (in this document denoted as “AX-3.7u1”). Station security-related improvements are included. The document now mentions a new JACE-3E controller model introduced in 2013, mainly affecting sections “[About platform differences](#)” on page 1-6, including subtopics “[QNX-based](#)”, “[Sun Hotspot JVM or IBM J9 JVM](#)”, “[Backup Battery \(or not\)](#)”, and also “[Models of platforms](#)” on page 1-11. Additionally, the prior release of the Sedona Framework TXS 1.2 includes a possible new **Sedona Environment Manager** platform view for remote NiagaraAX hosts, as explained in a [Note](#): in the section “[Plat-](#)

[form overview](#)” on page 1-2 (and referenced in a few other topics). Note that this platform view, along with the reasons for it, are covered in other documents specific to Sedona Framework TXS 1.2. Other changes related to the AX-3.7u1 release in this document are as follows:

- The [“Distribution File Installer”](#) section has new Notes added about restoring backup dist files saved when using a 2013 update release (AX-3.7u1), or later. See a new related subsection [“Security update 1 changes to backup dist usage”](#) on page 1-23. In the subsection [“Downgrading a JACE \(Clean Dist\)”](#) on page 1-26, a Note was added about the deletion of the `!security` folder of a JACE, and how this affects re-installs of saved station (`config.bog`) files. A subsection [“Example JACE downgrade from AX-3.7u1 to AX-3.6u4, maintaining interim station data”](#) on page 1-28 was edited to reflect working with 2013 update (or later) releases. In the section [“Upgrading a JACE”](#) on page 1-29, a **Note**: was added in about multi-station job upgrades.
- The [“Station Copier”](#) section has new Notes added about installing saved stations (`config.bog`) files when using a 2013 update release (AX-3.7u1) or later. A related subsection [“Security update 1 changes to Station Copier usage”](#) on page 1-63 summarizes cases when `config.bog` edits may be needed (it also references the standalone document *NiagaraAX 2013 Security Updates*). The subsection [“Station copy direction”](#) on page 1-65 also has a related **Note**: on this, as a reminder.
- NiagaraAX-3.7 revision, August 30, 2012  
Document completely reworked since the prior version describing AX-3.5 and AX-3.6, where most mention of NiagaraAX release behavior prior to AX-3.6 has been removed. Most screen captures throughout the document were updated using AX-3.7 Workbench. Changed areas are numerous, with the largest areas of change (since the last AX-3.6 revision) found in the following areas:  
In the main [“Niagara platform”](#) section, it was noted that dialup modem support ended in AX-3.7, with the [“Dialup Configuration”](#) platform view missing. Other changes in child sections are
  - [“Platform overview”](#) on page 1-2, where the [“Certificate Management”](#) platform view is mentioned as a possible view.
    - [“About a platform connection”](#) on page 1-3 was reworked with additional information, including new subsections [“Platform connection session info”](#) on page 1-3 and [“Platform daemon \(niagarad\)”](#) on page 1-4.
    - [“Types of platform views”](#) on page 1-5 now lists the Certificate Management view and omits the Dialup Configuration view.
  - [“About platform differences”](#) on page 1-6 was updated with information on the newest JACE platforms, including capabilities introduced starting in AX-3.7, in the following subsections:
    - [“QNX-based”](#) on page 1-6, where new differences are noted between [Sun Hotspot JVM](#) or [IBM J9 JVM](#) type controllers. The [“Dialup Configuration”](#) subsection was removed.
    - [“Windows-based”](#) on page 1-9, where more information on supported Windows OS types was included, and more details in subsection [“Win64-based Supervisor notes”](#) on page 1-10.
    - [“Models of platforms”](#) on page 1-11 was updated to include the latest QNX-based JACEs.
  - The [“Application Director”](#) view section was updated, including details in the subsection [“Installed applications \(stations\)”](#) on page 1-14.
  - A new section, [“Certificate Management”](#) on page 1-19 was added, providing overview information only. Refer to the *NiagaraAX SSL Connectivity Guide* for detailed information.
  - The [“Distribution File Installer”](#) view section was updated, including new notes relating to SSL files in the section [“Downgrading a JACE \(Clean Dist\)”](#) on page 1-26. A new procedure is now included: [“Example JACE downgrade from AX-3.7u1 to AX-3.6u4, maintaining interim station data”](#) on page 1-28.
  - The [“File Transfer Client”](#) view section was updated with a new subsection [“system.properties notes”](#). This section describes some new entries, along with notes about updating this file in a remote JACE controller.
  - The [“Lexicon Installer”](#) view section was updated to note the alternative method of using lexicon *modules*, available starting in AX-3.7.
  - The [“Platform Administration”](#) view section was updated to note new functions starting in AX-3.7, in subsection [“Types of Platform Administration functions”](#) on page 1-41 and new subsection [“Change SSL Settings”](#) on page 1-47.
  - The [“Software Manager”](#) view section was updated to note lexicon modules starting AX-3.7, in subsection [“Software Manager notes”](#) on page 1-56.
  - The section [“WiFi Certificate Manager”](#) on page 1-82 was updated to show the changed interface (view) for managing PKI certificates for a WiFi option installed in a JACE-700 controller.

In the section [“Platform Services”](#), the subsection [“SystemService \(under PlatformServices\)”](#) on page 2-8 was updated to note the [“Local Daemon Ssl Port”](#) property starting in AX-3.7. In the [“Platform](#)

[Component Guides](#)” section, new summary entries were added for context-sensitive Workbench help on components new in AX-3.7, including [“Components in platCrypto”](#) on page 3-1 and the [“platform-DaemonSecureSession”](#) on page 3-1. Other corrections were also made. In the [“Platform Plugin Guides”](#) (views) section, new summary entries were added for context-sensitive Workbench help on views new in AX-3.7, including [“Plugins in platCrypto”](#) on page 4-1. Other corrections were also made. In the [“License Tools and Files”](#) appendix, the subsection [“About NiagaraAX license files”](#) on page A-8 was updated to make it more current. Note it is still far from inclusive of all possible licensed features and/or variations. In the [“Platform Tunneling”](#) appendix, a new subsection [“SSL considerations for platform tunneling”](#) on page C-5 was added.



# CHAPTER 1

## Niagara platform

---

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.

The following main sections provide more platform details:

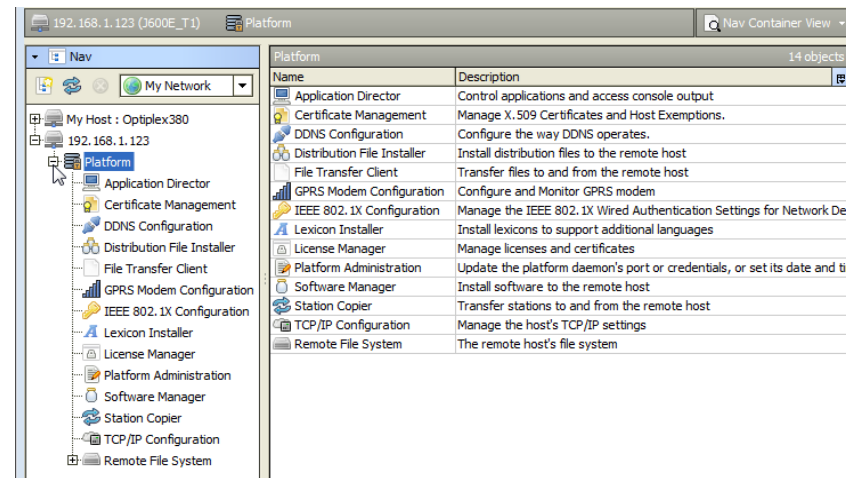
- [“Platform overview”](#) on page 1-2
  - [“About a platform connection”](#) on page 1-3
  - [“Provisioning versus platform interface”](#) on page 1-5
  - [“Types of platform views”](#) on page 1-5
- [“About platform differences”](#) on page 1-6
  - [“QNX-based”](#) on page 1-6
  - [“Windows-based”](#) on page 1-9
  - [“Linux-based Supervisor”](#) on page 1-10
  - [“Models of platforms”](#) on page 1-11
- Possible platform views:
  - [“Application Director”](#) on page 1-13
  - [“Certificate Management”](#) on page 1-19 (AX-3.7 or later platforms, if licensed for SSL)
  - [“DDNS Configuration”](#) on page 1-21 (If a QNX-based JACE)
  - [“Distribution File Installer”](#) on page 1-22
  - [“File Transfer Client”](#) on page 1-30
  - [“GPRS Modem Configuration”](#) on page 1-32 (If a QNX-based JACE)
  - [“Lexicon Installer”](#) on page 1-34
  - [“License Manager”](#) on page 1-35
  - [“Platform Administration”](#) on page 1-40
  - [“Software Manager”](#) on page 1-55
  - [“Station Copier”](#) on page 1-63
  - [“TCP/IP Configuration”](#) on page 1-72
  - [“User Manager”](#) on page 1-77 (If a Windows-based platform)
  - [“Remote File System”](#) on page 1-82

**Note:** Starting in AX-3.7, dialup modem support ended (no longer any “Dialup Configuration” platform view). Also, in some cases a “Sedona Environment Manager” platform view may appear—see the related [Note](#) in the section [“Platform overview”](#) on page 1-2.

## Platform overview

In Workbench, when you open a platform connection to a Niagara host (whether JACE or Supervisor), that host's available platform functions are listed in the platform's Nav Container View, see [Figure 1-1](#).

**Figure 1-1** Platform functions listed in platform's 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 JACE or a Supervisor, with these exceptions:

- If you open a *local* platform connection at your computer, note that some platform views are “missing,” e.g. the [Distribution File Installer](#), [File Transfer Client](#), and [Station Copier](#). The views have no real 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 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 for any 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 `platformIEEE8021X` module installed. (This is another option unavailable for any older QNX-based JACE platform with IBM J9 Java VM.)
- 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 Workbench AX-3.6 (build 3.6.44 or later), platform support was added for Niagara R2 configuration and upkeep of “retrofit board” R2 JACE-403 and JACE-545 controllers, also known as “JACE-603” and “JACE-645” controllers. A related “**R2 Platform Tool**” platform view is available only in a platform connection to these two models (if configured for Niagara R2). Details are not in this document. Refer to the *Retrofit Board Niagara R2 Install & Startup Guide* for information.

Also, a few of the platform views differ depending on platform type. See “[About platform differences](#)” on page 1-6 for details.

The following sections provide additional background on NiagaraAX 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 in Workbench), note any platform connection to a remote NiagaraAX 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** tool 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 that 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 (i.e. “Web Workbench” applet).

**Note:** *Browser access of a Supervisor station can provide platform connectivity, albeit indirectly, through its **ProvisioningService**. See “[Provisioning versus platform interface](#)” on page 1-5.*



*Also, Workbench can “tunnel” a platform connection to a JACE through a station connection to a Supervisor—providing that the Supervisor host is licensed for web tunneling. However, note that full Workbench is still required. For details, see “[Platform Tunneling](#)” on page C-1.*

The following sections provide more details on a platform connection:

- [Platform connection session info](#)
- [Platform daemon \(niagarad\)](#)
  - [Platform daemon port](#)
  - [Platform credentials](#)
  - [Platform access without a platform connection](#)
- [Platform daemon on a PC](#)

## Platform connection session info

Starting in AX-3.7, it is now possible to open a *secure* (encrypted, SSL or TLS) platform connection to most<sup>1</sup> types of NiagaraAX hosts, providing each 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 (“platformssl”), or  for regular (unencrypted).

**Note:** *For best security, use of SSL is always recommended whenever possible. In AX-3.8 Workbench, default “Open Platform” and “Open Station” (Fox) dialogs assume an SSL type connection, where to connect in a regular (unencrypted) fashion you must change the connection “Type” first.*

Once platform connected, the available platform functions are identical—regardless of connection method. Workbench 3.7 or later provides a right-click “Session Info” action on any platform connection, as well as any station (Fox) connection.

**Figure 1-2** Example right-click Session Info dialog for a secure (SSL) platform connection

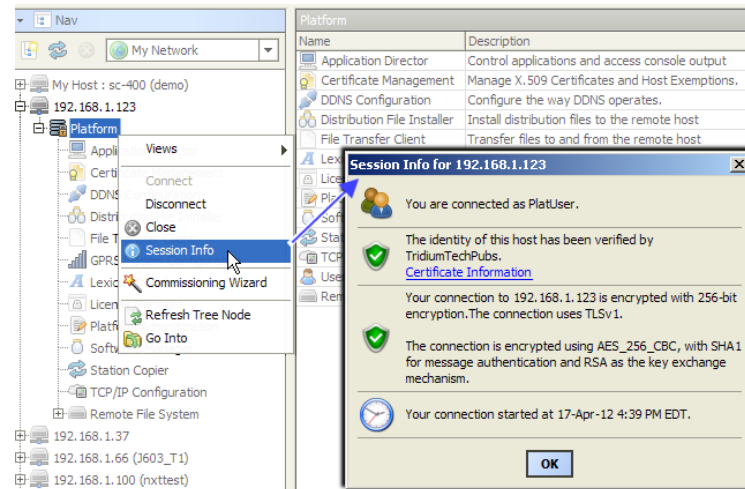
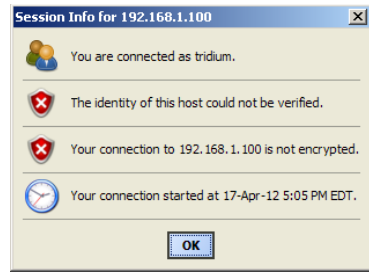


Figure 1-2 shows an example of this client session info from a secure (SSL) platform connection. In this example, the identity of the (server JACE) has been verified by a signed certificate, and all data on this connection is being encrypted.

If a regular (unencrypted) platform connection, the Session Info dialog would look similar to [Figure 1-3](#).

1. Platform and Fox SSL not supported on JACE models running the IBM J9 JVM (JACE-2, JACE-4, JACE-5 series).

**Figure 1-3** Example right-click Session Info dialog for an unencrypted platform connection



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.

Starting in AX-3.7, on most newer JACE platforms (JACE-6, JACE-7, any NPM6E-based or JACE-3E series) as well as any Windows-based platform), the platform 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.

The newer (Java-based) platform daemon facilitates the extended SSL support starting in AX-3.7, which remains *unavailable* in the older (“J9 JACE”) controllers. Note that apart from the additional SSL support (and in AX-3.8 support for IEEE 802.1X), the Java-based platform daemon functions the same as the previous “native code” platform daemon—thus this change should be largely “transparent”. For additional platform difference details, see “[About platform differences](#)” on page 1-6.

**Platform daemon port** A Niagara host’s platform daemon monitors a different TCP/IP port for client connections than does any running station. By *default*, this TCP port is either:

- 3011 - for a “regular” (un-encrypted)  **Platform** connection.
- 5011 - for a secure (SSL or TLS)  **Platform** connection (if available).

If necessary, you can change either TCP port monitored to a different (non-default) port during Niagara platform configuration. See “[Platform Administration](#)” on page 1-40.

**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 JACE controller ships with default platform credentials that are “widely known”—and if left unchanged the JACE is extremely susceptible to being hacked. During the startup 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](#)” on page 1-42.*

**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.” For details about this *different* type of platform access, see “[Platform Services](#)” on page 2-1.

### Platform daemon on a PC

When you install NiagaraAX 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 for either of the following:

- 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.
- For a PC to run as a SoftJACE, essentially a JACE running on a PC dedicated to this application. (Note that the SoftJACE Install wizard *automatically* installs and starts the platform daemon. Also,

in this particular case, *Workbench is not licensed* to run locally at that PC.)

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 NiagaraAX installation on your PC, you can install and start the platform daemon at any time, if needed. From the Windows Start menu, do this with **Start > All Programs > Niagara <3.n.n> > Install Platform Daemon** (shortcut for “plat.exe installdaemon”).*

In summary, your Workbench PC’s local platform daemon is not necessary for making client 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 NiagaraAX 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, please see “Niagara Provisioning overview” in the *NiagaraAX 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 NiagaraAX, 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

A Workbench direct (or tunneled) platform connection to a Niagara host, either JACE or Supervisor, provides various functional views, as shown in [Figure 1-1](#) on page 1-2.

**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 NiagaraAX commissioning of a new JACE controller, or when upgrading NiagaraAX in a JACE. For more details, see “About the Commissioning Wizard” in the JACE NiagaraAX Install & Startup Guide. Starting in build 3.6.44 of AX-3.6, Workbench platform support of “retrofit board”-equipped JACE-403 and JACE-545 controllers (JACE-603 and JACE-645) was added, allowing for Niagara R2 configuration. In this case, the platform “Commissioning Wizard” is not used. Instead, an “Install Niagara R2” wizard and new “R2 Platform Tool” platform view are available. Refer to the Retrofit Board Niagara R2 Install & Startup Guide for complete details.*

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

- [Application Director](#)  
To start, stop, restart, or kill a station on the Niagara 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](#)  
(AX-3.7 or later) This platform view appears if the host is licensed for SSL with the “crypto” feature, and has the necessary modules. Use it to import signed certificates into the platform’s key store and trust store (for SSL or TLS secure connections), and to perform other related functions.
- [DDNS Configuration](#)  
Allows for DNS IP addresses to be dynamically updated (DDNS), an option sometimes used for JACE hosts—although infrequently.
- [Distribution File Installer](#)  
Used to restore a “backup” .dist file to the target JACE, or to install a “clean dist” file to downgrade a JACE to a known bare minimum state. (Do not attempt to use this to upgrade a JACE—instead, you must use the Commissioning Wizard, as mentioned in the previous [Note](#).)
- [File Transfer Client](#)  
To copy files between your Workbench PC and the remote Niagara platform (in either direction). For example, you use this platform view when editing a JACE controller’s “system.properties” file—once to copy it from the JACE to your Workbench PC (for local editing), then afterwards to copy it back to the JACE. See “[system.properties notes](#)” on page 1-31.

- [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 text-based Niagara lexicon files from your Workbench PC to the remote Niagara platform, to either provide non-English language support, or customize English lexicons to globally change display tags. Starting in AX-3.7, lexicons are also 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.
- [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) on your Workbench PC.
- [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 the login credentials and TCP port monitored by the platform daemon, and (if applicable) to enable and/or require a secure (SSL) platform connection.
- [Station Copier](#)  
To *install* (copy) a station from your Workbench PC and the remote Niagara platform, including different file-level options. Also to *backup* (copy) a station in a remote platform to your Workbench PC, or to *delete* a remote station. You can also easily rename any copied 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](#)  
To navigate among all files and folders under the platform’s Niagara root (system home) directory, including the ability to make local copies on your PC.

In some cases, you may also have a **Sedona Environment Manager** platform view listed too. See the related [Note](#): in the section “[Platform overview](#)” on page 1-2.

## 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](#) (most JACE controllers)
- [Windows-based](#) (Supervisor host, SoftJACE, and select JACE controllers)
- [Linux-based Supervisor](#)

Among the first two types, there are various JACE host *models*, each with a “model” string descriptor. For a list of host models that are current with this document, see “[Models of platforms](#)” on page 1-11.

### QNX-based

Sometimes called “embedded” JACE controllers, these include JACE-2,-3,-6,-7 series controllers as well as older JACE-4 and JACE-5 series models, all shipped with the *QNX operating system*. All these devices use onboard *flash memory* for file storage and provide wired Ethernet connectivity. The JACE-2,-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.

**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 JACE controller or Supervisor).

The latest JACE-3E platform is available in 2013, introduced at the AX-3.7 update 1 release (AX-3.7u1). Positioned between a JACE-2 and JACE-6/6E in performance, a JACE-3E has onboard SRAM like a JACE-6E. At a prior 2012 AX-3.6 maintenance release (build 3.6.47), three other QNX-based JACE platforms were introduced: a JACE-6E controller (similar to JACE-6, but with onboard SRAM), and two “retrofit boards” for older Niagara R2 models of JACE-403 and JACE-545 controllers. With the retrofit board installed, these last two controllers are now considered “JACE-603” and “JACE-645” models.

See the following for further details on QNX-based JACEs:

- [“Sun Hotspot JVM or IBM J9 JVM”](#) on page 1-7
- [“Backup Battery \(or not\)”](#) on page 1-7
- [“Platform view differences, QNX-based vs. Windows-based”](#) on page 1-8
- [“Models of platforms”](#) on page 1-11

## Sun Hotspot JVM or IBM J9 JVM

Before AX-3.6, 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 (JACE-6, JACE-7), 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), 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 \(niagarad\)”](#) on page 1-4 for related details).

For the most part, these differences in Java VM and platform daemon are typically “transparent” to the normal configuration of the JACE’s hosted station or platform.

However, in AX-3.7 and later there are now notable advantages for a JACE using the Hotspot JVM, as follows:

- A Hotspot JACE supports IPv6 in its TCP/IP platform configuration. See [“TCP/IP changes in AX-3.6”](#) on page 1-73 for related details.
- A Hotspot JACE running AX-3.7 or later is capable of supporting secure encrypted (SSL) connections. See [“About a platform connection”](#) on page 1-3.

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

## Backup Battery (or not)

Until availability of the JACE-6E series in early 2012 and now the JACE-3E series in 2013, all QNX-based JACE controllers include a standard, integrated, backup battery. Typically, this is a NiMH (nickel metal hydride) onboard battery. A few models also support an additional external 12V SLA (sealed lead acid) battery. The backup battery allows continuous operation during brief power outages.

The JACE 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 QNX-based JACE in the **PowerMonitorService** in a running *station*, see [“About Platform Services”](#) on page 2-2.

**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, onboard 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, QNX-based vs. Windows-based

For any QNX-based platform, the following platform views differ from [Windows-based](#) platforms:

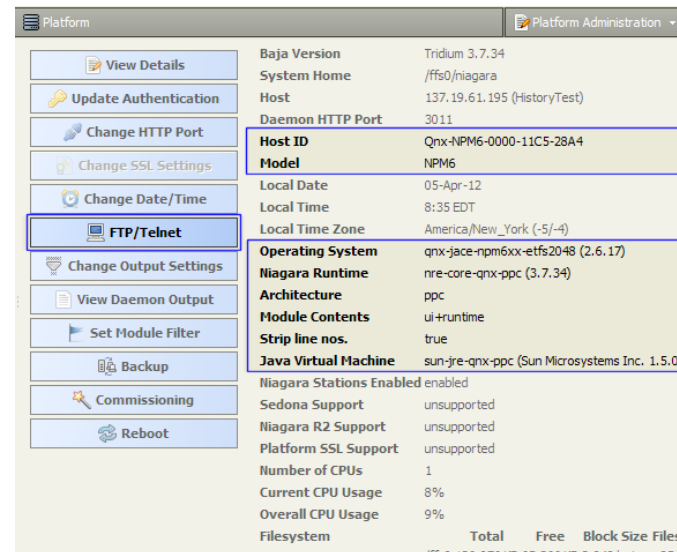
- [Platform Administration](#)

Also, in the [Application Director](#), you cannot **Start** a station after manually stopping it—you must reboot the JACE instead. See “[Application and output controls](#)” on page 1-17.

## Platform Administration

Platform Administration for a QNX-based platform ([Figure 1-4](#)) differs as follows:

**Figure 1-4** Platform Administration for QNX-based platform



- An **FTP / Telnet** button is available. For details, see “[FTP/Telnet](#)” on page 1-49
- Selections possible in the [Update Authentication](#) function are simpler.
- Various data in summary information (repeated in View Details) differs greatly from Win32 hosts.

See “[Platform Administration](#)” on page 1-40 for more details.

**Note:** Starting in AX-3.6 build 3.6.44 and later, if platform connected to a JACE-603 or JACE-645 controller (QNX-based “retrofit board” controller using an NPM6E processor), an additional button is available in the Platform Administration view, which is labeled either:

- **Install Niagara R2** — For a new uncommissioned controller, or one currently configured for NiagaraAX. Selecting this launches a wizard that *removes* all NiagaraAX configuration (including its station), and requires Niagara R2 2.301.535 or later installed on your Workbench PC.
- **Uninstall Niagara R2** — For a controller configured for Niagara R2, instead of NiagaraAX. Selecting this removes all Niagara R2 configuration, including its Niagara R2 license and station.

Refer to the Retrofit Board Niagara R2 Install & Startup Guide for complete details.

## 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 “Supervisor” 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.

- 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](#)” on page 1-10.
- 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 [QNX-based](#) platforms:

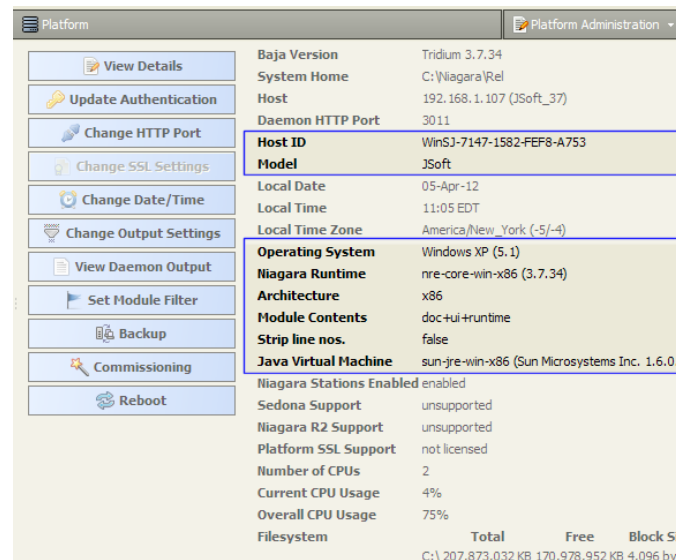
- [Platform Administration](#)

**Note:** When connected to any Windows host, a [User Manager](#) platform view is also available. Intended use is for a Windows-based JACE only. On a Windows Supervisor, you typically configure Windows users and groups using normal Windows administrative tools.

## Platform Administration

Platform Administration for a Win32-based platform ([Figure 1-5](#)) differs as follows:

**Figure 1-5** Platform Administration for Windows-based platform



- No **FTP / Telnet** button is available (this configuration can be done directly using Windows).
- Choices available from the [Update Authentication](#) function are more involved.
- Various data in summary information (repeated in View Details) differs greatly from QNX hosts.

**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](#)” on page 1-52.
- 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](#)” on page 1-54.
- Reboot is intended only for remote JACE platforms (see “[Reboot](#)” on page 1-55). To locally reboot a Supervisor, you should stop its local station, exit Workbench, then restart the operating system.

See “[Platform Administration](#)” on page 1-40 for more details.

## Win64-based Supervisor notes

Starting in AX-3.4, Supervisor support was added for installation on PCs running 64-bit Windows operating systems, for example Windows Server 2003 or Windows Vista 64-bit. Support for 64-bit Windows OS was expanded in AX-3.6 for Windows 7 Professional and Windows Server 2008, and again in AX-3.8 for Windows 8 Professional and Windows Server 2012. The primary application for 64-bit install is for a Supervisor station that has a very large NiagaraNetwork (job has large numbers of JACEs, each with many Niagara 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 with 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. See the following sections “[Known Limitations](#)” and “[Installation and interface differences](#)” for more details.

**Known Limitations** Please note that 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:

- Starting in AX-3.7, NRE serial support was *added* for a 64-bit Windows platform (in prior NiagaraAX releases, there was no 64-bit serial support). However, serial-based drivers (e.g. modbusAsync, flexSerial, various legacy drivers) are not typically licensed on a Supervisor or SoftJACE, and therefore are not fully tested or supported on a 64-bit platform—note the JACE-NXT is Win32 (Windows XP Embedded).  
Exceptions to such license rules can occur with 64-bit “engineering workstations” or “demo machines”. Again, in these cases note that 64-bit serial operation is not fully guaranteed. Additionally, a known issue with the 64-bit serial library may present itself in initialization phases, as has been noticed with usage of a 64-bit Niagara Serial Tunnel client (also new starting in AX-3.7). For related details see the section “Client side (PC application)” in the latest *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 or SoftJACE is not typically licensed for Lonworks. However “LonIP” is supported.
- LON tunneling drivers are not available for 64-bit Windows platforms.

**Note:** *OPC client driver support for Windows 64-bit platforms became available starting in AX-3.6 (build 3.6.36), as well as in builds 3.5.36 and 3.4.66 for earlier releases.*

**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 CD are used to install (setup\_x64.exe instead of setup.exe).

A platform connection to a Win64-based Supervisor or SoftJACE provides the identical collection of views as with a Win32-based host. Also, starting in AX-3.7 when opening a station running on a Win64 host, you now see a “SerialPortService” under its [PlatformServices](#) (formerly this service was missing).

## 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 machine.

**Note:** *Refer to the Engineering Notes document “Linux AX Supervisor Notes” for further installation details.*

The following sections provide platform-related details about a Linux Supervisor:

- [NiagaraAX platform rights on Linux Supervisor](#)
- [Default Linux Supervisor platform administrator](#)
- [Linux Supervisor platform views](#)
- [Linux Supervisor port usage notes](#)

## 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

At the time of this document update, a platform connection to a [Linux-based Supervisor](#) provides the same collection of [platform views](#) as to an AX-3.4 or later 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.

## Models of platforms

Among the two groups of JACE controllers ([QNX-based](#) 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 Niagara Network (**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.

[Table 1-1](#) lists various platform models (including JACE controllers), known at the time of this document, starting with the host model text descriptor.

**Table 1-1** Host models of JACE platforms

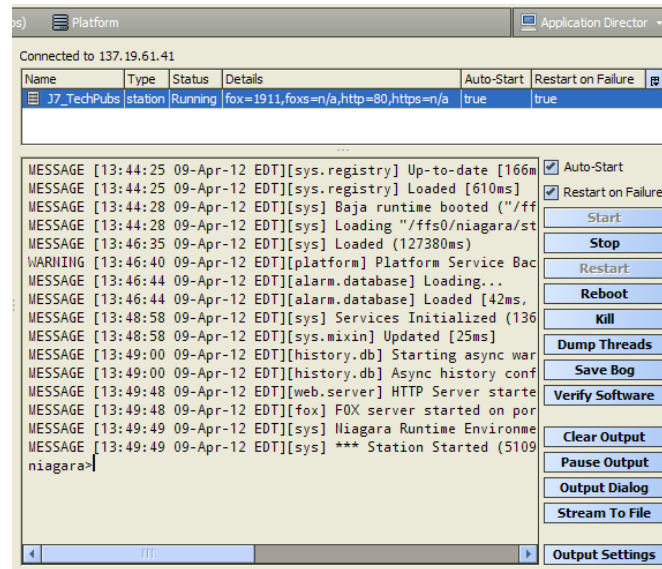
Model desc.	Actual Model	Notes
J402	JACE-402P	QNX-based. Uses the IBM J9 JVM (Java Virtual Machine). See the <i>JACE NiagaraAX Install and Startup Guide</i> for commissioning details.
J403	JACE-403 series	
J404	JACE-545 series, J5-R-AX (Rack Mount)	
J511	JACE-511 series	Discontinued models, QNX-based. Uses the IBM J9 JVM (Java Virtual Machine). See the <i>JACE NiagaraAX Install and Startup Guide</i> for commissioning details.
J512	JACE-512 series	
JNX	JACE-NX series	Discontinued model, Win32-based (Windows XP Embedded). See the <i>JACE-NX NiagaraAX Install and Startup Guide</i> for commissioning details.
JNXS	JACE-NXS series	Win32-based (Windows XP Embedded), previous model before latest JACE-NXT series. See the <i>JACE-NXS NiagaraAX Install and Startup Guide</i> for commissioning details.
JNXT	JACE-NXT series	Latest Win32-based JACE controller (Windows XP Embedded). See the <i>JACE-NXT NiagaraAX Install and Startup Guide</i> for commissioning details.
JVLN	JACE-7 series (JACE-700)	QNX-based, with more processing power than JACE-2/6 series. Introduced in the AX-3.5 time frame. See the <i>JACE NiagaraAX Install and Startup Guide</i> for commissioning details. <b>Note:</b> Starting in AX-3.6, this JACE series uses the Sun “Hotspot” JVM instead of the IBM J9 JVM, and also supports IPv6. A 802.11b/g wireless (WiFi) option is also available.
NPM2	JACE-2 series, including Security JACE (SEC-J-201), JACE-202 Express (M2M JACE 2)	QNX-based. Uses the IBM J9 JVM (Java Virtual Machine). Among the most popular. Includes the JACE-202 Express, or “M2M JACE 2” with onboard I/O points. See the <i>JACE NiagaraAX Install and Startup Guide</i> for commissioning details. For Security JACE (SEC-J-201) commissioning details, see the <i>Security Enterprise Guide</i> .
NPM3	JACE-3E series, introduced in mid 2013.	Latest QNX-based controller, a 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). See the <i>JACE NiagaraAX Install and Startup Guide</i> for commissioning details. <b>Note:</b> The JACE-3E requires AX-3.7u1 or later.
NPM6	JACE-6 series, including Security JACE (SEC-J-601), JACE-602 Express (M2M JACE 6)	QNX-based, with more processing power than the JACE-2 series. Among the most popular. Includes the JACE-602 Express, or “M2M JACE 6” with onboard I/O points. See the <i>JACE NiagaraAX Install and Startup Guide</i> for commissioning details. For Security JACE (SEC-J-601) commissioning details, see the <i>Security Enterprise Guide</i> . <b>Note:</b> Starting in AX-3.6, this JACE series uses the Sun “Hotspot” JVM instead of the IBM J9 JVM, and also supports IPv6.
NPM6E	JACE-6E, as well as “retrofit board” controllers JACE-603 and JACE-645.	Recent group of QNX-based controllers. The JACE-6E is like a JACE-6, but includes onboard SRAM for battery-less operation (if desired). The JACE-603 and JACE-645 are “retrofitted” Niagara R2 JACE-403 and JACE-545 controllers, configurable to run either Niagara R2 or NiagaraAX. If configured for NiagaraAX, use can be made of SRAM. All controllers use the Sun “Hotspot” JVM, and support IPv6 (if NiagaraAX). Refer to the <i>JACE NiagaraAX Install and Startup Guide</i> for commissioning details. For <i>Niagara R2</i> commissioning of a JACE-603 or JACE-645, refer to the <i>Retrofit Board Niagara R2 Install &amp; Startup Guide</i> .
Jsoft	SoftJACE installed on user-supplied PC	Windows-based. This is different than a Supervisor for example, which appears instead as “Workstation”.
Workstation	User-supplied PC, e.g. a Supervisor or engineering workstation.	Windows-based customer supplied PC that runs Workbench, minimally.
Linux version	<a href="#">Linux-based Supervisor</a>	Red Hat Linux Enterprise 5 operating system.

## Application Director

The Application Director is one of several [platform views](#). You use it to *start* or *stop* a *station* in any NiagaraAX 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.

**Note:** *Starting in build 3.6.44, the Application Director also applies to a Workbench platform connection to a JACE-603 or JACE-645 controller configured for Niagara R2. See the “Application Director” section in the document Retrofit Board Niagara R2 Install and Startup Guide for related details.*

**Figure 1-6** Application Director view, looking at NiagaraAX station

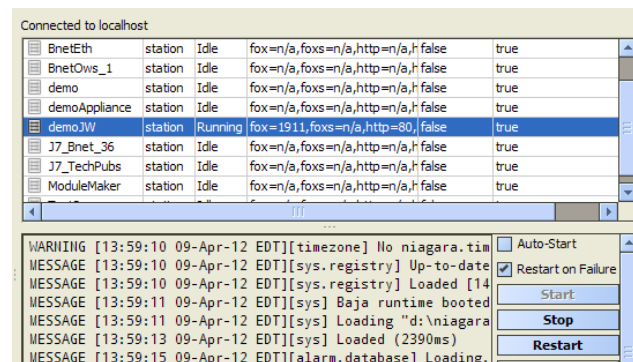


As shown in [Figure 1-6](#), the **Application Director** is split into three main areas:

- [Installed applications \(stations\)](#) — at top
- [Application output](#) — main area
- [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 in [Figure 1-6](#). However, the Application Director for a PC platform (Supervisor, NiagaraAX engineering workstation) may show multiple stations, as shown in [Figure 1-7](#).*

**Figure 1-7** Application Director for Supervisor host showing multiple stations



**Note:** *Prior to AX-3.6, the **Application Director** could show a maximum of 32 stations—regardless if more stations were under the host's `stations` directory. Starting in AX-3.6, the Application Director was changed to allow access to all stations, even if over 32.*

## Installed applications (stations)

The top area of the [Application Director](#) shows a table of installed applications (stations), as shown in [Figure 1-8](#). Apart from the data shown in the table, [application selections](#) are possible.

