

Technical Document

NiagaraAX Provisioning Guide for Niagara Networks

Niagara^{AX-3.5--AX-3.7}

Updated: August 4, 2012



Niagara^{AX} Provisioning Guide for Niagara Networks

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^{AX} Framework, and Sedona Framework are registered trademarks, and Workbench, WorkPlace^{AX}, and ^{AX}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 © 2012 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

Preface	v
About this document	v
Provisioning FAQs	vi
Provisioning terms	vi
Document Change Log	vii
 Provisioning Niagara Installation.....	 1-1
Provisioning license considerations	1-1
Provisioning software installation	1-1
 Provisioning Niagara Quick Start.....	 2-1
Add and configure necessary components	2-1
Add the Batch Job Service and ProvisioningNwExt	2-1
<i>To add the Batch Job Service and ProvisioningNwExt</i>	<i>2-1</i>
Configure the NiagaraDevice extensions	2-2
<i>To configure the NiagaraStation Platform Connection extensions</i>	<i>2-2</i>
Run provisioning jobs	2-3
Run custom Niagara provisioning jobs	2-3
<i>Synchronizing Supervisor license database and updating JACE licenses</i>	<i>2-3</i>
Add custom persisted Niagara provisioning jobs	2-3
<i>Example Supervisor configured for "Job Prototype" backups of stations</i>	<i>2-4</i>
 Niagara ProvisioningNwExt Concepts	 3-1
Niagara Provisioning overview	3-1
About the ProvisioningNwExt	3-2
Niagara Network Job Builder	3-2
Initial (run once) steps List	3-3
Steps for each station List	3-3
Adding Job List Steps	3-3
Removing Job List Steps	3-3
Reorder Job List Steps	3-3
Stations List	3-4
Adding Stations	3-4
Removing Stations	3-4
Reorder Stations	3-4
Alarm checkboxes	3-5
Buttons	3-5
Niagara Network Job List	3-5
Jobs Table	3-5
Jobs table columns	3-6
Buttons	3-6
Job Log notes	3-6
Niagara Network Job View	3-7

Job elements (read-only)	3-7
Step summary table	3-8
Buttons	3-8
Batch Job Log File View	3-9
Batch Job Step Log File View	3-9
Step elements (read-only)	3-10
Step log table	3-10
Buttons	3-10
Job Step Details View	3-11
ProvisioningNwExt slots	3-12
ProvisioningNwExt properties	3-12
ProvisioningNwExt container components	3-12
ProvisioningNwExt action	3-13
<i>Why the Start Backup action is typically NOT recommended</i>	3-13
Supervisor Software container	3-14
Software tree	3-14
Details pane	3-14
Buttons	3-14
Licenses slot	3-16
Supervisor License Manager	3-17
About the Niagara Network Job Prototype	3-17
Scheduling a job prototype	3-18
Job retention	3-19
Job actions	3-20
Niagara Network Prototype View	3-21
Initial (run once) steps List	3-21
Steps for each station List	3-21
<i>Adding Job List Steps</i>	3-21
<i>Removing Job List Steps</i>	3-22
<i>Reorder Job List Steps</i>	3-22
Stations List	3-22
<i>Adding Stations</i>	3-22
<i>Removing Stations</i>	3-23
<i>Reorder Stations</i>	3-23
Alarm checkboxes	3-23
Buttons	3-23
Prototype Job List	3-24
Jobs Table	3-24
Buttons	3-24
Niagara provisioning job steps	3-25
Update Licenses step	3-26
Backup Stations step	3-26
Copy Supervisor File step	3-26
Copy Local File step	3-27
Install Software step	3-28
Reboot step	3-29
Run Robot step	3-30
Upgrade Out-of-date Software step	3-30
Provisioning Robot notes	3-30
Provisioning Robot Editor	3-30
Run Robot step view features	3-32
<i>Robot Source</i>	3-32
<i>Robot Log</i>	3-32
Niagara Provisioning Job management	3-33
BatchJobService	3-33
<i>BatchJobService properties</i>	3-33
Job execution	3-34
Batch Job log files	3-34

Provisioning Niagara station data files	3-34
Histories related to provisioning	3-35
About provisioning-related alarms	3-35

Provisioning Station Extension Concepts 4-1

Types of provisioning extensions	4-1
Provisioning Manager	4-2
Extension Access in Provisioning Manager	4-2
Edit in Provisioning Manager	4-2
Platform Connection	4-3
Properties	4-3
Ping action	4-4
New station notes	4-4
StationProxy	4-5
Properties	4-5
Actions	4-6
Views	4-7
Provisioning Station Director	4-7
Station Job List	4-8
Step Table	4-8
Step table columns	4-8
Buttons	4-9
Software	4-9
Properties	4-9
Station Software View	4-10
Backup	4-11
Properties	4-11
Start Backup action	4-11
Licenses	4-12
Properties	4-12
Poll action	4-12

Provisioning Plugin Guides 5-1

Plugin Reference Summary	5-1
Plugins in batchJob module	5-1
Plugins in provisioningNiagara module	5-2

Provisioning Niagara Component Guides 6-1

Component Reference Summary	6-1
Components in batchJob module	6-1
Components in provisioningNiagara module	6-1

Converting from old style (AX-3.1) Provisioning A-1

Conversion- Supervisor previously running AX-3.1/AX-3.2 style provisioning ..	A-1
Run the conversion to new style Provisioning	A-1
<i>To convert the Supervisor to use the newer style provisioning</i>	A-1
Post conversion notes	A-2

CONTENTS

Preface

This document explains provisioning for a NiagaraNetwork in a Supervisor running AX-3.7, AX-3.6, or AX-3.5. For the most part, provisioning is identical in these three releases, with a few AX-3.7 differences noted. In addition, the basic provisioning model for NiagaraNetworks has remained unchanged since AX-3.3. See [“Provisioning FAQs”](#) later in this preface for a summary of related changes.

Note: *Before AX-3.3, provisioning was significantly different, using a different station architecture and NiagaraAX module (provisioning, vs. the current provisioningNiagara). Details about provisioning in AX-3.1 or AX-3.2 releases are in a different document: the NiagaraAX Provisioning Guide. However, note that this document has an appendix that covers conversion of a Supervisor that was previously running the older type of provisioning, and has now been upgraded to AX-3.5 or later. See the section [“About this document”](#) for a listing of the main sections in this document.*

This preface has the following sections:

- [About this document](#)
- [Provisioning FAQs](#)
- [Provisioning terms](#)
- [Document Change Log](#)

About this document

As noted earlier in this [Preface](#), this document applies to Niagara provisioning in AX-3.7, AX-3.6, or AX-3.5, and has the following main sections:

- [Provisioning Niagara Installation](#)
Explains the NiagaraAX platform, software, and licensing requirements.
- [Provisioning Niagara Quick Start](#)
Provides several quick procedures for online Supervisor station configuration to add the ProvisioningNwExt to the station’s NiagaraNetwork, configure provisioning NiagaraStation extensions, and begin typical provisioning tasks.
- [Niagara ProvisioningNwExt Concepts](#)
Provides concepts behind the ProvisioningNwExt in the Supervisor’s NiagaraNetwork, including its special views, properties, and various container slots. Many other concepts are also explained, including the available “job prototype” components and their views, individual Niagara provisioning steps, the ProvisioningRobot, provisioning job management, and provisioning-related alarms.
- [Provisioning Station Extension Concepts](#)
Provides concepts behind *other* provisioning-related components under the NiagaraNetwork in the Supervisor, including their special views and properties. These components are automatically created as *device extensions* under NiagaraStation components.
- [Provisioning Plugin Guides](#)
Provides brief summaries of the various provisioning views, each with links back to the more detailed concepts sections. Entries are used in NiagaraAX context-sensitive help “On View”.
- [Provisioning Niagara Component Guides](#)
Provides brief summaries of the different provisioning components, with links back to the more detailed concepts sections. Entries are used in NiagaraAX context-sensitive help “Guide On Target”.
- Appendix A - [Converting from old style \(AX-3.1\) Provisioning](#)
Provides an upgrade procedure to convert the older style ProvisioningService-based station of an upgraded (AX-3.1 or AX-3.2) Supervisor to the newer and more flexible “NetworkExt” model used for provisioning used since AX-3.3.

Provisioning FAQs

Below are some frequently asked questions (FAQs) about the provisioningNiagara module:

Q: What is meant by provisioning Niagara?

A: Provisioning Niagara means an administrator can configure Supervisor automation of:

- one or more pre-defined tasks
- against one or more (potentially many) stations in its Niagara Network
- done from the Supervisor
- in a way where results are recorded and can be referred to later.

Tasks would otherwise need to be performed manually by a user with Workbench, often by making platform connections to JACEs. All provisioning work is performed by the Supervisor station, as [jobs](#).

For more details, see [“Niagara Provisioning overview”](#) on page 3-1.

Q: What type of pre-defined tasks can provisioning perform?

A: Provisioning includes the ability to do station backups, install software, update license s, and copy files, among other things. Also, you can write custom Program code to execute as “provisioning robots,” allowing station database changes. For more details, see [“Niagara provisioning job steps”](#) on page 3-25.

Q: What changes were made in provisioning for AX-3.7?

A: Provisioning now accommodates “secure platform access” (platform SSL) introduced starting in AX-3.7, providing subordinate JACE hosts are licensed and configured to support this new certificate-based SSL platform access. Primarily, this affects the configuration of each NiagaraStation’s “PlatformConnection” provisioning extension. This type of platform access is optional, where the previous (non-SSL) method is still supported. For related details, see [“Platform Connection”](#) on page 4-3.

Note: For complete details on SSL starting in AX-3.7, refer to the NiagaraAX SSL Connectivity Guide.

Q: What changes were made in provisioning starting in AX-3.5?

A: Now, when choosing stations in a provisioning job, the “Add Device” dialog shows a tree structure that reflects any station folders, as well as any station “display names”. Both features can be useful when navigating a very large system with many replicated stations. Also, the “Station Software View” of a NiagaraStation’s “Software” extension was updated to operate similar to the corresponding platform “Software Manager” view in AX-3.5. Finally, a timeout routine with associated “timeout properties” was added to the NiagaraNetwork’s ProvisioningExt. The timeout mechanism permits a step in a provisioning job to fail, yet continue the job with the next station (instead of suspending the job indefinitely). These changes are covered in detail in the appropriate sections of this document.

Q: If I upgrade my Supervisor to AX-3.7, and it is already configured for provisioning (in a previous release), are there any known issues?

A: If you upgrade a Supervisor that was previously running AX-3.3 or later, the provisioning architecture remains the same, and operates as before. A few changes and additions were made starting in AX-3.7 as well as AX-3.5, as outlined in previous FAQ answers.

If you upgrade a Supervisor that was previously running AX-3.2 or AX-3.1 (using the “old style” provisioning architecture), you should convert to the newer architecture (NiagaraNetwork Provisioning-NwExt-based). A right-click command on the Supervisor’s “ProvisioningService” automates this. For details, see the appendix [“Converting from old style \(AX-3.1\) Provisioning”](#) on page A-1.

Q: Is there any other sort of provisioning besides “provisioning Niagara”?

A: At the time of this document, provisioning in AX-3.7 applies only to the Niagara driver, meaning for Niagara hosts represented in the NiagaraNetwork of a Supervisor station. However, with the separation of provisioning functions into different modules, reflected by a “BatchJobService” as well as a “network-level” “ProvisioningNwExt,” other driver types may support provisioning in the future.

Provisioning terms

The following list of terms and abbreviations are used when describing Niagara provisioning, and are used in this document. For general NiagaraAX terms, see the Glossary in the *User Guide*. Note that this glossary may grow over time, or may else simply be eliminated.

batch job A batch job is a station job that is managed and launched (sent to the station’s JobService) by the station’s BatchJobService. The BatchJobService and batch jobs replace job control functions that were formerly done (exclusively) by the ProvisioningService in early NiagaraAX releases (AX-3.1 and AX-3.2).

At the time of this document, only NiagaraNetwork provisioning uses batch jobs. However, at some future time other network drivers may also offer provisioning using batch jobs.

job A job is used to manage a task performed by the station. It runs asynchronously in the background, but provides user visibility. Provisioning is done entirely with “batch jobs”, which requires the Supervisor to have the BatchJobService (in addition to the “core” JobService). The BatchJobService provides additional job control functions to launch and manage jobs.

Apart from provisioning, other station services and drivers also perform jobs, all of which can be viewed in the station’s JobService. However, unlike other jobs, all provisioning (batch) jobs are persistent across a station restart. Also, provisioning provides specialized views for building and administering provisioning batch jobs.

platform snapshot A “snapshot” of installed software on a remote host running a station, which gets built/updated when you access the “Software” extension under a NiagaraStation component. This snapshot is used by Niagara provisioning when performing queries and installation of software.

software registry Used to describe the catalog of available software installable files, such as modules (.jars) or Niagara distribution files (.dists) under the !sw directory of a NiagaraAX workstation or Supervisor.

Document Change Log

Updates (changes/additions) to this *Niagara^{AX} Provisioning for Niagara Networks Guide* document are listed below.

- Updated: August 4, 2012
Document was updated for AX-3.7, with changes and additions that mostly relate to emphasizing usage of NiagaraNetworkJobPrototypes for regular station backups (for any release on NiagaraAX), plus the support of “secure platform access” (platform SSL) introduced in AX-3.7. Such changes are included in this Preface’s “[Provisioning FAQs](#)” section, changes in the “[Provisioning Niagara Quick Start](#)” on page 2-1, and the section “[Platform Connection](#)” on page 4-3 that describes that provisioning extension for any NiagaraStation found in a AX-3.7 or later Supervisor. Additionally, most screen captures were updated to show the default AX-3.7 “Lucid” Workbench theme. Numerous other minor changes were also made.
- Updated: February 10, 2010
Document is a “version-split” from the previous *Provisioning for Niagara Networks Guide* revision, and applies to NiagaraAX-3.5 (AX-3.5) and later Niagara provisioning functions. Many NiagaraNetwork and platform-related changes were made in AX-3.5 that affected Niagara provisioning. With this version split, many notes and references about older revisions of NiagaraAX (AX-3.0 - AX-3.4) were removed. However, a few references remain, with the possibility of a Supervisor upgrade from and earlier release to AX-3.5.
AX-3.5-related changes to this document are many, but are concentrated in the following sections:
 - The “[Provisioning FAQs](#)” section in this Preface was reworked to include changes starting in AX-3.5.
 - Details on converting a Supervisor using the older “AX-3.1/AX-3.2 style” provisioning in were removed from the “[Provisioning Niagara Quick Start](#)” section, and put in a new appendix, “[Converting from old style \(AX-3.1\) Provisioning](#)”. Moving forward, this information should be rarely needed.
 - In the main section “[Niagara ProvisioningNwExt Concepts](#)” on page 3-1, several screen captures were updated and other various minor edits made. Changes are generally in subsections “[Niagara Provisioning overview](#)” on page 3-1, “[Niagara Network Job Builder](#)” on page 3-2, “[Stations List](#)” on page 3-4 (new dialog when [Adding Stations](#)), “[Niagara Network Job View](#)” on page 3-7, “[ProvisioningNwExt properties](#)” on page 3-12 (new timeout properties), job prototype “[Job retention](#)” on page 3-19, and job prototype “[Adding Stations](#)” on page 3-22.
 - In the other main section “[Provisioning Station Extension Concepts](#)” on page 4-1, several screen captures were updated and other minor edits made. Changes are generally in subsections “[Provisioning Manager](#)” on page 4-2, and Software extension’s “[Station Software View](#)” on page 4-10.
 - Summary description in sections “[Provisioning Plugin Guides](#)” and “[Provisioning Niagara Component Guides](#)” were reviewed and updated, reflecting a few name changes given to plugins (views) and components since the initial provisioningNiagara and batchJob modules.
 - As previously mentioned, a new appendix “[Converting from old style \(AX-3.1\) Provisioning](#)” was added.

CHAPTER 1

Provisioning Niagara Installation

To use the latest provisioning features described in this document, you must have a NiagaraAX-3.5 or later Supervisor. By default, your Supervisor PC host is licensed with the feature “provisioning”. See the next section [“Provisioning license considerations”](#) for further details.

Note: *In general, it is recommended that your Supervisor and subordinate JACEs are running the same NiagaraAX release, for example AX-3.7. However, JACEs running an earlier release than the Supervisor are typically supported for most provisioning operations.*

JACEs do not require any provisioning-related module (provisioningNiagara, batchJob, provisioning), nor do they require a license feature for provisioning. However, JACEs do require the program module (and the ProgramService in their station) in order to process provisioning jobs with “Run Robot” steps.

The following sections provide details about Provisioning licensing and software installation:

- [Provisioning license considerations](#)
- [Provisioning software installation](#)

Provisioning license considerations

Only a Supervisor can be (or needs to be) licensed for provisioningNiagara, with an entry similar to below in its Tridium.license (license) file:

```
<feature name="provisioning"
  expiration="never"
  parts="ENG-WORKSTATION" />
```

If this feature is missing or becomes expired, the Supervisor station’s ProvisioningNwExt (under its NiagaraNetwork) will have a fault status, as will all provisioning-related extensions for NiagaraStations under its NiagaraNetwork.

Provisioning software installation

Provisioning software requires multiple modules installed in the modules folder of the Supervisor PC. These modules are

- provisioningNiagara
- batchJob

Note: *If the Supervisor was upgraded from AX-3.1 or AX-3.2, it also requires the provisioning module, in order to perform the conversion to the new provisioningNiagara architecture used in AX-3.3 and later.*

Basic configuration procedures and a few global provisioning tasks are covered in the next section, [“Provisioning Niagara Quick Start”](#). Conceptual and reference details about the ProvisioningNwExt and and provisioning station (device) extensions are in the main sections [“Niagara ProvisioningNwExt Concepts”](#) on page 3-1, including the beginning section [“Niagara Provisioning overview”](#) on page 3-1, and also [“Provisioning Station Extension Concepts”](#) on page 4-1.

CHAPTER 2

Provisioning Niagara Quick Start

This section provides a collection of procedures to configure a Supervisor station for provisioning Niagara. Also, procedures are included to begin performing basic provisioning—including a few possible first usages. These are the main subsections:

- [“Add and configure necessary components”](#) on page 2-1
 - [“Add the Batch Job Service and ProvisioningNwExt”](#) on page 2-1
 - [“Configure the NiagaraDevice extensions”](#) on page 2-2
- [“Run provisioning jobs”](#) on page 2-3
 - [“Run custom Niagara provisioning jobs”](#) on page 2-3
 - [“Add custom persisted Niagara provisioning jobs”](#) on page 2-3

Note: *If an existing Supervisor upgraded from AX-3.1 or AX-3.2, and it was configured for provisioning using the “old style” provisioning architecture (“ProvisioningService” in its “Services” container), you can run a conversion utility. See the appendix [“Converting from old style \(AX-3.1\) Provisioning”](#) on page A-1.*

Add and configure necessary components

To configure Niagara provisioning for a Supervisor not previously configured for provisioning, perform the following main tasks:

- [Add the Batch Job Service and ProvisioningNwExt](#)
- [Configure the NiagaraDevice extensions](#)

Add the Batch Job Service and ProvisioningNwExt

To add the Batch Job Service and ProvisioningNwExt

- Step 1 In Workbench, open the Supervisor station.
- Step 2 Open the **provisioningNiagara** palette in the Workbench palette side bar (see “Using the palette side bar” in the *User Guide* for general details).
- Step 3 Expand the station’s Config space to see its **Services** folder, and double-click it for the **Service Manager** view. If a BatchJobService is already listed, verify it is enabled (status “ok”), and skip ahead to [step 6](#).
- Step 4 From the palette, *drag* the **BatchJobService** onto the station’s **Services** folder. In the popup **Name** dialog, you can rename the service—or, simply use the default name.
- Step 5 Click **OK** to add the BatchJobService to the station.
A BatchJobService named “BatchJobService” (or whatever you named it), is under your Services folder.
Note: *From the property sheet of the BatchJobService, you can set the service’s **Job Queue > Max Threads** property to a value greater than 1, say 2 or 3, to allow more than a single provisioning job to run concurrently. It is also recommended that you change its Alarm Class from the default “Default Alarm Class” to another alarm class to use for monitoring provisioning alarms.*
For more details, see [“BatchJobService”](#) on page 3-33.
- Step 6 Under the station’s Config space, expand its **NiagaraNetwork**. If a ProvisioningNwExt is already a child of the network, verify that it is enabled (status “ok”), and skip ahead to the next procedure, [“Configure the NiagaraDevice extensions”](#).
- Step 7 From the palette, *drag* the **ProvisioningNwExt** onto the station’s **NiagaraNetwork**. In the popup **Name** dialog, you can rename the ProvisioningNwExt—or, simply use the default name.
- Step 8 Click **OK** to add the ProvisioningNwExt to the Supervisor’s NiagaraNetwork.

You should have a ProvisioningNwExt named “ProvisioningNwExt” (or whatever you named it), under your NiagaraNetwork.

Configure the NiagaraDevice extensions

When you add the ProvisioningNwExt to the NiagaraNetwork in a Supervisor station, every existing NiagaraStation device (under the network) automatically has five new device extensions created. See “Types of provisioning extensions” on page 4-1 for more details.

In addition, as you add more NiagaraStations, they too will have these extensions (see “New station notes” on page 4-4). In order for provisioning to work with the hosts running these stations, you need to enter some property values in one of these extensions.