**Figure 1-8** Application Director installed applications

Name	Type	Status	Details	Auto-Start	Restart on Failure
J7_TechPubs	station	Running	fox=1911,foxs=n/a,http=80,https=n/a	true	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 it does not run the Fox service.
  - foxs= TCP/IP port monitored for secure (SSL) Fox connections to Workbench and other Niagara stations, if so configured (applies to AX-3.7 and later platforms). Shows “n/a” if host does not support (or is enabled) for Fox SSL, or if the station is not running, or if it does not run the Fox service.
  - 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.
  - https= HTTP port that the station’s WebService monitors for secure (SSL) browser connections to the station, if so configured (applies to AX-3.7 and later platforms). Shows “n/a” if host does not support (or is enabled) for Web SSL, or if the station is not running, or if it 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 (see “[Start checkboxes](#)” on page 1-17).
- **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 (see “[Start checkboxes](#)” on page 1-17).

## Application selections

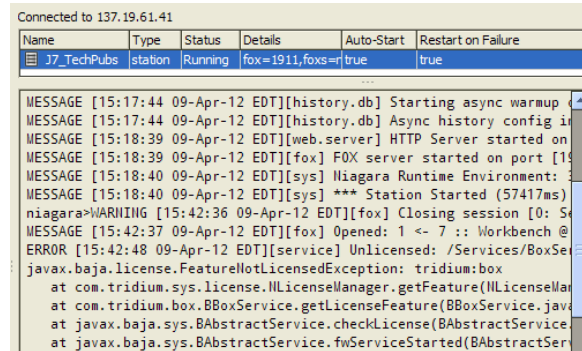
Click in the [installed applications](#) table for different results, as follows:

- **click**  
Click a station to select it, highlighting it. When a station is selected, its standard output appears, and all enabled right-side buttons apply to it. For details, see “[Application output](#)” on page 1-15 and “[Application and output controls](#)” on page 1-17.
- **right-click**  
Right-click a station for its shortcut menu (a *subset* of the application and output actions buttons). For details on included menu commands, see “[Application and output controls](#)” on page 1-17.
- **double-click**  
If running, double-click a station to open a Workbench (Fox) connection to it, showing its **Station Summary** view. Or, press Ctrl and double-click for a *new tab* showing the station connection. If not running, a station 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 ([Figure 1-9](#)).

**Figure 1-9** Station output in Application Director's application output area



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:** 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. For more details, see “[Output control buttons](#)” on page 1-18.

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 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 “level” of station output—see “[Station log levels \(spy:/logSetup\)](#)” on page 1-16.

The general format of a station output message is:

`TYPE [timestamp] [station_process] message_text`

For example:

`MESSAGE [13:53:08 08-Mar-12] [fox] FOX server started on port [1911]`

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

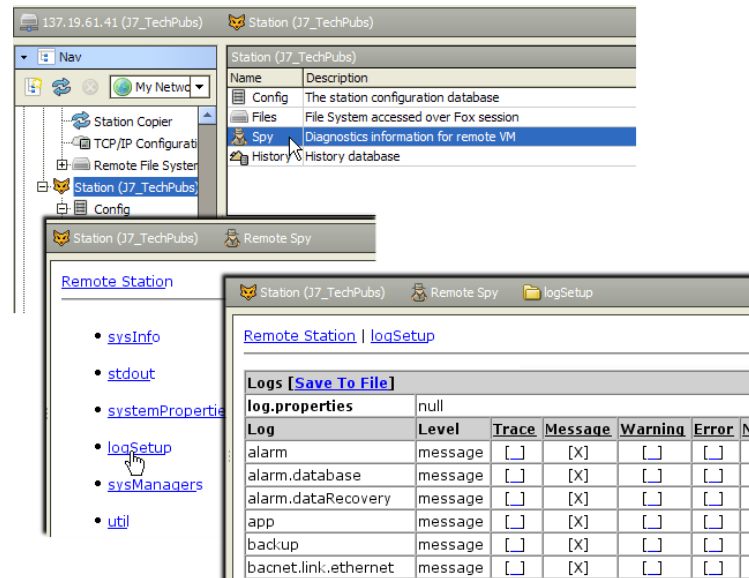
- **MESSAGE**  
Typically comprise most default station output messages. Usually, each message lets you know some process milestone was started or reached, such as a service or the station itself.
- **WARNING**  
Informs you of a potential problem, such as inability to open a specific port. Typically, 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. 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" page (Figure 1-10), you can change the "log level" of the station processes of interest in order to "tune" station output.

**Figure 1-10** Station spy logSetup (from Station Summary)



**Note:** To get to a running station's logSetup page in Workbench, double-click the station in the Nav tree for its **Station Summary** view. From there, double-click **Spy**, then click **logSetup**.

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 decreasing order of message volume, as follows:

- **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).
- **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**.

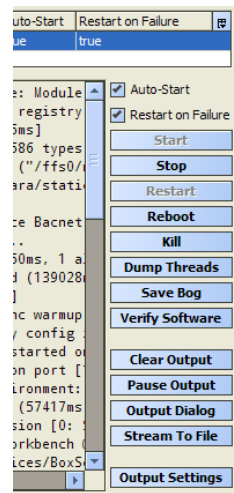
This is mentioned because when looking at a station's output, you are usually troubleshooting. As part of troubleshooting, you should always check the station's histories for LogHistory. It should contain recently recorded station errors and (if configured) warnings. This information may help when evaluating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



## Application and output controls

Unlike in most Workbench views, where changes are entered first and then applied with a “Save” button, in the [Application Director](#) when you click checkboxes and buttons ([Figure 1-11](#)), changes are applied *immediately* to the selected station.

**Figure 1-11** Application Director checkboxes and buttons



From top-to-bottom, these controls are grouped as follows:

- [Start checkboxes](#) (Auto-Start, Restart on Failure)
- [Application control buttons](#) (Start, Stop, Restart, Reboot, Kill, Dump Threads, Save Bog, Verify Software)
- [Output control buttons](#) (Clear Output, Pause Output, Output Dialog)
- [Output Settings](#) button

### Start checkboxes

For the currently selected station in the [Application Director](#), you can enable (check) or disable (clear) two start settings using checkboxes ([Figure 1-11](#)). 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 (or host rebooting) after successive failures.

**Note:** Changes are reflected 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 *QNX-based* JACE, 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 nonzero return code (e.g., engine watchdog had killed the station because of a “deadlock” condition).  
**Note:** *QNX-based* JACEs 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 JACE reboots.  
If a JACE 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 [“PlatformServiceContainer configuration parameters”](#) on page 2-4.

## Application control buttons

For the selected station in the [Application Director](#), application control buttons (Figure 1-11) apply as follows:

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

- **Start**  
Enabled only if the 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 [standard output](#) is cleared, and output messages begin with the new startup of that station.  
**Note:** *If you manually stop a station on a QNX-based 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 only if the 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 its configuration to its `config.bog` file, and potentially saving history data). See the preceding note about stopping the station on a QNX-based host.
- **Restart**  
(Available [Windows-based](#) platforms only) Enabled only if the 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](#)” on page 1-55.
- **Kill**  
Enabled only if the station has a status of “Starting”, “Stopping”, or “Running” in the [installed applications](#) area. When pressed, a popup confirmation dialog appears. If you confirm, the host’s platform daemon terminates the station process immediately.  
**Note:** *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 standard output area.*
- **Dump Threads**  
(station only) 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 [standard 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.  
**Note:** *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 NiagaraAX software, those modules are not copied.*

## Output control buttons

For the selected station in the [Application Director](#), output control buttons (Figure 1-11) apply as follows:

- **Clear Output**  
Enabled regardless of station status. When pressed, all text in the [standard output](#) area is cleared.  
**Note:** *Data in the 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 only if the station has a “Running” status in the [installed applications](#) area. When pressed, the button toggles to “**Load Output**”, and the next press 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.

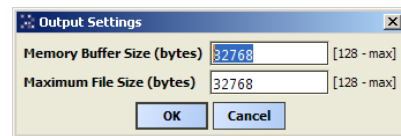


- When you press **Load Output**, text in the **standard output** area is reloaded with the station's buffered output, and output remains updating in real time.
- **Output Dialog**  
Enabled regardless of station status. When pressed, this produces a separate “non-modal” output window displaying the 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.*

## Output Settings

For the currently selected station in the **Application Director**, the **Output Settings** button produces a dialog (Figure 1-12) in which you specify how the platform daemon buffers the output from that station.

**Figure 1-12** 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.

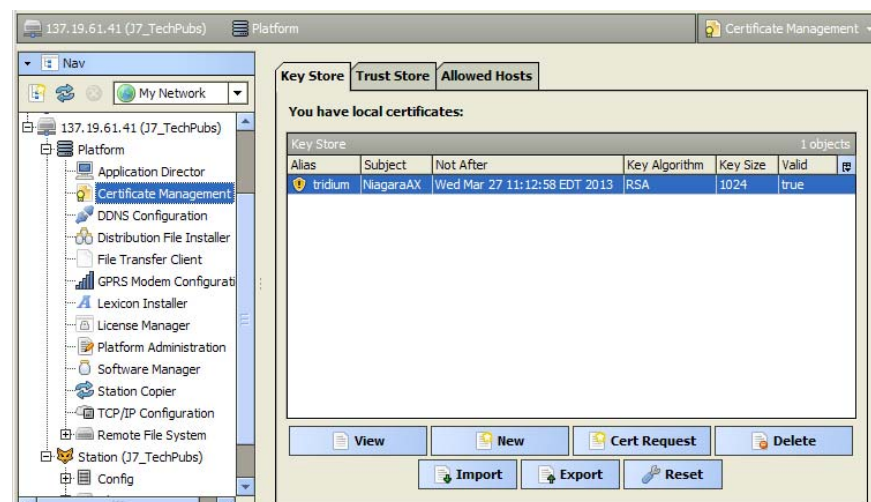
*Note: Changes to either output setting may result in the output buffer's contents to be cleared.*

## 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`.

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

**Figure 1-13** 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 (SSL or TLS) connections to this host.

Following the certificate management portion of configuration, starting in AX-3.7, 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"](#) on page 1-40.
- 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 *Drivers Guide*.
- Any browser (HTTP) connection to the station's web server. You enable this in properties of the station's WebService, typically 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:

- [Key Store](#)
- [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 more 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 all installed signed certificates by trusted entities (CA authorities) in the currently opened platform.

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

### **Allowed Hosts**

The Allowed Hosts tab in the [Certificate Management](#) view lists all approved hosts (with connection port) and associated self-signed certificates in the currently opened platform. These hosts (and certificates) cannot be validated against a CA authority.

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

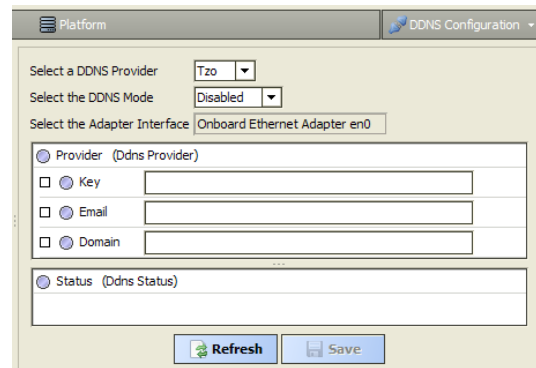
## DDNS Configuration

**Note:** 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.

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 1-14](#) shows the default **Ddns Configuration** view.

**Figure 1-14** Ddns Configuration platform view



**Note:** Refer to the Engineering Notes document “NiagaraAX-3.1 DDNS” for more details and examples.

The following sections provide a few basic DDNS configuration details:

- [DDNS core configuration items](#)
- [About TZO](#)

### DDNS core configuration items

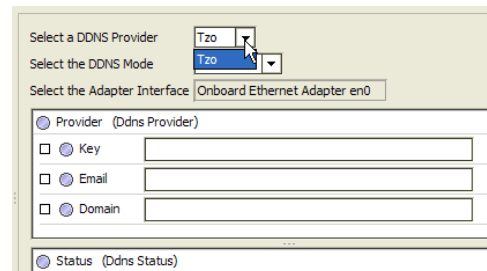
The platform [DDNS Configuration](#) view has the following configuration sections:

- [Provider](#)
- [Mode](#)

#### 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](#)” on page 1-22), although future builds may support other providers.

**Figure 1-15** DDNS Provider selection and Provider-supplied configuration properties



**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 ([Figure 1-15](#)).

- **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 (see [Note](#): on page 1-21). Note the original development and testing of NiagaraAX [DDNS Configuration](#) focused primarily with TZO accounts.

## Distribution File Installer

**Note:** *Starting in AX-3.6, the Distribution File Installer also applies to a Workbench platform connection to a JACE-603 or JACE-645 controller configured for Niagara R2. See the “Distribution File Installer” section in the document Retrofit Board Niagara R2 Install and Startup Guide for related details.*

The Distribution File Installer is one of several NiagaraAX [platform views](#), used to install dist files. A dist file is a zip file 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 of a remote JACE. Such a backup can be initiated using the [Backup](#) command from the platform’s [Platform Administration](#) view, or directly from the **BackupService** in the NiagaraAX station running on that host. For details, see “[Restoring a backup dist](#)” on page 1-25.

**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, in order to downgrade a JACE to an older release level, or restore it to a “known good” state before installing other software. Note that this *wipes* the file system (including all station files), leaving the JACE in a “near factory” state. For details, see “[Downgrading a JACE \(Clean Dist\)](#)” on page 1-26.

Note you *do not use* this view to *upgrade* a JACE. Instead, you *must use* the **Commissioning Wizard** to upgrade NiagaraAX in a JACE controller. In early NiagaraAX releases, upgrading core NiagaraAX software was possible (but not recommended) via the Distribution File Installer. The Commissioning Wizard is a right-click option on a platform when opened in Workbench. See “[Upgrading a JACE](#)” on page 1-29.

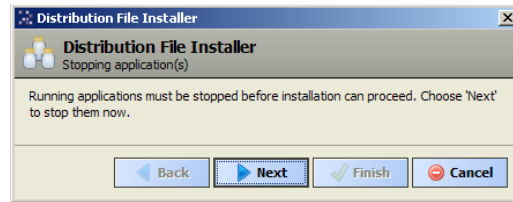


#### Caution

*If a station is already running on the remote host, any dist file install requires all applications to be stopped, after which all are invariably overwritten! After selecting a dist file, the Installer provides a confirmation dialog for this, as shown in [Figure 1-16](#). 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 controlled equipment that might be adversely affected by the JACE’s station stopping (and removal) is put in a manually controlled state.*

**Figure 1-16** Stop station dialog in Distribution File Installer



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

- [Security update 1 changes to backup dist usage](#)
- [AX-3.8 changes to backup dist usage](#)
- [Operation of the Distribution File Installer](#)
- [Restoring a backup dist](#)
- [Downgrading 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 JACE 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](#)” on page 1-43.

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 AX-3.8 host 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 an AX-3.8 host (unlike in a backup `.dist` file for any AX-3.7u1 or earlier release JACE).

In some cases this may result in confusion when *restoring* a station backup made from a AX-3.8 host, as this varies from restoring a station backup made from a host running AX-3.7u1 or AX-3.7 (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 (pre-AX-3.8) 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**

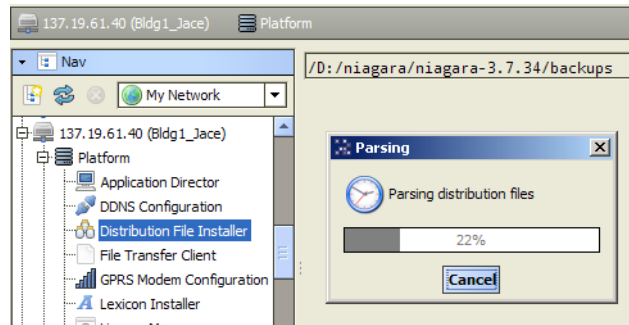
The following subsections explain more about dist file selection and the install process:

- [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 (Figure 1-17).

**Figure 1-17** Parsing dialog in Distribution File Installer

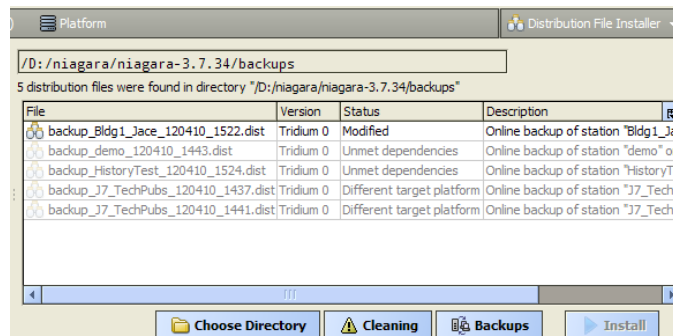


By default, the first time you use the Installer, the “backup” folder under the Niagara software location (!\backups) is parsed. If that folder does not exist yet (no backups have been made), then the “cleanDist” folder (!\cleanDist) is parsed instead.

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.

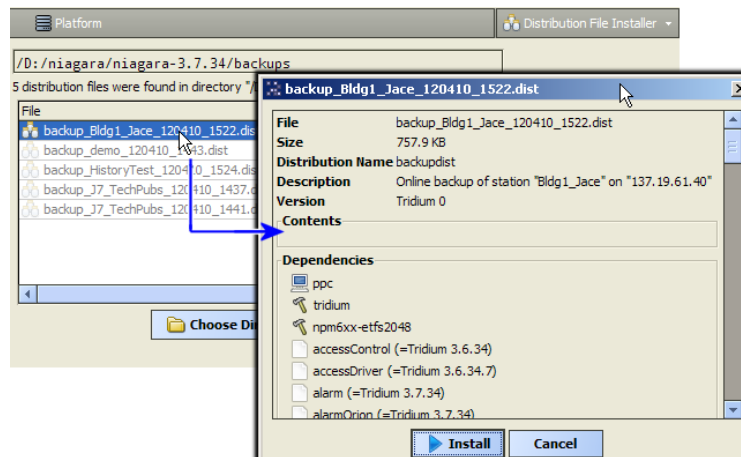
When parsing completes, a table of the found dist files appears, with the appropriate dist files available for selection in the table, as shown in Figure 1-18 (using default software location).

**Figure 1-18** 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 (Figure 1-19).

**Figure 1-19** Example details dialog for a dist file, showing dependencies.



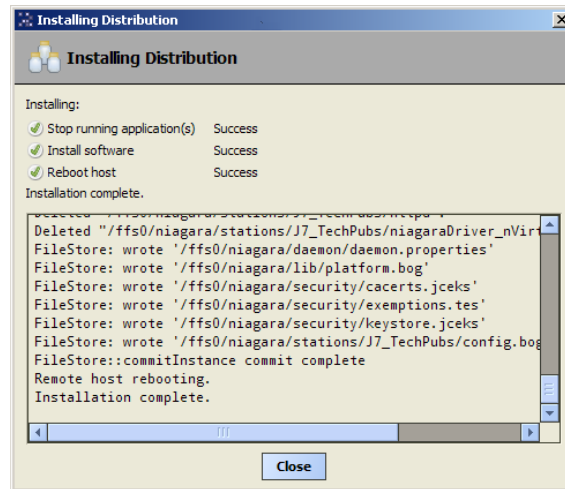


To be able to restore a backup dist file, your Workbench installation requires the same versions of software, including modules, to be available in its “software database.” Therefore, it is recommended that you make and keep frequent backups as you upgrade JACEs.

## 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 in [Figure 1-20](#).

**Figure 1-20** 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](#).

## Restoring a backup dist

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. You can do a backup from the **Platform Administration** view—for details, see [“Backup”](#) on page 1-53.

More typically, backups are done from Workbench *station connections* (must be a station running, that has the BackupService). In the Nav tree, right-click the opened station, and select **Backup Station**.

To *restore* a backup, in the [Distribution File Installer](#) click the **Backups** button for the ! /backups folder (if not already there), or if needed, click the **Choose Directory** button to point to another backup dist file location.

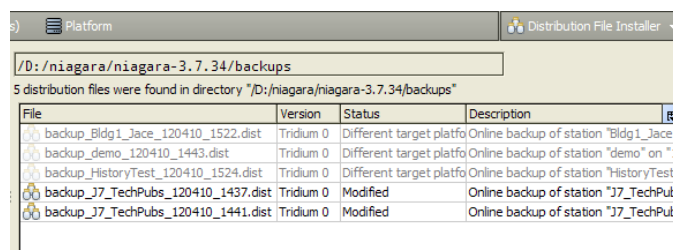
**Note:** Depending on NiagaraAX release of the host with the station backup, these restore considerations apply:

- If an AX-3.7u1 (update release) backup .dist, installation to a host other than the original host may first require manual editing. See [“Security update 1 changes to backup dist usage”](#) on page 1-23.
- If an AX-3.8 host station backup, it is *unnecessary* to first edit the backup .dist file. However, be aware that the handling of JACE platform credentials has changed in AX-3.8 station backups. See [“AX-3.8 changes to backup dist usage”](#) on page 1-23

For complete details, refer to the NiagaraAX 2013 Security Updates engineering notes document.

The Installer parses through the available dist files, and makes selectable only those files available that are compatible with the opened JACE platform. When done parsing, available backup dists appear listed, as shown in [Figure 1-21](#).

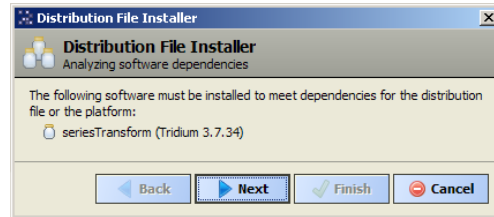
**Figure 1-21** Backup dist files listed in Distribution File Installer



Double-click any `.dist` for more details in a popup dialog. To restore the backup, click **Install** to continue. If the host is already running a station, a dialog appears telling you that it must be stopped, as previously shown in [Figure 1-16](#) on page 1-23.

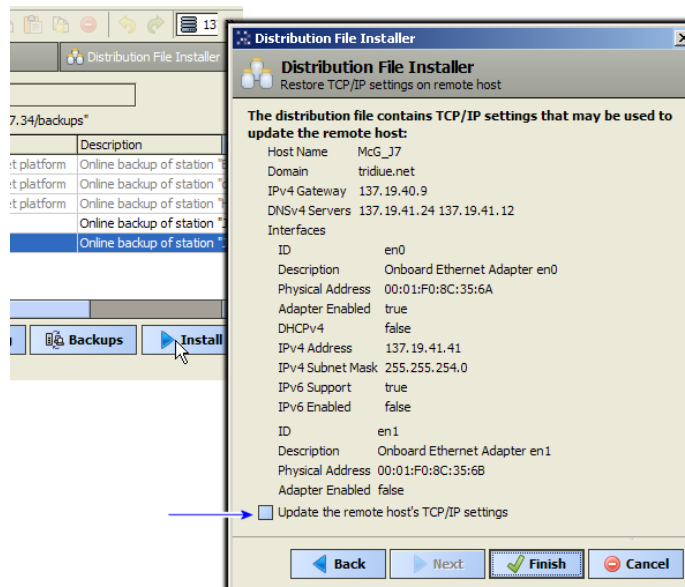
If the `.dist` file contains software modules different from (or in addition to) those already installed in the remote host, another dialog appears to inform you, as shown in [Figure 1-22](#). Click **Next** to continue.

**Figure 1-22** Software dependencies dialog in Distribution File Installer



As shown in [Figure 1-23](#), whenever restoring a backup, another dialog always asks if you wish to restore the TCP/IP settings stored in the dist file (as displayed) into the remote host.

**Figure 1-23** Restore TCP/IP settings in remote host from dist file



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

- If you *leave* this cleared, after the dist file installs and host reboots, it *retains its current* TCP/IP settings. This allows you to use the same dist file on differently addressed hosts, if needed.
- If you *check* (select) this checkbox, after the dist file installs (and host reboots), it will use the TCP/IP settings stored in the dist file—meaning the same ones shown in this dialog.

When you click **Finish**, the [distribution file install process](#) begins.

### Downgrading a JACE (Clean Dist)

Although the NiagaraAX Framework has historically provided smooth *upgrades* between software releases, *downgrading* from one software release to an earlier release can present compatibility problems. This applies particularly to [QNX-based](#) JACEs, as binaries for the (QNX) OS are included in dist files.

However, at times it may be necessary to install an older NiagaraAX release onto a JACE, or to restore a JACE to a known good state. “Clean dist” files became available for this. To downgrade a JACE, or otherwise “wipe it clean to start over,” you can install a clean distribution file.



#### Caution

*Installing a clean dist will wipe the file system and install an appropriate version of Niagara platform daemon, resetting the unit to a “near factory state.” If the JACE 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 SSL configuration (if an AX-3.7 or later SSL-configured unit)

All other data is removed from the file system, including station bog files, Px files, modules, etc. Note that if an SSL-configured JACE (AX-3.7 or later), its “private key” information is not backed up.

In addition, a clean dist restores the factory-default platform credentials and port (3011).

Therefore before installing a clean dist file, make sure to backup station files plus any other modules on the JACE you wish to keep. You should backup (export) certificate keys from a AX-3.7 or later SSL-configured unit, such that if the JACE needed to be replaced (hardware swap-out), you could re-import those keys.

Remember, that a “Station Backup” creates a dist file that (when restored) includes the same level of software installed at backup time, so instead of (or in addition to) this type of backup, you may wish to use the **Station Copier**. For related details, see “[Station Copier](#)” on page 1-63.

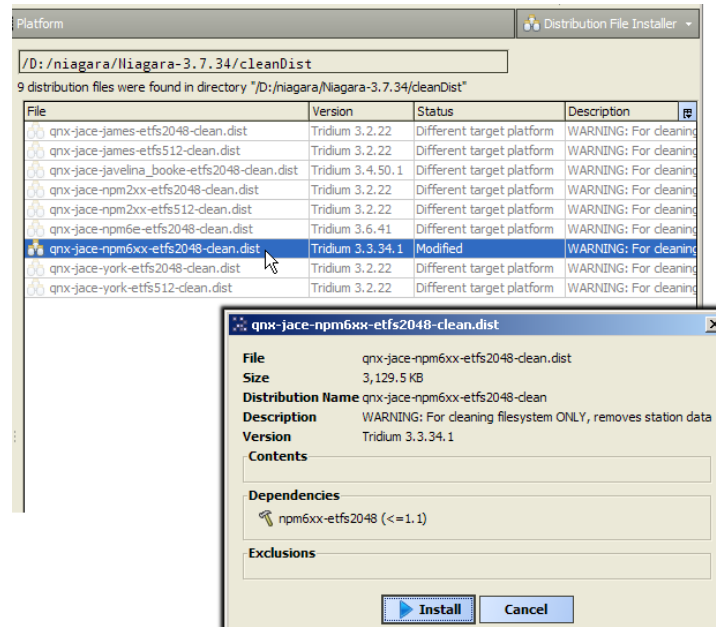
Note that after installing a clean dist, you must recommission the JACE for NiagaraAX, using the **Commissioning Wizard**.

**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](#)” on page 1-23.

A clean dist file has the suffix “-clean” in its name. A clean dist file for most of the various QNX-based JACE hardware platforms is located under a !\cleanDist folder—apart from other dist files under your software database.

Clean dist files appear listed in the Installer with a “WARNING” in the Description column, as shown for the one highlighted in [Figure 1-24](#).

**Figure 1-24** Clean dist file shown selected in the Distribution File Installer



**Note:** Clean dists support re-installation of AX-3.5 or later. Note that newer QNX-based JACEs require a minimum release level of NiagaraAX for installation, because of platform dependencies. For example, a JACE-3E requires build 3.7.105 or later of AX-3.7 (AX-3.7u1), a JACE-6E requires AX-3.6 or later, a JACE-700 requires AX-3.5 or later, and so on.


As for any dist file, only the appropriate one for the currently opened platform will be selectable.

The following procedures describes the downgrade process for a QNX-based JACE from AX-3.7u1 to an earlier release. Note that because station architecture changed in AX-3.7, a second procedure was added about downgrading from AX-3.7u1 to AX-3.6u4 while still maintaining interim station data.

- [Example JACE downgrade from AX-3.7 to an earlier release](#)
- [Example JACE downgrade from AX-3.7u1 to AX-3.6u4, maintaining interim station data](#)

### Example JACE downgrade from AX-3.7 to an earlier release

To downgrade a QNX-based JACE from any AX-3.7 build to an earlier release (and install a station compatible with that earlier release):

- Step 1 Before installing a clean dist file, make sure you have backed up the station files plus any other data and modules on the JACE you wish to keep.
- Step 2 Start AX-3.7 Workbench and open a platform connection to the JACE.
- Step 3 Open the [Distribution File Installer](#) and click  **Cleaning** button for the !/cleanDist directory.
- Step 4 Select the appropriate clean dist file for the platform and install.
- The file system clean will take a few minutes, then the JACE will automatically reboot. Wait for the reboot to complete.

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

- Step 5 To re-install the software versions to the JACE:
1. Use a version of Workbench that runs the same software versions that you want on the JACE, and 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*.
  2. If you have a backup dist file for the JACE that was made when it had the desired older software versions, use the Distribution File Installer to install it. See the previous section, “[Restoring a backup dist](#)” on page 1-25.


### Example JACE downgrade from AX-3.7u1 to AX-3.6u4, maintaining interim station data

**Note:** *This assumes you have not changed the JACE's station database to add AX-3.7 components. Also, once a station is started in a 2013 update release (such as AX-3.7u1), password storage is changed such that an older NiagaraAX release (e.g. AX-3.6) will be unable to start a saved station (config.bog). A workaround is possible, but all AX-3.6 and AX-3.5 JACEs should be upgraded to 2013 “update 4” releases for security reasons (AX-3.6u4 and AX-3.5u4) anyway—and these releases are password compatible with AX-3.7u1. For background information on this, refer to the NiagaraAX 2013 Security Updates document.*

In the case where you have upgraded an existing JACE from AX-3.6 to AX-3.7u1, but for whatever reason now wish to downgrade it back to a lower release level (e.g. AX-3.6u4) and still maintain interim saved station data (e.g. alarms and histories), please note there are special considerations and steps required.

- Step 1 Use AX-3.7u1 Workbench to make a platform connection to the JACE, then use the **Station Copier** to save its station database to your Workbench PC.
- Step 2 Copy the saved JACE station subfolder from your AX-3.7u1 Niagara !/stations folder to your AX-3.6u4 Niagara !/stations folder.

**Note:** *Perform all remaining steps below using AX-3.6u4 Workbench.*

- Step 3 On your Workbench PC, start the AX-3.6u4 platform daemon. For example:  
**Start > All Programs > Niagara AX 3.6.405 > Install Platform Daemon**
- Step 4 Use AX-3.6u4 Workbench to open a local (My Host) platform connection.
- Step 5 Start the copied JACE station (locally) on your PC.
- You will see error and warning messages, but the station should start.
- Step 6 After the station starts, **Save** it. Then stop the station.
- Step 7 Continue to use AX-3.6u4 Workbench to open a platform connection to the JACE to be downgraded.
- Step 8 Open the [Distribution File Installer](#) and click  **Cleaning** button for the !/cleanDist directory.
- Select the appropriate clean dist file for the platform and install. Cleaning wipes the station and all software modules from the JACE. The file system clean will take a few minutes, then the JACE will automatically reboot. Wait for the reboot to complete.

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

- Step 9 Reopen a platform connection to the JACE and use the AX-3.6u4 **Commissioning Wizard** to commission the JACE, selecting to install the newly-saved station.

Station selection in the wizard should pre-select all required software modules, but in the software installation (Select Modules) step you can review and select additional modules if desired.

Step 10 Review the final step in the Commissioning Wizard and initiate the wizard.

Allow sufficient time for the software install and station startup; then open a station connection to the JACE and perform a checkout.

**Note:** *If necessary, remember to restart the AX-3.7u1 platform daemon (niagarad) on your local Workbench PC, as it will be currently running the AX-3.6u4 platform daemon, from [Step 3](#) above.*

## Upgrading a JACE

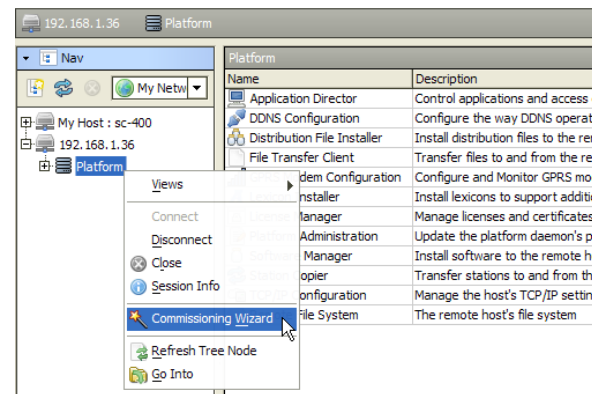
You must use the **Commissioning Wizard** to upgrade NiagaraAX software in a JACE. This means either an “update” upgrade (say from build 3.7.44 to 3.7.106), or a full “minor” release upgrade, for example build 3.7.106 to build 3.8.31.

**Note:** *When updating a multi-station system for the first time to an update release (e.g. AX-3.7u1) or AX-3.8, it is recommended to upgrade a Supervisor before its subordinate JACEs. For related details, see the “Upgrade considerations” section in the NiagaraAX 2013 Security Updates document.*

Any JACE to be upgraded from one minor version to another, say from 3.6.nn to 3.8.nn, *requires a JACE 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.

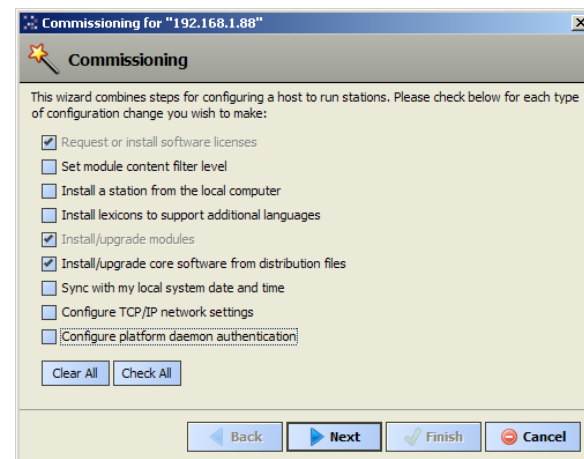
With a platform connection to any NiagaraAX JACE, access the Commissioning Wizard by simply right-clicking on that platform and selecting it from the menu ([Figure 1-25](#)).

**Figure 1-25** Commissioning Wizard is right-click option of opened platform



If a JACE upgrade, in the wizard’s opening “selection of steps” you typically *deselect* most items that were previously run at the JACE’s initial commissioning time—for example to set the module content filter level, set date and time, install lexicons, and so on. See [Figure 1-26](#).

**Figure 1-26** Typical upgrade selections for existing NiagaraAX JACE (already running a station)



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

- In the case where the upgrade requires an updated license installed:  
“Request or install software licenses”
- 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. 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*.

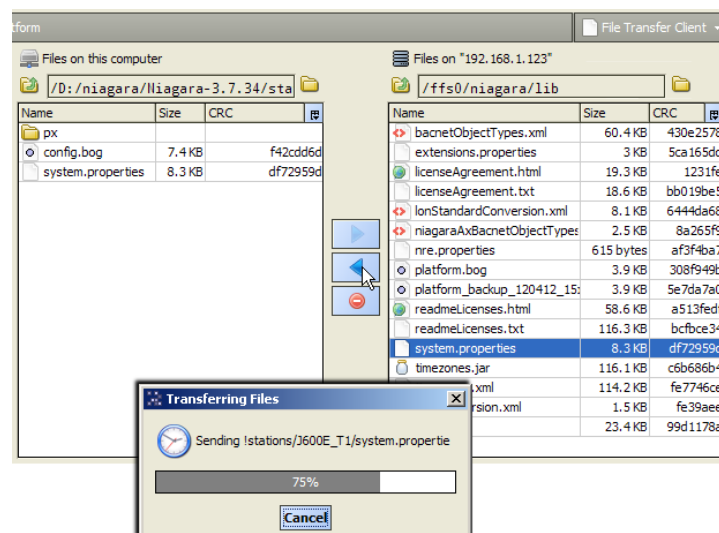
**Note:** Platform tools prior to Workbench AX-3.5 skipped license checks before JACE upgrades. However, following an upgrade without an adequate license, a license error could keep the JACE’s station from starting. Other platform views in Workbench were updated to prevent inappropriate upgrades—checking first that the necessary license is installed. This affects behavior of not only the Commissioning Wizard and *Distribution File Installer*, but also other platform views where “inadvertent” upgrades may have previously occurred, such as when using the *Station Copier* or *Software Manager* views.

## File Transfer Client

**Note:** Starting in build 3.6.44, the File Transfer Client also applies to a Workbench platform connection to a JACE-603 or JACE-645 controller configured for Niagara R2. See the “File Transfer Client” section in the document *Retrofit Board Niagara R2 Install and Startup Guide* for related details.

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 NiagaraAX platform. You can also use it to *delete* files and folders.

**Figure 1-27** File Transfer Client



This may be useful if you wish to copy graphics images to a JACE, as one example. Or, use it to copy text files from the `! / lib` folder of a remote JACE to your local PC, to allow editing. Then you can use the File Transfer Client to copy the edited version back to the JACE’s `! / lib` folder. An example of doing this is for a JACE’s `system.properties` file. See “[system.properties notes](#)” on page 1-31.

However, *do not use* the File Transfer Client to *copy modules* to a JACE, as the “module content filter” is not applied. As a result, larger files are transferred, and unnecessary flash drive space is used on the JACE.

The File Transfer Client provides a two-pane view, as shown in [Figure 1-27](#).



### Caution

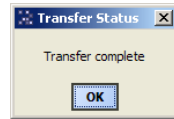
Be careful when using the File Transfer Client, especially when copying files to a target JACE platform, or whenever using the delete (X) control. Note that in either direction, when transferring a file and an identically-named file already exists, a popup confirmation dialog appears before the copy.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

In the File Transfer Client view, 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 in [Figure 1-28](#).

**Figure 1-28** Transfer complete dialog



### system.properties notes

Occasionally there may be a need to edit the `system.properties` file on a NiagaraAX platform, which is located in its `!/lib` folder (`niagaraRel/lib/system.properties`). Only a few entries in this 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.*

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 a JACE's `system.properties` file in place. Instead, you must copy it to your PC first (using the platform **File Transfer Client**) to edit.

### Station restart needed

A station must be *restarted* before changes to `system.properties` become effective. Thus, after copying (transferring back) an edited `system.properties` to any QNX-based JACE using the platform **File Transfer Client**, do the following:

1. **Stop** the station using the platform **Application Director**.  
Wait for the station to stop completely, ensuring that it saves its database.
2. From the platform **Platform Administration** view, select **Reboot**.  
Allow sufficient time for the JACE to reboot and station to start.
3. Reconnect to the JACE's station with Workbench to verify operation.

## GPRS Modem Configuration

The GPRS Modem Configuration view (Figure 1-29) is one of several NiagaraAX platform views for a QNX-based JACE. It is used to configure the (wireless) GPRS 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 JACE 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.
- 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 1-29** 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 (again, this also applies to the equivalent **Gprs Platform Service Plugin** view).

The following sections provide a few basic GPRS modem configuration details:

- [GPRS modem configuration sections](#)
- [Status and Runtime Data area](#)

### GPRS modem configuration sections

As shown in Figure 1-29, the **GPRS Modem Configuration** view (or the **Gprs Platform Service Plugin** view) has the following configuration sections:

- Modem Configuration
- Provider Configuration
- SMS Configuration

Also, a [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.

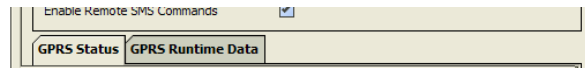
## SMS Configuration

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 1-30** Tabs near the bottom of the GPRS Modem Configuration view

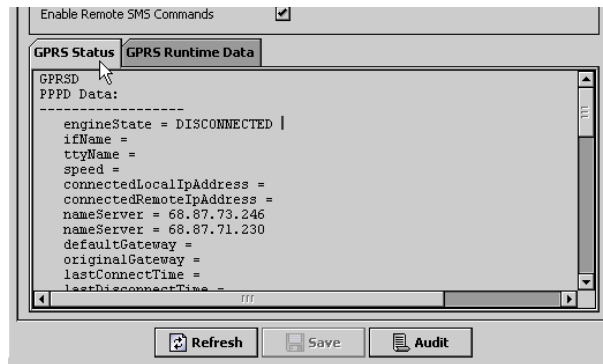


This data is found on two separate tabs: [GPRS Status](#) and [GPRS Runtime Data](#).

- 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 1-31** 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 1-32** GPRS Runtime Data tab in GPRS Modem Configuration view (or Gprs Platform Service Plugin)

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

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

**Note:** If a JACE-603 or JACE-645 configured for Niagara R2 (and not NiagaraAX), this view is not used. Instead, you use a tab under a different “R2 Platform Tool” platform view for installing Niagara R2 lexicon sets. For details, see the “R2 Installation Manager” section in the Retrofit Board Niagara R2 Install & Startup Guide.

The Lexicon Installer is one of several NiagaraAX [platform views](#). This view lets you install *text-based* NiagaraAX “lexicon sets” from your Workbench PC to a remote JACE platform, as needed.

Note that starting in AX-3.7, 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 in AX-3.7 (and later) builds as modules, using a module file name convention of:

```
niagaraLexiconLc.jar
```

where *Lc* is the two-character “language code”, such as *Fr* for French or *Es* for Spanish. In addition, Workbench 3.7 or later provides an updated **Lexicon Tool** with a special “Lexicon Module Maker” view you can use to modify or make new lexicon modules, from edited text-based lexicon files.

For complete details, refer to the *NiagaraAX Lexicon Guide*.

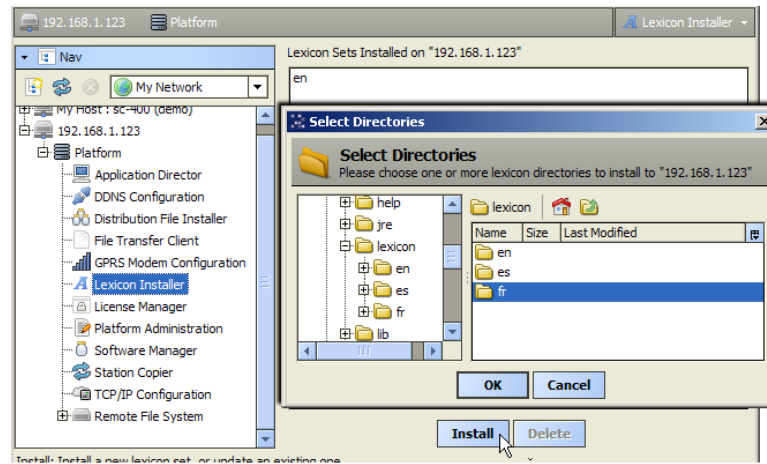
Lexicons in NiagaraAX typically have one of two uses, depending on job location:

- 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. See the *User Guide* sections “About lexicons” and “Lexicon editor” for more details.

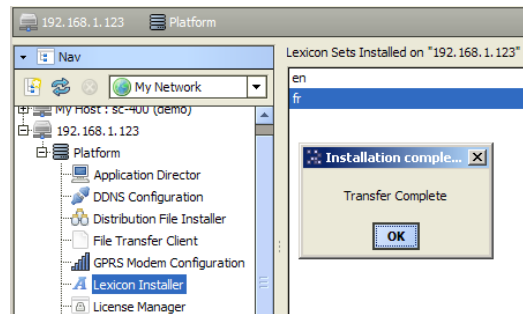
When you select Lexicon Installer, any existing text-based lexicon sets (already installed in that platform) are listed in the view pane. When you click Install, a “Select Directories” dialog appears for you to select lexicon sets (on your Workbench PC) to install in the remote platform, as shown in [Figure 1-33](#).

**Figure 1-33** Lexicon Installer, selecting lexicon



When you click **OK**, the selected lexicon directory is installed in the remote JACE platform. An “Installation Complete” dialog appears when all files are transferred, as shown in [Figure 1-34](#).

**Figure 1-34** Lexicon Installer, lexicon installed



## License Manager

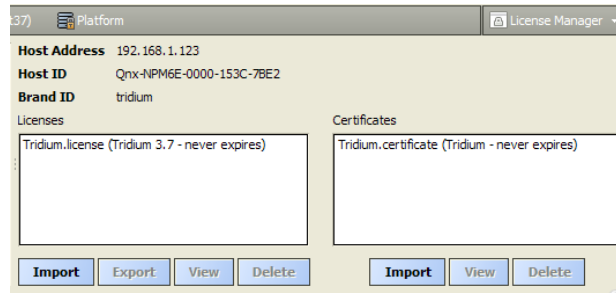
**Note:** If a JACE-603 or JACE-645 configured for Niagara R2 (and not NiagaraAX), note that this view applies only to the NiagaraAX license for that controller—something always required. However, you use a tab under a different “R2 Platform Tool” platform view for viewing/installing the controller’s Niagara R2 license. Niagara R2 licensing is not covered in this document. For details, refer to the “R2 Installation Manager” section in the Retrofit Board Niagara R2 Install & Startup Guide.

The License Manager is one of several [platform views](#). This view lets you install (import) licenses and certificates to a remote JACE platform, sourced either from your Workbench PC or the Niagara licensing server. You can also view contents of licenses and certificates, and if desired, delete them from a JACE.

**Note:** A “floating license repository” (FLR) option is available in NiagaraAX, to permit licensing of Workbench hosts and Supervisors with a “host-independent” client license. Such licenses are “leased” from a job’s local FLR server station, running on a specially-licensed host with one or more “license packs”. This different licensing is explained in an engineering notes document “Floating License Repository”.

Please note that the License Manager, as well as the LicenseService under a station’s PlatformServices, does not work with floating licenses or license packs, only traditional, host-specific, “node-locked” licenses typical to JACEs. This NiagaraAX Platform Guide describes working with traditional licenses only.

**Figure 1-35** License Manager lists existing NiagaraAX licenses and certificates



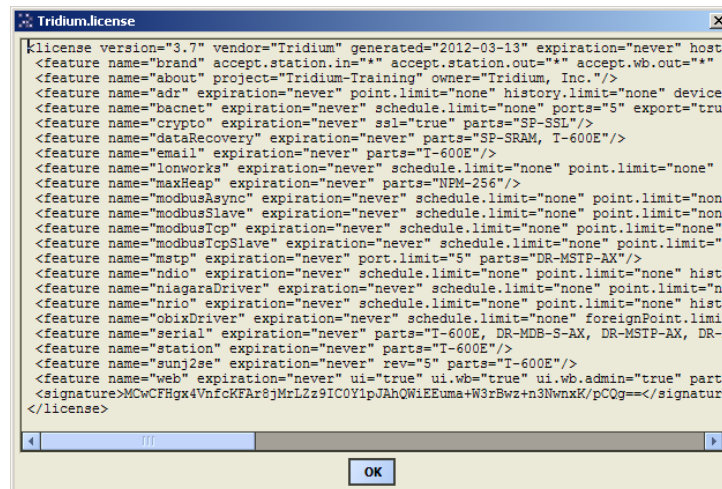
The License Manager lists any existing NiagaraAX licenses and certificates (already installed in that platform), as shown in Figure 1-35.

If selected, you can also view or delete an *existing* license file (**View** button is the same as simply double-clicking item, see Figure 1-36), or save (export) a license file as a “license archive” (.lar) file.

Buttons below each side let you install (import) a *new* license or certificate file. Typically, license files are imported from either the online licensing server or from your local license database.

Click a license or certificate to select it, or *double-click* to view in a dialog, as shown in Figure 1-36.

**Figure 1-36** Viewing a license in License Manager



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 Niagara host*, and enables a set of vendor *features*. All NiagaraAX hosts require a branded “Tridium” license. If third-party modules are installed, one or more additional licenses may be needed. For details about license file contents, see the section “[About NiagaraAX license files](#)” on page A-8.
- 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 NiagaraAX 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 JACE inoperable until a proper license or certificate is reinstalled!*

For further [License Manager](#) details, see the following sections:

- [License operations](#)
- [About the licensing server](#)

**Note:** 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 an appendix to this document, “[License Tools and Files](#)”, along with details about the contents (features) of license files.

## License operations

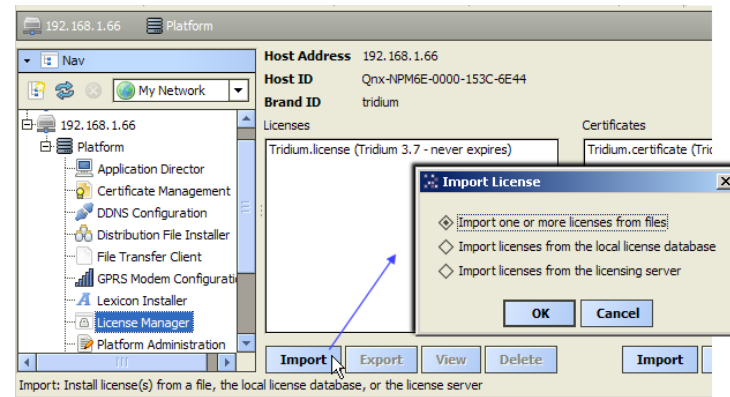
Below the *license-side* of the [License Manager](#) (Figure 1-37), these two buttons (commands) are displayed in addition to **View** and **Delete**:

- **Import** — 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** — Available if you have a license selected, to save locally as a “license archive file.”

### Import

If you choose **Import** from the License Manager, the Import License dialog asks you to select where the source license is, as shown in Figure 1-37.

**Figure 1-37** Import dialog from License Manager



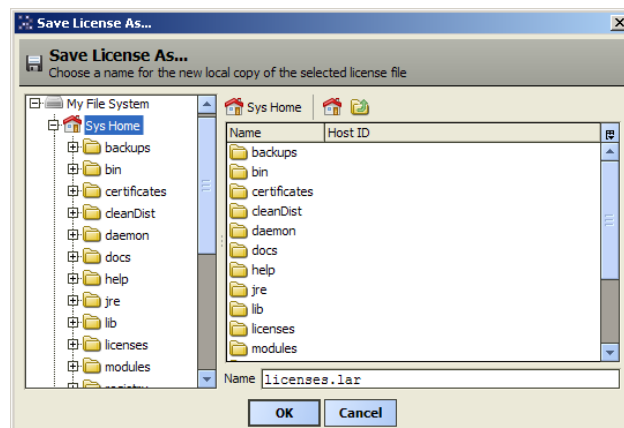
Select *one* of the following options (depending on scenarios, some may be unavailable, as noted):

**Note:** See “[License Import results](#)” on page 1-38 for details on results after making a selection below.

- **Import one or more licenses from files**  
Always an available option, this provides a **Select File** dialog 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 will be unavailable (dim) if this 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. See “[About the local license database](#)” on page A-6 for related details.
- **Import licenses from the licensing server**  
Typically, this option is available if your Workbench PC has Internet connectivity. When you select this option, Workbench silently searches the licensing server and installs the license.

### Export

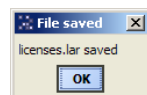
With a license selected in the [License Manager](#), the **Export** button provides a **Save License As...** dialog to save that license file locally on your Workbench PC, as a *license archive* (.lar) file. See [Figure 1-38](#). For related details, see “[About license archive \(.lar\) files](#)” on page A-7.

**Figure 1-38** 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 Workbench License Manager view of Workbench.

By default, a license archive file is saved in the root of your Niagara release directory. If needed, you can use the dialog'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 `MyJaceNxs.lar`.

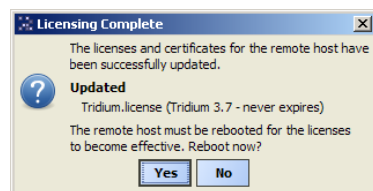
After exporting a license, a notification dialog appears in Workbench, as shown in [Figure 1-39](#).

**Figure 1-39** Exported license archive notification dialog

## License Import results

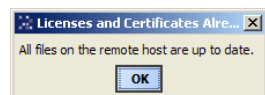
Depending on the **Import** option chosen in the **License Manager** ([Figure 1-37](#)) and the success of the import attempt, after you click **OK**, one of several dialogs may appear to signal completion, as follows:

- **Licensing Complete**  
The license was successfully added, as shown in [Figure 1-40](#).

**Figure 1-40** Licensing Complete dialog

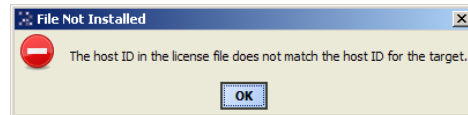
**Note:** If a station is running on the host platform, this dialog informs you that the host must be rebooted (if a QNX-based platform) or station restarted (if Win32-based platform) to become effective, and provides a **Yes** button to do this now. Or, you can select **No** and do this manually later.

- **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 in [Figure 1-41](#).

**Figure 1-41** 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 [Figure 1-42](#).

**Figure 1-42** 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. See [Figure 1-43](#) on page 1-39.

## About the licensing server

For traditional NiagaraAX license files validated against the Tridium certificate, installation can be automated from Workbench. All such purchased licenses (including JACEs, Supervisor, or Workstation-only) are stored and available to Workbench through the online *licensing server*.

**Note:** *The licensing server is the final license authority—the most current version of any NiagaraAX host platform’s license is always stored there. In addition to accessing licenses via the licensing server when using the License Manager, operations in other Workbench views access the licensing server too. Examples include the [Workbench License Manager](#) tool of Workbench, or the Network License Summary view of the “Licenses” slot of the NiagaraNetwork’s ProvisioningExt.*

Providing that your PC currently has Internet connectivity while running a platform connection to any Niagara host, the [License Manager](#) allows you to automatically retrieve and install any needed licenses.

Do this with the **Import** button, then selecting the license server option. See [“Import” on page 1-37](#) for details. As a side benefit, your “local license database” is also updated.

**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, as shown in [Figure 1-43](#).*

**Figure 1-43** License request form in browser (from Workbench, Tools > Request License)

This lets you submit a license request to the licensing server that includes the platform’s Niagara Host ID. In this dialog, be sure to enter your name, and email address ([Figure 1-43](#)).

If you already have been sent a “License Key”, note that 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 license file for this JACE will be emailed back to the entered address. This license will typically be a zipped “license archive” (have a “.1ar” extension)—see “[About license archive \(.lar\) files](#)” on page A-7. At that point forward, it will also be 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.

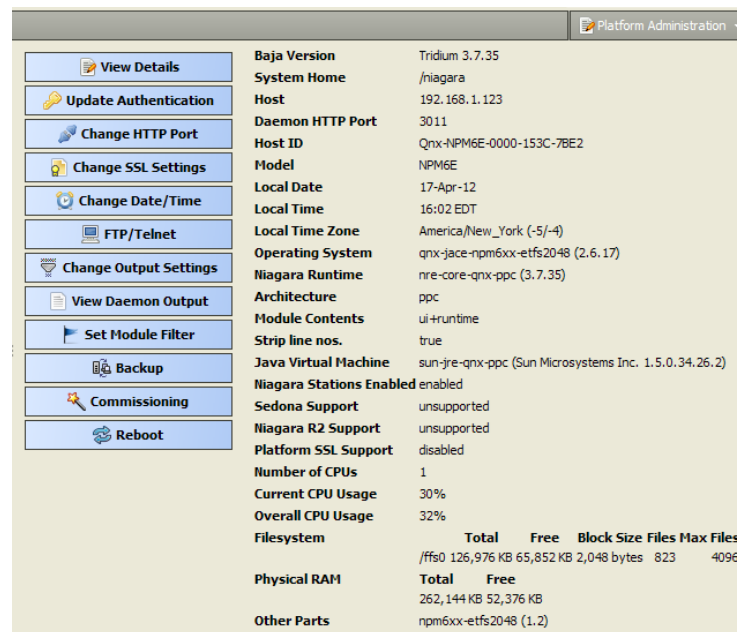
## Platform Administration

**Note:** Starting in AX-3.6, this view is key in any platform connection to a JACE-603 or JACE-645 controller, providing an option to either “Install Niagara R2” or “Uninstall Niagara R2”. See the Retrofit Board Niagara R2 Install and Startup Guide for details specific to these two controllers, including sections “About the Install Niagara R2 Wizard” and “Platform Administration view”.

Platform Administration is one of several [platform views](#). This view provides access to various platform daemon (and host) settings and summary information. As shown in [Figure 1-44](#), available functions appear as “buttons” on the left side, and summary information is listed in the right side. Typical use is when commissioning a new JACE, or when troubleshooting platform or host problems.

During a platform connection, upon first access to Platform Administration, a small delay occurs while downloading data about that platform’s installed modules. You may briefly see a “Loading Modules” dialog before the main view ([Figure 1-44](#)) appears.

**Figure 1-44** Platform Administration view



**Note:** Starting in AX-3.7, a “Change SSL Settings” button is enabled if the platform is licensed for SSL (feature “crypto”), and has the necessary modules installed. Otherwise, this button remains unavailable (dimmed).

The following sections provide more details:

- [Types of Platform Administration functions](#)
  - [View Details](#)
  - [Update Authentication](#)
  - [Change HTTP Port](#)
  - [Change SSL Settings](#)
  - [Change Date/Time](#)
  - [FTP/Telnet](#)
  - [Change Output Settings](#)
  - [View Daemon Output](#)
  - [Set Module Filter](#)
  - [Backup](#)
  - [Commissioning](#)
  - [Reboot](#)

**Note:** Some functions vary by platform, see “[About platform differences](#)” on page 1-6.





**Caution** *Reboot does exactly what it says, regardless of the opened platform. This is a drastic action to take on any Niagara host. See “Reboot” on page 1-55 for more details.*

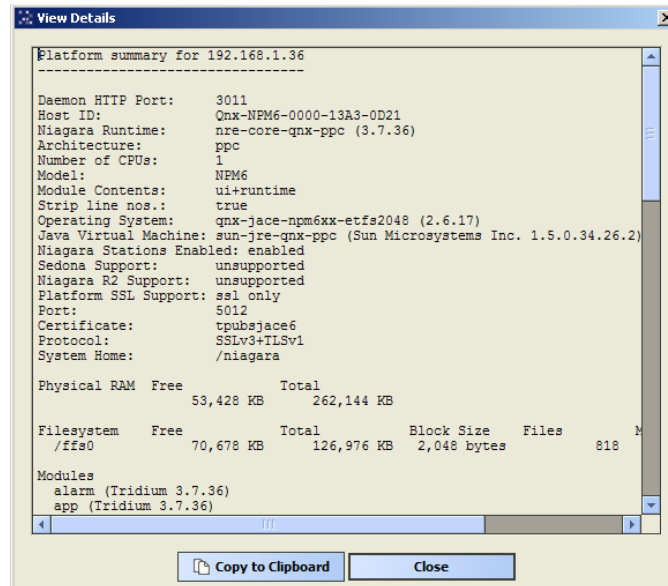
## Types of Platform Administration functions

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.
- [Update Authentication](#)  
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).
- [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 SSL Settings](#)  
(Available only if a AX-3.7 or later host that is licensed for SSL) For a dialog to enable/disable secure SSL/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.
- [FTP/Telnet](#)  
([QNX-based](#) only) For a dialog to enable/disable both FTP and Telnet access to the JACE, or change the default port number used by each one.  
*Note: FTP and Telnet pose security risks. We strongly recommend you keep each one 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 module content level 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](#)  
One way to launch the Commissioning Wizard, as an alternative to right-clicking 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. Included in the **View Details** window ([Figure 1-45](#)) is all installed modules, lexicons, licenses, and certificates, in addition to all stations (each station line lists configuration for autostart and autorestart, plus current status).

**Figure 1-45** View Details dialog in Platform Administration

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

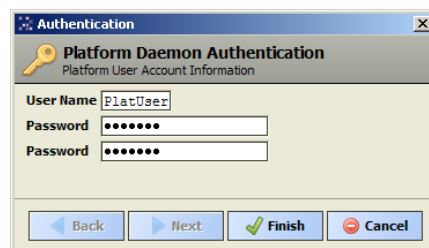
This selection from the main [Platform Administration](#) view lets you change that platform's authentication. This affects the login used to access the host's platform daemon. Depending on the type of platform currently opened ([QNX-based](#) or [Windows-based](#)), update authentication provides different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

- QNX-based platforms: [Digest platform authentication](#)  
*Note:* Also see *"Improvements to AX-3.8 digest authentication"* on page 1-43
- Win32-based platforms, either:
  - [Basic platform authentication](#) or
  - [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 in [Figure 1-46](#).

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

**Figure 1-46** Platform authentication dialog for digest authentication

### Caution

When commissioning a new JACE, always change platform credentials from the defaults! Note that the **Commissioning Wizard** includes a step for this ("Platform daemon authentication"). Do not omit this step. 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 AX-3.8 digest authentication"* on page 1-43.

The following sections provide more details:

- [User Name](#)
- [Password](#)
- [Usage Notes](#)

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

- 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 Win32-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:

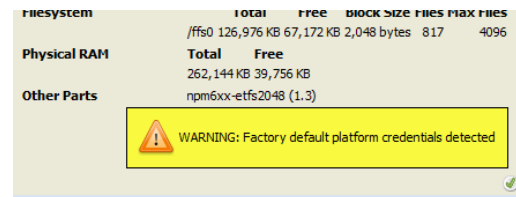
- Use both upper and lower case.
- Include numeric digits.
- Include special characters.
- Don’t use dictionary words.
- Don’t use company name.
- Don’t make the same as the user name.
- Don’t use common numbers like telephone, address, birthday, and so on.

**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. For related details, see the “Credentials manager” section in the *User Guide*.

## Improvements to AX-3.8 digest authentication

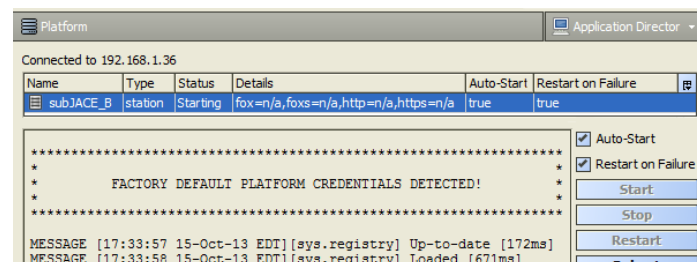
In AX-3.8, improvements were made in platform digest authentication and JACE security, as follows:

1. 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).
2. Platform digest credentials were *relocated* to a more secure location in the *registry* of the host platform (e.g. JACE).
3. 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 box WARNING  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.



As shown above, this warning precedes all other station output, and it repeats upon each station start until you *change* the platform credentials to non-defaults.

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](#)” on page 1-23.

### Basic platform authentication

A Win-32 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 Win32-based host, including a JACE-NXS, when you update platform authentication, a dialog asks you to select one of the two methods, as shown in [Figure 1-47](#).

**Figure 1-47** Authentication dialog for Win32 Niagara host



- If you select digest authentication, upon **Next** you go to the authentication dialog to set the single platform login account ([Figure 1-46](#) on page 1-42). There is no linkage between Windows OS users accounts and the platform administrator.
- If you select basic authentication, you go to a different dialog where you can assign one existing Windows user group to each of the two possible levels of platform access.

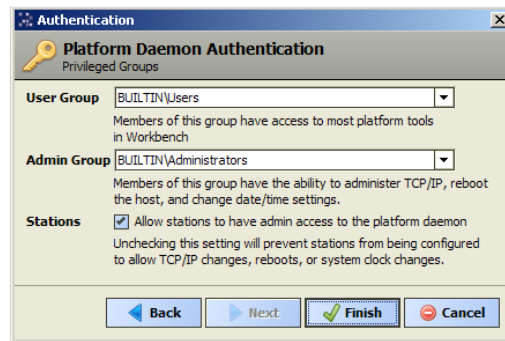
**Note:** If the host platform is currently configured for digest authentication, and you change to basic authentication, you first see a login dialog, as shown in [Figure 1-48](#). If already configured for basic authentication, you go directly to the basic authentication dialog ([Figure 1-49](#)).

**Figure 1-48** Login dialog when changing from digest authentication 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 ([Figure 1-49](#)).

**Figure 1-49** 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. For more details, see the next sections “[Station access](#)” and “[Levels of platform access](#)”.

**Station access** A “**Stations**” checkbox in the basic authentication dialog ([Figure 1-49](#)) allows you to disable *any* station user from changing TCP/IP settings, system time, or rebooting the host by accessing the station’s [PlatformServices](#).

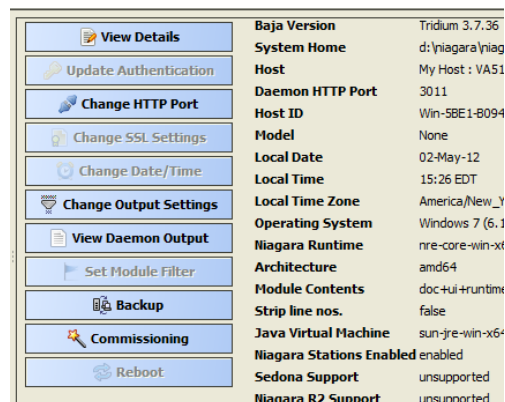
**Note:** In general, if a [Windows-based](#) JACE, you should leave the Stations checkbox enabled, as shown. However, if an Supervisor (PC) platform, you may wish to clear the Stations checkbox, particularly if the local IT department has host access concerns.

**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 in [Figure 1-50](#).

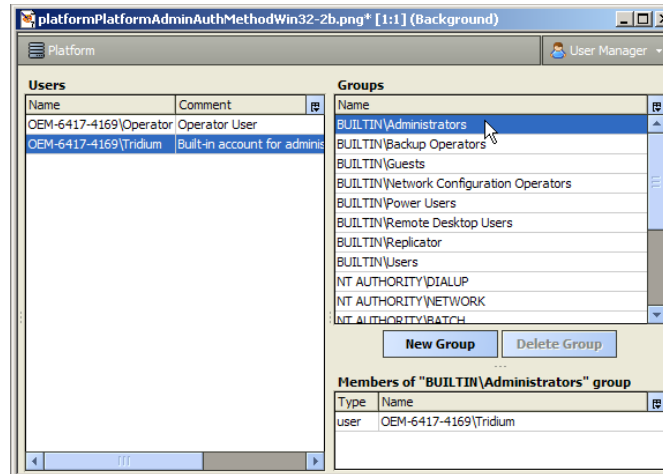
**Figure 1-50** Platform Administration view if user-level platform login



Platform access to a remote [Windows-based](#) host (JACE-NXT, JACE-NXS) 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 in [Figure 1-51](#).

**Figure 1-51** 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 Win32 installation (either Workbench/Supervisor installation or a factory-shipped JACE-NXS) are as follows:

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

## Change HTTP Port

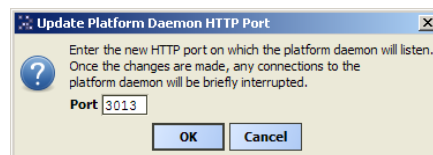
This selection from the main [Platform Administration](#) view lets you change the HTTP port monitored by the host’s platform daemon for “regular” (that is, non-SSL) platform client connections. By default, port 3011 is monitored for such connections. This differs from any port used for station (Fox) connections or for any secure (SSL) connection. For more details, see [“About a platform connection”](#) on page 1-3.



### Caution

If there is a firewall on the host (or its network), before changing this port make sure that it will allow traffic to the new port. For JACE-NXT (or JACE-NXS) related details, see the section “Review the Windows XP Firewall” in the JACE-NXT NiagaraAX Install and Startup Guide.

**Figure 1-52** 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 [Figure 1-52](#), 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 (note this does not affect the operation of any running station). The platform icon shows in the Nav tree with the new HTTP port number (:n) in parenthesis.

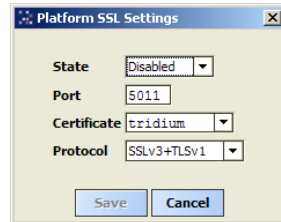
**Note:** Before closing the host (removing it from the Nav tree), carefully note the new (non-default) port number you entered. You must specify that port number whenever reopening a platform in a non-SSL connection. You can check this port number in a station running on the host, by opening its Config, [PlatformServices](#) property sheet.

## Change SSL Settings

(AX-3.7 or later platforms that use the Hotspot Java VM, and with the necessary modules installed). This selection from the [Platform Administration](#) view lets you configure for secure (SSL/TLS) platform connections, as well as change related secure platform connection (platformssl) parameters.

[Figure 1-53](#) shows the dialog with default values.

**Figure 1-53** Platform SSL Settings with default values (disabled)



Fields in this dialog are as follows:

- **State**  
Either Disabled, Enabled, or Ssl 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.
  - Ssl Only — Only secure platform connections are allowed. Behavior from an attempt to open a regular platform connection *differs* between AX-3.7 and AX-3.8, as follows:
    - With a AX-3.8 platform, any such attempt goes unresolved (errors out).
    - With a AX-3.7 platform, any such attempt is *automatically redirected* to the port servicing secure platform connections. (Note this not supported for the host *as a target in a tunneled* platform connection).

This state is reflected among the properties listed on the main Platform Administration view, as “Platform SSL Support” *state*.
- **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” (unsecure) platform connection.



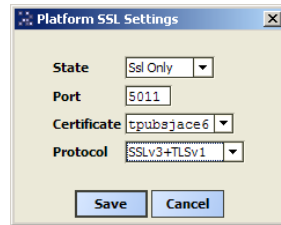
**Caution** Again, if there is a firewall on the host (or its network), before changing this port make sure that it will allow traffic to the new port.

- **Certificate**  
The “alias” for the server certificate in the platform’s “key store” to use for any platformssl connection. The default is the “tridium” self-signed certificate, which is automatically created when SSL is first loaded (by presence of certain modules and proper host licensing). 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 [“Certificate Management”](#) on page 1-19, or for complete details refer to document *NiagaraAX SSL Connectivity Guide*.
- **Protocol**  
Either SSLv3, TLSv1, or SSLv3+TLSv1, specifying the protocol for the secure platform connection.
  - SSLv3 — Only SSL version 3 (Secure Socket Layer) is used.
  - TLSv1 — Only TLS version 1 (Transport Layer Security) is used.
  - SSLv3+TLSv1 — Either TLS version 1 or SSL version 3 can be used for the platform connection. This is typically recommended, and is the default.

**Note:** If an AX-3.8 platform configured for FIPS 140, protocol is automatically set to TLSv1, and made read-only. For related FIPS 140 details, refer to the NiagaraAX FIPS 140 Configuration Guide.

[Figure 1-54](#) shows an example dialog for a JACE enabled for platform SSL (only).



**Figure 1-54** Example settings for a JACE enabled for SSL, with a signed certificate

In this example, the JACE uses a signed certificate with alias "tpubsjace6" (previously imported), with the port and protocol settings left at defaults.

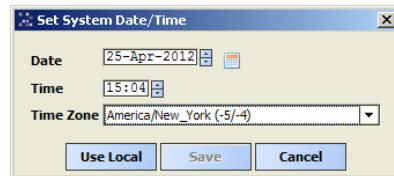
### Save notes on SSL platform settings

When you click **Save** after making any changes in Platform SSL Settings, those changes are immediately applied. Often this means your current platform connection *closes*, and then *reopens* in Workbench. For example if you change state from "Ssl Only" to "Disabled", your secure connection will close and then reopen as a regular (unsecure) platform connection. Or, if while secure connected, if you change Port from (default) 5011 to another port number, your reopened platformssl connection will use this new port, shown in parentheses (*nnnn*) to indicate a "non-default" port is being used.

**Note:** Before closing the host (removing it from the Nav tree), carefully note the new (non-default) secure platform port number you entered. In the future you must specify that port number whenever reopening this platform using the "Platform SSL Connection" method.

### Change Date/Time

This selection from the main [Platform Administration](#) view lets you change the date and time in the Niagara platform, as well as specify its time zone ([Figure 1-55](#)). The Save button becomes available after you change one or more fields in the dialog, or when you click [Use Local](#).

**Figure 1-55** Set System Date/Time dialog

Upon **Save**, any change is processed by that host's operating system.

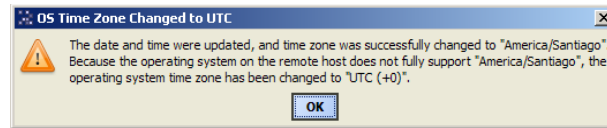
The three dialog fields are as follows:

- **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.
- **Time**  
Always displays in 24-hour format. Click in a hour or minute position to select, then click up/down controls, or click and type in numerals directly.
- **Time Zone**  
Provides a drop-down selection list of all available time zones in NiagaraAX. 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).  
For more time zone details, see the appendix ["Time Zones and NiagaraAX"](#) on page B-1.

**Note:** Although rare, in some cases a platform's OS may not support a particular time zone. This relates only to a few time zones with daylight savings switch-over. For example, America/Santiago (-4, -3) is not supported in [Windows-based](#) hosts because the start and end daylight savings dates are exact dates, versus a week number and weekday (e.g. 2nd Sunday).