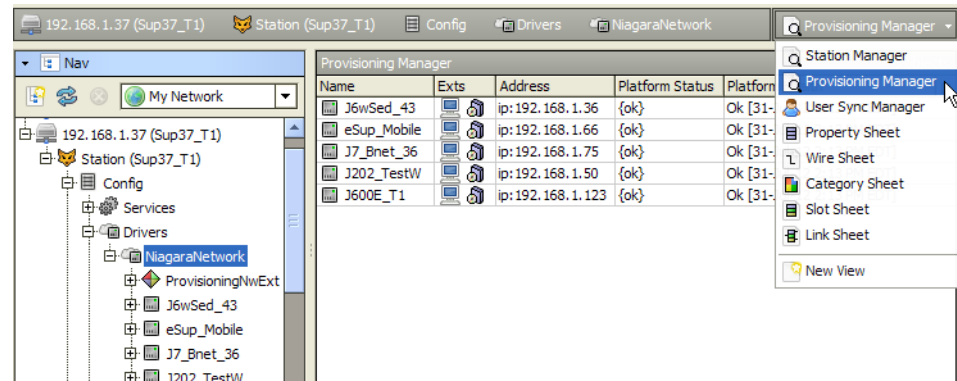
To configure the NiagaraStation Platform Connection extensions

For each NiagaraStation device in the Supervisor station’s NiagaraNetwork, you must enter the corresponding platform daemon credentials of its host. The simplest way to do this is by using the NiagaraNetwork’s **Station Manager** view, as follows:

- Step 1 In the Nav tree, expand **Config, Drivers** then double-click **NiagaraNetwork**. The **Station Manager** view appears, listing all stations.
 - Step 2 Double-click the name of a station to open its **Edit** dialog. The last few “Platform” fields are for editing that station’s Platform Connection extension properties, namely for platform User, Password, and Port.
- Note:** In AX-3.7 or later, an additional “Secure Platform” boolean property is present, set to “false” by default. If the station’s host platform is configured for SSL access (platformSSL) set this to true. In this case, note the default Platform Port is not 3011, but is 5011—unless changed to some other port in that platform.
- Step 3 Enter its platform connection credentials, and if using a non-default port address, change from the default port 3011 to the port number used.
- Note:** Platform credentials (username, password) and port should be the same credentials and port you use when you open a platform connection directly to this remote (JACE) host.
- Step 4 Click the **OK** button.
- Upon the next monitor ping of that host’s platform daemon, its “Station Run State” should change from Unknown to Running (the station’s “Platform Connection” extension should now have status “ok”). Alternatively, right-click a NiagaraStation’s Platform Connection extension and invoke the action “Ping” for an immediate ping of its platform daemon.
- Step 5 Repeat this procedure for each NiagaraStation listed in the **Station Manager** view.
- Note:** If using the identical platform credentials and port numbers in multiple (or even all) hosts running each station, you can select multiple stations (rows) in the **Station Manager** to do this edit (right-click and select **Edit** with multiple stations highlighted).

Alternatively, you can use the **Provisioning Manager** view on the Supervisor’s NiagaraNetwork to edit these properties for each NiagaraStation (Figure 2-1 below).

Figure 2-1 Provisioning Manager view of the Supervisor’s NiagaraNetwork



When done configuring platform connections, all stations shown in the **Provisioning Manager** should have a Platform Status of “ok”, as shown in Figure 2-1 above. For related details on this view, see “Provisioning Manager” on page 4-2.

Run provisioning jobs

After you [Add the Batch Job Service and ProvisioningNwExt](#) and [Configure the NiagaraDevice extensions](#), you can run provisioning jobs.

- [Run custom Niagara provisioning jobs](#)
- [Add custom persisted Niagara provisioning jobs](#) (the recommended method for backups)

Note: Previous revisions of this document included a third “Backup all stations” procedure, which was simply to invoke the right-click “Start Backup” action on the **ProvisioningNwExt**. Although “conveniently easy”, this method is generally not recommended, except for a small, lightly-loaded system with only a few JACEs. On a larger enterprise system, this could take an excessive amount of time and put an undue load on the system at a peak time. For related details, see “[ProvisioningNwExt action](#)” on page 3-13, including new subsection “[Why the Start Backup action is typically NOT recommended](#)” on page 3-13.

Run custom Niagara provisioning jobs

Note: Before running provisioning jobs, ensure that all **NiagaraStations** are properly configured for platform connections. See “[To configure the NiagaraStation Platform Connection extensions](#)” on page 2-2.


By double-clicking the **ProvisioningNwExt** for its [Niagara Network Job Builder](#) (default view) you can run various sequences of provisioning steps against one or more stations in the **NiagaraNetwork** of the Supervisor. Typically, you use this method for provisioning jobs that you run infrequently, including “special jobs” such as updating license files in selected (or all) remote JACEs in the network, or updating one or more software modules in them.

Note: For regular scheduled backups of networked JACEs, especially for large enterprise Supervisors, it is recommended that you copy “job prototype” components in the station and configure them accordingly. This provides “job retention” advantages not available if working from the **Niagara Network Job Builder**. See the next section “[Add custom persisted Niagara provisioning jobs](#)” for more details and an example.

For example, you may wish to run the following job for all stations, after you have a new **NiagaraAX** build installed on your Supervisor:

Synchronizing Supervisor license database and updating JACE licenses

This procedure assumes you have purchased JACE license upgrades, and they are available on the online licensing server.

- Step 1 Expand the **ProvisioningNwExt** in the Nav tree to see its  **Licenses** node.
- Step 2 Right-click **Licenses** and select **Views > Supervisor License Manager**.
- Step 3 In the Supervisor License Manager, click **Synchronize**.
Click **Yes** to “Sync All Licenses”, and click **OK** at Synchronization Complete.
- Step 4 Double-click the **ProvisioningNwExt** for its **Niagara Network Job Builder** view.
- Step 5 In the top “Initial steps to run only once” pane, click the Add button,
In the **New Job Step** popup dialog, click the **Update Licenses** step and click **OK**.
- Step 6 In the lower “Stations to include in the job” pane, click the Add button,
In the **Add Device** popup dialog, click to select JACE stations and click **OK**.
- Step 7 Review and click the **Run Now** button at the bottom of the **Niagara Network Job Builder** view to initiate this provisioning job.
The view changes to the **Niagara Network Job View**, where steps and results appear as they are executed. For more details see the following related sections:
 - “[Licenses slot](#)” on page 3-16
 - “[Niagara Network Job Builder](#)” on page 3-2
 - “[Niagara Network Job View](#)” on page 3-7

Add custom persisted Niagara provisioning jobs

Note: Before running provisioning jobs, ensure that all **NiagaraStations** are properly configured for platform connections. See “[To configure the NiagaraStation Platform Connection extensions](#)” on page 2-2.

Using “job prototype” components copied from the provisioningNiagara palette into the Supervisor’s station, you can specify various sequences of provisioning steps against one or more stations in the **NiagaraNetwork** of the Supervisor, and save as normal “persisted” components (duplicate, re-edit, etc.). Each job prototype provides a default [Niagara Network Prototype View](#) for building that job.

As needed, link any job prototype component to a **TriggerSchedule** in the station for future execution.

Note: This is the recommended method for regular, scheduled backups on networked JACEs.

Example Supervisor configured for “Job Prototype” backups of stations

This example procedure is for a Supervisor station with 36 JACE stations in its NiagaraNetwork.

- Step 1 With the Supervisor station opened in Workbench, expand it to reveal its **Config** node.
- Step 2 Right-click Config and select **New > Folder**.
Name it “ProvBackups” (or whatever desired. You can also create this folder anywhere under Config.)
- Step 3 Open the Palette side bar, and then in the side bar open the provisioningNiagara palette.
- Step 4 From the provisioningNiagara palette, drag (copy) a **NiagaraNetworkJobPrototype** component to drop (paste) in the new ProvBackups folder.
- Step 5 From the provisioningNiagara palette, drag (copy) a **TriggerSchedule** component to drop (paste) in the same ProvBackups folder.
- Step 6 Double-click the ProvBackups folder for its wire sheet view.
In the wire sheet, link the Trigger slot of the TriggerSchedule to the SubmitJob slot of the JobPrototype component.
- Step 7 Double-click the NiagaraNetworkJobPrototype for its **Niagara Network Prototype View**.
1. In the middle “Steps to run for each station” pane, click the Add button.
Select **Backup Stations**, then **OK**.
 2. In the bottom “Stations to include in the job” pane, click the Add button.
In the **Add Devices** popup dialog, select *some* of the stations (in this case 12 stations), then **OK**.
- Note:** Later, after duplicating this component and reconfiguring, you will select different stations to backup.
3. Click the **Save** button at the bottom of the view.
- Step 8 For the same NiagaraNetworkJobPrototype, go to its (secondary) **Prototype Job List** view.
1. Near the top of its **Prototype Job List** view, click the “**Retention policy**” control for a drop-down list, and choose either:
 - **Dispose after a specified amount of time** (default is 7 days)
 - **Keep a limited number of executions** (default is 10)
 2. Click the **Save** button at the bottom of the view.
- Step 9 Double-click the linked TriggerSchedule to go to its Trigger Schedule view, and add a new event on the left side, for example: Type: **Week and Day**: Sat, Any Week, Any Month (for a weekly backup) and on the right side add an “off hours” time, for example: 12:30AM.
Click to **Save** this schedule configuration.

Note: Later, after duplicating this component and reconfiguring, you will select a slightly different off hours time.

- Step 10 With the **NiagaraNetworkJobPrototype** linked to the **TriggerSchedule**, select both components, then right-click and select **Duplicate**. This puts another linked pair on the wire sheet.
Move the copied linked pair to a new spot on the wire sheet, and repeat this duplicate again.

Note: In this example you now need to configure each of the 3 job prototypes to backup 12 different stations per provisioning job, and adjust each TriggerSchedule to a slightly different time of execution.

For example, you may have one backup provisioning job scheduled at 12:30AM, another at 1:00AM, and another at 1:30AM.

For related details, see the “[About the Niagara Network Job Prototype](#)” on page 3-17, “[Scheduling a job prototype](#)” on page 3-18, and “[Job retention](#)” on page 3-19.

CHAPTER 3

Niagara ProvisioningNwExt Concepts

This section describes Niagara Network provisioning concepts, including the **ProvisioningNwExt** component for the Supervisor's NiagaraNetwork, which provides the core of Niagara provisioning. Also described are the available "job prototype" components for Niagara provisioning, the different provisioning job steps, and the various views encountered while working with all of these items.

These are the main subsections:

- "Niagara Provisioning overview" on page 3-1
- "About the ProvisioningNwExt" on page 3-2
- "Niagara Network Job Builder" on page 3-2
- "Niagara Network Job List" on page 3-5
- "Niagara Network Job View" on page 3-7 (and similar "Batch Job Log File View" on page 3-9)
- "Batch Job Step Log File View" on page 3-9 (and similar "Job Step Details View" on page 3-11)
- "Supervisor Software container" on page 3-14
- ProvisioningNwExt "Licenses slot" on page 3-16
- "About the Niagara Network Job Prototype" on page 3-17
- "Niagara Network Prototype View" on page 3-21
- "Prototype Job List" on page 3-24
- "Niagara provisioning job steps" on page 3-25
- "Provisioning Robot notes" on page 3-30
- "Niagara Provisioning Job management" on page 3-33

Niagara Provisioning overview

Niagara provisioning is available only to a Supervisor station. It provides automation of various tasks to remote (JACE) hosts in the station's NiagaraNetwork. For the most part, these are *platform* tasks—that is, otherwise done using (full) Workbench, making individual platform connections *directly* to remote JACE hosts, then using the appropriate platform views.

Additionally, "provisioning robot" tasks are available. This allows running custom program code in the *station* running on each JACE host (executed by those stations' ProgramService). It is important to note that the Supervisor *station* does all these tasks, modeled in the station as provisioning **jobs**.

Outside of provisioning, you must perform similar tasks using (full) Workbench, either by:

- Making individual platform connections *directly* to remote JACE hosts, then using the appropriate platform views.
- Making individual *tunneled* platform connections to remote JACE hosts, then using the appropriate platform views (platform tunneling available starting in AX-3.5).
- Or in the case of "provisioning robots", by opening station connections and then copying and executing Program objects.

Note: For details about the platform user interface, see "Platform overview" in the *NiagaraAX Platform Guide*. Provisioning provides some advantages over "individual" platform (or station) connections, for example:

- When provisioning, you need only one *station connection*—to the Supervisor, and no other connections (platform or otherwise).
This means you can do provisioning from anywhere you can open the Supervisor station...even using Web Workbench! (ordinary platform tasks cannot be done using Web Workbench.)
- Provisioning allows the same series of tasks (executed as the job's *steps*), to be executed to any number of target JACE hosts. Most job steps execute sequentially on one host, then are repeated on the next host, until all specified hosts are done—or, just to a single JACE, if specified. This ability is useful

when performing the same tasks with multiple JACEs, such as a job-wide software upgrade, or a periodic backup of all hosts' station configurations.

- You can build a provisioning job to run immediately, using the job builder view of the NiagaraNetwork's ProvisioningNwExt, as well as use the BackupSchedule slot of the ProvisioningNwExt to specify a regular, repeating schedule backup for all stations. Or, you can use “job prototype” components copied from the provisioningNiagara palette and pasted in the station. Any job prototype can be linked to a TriggerSchedule, and can also be run immediately.
- By default, provisioning provides persistent storage of all jobs on the Supervisor, including all statistics associated each job and step (creating user, begin and end job times, step details, log output, and so on). In the case of station backups, any saved .dist file can also be restored directly from its batch job step log— via a “Restore” function, whereby it is executed as another provisioning job.
- Provisioning tasks in the Supervisor station can be done in *three* different ways:
 - From the ProvisioningNwExt, under the Supervisor's NiagaraNetwork component, described next—see [“About the ProvisioningNwExt”](#).
 - From individual job components. See [“About the Niagara Network Job Prototype”](#) on page 3-17.
 - From NiagaraStation *provisioning* (device) *extensions*, under each station modeled in Drivers, NiagaraNetwork. See [“Provisioning Station Extension Concepts”](#) on page 4-1.

About the ProvisioningNwExt

The **ProvisioningNwExt** of a NiagaraNetwork in a Supervisor station is the core of Niagara provisioning. The following main sections provide related details about its views and slots:

- Views: [“Niagara Network Job Builder”](#) on page 3-2 and [“Niagara Network Job List”](#) on page 3-5.
- Slots: See [“ProvisioningNwExt properties”](#) on page 3-12, [“ProvisioningNwExt container components”](#) on page 3-12, and [“ProvisioningNwExt action”](#) on page 3-13.

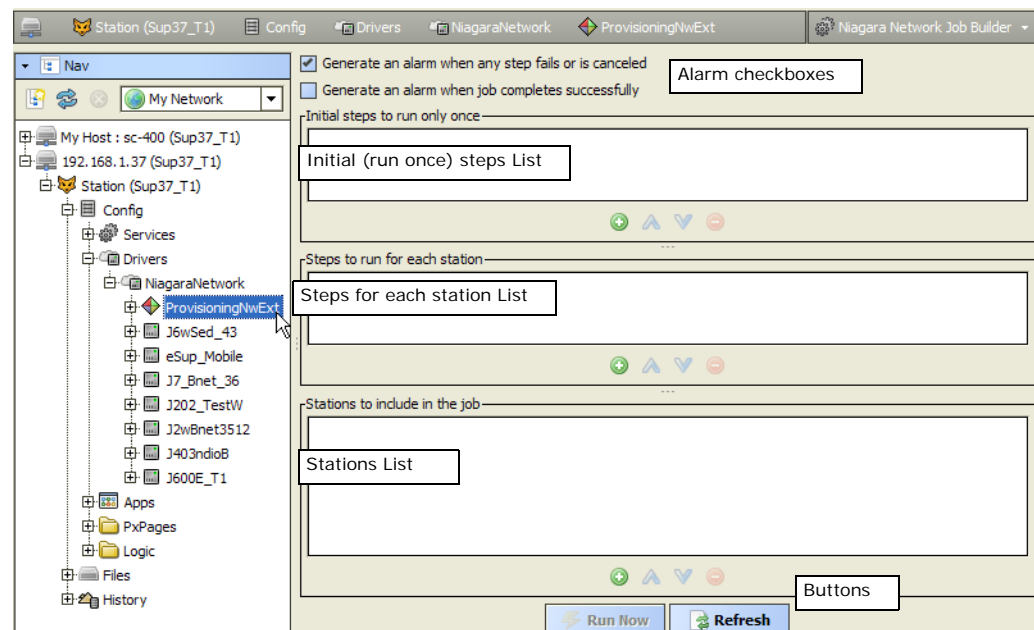
Niagara Network Job Builder

The default view on the **ProvisioningNwExt**, you can start here to specify and launch a “one time” provisioning job needed now, meaning that you do not need to save as a reusable component.

Note: *If you would rather build a job that you can schedule or run at a later time, save, duplicate and modify, and so on, use a NiagaraNetworkJobPrototype component copied anywhere in the Supervisor's station (each has its own equivalent view). Find them in the provisioningNiagara palette.*

In general, those are the components you should use to create regular station backup jobs. For related details, see [“About the Niagara Network Job Prototype”](#) on page 3-17.

Figure 3-1 Niagara Network Job Builder is default view of ProvisioningNwExt



As shown in [Figure 3-1](#), this view has *three* list areas:

- [Initial \(run once\) steps List](#)

- [Steps for each station List](#)
- [Stations List](#)

There are also [alarm checkboxes](#) and [buttons](#).

Initial (run once) steps List

This *top* area of the [Niagara Network Job Builder](#) shows a one-line summary for each step to be performed *once* in the job, regardless of how many stations will be specified. Currently, when provisioning Niagara this means only *one type* of step: “Update Licenses”, and is optional. (Often you may wish to skip this step, but add other types of steps in the middle “[Steps for each station List](#)” pane).

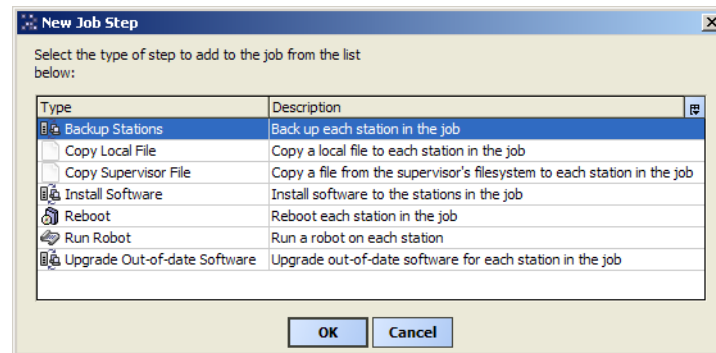
To add this step, click the “+” (add) button below this list—see “[Adding Job List Steps](#)” on page 3-3 for further details. For details on this initial step, see “[Update Licenses step](#)” on page 3-26.

Steps for each station List

This *middle* area of the [Niagara Network Job Builder](#) shows a one-line summary for each step to be run for each station specified in this job. In most provisioning jobs, you [add](#) one or more steps. As needed, you can also [remove](#) and [reorder](#) steps (when the job runs, steps execute in the top-to-bottom order).

Adding Job List Steps

Figure 3-2 New Job Step popup menu (from Niagara Network Job Builder)



Add a provisioning job step by using any of the following methods:

- Click the “+” (add) button below the list, and choose the step type from the **New Job Step** popup menu (see [Figure 3-2](#)). Note that this dialog provides two different types of file copy steps:
 - Copy Local File — To copy a file that exists locally on your local Workbench PC (not on the Supervisor), to each target host running an included station. This is the only step type not available in a [job prototype](#).
 - Copy Supervisor File — To copy a file that exists on the Supervisor (not on your local PC), to each target host running an included station. Note if you are running Workbench at the Supervisor PC, both are effectively the same step options.

For details about step types, see “[Niagara provisioning job steps](#)” on page 3-25.

- Right-click in the steps list, select **Add**, and choose the step type from the popup menu.
- Drag a file from Workbench’s Nav tree into the steps list (implicit File Copy step).
- Drag a software item (module or dist) from Workbench’s Nav tree that appears under the ProvisioningNwExt’s “**Software**” container, into the Job Steps List (implicit [Install Software step](#)).
- Drag a ProvisioningRobot that exists in the station’s Config (component) architecture into the Job Steps List (implicit [Run Robot step](#)).

Removing Job List Steps

Remove a provisioning job step using either of these two methods:

- Click to select the step, then click the “X” (remove) button below the list.
- Right-click the step, and select **Remove** from the popup menu.

Reorder Job List Steps

Reorder a selected job step by clicking the (up) or (down) *arrow* button at the bottom of the list. Or, right-click a job step and select **Move Up** or **Move Down**, as needed.

Note: *Job steps are executed in the same top-to-bottom order as defined in the two step lists: “initial steps” first, then “steps for each station”.*

Stations List

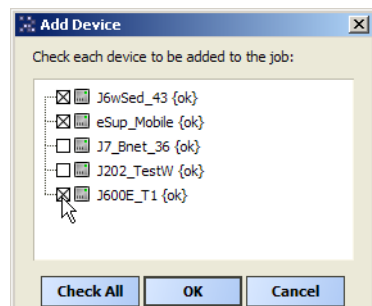
This lower area of the [Niagara Network Job Builder](#) lists all the stations to be processed by the job—note this means each station processes all steps in the (middle pane) “[Steps for each station List](#)”. Only stations in the Supervisor’s NiagaraNetwork can be added. For any job, you [add](#) one or more stations, and you can also [remove](#) and [reorder](#) stations (stations are processed in a top-to-bottom order).