Upon clicking **Save** for a time zone not fully supported by the host's OS, a popup dialog ([Figure 1-56](#)) explains that the platform's OS time zone was set to UTC (+0) during the update.

**Figure 1-56** OS time zone changed to UTC dialog



## Use Local

Typically, if your Workbench PC's current date/time setting are accurate, you click the "Use Local" 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 Niagara platforms, configure the [NtpPlatformService](#) in the station running on each platform, as appropriate.

## FTP/Telnet

This [Platform Administration](#) view selection is available for [QNX-based](#) JACE only. (For [Windows-based](#) platforms, you configure/enable FTP and Telnet within local Windows TCP/IP configuration.)

As factory-shipped, a QNX-based 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.



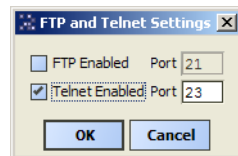
**Caution** *FTP and Telnet pose security risks. We strongly recommend you keep each one disabled, unless otherwise directed by Systems Engineering.*

Note that Telnet access to a QNX-based JACE 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.

[Figure 1-57](#) shows the FTP and Telnet dialog with Telnet enabled (both services showing "well-known" ports).

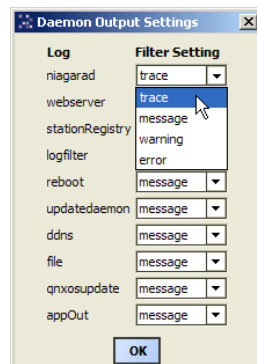
**Figure 1-57** FTP and Telnet dialog



## Change Output Settings

This selection from the main [Platform Administration](#) view lets you "tune" the amount and content of the platform daemon output ([View Daemon Output](#)). You can do this by changing the [log filter settings](#) of the various daemon processes ([Figure 1-58](#)).

**Figure 1-58** Daemon Output Settings dialog for QNX-based JACE



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 changes made in this dialog.
- **reboot** — Log for the reboot servlet, one of the servlets the platform daemon manages.
- **updatedaemon** — Log for handling Workbench requests for current platform daemon configuration, used mainly by 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, like “request for file xxx”, or “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”, or “waitpid when launching osupdate command failed”.
- **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.

### Log filter settings

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

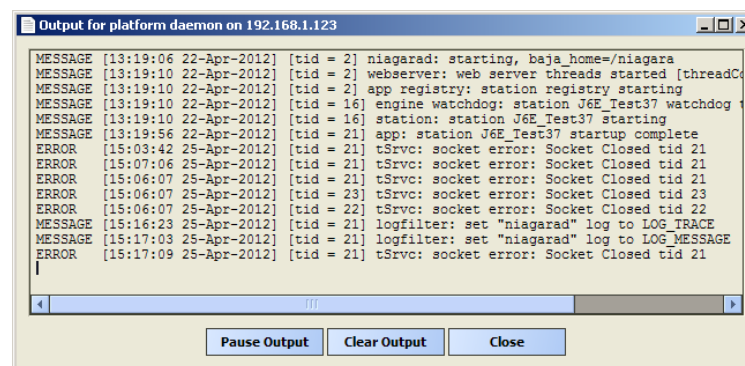
- **Trace**  
Returns all message activity (*verbose*). This includes all transactional messages, which may result in too many messages to be useful. Be careful using Trace!
- **Message**  
(Default) Returns informational “MESSAGE”s, plus all “ERROR” and “WARNING” types.
- **Warning**  
Returns only “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* ([Figure 1-59](#)), 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 1-59** Example Output for platform daemon



Depending on the [log filter settings](#) set in platform administration’s **Daemon Output Settings** dialog ([Change Output Settings](#)), 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.

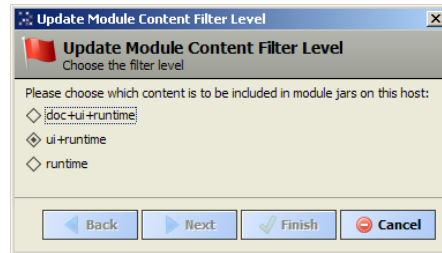
## Set Module Filter

This selection from the [Platform Administration](#) view lets you globally change the “module content level” of the connected platform. This affects how much file space is consumed by installed Niagara modules.

- For any [Windows-based](#) host (providing it has hard drive for file storage), you typically want the *full-est* possible content level, meaning including all documentation (doc+ui+runtime).
- For a [QNX-based](#) JACE, with more limited flash-based file storage, you may wish to change module content level. Selection produces the **Update Module Content Filter** dialog ([Figure 1-60](#)).

**Note:** Typically, you specify the module content level once during initial JACE commissioning, then never change it.

**Figure 1-60** Update Module Content Filter Level dialog



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 JACEs that serve PxPages directly to browser clients.
- runtime — Typically only for a [QNX-based](#) JACE *not* running Web Service (all PxPages served instead by a Supervisor).

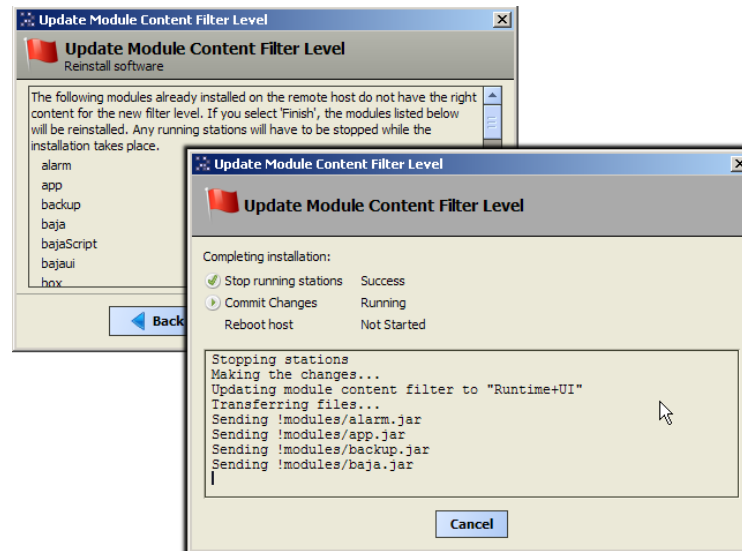
[Operations from a change in module content level](#) differ according to the “direction” of the change.

### Operations from a change in module content level

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

- If you *restrict* 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.  
Therefore, 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 in [Figure 1-61](#). Note that if a [QNX-based](#) platform, this also automatically results in a *reboot* of that host platform.

**Figure 1-61** Dialog messages resulting from increasing module content level



## Backup

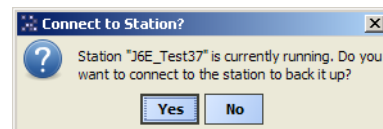
This selection from the [Platform Administration](#) view performs a complete backup of the connected JACE, saved as a *dist file* on your PC. The backup dist contains the entire station folder plus the specific NRE config used by that JACE platform, including license(s) and certificate(s). The dist 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](#)” on page 1-25.*

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 ([Figure 1-62](#)). 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 1-62** Backup with station running, station connection

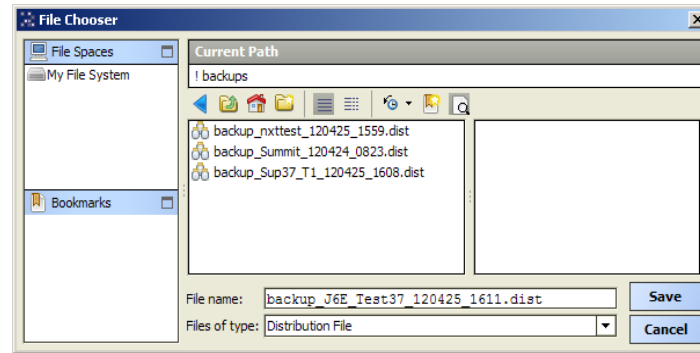


After station login and connection to the station (or if no station is running), the **File Chooser** appears ([Figure 1-63](#)) for you to navigate to the target location to save the backup dist file, and to rename if desired.



### Caution

*Control access to (keep secure) any saved backup dist files, as they contain sensitive information that could be valuable to any “system hacker.”*

**Figure 1-63** File Chooser to select target folder and dist file name

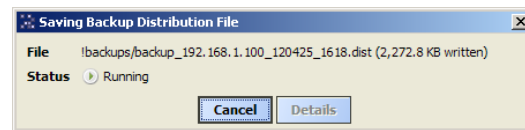
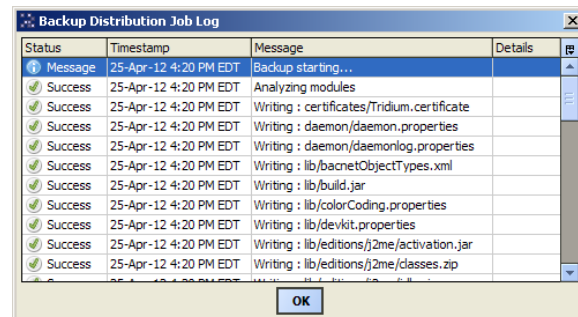
By default, the Backup function automatically creates (if not already present) a `backups` subdirectory under your Niagara build directory. The default *name* for a backup file uses a format of:

`backup_stationName_YYMMDD_HHMM.dist`

For example, “`backup_J6E_Test37_120425_1611.dist`” for a backup made of station “J6E\_Test37” on April 25, 2012 at 4:11 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 in [Figure 1-64](#). 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 ([Figure 1-65](#)).

**Figure 1-64** Backup from Platform Administration, no station running**Figure 1-65** 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 steps. Included views are similar to a subset of those listed in “[Types of platform views](#)” on page 1-5.

Typically, you use the Commissioning Wizard for the *initial* Niagara installation and startup of a JACE controller. For related details, see “About the Commissioning Wizard” in the *JACE NiagaraAX Install & Startup Guide*.

You also use the Commissioning Wizard to *upgrade* a JACE. For related details, see “[Upgrading a JACE](#)” on page 1-29.

**Note:** The Commissioning Wizard is intended for a remote JACE only. Please note that this button is unavailable whenever you are connected to your “localhost” (Supervisor) platform.



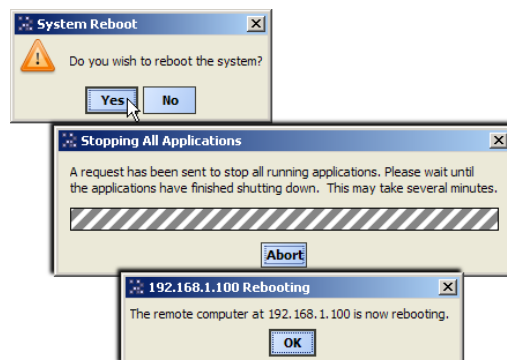
## Reboot

This selection from the [Platform Administration](#) view *reboots the host* of the connected platform. One confirmation dialog appears, after which the daemon attempts to stop any running station before issuing the final reboot ([Figure 1-66](#)).



**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 1-66** Reboot performs operating system 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](#)” on page 1-13).

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.

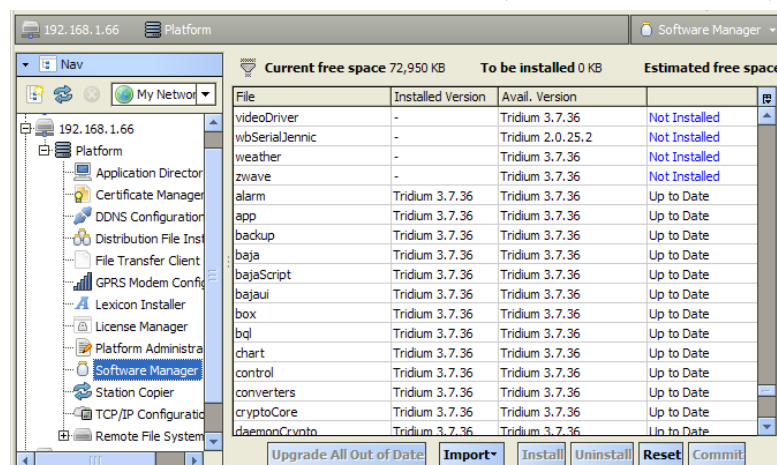
**Note:** *Reboot is intended for a remote JACE only. Please note that this button is unavailable whenever you are connected to your “localhost” (Supervisor) platform.*

## Software Manager

**Note:** *If platform connected to a (retrofit board) JACE-603 or JACE-645 controller configured for Niagara R2 (and not NiagaraAX), this view is not used. Instead, you use a “R2 Software Manager” tab under a different platform view, the “R2 Platform Tool”. For details, refer to the “R2 Software Manager” section in the Retrofit Board Niagara R2 Install & Startup Guide.*

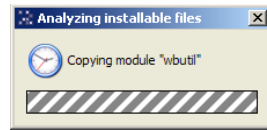
As shown in [Figure 1-67](#), the **Software Manager** is one of several [platform views](#). This view lets you install, uninstall, or simply review all NiagaraAX software modules installed in a remote JACE platform. By default, this view compares the platform’s modules against your “locally available” modules, meaning the most current NiagaraAX modules in the [software database](#) on your Workbench PC.

**Figure 1-67** Software Manager compares remotely installed modules to locally available



The *first time* you run the Software Manager, it copies modules in your Workbench's `! /modules` folder into the build-named subfolder in your “software database” (`! /sw`), for example `! /sw/3.7.105`. Note this can take several minutes, with a popup similar to the one below.

**Figure 1-68** Copying modules into your software database



**Note:** Copying also occurs whenever you “import” software into your Workbench’s 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 Niagara platform.

The following sections explain further:

- [Software Manager notes](#)
- [About your software database](#)
- [Default module listing and layout](#)
- [Filtering displayed software](#)
- [Software actions](#)

## Software Manager notes

Since AX-3.5, the “scope” of installables in the Software Manager has been *modules* only—whereas in Workbench AX-3.4 and earlier, all types of software was shown, including “dist” files for NRE, VM, OS, and so on. These and other [Software Manager](#) changes are summarized as follows:

- Only software modules are shown, versus all “installable parts” including dist files, etc. However, starting in AX-3.7, note that “standard lexicons” are distributed in NiagaraAX builds as *modules*, named (by convention) as `niagaraLexiconLc.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 in AX-3.7 and later Workbench 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”](#) on page 1-62.

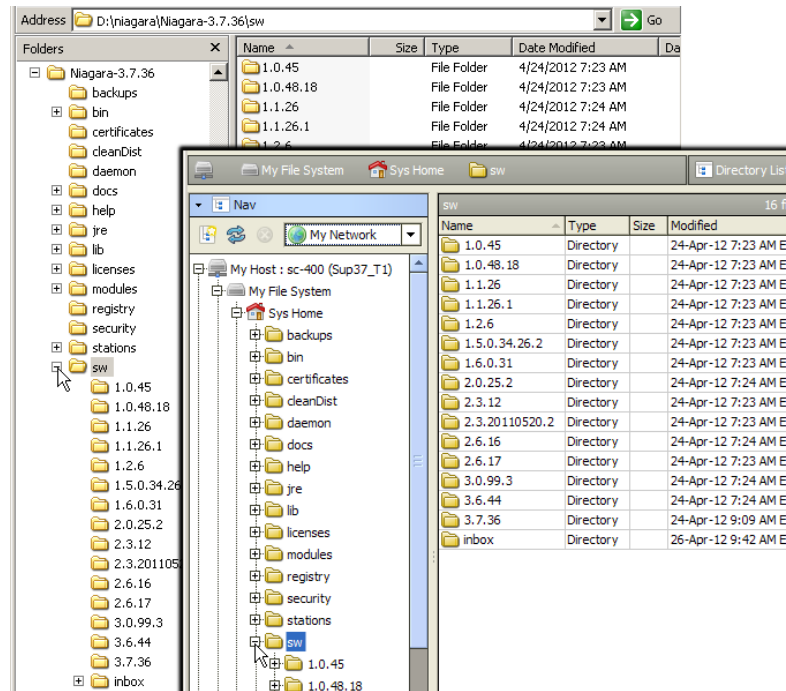
These changes are described and noted in other sections of this document, and are summarized here only to assist if you are already familiar with previous Workbench versions.

## About your software database

The software database for your Niagara Workbench is located under the Niagara build folder’s “sw” subdirectory. If Workbench was installed using the “use as an installation tool” option, this directory will contain a several subdirectories, each named using “version numbers”.

You can see your `sw` subdirectory structure using either Windows Explorer or the local “My File System” in Workbench, as shown in [Figure 1-69](#).

**Figure 1-69** 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 Figure 1-69 example of an AX-3.7 installation, this sw software database has many versioned subdirectories, a few of which are described in this example as follows:

- 1.1.26 — Reflects version of Sedona Framework-related modules. Contains 6 module files.
- 1.6.0.31 — Reflects version of Sun JVM for Win32/Win64 hosts. Contains 2 "sun-jre" dist files.
- 2.3.20110520.2 — Reflects version of IBM J9 JVM used on older QNX-based platforms. Contains an "ibm-j9" dist file.
- 2.6.17 — Reflects version of QNX operating system. Contains 9 different qnx dist files.
- 3.6.44 — Reflects a previous *Niagara release, by build number*. Contains 358 files, including module files as well as dist files for nre "core" and "config". Likely this was created in a *software import*.
- 3.7.36 — 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, contents of the build's modules directory (module .jars) are automatically copied here too.
- 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](#)" on page 1-60.

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](#)" on page 1-62.

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 the JACE's *out-of-date* modules at the *top* of the table, then *uninstalled* modules, and lastly *up-to-date* modules (sorted alphabetically); see Figure 1-70.

**Figure 1-70** Software Manager default listing out-of-date, then uninstalled modules

Platform		Software Manager	
Current free space 24,490 KB		To be installed 0 KB	Estimated free space
File	Installed Version	Avail. Version	
svg	Tridium 3.6.44	Tridium 3.7.36	Out of Date (Requires Commissioning)
wbutil	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)
aaphp	-	Tridium 3.7.36	Not Installed
aasup	-	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
bacnetbus	-	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. For related details, see “About module versions” in the *User Guide*.
- **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. See “[Upgrading a JACE](#)” on page 1-29.
  - **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. See “[Upgrading a JACE](#)” on page 1-29.
  - **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 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.

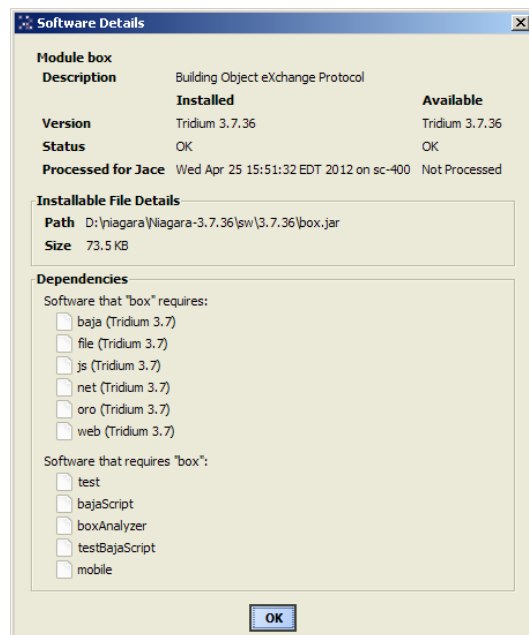
**Note:** “Intended” status values like “Install <version>” reflect un-committed actions made during your Software Manager session. Blue text is used to list these statuses.

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](#)” on page 1-59 for more details.

## Software Details

From the [Software Manager](#), double-click any module to see a popup dialog with details ([Figure 1-71](#)).

**Figure 1-71** Software Details dialog from Software Manager



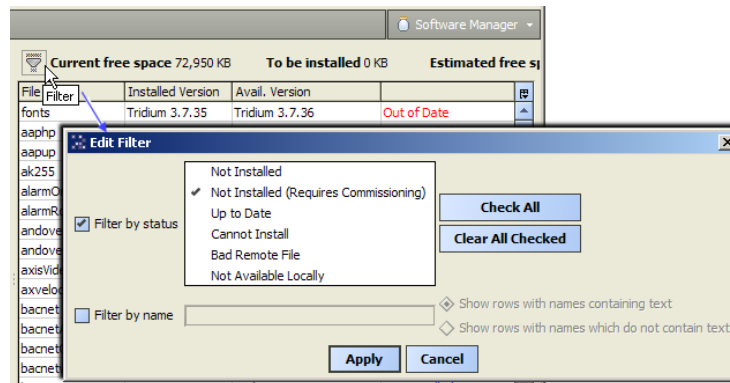
Details include a brief module description, comparisons between installed and available module, module file and size, and whatever module *dependencies* exist, by part names. Dependencies are listed for both cases: what software is required *by* this module, plus software dependent *on* this module.

**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 ([Figure 1-72](#)), in which you select items for listing, thereby filtering undesired items.

**Figure 1-72** 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.

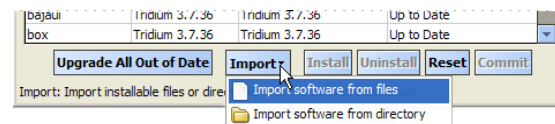
This feature can be useful to filter many modules with common name characters, for example “lon” or “doc” part-named modules.

## Software Import

As shown in [Figure 1-73](#), 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 the next section, “Import vs. copy into modules”.*

**Figure 1-73** 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. See [“Right-click option to install earlier version”](#) on page 1-62.

## Import vs. copy into modules

When receiving updated or new *module* jar files (say sent from Systems Engineering or downloaded from Niagara Central), 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.



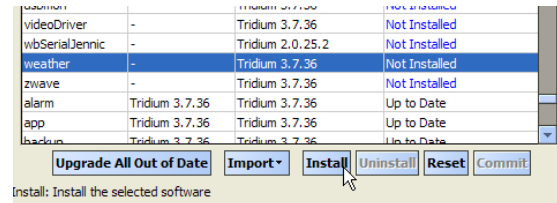
2. Copy into your `! /sw/ inbox` directory (or, use the equivalent “[Software Import](#)” feature in the [Software Manager](#)). In this case, the module(s) are *not* used in your Workbench environment, but *are* available in your software database for installation in remote platforms.

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 in [Figure 1-74](#). Action buttons become enabled when you have one or more items selected.

**Figure 1-74** 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 host must be rebooted and/or station stopped. Then, after the software operation completes, the host is rebooted (if a [QNX-based JACE](#)), or if a [Windows-based JACE](#), its station is restarted.

**Note:** 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.

The following action buttons are explained in further detail:

- [Upgrade All Out of Date](#)
- [Install](#)
- [Uninstall](#)
- [Re-Install, Upgrade, Downgrade](#)
- [Commit and Reset](#)

**Note:** Also see “[Right-click option to install earlier version](#)” on page 1-62.

### 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.

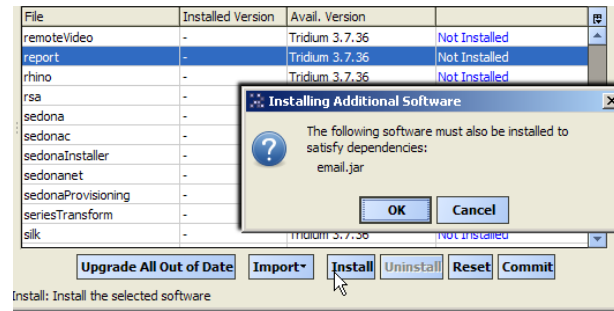
### Install

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 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 in [Figure 1-75](#). After you click **OK** from this dialog, the additional modules are flagged, the status of all affected modules changes to “Install <version>,” and the button changes to **Cancel Install**.



**Figure 1-75** Installing Additional Software dialog

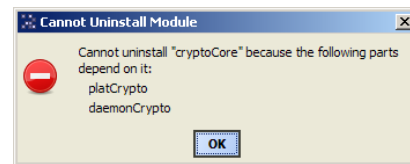


## Uninstall

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 in [Figure 1-76](#). You can then decide if you want to reflag another uninstall, selecting also all modules that are dependent.

**Figure 1-76** 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 a 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

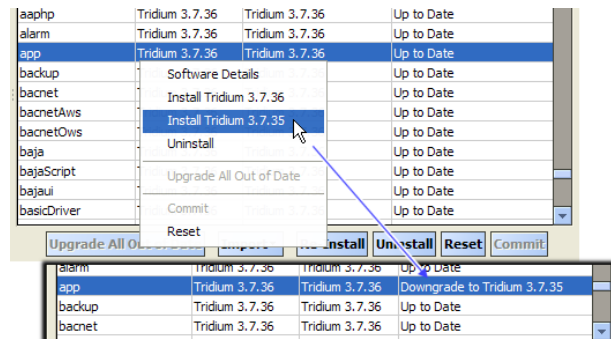
In the Software Manager, when you have one or more *pending* actions in place on software items, the **Commit** button is available. This is how you initiate the software action.

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 1-77** 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 shown (Figure 1-77). Note if a “downgrade”, a host reboot/station restart will result after you commit. For related details, see “[About your software database](#)” on page 1-56.

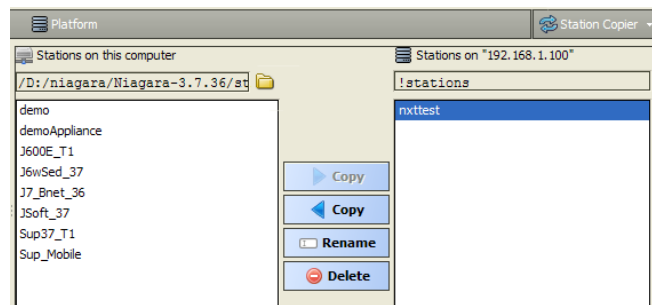
## Station Copier

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.

**Note:** *If platform connected to a (retrofit board) JACE-603 or JACE-645 controller configured for Niagara R2 (and not NiagaraAX), this view is not used. Instead, you use a “R2 Station Copier” tab under a different platform view, the “R2 Platform Tool”. For details, refer to the “R2 Station Copier” section in the Retrofit Board Niagara R2 Install & Startup Guide.*

As always, this platform view is not seen when opening a *local* platform connection at your Supervisor computer—usage has always been intended for remote NiagaraAX JACE hosts only.

**Figure 1-78** Station Copier view



As shown in Figure 1-78, this view is split into two main areas:

- Local stations on your Workbench computer (left side)
- Remote station on the opened JACE platform (right side)

**Note:** *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.*

The following sections provide more details:

- [Security update 1 changes to Station Copier usage](#)
- [Station copy direction](#)
- [Station Copier dependencies check](#) (if familiar with the Station Copier prior to Workbench AX-3.5)
- [Station Transfer Wizard](#)
- [Renaming stations](#)
- [Deleting stations](#)

### 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

**Note:** 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 a 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.

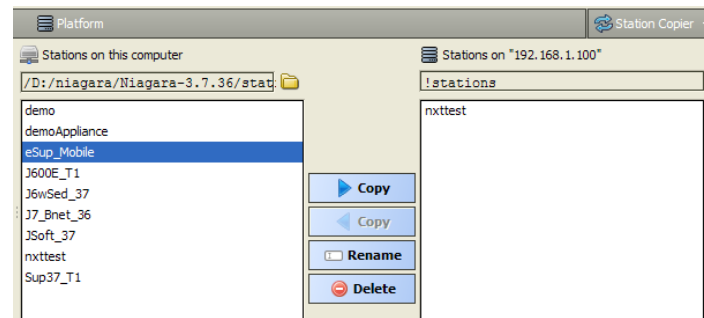
However, in the AX-3.7u1 update release, because of the different station password storage methods, the following scenarios typically require you to perform some *offline editing* of the saved station file (`config.bog`) first, that is *before* using the Station Copier to install/copy it to other different 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”.
- When a station saved with Workbench AX-3.7u1 or later is to be installed on *any* host running a “pre-update” release, e.g. AX-3.7 or AX-3.6. Again, in this case the 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).

## 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 [Figure 1-79](#).

**Figure 1-79** Copy direction by station side selection



**Note:** In AX-3.7u1, in some cases a copy local-to-remote (install) may first require offline editing of the saved config.bog file, or alternatively some subsequent password editing after the install. (Note this does not apply if the station's host is running AX-3.8.) See ["Security update 1 changes to Station Copier usage"](#) on page 1-63 for more details.

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 a **Copy** button, a [Station Transfer Wizard](#) guides you the rest of the way.

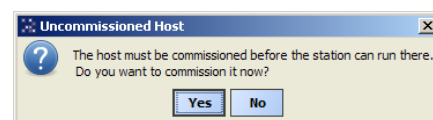
**Note:** Dependency checks may prevent the installation of a selected station—where the Station Transfer Wizard does not start. See the next section ["Station Copier dependencies check"](#) for more details.

## Station Copier dependencies check

The [Station Copier](#) checks whenever *installing* a station, *and* the target JACE platform does not already have all modules installed that are required by that 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 1-80](#).

**Figure 1-80** 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.

Two different examples of when this might occur:

- You have a platform connection to a JACE running AX-3.6. The station you are trying to copy (install) has been engineered with "apps", using the app module introduced in AX-3.7. The app module has 3.7 dependencies on *baja* and other modules.  
In this case, providing you bought a 3.7 license for this JACE, you could click **Yes** to start the Commissioning Wizard, then *upgrade* the JACE—selecting to also install the same station.
- You are trying to install a station in a new, uncommissioned "out-of-the-box" JACE.  
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 and 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"](#) on page 1-68.

## 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 select a different station 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 wizard's **Transferring station** dialog. When complete, you click **Close** in that dialog to exit the wizard.

The following sections describe all possible steps in the Station Transfer Wizard:

- [Name step](#)
- [Delete step](#)
- [Content step](#)
- [Disposition step](#)
- [Station settings step](#)
- [Details step](#)
- [Modules step](#)
- [Stop station step](#)
- [Review step](#)

**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 the 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 1-81** 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.

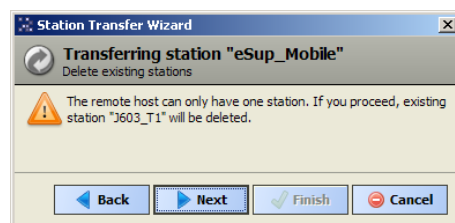
### Delete step

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

- No existing station exists.
- 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 ([Figure 1-82](#)).

**Figure 1-82** 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.

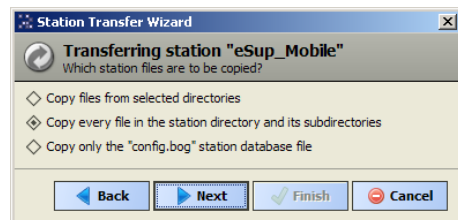
The next step is the [content step](#).

## 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 ([Figure 1-83](#)).

**Figure 1-83** 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.

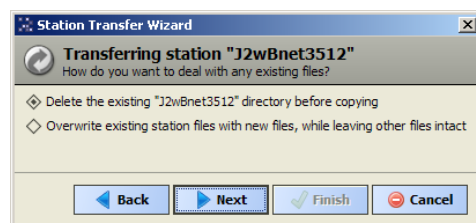
The next step is the [disposition step](#).

## 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 ([Figure 1-84](#)).

**Figure 1-84** Station Transfer Wizard dialog, disposition step



The two possible selections are:

- Delete existing station directory before starting the copy
- 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.

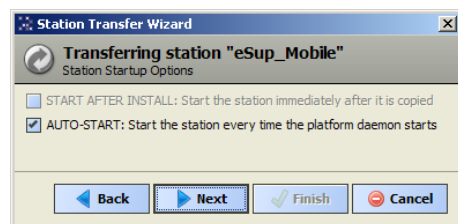
The next step is the [station settings step](#).

## Station settings step

**Note:** This [wizard](#) step is skipped for any station backup.

This step specifies the station's Auto-Start setting ([Figure 1-85](#)).

**Figure 1-85** 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.

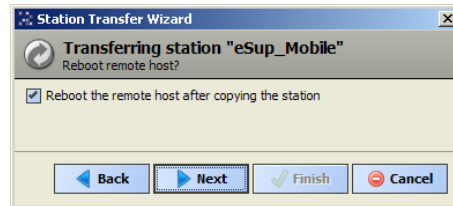


Auto-start is one of two station settings for any station, as specified in the [Application Director](#) view by using [start checkboxes](#).

Depending on the target platform, default settings in this step vary:

- If a Win-32 based JACE (JACE-NXT, SoftJACE)
  - START AFTER INSTALL: Enabled.
  - AUTO-START: Enabled.
- If a QNX-based JACE.
  - START AFTER INSTALL: shown cleared and unavailable (dimmed). A QNX-based JACE requires a reboot for the station to start. After clicking Next, a separate dialog asks if the remote host should be rebooted ([Figure 1-86](#)).
  - AUTO-START: Enabled.

**Figure 1-86** Reboot remote host after copying the station



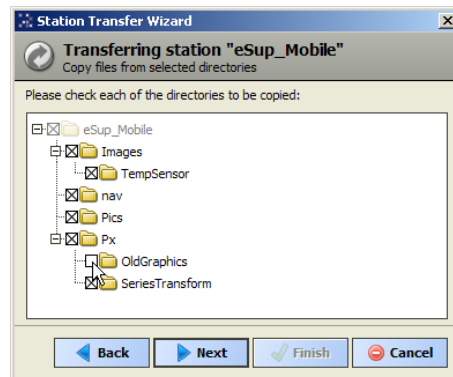
Typically, you enable all settings and continue to the next step, either the [details step](#) or [modules step](#).

### Details step

**Note:** This [wizard](#) step is skipped unless you selected “copy selected directories” in the [content step](#).

This step provides a tree ([Figure 1-87](#)) 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).

**Figure 1-87** Station Transfer Wizard dialog, details step



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

### Modules step

**Note:** This [wizard](#) step is skipped if a station backup, or if all modules required by the station being installed are already in the remote JACE platform. In this case, you see either the [stop station step](#) or [review step](#) instead.

This step occurs if the target JACE 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 JACE platform. The Station Copier looks for versions of those missing modules in your Workbench software database that can be installed without re-commissioning the JACE, 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 up-

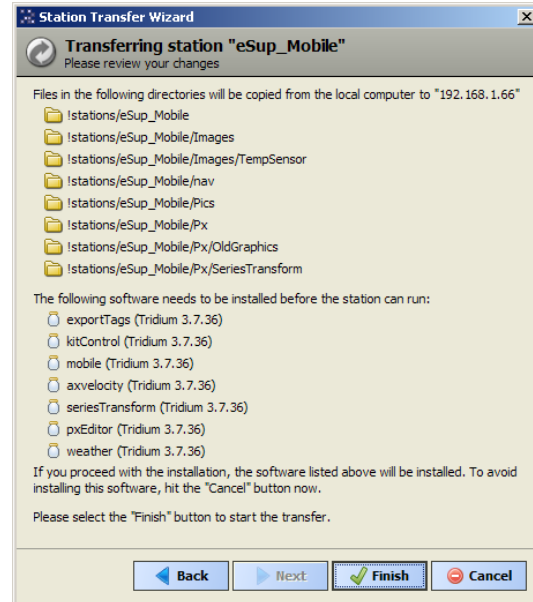


dates appear in a [Transferring station](#) dialog.

- **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 JACE, including the installation of a station.

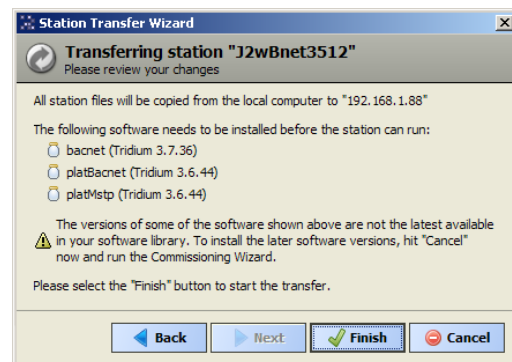
**Station can be installed with most current modules** If all missing modules can be installed using the most current versions, they list without any warning, as shown in [Figure 1-88](#).

**Figure 1-88** 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. An example of this dialog is shown in [Figure 1-89](#).

**Figure 1-89** Station install example, one or more missing modules not current versions



In the example above, a station using the BACnet driver is being installed in AX-3.6 JACE which does not already have BACnet-related modules installed (bacnet, platBacnet, platMstp). Here, the Workbench AX-3.7 *software database* includes some 3.6 modules—likely “imported” using the Software Manager.

For related details, see Software Manager topics [“About your software database”](#) on page 1-56 and [“Software Import”](#) on page 1-60. Also see [“Upgrading a JACE”](#) on page 1-29.

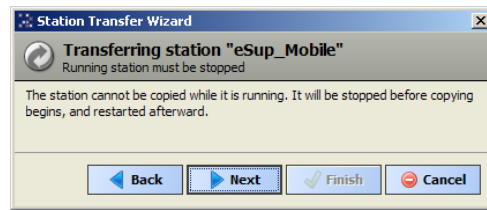
## Stop station step

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 (Figure 1-90).

**Figure 1-90** Station Transfer Wizard dialog, stop station step



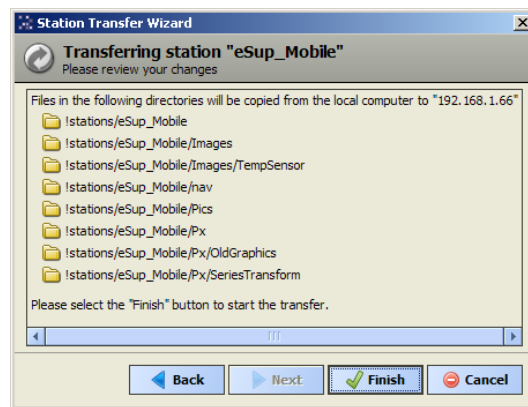
Click Next to go to the review step.

## 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. See Figure 1-88 on page 1-69 for an example.

This step provides a summary of choices from previous steps, and a **Finish** button to begin the station copy process. As shown in Figure 1-91, if you selected only specific station subdirectories to copy (from the details step), they are listed.

**Figure 1-91** Station Transfer Wizard dialog, review step

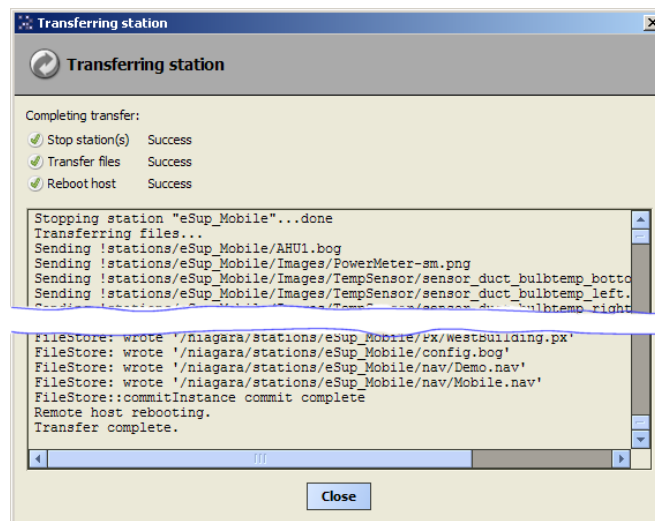


If needed, click **Back** to make changes, or click **Finish** to begin the copy process and observe progress in the Transferring station dialog.

## Transferring station

After clicking **Finish** in the wizard's modules step or review step, the station copy process begins and updates appear in this dialog, as shown in Figure 1-92.

**Figure 1-92** Station Transfer Wizard, Transferring station (copy) process



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

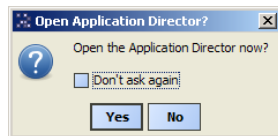
- If installing a station (copy local-to-remote):
  - Stop all stations — whenever modules require to be installed.
  - Stop one station — any JACE where same station is being reinstalled.
  - Delete station(s) — if you chose to delete station in the [disposition step](#), or if a station needs to be 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 required to be stopped (module installation), or if you chose to start the station in the [station settings step](#). Note that if a [QNX-based](#) platform, the process will finish with a host reboot.
- If backing up a station (copy remote-to-local):
  - Save station — whenever remote station is currently running.
  - Transfer files — includes station and module files (actual copy portion).

**Note:** A popup explaining that the existing station must be stopped 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 ([Figure 1-93](#)) asking if you wish to switch to the [Application Director](#) platform view. Because it is a good idea to observe a station’s output upon first startup, you typically select **Yes**.

**Figure 1-93** Switch to Application Director popup



**Note:** To always 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). A Rename dialog ([Figure 1-94](#)) appears when you select a station and click **Rename**.

**Figure 1-94** Rename station dialog

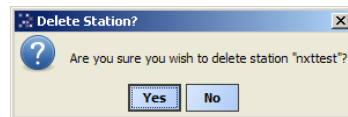


**Note:** Be careful when renaming stations, as there is no undo. Furthermore, please note the following:

- Any *running* station that is renamed must first be *stopped*—a confirmation dialog will inform you of this after you enter the new station name and click OK. In the case of a QNX-based JACE, this also results in a host *reboot*.
- 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). A confirmation dialog ([Figure 1-95](#)) appears when you select a station and click **Delete**.

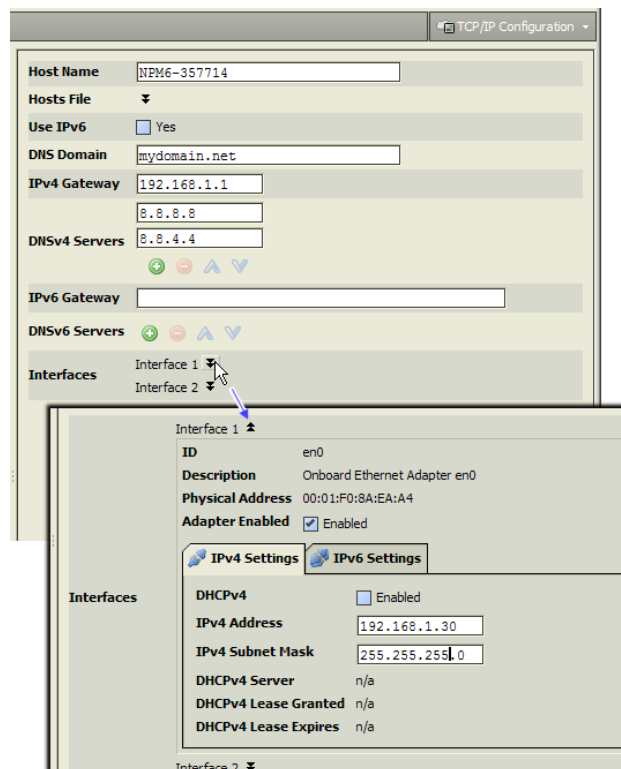
**Figure 1-95** Confirm delete station dialog

**Note:** Be careful when deleting stations, as there is no undo. Furthermore, please 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 remote 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.

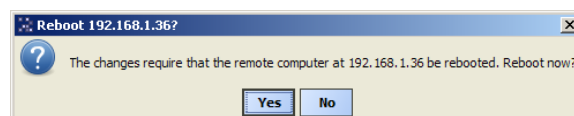
## TCP/IP Configuration

TCP/IP Configuration is one of several [platform views](#). Typically, you use it to initially configure a remote JACE's TCP/IP settings. Some earlier JACE models have only a single Ethernet port (interface) for configuring TCP/IP, but if equipped with two Ethernet ports, both interfaces are accessible ([Figure 1-96](#)).

**Figure 1-96** TCP/IP Configuration shows available Ethernet interface(s)

If connected to your localhost (PC) platform, this view also provides access to your PC's Windows TCP/IP settings. However, you typically use the Windows Control Panel for making these changes.

**Note:** Regardless of platform connection, if you make any changes in this view, a reboot of that platform is necessary before those changes are effective. A popup dialog appears when you click **Save** ([Figure 1-97](#)), asking if you wish the platform daemon to reboot that host now. When making multiple changes, wait until entering all changes before you click **Save, Yes**.

**Figure 1-97** Reboot confirmation dialog from TCP/IP configuration change

More details on the [TCP/IP Configuration](#) view are in the following sections:

- [TCP/IP changes in AX-3.6](#) (IPv6 support for “Hotspot JACE” vs. “J9 JACE”)
- [TCP/IP Host fields](#)
- [TCP/IP DNS fields](#) (host-level for QNX-based platforms only)
- [TCP/IP Interface fields](#)

## TCP/IP changes in AX-3.6

### IPv6 support

Starting in AX-3.6, IPv6 support is available in some of the newer QNX-based JACE platforms, providing that they are running AX-3.6 or later. Starting in AX-3.6, these same JACE platforms also use the **Sun Hotspot JVM** (Java virtual machine), versus the IBM J9 JVM used in AX-3.5 and earlier.

Thus, there are now two “subgroups” of QNX-based controllers, sometimes referred to as follows:

- **Hotspot JACE**  
Includes the newest NPM3 and NPM6E-based models (JACE-3E, JACE-6E, JACE-603, JACE-645), and also the JACE-7 (JACE-700) and JACE-6 and JACE-602 Express (both NPM6-based) controllers, providing they are running AX-3.6 or later. These models support IPv6 in addition to IPv4.
- **J9 JACE**  
Includes *any* QNX-based JACE controller running AX-3.5 or earlier, including some of the models above, as well as all other JACE controllers (regardless of AX-level), such as the JACE-2 and JACE-202 Express (both NPM2-based). None of these controllers support IPv6.

The two QNX-based JACE sub groups are sometimes referred to as “Hotspot JACE” or “J9 JACE” in this document. Also see [“Sun Hotspot JVM or IBM J9 JVM”](#) on page 1-7 for related details.

Related to this, AX-3.5 and later Workbench began support for “IPv6”, where for any selected Ethernet interface, there are now separate tabs for (standard) **IPv4 Settings** as well as **IPv6 Settings**. Note that TCP/IP property field descriptors maintain an “IPv4”, “v4”, “IPv6”, or “v6” qualifier, to clarify separate properties.

For example: IPv4 Address (IP Address)

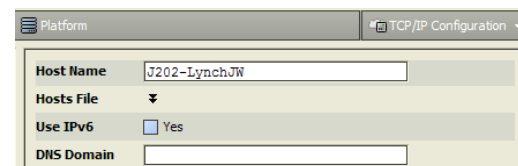
### JACE WiFi

Also starting in AX-3.6, a wireless 802.11b/g (WiFi) option became available for the JACE-700 controller, effectively adding a *third available interface* for its platform TCP/IP configuration. For related details, see the Engineering Notes II document *“NiagaraAX JACE WiFi Option”*.

## TCP/IP Host fields

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

**Figure 1-98** 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. By default, hostname for any [QNX-based](#) platform is “localhost”.  
**Note:** *On Win32 systems, this is also the computer name. Be aware that changing it will have important consequences with regards to the computer’s identification in its workgroup or domain!*  
If using hostnames in your network, you should assign each Niagara platform 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 and edit, you click on the expand control to see all entries.
  - 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), NiagaraAX (platform daemon and station) respond to IPv6 requests, that is, creates IPv6 server sockets (daemon) and IPv6 fox multicast sockets. This property is applicable only to the following hosts:
  - AX-3.6 and later “Hotspot JACE”, that is JACE-7 (JACE-700) and JACE-6 and JACE-602 Express running AX-3.6 or later. See “IPv6 support” on page 1-73 for background details.
  - AX-3.4 and later Windows-based hosts, providing that the host platform has the Microsoft TCP/IP version 6 protocol installed.
  - AX-3.4 and later [Linux-based Supervisor](#).

### TCP/IP DNS fields

If connected to a [QNX-based](#) JACE platform, the DNS and gateway settings are also “host-level” parameters in the TCP/IP Configuration view, as shown in [Figure 1-99](#).

**Note:** If a [Windows-based](#) host, DNS and gateway settings are available under each Interface section. See “TCP/IP Interface fields” on page 1-74.

**Figure 1-99** Host-level fields for QNX-based platform includes DNS and gateway

The screenshot shows the TCP/IP Configuration window. The 'Host Name' field contains 'NPM6-357714'. The 'Hosts File' is a dropdown menu. The 'Use IPv6' checkbox is checked. The 'DNS Domain' field contains 'mydomain.net'. The 'IPv4 Gateway' field contains '192.168.1.1'. The 'DNSv4 Servers' field contains '8.8.8.8' and '8.8.4.4'. The 'IPv6 Gateway' field is empty. The 'DNSv6 Servers' field is empty. There are callouts: one pointing to the 'IPv4 Gateway' and 'DNSv4 Servers' fields with the text 'All QNX-based JACEs', and another pointing to the 'IPv6 Gateway' and 'DNSv6 Servers' fields with the text 'Hotspot JACEs only (AX-3.6 and later)'. At the bottom, there are 'Interface 1' and 'Interface 2' dropdown menus.

The available fields for QNX-based JACE platforms are as follows:

- **DNS Domain**  
The TCP/IP Domain Name System (DNS) domain this host belongs to, if used.
- **IPv4 Gateway**  
The IP address for the device 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 for 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**  
Available only if a AX-3.6 and later “Hotspot JACE”. The IPv6 address for the device that forwards packets to other IPv6 networks or subnets.
- **DNSv6 Servers**  
Available only if a AX-3.6 and later “Hotspot JACE”. 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).

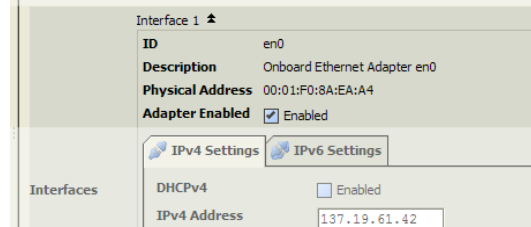
### TCP/IP Interface fields

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

Some JACE models have two Ethernet ports: LAN1 and LAN2, which are **Interface 1** (en0) and **Interface 2** (en1). Other JACE models may have a single Ethernet LAN port, **Interface 1** (en0).

**Note:** A JACE-700 can have an optional “WiFi” adapter installed. It appears in the **TCP/IP Configuration** view as yet a third **Interface 3** (bc0).

**Figure 1-100** TCP/IP Interface fields, top properties



As shown in [Figure 1-100](#), each **Interface** has the following properties at the top:

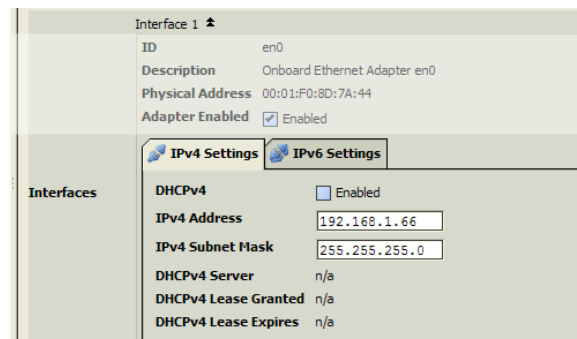
- **ID**  
A read-only OS identifier for the hardware interface, such as “en0” if a [QNX-based](#) platform, or if a [Windows-based](#) platform, either a 128-bit GUID (globally unique identifier) or a Windows network connection name, such as “Local Area Connection 2”.
- **Description**  
A read-only text string such as “Onboard Ethernet Adapter en0” for a QNX-based host, or “Intel(R) PRO/100 VE Network Connection” for a Win32-based host, describing a NIC model.
- **Physical Address**  
(Value is available only if platform is running AX-3.5 or later) The unique 48-bit MAC address of the Ethernet adapter, in six two-hexadecimal digits. For example, for a JACE: 00:01:F0:80:13:E6
- **Adapter Enabled**  
Checkbox to specify whether the Ethernet port is usable. On Windows-based hosts, this is read-only.

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

- [IPv4 Settings](#)
- [IPv6 Settings](#)

## IPv4 Settings

**Figure 1-101** IPv4 tab for Interface of QNX-based JACE, 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 QNX-based JACE 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.  
**Note:** 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 JACE 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 JACE may become unreachable over the network.

- **DNS Domain**  
([Windows-based](#) hosts only) The TCP/IP Domain Name System (DNS) domain this host belongs to, if used.

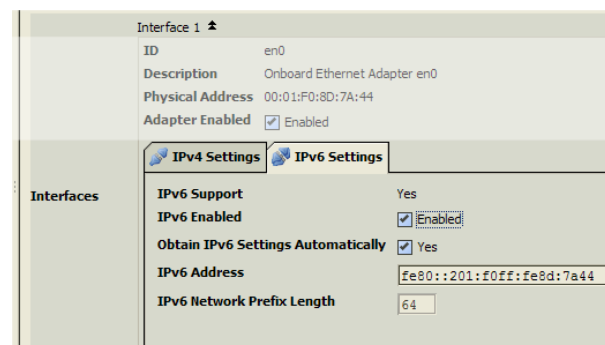


- **IPv4 Address**  
The “static” IP address for this host, unique on your network.  
**Note:** *Be careful to understand the following:*
  - If enabling both LAN ports, note that the LAN1 IP address and LAN2 IP address must be on *different subnets*, otherwise the ports will not function correctly.  
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 JACE controller *does not* provide IP routing or bridging operation between different Interfaces (LAN ports, GPRS, dialup, WiFi).
- **IPv4 Gateway**  
(Windows-based 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 DCHP 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 DCHP is enabled. Shows a read-only timestamp of when the DHCP lease started.
- **DHCPv4 Lease Expires**  
Applies only if DCHP is enabled. Shows a read-only timestamp of when the DHCP lease will expire, and will need renewal.
- **DNSv4 Servers (DNS Servers)**  
(Windows-based 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

- IPv6 support is provided in most QNX-based JACE models running AX-3.6 or later (“Hotspot JACE”), but is not available in any “J9 JACE” or any QNX-based JACE running AX-3.5 or earlier. See “IPv6 support” on page 1-73 for background.
- Windows-based hosts have IPv6 properties that are *read-only*. Any adjustments, if necessary, must be made directly from the Windows OS interface, that is the Control Panel.

**Figure 1-102** IPv6 tab for Interface of (Hotspot) QNX-based JACE, 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.
  - Some QNX-based JACEs running AX-3.6 or later (with “Hotspot” JVM) support IPv6. For those hosts that do not, the rest of the properties below do not appear.
  - Windows-based OS hosts typically provide IPv6 support, as does Linux.
- **IPv6 Enabled**  
Checkbox for Enabled. Read-write for AX-3.6 or later “Hotspot JACE” only, where default is disabled (cleared). If a Windows-based host this is read-only, and indicates whether the host is configured with the IPv6 protocol.

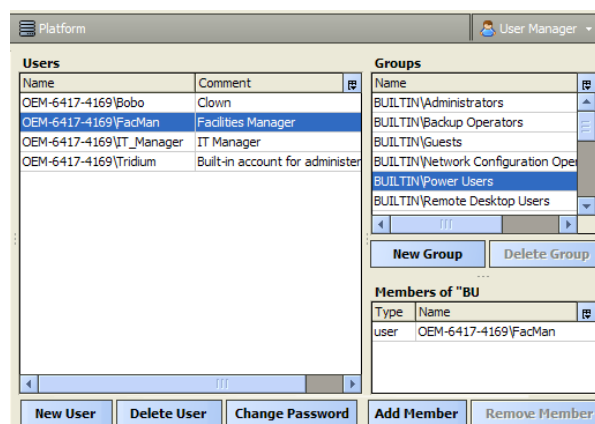
- **Obtain IPv6 Settings Automatically**  
Checkbox for Enabled (default). Changeable for a AX-3.6 or later “Hotspot JACE” only. Provides for “auto-configuration” of IPv6 address, if acceptable. If enabled, the next two properties are read-only. If cleared, the two properties below must be entered manually.
- **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-based](#) 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 remote [Windows-based](#) JACE. 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](#)” on page 1-45 for related details.

**Figure 1-103** User Manager for remote Win32-based host



As shown in [Figure 1-103](#) above, 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 popup 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](#)
- [Groups management](#)

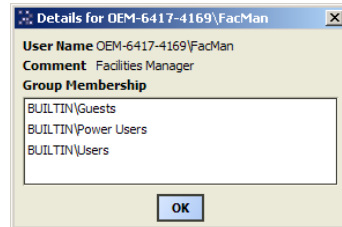
### Users management

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 in the [User Manager](#) for a Details dialog, as shown in [Figure 1-104](#).

**Figure 1-104** Details dialog for Windows user

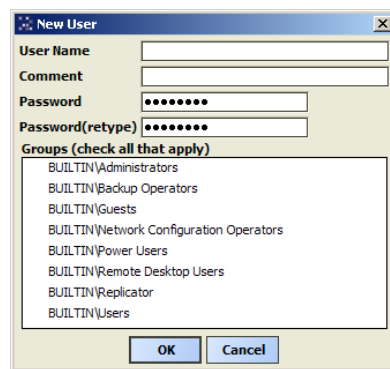


This displays the user's account name, comment, and group memberships (including domain groups).

## Add user

Click the **New User** button in the [User Manager](#) for a **New User** dialog, as shown in [Figure 1-105](#).

**Figure 1-105** New User dialog



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

In the [User Manager](#), 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 in [Figure 1-106](#).

**Figure 1-106** Confirm Delete dialog

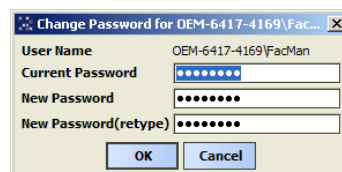


When you click **OK**, the selected user(s) is deleted and removed from the user table.

## Change user password

In the [User Manager](#), click to select a user, then click the **Change Password** button for a popup dialog ([Figure 1-107](#)).

**Figure 1-107** Change Password dialog



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

In the [User Manager](#), 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

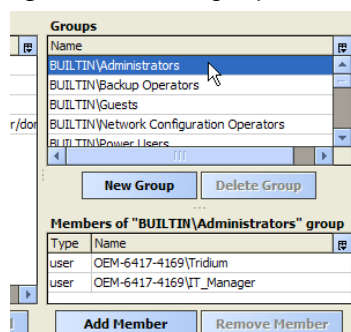
In the groups side of the [User Manager](#), click in the groups table, membership table, and buttons below to perform various Windows group management tasks. You can [review](#), [add](#), and [delete](#) groups, and in any group, you can [add](#) or [remove](#) members.

**Note:** Starting in AX-3.5, Workbench and the NiagaraAX platform daemon changed for a Windows host, such that 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 ([Figure 1-108](#)).

**Figure 1-108** Select group to see membership

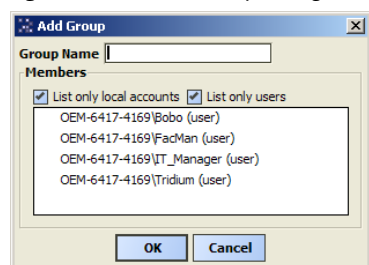


All users for the selected group are shown.

## Add group

Click the **New Group** button in the [User Manager](#) for a popup dialog, as shown in [Figure 1-109](#).

**Figure 1-109** New Group dialog



**Note:** AX-3.5 and later hosts limit shown domain groups to those 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

**Note:** You cannot delete any Windows “Built-In” group.

In the [User Manager](#), 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 in [Figure 1-106](#).

**Figure 1-110** Confirm Delete dialog

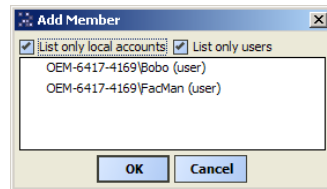


When you click **OK**, the selected group(s) is deleted and removed from the groups table.

## Add member

Select (click) a group in the [User Manager](#), then the **Add Member** button for a popup ([Figure 1-111](#)).

**Figure 1-111** 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.

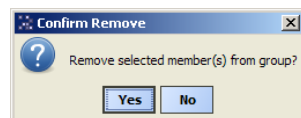
By default, as in the **New Group** dialog ([Figure 1-109](#)), the users checklist is filtered to *not list* domain users and groups. If needed, click these checkboxes to add or remove these choices in the users checklist. When you click **OK**, the group’s membership is updated with the member(s) you added.

**Note:** You can also [drag and drop](#) users (rows in users table) onto groups (rows in groups table).

## Remove member

Select (click) a group in the [User Manager](#), then in the membership table, select one or more users. With the user(s) selected, click the **Remove Member** button for confirmation dialog ([Figure 1-112](#)).

**Figure 1-112** Confirm Remove dialog

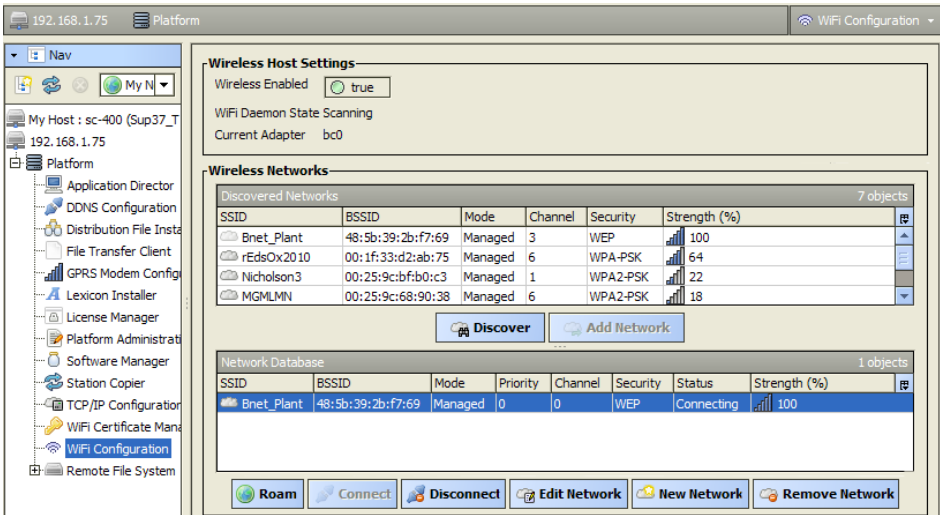


When you click **OK**, the selected user(s) is removed that group’s membership.

WiFi Configuration

**WiFi Configuration** appears among [platform views](#), along with a **WiFi Certificate Manager** view., only if a AX-3.6 or later JACE host that has an installed 802.11b/g wireless WiFi adapter, At the time of this document, this applies only to a JACE-700 with a Mini-PCI WiFi adapter card (T7-WIFI option). This JACE-7 option became available in the initial AX-3.6 software release time frame.

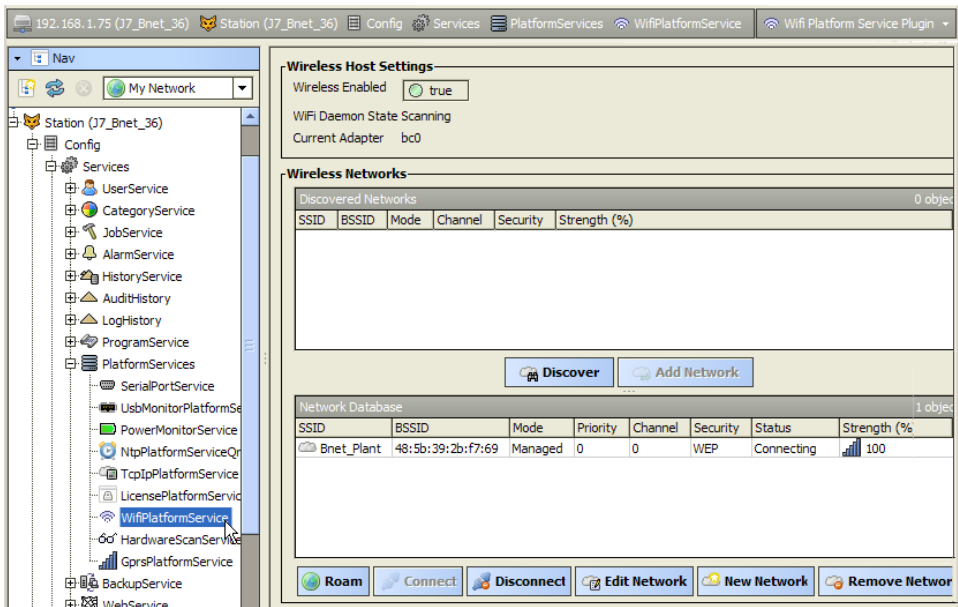
Figure 1-113 WiFi Configuration platform view (AX-3.6 or later)



This view lets you discover 802.11b/g networks available to the JACE, and add one or more networks, as necessary.

**Note:** The JACE's running station also has a **"WifiPlatformService"** among its platform services. Its default **"Wifi Platform Service Plugin"** view is identical to the **WiFi Configuration** view.

Figure 1-114 Wifi Platform Service Plugin view



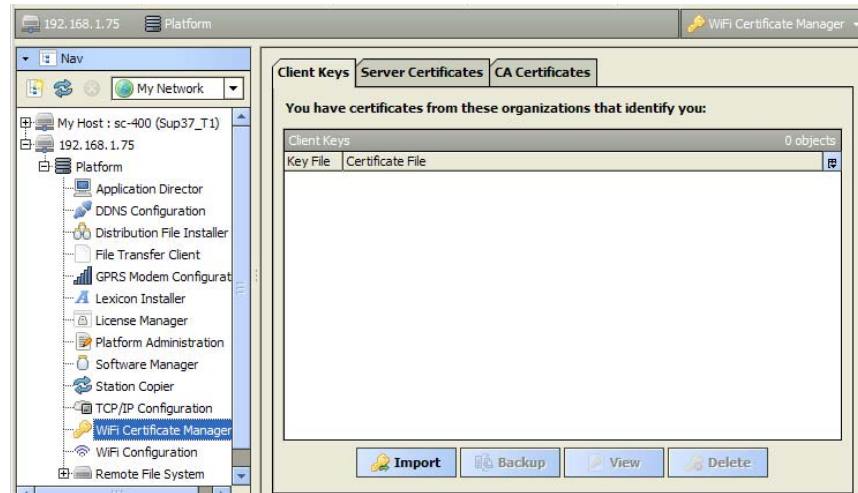
Refer to the Engineering Notes II document "NiagaraAX JACE WiFi option" for complete details, including usage scenario, configuring the JACE for WiFi, and further details on this view.

## WiFi Certificate Manager

**WiFi Certificate Manager** appears among [platform views](#), along with a **WiFi Configuration** view, only if a AX-3.6 or later JACE host that has an installed 802.11b/g wireless WiFi adapter. At the time of this document, this applies only to a JACE-700 with a Mini-PCI WiFi adapter card (T7-WIFI option).

**Note:** *Starting in AX-3.7, this view changed in appearance from the previous (AX-3.6) **WiFi Certificate Manager** view. 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 1-115** 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 JACE's running station also has a “**WifiPlatformService**” among its platform services. However, at the time of this document, 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.*

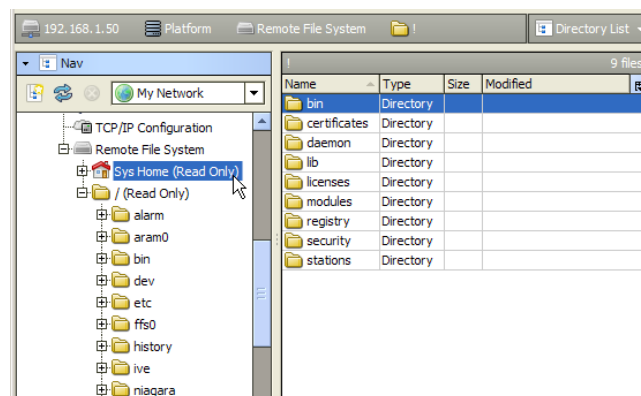
Refer to the Engineering Notes II document “NiagaraAX JACE WiFi option” for complete details, including further details on using this view.

## 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.

For [QNX-based](#) platforms (only), it also provides browse capability of the file system root (/) of the remote host, using the Nav tree side bar of Workbench ([Figure 1-116](#)).

**Figure 1-116** Remote File System for QNX-based host





Note to actually copy (transfer) files between your Workbench PC and the remote NiagaraAX platform, you use the platform **File Transfer Client** view. See [“File Transfer Client”](#) on page 1-30.



# CHAPTER 2

## Platform Services

---

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.

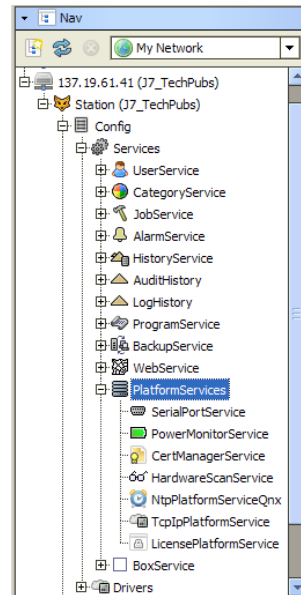
The following main sections provide more details:

- [“About Platform Services”](#) on page 2-2
  - [“Component differences for platform services”](#) on page 2-2
- [“PlatformServiceContainer parameters”](#) on page 2-3
  - [“PlatformServiceContainer status values”](#) on page 2-3
  - [“PlatformServiceContainer configuration parameters”](#) on page 2-4
  - [“Additional PlatformServiceContainer properties”](#) on page 2-6
  - [“Model-specific PlatformServiceContainer properties”](#) on page 2-7
  - [“PlatformServiceContainer actions”](#) on page 2-7
- [“SystemService \(under PlatformServices\)”](#) on page 2-8
- [“Platform service types”](#) on page 2-9
- [“Using platform services in a station”](#) on page 2-10
  - [“JACE power monitoring”](#) on page 2-10
  - [“JACE-NXT \(and JACE-NXS\) power monitoring”](#) on page 2-11
  - [“PlatformServices binding and link caveats”](#) on page 2-12
- [“About the NtpPlatformService”](#) on page 2-13
  - [“Interaction with station's TimeSyncService”](#) on page 2-13
  - [“NTP port/firewall considerations”](#) on page 2-13
  - [“About the Ntp Platform Service Editor”](#) on page 2-13
  - [“About the Ntp Platform Service Editor Win32”](#) on page 2-14
  - [“About the Ntp Platform Service Editor Qnx”](#) on page 2-15 (most JACEs)

## About Platform Services

Under **Config, Services**, every *running* Niagara station has a **PlatformServices** container (Figure 2-1), which any station user with admin-level permissions to this component can access.

**Figure 2-1** 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” (note that if a [QNX-based](#) JACE, this results in a host reboot!). For details about station security, see “About Security” in the User Guide.

[Component differences for platform services](#) (as explained in the next section) should be understood.

### 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!

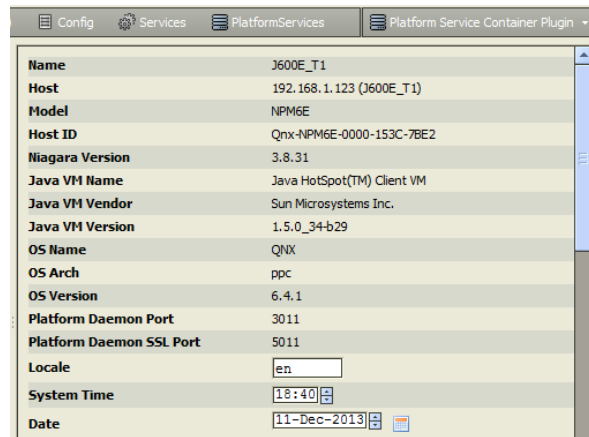
In summary, when you make changes under a station's PlatformServices, those changes are independent of the running station. If you install another station, platform services are dynamically recreated again when the new station starts, based upon the last settings.

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 QNX-based JACE. A “Reboot Now?” popup dialog appears upon saving such a change.

## PlatformServiceContainer parameters

In addition to being a container, the [PlatformServices](#)Container provides various status and configuration entries for the host platform. In the Nav tree, double-click PlatformServices to access the “Platform Service Container Plugin” which lists these entries, as shown in [Figure 2-2](#).

**Figure 2-2** PlatformServicesContainerPlugin view (many entries not shown)



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](#)
- [Additional PlatformServiceContainer properties](#)

**Note:** By default, any [PlatformServices](#)Container also provides three right-click [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](#)” on page 1-11 for further details.
- **Host ID**  
NiagaraAX host identifier, a string unique to this one machine.
- **Niagara Version**  
Version and build number of the NiagaraAX distribution running in the host platform.
- **Java VM Name**  
Java virtual machine used, e.g. either “Java HotSpot(TM) Client VM” or “J9” ([QNX-based](#) hosts) or “Java HotSpot(TM) Client VM” for Windows-based hosts.
- **Java VM Vendor**  
Vendor for Java VM, e.g. “Sun Microsystems Inc.” (Java HotSpot) or “IBM Corporation” (J9).
- **Java VM Version**  
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)
- **OS Version**  
Operating System version, such as “6.4.1” (QNX) or “5.1” (Windows XP)
- **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](#)” on page 1-46), but then forgot what the new port is.