Adding Stations

Add a station to a job in the [Niagara Network Job Builder](#) by doing any of the following:

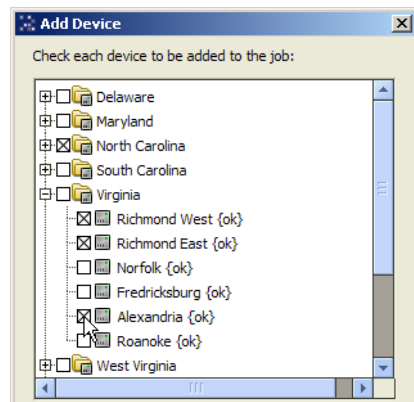
- Click the “+” (add) button below the list, and choose the station(s) in the **Add Device** popup dialog (see [Figure 3-3](#) and [Figure 3-4](#)). A **Check All** button allows you to choose all stations.
- Right-click in the Stations List, select **Add**, and choose the station(s) in the same popup dialog.
- From the Nav tree, simply drag a station from under the NiagaraNetwork into the Stations List.

Figure 3-3 Add Device dialog (small system example)



Note the dialog for adding stations also reflects any station folders (NiagaraStationFolder) in an expandable tree structure. Also, stations are listed using any assigned “display names” ([Figure 3-4](#)).

Figure 3-4 Add Device dialog (large system using station folders, display names)



This station tree structure using station display names (if assigned) can be useful on Supervisors that have very large numbers of subordinate JACE stations. Click on controls to expand and collapse station folders, as needed. A selected folder will select *all nodes* under it (stations and any subfolders).

Removing Stations

Remove a station from a job in the [Niagara Network Job Builder](#) using either of these two methods:

- Click to select the station, then click the “X” (remove) button below the list.
- Right-click the station, then select **Remove** from the popup menu.

Reorder Stations

Reorder a selected station in the [Niagara Network Job Builder](#) by clicking the (up) or (down) *arrow* button at the bottom of the list. Or, right-click a station and select **Move Up** or **Move Down**, as needed.

Note: Stations are processed in the same top-to-bottom order as defined in the Stations List.

Alarm checkboxes

The two checkboxes at the top of the [Niagara Network Job Builder](#) determine if alarms (actually *alerts*) are to be issued by the **BatchJobService** for this provisioning job, and under what circumstances. Alerts use the alarm class specified in the property sheet of the BatchJobService.

- **Generate an alarm when any step fails or is canceled** — if checked, an alert is raised whenever a job step fails or is canceled.
- **Generate an alarm when a job completes successfully** — if checked, an alert is raised whenever a job completes with no step failures.

Note: *Checkbox settings apply to the provisioning job being built, and do not affect other provisioning jobs that may exist (already queued to run, or as [job prototypes](#), that is [NiagaraNetworkJobPrototypes](#)).*

For further alarm details, see [“About provisioning-related alarms”](#) on page 3-35.

Buttons

The **Run Now** button at the bottom of the [Niagara Network Job Builder](#) becomes enabled when there is at least *one* job step in either the [“Initial \(run once\) steps List”](#) or [“Steps for each station List”](#), and *one* station in the [“Stations List”](#). When you are finished specifying a provisioning job, you can click it.

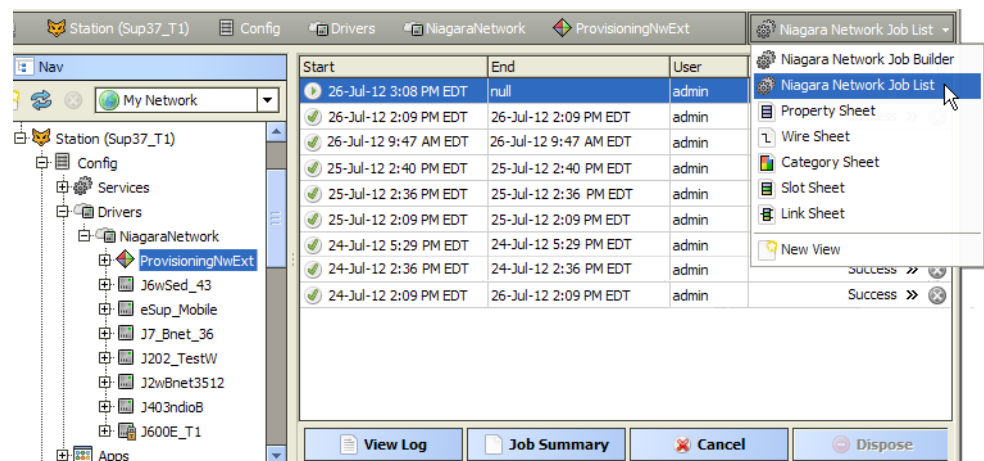
- **Run Now** — click to dispatch the job to the batch job queue for immediate execution. The Workbench view automatically changes to the [Niagara Network Job View](#).

Note: *The **Refresh** button is always enabled, note that it removes all entries from all three lists.*

Niagara Network Job List

Another view on the NiagaraNetwork’s **ProvisioningNwExt**, use the view selector ([Figure 3-5](#)) to access it, or right-click the ProvisioningNwExt and select **Views > Niagara Network Job List**.

Figure 3-5 *Niagara Network Job List is another view of the ProvisioningNwExt*



As shown in [Figure 3-5](#), this view has one list area: **Jobs Table**, with **buttons** at the bottom.

Jobs Table

This main area of the [Niagara Network Job List](#) shows a supervisor-wide list of provisioning jobs that have been sent to run, are running, or are have completed. Note that any “pending jobs” do not appear until the linked trigger schedule actually fires.

This jobs table differs from the one in the **Job Service Manager** view on the station’s JobService in the following ways:

- This table shows *all* NiagaraNetwork provisioning jobs, whereas the Job Service Manager table shows only 10 jobs maximum (of various types, and not just provisioning jobs).
- Jobs persist (remain) following a station restart, whereas all jobs are cleared under the JobService.
- More information shows in this table, with **columns** for start time, end time, and so on.
- You can double-click any job row to view its **Job Log**, a series of log messages about the job stored in its job log file (same as using the **View Log** button at the bottom of the view).
- You can right-click a job for a popup menu—providing same functions as **buttons** (at view bottom).

Jobs table columns

Jobs table columns in the [Niagara Network Job List](#) include the following:

- **Start** — time the job was queued.
- **End** — time the job terminated (successfully, failed, or canceled).
- **User** — station user that requested the job (shows “unknown” if job triggered by a linked schedule).
- **Status** — the status of the jobs, as one of the following:
 - Unknown — job is pending execution, because all threads in the job queue are in use.
 - Running — job is executing.
 - Canceling — request to cancel the job was sent, but has not been processed yet, and the job is still executing.
 - Success — job finished successfully, with all steps completed for all stations.
 - Canceled — job was canceled before it completed, and is no longer running.
 - Failed — at least one step failed in one station; job is no longer running.

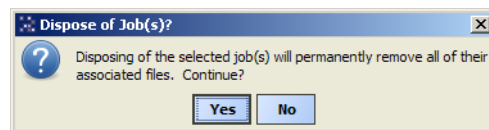
The status column ends with a “>>” (Details) and “X” (Dispose) button for each job, these function the same as the **Job Summary** and **Dispose** buttons at the bottom of this view.

Buttons

Most buttons near the bottom of the [Niagara Network Job List](#) become enabled when you have a job row selected in the [Jobs Table](#). Buttons are described as follows:

- **View Log** — For a popup **Job Log** dialog, showing log messages output by the selected job. See “[Job Log notes](#)” for more details.
- **Job Summary** — To change to the [Niagara Network Job View](#) (or [Batch Job Log File View](#)), to display the selected job in detail.
- **Cancel** — Enabled only if selected job has Running status. If clicked, the job is notified it should cancel when safe to do so. Note that not all job steps can be canceled.
- **Dispose** — For a confirmation dialog to delete the selected job(s), including all associated job files ([Figure 3-6](#)).

Figure 3-6 Dispose Job Confirmation



If you confirm to dispose:

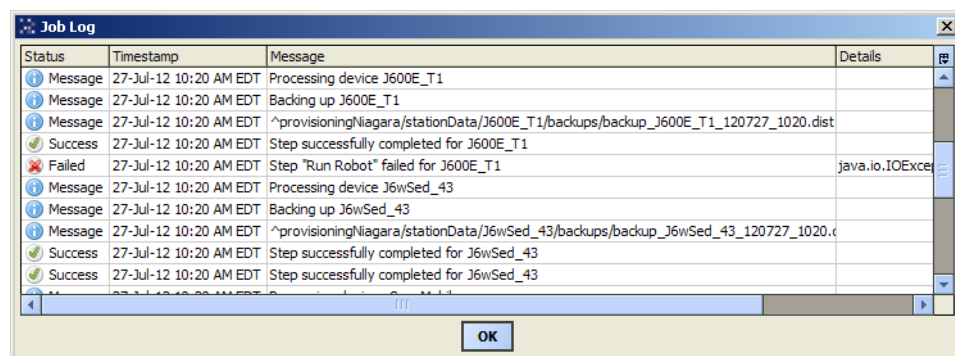
- Job(s) is deleted from the JobService (if still there, not rolled off as 11th job, or station restart).
- All associated job files are *deleted*, including the batch job log file, batch job step log files, and other files if applicable. Note for a backup job, this includes *deleting the backup .dist file(s)*.
- Job(s) is removed from the [Jobs Table](#) in the extension’s [Niagara Network Job List](#).

Note: You can use the **Dispose** button with multiple selected jobs, if needed.

Job Log notes

The popup **Job Log** table shows you a running log of messages output with the execution of the job. As shown in [Figure 3-7](#), each row includes a column for status, timestamp, message, and details.

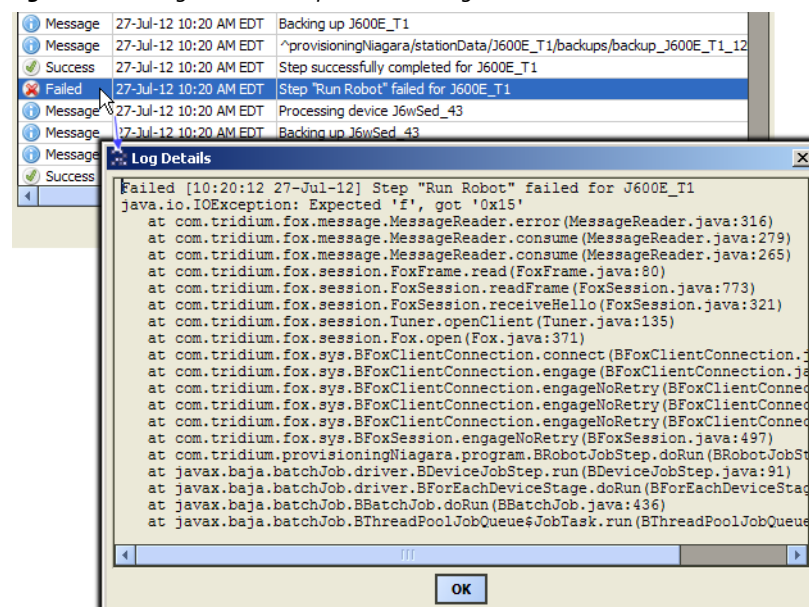
Figure 3-7 Job Log dialog example



Note: New log messages do not appear dynamically—to see newer messages you must reopen the **Job Log**.

To see more details (if available) on any row, double-click it for a **Job Details** popup dialog, as shown in [Figure 3-8](#).

Figure 3-8 Log Details example from Job Log

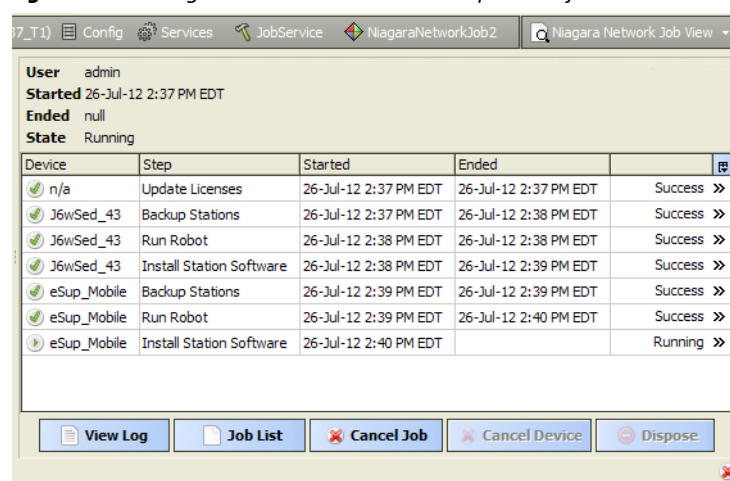


Niagara Network Job View

The Niagara Network Job View ([Figure 3-9](#)) shows details for a single Niagara provisioning job execution. It is the default view for any `NiagaraNetworkJob`. You see this view when you select “Run Now” in the [Niagara Network Job Builder](#) or [Niagara Network Prototype View](#), or by clicking the “>>” (Details) button in the [Niagara Network Job List](#) or the [Prototype Job List](#) on any *recent* provisioning job.

Note: If the job has completed (and no longer appears in its `JobServiceManager` view), instead of this view you see the [Batch Job Log File View](#) (see [Figure 3-11](#) on page 9) which operates in the same manner.

Figure 3-9 Niagara Network Job View lists steps in one job.



As shown in [Figure 3-9](#), from top-to-bottom this view has three areas:

- [Job elements \(read-only\)](#)
- [Step summary table](#)
- [Buttons](#)

Job elements (read-only)

The four read-only elements at the top of the [Niagara Network Job View](#) (or [Batch Job Log File View](#)) include the following:

- **User** — Station user that submitted the provisioning job (is “unknown” if a scheduled job).
- **Started** — Date/time when the job was submitted to the job queue.
- **Ended** — Date/time when the job stopped running, or blank if still running.
- **State** — Current or final state of the job, as one of the following:
Note: First three states apply only if the Device Network Job View.
 - Unknown — job is pending execution, because all threads in the job queue are in use.
 - Running — job is executing.
 - Canceling — request to cancel the job was sent, but has not been processed yet, and the job is still executing.
 - Success — job finished successfully, with all steps completed for all stations.
 - Canceled — job was canceled before it completed, and is no longer running.
 - Failed — at least one step failed in one station; job is no longer running.

Step summary table

This main area of the [Niagara Network Job View](#) (or [Batch Job Log File View](#)) shows a summary for each step in the job that has ever started processing for a station (in other words, no rows appear for steps that have not yet started, or were skipped because prior steps either failed or were canceled).

Columns in the step summary table include the following:

- **Station** — the station the step is (or has) processed.
- **Step** — type of provisioning job step (see “[Niagara provisioning job steps](#)” on page 3-25).
- **Started** — time the step started processing for this station.
- **Ended** — time the step finished, or blank if the step is still running.
- **State** — the state of the steps, as one of the following:
Note: First two states apply only if the Device Network Job View.
 - Running — step is executing.
 - Canceling — request to cancel the step was sent, but job has not processed this yet, and the step is still executing.
 - Success — step finished successfully, and is no longer running.
 - Canceled — job was canceled before it completed, and is no longer running.
 - Failed — the step failed and is no longer running.

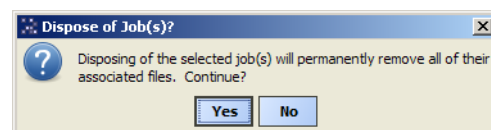
The State column ends with a “>>” (Step Details) button for each step. If clicked, this changes the view to the [Batch Job Step Log File View](#), showing all log messages on this single step.

Buttons

Buttons at the bottom of the [Niagara Network Job View](#) (or [Batch Job Log File View](#)) are described as follows:

- **View Log** — (always enabled) Click for a popup **Job Log** dialog showing log messages output by the job. See “[Job Log notes](#)” on page 3-6 for more details.
- **Job List** — (always enabled) Click to go “up a level” to the [Niagara Network Job List](#).
- **Cancel Job** — Enabled only if job is running. Clicking notifies the job to cancel when safe to do so.
- **Cancel Device** — Enabled only if job is running, and a step row is selected in the [Step summary table](#). Clicking notifies the job to cancel for the *selected station* (when safe to do so), and to begin job processing for the *next* station.
- **Dispose** — Enabled if job is finished. Click for a confirmation dialog to delete the selected job(s), including all associated job files ([Figure 3-6](#)).

Figure 3-10 Dispose Job Confirmation



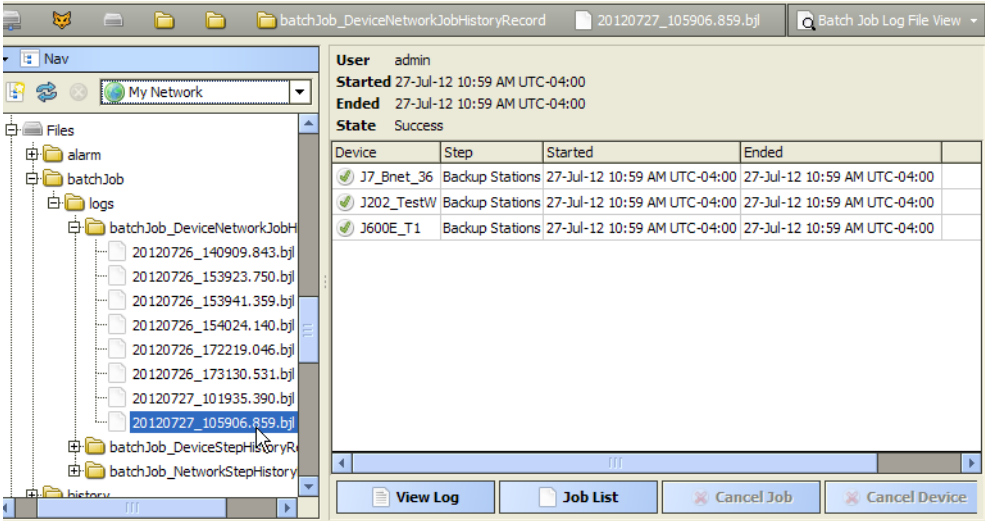
If you confirm to dispose:

- The job is deleted from the JobService (if still there, not rolled off as 11th job, or station restart).
- All associated job files are *deleted*, including the batch job log file, batch log step log files, and other files if applicable. Note for a backup job, this includes *deleting the backup .dist file(s)*.
- The job is removed from the [Jobs Table](#) in the ProvisioningNwExt’s [Niagara Network Job List](#).

Batch Job Log File View

The Batch Job Log File View (Figure 3-11) shows details for a single Niagara provisioning job that has completed. See this view by clicking the “>>” (Details) button for any batch job log file (.bjl extension).

Figure 3-11 Batch Job Log File View lists steps in one job



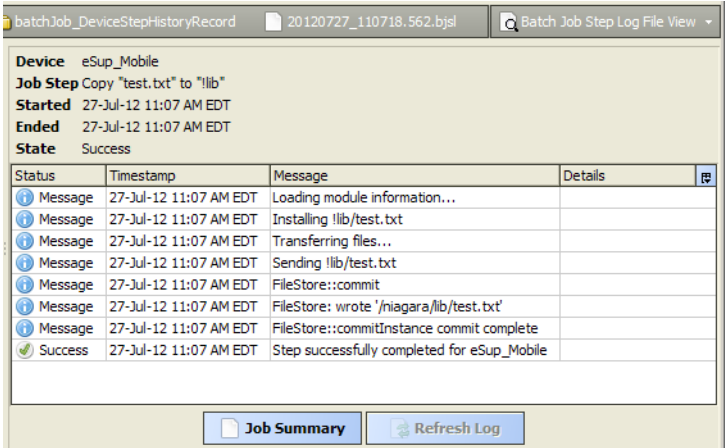
As shown in Figure 3-11, this view is nearly identical to the Niagara Network Job View (for a Niagara provisioning job still under the station's JobService). See the following Niagara Network Job View sections for related details:

- “Job elements (read-only)” on page 3-7
- “Step summary table” on page 3-8
- “Buttons” on page 3-8

Batch Job Step Log File View

The Batch Job Step Log File View (Figure 3-12) shows details for one provisioning job step executed against a single station. It is the default view for any batch job step log file (.bjsl extension). Typically, you see this view by clicking the “>>” (Details) button on a completed step in the Niagara Network Job View or Batch Job Log File View.

Figure 3-12 Batch Job Step Log File View lists details on one step for a station.



As shown in Figure 3-12, from top-to-bottom this view has three areas:

- Step elements (read-only)
- Step log table
- Buttons

Step elements (read-only)

Read-only elements at the top of the [Batch Job Step Log File View](#) include the following:

- **Device** — Station the step is (or has) processed.
- **Job Step** — Type of provisioning step (Backup Stations, File Copy, Install Software, etc.).
- **Started** — Date/time the step started processing.
- **Ended** — Date/time the step finished, or blank if it is still running.
- **State** — Current or final state of the step, as one of the following:
 - Running — step is executing.
 - Canceling — request to cancel the step was sent, but has not been processed yet, and the job is still executing.
 - Success — step finished successfully, and is no longer running.
 - Canceled — job was canceled before it completed, and is no longer running.
 - Failed — the step failed and is no longer running.
- **File** (only if a Backup Stations step) — File path and name on Supervisor for the saved backup .dist file. Uses convention:

^provisioningNiagara/stationData/*stationName*/backups/backup_*stationName*_yyymmdd_hhmm.dist

Step log table

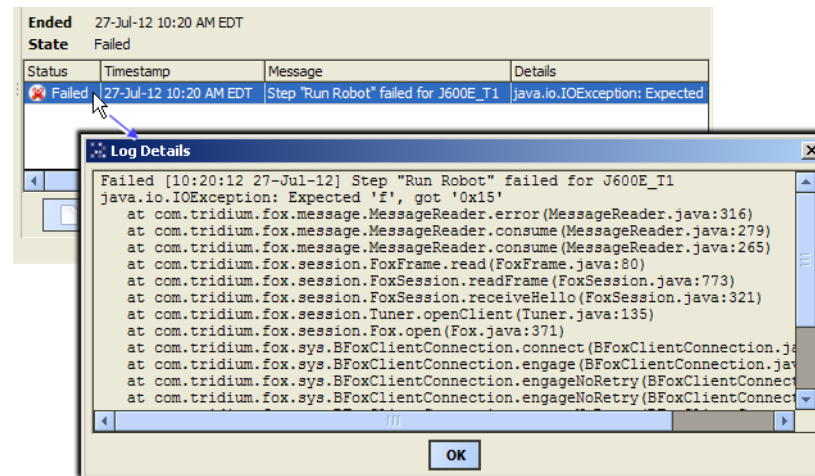
This main area of the [Batch Job Step Log File View](#) shows the log items written during the step execution.