- **Platform Daemon SSL Port**  
Port number on which the platform daemon is listening for its platform SSL server (5011, or another port number, providing that platform SSL enabled). If platform SSL is disabled, it reads “Unknown”. This can prove useful in case you changed the platform SSL port (see “[Change SSL Settings](#)” on page 1-47), 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).
- **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 QNX-based JACE host, it may look similar to:

	Total	Free	Block Size	Files	Max Files
/ffs0	126,976 KB	77,160 KB	2,048 bytes	732	4096
/aram0	31,488 KB	29,832 KB	1,024 bytes	28	4096

  
*Note:* The “Files” and “Max Files” values require the JACE to be running AX-3.6 or later.
- **Physical RAM**  
Current total and free RAM statistics for the host. For a QNX-based JACE, it may look similar to:

Total	Free
262,144 KB	70,160 KB
- **Serial Number**  
(Appears only if QNX-based JACE running AX-3.6 or later). The controller’s unique serial number.
- **Hardware Revision**  
(Appears only if QNX-based JACE running AX-3.6 or later). Hardware revision of the controller.
- **Hardware Jumper Preset**  
(Appears only if QNX-based JACE) Either true or false—indicates whether or not the mode jumper is installed for “serial shell mode” access. Read at boot time only. See “About JACE serial shell mode” in the *JACE NiagaraAX Install & Startup Guide*.
- **EWf Overlays**  
(Appears only if a CompactFlash-based JACE-NXT or JACE-NXS host) Status of the Enhanced Write Filter overlays which protect the CompactFlash card from excessive writes. Two overlays are listed, one for OS, and one for History. The current **State** can either be “Enabled,” “Disabled,” or “Unknown.” The state should always say “Enabled” for both overlays under normal operation. The **Command** should normally be “Commit.” Other possible values are “noCmd” and “CommitAndDisable.” **Ram Usage** is the amount of RAM used by the overlay, and should “cap out” at some maximum number during normal operation. The history overlay theoretical maximum is about 5% larger than the size of the History partition, about 85MB.  
Also see the *JACE-NXT NiagaraAX Install & Startup Guide* and *JACE-NXS NiagaraAX Install & Startup Guide*.

## 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”. For details, see Sun documentation at <http://docs.oracle.com/javase/1.4.2/docs/api/java/util/Locale.html>.
- **System Time**  
Current local time in host.
- **Date**  
Current local date in host.
- **Time Zone**  
Current local time zone for host. For more details, see “[Time Zones and NiagaraAX](#)” on page B-1.

- **Engine Watchdog Policy**  
*Note:* 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.
  - 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 QNX-based host, or every (1) hour if Windows-based 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 QNX-based JACE hosts 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-based hosts, the default is 3, and typically (unless a CompactFlash-based JACE-NXT or JACE-NXS) can be safely adjusted up, if desired.
- **Battery Present**  
Applies to configuration of a QNX-based JACE’s backup battery, requiring the JACE to also be running build 3.6.44 or later. 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).  
*Note:* The configuration described above is only one of three possible backup options for an SRAM-equipped controller (e.g. JACE-6E, JACE-2,-6,-7 with an SRAM option card), as well as the newest JACE-3E controller. 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 Engineering Notes II document *JACE Data Recovery Service (SRAM support)*.
- **Failure Reboot Limit**  
Limits the number of system reboots 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” on page 1-17). Default value is 3.
- **Failure Reboot Limit Period**  
Specifies the repeating frequency of the Failure Reboot Limit period, with default value at 1 hour.  
*Note:* 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)

Also see next section, “Additional PlatformServiceContainer properties” on page 2-6.



## Additional PlatformServiceContainer properties

A station running on a QNX-based platform has the following additional PlatformServicesContainer properties, as shown in Figure 2-3.

**Figure 2-3** Additional PlatformServiceContainer properties for QNX-based host.

Failure Reboot Limit	<input type="text" value="3"/> [1 - max]		
Failure Reboot Limit Period	<input type="text" value="00000h 10m"/> [0ms - +inf]		
<b>RAM Disk</b>	<b>Min Free</b>	<b>Size</b>	<b>Status</b>
	<input type="text" value="5"/> % [0 - 100]	<input type="text" value="6"/> [1 - 12] MB	Ok
<b>Java Heap</b>	<b>Min Free</b>	<b>Max</b>	<b>Free</b> <b>Status</b>
	<input type="text" value="3"/> MB	48 MB	39 MB Ok
<b>Open File Descriptors</b>	<b>Min Free</b>	<b>Max Open</b>	<b>Free</b> <b>Status</b>
	<input type="text" value="50"/> [50 - max]	2000	1934 Ok
<b>Free RAM</b>	<b>Min Free</b>		<b>Status</b>
	<input type="text" value="1024"/> [512 - max] KB		Ok
<b>Disk Space</b>	<b>Min Free</b>		<b>Status</b>
	<input type="text" value="10"/> % [0 - 100]		Ok
<b>Files</b>	<b>Min Free</b>		<b>Status</b>
	<input type="text" value="50"/> [50 - max]		Ok

These properties are in addition to the other standard configuration properties. Each of these additional properties has a read-only **Status** field on the far right side, plus the following configuration fields:

- RAM Disk Size**  
 Has two configurable fields:
  - 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 —in MB, where default is 16 for a JACE-4/5 series, 8 for a JACE-2 series, 32 for a JACE-6 series, or 48 for a JACE-7 series. Specifies the size of RAM disk used to store history and alarm files. Specifies 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.  
***Note:** 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 starting in AX-3.6, the PlatformServiceContainer status property “Filesystem” includes both the current number of files and the maximum number of files for each partition on a QNX-based JACE.

***Note:** Current statistics for memory, heap, and file descriptors (fd) are accessible on a station opened in Workbench, via the **Resource Manager** view of the Station component.*

Also see previous section, “PlatformServiceContainer configuration parameters” on page 2-4.

## Model-specific PlatformServiceContainer properties

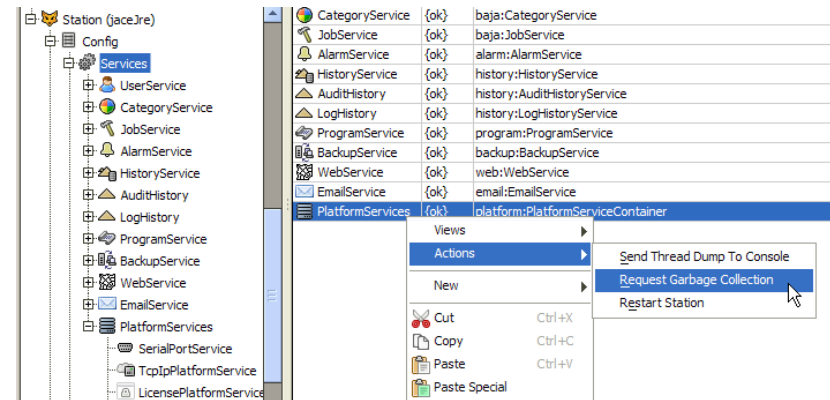
Some QNX-based JACE models may have yet more PlatformServices properties, specific to special hardware features. This is in addition to the standard and additional properties described above. Typically, these are configured at JACE commissioning time.

For more details, see “Controller-specific PlatformServices properties” in the *JACE NiagaraAX Install & Startup Guide*.

## PlatformServiceContainer actions

The PlatformServicesContainer also provides three (right-click) actions, as shown in Figure 2-4.

**Figure 2-4** PlatformServicesContainer actions.



These actions are described as follows:

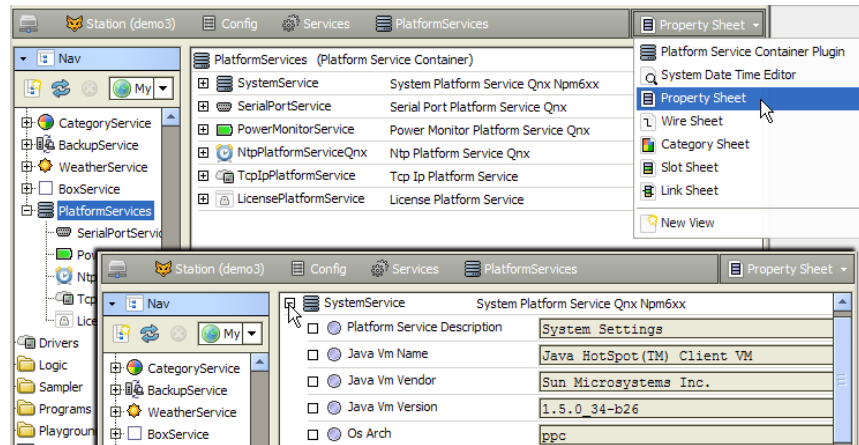
- Send Thread Dump to Console**  
 Causes that host's platform daemon to have the station send a VM thread dump to its [standard output](#) (console), equivalent to the “Dump Threads” command in the [Application Director](#). Typically used only during troubleshooting.  
*Note:* Apart from *Application 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://<host IP>/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 PlatformServicesContainer have an available “Poll” action, which refreshes their property values. See “Platform service types” for a listing of possible child services.

## SystemService (under PlatformServices)

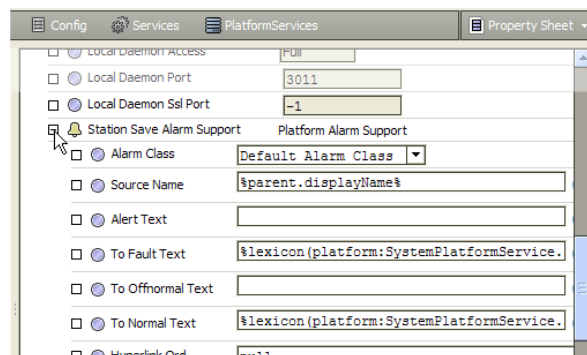
[PlatformServices](#) also contains a child “SystemService” container, accessible from its property sheet as shown in [Figure 2-5](#). Unlike other child services, SystemService does not appear in the Nav tree.

**Figure 2-5** 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](#)” on page 2-3). In addition, as shown in [Figure 2-6](#), there are two other slots: “Local Daemon Ssl Port” (if a AX-3.7 or later platform), and a container slot “Station Save Alarm Support”.

**Figure 2-6** Station Save Alarm Support expanded in property sheet of SystemService.



The “Local Daemon Ssl Port” value will be “-1” if the platform is not configured (and enabled) for platform SSL, or else otherwise shows the port used for secureSSL connections, for example, 5011.

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. For property descriptions, see the *User Guide* section “About alarm extension properties”.

**Note:** Other platform warnings from defined limits, such as for low memory, low disk space, and so on (see [Figure 2-3](#) on page 6 for related properties) 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](#)” on page 2-12.

## Platform service types

In addition to the [SystemService](#) found under its property sheet, the [PlatformServicesContainer](#) 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 a Workbench [platform connection](#) to a JACE's platform daemon). Examples include types [TcpIpService](#) and [LicenseService](#).*

The list of visible platform service types includes the following:

- **TcpIpService**  
Provides access to the same configuration using the platform's **TCP/IP Configuration** view. See "[TCP/IP Configuration](#)" on page 1-72.
- **LicenseService**  
Provides access to the same configuration using the platform's **License Manager** view. See "[License Manager](#)" on page 1-35.
- **SerialPortService**  
Allows review of available serial ports on the host platform. For older [QNX-based](#) JACEs with configurable serial ports (e.g. JACE-403), this is where you *configure port usage*. For a related procedure, see "JACE serial port configuration" in the *JACE NiagaraAX Install & Startup Guide*.
- **PowerMonitorService**  
Currently applies to [QNX-based](#) JACEs, providing configuration and status of the JACE's battery monitoring and AC power-fail shutdown routines. See "[JACE power monitoring](#)" on page 2-10 for details. This service also applies to some models of the (Win32-based) JACE-NXT and JACE-NXS, see "[JACE-NXT \(and JACE-NXS\) power monitoring](#)" on page 2-11.
- **NtpPlatformService**  
Provides the Niagara interface to the NTP (Network Time Protocol) service or daemon of the platform's OS (QNX, Win32, Linux), including several configuration parameters and a list specifying one or more NTP time servers. For details, see "[About the NtpPlatformService](#)" on page 2-13.
- **GprsPlatformService**  
Available only for a QNX-based JACE with a GPRS modem option card installed, providing that the `platGprs` module is installed. Refer to the section "About the GprsPlatformService" in the Engineering Notes document *GPRS modem option* for more complete details.
- **UsbMonitorPlatformService**  
(Applies to JVLN, i.e. JACE-7 platforms only). Provides read-only debug type details related to USB port monitoring on the JACE-7 series controller. If client applications are installed that interface with this service, system notifications may occur when USB devices are inserted or removed.
- **DataRecoveryService**  
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. For details, refer to the Engineering Notes II document *JACE Data Recovery Service (SRAM support)*.  
**Note:** *In the initial AX-3.6 release, this service automatically removed any former **PowerMonitorService**. However, starting in build 3.6.44, this changed such that both services are retained. However, if the PlatformServiceContainer's configuration property "Battery Present" is set to false, the PowerMonitorService no longer monitors for a battery, preventing "battery bad" nuisance alarms.*
- **HardwareMonitorService**  
Applies to station in a Win32-based JACE only (JACE-NXT, JACE-NXS, JACE-NX). Provides status of several internal environmental variables, including alarm limit configuration. See "[JACE-NXT hardware monitoring](#)" on page 2-11 and "[JACE-NXS hardware monitoring](#)" on page 2-11 for details.
- **WifiPlatformService**  
(Applies if a JACE-700 with installed WiFi option only). Allows configuration of the optional WiFi adapter installed in the JACE controller, providing that the `platWifi` module is installed. For details, see the "About the WifiPlatformService" section in the Engineering Notes II document *NiagaraAX JACE WiFi option*.

## 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:

- [JACE power monitoring](#)
- [JACE-NXT \(and JACE-NXS\) power monitoring](#)
- [JACE-NX hardware monitoring](#)
- [JACE-NXS hardware monitoring](#)

### JACE power monitoring

By default, through the **PowerMonitorService**, any [QNX-based](#) 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”](#) on page 2-12).

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 & Startup Guide*.

**Note:** *A station running in a JACE controller with an “SRAM option card” may not have a PowerMonitorService, if running an AX-3.6 build prior to 3.6.44. Such a JACE is always considered “battery-less”, where its “DataRecoveryService” (for SRAM support) automatically removes the PowerMonitorService.*

*This changed starting in build 3.6.44, such that any SRAM-equipped JACE (e.g. JACE-6E, JACE-3E, or another JACE with SRAM option card), always retains its PowerMonitorService, along with a DataRecoveryService (if the platDataRecovery module is installed). So, you can selectively disable battery monitoring. See the next section “Battery monitoring disabled”. Additionally, other SRAM support options exist. For related details, refer to the Engineering Notes II document JACE Data Recovery Service (SRAM support).*

**Note:** *As new QNX-based JACE platforms are introduced, additional power-monitoring capabilities may be present in the station’s PowerMonitorService. For example, both the JACE-7 series and JACE-x02 Express series (“M2M JACE”) may be installed with two backup batteries: the NiMH battery like a JACE-2/6, plus an optional 12V sealed lead-acid (SLA) battery that provides system operation for some duration during a power outage. In this “dual battery” PowerMonitorService, separate slots exist for the monitoring and alarming of both batteries. For related details, see the latest JACE NiagaraAX Install & Startup Guide.*

### Battery monitoring disabled

As mentioned in the first Note above, 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). Prior to build 3.6.44, battery monitoring stopped simply because the station’s PowerMonitorService was automatically removed.

However, starting in build 3.6.44, a station’s PowerMonitorService is always retained. This service 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”](#) on page 2-4,



## JACE-NXT (and JACE-NXS) power monitoring

**Note:** Often, a hard-drive based JACE-NXT or JACE-NXS does not have the special SITOP DC UPS module (with battery accessory), in which case its `PowerMonitorService` will have no application. The following slots do apply to the CompactFlash-based JACE-NXT or JACE-NXS, which will be so equipped.

Currently, through the `PowerMonitorService`, a station running in a JACE-NXT or JACE-NXS provides status monitoring of the following items, via “Boolean” type slots:

- **DC power**  
 (“Primary Power Present” slot) — True whenever DC power is currently supplied to the UPS.
- **Battery level**  
 (“Battery OK” slot) — True if last UPS backup-battery test was good.
- **UPS Connectivity**  
 (“Ups Talking” slot) — value is “talking” if last JACE attempt to communicate to the UPS was successful. Another possible state is “no comm.” Note that a USB cable must be connected between the JACE-NXS and the SITOP UPS module.

If needed, you can make Px bindings or links to these slots (however, see [“PlatformServices binding and link caveats”](#) on page 2-12). In addition to these read-only status slots, this `PowerMonitorService` provides related *configuration* slots, which you typically review at commissioning time.

For more details and a related procedure, refer to:

- “Power monitoring JACE-NXT configuration” in the *JACE-NXT NiagaraAX Install & Startup Guide*.
- “Power monitoring JACE-NXS configuration” in the *JACE-NXS NiagaraAX Install & Startup Guide*.

## JACE-NXT hardware monitoring

A station in the [Windows-based](#) JACE-NXT provides status monitoring of *internal* hardware parameters, via “Float” type slots under its `HardwareMonitorService`.

The three items monitored are:

- **CPU temperature**  
 (“Cpu Temp” slot) — Value in degrees (C or F) of the mainboard under the JACE-NXS’s CPU.
- **Board temperature**  
 (“Board Temp” slot) — Value in degrees (C or F) of the space inside the chassis.
- **RAM temperature**  
 (“Ram Temp” slot) — Value in degrees (C or F) of the space inside the chassis.

If needed, you can make Px bindings or links to these slots (however, see [“PlatformServices binding and link caveats”](#) on page 2-12). In addition to these read-only status slots, the `HardwareMonitorService` provides related *configuration* slots, which you typically review at commissioning time. For more details and a related procedure, see “Hardware monitoring JACE-NXT configuration” in the *JACE-NXT NiagaraAX Install & Startup Guide*.

## JACE-NXS hardware monitoring

A station in the [Windows-based](#) JACE-NXS provides status monitoring of *internal* hardware parameters, via “Float” type slots under its `HardwareMonitorService`.

The two items monitored are:

- **CPU temperature**  
 (“Cpu Temp” slot) — Value in degrees (C or F) of the mainboard under the JACE-NXS’s CPU.
- **Board temperature**  
 (“Board Temp” slot) — Value in degrees (C or F) of the space inside the chassis.

If needed, you can make Px bindings or links to these slots (however, see [“PlatformServices binding and link caveats”](#) on page 2-12). In addition to these read-only status slots, the `HardwareMonitorService` provides related *configuration* slots, which you typically review at commissioning time. For more details and a related procedure, see “Hardware monitoring JACE-NXS configuration” in the *JACE-NXS NiagaraAX Install & Startup Guide*.

## JACE-NX hardware monitoring

A station in the (discontinued) [Windows-based](#) JACE-NX provides status monitoring of various *internal* hardware parameters, via “Float” type slots under its `HardwareMonitorService`.

A few of the items monitored include:

- **CPU temperature**  
 (“Cpu Temp” slot) — Value in degrees (C or F) of the mainboard under the JACE-NX’s CPU.

- **CPU fan Speed**  
("Cpu Fan Speed" slot) — RPM value of the CPU fan inside the JACE-NX.
- **Chassis fan Speed**  
("Sys Fan Speed" slot) — RPM value of the JACE-NX's rear-mounted chassis fan.

Also included are various internal voltage values (Vtt, CPU core, Vcc 3 and 5, +12, -12, Vsb).

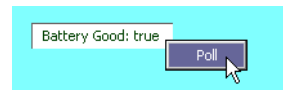
If needed, you can make Px bindings or links to these slots (however, see "[PlatformServices binding and link caveats](#)" on page 2-12). In addition to these read-only status slots, the HardwareMonitorService provides related *configuration* slots, which you typically review at commissioning time. For more details and a related procedure, see "Hardware monitoring in the JACE-NX" in the *JACE-NX NiagaraAX Install & Startup Guide*.

## 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:

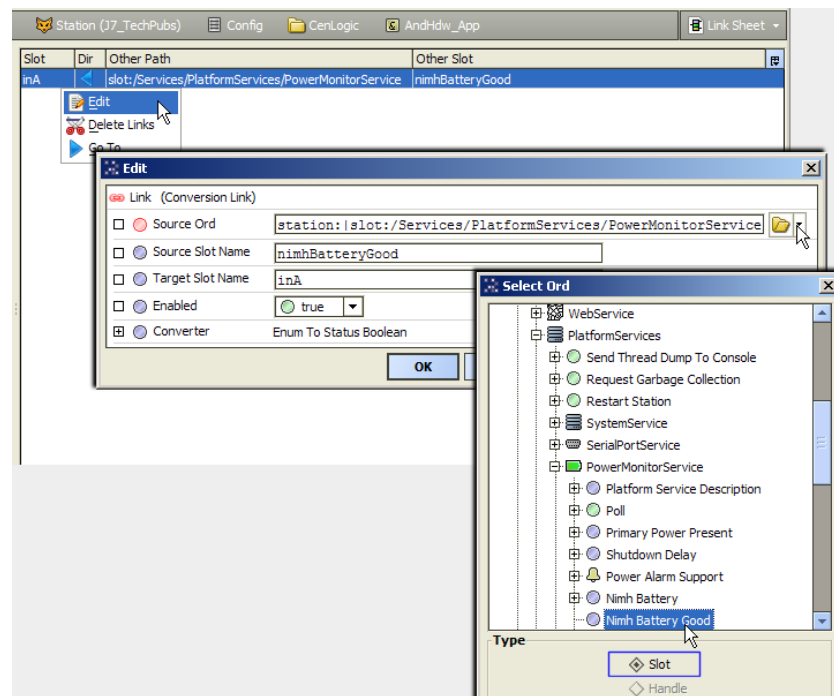
- Subscription behavior is unique to a station's PlatformServices slots, in that property values initially load, but do *not automatically update*. To explicitly refresh such properties, you must invoke the "poll" action of the container for those properties.  
For example, if on a Px page you bind a 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 2-7** Poll action on bound PlatformServices property



- Links from PlatformServices (and child slots) to other station components must use a source ord *"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 [Figure 2-8](#).  
**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 2-8** From LinkSheet 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 (AX-3.4 or later) 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 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:

- [Interaction with station's TimeSyncService](#)
- [NTP port/firewall considerations](#)
- [About the Ntp Platform Service Editor](#)

### Interaction with station's TimeSyncService

Typically, an NTP-only solution is used to synchronize time among NiagaraAX hosts running AX-3.4 or later, such that all platforms use their enabled *NtpPlatformService* for their OS type, and each has no *TimeSyncService* component in their station. A Supervisor may be NTP-configured to sync with one or more domain controllers, or perhaps to sync to one or more well-known Internet public NTP time servers. Whereas JACEs typically specify the Supervisor as the NTP time server. Or alternatively, a JACE may be configured to sync to other JACEs, or (providing it has access to DNS and a gateway), a public NTP time server.

However, it may be possible that RFC 868 time server support is still required, for example an AX-3.4 or later Supervisor on a job that includes one or more AX-3.3 JACEs. In this scenario, the Supervisor could synchronize its own time using NTP (as a client) using its *NtpPlatformServiceWin32*, and also behave as an RFC 868 time server via the *TimeSyncService* in its running station. Note in this configuration, any child “*TimeSyncClient*” components under its *TimeSyncService* are irrelevant—as whenever the *NtpPlatformService* is enabled, client synchronization is only possible using NTP.

However, in the *TimeSyncService* of any remote AX-3.3 (or earlier) JACE station, it could reference the Supervisor as the *TimeSyncClient* (time server) to request/receive RFC 868 synchronization.

### NTP port/firewall considerations

On any host, NTP requires the use of UDP port 123—this port is not configurable. On a QNX-based JACE this is not an issue. However, on a Windows or Linux host platform, you typically need to make the necessary firewall exception or “iptables” entry to allow UDP port 123 traffic.

**Note:** Otherwise, NTP time synchronization can fail because of firewall-blocked messages.

### About the Ntp Platform Service Editor

Regardless of platform OS type (Windows, QNX, Linux), the default view for any *NtpPlatformService* is an **Ntp PlatformService Editor OS** view, your typical interface. Double-click any *NtpPlatformService* to see this editor.

Although a few differences exist between the different NTP platform service editors in property names, 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 Win32](#)
- [Ntp Platform Service Editor QNX](#)
- [Ntp Platform Service Editor Linux](#)

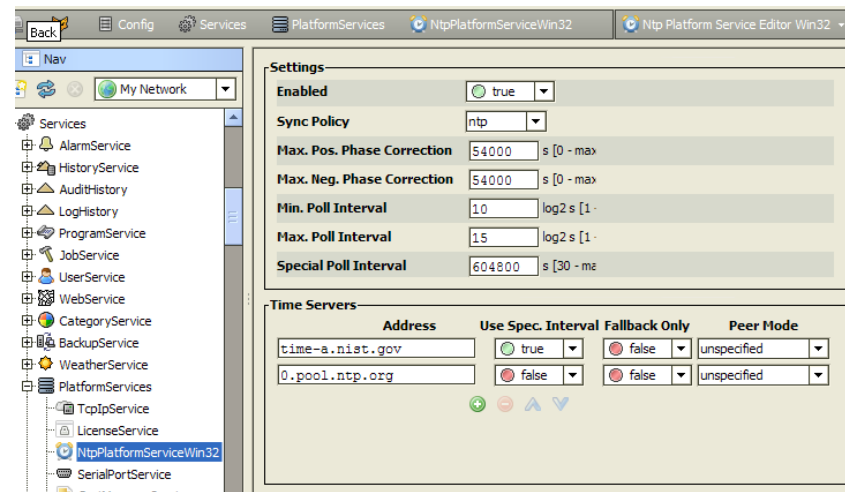
**Note:** All *NtpPlatformServices* have a right-click action “Poll”. This action refreshes the values shown in the *Ntp Platform Service Editor* (received from the NTP daemon), but does not affect messaging between the host and an NTP server.

However, starting in AX-3.6, the *NtpPlatformServiceQnx* (for a JACE) does have an additional “Sync Now” action, which does attempt an immediate sync. See [“Sync Now action”](#) on page 2-16 for more details.

## About the Ntp Platform Service Editor Win32

An example Win32 host Ntp Platform Service Editor is shown in Figure 2-9. This is the default view for the RFC-1305 compliant NtpPlatformService on a Windows-based (Win32 or Win64) host.

**Figure 2-9** 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.





## About Ntp Platform Service Editor Win32 settings

Settings in the Ntp Platform Service Editor Win32 include the following properties:

- **Enabled**  
If true, the host will use NTP to sync its clock with time values retrieved from other servers.
- **Sync Policy**  
Specifies how the host should sync, where possible choices include:
  - none — available only if Enabled = false. Clock is not sync’ed with any time servers.
  - ntp — available only if Enabled = true. Use NTP to sync with servers in the Time Servers list.
  - domain — available only if Enabled = true and the host is a member of a Windows domain. Use NTP to sync with domain controllers, but *not* with servers in Time Servers list.
  - both — available only if Enabled = true and the host is a member of a Windows domain. Use NTP to sync with domain controllers, and also with servers in Time Servers list.
- **Max. Pos. Phase Correction**  
Maximum amount of time, in seconds, that the clock can be set forward in a sync. Default is 54000, or 15 hours. If the service determines a larger correction is needed, an event log is created instead.
- **Max. Neg. Phase Correction**  
Maximum amount of time, in seconds, that the clock can be set backward in a sync. Default is 54000, or 15 hours. If the service determines a larger correction is needed, an event log is created instead.
- **Min. Poll Interval**  
The shortest period allowed between polls, where units are in “log-base-two seconds,” or 2 to the power of  $n$  seconds (NTP convention). For example, if this value is 10, then the interval is 2 to the 10th seconds, or 1024 seconds.
- **Max. Poll Interval**  
The longest period allowed between polls, where units are in “log-base-two seconds,” or 2 to the power of  $n$  seconds (NTP convention). For example, if this value is 14, then the interval is 2 to the 14th seconds, or 16384 seconds.
- **Special Poll Interval**  
Poll interval, in seconds, for servers in the Time Servers list that have “Use Spec. Interval” set to true.

## About Ntp Platform Service Editor Win32 time servers

Each entry in the time servers list in the [Ntp Platform Service Editor Win32](#) specifies a server to which the host's clock will be sync'd when the Sync Policy is either "ntp" or "both." These servers are *not* used if the Sync Policy is "none" or "domain."

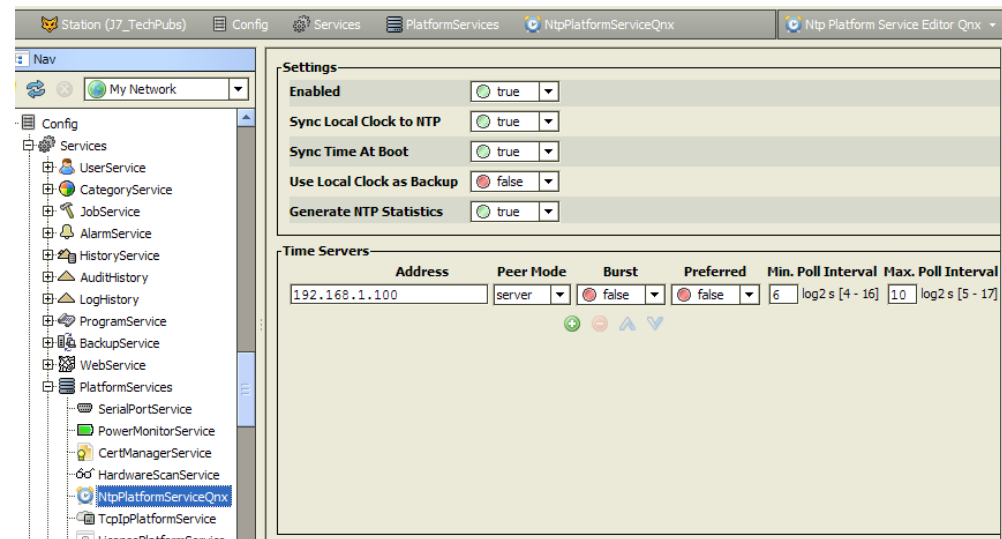
Controls below the list allow you to add  and delete  servers, as well as reorder up  or down . Fields for each time server includes the following:

- **Address**  
Fully qualified domain name, IP address, or host files alias for the NTP time server.
- **Use Spec. Interval**  
Default is false. If true, the poll interval rules specified by RFC 1305 are ignored, and the specified Special Poll Interval is used instead.
- **Fallback Only**  
Default is false. If true, this server is polled only if other servers cannot be reached.
- **Peer Mode**  
Specifies the server's sync relationship:
  - unspecified — Use the system default behavior when sync'ing with the server.
  - symmetricActive — Both this host's and the server's clocks may be changed as a result of each sync.
  - client — Only this host's clock may be changes as a result of each sync.

## About the Ntp Platform Service Editor Qnx

An example QNX-based host Ntp Platform Service Editor is shown in [Figure 2-10](#). This is the default view for the RFC-1305 compliant [NtpPlatformService](#) on any QNX-based JACE.

**Figure 2-10** 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.

As in all [Ntp Platform Service Editors](#), there are two main areas: [Settings](#) at top, [Time Servers](#) at bottom. Also, starting in AX-3.6, the NtpPlatformServiceQnx has an available "Sync Now" action. For more details, see ["Sync Now action"](#) on page 2-16.

## About 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.  
*Note: 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 NiagaraAX has no control over what it reports.

### About 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.

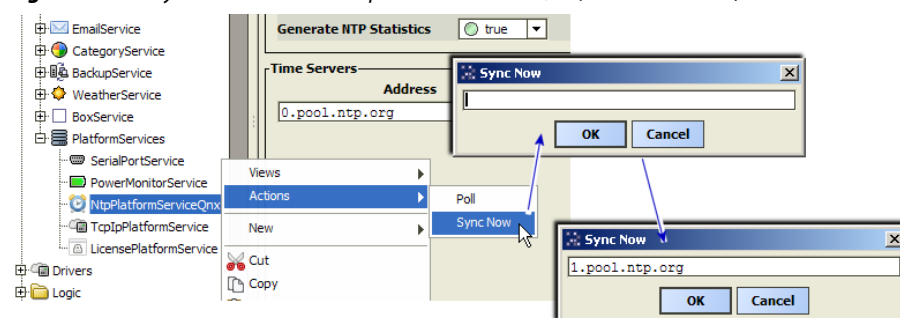
Controls below the list allow you to add  and delete  servers, as well as reorder up  or down  (to establish priority order, highest at top). Fields for each time server includes the following:

- **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, starting in AX-3.6 the **NtpPlatformServiceQnx** component has an additional "**Sync Now**" action.

**Figure 2-11** Sync Now action on NtpPlatformServiceQnx (AX-3.6 and later)



As shown in [Figure 2-11](#), 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**.

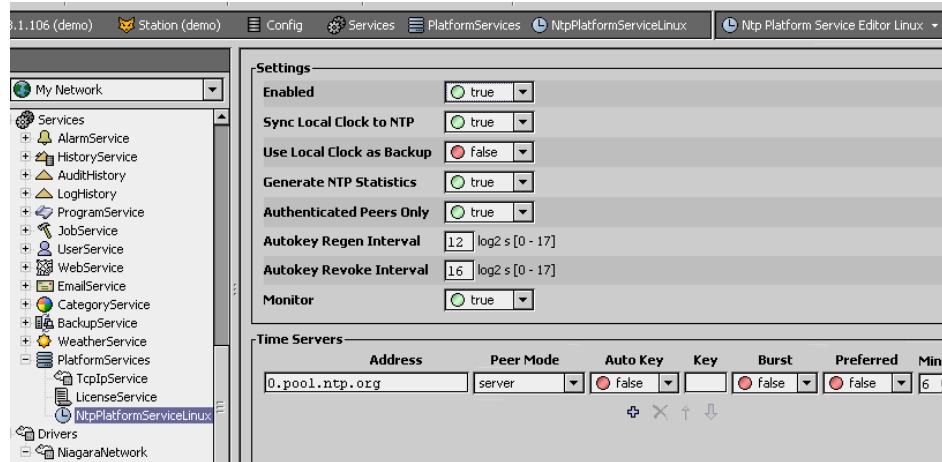
To verify, look for a related entry in the station's spy "platform diagnostics" log. Do this in Workbench by right-clicking the station, then selecting **Spy > platform diagnostics > log** or from the Workbench **File** menu, **File > Open ord** (Ctrl + L) and enter:

```
ip:JACE_IP_address|fox:|spy:/platform diagnostics/log
```

## About the Ntp Platform Service Editor Linux

An example Linux host Ntp Platform Service Editor is shown in [Figure 2-12](#). This is the default view for the RFC-1305 compliant [NtpPlatformService](#) on a Linux Supervisor host.

**Figure 2-12** Ntp Platform Service Editor Linux



This dialog provides access to some key settings of the Network Time Protocol (NTP) Daemon on the Supervisor's Linux OS.

**Note:** *Settings are only a small subset of those available—for all settings, plus the ability to make changes in the NTP Daemon, use root login access of the Linux OS on the host. Refer to the Linux documentation about the NTP Daemon for specific details.*

As in all [Ntp Platform Service Editors](#), there are two main areas: [Settings](#) at top, [Time Servers](#) at bottom.

## About Ntp Platform Service Editor Linux settings

Settings in the [Ntp Platform Service Editor Linux](#) 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.
- **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.
- **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 NtpPlatformServiceLinux and select **Views > Spy Remote**. Keep in mind that the ntpd is a Linux process; thus NiagaraAX has no control over what it reports.
- **Authenticated Peers Only**  
If true (default), enables the server to synchronize with unconfigured peers only if the peer has been correctly authenticated using a trusted key and key identifier.
- **Autokey Regen Interval**  
Specifies the interval between regenerations of the session key list used with the autokey feature. Note that the size of the key list for each association depends on the interval and the current poll

interval. The default value is 12 (units in NTP are in “log-base-two seconds,” or 2 to the power of  $n$  seconds where 12 means 4096 seconds, or about 1.1 hours). For poll intervals about the specified interval, a session key list with a single entry will be regenerated for every message sent.

- **Autokey Revoke Interval**  
Specifies the interval between regenerations of the private value used with the autokey feature.
- **Monitor**  
If true (default), enables the ntpd monitoring facility.

### About Ntp Platform Service Editor Linux time servers

Each entry in the time servers list in the [Ntp Platform Service Editor Linux](#) specifies a server to which the host's clock will be sync'ed when the service is Enabled (true), and “Use NTP on Server” 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 . 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).
- **Auto Key**  
If true, all packets sent to the address include authentication field encrypted using autokey scheme. If false, packets sent to the address include the authentication field encrypted using the specified Key identifier.
- **Key**  
Applies only if Auto Key is disabled (false). To specify a key identifier, as an unsigned 32-bit integer less than 65536, used in authentication with packets set to the address.
- **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 6. 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 6th (64 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 (1024 seconds) to 2 to the 17th (131,072 seconds).