Columns in the step log table include the following:

- **Status** — Status of the step processing when the log message was written, for example: Message, Success, or Failed.
- **Timestamp** — Date/time when the log message was written.
- **Message** — The actual log message.
- **Details** — Any additional details (or the beginning of such details).

Note: Double-click any step log message to see additional details, if any (see [Figure 3-13](#)). For example, an error message may include a “stack trace” in the details.

Figure 3-13 Log Details dialog on message in step log

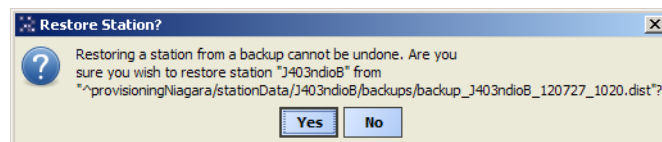


Buttons

Buttons at the bottom of the [Batch Job Step Log File View](#) are described as follows:

- **Job Summary** — (always enabled) Click to go “up a level” to the [Batch Job Log File View](#) or [Niagara Network Job View](#) (the latter if job is running or recent).
- **Refresh Log** — Enabled only if job is running. Clicking reloads the information from the job (available because this view cannot get new log entries as they are written on the Supervisor).
- **Restore** — Enabled if it was a [Backup Stations step](#), and it completed successfully. Click for a confirmation dialog to install the station from the backup .dist saved in this job step ([Figure 3-14](#)).

Figure 3-14 Restore Station Confirmation



Note: Read the **Restore Station** confirmation carefully. There is no “undo” after restoring a station backup.

If you confirm to restore:

- A provisioning job is queued to install the backup .dist.
- The view changes to the [Niagara Network Job View](#) for the new (install backup) job.

Note: If the step was a **Run Robot** step, these additional buttons are available at the bottom of the [Batch Job Step Log File View](#) or [Job Step Details View](#):

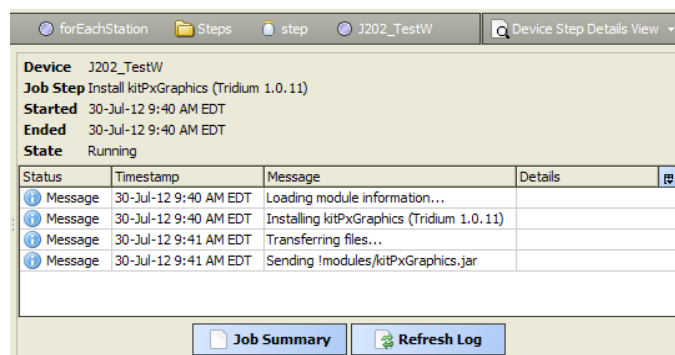
- **Robot Source** — (always enabled) Click to view the program code of the ProvisioningRobot.
- **Robot Log** — Enabled after job has completed. Click to see the log output from the execution of the ProvisioningRobot.

For more details, see “[Run Robot step view features](#)” on page 3-32.

Job Step Details View

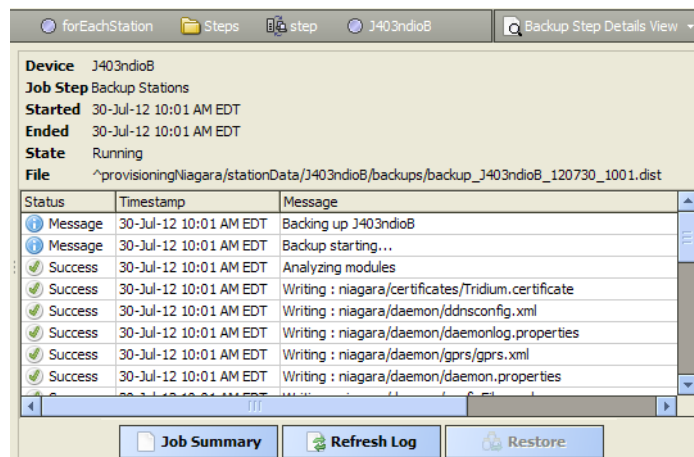
The Job Step Details View ([Figure 3-15](#)) shows details for one provisioning job step *currently running* against a single station. Typically, you do *not* see this view, unless from the [Niagara Network Job View](#), you click the “>>” (Details) button on *most running* steps. After the job completes and you ask for step details, instead of this view you typically see the [Batch Job Step Log File View](#).

Figure 3-15 Job Step Details View lists details on one running step for a station.



Note: If a currently running backup station step, you see a slightly different view still: the **Backup Step Details View** ([Figure 3-16](#) below). Note that the view for a “Run Robot step” also provides additional buttons. See “[Run Robot step view features](#)” on page 3-32.

Figure 3-16 Backup Step Details View lists details on one running backup step for a station.



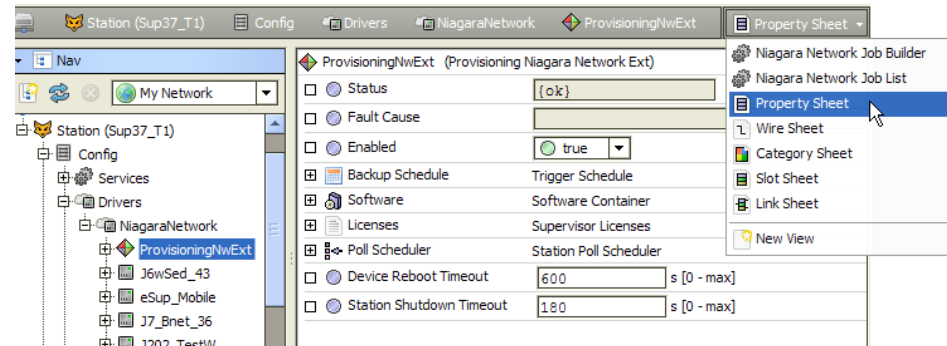
All views on running steps are nearly identical to the [Batch Job Step Log File View](#). For more details, please see the following **Batch Job Step Log File View** sections:

- “[Step elements \(read-only\)](#)” on page 3-10
- “[Step log table](#)” on page 3-10
- “[Buttons](#)” on page 3-10

ProvisioningNwExt slots

The **ProvisioningNwExt** for the NiagaraNetwork has a number of frozen slots, both containers (some with special views) as well as a few properties. Access them through the component's property sheet (Figure 3-17).

Figure 3-17 ProvisioningNwExt property sheet



See the following sections for more details:

- [ProvisioningNwExt properties](#)
Note: Two timeout properties near the bottom of the property sheet are available starting in AX-3.5. They help prevent the scenario of a failed remote station hanging an entire provisioning job.
- [ProvisioningNwExt container components](#)

ProvisioningNwExt properties

Properties of the ProvisioningNwExt (Figure 3-17) are described as follows:

- **Status** — Is either `fault` if provisioning is not licensed, `disabled` if `Enabled` property is false, and in all other cases is `ok`.
- **Enabled** — If set to false, provisioning activity cannot occur, and provisioning-related extensions in NiagaraStations are set to disabled.
- **Fault Cause** — Text explanation for why the extension is in fault.
- Four container components (see next section “[ProvisioningNwExt container components](#)”).
- **Device Reboot Timeout** — Specifies the maximum time in seconds that a provisioning job step will wait while attempting to reconnect to a station that has been rebooted before setting the step status to “failed”, and advancing to the next station. The default time is 600 seconds (10 minutes).
- **Station Shutdown Timeout** — Specifies the maximum time in seconds that a provisioning job step will wait for a station to stop after receiving a shutdown request before setting the step status to “failed”, and advancing to the next station. The default time is 180 seconds (3 minutes).

ProvisioningNwExt container components

As shown in Figure 3-17, the ProvisioningNwExt contains these other frozen child components:

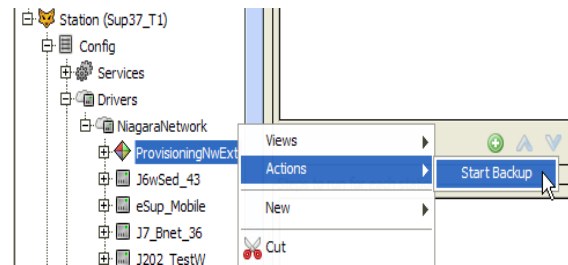
- **Backup Schedule** — A standard TriggerSchedule with its trigger output pre-linked to the ProvisioningNwExt's “Start Backup” action, which when invoked performs a backup of all stations. This is just a convenience you can use to set a regular schedule for a backup of all stations at the same time. Simply double-click it for its Trigger Scheduler view.
Note: Use of this Backup Schedule is not advisable on a larger enterprise system, where the Supervisor has many subordinate JACE stations. Instead, it is recommended that you use “job prototype” components (NiagaraNetworkJobPrototype) copied from the provisioningNiagara palette. For more details, see “[About the Niagara Network Job Prototype](#)” on page 3-17.
- **Software** — Software Container that summarizes software versions available on the Supervisor to be installed into stations. For more details, see “[Supervisor Software container](#)” on page 3-14 and “[Supervisor Software Manager](#)” on page 3-14.

- **Licenses** — Provides views into licenses installed in NiagaraStations under the NiagaraNetwork, as well as the Supervisor’s “license database.” For more details, see “[Licenses slot](#)” on page 3-16.
- **Poll Scheduler** — The Station Poll Scheduler controls the polling of the [StationProxy](#) extensions by the ProvisioningNwExt. This poll scheduler operates as in most driver networks. See “About poll components” in the *Drivers Guide* for general information. Polling only happens for station proxy objects that are subscribed (linked somewhere, or being viewed or charted, etc.).
Note: Often, you can leave poll configuration at defaults. Alternatively, assign the Poll Frequency property in the [StationProxy](#) extensions (NiagaraStation provisioning extensions) differently than the default “Normal.”

ProvisioningNwExt action

As shown in [Figure 3-18](#), the ProvisioningNwExt has a single **Start Backup** action.

Figure 3-18 Start Backup action of ProvisioningNwExt



When invoked, a provisioning job to backup *every* NiagaraStation (providing its Backup extension is enabled) is immediately submitted to the batch job queue.

Note that invoking this action is generally equivalent to doing the following:

- Double-clicking the **ProvisioningNwExt** for its **Niagara Network Job Builder** view.
- Adding a Backup Stations step in the view’s “Steps for each station List”.
- Adding all stations in the view’s Stations List.
- In the buttons area, clicking **Run Now**.

However, note either method (single action shown, or equivalent steps) is generally *not recommended*.

Why the Start Backup action is typically NOT recommended

On a large system where a Supervisor has many JACEs, invoking the action shown above (or equivalent steps listed) creates a provisioning job that can take an excessive amount of time. Further, it can put an undue load on the system at a peak time—when an administrator invokes it. Finally, unlike a provisioning job from a NiagaraNetworkJobPrototype (the preferred way), the accumulated backup .dist files remain stored on the Supervisor until manually deleted—not available to “job retention” policies. Without manual intervention, over a long period of time this could lead to a “disk full” condition on a Supervisor.

What *is recommended*, particularly for a large enterprise system where a Supervisor has many JACEs, is to add multiple **NiagaraNetworkJobPrototype** components in the Supervisor’s station, copying from the provisioningNiagara palette. Then you can configure each one for a custom backup job, selecting *some* of the system’s JACE stations in each one.

To run each backup job at some periodic interval, perhaps at an “off hours” time, you could add and link a standard **TriggerSchedule** component (also available on the provisioningNiagara palette). By using multiple TriggerSchedules (one configured slightly differently for each linked NiagaraNetworkJobPrototype), backups could constructively “stage” in sequence—say 10 minutes apart from one another.

Additionally, each NiagaraNetworkJobPrototype component has configurable “job retention” policies, via its “Prototype Job List” view. You can (and typically should) configure them to provide automatic disposal of older saved backup .dist files, based on age or some number of earlier saved backup .dist file. For an example procedure including NiagaraNetworkJobPrototypes, see “[Example Supervisor configured for “Job Prototype” backups of stations](#)” on page 2-4.

For other related details, see the following sections:

- “[About the Niagara Network Job Prototype](#)” on page 3-17
 - “[Scheduling a job prototype](#)” on page 3-18
 - “[Job retention](#)” on page 3-19
 - “[Job actions](#)” on page 3-20
 - “[Niagara Network Prototype View](#)” on page 3-21

Supervisor Software container

The Software container under the ProvisioningNwExt provides a “window” into the Supervisor’s software registry (installable software files stored under its `!sw` directory), and is populated by components that represent these files. When the service starts (station startup), it starts a thread that scans the software registry and populates this container. The Software container also lists software files added to the registry after the initial scan.

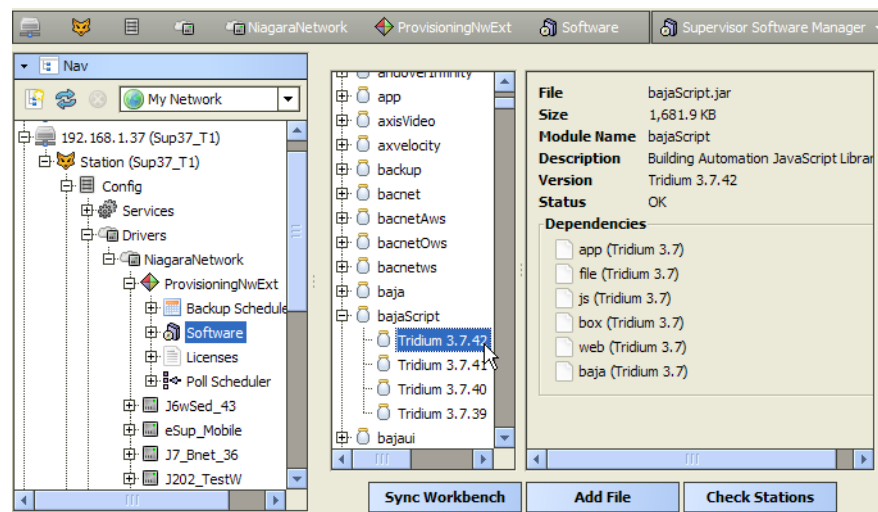
Direct children of the Software container are summary objects ([InstallableSummary](#) components) for named, typed software files (for example, file type module named “baja”). For each summary object, there is a specification object ([InstallableSpec](#) components) for each version in the registry.

Apart from these summary children, the Software container has but a single frozen property: “loaded”, a boolean slot indicating if the startup thread has finished scanning the registry (by default, it is hidden). Your key interface to the Software container is its default view: the [Supervisor Software Manager](#).

Supervisor Software Manager

The default view of the **Software** container under the ProvisioningNwExt is the **Supervisor Software Manager** ([Figure 3-19](#)).

Figure 3-19 Supervisor Software Manager is default view of Software container



This view provides the following:

- Visibility into the software registry on the Supervisor.
 - Ability to add software files in the software registry on the Supervisor.
 - Ability to compare software versions against stations under the Supervisor’s NiagaraNetwork.
- Parts of this view include the [Software tree](#) pane, [Details pane](#), and bottom-located [buttons](#).

Software tree

The left pane of the [Supervisor Software Manager](#) shows all child module and distribution file nodes, where each root node represents a software file’s name, and its children represent the versions of the software file in the registry. You can drop a module or distribution file from Workbench’s Nav tree into this software tree.

If it is an acceptable file (that is readable and correctly formatted), its version will be checked against the software database of the Supervisor. If not a duplicate, it is transferred to the Supervisor and becomes registered in its software database.

Details pane

This right pane of the [Supervisor Software Manager](#) shows details about any version node selected in the [Software tree](#), including file name, size, description, version, and all dependencies.

Buttons

Buttons near the bottom of the [Supervisor Software Manager](#) are described as follows:

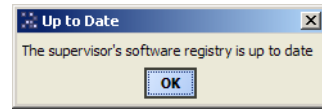
- **Sync Workbench**— (always enabled) Available to update the Supervisor’s software database to include all installable files in your Workbench’s software database (under its `!sw` directory). See the next section “[Sync Workbench](#)”.

- **Add File** — (always enabled) For the standard **File Chooser** dialog, in which you can navigate to the location of the software file for a module or distribution file.
- **Check Stations** — enabled when you have a software node selected in the **Software tree**.

Sync Workbench When you click this in the **Supervisor Software Manager**, the comparison between the different software databases is made.

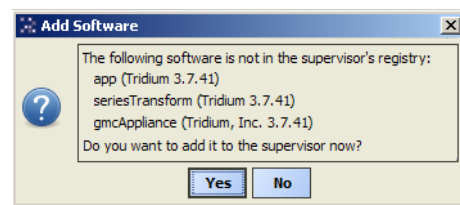
- If the Supervisor already has all the software installable files that your Workbench environment has, a popup dialog informs you that the “provisioning software is up to date” (Figure 3-20).

Figure 3-20 Up to Date dialog from Sync Workbench



- If your Workbench environment has installable files that the Supervisor does not, an **Add Software** popup dialog lists these files, and asks if you wish to transfer them to the Supervisor, as shown in Figure 3-21. You can select **Yes** (to transfer) or **No** to cancel, as desired.

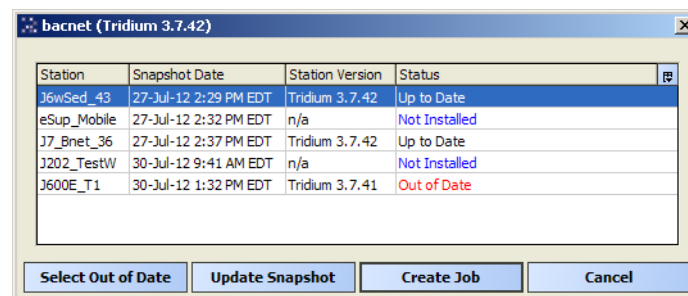
Figure 3-21 Add Software dialog from Sync Workbench



Check Stations When you click this in the **Supervisor Software Manager**, the selected version (or latest version, if a root node is selected) is compared against the “platform snapshot” in each of the NiagaraStation’s **Software** station extensions.

A dialog is then displayed for that module, as shown in Figure 3-22.

Figure 3-22 Example dialog for installable from Check Stations function



This dialog shows a table with rows for each NiagaraStation, displaying the version of the software in its platform snapshot (viewable in its Software extension).

The last column in the table shows the status of the platform snapshot for each station, which can have one of the following values:

- **Up to Date** — station version is equal to or greater than the software file in the Supervisor.
- **Out of Date** — Supervisor software file version is greater than the version installed on the station’s host.
- **Not Installed** — this software file is not installed on the station’s host.
- **No Snapshot** — No platform snapshot has been taken for this station, so there is no basis for comparison. Click to select, then click **Update Snapshot**.
- **Bad Remote File** — Station’s host has a version that is corrupt or otherwise unusable.

Buttons in this dialog provide added functionality.

Buttons As shown in Figure 3-22, buttons in the dialog produced by **Check Stations** are as follows:

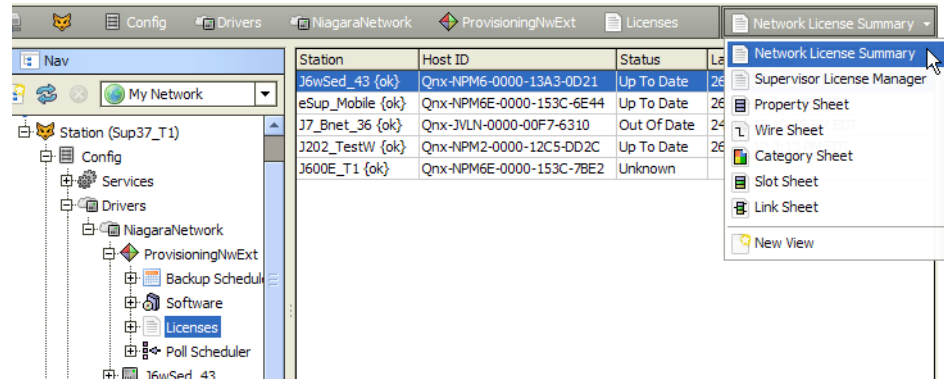
- **Select Out of Date** — Selects each of the rows (stations) with an “Out of Date” status.
- **Update Snapshot** — For the selected station(s), queries that host’s platform daemon and builds a “snapshot” of installed software. Use this for any station showing a status of “No Snapshot”.

- **Create Job** — Closes this dialog, and changes the view to the [Niagara Network Job Builder](#).
- **Cancel** — Closes this dialog, and takes no other action.

Licenses slot

Among [ProvisioningNwExt slots](#), the **Licenses** slot under the NiagaraNetwork's **ProvisioningNwExt** provides two different *views* of licenses related to this Niagara Network. It has no direct component children, and a single property: “Allow License Server Access” (by default, is `true`).

Figure 3-23 Licenses slot of ProvisioningNwExt provides two views



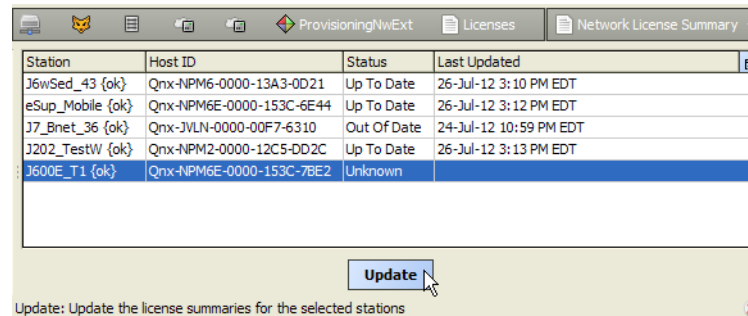
The two Licenses views are:

- [Network License Summary](#) — Provides a summary table listing the currently known license information for each station (NiagaraStation) in the network.
- [Supervisor License Manager](#) — Provides a manager view into the Supervisor’s “local license database”, similar to the “Workbench License Manager” view (Tool) for any Workbench PC.

Network License Summary

The default view of the **Licenses** slot under the ProvisioningNwExt is the **Network License Summary** (Figure 3-24).

Figure 3-24 Network License Summary is default view of ProvisioningNwExt’s “Licenses” slot



This view provides summary information about the license in each host running a station, and is sourced by the “Licenses” device extension of each child NiagaraStation station. If you double-click on a row, the view changes to the property sheet of the “[Licenses](#)” extension for that particular NiagaraStation.

Note: A status of “Up To Date” simply means that the license on the remote host agrees with the license that the Supervisor has for it in its (own) local license database. It may be possible that a more recent license is available for it on the licensing server.

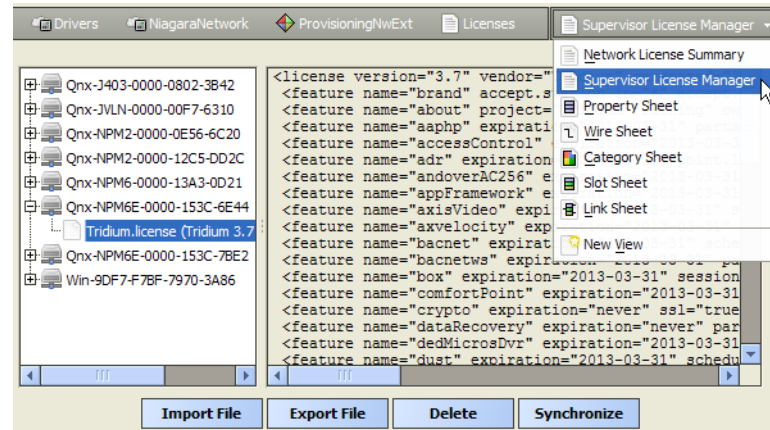
For more details and related information, refer to the Platform Guide sections “About the licensing server”, “About the local license database”, and “About license archive (.lar) files”.

As needed in the Network License Summary view, you can select one more stations (license rows), and **Update**. An update requests the latest license from the online licensing server, and if a newer one is found (than already installed), installs it in the JACE, as well as update that license in the Supervisor’s local license database. The “Last Updated” timestamp is also reset to the time of the update command.

Supervisor License Manager

The **Supervisor License Manager** (Figure 3-24) is an available view of the **Licenses** slot under the ProvisioningNwExt.

Figure 3-25 Supervisor License Manager view is available view on ProvisioningNwExt's License slot



This view provides access and management of the Supervisor's "local license database"—that is, the structured organization of "host ID-named" subfolders and contained license files that reside under the Supervisor's `! /licenses` folder. As in the equivalent "Workbench License Manager" view, it 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 the Supervisor's 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 the Supervisor's local license database.
- **Synchronize** — Typically always available if you have Internet connectivity. This lets you *update* all licenses (or any selected licenses) in Supervisor's local license database with the *most current* versions, via the online licensing server.

For more details and related information, refer to the *Platform Guide* sections "About the licensing server", "Workbench License Manager", "About the local license database", and "About license archive (.lar) files".

About the Niagara Network Job Prototype

A NiagaraNetworkJobPrototype component (or simply a "job prototype") is a persisted component that specifies a Niagara provisioning job—meaning a specific set of steps to be performed on a given list of Niagara stations. You typically link the out (Trigger) slot of a standard TriggerSchedule to the component's "Submit Job" action, to specify some *future* (and possibly *periodic*) execution.

Because of available "job retention policies", these are the *preferred components* to configure regular station backup provisioning jobs—something otherwise unavailable when setting up a provisioning job from the ProvisioningNwExt (Niagara Network Job Builder view).

Add these components in the Supervisor station for provisioning its NiagaraNetwork. A copy is in the provisioningNiagara *palette*, as a "**NiagaraNetworkJobPrototype**" component. Like other components, you can duplicate, modify, and reuse these jobs as needed in the Supervisor station.

Note: *Locate these job prototype components anywhere needed in the architecture of the Supervisor's station database—for example, they do not need to be under the ProvisioningNwExt, or even the NiagaraNetwork.*

Figure 3-26 Copying a NiagaraNetworkJobPrototype from the provisioningNiagara palette

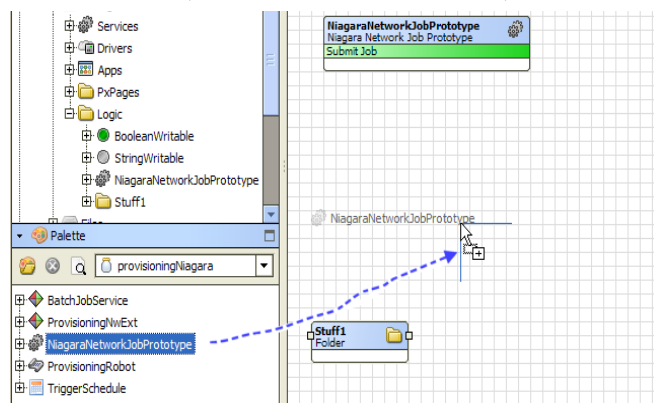


Figure 3-26 shows a job prototype (NiagaraNetworkJobPrototype component) added, copied/dragged from the palette.

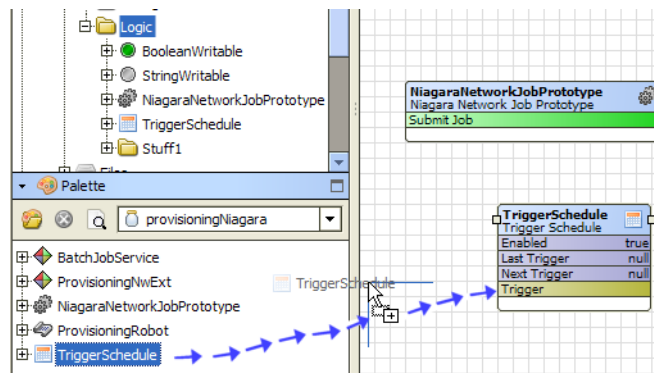
Note: Alternatively, you can quickly make a “one-time/run-now” job using the [Niagara Network Job Builder](#). However, unlike job prototypes, that job cannot be saved as a component for reuse. Furthermore, as a best practice you should run a “Backup Stations” step only in job prototypes, for “job retention” purposes. Further details on job prototype components are in the following subsections:

- [Scheduling a job prototype](#)
- [Job retention](#)
- [Job actions](#)
- [Niagara Network Prototype View](#) (default view, and “main interface”)
- [Prototype Job List](#) (another available view)

Scheduling a job prototype

You can link a [job prototype](#) component to a TriggerSchedule for scheduled execution. For convenience, the provisioningNiagara palette provides a TriggerSchedule for drag and drop copying (Figure 3-27).

Figure 3-27 Copying a TriggerSchedule from the provisioningNiagara palette

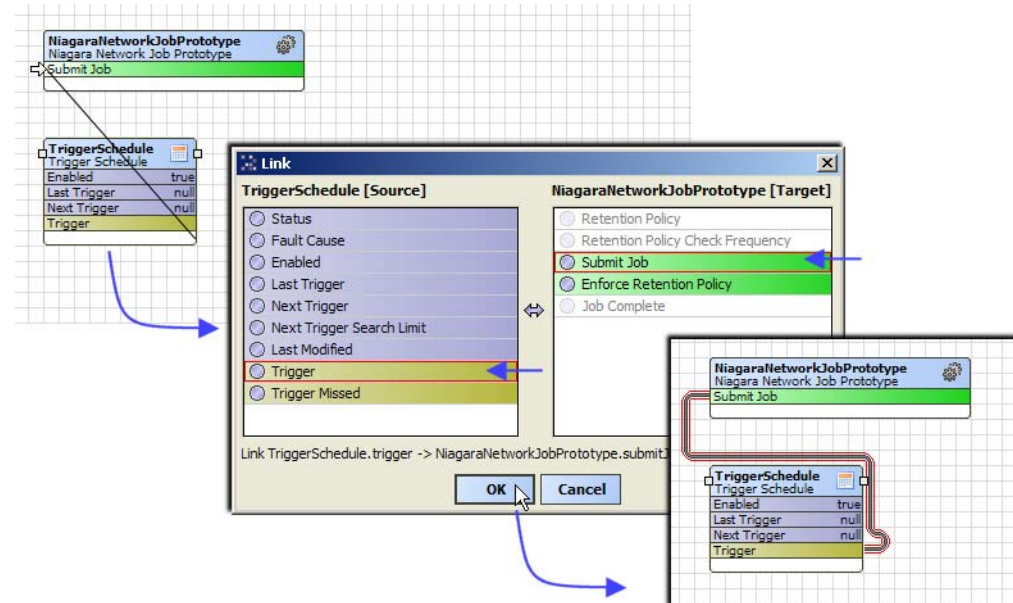


This is a standard TriggerSchedule component, identical to the one found in the schedule palette.

To schedule, you link the “Trigger” slot of a source TriggerSchedule to the “Submit Job” (action) slot of the job prototype component, as shown being done in Figure 3-28.

Note: Do not link a job prototype to the default “Backup Schedule” under the ProvisioningNwExt—that TriggerSchedule is already linked to the “Start Backup” action of the extension. Those components are not recommended for regular provisioning jobs that include station backups. For background details, see “[Why the Start Backup action is typically NOT recommended](#)” on page 3-13.

Figure 3-28 Linking copied TriggerSchedule to job prototype component



To define the schedule, double-click on the linked TriggerSchedule for its Scheduler view. You can set up a one-time trigger in the future, or define regular (repeating) triggers, or define any combination of such trigger events. For more details, see the “Trigger Scheduler view” section in the *User Guide*.

Note: *If desired, you can copy a TriggerSchedule under a job prototype component (using it as the container), and make this same link. This provides portability. Remember if you duplicate/copy the parent **job prototype** after configuring the schedule, all copied job components will have that same scheduling configuration.*

Job retention

Unique to a **job prototype** (or more accurately, any type of batch job component), is the feature of *job retention*—essentially a “disposal” configuration for its previously executed jobs. When a batch job is disposed, any associated files stored on the Supervisor are deleted, and that job is removed from the various job list views.

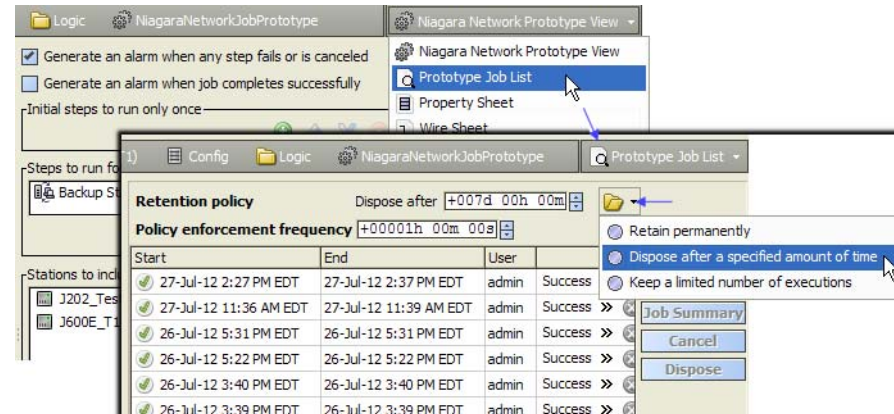
Disposal is especially *important* in regards to jobs with *backup steps*, each of which results in an associated backup .dist file to be stored for each station. Over time, retaining all such jobs can consume an *excessive amount of file space* on a Supervisor.

Although a Supervisor platform typically has large amounts of hard drive storage, it is common (and recommended) for it to be periodically backed up to removable media. Thus, it would be unwise to retain forever all daily (or even weekly) backups for even a modestly-sized system—that would needlessly create a *huge* number of backup files under the station’s provisioningNiagara directory. This is where automatic disposal via *job retention* configuration is useful.

By default (as copied from the provisioningNiagara palette), a job prototype component’s retention is set for all of its jobs to be *permanently* retained (until manually disposed). However, you can (and often *should*) modify a job prototype’s retention policy such that some executed jobs will be automatically disposed—depending on either the “elapsed time” or by reaching some number of job executions.

Access a job's retention properties on either the top of its **Prototype Job List** view (Figure 3-29), or in the component's property sheet.

Figure 3-29 Retention properties for NiagaraNetworkJobPrototype component

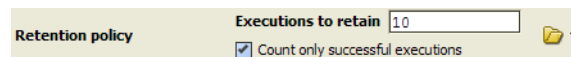


The two retention properties are described as follows:

- **Retention Policy** — This determines how long (or many) executed batch jobs are *retained* in the station's job management system, after which they are disposed. Choices include:
 - **Retain permanently** — All executed batch jobs remain in the Supervisor's station job management system, until manually disposed.
 - **Dispose after a specified amount of time** — Executed batch jobs are disposed after some period of time (relative to their "end" timestamp). The field editor allows selection of a period including days, hours, and minutes (default 7 days).



- **Keep a limited number of executions** — Only the most recent number *n* of executed batch jobs are retained, after which older jobs are disposed. The field editor provides a numerical entry field (for example, 10 executions). Included is a checkbox (flag) to specify whether only successful job executions are to be counted.



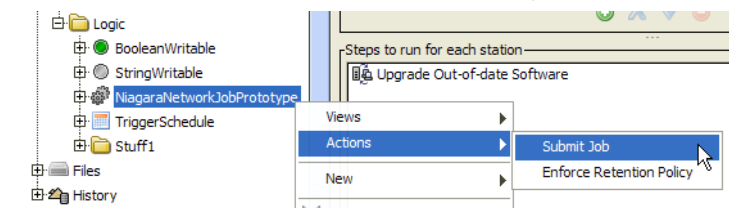
- **Retention Policy Check Frequency** — (1 hour for job from palette). The periodic frequency at which the job's Retention Policy is evaluated and enforced. Note that you can also manually invoke the "Enforce Retention Policy" action on the job component. See the next "Job actions" section.

Note: Keep in mind that if you configure retention properties and then duplicate/copy the *job prototype*, that all copied job components will have that same retention configuration.

Job actions

Each *job prototype* has two available right-click actions, as shown in Figure 3-30.

Figure 3-30 Actions for a NiagaraNetworkJobPrototype



The **Submit Job** action is equivalent to the **Run Now** button when in the job prototype's default *Niagara Network Prototype View*. The **Enforce Retention Policy** action immediately applies the job's retention policy. For related details, see the section "Job retention" on page 3-19.

Niagara Network Prototype View

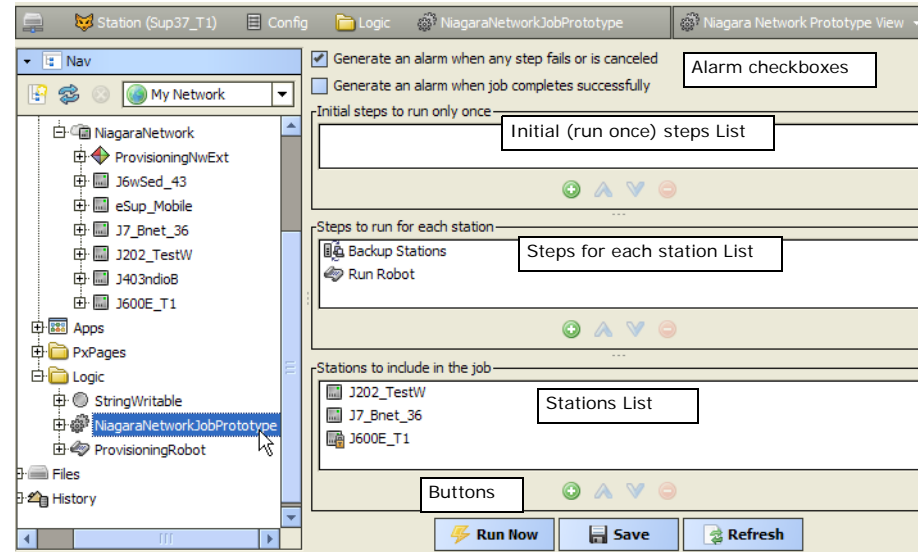
The default view on any (reusable) Niagara [job prototype](#), you use it to specify or otherwise edit this particular Niagara provisioning job. As shown in [Figure 3-31](#), this view is similar to the [Niagara Network Job Builder](#) view (of the ProvisioningNwExt), in that it has *three* list areas:

- [Initial \(run once\) steps List](#)
- [Steps for each station List](#)
- [Stations List](#)

Also as in the Niagara Network Job Builder view, there are [alarm checkboxes](#) and buttons.

Note: The additional **Save** button in the **Niagara Network Prototype View** is mainly how this view differs from the **Niagara Network Job Builder** view. For more details, see [“Buttons”](#) on page 3-23.

Figure 3-31 NiagaraNetworkPrototypeView is default view of a NiagaraNetworkJobPrototype



Initial (run once) steps List

This *top* area of the [Niagara Network Prototype View](#) shows a one-line summary for each step to be performed *once* in the job, regardless of how many stations will be specified. Currently, when provisioning Niagara this means only one step type: “Update Licenses”, and is optional. (You may wish to skip this step, but add other types of steps in the middle “[Steps for each station List](#)” pane).

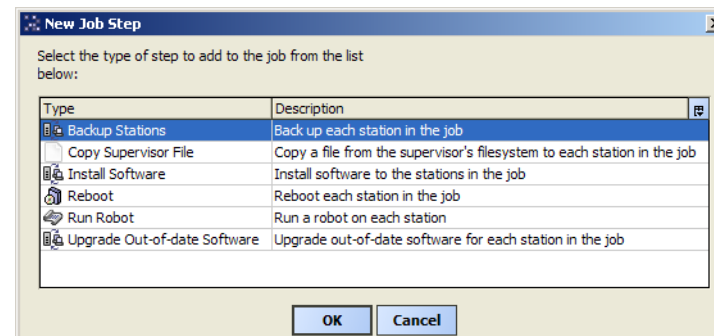
To add this step, click the “+” (add) button below this list—see [“Adding Job List Steps”](#) on page 3-21 for further details. For details on this initial step, see [“Update Licenses step”](#) on page 3-26.

Steps for each station List

This *middle* area of the [Niagara Network Prototype View](#) shows a one-line summary for each step to be run for each station specified in this job. In most provisioning jobs, you [add](#) one or more steps. As needed, you can also [remove](#) and [reorder](#) steps (when the job runs, steps execute in the top-to-bottom order).

Adding Job List Steps

Figure 3-32 New Job Step popup menu (Niagara Network Prototype View)



Add a provisioning job step by using any of the following methods:

- Click the “+” (add) button below the list, and choose the step type from the **New Job Step** popup menu (see [Figure 3-32](#)). Note that all the same types of job steps are available as in the [Niagara Network Job Builder](#), with the exception of the [Copy Local File step](#). For details about step types, see “[Niagara provisioning job steps](#)” on page 3-25.
- Right-click in the steps list, select **Add**, and choose the step type from the popup menu.
- Drag a file from Workbench’s Nav tree into the steps list (implicit [Copy Supervisor File step](#)).
- Drag a software item (module or dist) from Workbench’s Nav tree that appears under the ProvisioningNwExt’s “**Software**” container, into the Job Steps List (implicit [Install Software step](#)).
- Drag a ProvisioningRobot that exists in the station’s Config (component) architecture into the steps list (implicit [Run Robot step](#)).

Removing Job List Steps

Remove a provisioning job step using either of these two methods:

- Click to select the step, then click the “X” (remove) button below the list.
- Right-click the step, and select **Remove** from the popup menu.

Reorder Job List Steps

Reorder a selected job step by clicking the (up) or (down) *arrow* button at the bottom of the list. Or, right-click a job step and select **Move Up** or **Move Down**, as needed.

Note: *Job steps are executed in the same top-to-bottom order as defined in the two step lists: “initial steps” first, then “steps for each station”.*

Stations List

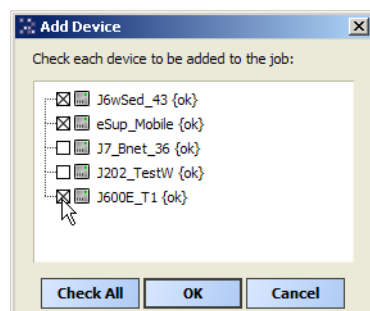
This lower area of the [Niagara Network Prototype View](#) lists all the stations to be processed by the job—note this means each station processes all steps in the (middle pane) “[Steps for each station List](#)”. Only stations in the Supervisor’s NiagaraNetwork can be added. For any job, you [add](#) one or more stations, and you can also [remove](#) and [reorder](#) stations (stations are processed in a top-to-bottom order).