# CHAPTER 3

## Platform Component Guides

---

Component Guides provides summary information on platform-related components.

### Component Reference Summary


Summary information is available on components in the following modules:

- [platCrypto](#)
- [platDataRecovery](#)
- [platGprs](#)
- [platform](#)
- [platIEEE8021X](#) (AX-3.8)
- [platHwScan](#)
- [platPower](#)
- [platPowerNxs](#)
- [platSerialQnx](#)
- [platSerialWin32](#)
- [platSysmonNx](#)
- [platSysmonNxs](#)
- [platSysmonNxt](#)
- [platWifi](#)

### Components in platCrypto

- [CertManagerService](#)


#### **platCrypto-CertManagerService**

 The **CertManagerService** is the platform service of a station running on an AX-3.7 or later host, provided it is licensed with the `crypto` feature, and has the necessary modules installed (`platCrypto`, possibly others depending on build). It has few visible properties, but provides a default **Certificate Management** view, equivalent to that same-named platform 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.

**Note:** For complete details, refer to the document “NiagaraAX SSL Connectivity Guide”, including the section “About the Certificate Management dialog”.

#### **platform-DaemonSecureSession**

 **DaemonSecureSession** represents a *secure* platform connection (SSL or TLS) to a host made in Workbench, possible starting in AX-3.7. In the Nav tree view, the platform session icon is labeled **Platform**, shows a small “padlock”, and is directly under the host to which the platform session is in progress. To support such connections, the host must be licensed with the `crypto` feature, have the necessary installed modules, and have its “Platform SSL 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. For related details, see “[Niagara platform](#)” on page 1-1, including “[Platform overview](#)” on page 1-2 and “[About a platform connection](#)” on page 1-3.




**Note:** For details on SSL connections, refer to the document “NiagaraAX SSL Connectivity Guide”.

## Components in platDataRecovery module

- [DataRecoveryService](#)

### platDataRecovery-DataRecoveryService

 The DataRecoveryService automatically creates and manages buffers in a JACE controller’s available SRAM (Static RAM), allowing “battery-less” operation. It applies to these platforms:

- A JACE-3E series controller (based on the NPM3 processor, with built-in SRAM). This model was introduced in 2013, around the time of a NiagaraAX-3.7 update release (build 3.7.105).
- A JACE-6E series controller (based on the NPM6E processor, with built-in SRAM). This model was introduced in early 2012, around the time of a NiagaraAX-3.6.44 maintenance release.
- A JACE-603 or JACE-645 controller *configured for NiagaraAX* (and *not* Niagara R2). These “retrofit board” JACEs are also based on the NPM6E processor, and are supported in build 3.6.44 or later.
- A JACE-2,-6,-7 series or JACE-x02 Express (XPR or M2M) series controller, with an installed “SRAM option card”.


Starting in build 3.6.44, 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 run in the station’s PlatformServices container, operating independently or in unison, as configured.

For details, see the “About the DataRecoveryService” section in the Engineering Notes II document *JACE Data Recovery Service (SRAM support)*.

## Components in platGprs module

- [GprsPlatformService](#)
- [GprsHostSettings](#)
- [GprsRuntimeData](#)

### 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](#)” on page 1-32.

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 JACE that is equipped with a GPRS modem option. Access it under 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 \(latest\)](#)” 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 JACE 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 \(latest\)](#)” section in the Engineering Notes document “[GPRS modem option](#)”.


## Components in platform module

- [DaemonFileSpace](#)
- [DaemonSession](#)
- [LicensePlatformService](#)
- [NtpPlatformServiceLinux](#)
- [NtpPlatformServiceQnx](#)
- [NtpPlatformServiceWin32](#)
- [PlatformAlarmSupport](#)
- [PlatformServiceContainer](#)
- [SystemPlatformServiceQnxJavelina](#)
- [SystemPlatformServiceQnxNpm2xx](#)
- [SystemPlatformServiceQnxNpm6xx](#)
- [SystemPlatformServiceWin32](#)
- [TcpIpPlatformService](#)

### **platform-DefaultDaemonFileSpace**

 The Remote File System (DefaultDaemonFileSpace) represents the files accessible for read-only access when platform-connected to a remote NiagaraAX host. For more details, see [“Remote File System”](#) on page 1-82.

### **platform-DaemonSession**

 DaemonSession represents a platform connection to a host made in Workbench. In the Nav tree view, the DaemonSession icon is labeled **Platform**, and is directly under the host to which the platform session is in progress.


The default view is the **Nav Container View**, which provides a table of all the various platform views. For more details, see [“Niagara platform”](#) on page 1-1.

**Note:** *If using a simplified, profiled Workbench (without sidebar areas, including the Nav tree), a Workbench platform connection to a host is represented by a simple table of available platform views. Such a profiled Workbench may be used for a platform connection to JACE-603 or JACE-645 controller (“upgraded with a retrofit board”), to configure it for Niagara R2 operation. For related details, refer to the Retrofit Board Niagara R2 Install & Startup Guide.*


### **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. For more details, see [“Workbench License Manager”](#) on page A-2.

### **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 [PlatformServiceContainer](#). From the default plugin (view), you can perform the same operations as from the License Manager view using a platform connection. For more details, see [“License Manager”](#) on page 1-35.

### **platform-NtpPlatformServiceLinux**

 The NtpPlatformServiceLinux is the NiagaraAX interface to the NTP (Network Time Protocol) daemon of the Linux OS running on a Linux Supervisor. If enabled, it provides client and server support for NTP. The default view of this platform service is the [Ntp Platform Service Editor Linux](#) plugin, in which you can adjust a *few* settings, as well as specify time servers.


For more details, see [“About the NtpPlatformService”](#) on page 2-13.

### **platform-NtpPlatformServiceQnx**

 The NtpPlatformServiceQNX is the NiagaraAX interface to the NTP (Network Time Protocol) daemon of the QNX OS running on a JACE. 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.


For more details, see [“About the NtpPlatformService”](#) on page 2-13.

## **platform-NtpPlatformServiceWin32**

 The NtpPlatformServiceWin32 is the NiagaraAX 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”](#) on page 2-13.

## **platform-PlatformAlarmSupport**

 PlatformAlarmSupport is a container slot that appears for each alarmable value under a Platform Service, such as (for a QNX-based JACE), the [PowerMonitorPlatformServiceQnx](#), or for a JACE-NXS, the [HardwareMonitorNxsPlatformServiceWin32](#) (internal JACE-NXS temperature, etc.).

For a QNX-based JACE, 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.

For a JACE-NXT or JACE-NXS, example PlatformAlarmSupport components include:


- **Cpu Temperature Alarm Support**  
Under the HardwareMonitorService, to configure how “Cpu Temperature Fail” alarms are handled in the station.
- **Ups Not Found Alarm Support**  
Under the PowerMonitorService, to configure how “UPS not responding” alarms (if applicable) 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 provides a container for a station's PlatformService instances. The [Platform Service Containe rPlugin](#) is its primary view. The PlatformServiceContainer is available when online with any running station, under its Config, **Services** folder. For more details, see [“PlatformServiceContainer parameters”](#) on page 2-3.


## **platform-SystemPlatformServiceQnxJavelina**

 SystemPlatformServiceQnxJavelina is the QNX implementation of SystemPlatformService in a station running on a JVLN-based (JACE-700) controller. For more details, see [“SystemService \(under PlatformServices\)”](#) on page 2-8.


## **platform-SystemPlatformServiceQnxNpm2xx**

 SystemPlatformServiceQnxNmp2xx is the QNX implementation of SystemPlatformService in a station running on a NPM2-based JACE controller. For more details, see [“SystemService \(under PlatformServices\)”](#) on page 2-8.

## **platform-SystemPlatformServiceQnxNpm6xx**

 SystemPlatformServiceQnxNmp6xx is the QNX implementation of SystemPlatformService in a station running on a NPM6-based JACE controller. For more details, see [“SystemService \(under PlatformServices\)”](#) on page 2-8.


## **platform-SystemPlatformServiceWin32**

 SystemPlatformServiceWin32 is the Win32 implementation of SystemPlatformService. For more details, see [“SystemService \(under PlatformServices\)”](#) on page 2-8.

## **Reboot action**

Reboot action causes a system reboot. This is not available in Win32 systems if the platform authentication setting labeled “Stations - allow stations to have admin access to platform daemon” is disabled. See [Figure 1-49](#) on page 45 and the discussion following it for more details.


### platform-TcplpPlatformService

 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”](#) on page 1-72.

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. See [Figure 1-49](#) on page 45.

## Components in platHwScan

### platHwScan-HardwareScanService

 Starting in AX-3.7, 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-2/6 (200/600), JACE-6E, JACE-3E	platHwScanNpm
JACE-7 (700)	platHwScanJvln
JACE-603 (JACE-403 with retrofit board)	platHwScanJ603
JACE-645 (JACE-545 with retrofit board)	platHwScanJ645
JACE-x02 Express (202/602-XPR or M2M)	platHwScanXpr
JACE-NXS/NXT	platHwScanNx
Security JACE (201/601)	platHwScanSec

This default **Hardware Scan Service View** provides a diagram of the controller that shows its communication ports and other features (including, if applicable, installed option cards), and has callouts to a table that explains each item's location, description (such as COM2), port type, and status.


**Note:** For complete details, refer to the Engineering Notes 3.7 document “JACE Hardware Scan Service”.

## Components in platIEEE8021X


**Note:** IEEE 802.1X support is available only for QNX-based “Hotspot” JACEs (JACE-3,-6,-7 series) using AX-3.8.

- [IEEE8021XAdapterSettings](#)
- [IEEE8021XHostSettings](#)
- [IEEE8021XPlatformService](#)


### 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 Engineering Notes 3.8 document *NiagaraAX IEEE 802.1X Configuration*.

### 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](#) subcontainers. For complete details, refer to the Engineering Notes 3.8 document *NiagaraAX IEEE 802.1X Configuration*.

### platIEEE8021X-IEEE8021XPlatformService

 In AX-3.8, the **IEEE8021X Platform Service** is an available platform service on a JACE station, providing that the JACE host has the platIEEE8021X module installed and is licensed with the “ieee8021x” feature.

**Note:** *IEEE 802.1X support is available only for QNX-based “Hotspot” JACEs (JACE-3,-6,-7 series) using AX-3.8.*

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 JACE 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 Engineering Notes 3.8 document *NiagaraAX IEEE 802.1X Configuration*.

## Components in platPower module

- [JaceSlaBattery](#)
- [JavelinaBatteryPlatformService](#)
- [Npm2NimhBattery](#)
- [NpmDualBatteryPlatformService](#)
- [NpmExternalSlaBattery](#)
- [PowerMonitorPlatformServiceQnx](#)

### 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-JaceSlaBattery

● JaceSlaBattery is the “Battery” slot under the PowerMonitorService in a JACE-4 or JACE-5 series station's PlatformServices container. This slot simply indicates the host JACE platform uses a sealed lead acid type (SLA) 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 [PlatformServiceContainer](#).*

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

● Npm2NimhBattery is a “Battery” container slot under the PowerMonitorService in a JACE-2/6 series or JACE-x02 Express (NPM2 or NPM6) 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).

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, such as a JACE-x02 Express series (M2M JACE). 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 QNX-based JACEs. 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 [PlatformServiceContainer](#) in a station running on any QNX-based JACE *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 *JACE NiagaraAX Install & 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](#)” on page 2-10 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) *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](#)
- [SerialPortPlatformServiceQnx403](#)
- [SerialPortPlatformServiceQnx404](#)

### **platSerialQnx-SerialPortPlatformServiceQnx**

 SerialPortPlatformServiceQnx is the station’s interface to the platform’s serial port configuration, such as used by a JACE-2,-6,-7 series host. This service is found under the running station’s [PlatformServiceContainer](#) as the **SerialPortService**.

### **platSerialQnx-SerialPortPlatformServiceQnx403**

 SerialPortPlatformServiceQnx403 is the station’s interface to the platform’s serial port configuration, in this case specific to a JACE-403 series host. This service is found under the running station’s [PlatformServiceContainer](#) as the **SerialPortService**.

### **platSerialQnx-SerialPortPlatformServiceQnx404**

- SerialPortPlatformServiceQnx404 is the station's interface to the platform's serial port configuration, in this case specific to a JACE-545 series host. This service is found under the running station's [PlatformServiceContainer](#) as the **SerialPortService**.

### **platSerialQnx-SerialPortQnx**

- SerialPortQnx contains properties that describe how a serial port (RS-232 or RS-485) on a QNX-based JACE is being used in software as COM $n$ . 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 "dbgjmpr" (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**

- 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 COM $n$ . 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**

- 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 COM $n$ . 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**

- HardwareMonitorNxPlatformServiceWin32 (HardwareMonitorService) is the station's interface to internal environmental parameters in the host JACE-NX, such as CPU temperature, fan speeds, and various voltages. This service appears under the running station's [PlatformServiceContainer](#) as the **Hardware Monitor Service**.




See “Using platform services in a station” on page 2-10 for related details. For specific details, see the section “Hardware monitoring JACE-NX configuration” in the *JACE-NX NiagaraAX Install & Startup Guide*.

## Components in platSysmonNxs module

- [HardwareMonitorNxsPlatformServiceWin32](#)

### **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 “Hardware monitoring JACE-NXS configuration” in the *JACE-NXS NiagaraAX Install & Startup Guide*.

## Components in platSysmonNxt module

- [HardwareMonitorNxtPlatformServiceWin32](#)

### **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 “Hardware monitoring JACE-NXT configuration” in the *JACE-NXT NiagaraAX Install & Startup Guide*.

## Components in platUsbmon module


- [UsbMonitorPlatformServiceQnx](#)

### **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**

 (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 JACE. 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-700 series (JVLN) controller with a WiFi option. For general information, see “[WiFi Configuration](#)” on page 1-81.

**Note:** For complete details, refer to the Engineering Notes document “*NiagaraAX JACE WiFi option*”.



# CHAPTER 4

## Platform Plugin Guides

---

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.

### Plugin Reference Summary


Summary information is provided on views in the following modules:

- [platCrypto](#)
- [platDaemon](#)
- [platDataRecovery](#)
- [platDDNS](#)
- [platform](#)
- [platGprs](#)
- [platHwScan](#)
- [platPower](#)
- [platWifi](#)

### 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.

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.

For complete details, refer to the document *NiagaraAX SSL Connectivity Guide*, including the section "About the Certificate Management dialog".

**Note:** *Workbench 3.7 or later provides a similar view, via **Tools > Certificate Management**.*


*Also included is a related **Tools > Certificate Signer Tool** view. Refer to "About the Certificate Signing dialog" in the *NiagaraAX SSL Connectivity Guide* for related details.*

### Plugins in platDaemon module

- [Application Director](#)
- [Certificate Management](#)
- [Distribution File Installer](#)
- [Distribution View](#)
- [File Transfer Client](#)
- [Lexicon Installer](#)
- [License Manager](#)

- [Software Manager](#)
- [Software View](#)
- [PlatformAdministration](#)
- [R2 Platform Tool](#)
- [Station Copier](#)
- [TCP/IP Configuration](#)
- [User Manager](#)


### platDaemon-ApplicationDirector

 The Application Director is the platform view that allows you to start, stop, restart, and kill a station on the connected NiagaraAX platform. You also use it to examine standard *station output*, for troubleshooting and debug purposes.

For more details, see “[Application Director](#)” on page 1-13.


**Note:** *Starting in NiagaraAX build 3.6.44, the **Application Director** view also applies in a Workbench platform connection to a Niagara R2-configured JACE-603 or JACE-645 controller (retrofit board-equipped JACE-403 or JACE-545 controller). However, in this case there are some differences. For more details, refer to the “Application Director” section in the Retrofit Board Niagara R2 Install & Startup Guide.*

### platDaemon-DistInstaller


 The platform **Distribution File Installer** allows you to install distribution (.dist) files from this computer to the remote NiagaraAX host. Typical use is for restoring backups, or for installing a “clean dist” file to essentially restart configuration “from scratch.” For more details, see “[Distribution File Installer](#)” on page 1-22.

**Note:** *Starting in NiagaraAX build 3.6.44, the **Distribution File Installer** view also applies in a Workbench platform connection to a Niagara R2-configured JACE-603 or JACE-645 controller (retrofit board-equipped JACE-403 or JACE-545 controller). Chiefly, it is used to restore backups. For more details, refer to the “Distribution File Installer” section in the Retrofit Board Niagara R2 Install & Startup Guide.*

### 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 NiagaraAX platform, as needed. For more details, see “[File Transfer Client](#)” on page 1-30.

**Note:** *Starting in NiagaraAX build 3.6.44, the **File Transfer Client** view also applies in a Workbench platform connection to a Niagara R2-configured JACE-603 or JACE-645 controller (retrofit board-equipped JACE-403 or JACE-545 controller). One recommended use is to backup the Niagara R2 license installed on the controller. For more details, refer to the “File Transfer Client” section in the Retrofit Board Niagara R2 Install & Startup Guide.*


### platDaemon-LexiconInstaller

 Lexicon Installer allows you to install lexicon files (for Niagara localization) on a remote host. For more details, see “[Lexicon Installer](#)” on page 1-34.


**Note:** *Starting in AX-3.7, standard lexicons in Workbench are distributed as modules, for example: niagaraLexiconFr as the French lexicon, or niagaraLexiconDe for German. Workbench lexicon tools now 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. For more details, see the NiagaraAX Lexicons Guide.*

### platDaemon-LicenseManager


 License Manager allows you to view and install files required for Niagara licensing. For more details, see “[License Manager](#)” on page 1-35.

### platDaemon-PlatformSessionListView


 PlatformSessionListView is a tabular view of available platform views when platform connected to a host using a simplified “profiled” Workbench. This Workbench can be used to make platform connections to a *Niagara R2-configured* JACE-603 or JACE-645, using Workbench 3.6.44 or later.

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. For more details, see “[Software Manager](#)” on page 1-55.

### **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 “[Platform Administration](#)” on page 1-40.

**Note:** *Starting in build 3.6.44 or later of 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 (by removing Niagara R2). Refer to the Retrofit Board Niagara R2 Install & Startup Guide for complete details, including the section “Platform Administration”.*


### **platDaemon-R2PlatformTool**

-  Starting in build 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 NiagaraAX “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 JACE.


### **platDaemon-StationCopier**

-  The Station Copier is the platform view that you use to install a station in a remote NiagaraAX JACE host, as well as make a local backup copy of a remote NiagaraAX JACE station onto your Workbench PC. You can also delete stations using this view. For more details, see “[Station Copier](#)” on page 1-63.

### **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 for NiagaraAX, where this view is one step in the platform’s Commissioning Wizard.

For more details, see “[TCP/IP Configuration](#)” on page 1-72.

**Note:** *Starting in NiagaraAX build 3.6.44, the TCP/IP Configuration view also applies in a Workbench platform connection to a Niagara R2-configured JACE-603 or JACE-645 controller (retrofit board-equipped JACE-403 or JACE-545 controller). However, this view is not included as a wizard step—you must select it from the main platform view. For more details, refer to the “TCP/IP Configuration” section in the Retrofit Board Niagara R2 Install & Startup Guide.*

### **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 “[User Manager](#)” on page 1-77.

## Plugin in platDataRecovery

- [Data Recovery Service Editor](#)

### **platDataRecovery-DataRecoveryServiceEditor**

 The Data Recovery Service Editor is the default view on the **DataRecoveryService**, as found in the PlatformServices of the following JACE controllers:


- A JACE-3E series controller (based on the NPM3 processor, with built-in SRAM). This model was introduced in 2013, around the time of a NiagaraAX-3.7 update release (build 3.7.105).
- A JACE-6E series controller (based on the NPM6E processor with built-in SRAM). This model was introduced in early 2012, around the time of a NiagaraAX-3.6.44 maintenance release.
- A JACE-603 or JACE-645 controller *configured for NiagaraAX* (and *not* Niagara R2). These “retrofit board” JACEs are also based on the NPM6E processor, and supported in build 3.6.44 or later.
- A JACE-2,-6,-7 series or JACE-x02 Express (XPR or M2M) series controller, with an installed “SRAM option card”.

This view allows monitoring of the “battery-less” support provided by this service. Starting in build 3.6.44, 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 Platform-Services container, operating independently or in unison, as configured.

For details, see the “About the DataRecoveryService” section in the Engineering Notes II document *JACE Data Recovery Service (SRAM support)*.

## Plugin in platDDNS


### **platDdns-DdnsConfigurationView**

 DDNS Configuration allows for DNS IP addresses to be dynamically updated. For more details, see “[DDNS Configuration](#)” on page 1-21.


## Plugins in platform module

- [License Platform Service Plugin](#)
- [Ntp Platform Service Editor Linux](#)
- [Ntp Platform Service Editor Qnx](#)
- [Ntp Platform Service Editor Win32](#)
- [Platform Service Containe rPlugin](#)
- [Platform Service Properties](#)
- [System Platform Service Plugin](#)
- [System Platform Service Qnx Plugin](#)
- [TcpIp Platform Service Plugin](#)

### **platform-LicensePlatformServicePlugin**


 License Platform Service Plugin allows you to manage the host’s licenses and certificates under a station’s **PlatformServiceContainer**. It provides the same interface as the **License Manager** view in a platform connection. See “[License Manager](#)” on page 1-35.

### **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 “[About the Ntp Platform Service Editor Linux](#)” on page 2-17.


### **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 QNX-based JACE. This view provides access to a few related settings, plus allows specifying one or more remote time servers.




For more details, see [“About the Ntp Platform Service Editor Qnx”](#) on page 2-15.

### **platform-NtpPlatformServiceEditorWin32**

 Ntp Platform Service Editor Win32 is the default view of the station's [NtpPlatformServiceWin32](#), which provides the platform interface to the Windows Time service (W32Time) running on the host platform's Windows OS. This Windows service uses the SNTP (Simple Network Time Protocol) to synchronize to one or more designated time servers.

For more details, see [“About the Ntp Platform Service Editor Win32”](#) on page 2-14.


### **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. For related details, see [“PlatformServiceContainer parameters”](#) on page 2-3.


### **platform-PlatformServiceProperties**

 PlatformServiceProperties allows you to view and edit platform parameters on the host running the opened station, using a property sheet. See [“PlatformServiceContainer parameters”](#) on page 2-3.


### **platform-SystemDateTimeEditor**

 As an available view on a station's PlatformServiceContainer, the **System Date Time Editor** allows you to set the date, time, and time zone for the host running the station. For related time zone details, see [“Selecting a time zone in NiagaraAX”](#) on page B-2.


### **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). See [“SystemService \(under PlatformServices\)”](#) on page 2-8.


### **platform-SystemPlatformServiceQnxPlugin**

 System Platform Service Qnx Plugin allows you to view and edit platform parameters on any QNX-based JACE host running the station, and is the default view on the station's **SystemService** (SystemPlatformServiceQnx). See [“SystemService \(under PlatformServices\)”](#) on page 2-8.

### **platform-TcpIpPlatformServicePlugin**


 Tcp Ip Platform Service Plugin allows you to manage the host's TCP/IP settings under a station's [PlatformServiceContainer](#). It provides the same interface as the **TCP/IP Configuration** view in a platform connection. See [“TCP/IP Configuration”](#) on page 1-72.

### **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”](#) on page A-2.


## **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 JACE. For general details, see [“GPRS Modem Configuration”](#) on page 1-32.

**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 3.5.35 or later). It provides the identical interface as the platform view **GPRS Modem Configuration**. For general details, see [“GPRS Modem Configuration”](#) on page 1-32.

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 running AX-3.7 or later on most JACE models, 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.


**Note:** For complete details, refer to the Engineering Notes 3.7 document “JACE Hardware Scan Service”.

## Plugins in platIEEE8021X


**Note:** IEEE 802.1X support is available only for QNX-based “Hotspot” JACEs (JACE-3,-6,-7 series) using AX-3.8.

- [IEEE 8021X Adapter Settings Editor](#)
- [IEEE 802.1X Configuration](#)
- [IEEE 8021X Platform Service Plugin](#)

### 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 Engineering Notes 3.8 document *NiagaraAX IEEE 802.1X Configuration*.

### platIEEE8021X-IEEE8021XDaemonSessionPlugin

 The **IEEE 802.1X Configuration** view is an available platform view on an AX-3.8 JACE platform, providing that has the platIEEE8021X module 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 Engineering Notes 3.8 document *NiagaraAX IEEE 802.1X Configuration*.

**Note:** IEEE 802.1X support is available only for QNX-based “Hotspot” JACEs (JACE-3,-6,-7 series) using 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 Engineering Notes 3.8 document *NiagaraAX IEEE 802.1X Configuration*.

**Note:** IEEE 802.1X support is available only for QNX-based “Hotspot” JACEs (JACE-3,-6,-7 series) using 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](#)” on page 2-10.

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 QNX-based JACE models. 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](#)” on page 2-10.

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*.

## 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—at the time of this document, this means a JACE-700 series (JVLN) controller with a WiFi option. For general information, see “[WiFi Configuration](#)” on page 1-81.

**Note:** For complete details, refer to the Engineering Notes II document “NiagaraAX JACE WiFi 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 WiFi-equipped JACE, to discover and connect to a wireless 802.11 network. At the time of this document, this means a JACE-700 series (JVLN) controller with a WiFi option. For general details, see “[WiFi Configuration](#)” on page 1-81.

**Note:** 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](#)” on page 1-82.

This view is also a *secondary* view on a station’s **WifiPlatformService**, with the primary view the **WifiPlatformServicePlugin**.

**Note:** For complete details, refer to the Engineering Notes II document “NiagaraAX JACE WiFi option”.



# APPENDIX A

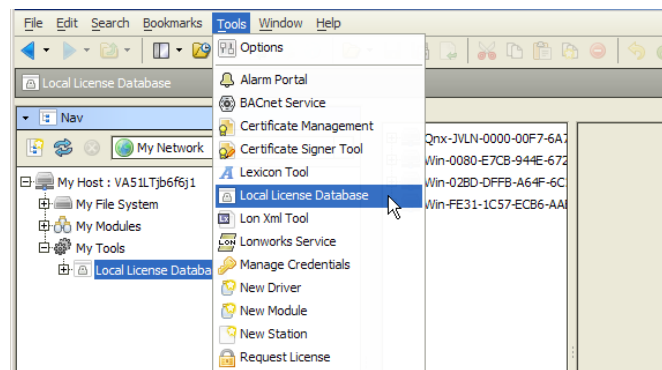
## License Tools and Files

This appendix provides details about the Workbench tools related to NiagaraAX 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 in [Figure A-1](#).
  - **Workbench License Manager** tool (see [“Workbench License Manager”](#) on page A-2)
  - **Request License** tool (see [“Request License”](#) on page A-5)

**Figure A-1** Tools menu in Workbench

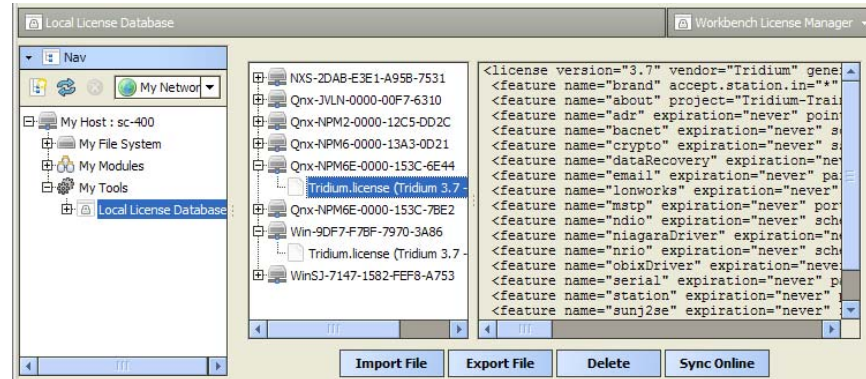


- **License management topics (in addition to Workbench License tools):**
  - [“About the local license database”](#) on page A-6
  - [“About license archive \(.lar\) files”](#) on page A-7
- [“About NiagaraAX license files”](#) on page A-8
  - [“Items common to all license files”](#) on page A-8
  - [“JACE hardware features”](#) on page A-9
  - [“Driver attributes”](#) on page A-10
  - [“Driver types”](#) on page A-11
  - [“Applications”](#) on page A-14

## Workbench License Manager

The **Workbench License Manager** view is available via the Workbench Tools menu, by selecting **Local License Database**.

**Figure A-2** Workbench License Manager



As shown in [Figure A-2](#), 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”](#) on page A-6.

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.

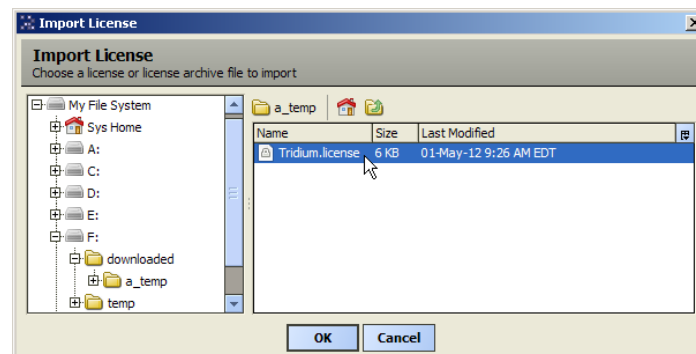
Buttons at the bottom of this view provide a way to manage the contents of your local license database, and are described as follows:

- **Import File** — Always available, this allows you to add license file(s) from a local license file or license archive (.lar) file.
- **Export File** — Always available, this allows you to save all licenses (or any selected licenses) locally, as a license archive file.
- **Delete** — This allows you to delete licenses from your Workbench local license database.
- **Sync Online** — 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

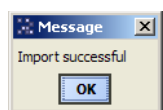
The **Import File** button in the **Workbench License Manager** is always enabled, and produces the **Import License** dialog for you to navigate to a source file (.license or .lar), as shown in [Figure A-3](#). Note only these two types of files appear for selection.

**Figure A-3** Import License dialog to find local license file or license archive file



Select a license file and click **OK** to add to (or update in) your local license database. A popup dialog ([Figure A-4](#)) informs you of success, and the license(s) are added or updated in your database.

**Figure A-4** Import success



**Note:** 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

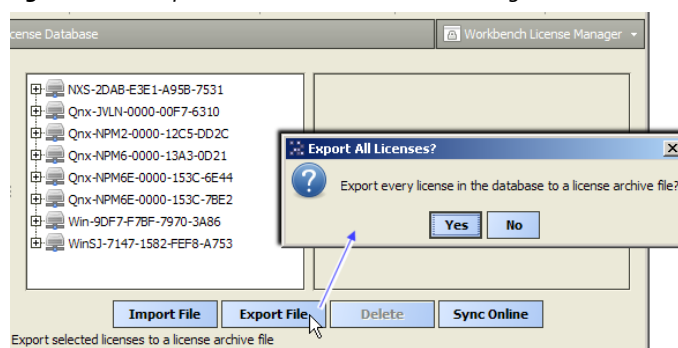
The **Export File** button in the [Workbench License Manager](#) allows you to save any number (or all) licenses in your local license database locally on your Workbench PC, as a license archive (.lar) file.

**Note:** The license archive format allows you to easily share saved .lar files (however named) among multiple Workbench 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 when in the platform [License Manager](#) (or similar License Platform Service Plugin).

For more details, see “[About license archive \(.lar\) files](#)” on page A-7.

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 [Figure A-5](#).

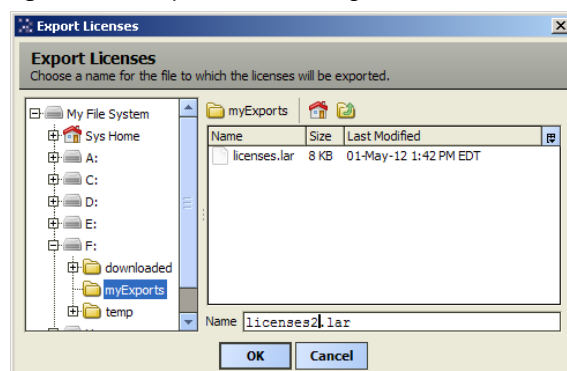
**Figure A-5** Export All Licenses confirmation dialog



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.

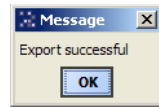
When you click **Yes** (if all) or **Export File** for selected licenses, an **Export Licenses** dialog ([Figure A-6](#)) lets you navigate to the spot to save the .lar file.

**Figure A-6** Export Licenses dialog



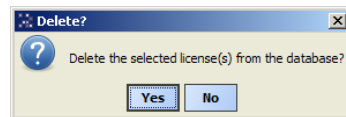
By default, a license archive file is saved in the root of your Niagara release directory. Use the dialog’s navigation controls to specify another target folder or drive, as needed. Before saving, you can also *rename* the license archive file, to make it more identifiable. For example, instead of: licenses.lar, you could rename it MyJaceNxt.lar.

Upon export of license(s) to a license archive file, a popup dialog appears, as shown in [Figure A-7](#).


**Figure A-7** 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 in [Figure A-8](#).

**Figure A-8** Delete licenses confirmation

Click **Yes** to delete the license(s), or **No** to leave the local license database unchanged.

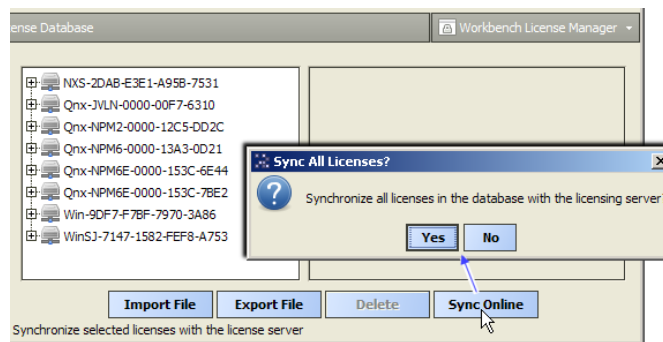
**Note:** Following a delete, you may need to click the  **Refresh** button in order to update the left pane contents. Note that if the selected “host ID” folder contained only a `.license` file, the entire folder is removed with a delete. However, if the folder contained other files (or subfolders), 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.

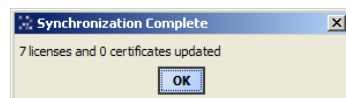
**Note:** For related details, see “[About the licensing server](#)” on page 1-39.

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 [Figure A-9](#).

**Figure A-9** 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 updated in your Workbench local license database. See [Figure A-10](#).

**Figure A-10** Synchronization Complete dialog

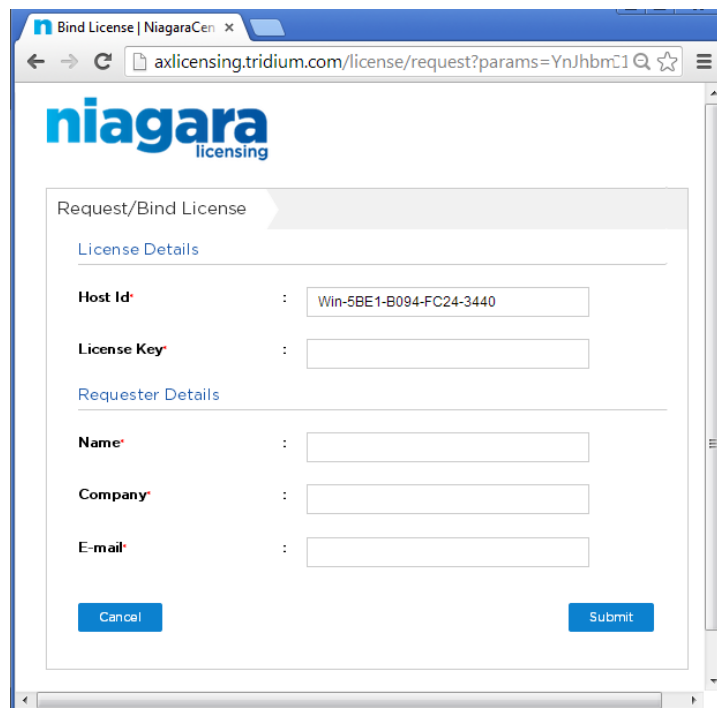
If all licenses (and certificates) were already up-to-date, this dialog will say “0 licenses and 0 certificates updated”.