Adding Stations

Add a station to a job in the [Niagara Network Prototype View](#) by doing any of the following:

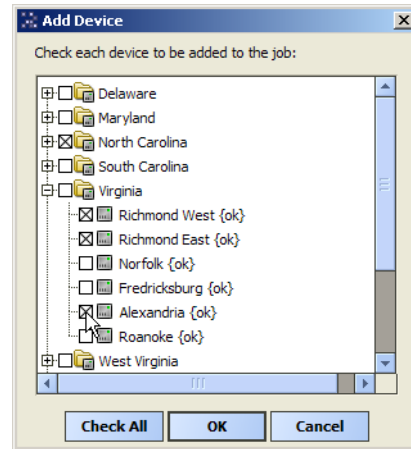
- Click the “+” (add) button below the list, and choose the station(s) in the **Add Device** popup dialog (see [Figure 3-3](#) and [Figure 3-4](#)). A **Check All** button allows you to choose all stations.
- Right-click in the Stations List, select **Add**, and choose the station(s) in the same popup dialog.
- From the Nav tree, simply drag a station from under the NiagaraNetwork into the Stations List.




Figure 3-33 Add Device dialog (small system example)



Note that the dialog for adding stations reflects any station folders (NiagaraStationFolder) in an expandable tree structure. Also, folders and stations are listed using any assigned “display names” ([Figure 3-4](#)).

Figure 3-34 Add Device dialog (large system using station folders, display names)



This station tree structure using station display names (if assigned) can be useful on Supervisors that have very large numbers of subordinate JACE stations. Click on controls to expand  and collapse  station folders, as needed. A selected  folder will select *all nodes* under it (stations and any subfolders).

Removing Stations

Remove a station from a job in the [Niagara Network Prototype View](#) using either of these two methods:

- Click to select the station, then click the “X” (remove) button below the list.
- Right-click the station, then select **Remove** from the popup menu.

Reorder Stations

Reorder a selected station in the [Niagara Network Prototype View](#) by clicking the (up) or (down) *arrow* button at the bottom of the list. Or, right-click a station and select **Move Up** or **Move Down**, as needed.

Note: *Stations are processed in the same top-to-bottom order as defined in the Stations List.*

Alarm checkboxes

The two checkboxes at the top of the [Niagara Network Prototype View](#) determine if alarms are to be issued by the **BatchJobService** for this provisioning job, and under what circumstances. Alarms use the alarm class specified in the property sheet of the BatchJobService, and appear listed in an alarm console as *alerts*.

- **Generate an alarm when any step fails or is canceled** — if checked, an alarm is raised whenever a job step fails or is canceled.
- **Generate an alarm when a job completes successfully** — if checked, an alarm is raised whenever a job completes with no step failures.

Note: *Checkbox settings apply to the provisioning job being built, and do not affect other provisioning jobs that may exist (either as other provisioning components, jobs already queued to run, or built in the [Niagara Network Job Builder](#)).*

For further details, see “[About provisioning-related alarms](#)” on page 3-35.

Buttons

The [Niagara Network Prototype View](#) has three buttons along the bottom, which from left-to-right are:

- **Run Now** — Enabled when there is at least *one* job step in either the “[Initial \(run once\) steps List](#)” or “[Steps for each station List](#)”, and at least *one station* in the “[Stations List](#)”. Click to dispatch the job to the batch job queue for immediate execution. The Workbench view automatically changes to the [Niagara Network Job View](#).

Note: *If unsaved changes (**Save** button enabled), a popup dialog asks if you wish to save the configuration in this view (before queuing to run).*

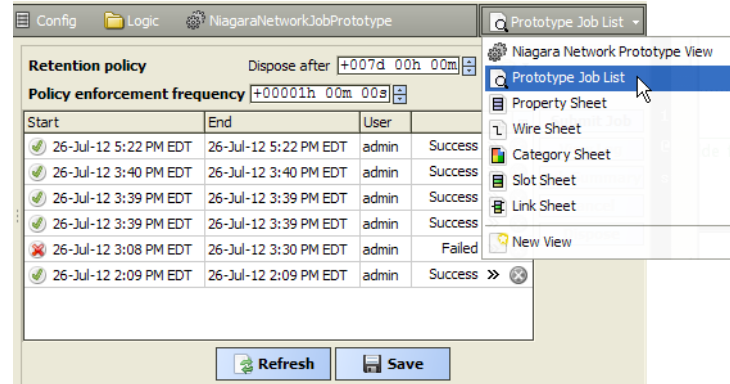
- **Save** — Enabled whenever you have unsaved changes in this view. Click to immediately save the changes to the corresponding [job prototype](#) component.
- **Refresh** — Always enabled. If unsaved changes exist, this prompts if you wish to save. If you answer no, the view is restored to the last saved configuration.

Prototype Job List

Another view on any Niagara [job prototype](#), access it via the view selector ([Figure 3-35](#)), or by right-clicking the NiagaraNetworkJobPrototype and selecting **Views > Prototype Job List**.

As shown in [Figure 3-35](#), this view has one list area: **Jobs Table**, with [buttons](#) on the right and the bottom.

Figure 3-35 Prototype Job List is another view of a NiagaraNetworkJobPrototype component



This view differs from the [Niagara Network Job List](#) (NiagaraNetwork's **ProvisioningNwExt** view) in the following ways:

- The two “retention properties” for the associated job prototype component are shown at top, where you can adjust if needed.
Note: Setting retention policy is recommended for any job that includes ongoing periodic backups. For more details, see “Job retention” on page 3-19.
- This **Jobs Table** area shows only jobs from the (one) associated Niagara [job prototype](#), whereas the **Niagara Network Job List** table shows *all* retained Niagara provisioning jobs.

Jobs Table

This main area of the [Prototype Job List](#) shows Niagara provisioning jobs that have been sent to run, are running, or are have completed. Note that any “pending jobs” do not appear until the job prototype’s linked trigger schedule actually fires.

Jobs table columns in the job list include the following:

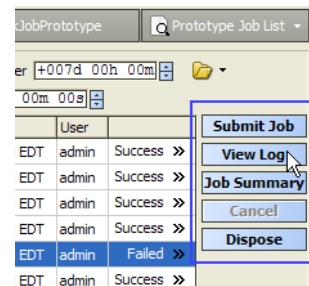
- Start** — time the job was queued.
- End** — time the job terminated (successfully, failed, or canceled).
- User** — station user that requested the job (shows “unknown” if job triggered by a linked schedule).
- Status** — the status of the jobs, as one of the following:
 - Unknown — job is pending execution, because all threads in the job queue are in use.
 - Running — job is executing.
 - Canceled — request to cancel the job was sent, but has not been processed yet, and the job is still executing.
 - Success — job finished successfully, with all steps completed for all stations.
 - Canceled — job was canceled before it completed, and is no longer running.
 - Failed — at least one step failed in one station; job is no longer running.

The status column ends with a “>>” (Details) and “X” (Dispose) button for each job, these function the same as the **Job Summary** and **Dispose** buttons at the right side of this view.

Buttons

Most buttons along the right side of the [Prototype Job List](#) become enabled when you have a job row selected in the **Jobs Table**. See [Figure 3-36](#).

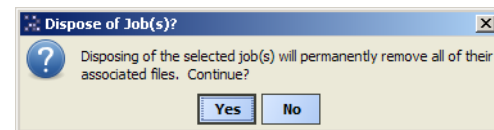
Figure 3-36 Buttons along right side of Prototype Job List view.



Buttons are described as follows:

- **Submit Job** — Always enabled. This lets you request this prototyped job to run now. A new job row is added atop the **Jobs Table**, with its status dynamically updated to show the job's progress.
- **View Log** — For a popup **Job Log** dialog, showing log messages output by the selected job. See “**Job Log notes**” on page 3-6 for more details.
- **Job Summary** — To change to the **Niagara Network Job View**, to display the selected job in detail.
- **Cancel** — Enabled only if selected job has Running status. If clicked, the job is notified it should cancel when safe to do so. Note that not all job steps can be canceled.
- **Dispose** — For a confirmation dialog to delete the selected job(s), including all associated job files (**Figure 3-37**).

Figure 3-37 Dispose Job Confirmation



If you confirm to dispose:

- Job(s) is deleted from the JobService (if still there, not rolled off as 11th job, or station restart).
- All associated job files are *deleted*, including the job log file, station step log files, and other files if applicable. Note for a backup job, this includes *deleting the backup .dist file(s)*.
- Job(s) is removed from the **Jobs Table** in the both the **Prototype Job List** and also in the **Niagara Network Job List**.

Note: You can use the **Dispose** button with multiple selected jobs, if needed.

Niagara provisioning job steps

Any Niagara Network provisioning job contains one or more component *steps*. Typically, you add steps in either of the “step lists” areas (“Initial—Run once” and/or “Steps to run for each”) in either of these two views:

- **Niagara Network Job Builder** (view on the NiagaraNetwork’s **ProvisioningNwExt**)
- **Niagara Network Prototype View** (view on a particular Niagara **job prototype**).

Both views offer identical controls to add , remove , or reorder   steps.

After any step completes, you can view details on it by accessing the **Batch Job Step Log File View**.

The different *types* of provisioning steps you can include in a Niagara provisioning job include the following:

- **Update Licenses step**
- **Backup Stations step** (recommended to run from a **job prototype**, typically)
- **Copy Supervisor File step**
- **Copy Local File step** (available in the **Niagara Network Job Builder** view on **ProvisioningNwExt**)
- **Install Software step**
- **Reboot step**
- **Run Robot step**
- **Upgrade Out-of-date Software step**

Update Licenses step

Currently, when adding [Niagara provisioning job steps](#), the Update Licenses step (UpdateLicenses-JobStep) is the *only* step option for the “Initial steps to run only once” (top) step list in either the [Niagara Network Job Builder](#) or [Niagara Network Prototype View](#). When processed by the Supervisor, it gathers information on the licenses installed on the target station's host(s), and then accesses the online *licensing server* (in one message) to see if the licenses are up-to-date.

If any updated (different) licenses are found, they are installed to the target station's host(s)—and also updated in the Supervisor's “local license database.”

Note: *If the Supervisor is not configured for Internet connectivity, then only its local license database is used to compare against licenses installed on the target station's host(s). If any updated (different) licenses are found, they are installed on the target station's host(s).*

For related information, refer to the *Platform Guide* sections “About the licensing server” and “About the local license database”.

Backup Stations step

Note: *Use of the Backup Stations step is recommended when configuring a job prototype component, versus when running a provisioning job from the ProvisioningNwExt (or invoking the “Start Backup” action on that extension). For related details, see “About the Niagara Network Job Prototype” on page 3-17.*

Among [Niagara provisioning job steps](#), the Backup Stations step (ProvisioningBackupStep) makes an online backup for each specified station (providing station is running), or if a specified station is not running, an offline backup. It is available in the **New Job Step** menu when adding a step in either the [Niagara Network Prototype View](#) (preferred) or the [Niagara Network Job Builder](#). Also, it is automatically added when you invoke the [ProvisioningNwExt action Start Backup](#) (for every station).

In any case, the backup dist file for each station is stored on the Supervisor, using the following convention:

```
^provisioningNiagara/stationData/stationName/backups/  
  backup_stationName_timestamp.dist
```

where

- *stationName* is the name of the station processed in the step
- *timestamp* is the job's start time, formatted as YYYYMMDD_HHmmss.sss

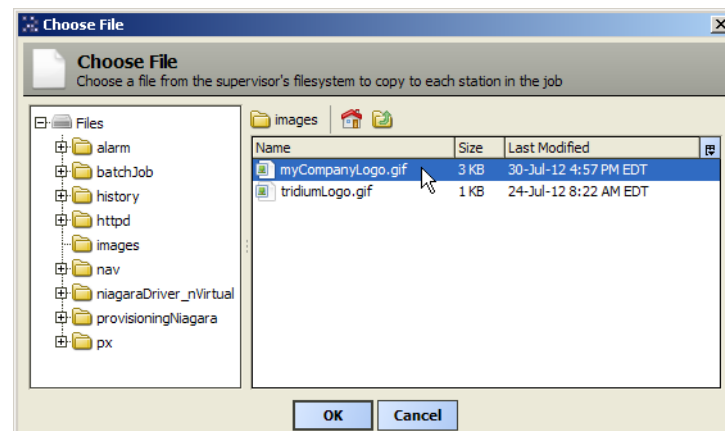
Note: *Starting in AX-3.7, “super user” permissions are required to see the provisioningNiagara subfolder on the Supervisor's station.*

Copy Supervisor File step

Among [Niagara provisioning job steps](#), the Copy Supervisor File step copies a single file from the Supervisor to a given location on the target JACE. It is available in the **New Job Step** menu when adding a step in either the [Niagara Network Job Builder](#) or [Niagara Network Prototype View](#), and also in the provisioningNiagara palette (as “CopyFile” step).

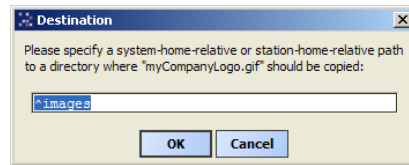
When you add this step, the standard **File Chooser** dialog appears, showing folders and files under the Supervisor's (currently running) station folder—use it to specify the source file on the Supervisor PC. [Figure 3-38](#) shows an example of a .gif file being selected.

Figure 3-38 Select Supervisor source file using File Chooser



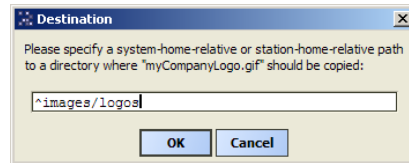
After selecting a local source file, a **Destination** dialog prompts you for the target destination to copy this file to, as shown in [Figure 3-39](#).

Figure 3-39 Select target Destination



Note that this destination folder applies to *all stations* in the job. You may need to edit destination, as shown done in [Figure 3-40](#).

Figure 3-40 Edited target Destination



Here, the destination was changed to specify a “logos” subfolder under images folder, all under the “station root” absolute (^). If a destination folder does not already exist on the target host, it is created.

Note: *The destination string must always begin with the character for either the system-home relative (!) or the station-home relative (^). No means is provided to modify any files outside of the Niagara release directory on any target JACE hosts.*

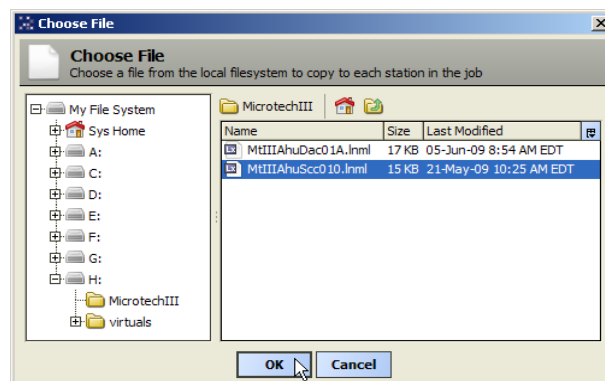
Upon execution, file copy steps are combined with other file copy steps, install software steps, or upgrade out-of-date software steps in order to run more efficiently.

Copy Local File step

Among [Niagara provisioning job steps](#), the Copy Local File step copies a single file from your (local) Workbench PC to a given location on the target JACE. It is available when adding a step in the [Niagara Network Job Builder](#) (**ProvisioningNwExt** view) only. It is also available in the provisioningNiagara palette (as “CopyFile”), but only if copied in the [Niagara Network Job Builder](#) view (and then selected instead of the “Copy Supervisor File” step in an intermediate popup dialog).

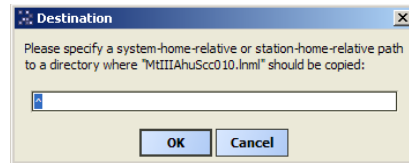
When you select this step, the standard **File Chooser** dialog appears, showing all drives mapped on your local PC—use it to navigate to the source file to copy. [Figure 3-38](#) shows an example of an .lnml file being selected.

Figure 3-41 Select local source file using File Chooser



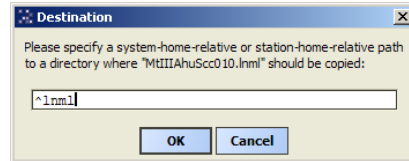
After selecting a local source file, a **Destination** dialog prompts you for the target destination to copy this file to, as shown in [Figure 3-39](#).

Figure 3-42 Select target Destination



Note that this destination folder applies to *all stations* in the job. You may need to edit destination, as shown done in Figure 3-40.

Figure 3-43 Edited target Destination



Here, the destination was changed to specify an “lnml” folder under the “station root” absolute (^). If a destination folder does not already exist on the target host, it is created.

Note: *The destination string must always begin with the character for either the system-home relative (!) or the station-home relative (^). No means is provided to modify any files outside of the Niagara release directory on any target JACE hosts.*

When the job is scheduled or run, unless the file being copied is local to the Supervisor (i.e. you are using Workbench on the Supervisor), a temporary copy of the file is made there. This temporary file is cleaned up once the job completes.

Upon execution, file copy steps are combined with other file copy steps, install software steps, or upgrade out-of-date software steps in order to run more efficiently.

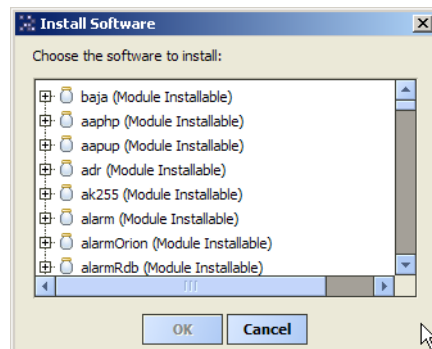
Install Software step

Among [Niagara provisioning job steps](#), the Install Software step (InstallBySpecStep) installs a *versioned* software file to the target JACE host(s), from the Supervisor's software database (files under its !sw folder). It is available in the **New Job Step** menu when adding a step in either the [Niagara Network Job Builder](#) or [Niagara Network Prototype View](#).

Note: *If using a remote Workbench PC to connect to the Supervisor station, and you copy/drag a (local) software file into the jobs step list pane from the Nav tree, the service automatically checks to see if that software file/version already exists on the Supervisor. If not, it is immediately downloaded to (and registered with) the Supervisor, in the background.*

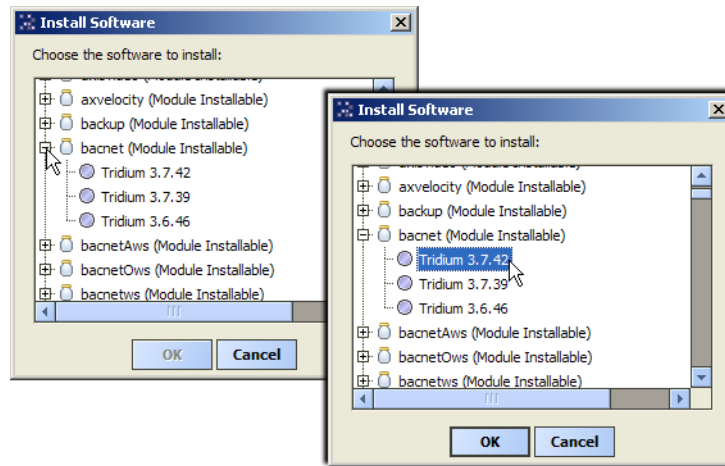
If adding this step in the job creation view using the ➕ (add) button, an **Install Software** dialog appears (Figure 3-44), in which you select the software module or .dist file to install.

Figure 3-44 Install Software dialog



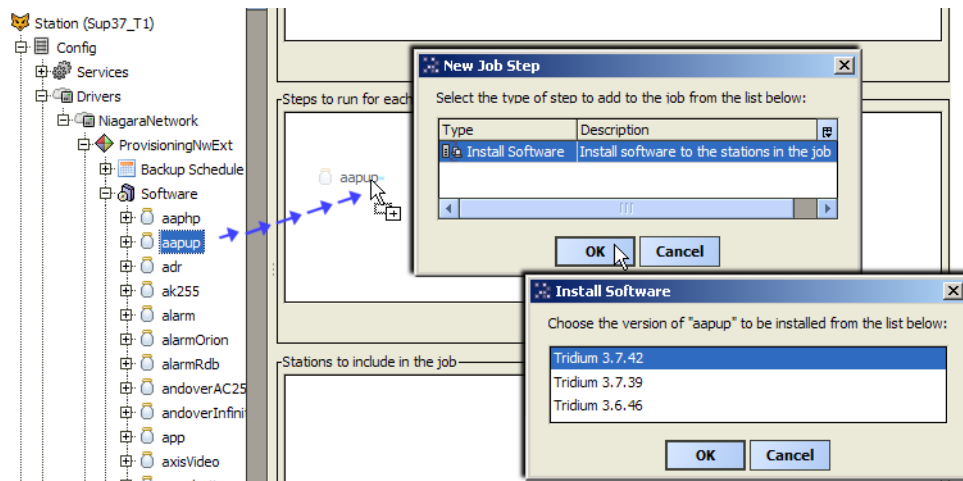
Select an item by expanding to see its *version*, then click that software file, then **OK** (Figure 3-45).

Figure 3-45 Expanding item to select software file by version



Note that when you drag a software item from the service's Software container, and more than one version (file) for that item is in the Supervisor's software database, a popup dialog prompts you to select the version (Figure 3-46).