## Request License

The “**Request License**” item on the Workbench Tools menu (Figure A-1 on page 1) simply opens a “Bind License form” in your Workbench PC’s default browser. By default, the only pre-filled field in this form is the host ID of your Workbench PC. See Figure A-11.

**Figure A-11** License request form in browser (from Workbench, Tools > Request License)



The screenshot shows a web browser window with the address bar displaying `axlicensing.tridium.com/license/request?params=YnJhbm...`. The page features the Niagara Licensing logo at the top. Below the logo, the form is titled "Request/Bind License". It is divided into two main sections: "License Details" and "Requester Details". In the "License Details" section, the "Host Id" field is pre-filled with the value "Win-5BE1-B094-FC24-3440", while the "License Key" field is empty. The "Requester Details" section contains three empty text input fields for "Name", "Company", and "E-mail". At the bottom of the form, there are two buttons: "Cancel" on the left and "Submit" on the right.

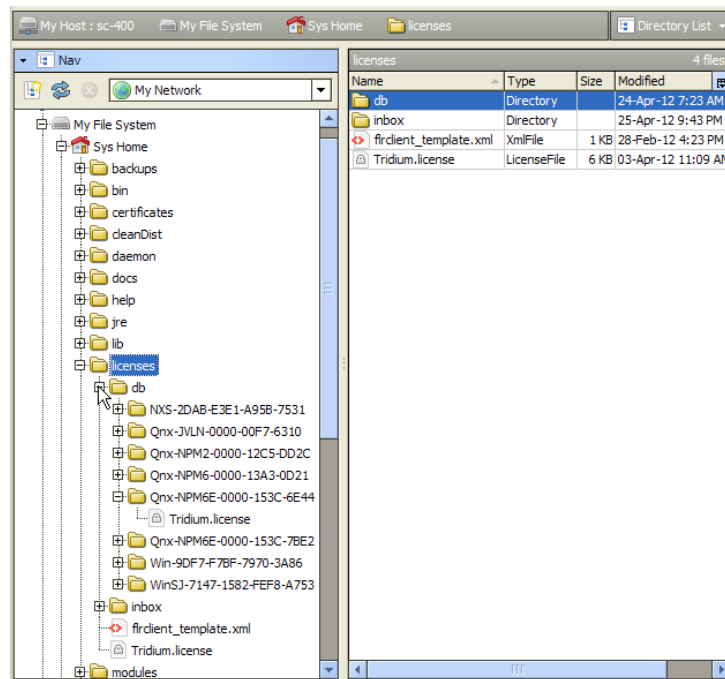
Typically, your Workbench PC is already licensed. Otherwise, you would not be able to successfully start Workbench, and then select **Request License** from the “Tools” menu.

However, you could use this as quick method to request a license for *another* PC on which you have installed NiagaraAX. In that case, you could substitute (type in) the host ID for the other PC in this form, along with other pertinent information.

## About the local license database

Any Workbench PC (including an Supervisor) has a “local license database” that is a *structured* collection of subdirectories and file under its Niagara release (system home) `! /licenses/db` directory. Each subdirectory has a unique NiagaraAX “host ID” name, matching that for some remote host platform. [Figure A-12](#) shows an example of this license database structure, as viewed in the Workbench Nav tree.

**Figure A-12** 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, PlatformService 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`.

For details on the Workbench License Manager, see [“Workbench License Manager”](#) on page A-2.

### Local license database rationale

The local license database design makes it easier for Workbench to store licenses for multiple host platforms—without inadvertently overwriting one license file with another. Unlike in older Workbench releases, you do not have to manually create 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 Workbench PCs, and is used in updating/synchronizing licenses to the online licensing server, as well as with provisioning features for Niagara Networks. See [“About license archive \(.lar\) files”](#) on page A-7.

### 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 Workbench computers, or perhaps sent to you from the licensing server.

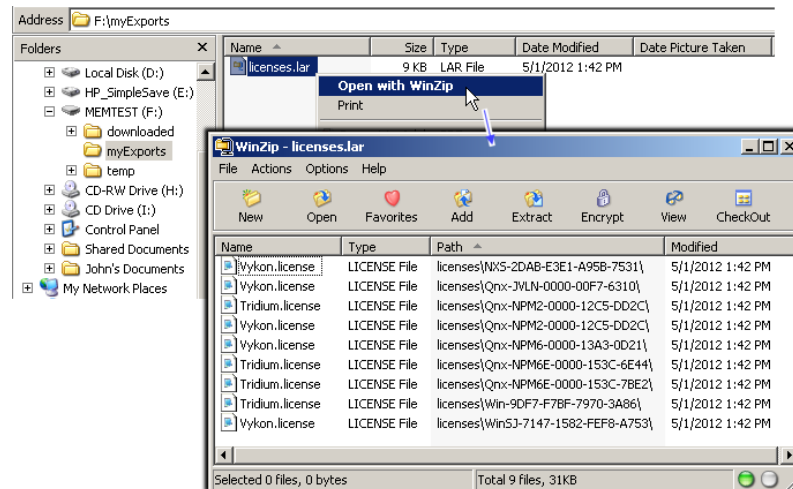
When you restart Workbench after copying license files and/or `.lar` files into your `inbox` subfolder, Workbench automatically creates the appropriate “host ID” named subfolders in your local license database, each containing the corresponding license file. Contents of the `inbox` folder are then deleted.

**Note:** If you upgraded Workbench (or an Supervisor) from a release prior to AX-3.3, this inbox feature is particularly useful. Simply copy all your license files you previously stored (from whatever locations) into your Workbench computer's `!licenses/inbox` folder, renaming files if necessary (to permit copying). 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” on page A-4.

## About license archive (.lar) files

When you use the platform [License Manager](#) view or Workbench's [Workbench License Manager](#) view (under 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/host ID`” folder (subdirectory) structure for any included licenses. See [Figure A-13](#) for an example.

**Figure A-13** License archive (.lar) is license file(s) in zip format, including folder paths relative to sys home.



[Figure A-13](#) shows a .lar file in Windows Explorer being opened using WinZip, 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 that one license.

### Licensing server use of license archives

The NiagaraAX licensing server also uses the “license archive” (.lar) format when it sends out license files, as a file attachment to emails.

- You can simply use the **Import File** command in your [Workbench License Manager](#) view (in your Tools menu) to copy in the `licenses.lar`. This adds or updates those licenses in your [local license database](#). You can then import licenses from your local license database when either platform (or station) connected.  
Or, if your Workbench PC is Internet-connected while using the platform [License Manager](#) (or a station's equivalent [PlatformServices](#), License Manager view) you can use the **Import** command and select “Import licenses from the licensing server,” which installs the updated license in the host platform, *and* updates your local license database at the same time.
- Alternatively, you can unzip the contents of the `licenses.lar` under your `!licenses` folder, and use the **Install File** command in the License Manager to install licenses.

## About NiagaraAX license files

A NiagaraAX 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 any license file that validates against the Tridium certificate:

- [Items common to all license files](#) (license, about, brand, signature)
- [JACE hardware features](#) (e.g. maxheap, 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](#) (crypto, eas, email, genericAppliance, provisioning, station, tenantBilling, 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 version="3.2" vendor="Tridium" generated="2007-04-11"
expiration="2008-03-31" hostId="Win-6827-91CB-C49A-6B4B" serialNumber="4856">
```

**version** version="3.7" - 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.

**vendor** vendor="Tridium" - This is always Tridium.

**generated** generated="2012-04-11" - The date upon which the license file was generated.

**expiration** expiration="2013-04-11" - The expiration date of the license file. After the expiration date the Workbench software will fail 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".

**hostId** hostId="Win-6827-91CB-C49A-6B4B" - Alphanumeric code unique to the specific host, which is generated upon installation of the NiagaraAX software on a Windows-based platform. QNX-based JACE controllers are assigned a hostId similar to this: hostId="Qnx-NPM2-0000-0E8F-2420". 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 QNX-based JACE 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.

#### 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" owner="Tridium" project="Tridium Testing"/>
```

**owner** owner="Tridium" - Optional attribute to designate the name of a person who is responsible for the project or possibly an end user.

**project** project="Tridium Testing" - Optional attribute to designate a project. This grouping should typically be assigned to all JACE controllers used for a particular project.

#### 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. The NiCS definitions are contained in the 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="brand" brandId="MyBrand" accept.station.in="*"
accept.station.out="*" accept.wb.in="*" accept.wb.out="*" />
```

**brandld** brandId="MyBrand" - Every licensed station and tool has a Brand Identifier (BrandID). This field holds a text descriptor that the OEM chooses as the identifier for its product line. Each station or tool can have only one BrandID entry.

**accept.station.in** accept.station.in="\*" - A list of brands that this local station will allow NiagaraAX 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 NiagaraAX 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".

### signature

This 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>MCwCFF0dq4wJcYgvhTVtrf0oSyuCDCWjAhRj+H9pNxQGStBnhEkIqK8rONB10g==</signature>
</license>
```

**Note:** All licenses for any JACE or Supervisor platform also require a "station" application feature, in order to run a local station. See the section "Applications" on page A-14 for related details.

### JACE hardware features

Alphabetically, JACE hardware license features listed here include [dataRecovery](#), [ibmj9j2me](#), [maxheap](#), [mstp](#), [ndio](#), [nr10](#), [serial](#), and [sunj2se](#).

#### dataRecovery

Starting in AX-3.6 for QNX-based JACE platforms, this feature licenses a station's DataRecoveryService, sourced from the JACE's platDataRecovery module. This service is required to support installed SRAM (Static RAM), whether integral "onboard SRAM" (such as for a JACE-6E or JACE-3E) or a different JACE model 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 a JACE-2 or JACE-6 series controller. In the absence of this feature, the maximum heap size is limited to 16 MB for a JACE-2, and 48 MB for a JACE-6. When this license feature is present, the maximum heap size is limited to 48 MB for a JACE-2 and 96 MB for a 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 a JACE-2 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 slots, if additional serial cards 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 *NiagaraAX NDIO Guide* for related details.

## nrrio

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 the onboard I/O of a JACE-x02 Express (M2M JACE) controller. Most QNX-based JACE controllers support NRIO modules (which communicate via RS-485). Refer to specific JACE controller data sheets to confirm whether this is an available option. Note that in the nrrio features line (below), a “device” equates to an “Nrio16Module”, and that history and schedule limits have no practical application. See Driver Feature Attributes for related details.

```
<feature name="nrrio" expiration="never" device.limit="16" history.limit="none"
  point.limit="none" schedule.limit="none" parts="DR-NRIO"/>
```

Refer to the *NiagaraAX 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" parts="AX-DEMO"/>
```

## 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. For simplicity, these common attributes are discussed first, and any unique attribute specific to a particular driver or service will be discussed in that [Driver types](#) section.

```
<feature name="driver name" expiration="expiration date" device.limit="none"
  history.limit="none" point.limit="none" schedule.limit="none" parts="AX-DEMO">
```

**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.limit`” and “`schedule.limit`” attributes have no importance in the feature line of those drivers.

## 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 NiagaraAX 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 NiagaraAX 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 Tridium internal use.

## Driver types

Each driver type is enabled by a separate feature element (or line, starting with *name* attribute), and has common [Driver attributes](#) as previously described.

**Note:** *New NiagaraAX 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](#), [bacnetws](#), [dust](#), [fileDriver](#), [jen6lp](#), [jennic](#), [lonworks](#), [modbusAsync](#), [modbusCore](#), [modbusSlave](#), [modbusTcp](#), [modbusTcpSlave](#), [obixDriver](#), [opc](#), [niagaraDriver](#), [rdbDb2](#), [rdbOracle](#), [rdbSqlServer](#), [sedonanet](#), [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 QNX-based JACE, 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" parts="AX-DEMO"/>
```

Refer to the *NiagaraAX BACnet Guide* for details on all BACnet integration with NiagaraAX.

**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) running AX-3.6 or later. The [bacnet](#) feature is also required in the license. More details are available in an appendix in the *NiagaraAX 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) running AX-3.6 or later. More details are available in an appendix in the *NiagaraAX BACnet Guide*.

**bacnetws**

(Superseded in favor of the BACnet AWS Supervisor and BACnet OWS Supervisor products that meet BTL certification, starting in AX-3.6). Provides added functionality as a *BACnet Supervisor* as described in the BACnet Specification B-OWS device profile, and is 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 *NiagaraAX 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 Niagara histories. For more details, see the “file-FileNetwork” section in the *NiagaraAX 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, Niagara 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 *NiagaraAX 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 *NiagaraAX 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 *NiagaraAX Obix Guide* for related details.

```
<feature name="obixDriver" expiration="never" device.limit="none" export="true"
history.limit="none" point.limit="none" schedule.limit="none" parts="AX-DEMO"/>
```

**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 *NiagaraAX OPC Guide* for related details.

## niagaraDriver

Enables communication via the Fox protocol to other NiagaraAX stations, 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 Niagara 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 NiagaraAX station 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 NiagaraAX station, 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" history-
Import="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"/>
```

To engineer and support, Niagara Workbench requires Sedona Framework TXS software installed, using the Sedona Installer tool—see the *NiagaraAX Sedona Installer Guide* for related details.

## snmp

Enables the SNMP (Simple Network Management Protocol) driver, which allows sending and receiving SNMP messages. Refer to the *NiagaraAX SNMP Driver Guide* for related details.

```
<feature name="snmp" expiration="never" device.limit="none"
history.limit="none" point.limit="500" schedule.limit="none" parts="AX-DEMO"/>
```

## videoDriver

Enables the NiagaraAX Video driver module (videoDriver) that provides the foundation to integrate selected commercial off-the-shelf video surveillance and recording systems into a NiagaraAX station. Depending on the specific video hardware used, one or more vendor-specific license feature entries are also typically required. Refer to the *NiagaraAX Video Driver Guide* for related details.

**zwave**

Applies to a QNX-based JACE 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 *NiagaraAX 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](#). However, applications [station](#), [web](#), and [workbench](#) have special importance, and are summarized first.

**station**

Enables a station to be run, and is typically present in any JACE platform, as well as a Supervisor.

```
<feature name="station" expiration="2014-03-31" resource.limit="none" guestEnabled="true" parts="AX-DEMO"/>
```

The station feature may not be present in a license for an engineering workstation (PC), unless specifically ordered with it.

**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 on SoftJACE license files.

**guestEnabled** `guestEnabled="true"` - (AX-3.8 only) 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" in AX-3.8 can enable and use the guest user—thus it 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 an HTTP connection). If not licensed, the server is set to fault with appropriate `faultCause`. If the feature is enabled, then WebServlets and spy pages are always available.

**Note:** *Full Workbench can connect to a station (via Fox connection) even if the web feature is disabled.*

```
<feature name="web" expiration="never" ui="true" ui.wb="true" ui.wb.admin="true" parts="AX-DEMO"/>
```

**ui** `ui="true"` - If the `ui` flag is false, then all access to ServletViews via `/ord` are disabled (with exception to spy pages).

**ui.wb** `ui.wb="true"` - If the `ui.wb` flag is false, then all user accounts using an `WbProfile` are disabled (including thin client views, but with exception to spy pages). Only thin client accounts operate when `ui.wb` is false; other accounts do not automatically degrade. Views which aren't licensed return 403 Forbidden.

**ui.wb.admin** `ui.wb.admin="true"` - If the `admin` flag is false, then all views which have an `admin` flag set in their required permissions are unavailable using the `wb` applet.

**workbench**

The workbench feature must be present to start the full version of Workbench (for example, a copy of Tridium's 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 a SoftJACE.

```
<feature name="workbench" expiration="never" admin="true" parts="AX-DEMO"/>
```

**box**

This enables an AX-3.7 or later 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 *any* NiagaraAX release since AX-3.3, 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 in AX-3.6 to AX-3.3 releases** 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** `allE2Reprot="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

## email

This features enables the NiagaraAX station to send email messages to an SMTP server. If the feature is not present then the service and all its incoming and outgoing accounts are marked as fault. No email can be sent or received if not licensed.

```
<feature name="email" expiration="never" parts="AX-DEMO"/>
```

## fips140-2

This features enables an AX-3.8 NiagaraAX 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="2014-03-13" 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

This feature enables an 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 an AX-3.8 host 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"/>
```

Refer to the *NiagaraAX LDAP / Active Directory Guide* for complete details.

## mobile

This enables an AX-3.7 or later host to support the NiagaraAX 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 NiagaraAX host provisioning, typically used to automate routine maintenance of a NiagaraAX 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" parts="AX-DEMO"/>
```

## 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"/>
```

## sox

Required by a Workbench host to open a Sox connection to a Sedona Framework device. Sox is the native protocol used to either program the App in the device, or to perform device provisioning.

```
<feature name="sox" 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.

# APPENDIX B

## Time Zones and NiagaraAX

---

Platform configuration of a NiagaraAX 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 appendix provides details on time zone selection in all revisions of NiagaraAX, including the currently used “historical time zone database.”

The following main sections are included:

- [“Time zones and terminology”](#) on page B-1
- [“Selecting a time zone in NiagaraAX”](#) on page B-2
- [“About the historical time zone database”](#) on page B-2 (current NiagaraAX releases)
- [“About timezones.xml”](#) on page B-4 (pre-AX-3.3)

**Note:** *Workbench provides a special “Time Zone Database Tool” that lets you explore the historical time zone database on the local Workbench host. For details, refer to the User Guide section on the “Time Zone Database Tool.”*

### 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).

#### 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 not 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).

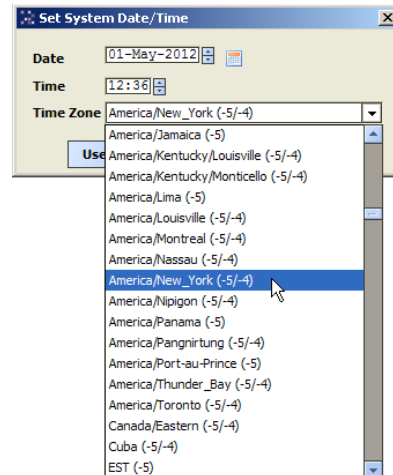
**Note:** *A change in DST rules for a time zone can cause issues in NiagaraAX when displaying historical data (histories and alarm records), particularly when applying new (current) DST rules to records collected using prior (old) DST rules. These issues have been overcome using a “historical timezones” mechanism. For more details, see [“About the historical time zone database”](#) on page B-2.*



## Selecting a time zone in NiagaraAX

Since the first NiagaraAX release, platform configuration of a NiagaraAX host includes the setting of date and time, which includes specifying its local time zone. Typically, this is done using either a platform connection and the **Platform Administration** view (function “[Change Date/Time](#)”), or possibly after a station is installed and running, using the one of the station’s [PlatformServices](#) views (“Platform Service Container Plugin” or “System Date and Time Editor”).

**Figure B-1** Selecting time zone from *Change Date/Time* selection in Platform Administration View



Time zones appear on the selection list using a “Zone ID (hours UTC offset/hours [DST + UST] offset)” format. Some examples:

```
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 available time zones is from a “*time zone database*” that *resides on that host*. For any host running AX-3.3 or later, this is a “historical database”, sourced in the file `!lib/timezones.jar`.

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. For more details, see “[About the historical time zone database](#)” on page B-2.

**Note:** *Host platforms running pre-AX-3.3 builds use a `!lib/timezones.xml` file instead. In this database file, for each known time zone there is one definition, starting with its zone ID. If necessary, you can edit the definition of one or more time zones in this `timezones.xml` file (or replace with an edited version of this file). Note although AX-3.3 or later hosts also have a `timezones.xml` file, it is not used locally. See the section “[About timezones.xml](#)” on page B-4 for more details.*

## About the historical time zone database

Starting in AX-3.3, any NiagaraAX host uses a time zone database with “historical perspective,” such that display of a station’s timestamped data (histories and alarms) collected in time zones under “prior rules” (typically DST-related) will always display with the original (and correct) collected time.

### Issue example and fix

You have a JACE that has been running since AX-3.0, installed in a U.S. time zone that observes DST. Included in the AX-3.0 builds was a `timezones.xml` with “old” DST rules. See the “[DST](#)” section for related details. On it you have a station running, that includes an original history for a boolean point with two records like this:

```
01-Nov-06 8:00 AM EST {} {ok} On
01-Nov-06 5:00 PM EST {} {ok} Off
```



The timestamp values shown are correct, this was a piece of equipment that came ON at 8:00am and OFF at 5:00pm, and on November 1st of 2006 it was indeed “Eastern Standard Time” (EST), as Daylight Savings Time had already ended on the last Sunday in October.

Since, you upgraded the JACE to AX-3.2, which uses updated 2007 rules in its `timezones.xml` file. Afterwards, when looking at the same two records for the example history, you see:

```
01-Nov-06 9:00:11 AM EDT {} {ok} On
01-Nov-06 6:00:13 PM EDT {} {ok} Off
```

These timestamp values are *incorrect*, as the DST offset of 1 hour was “retroactively” applied using the new DST rules (even though these rules were not actually in effect at this time). However, those same new rules are required during this period in 2007 to correctly change the system clock and maintain accurate 2007 timestamps.

After upgrading the JACE to a more current NiagaraAX release, which uses the `timezones.jar` (historical time zone database) instead of `timezones.xml`, when you look again at the same two records for the example history, you see:

```
01-Nov-06 8:00:11 AM EST {} {ok} On
01-Nov-06 5:00:13 PM EST {} {ok} Off
```

The timestamp values are correct, as the historical rules were used instead of the current (new) rules.

### **Parts of the historical time zone database**

The historical time zone database is sourced from the public “Olson Time Zone Database,” and updated (synchronized to it) upon each NiagaraAX build. It is implemented using a “`timezones.jar`” file that contains histories of changed rules for any so-affected timezones, along with changes to the NRE (Niagara Runtime Environment) that support this new method.

There are 2 associated files with historical time zones in an AX-3.3 or later 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 NiagaraAX, 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, unlike with the earlier `timezones.xml` database.

### **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-3.3 or later JACE, you can simply *transfer* the newer `timezones.jar` file to the host’s `!lib` folder.

**Note:** *As when making changes to the `timezones.xml` file in an earlier rev. host (pre-AX-3.3), after updating the historical time zone database files, you must restart the station on that host for the updated time zone database to become effective.*

## About timezones.xml

The `timezones.xml` file used by pre-AX-3.3 hosts provides an easily read and edited *single* definition for each time zone, where all values are contained within quotes *"value"*.

For example, consider the following entry for one time zone:

```
<zone id="America/Anchorage" utcOffset="-9h">
  <display name="Alaska Standard Time" short="AKST" dstName="Alaska Daylight Time" dstShort="AKDT"/>
  <dst savings="1h">
    <start time="2:00" weekday="sunday" month="march" week="second"/>
    <end time="2:00" week="first" weekday="sunday" month="november"/>
  </dst>
```

where attributes and example values are:

- `zone id="America/Anchorage"` : Name in drop-down selection list when picking a time zone.
- `utcOffset="-9h"` : Time added to UTC for this zone, typically in (h)ours, may be negative, positive, or 0 (i.e. none).
- `display name="Alaska Standard Time"` : Seen somewhere (but where?) when DST *not* in effect.
- `short="AKST"` : Seen within timestamps (e.g., alarms and histories) when DST *not* in effect.
- `dstName="Alaska Daylight Time"` : Seen somewhere (but where?) when DST *is* in effect.
- `dstShort="AKDT"` : Seen within timestamps (e.g., alarms and histories) when DST *is* in effect.
- `dst savings="1h"` : All dst parameters exist only if the time zone has DST. If so, the savings attribute is the DST adjustment amount, typically "1h" (1 hour), in very few time zones is "0.5h".
  - `start time="2:00" weekday="sunday" month="march" week="second" /` : Specifies the time and day when DST begins, often with "week, weekday, month" method (that is, "Nth weekday", as shown here).
  - `end time="2:00" week="first" weekday="sunday" month="november" /` : Specifies the time and day when DST ends, often with a "week, weekday, month" method (that is, "Nth weekday", as shown here).

Note that there are [DST start and end syntax variations](#) different from the ones above.

### DST start and end syntax variations

Sometimes a time zone with [DST](#) will use a *different* syntax to specify a DST start or stop day, versus the "Nth weekday" syntax shown in the previous [timezones.xml](#) example.

For example, the Baghdad time zone definition uses day ("Exact day") instead of "Nth weekday" to specify a particular date, in this case for the 1st of the month:

```
<zone id="Asia/Baghdad" utcOffset="3h">
  <display name="Arabia Standard Time" short="AST" dstName="Arabia Daylight Time" dstShort="ADT"/>
  <dst savings="1h">
    <start time="3:00 standard" month="april" day="1"/>
    <end time="3:00 standard" month="october" day="1"/>
  </dst>
</zone>
```

A variation is where weekday and day are both used (but *not* week) and the day numeric value includes trailing . . . to indicate "on or after (that day number)", as in time zone definition below. In this case, start time is evaluated as "on the Friday on or after the 26th of March."

```
<zone id="Asia/Tel_Aviv" utcOffset="2h">
  <display name="Israel Standard Time" short="IST" dstName="Israel Daylight Time" dstShort="IDT"/>
  <dst savings="1h">
    <start time="2:00" weekday="friday" month="march" day="26..." />
    <end time="2:00" month="september" day="13"/>
  </dst>
</zone>
```


A similar "on or before" variation can be specified using a leading . . . before a day number, for example, `day="...12"`. Of course many time zones do not even observe DST, so their definitions are noticeably shorter, for example the following one for Honolulu:

```
<zone id="Pacific/Honolulu" utcOffset="-10h">
  <display name="Hawaii Standard Time" short="HST"/>
</zone>
```

### Using Workbench to edit and transfer timezones.xml

Remember the [timezones.xml](#) used by any Niagara AX host (JACE) is the one in its *own !lib* folder, so to change it you typically edit a *local copy* (on your Workbench PC) and then *transfer this file* to that location on the remote JACE, using the platform view File Transfer Client.

The following is a quick summary of how to do this using Workbench:

- Step 1 In the Workbench nav tree, expand **My File System** and then **Sys Home, lib** folder.
- Step 2 In the **lib** folder, double-click `timezones.xml` to open it in the Text File Editor.
- Step 3 Make whatever changes are needed in the time zone(s) you will pick from, and click the Save tool  when done.
- Step 4 In Workbench, open a platform connection to the target JACE, and select the File Transfer Client.
- Step 5 In the File Transfer Client, open both (left and right) sides to list files in the **lib** folder.
- Step 6 Transfer your locally edited `timezones.xml` (left side) to the JACE on the right. A "Replace File?" popup dialog will appear, showing a different CRC value. Click **Yes** to transfer the file.  
The time zone database is now updated in the JACE.

**Note:** *Any change to `timezones.xml` is not effective until after the station is restarted.*



# APPENDIX C

## 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 in AX-3.3. This appendix provides details about how to configure and establish tunneled platform connections.

**Note:** 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” on page C-1
- “Supervisor configuration to support platform tunneling” on page C-2
- “Platform tunneling usage” on page C-3
- “Notes on platform tunneling” on page C-5
- “SSL considerations for platform tunneling” on page C-5

### Platform tunneling overview

Platform tunneling lets you make a Workbench 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.

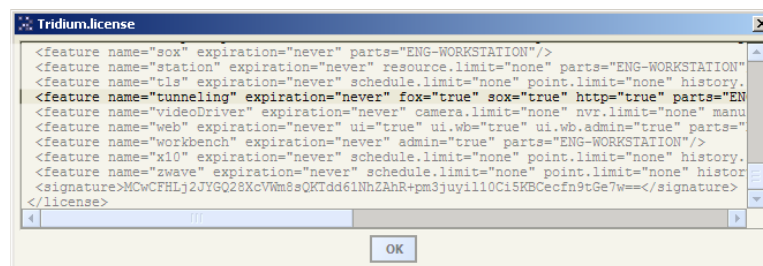
Key points to remember include:

- 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. See [Figure C-1](#).

**Figure C-1** 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

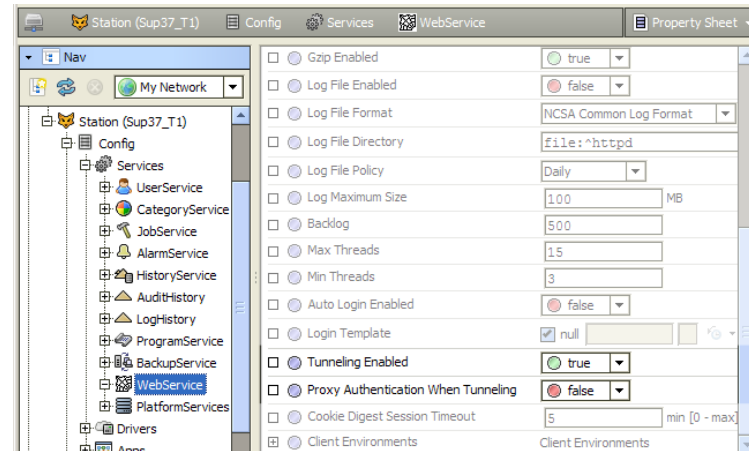
On the Supervisor station, these areas are involved:

- [WebService settings for platform tunneling](#)
- [FoxService settings for platform tunneling](#)

### WebService settings for platform tunneling

In the Supervisor's **WebService** (under **Config**, **Services**), ensure that the two properties below are as shown in [Figure C-2](#).

**Figure C-2** WebService in Supervisor to support platform tunneling



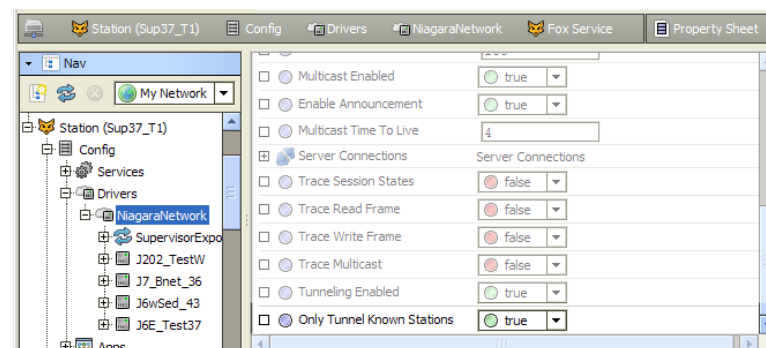
- **Tunneling Enabled**  
Set to true. If Fox and HTTP tunneling are already enabled, this should already be enabled.
- **Proxy Authentication When Tunneling**  
Set to false. In the initial AX-3.5 implementation of platform tunneling, proxy authentication is not supported. The only login authentication used in a tunneled platform connection are the tunnel endpoint's (JACE's) platform credentials.  
*Note: This property was first introduced in AX-3.4, and for an existing job upgraded from AX-3.4 to AX-3.5 may be true. To permit platform tunneling, you must set it to false.*

### FoxService settings for platform tunneling

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 highlighted in [Figure C-3](#).

**Figure C-3** Supervisor's FoxService, showing "Only Tunnel Known Stations" property



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`. This property was first introduced in AX-3.4. Depending on its setting, this affects the ability to open a tunneled platform connection, including how you enter the target (JACE) 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) of that JACE.
  - 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 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.

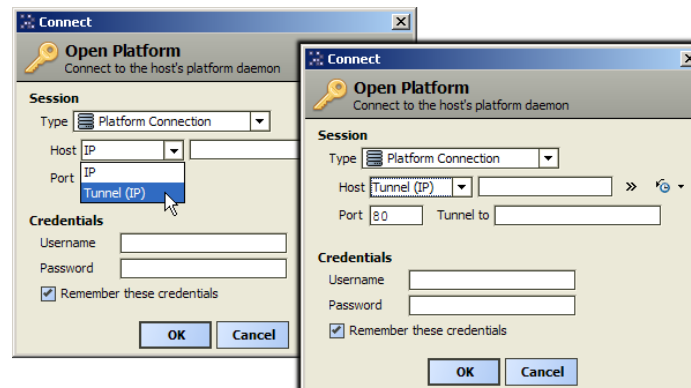
## Platform tunneling usage

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 (Ctrl + Shift + P), also available from the **File > Open** submenu.

### Open Platform dialog if tunneling

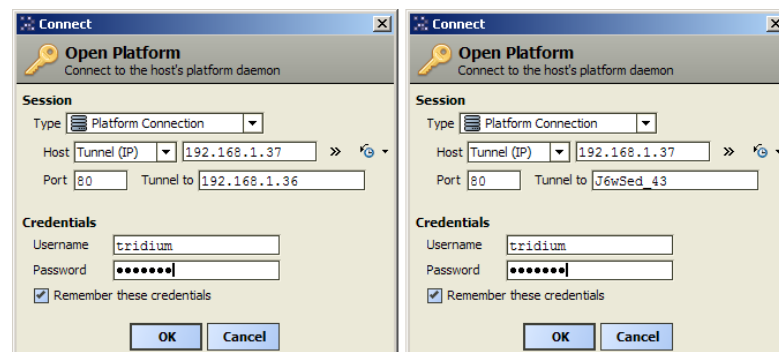
Click the drop-down control  in the Host field and select **Tunnel (IP)**

**Figure C-4** Select Tunnel (IP) in Open Platform dialog



In the top **Host** Tunnel (IP) field, enter the IP address or hostname of the Supervisor (the tunnel proxy server). In the example shown below, the Supervisor is at 192.168.1.37 IP address.

**Figure C-5** Standard port 80 on Supervisor, “Tunnel to” depends on “Only Tunnel Known Stations”



- In the **Port** field, enter the configured Http Port property value in the Supervisor's WebService, where 80 is the standard (default) value.
- 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. (Figure C-5, left), or
  - *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. (Figure C-5, right).For related details, see “FoxService settings for platform tunneling” on page C-2.



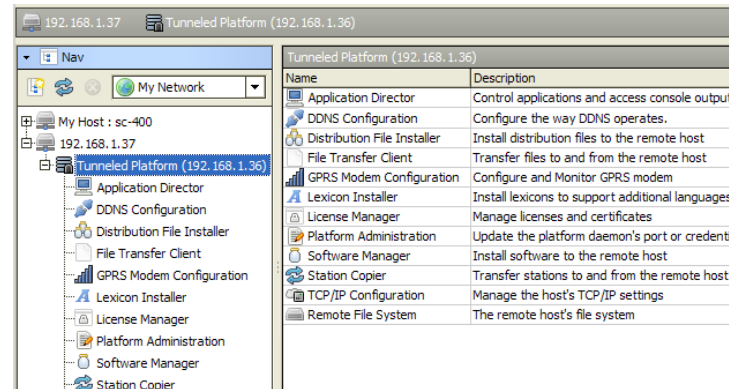
**Note:** If the endpoint JACE platform is using a “non-standard” HTTP port for the platform daemon, meaning not port 3011, append a colon (:) and that port number on the “Tunnel to” value. For example:

- 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 in Workbench

When connected, 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”). See [Figure C-6](#).

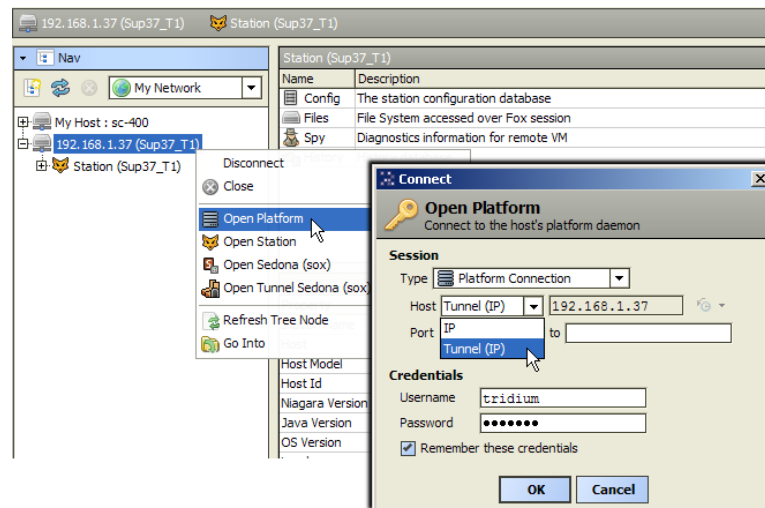
**Figure C-6** Tunnel-connected platform example showing IP address of endpoint JACE



Once connected, you can perform all the same platform operations as if directly platform connected to the endpoint JACE, including running the Commissioning Wizard. See “[Types of platform views](#)” on page 1-5.

Note 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 in [Figure C-6](#)

**Figure C-7** Right-click remote Supervisor for Open Platform menu item



Enter values the same way as previously described (see “[Open Platform dialog if tunneling](#)” on page C-3).

**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 necessarily 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—see [“Change SSL Settings”](#) on page 1-47.) 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*.