Figure 3-46 Select dialog from dragging item with multiple versions



Once added, the Install Software step appears in the job steps list pane. If the software has dependencies on module that are not yet installed on a particular JACE host, and those modules are in the Supervisor's software registry, those modules will be automatically included in the processing of that host (station), in order to satisfy dependencies.

Note: It is your (provisioning user's) responsibility to ensure that platform dependencies of the software are met by the JACE hosts running the target stations. For example, it is permissible to have a job with an install software step that includes stations running on different platform types, say JACE-545s and JACE-403s. However, if a step installs a distribution file specific to a JACE-403, note that the dependency check will fail on the JACE-545s, and no software will be installed on them.

When the job executes, install software steps are combined with other software install steps, copy file steps, and upgrade out-of-date-software steps to run more efficiently.

Note: A slightly different step (*InstallStep*) is created when you copy/drag a "backup .dist" file into the Job Steps List pane—a backup .dist is not a "versioned" install (nor is it a *FileCopyStep*).

Reboot step

Among [Niagara provisioning job steps](#), the Reboot step (*RebootJobStep*) reboots each station's host, then waits until its Niagara platform daemon comes back up and is available for connections again. It is available in the **New Job Step** menu when adding a step in either the [Niagara Network Job Builder](#) or [Niagara Network Prototype View](#).

Usage is expected to be infrequent, perhaps as a temporary measure for some misbehaving third-party software module. This job step will fail if a station's Software device extension is disabled or in fault.

Run Robot step

Among [Niagara provisioning job steps](#), the Run Robot step (RunRobotStep) causes the specified ProvisioningRobot (program) in the Supervisor station to be executed by each remote station in the job, by each station's ProgramService. Find it in the **New Job Step** menu when adding a step in either the [Niagara Network Job Builder](#) or [Niagara Network Prototype View](#).

Usage requires the Supervisor station to have a ProvisioningRobot customized to perform a task (or tasks) that specifically apply to all stations included in the provisioning job. This job step will fail if a target station is not running, does not have the ProgramService, or selectively in other cases were faulty or inapplicable program code is included in the specified ProvisioningRobot.

For more details, see “[Provisioning Robot notes](#)” on page 3-30.

Upgrade Out-of-date Software step

Among [Niagara provisioning job steps](#), the Upgrade Out-of-date Software step (UpgradeOutOfDateStep) compares the versions of software installed on the station's host with the latest versions of the same software in the Supervisor's software database. Any software found with a higher version on the Supervisor is then installed to the station. Find it in the **New Job Step** menu when adding a step in either the [Niagara Network Job Builder](#) or [Niagara Network Prototype View](#).

Note: *Although not typical, please note that the latest version of any software module found under the Supervisor's software database (under !sw) is always installed by this step, even if the Supervisor itself is using an earlier installed version (as found in its !modules directory). Normally, the Supervisor has the latest versions of software modules installed, so this distinction is moot.*

When the job executes, upgrade out-of-date-software steps are combined with other software install steps and copy file steps, to run more efficiently.

Provisioning Robot notes

Note: *By default starting in AX-3.7, you must have “super user” permissions to add/edit Program and Robot components (including ProvisioningRobots). If necessary, this is changeable in the Supervisor's system.properties file, via the niagara.program.requiresSuperUser entry.*

Note super user permissions are not required to configure a provisioning job that includes a “Run Robot step”, where the referenced ProvisioningRobot was previously added/edited by a super user. However, the station user in any remote JACE station used by the Supervisor for client access (fox) must be a super user for such a provisioning job to run successfully on that station.

This section provides various notes about the ProvisioningRobot in the provisioningNiagara palette. To use a ProvisioningRobot, copy (drag) it from the provisioningNiagara palette into the Supervisor station (wherever needed), and modify its program code as appropriate.

You can then reference the ProvisioningRobot when adding a “[Run Robot step](#)” when building a provisioning job. Or when building a job, simply drag a station's ProvisioningRobot into the step list area (in the [Niagara Network Job Builder](#) or [Niagara Network Prototype View](#))—this adds the Run Robot step.

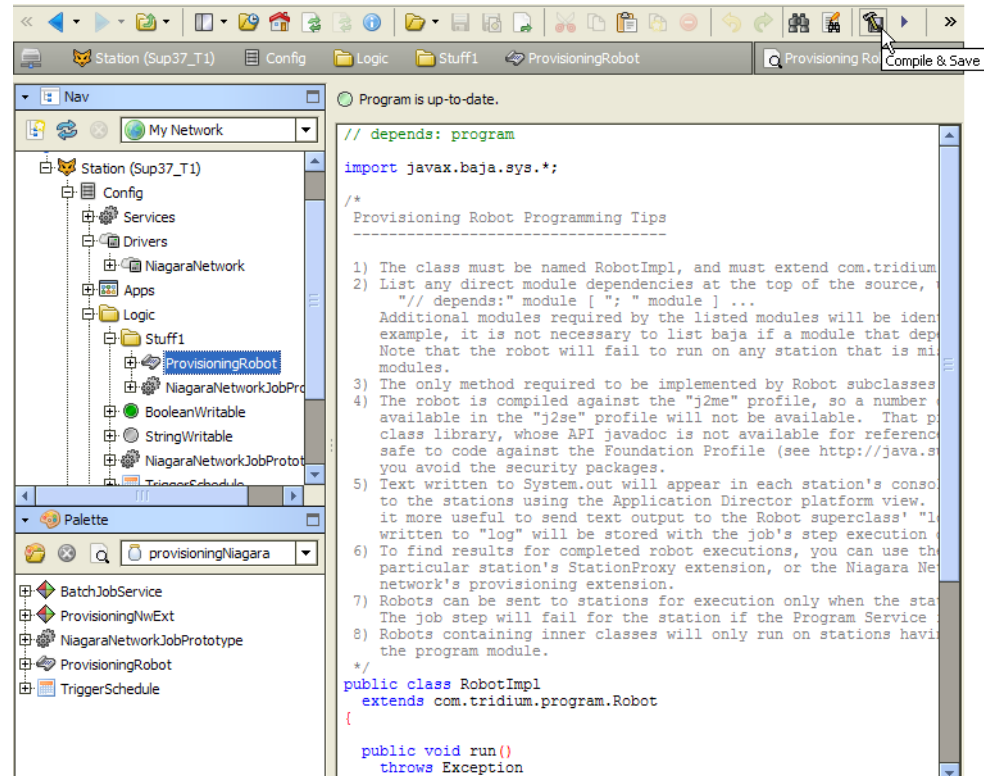
More details are in the following subsections:

- [Provisioning Robot Editor](#)
- [Run Robot step view features](#)

Provisioning Robot Editor

The default and main view of a ProvisioningRobot is the Provisioning Robot Editor ([Figure 3-47](#)), where you edit and compile the Baja program code to be executed by all target stations.

Figure 3-47 Provisioning Robot Editor is default view for a ProvisioningRobot



This view operates like the “Robot Editor” view of a station’s ProgramService. To successfully customize and use ProvisioningRobots, you should already be familiar with the NiagaraAX (Baja) class structures and methods, as well as the Java syntax used by Baja program code.


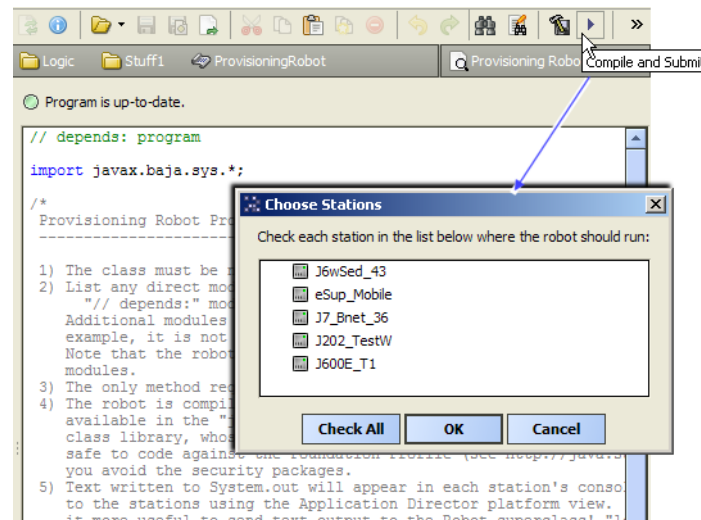
The Provisioning Robot Editor tool bar has buttons for Compile and Save  and Run Now  (Compile and Submit). As shown in Figure 3-48, clicking Run Now produces the “Choose Stations” dialog.

Figure 3-48 Run Now produces Choose Stations dialog, then switches to Niagara Network Job Builder view



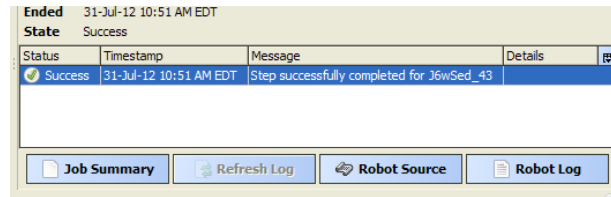
After you select one or more stations, the **Niagara Network Job Builder** view appears, with the Run Robot step and selected stations entered, ready for you to launch (or if needed, add other steps).

Note: As a simple test, the ProvisioningRobot “as copied” from the provisioningNiagara palette, if specified in a “Run Robot” step (or commanded to “Run Now”) should execute without errors by any target station with the ProgramService. In addition, the program code in this ProvisioningRobot includes a number of helpful programming tips within the remark lines near the top of the code.

Run Robot step view features

Step views for a “Run Robot step” provides additional buttons near the bottom of the view (Figure 3-49).

Figure 3-49 Run Robot step in Batch Job Step Log File View



Each of these buttons produces a popup dialog:

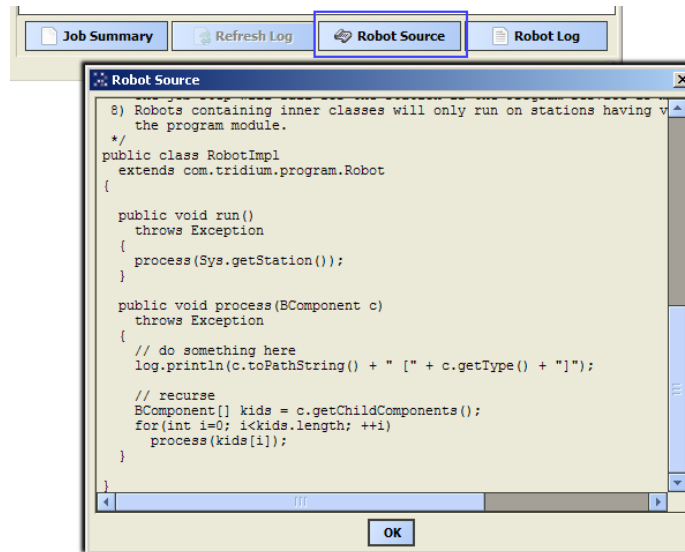
- [Robot Source](#)
- [Robot Log](#)

For additional details about step views, see “Batch Job Step Log File View” on page 3-9.

Robot Source

Click **Robot Source** for read-only display of the ProvisioningRobot code used in the “Run Robot step”, as shown in Figure 3-50.

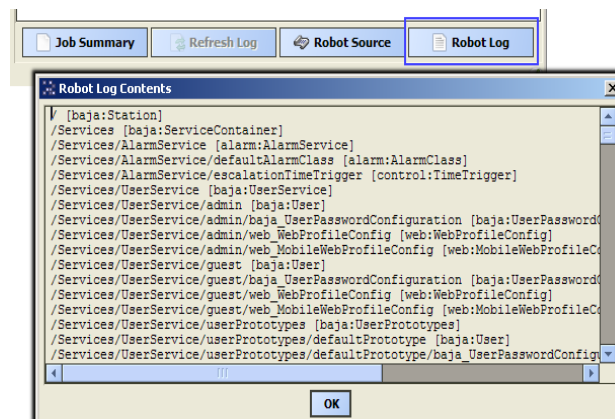
Figure 3-50 Robot Source popup showing program code used in referenced ProvisioningRobot



Robot Log

Click **Robot Log** to see any text written to the robot’s “log” PrintWriter during the step’s execution, as shown in Figure 3-51.

Figure 3-51 Robot Log Contents popup showing log output from executing the ProvisioningRobot



Niagara Provisioning Job management

Niagara Provisioning uses *batch jobs* as the method for doing most of its tasks. A combination of objects, including components, files, and histories are used to model provisioning jobs, and their contained steps.

Note that batch jobs are different than other jobs run by the Supervisor's JobService, because:

- File records of a batch job execution are retained until the job is explicitly disposed of, instead of being held temporarily as children of the JobService, then quickly disappearing.
- Batch jobs can optionally trigger alarms (alerts) upon successful or unsuccessful completion.
- Batch jobs are first dispatched to a thread pool job queue, before being given to the JobService.
- Histories are automatically created for provisioning batch jobs as well as for individual device steps.

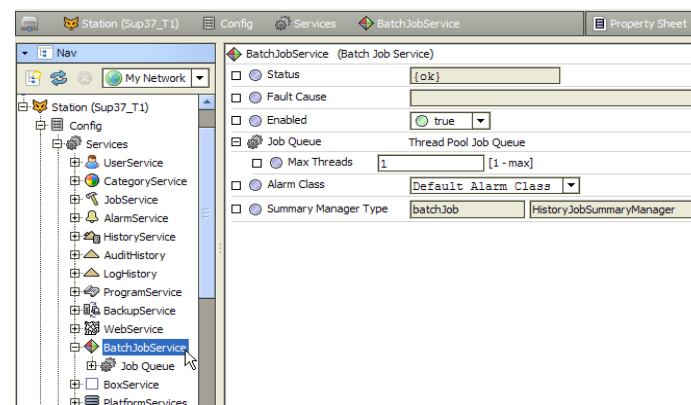
These sections describe those objects and their relationship to each other.

- [BatchJobService](#)
- [Job execution](#)
- [Batch Job log files](#)
- [Provisioning Niagara station data files](#)
- [Histories related to provisioning](#)
- [About provisioning-related alarms](#)

BatchJobService

The BatchJobService is required in the Supervisor station to facilitate provisioning. Copy it from the batchJob or provisioningNiagara palette into the Supervisor's Services folder. As shown in [Figure 3-52](#), it has no special views.

Figure 3-52 BatchJobService in Supervisor has no special views, and only a few properties



The BatchJobService is responsible for dispatching all provisioning jobs (as batch jobs) on a Supervisor, where batch jobs are run by the station's JobService. The BatchJobService provides “job control” functions for provisioning jobs. However, you do not normally interface with it, apart from specifying the station's alarm class to use for provisioning alarms. See the next section “[BatchJobService properties](#)”.

BatchJobService properties

Properties and slots of the BatchJobService ([Figure 3-52](#)) are described as follows:

- **Status** — Is either *fault* if provisioning is not licensed, *disabled* if Enabled property is false, and in all other cases is *ok*.
- **Enabled** — If set to false, provisioning activity via the BatchJobService cannot occur.
- **Fault Cause** — Text explanation for why service is in fault.
- **Job Queue** — (Thread Pool Job Queue) Manages the submission of provisioning batch jobs by using a thread pool, to ensure the Supervisor's CPU and network resources are not overtaxed by concurrent sessions. A single property, **Max Threads**, defaults to one (1), meaning only one provisioning job can run at a time. No special views or other features are provided.
Note: Only after determining the station has available resource overhead, should Max Threads be adjusted to values over 2 or 3. Otherwise, other tasks performed by the station may be affected.
- **Alarm Class** — (Recommended to be *changed* from default) The Alarm class in the Supervisor station's AlarmService to be used for alarms (technically, *alerts*) when provisioning batch jobs fail and/or complete—as set in the alarm checkboxes of the view used to build the provisioning job. For example (in the [Niagara Network Job Builder](#)), see “[Alarm checkboxes](#)” on page 3-5. For further details, see “[About provisioning-related alarms](#)” on page 3-35.

Job execution

Just prior to execution, some steps in a provisioning batch job are combined. Currently, the only steps that are combined are “Install Software,” “Copy File,” and “Upgrade Out-of-date Software” steps that are adjacent to each other. Combining steps avoids duplication of dependency-checking with a station, and minimizes the number of reboots required.

Upon execution, the provisioning batch job first executes any “initial, steps to run only once”—in NiagaraNetwork provisioning this simply means the “Update Licenses” step (if included). Typically the Supervisor has Internet connectivity, and makes a single, silent inquiry to the licensing server, passing the current licensing information for each host running an included station (in the job). The licensing server responds with updated licensing information (if any) for these hosts in the form of a license archive, where any updated licenses are installed—as well as updated within the Supervisor’s “local license” database. For more background details, refer to related *Platform Guide* sections “About the licensing server” and “About the local license database”.

Following this, the provisioning batch job executes remaining “steps to run for each station,” working through its list of stations in sequence, and for each station it executes its steps in sequence. When the job reaches a station in the list, its station state becomes “Running.” If any of the steps fail, the station’s state becomes “Failed,” and no additional steps are run for that station, and the job continues with the next station in the list.

If every step succeeds for a station, its station state becomes “Success.” If the job is canceled during a station step, its station state, the station state of all following stations, and the job state all become “Canceled.”

When all steps are complete for all stations without canceling, and all steps completed successfully, the job state is “Success.” However, if even one step failed, the job state will be “Failed.”

Batch Job log files

Batch job log files are binary records of a single completed provisioning batch jobs, as well as single steps in completed provisioning batch jobs. Batch job log files and batch step log files are always written under the Supervisor’s station directory using the following convention:

```
^batchJob/logs/batchJob_type/timestamp.bjl (or .bjsl)
```



Caution

Do not manually delete any batch job log files on the Supervisor, e.g. using Windows Explorer, or in Workbench using the “My File System” node in the Nav tree. Instead, use the “**Dispose**” job feature in various provisioning views, for example the [Niagara Network Job List](#), or [Prototype Job List](#) view for Niagara provisioning job components. When you dispose a job, all of its related batch job log files (and [Provisioning Niagara station data files](#)) are automatically deleted.

Provisioning Niagara station data files

Note: Starting in AX-3.7, a station user requires admin-level Read (R) permissions on a category assigned to the provisioningNiagara subfolder in order to see it under the station’s Files node.

In addition to batch job log files, Niagara provisioning creates other files under the Supervisor’s station directory to store backup .dist files, as well as “snapshots” of the software installed on the hosts running those stations.

Backup .dist files are written under the Supervisor’s station directory using the following convention:

```
^provisioningNiagara/stationData/station/backups/backup_station_timestamp.dist
```

Software snapshot files are written under the Supervisor’s station directory using this convention:

```
^provisioningNiagara/stationData/station/software/snapshot.bog
```



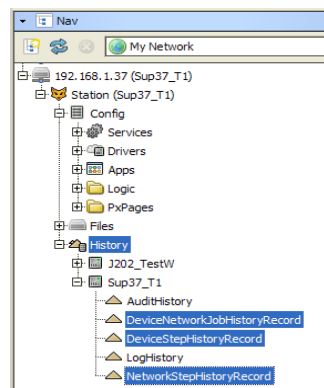
Caution

Do not manually delete any stationData files on the Supervisor, e.g. using Windows Explorer, or in Workbench using the “My File System” node in the Nav tree. Instead, use the “**Dispose**” job feature in various provisioning views, for example the [Niagara Network Job List](#), or [Prototype Job List](#) view for Niagara provisioning job components. When you dispose a job, all of its related stationData files (and [Batch Job log files](#)) are automatically deleted.

Histories related to provisioning

Niagara provisioning jobs are automatically recorded within the history space of the Supervisor, as shown in histories highlighted in [Figure 3-53](#).

Figure 3-53 Provisioning related histories in Supervisor's history space



These histories are automatically created using the following names:

- **DeviceNetworkHistoryRecord**
Contains one record for each Niagara provisioning job attempted. Includes various fields showing the job's finished state, submitting user, stations to process, and whether it has been disposed.
- **DeviceStepHistoryRecord**
Contains one record for each "run for each station" step, per station, for any Niagara provisioning job. Includes various available fields, showing the step's type, finished state, description, station, and whether disposed (included in a job that was disposed).
- **NetworkStepHistoryRecord**
Contains one record for each initial "steps to run only once" step (that is "Update Licenses" for any Niagara provisioning job). Includes various available fields, showing the step's type, finished state, description, network type, and stations to process.



Caution Do not rename, delete, or clear these histories on the Supervisor, for example using the "Database Maintenance" view on the Supervisor's history space. These histories are used by the station's BatchJob API code, and by various "job list" views.

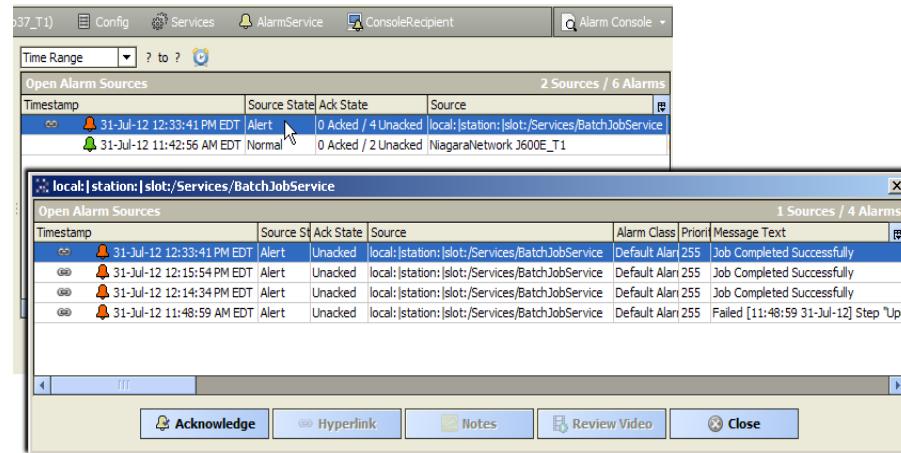
Deleting these histories can result in provisioning-related files becoming "orphaned," such that it becomes impossible to know if they can safely be deleted. Therefore, examination of these histories is optional, and in most cases you can simply ignore them.

About provisioning-related alarms

For any Niagara provisioning job, you can configure it to generate an alarm either upon job failure, successful job completion, or both. For details, see "Alarm checkboxes" on page 3-5 (Niagara Network Job Builder) and "Alarm checkboxes" on page 3-23 (Niagara Network Prototype View).

Alarm routing uses the alarm class specified in the Supervisor's [BatchJobService properties](#), and any alarm appears as an *Alert* 📢 (source state) in the Alarm Console, showing the BatchJobService ord as source (local: | station: | slot: /Services/BatchJobService). See [Figure 3-54](#).

Figure 3-54 Niagara Provisioning alarm is an “alert” with source as BatchJobService



Note: The single alarm source of the BatchJobService applies to all Niagara provisioning related alarms (alerts), meaning only one row is used in the Alarm Console for all provisioning alerts. Keep in mind if there are multiple alerts, and you click **Acknowledge** on that row, all alerts will be acknowledged.

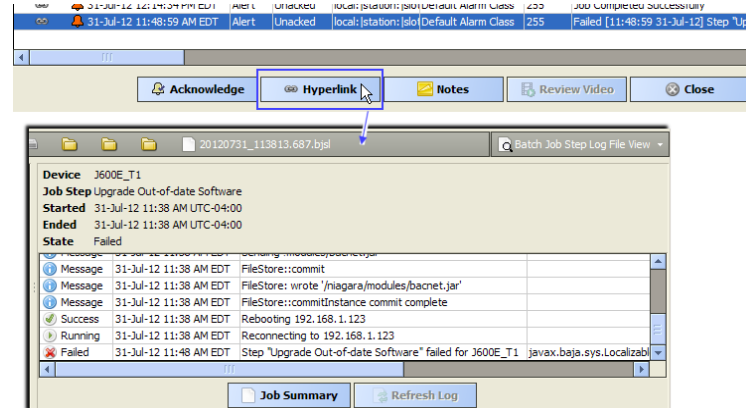
As needed, double-click that row to see multiple provisioning alerts in the Alarm Details view for the BatchJobService, as shown done in Figure 3-54.

Note that as shown, each Niagara provisioning alert includes a *hyperlink* for more details. When you click the **Hyperlink** button with the alert highlighted, the view changes to either:

- **Batch Job Log File View** — For that job, in cases where the alert is raised for *successful* job completion. For more details on this view, see “Batch Job Log File View” on page 3-9.
- **Batch Job Step Log File View** — For a *failed* job step, in cases where the alert is raised for a job failure. For more details on this view, see “Batch Job Step Log File View” on page 3-9.

Figure 3-55 shows an example hyperlink to the step log file view for a failed Niagara provisioning job.

Figure 3-55 Batch Job Step Log File View from Hyperlink in Alarm Console



Note: If you disposed a Niagara provisioning job before acknowledging it from the Alarm Console, if you click the **Hyperlink** button to it from an associated Alarm Details view, you will see a “Cannot Display Page” error in the view. This happens because disposing a job removes the batch job log (.bjsl) file and all batch job step log (.bjsl) files associated with that job. Therefore, it is recommended that you acknowledge any related alarms (alerts) before disposing jobs.

CHAPTER 4

Provisioning Station Extension Concepts

This section explains components that are automatically created by the ProvisioningNwExt under a different area of the Supervisor station (*apart* from the station's **ProvisioningNwExt** under its **NiagaraNetwork**, or separately copied “**job prototype**” components from the provisioningNiagara palette). These components are special “device extensions” created under *each* of the NiagaraStation devices contained in its **NiagaraNetwork**. Some have special views, also described in this section, as well as a “Provisioning Manager” view that is available on the NiagaraNetwork.

Along with the ProvisioningNwExt and its children (see “[About the ProvisioningNwExt](#)” on page 3-2) all station extension components are required to be present (and enabled) for full provisioning support.

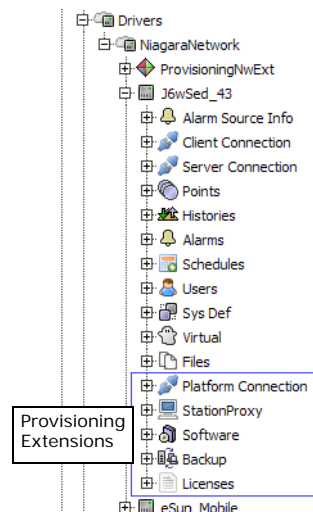
These are the main subsections:

- “[Types of provisioning extensions](#)” on page 4-1
- “[Provisioning Manager](#)” on page 4-2 (view on the NiagaraNetwork)
- “[Platform Connection](#)” on page 4-3 (extension)
- “[StationProxy](#)” on page 4-5 (extension), and its views “[Provisioning Station Director](#)” on page 4-7 and “[Station Job List](#)” on page 4-8
- “[Software](#)” on page 4-9 (extension), and its “[Station Software View](#)” on page 4-10
- “[Backup](#)” on page 4-11 (extension)
- “[Licenses](#)” on page 4-12 (extension)

Types of provisioning extensions

When you add the ProvisioningNwExt under the Supervisor’s NiagaraNetwork, *five* different device extensions are automatically created under each NiagaraStation component, as shown in [Figure 4-1](#).

Figure 4-1 Provisioning extensions added to NiagaraStation device



These device extensions are *in addition to* the standard extensions (**Points**, **Histories**, **Alarms**, **Schedules**, **Users**, **Sys Def**, **Files**) that exist for any NiagaraStation device.

Note: These same extensions are also automatically included when you add a new station under the *NiagaraNetwork* in the Supervisor (providing the network has the ProvisioningNwExt).

The five provisioning extension types are:

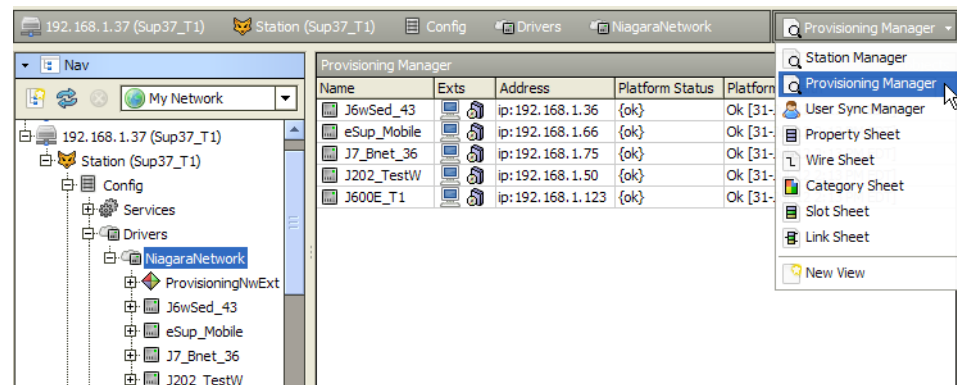
- [Platform Connection](#) — manages the platform connection between the Supervisor and the remote JACE running the station.
- [StationProxy](#) — polls the station for system statistics.
- [Software](#) — holds a “snapshot” of the current software versions installed on the JACE host.
- [Backup](#) — enables the Supervisor to make backups of the station.
- [Licenses](#) — enables the Supervisor to update licenses on the JACE running the station.

Although automatically created, you must (minimally) configure some properties in the [Platform Connection](#) provisioning extension.

Provisioning Manager

When you add the ProvisioningNwExt under a Supervisor’s NiagaraNetwork, in addition to the five [provisioning extensions](#) added to each child NiagaraStation, a new table-based view becomes available on the network: the **Provisioning Manager**, as shown in [Figure 4-2](#).

Figure 4-2 Provisioning Manager view of the Supervisor’s NiagaraNetwork



This view provides a “network-at-a-glance” summary of provisioning-related items, where each row represents a NiagaraStation component (similar to the network’s default **Station Manager** view).

Included in each row is the status and health of each NiagaraStation’s “[Platform Connection](#)” (provisioning extension), updated by the ongoing “ping” monitor to its platform daemon. For any station (row), you can also right-click it and manually issue a “Ping” action.

See the following sections for further details:

- [Extension Access in Provisioning Manager](#)
- [Edit in Provisioning Manager](#)

Extension Access in Provisioning Manager

In the **Exts** column, for each NiagaraStation, the [Provisioning Manager](#) provides double-click access to the following provisioning device extensions:

StationProxy extension, with a default **Provisioning Station Director** view.

For more details, see “[StationProxy](#)” on page 4-5 and “[Provisioning Station Director](#)” on page 4-7.

Software extension, with a default **Station Software Manager** view.

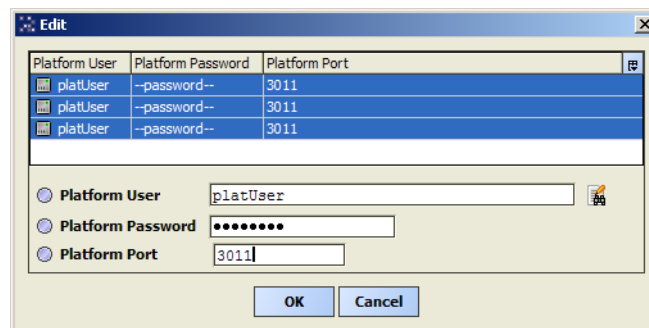
For more details, see “[Software](#)” on page 4-9 and “[Station Software View](#)” on page 4-10.

Other types of [provisioning extensions](#) are accessible by expanding a NiagaraStation in the Nav tree, or from the property sheet of a NiagaraStation.

Edit in Provisioning Manager


The [Provisioning Manager](#) provides an **Edit** button at the bottom of the view, enabled whenever you have one more rows selected. Edit allows you to modify the platform connection credentials used for provisioning access by the Supervisor station—otherwise accessible in each NiagaraStation’s “Platform Connection” (device extension) property sheet.

Figure 4-3 Edit dialog from Provisioning Manager



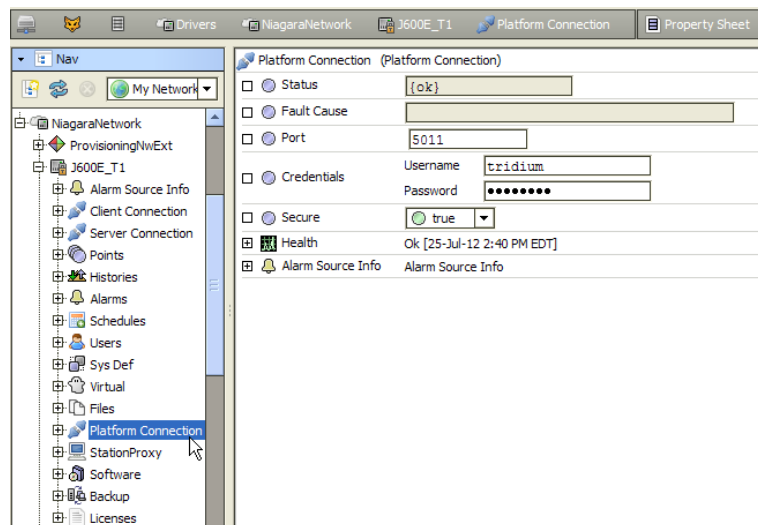
This Edit feature can be useful when first configuring Niagara provisioning on a NiagaraNetwork, especially if multiple (if not all) remote JACE hosts are using the *same* platform credentials. In this case, a single “gang edit” of these selected rows provides the configuration needed for Supervisor access.

Platform Connection

The  Platform Connection component, among the [provisioning extensions](#), specifies the HTTP port and credentials for the Supervisor to use for making a platform connection to each host represented by a NiagaraStation. Platform connections are used in most provisioning jobs, and in the polling performed by the ProvisioningNwExt. Starting in AX-3.7, you also specify if a *secure* platform connection (platform SSL) should be used.

In order for provisioning to work for any station, you must specify both its platform *port* and platform *credentials* in each newly-created Platform Connection extension. You can do this in its property sheet, as shown in [Figure 4-4](#) (no “special” views exist for this extension).

Figure 4-4 Platform Connection property sheet (AX-3.7 station shown)



The following sections provide more details:

- [Properties](#)
- [Ping action](#)
- [New station notes](#)

Properties

Properties of a [Platform Connection](#) provisioning extension are described as follows:

- Status — Current status of extension, where value is ok, disabled, or fault. The platform connection will be in *fault* if any of the following occurs:
 - Supervisor has no ProvisioningNwExt under its NiagaraNetwork (e.g., it has been deleted).
 - Supervisor is not licensed for provisioning.
 - NiagaraStation is in fault.
 - The station’s platform daemon rejects the platform connection’s credentials.

The extension will be disabled if the ProvisioningNwExt is disabled.

- **Fault Cause** — If in fault, gives brief text explanation.
- **Port** — Port on which the platform daemon in the station's host (JACE) is listening, where the default port for "regular" (non-SSL) platform daemon is 3011. (Shows as "Platform Port" in the **Add** or **Edit** dialog when working in the **Station Manager** view, see "New station notes" on page 4-4).
If this port was changed in commissioning, you must match it here. Or, if a AX-3.7 or later host configured for platform SSL access (Secure platform), change to match *that* port, where the default secure platform port is 5011 (also subject to change in commissioning).
- **Credentials** — Credentials used for a platform connection to the JACE running the station, where you enter a **Username** and **Password**.
(Credentials show as "Platform User" and "Platform Password" in the **Add** or **Edit** dialog when working in the **Station Manager** view, see "New station notes" on page 4-4).
- **Secure** — (AX-3.7 and later) Boolean property to specify if a secure (platform SSL) connection should be used to the host (JACE), where the default is false. (Shows as "Secure Platform" in the **Add** or **Edit** dialog in the **Station Manager** view, see "New station notes" on page 4-4).
If set to true, the Port property must be changed from its default value—typically to port 5011 ("standard") or whatever port was specified in the host's (JACE's) commissioning.
- **Health** — Contains information about the success or failure of the last pings, and is similar to the standard "Health" slot in most driver networks.
- **Alarm Source Info** — Specifies how and if alarms are to be generated as a result of ping monitor failures, similar to the standard "Alarm Source Info" slot in most driver networks.

Ping action

A single "Ping" action is available on the [Platform Connection](#), to immediately force a short message to the host's platform daemon. Its "Health" property updates with ping results. You can issue this action after entering port and credentials properties, to test their validity.

New station notes

As shown in [Figure 4-5](#), when working in Supervisor's NiagaraNetwork to add a *new station*, the **Add** dialog automatically at least three provisioning-related properties, providing that the Supervisor is configured with the ProvisioningNwExt.

Figure 4-5 Add dialog for new NiagaraStation (in NiagaraNetwork)

Name	Address	Fox Port	Use Foxs	Username	Password	Enabled	Virtuals Enabled	Platform User	Platform Password	Secure Platform
J600E_T1	ip:192.168.1.123	4911	true		--password--	true	false		--password--	false

Name J600E_T1

Address IP 192.168.1.123

Fox Port 4911

Use Foxs true

Username

Password

Enabled true

Virtuals Enabled false

Platform User

Platform Password

Secure Platform false

Platform Port 3011

OK Cancel

These properties are at the bottom of the **Add** (or **Edit**) dialog for a NiagaraStation, all with "Platform" in the descriptor. They are for the Port and Credentials properties in the [Platform Connection](#) extension, and if a AX-3.7 or later Supervisor, a "Secure Platform" property to specify if a "platform SSL" connection should be used. These properties do not appear in these dialogs when working in the NiagaraNetwork of any JACE station, or if a Supervisor is not configured with (or licensed for) the ProvisioningNwExt.

If you want to perform provisioning on the station, be sure to put proper values in these platform fields. For more details on Platform Connection properties, see "Properties" on page 4-3.

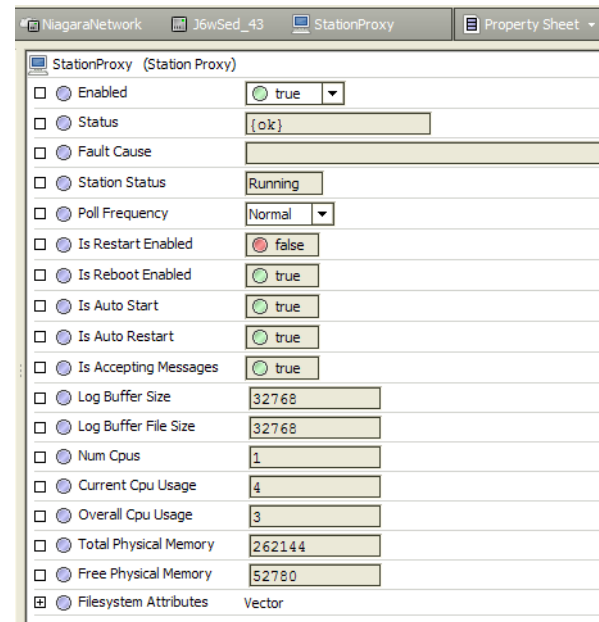
Note: A NiagaraNetwork "discover" can determine if a station is configured for secure (Foxs) access, including the port used—for example the standard 4911 as shown in [Figure 4-5](#) above. However, a discover cannot determine if that station's host has "secure platform" access enabled, nor its associated port.

StationProxy

The StationProxy extension (StationProxy component), among the [provisioning extensions](#), provides platform administration functions like those available when you open a direct platform connection in Workbench, using the **Station Director** and **Platform Administration** views. It also provides a number of actions for station control functions.

By default, this extension is enabled, to allow polling from the ProvisioningNwExt for values of the extension's properties. [Figure 4-6](#) shows the property sheet for the station's StationProxy extension. Note that other special views are provided, including the default view ([Provisioning Station Director](#)) as well as a [Station Job List](#).

Figure 4-6 StationProxy property sheet



The following sections provide more details:

- [Properties](#)
- [Actions](#)
- [Views](#)

Properties

Properties of the [StationProxy](#) are described as follows:

- **Enabled** — Defaults to `true`, set to `false` if you wish to prevent polling by the ProvisioningNwExt. Note that Enabled must be `true` in order to use the special views on the Station Proxy extension, namely the [Provisioning Station Director](#) and the [Station Job List](#).
- **Status** — Current status of extension, where value is `ok`, `disabled`, or `fault`. The platform connection will be in `fault` if any of the following occurs:
 - Supervisor has no ProvisioningNwExt under its NiagaraNetwork (e.g., it has been deleted).
 - Supervisor is not licensed for provisioning.
 - NiagaraStation is in fault.
 - Host's platform daemon rejects the platform connection's credentials.
 The extension will be `disabled` if its **Enabled** property is set to `false`, or if the ProvisioningNwExt is disabled.
- **Fault Cause** — Provides text explanation of why status is fault, otherwise is blank.
- **Station Status** — Reflects one of the following values:
 - **Idle** — Station is not currently running, and can be started without a reboot.
 - **Starting** — Station process is running, but has not completed its startup sequence.
 - **Running** — Station is running.
 - **Stopping** — Station is in process of shutting down, but its process is still alive.
 - **Halted** — Station is not currently running, and the host must be rebooted before it can start.
 - **Unknown** — [StationProxy](#) has status `disabled` or `fault`, and so station status is unknown. Status is also unknown if the station is unreachable, or if a poll has not happened yet.

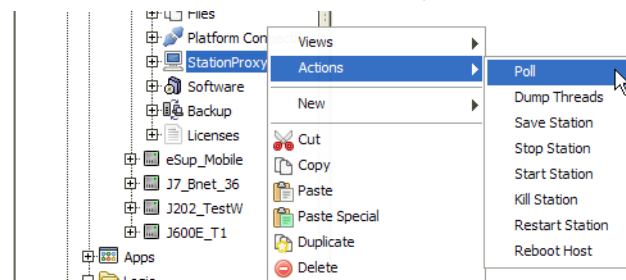
- Poll Frequency — Corresponds to the **Poll Scheduler** in the ProvisioningNwExt, as part of its monitor “ping” mechanism (a ping of the platform daemon in the host running each station). See the section “[ProvisioningNwExt container components](#)” on page 3-12 for related details.
Note: Remaining properties described below are read-only, and reflect various items about the platform.
- Is Restart Enabled — If `true`, the station can be restarted without a reboot of its host platform (such as with Win32-based platforms).
- Is Reboot Enabled — If `true`, the host’s platform daemon is capable of (and allows) rebooting of the host.
- Is Auto Start — If `true`, the station is restarted automatically after the host is rebooted.
- Is Auto Restart — If `true`, the host is restarted (or rebooted if “Is Restart Enabled”=`false`) whenever the station terminates with a failure exit code.
- Is Accepting Messages — If `false` (unlikely), thread dumps, station saves, and graceful shutdown is not possible using the platform daemon.
- Log Buffer Size — Size (in bytes) of the buffer used by the platform daemon to hold the console output.
- Log Buffer File Size — Maximum size of the `console.txt` file (in bytes) that the platform daemon saves console output to, when the station stops.
- Num Cpus — Number of CPUs on the host running the station.
- Current Cpu Usage — Percentage of time the CPU(s) have been in use in the last second.
- Overall Cpu Usage — Percentage of time the CPU(s) have been in use since the platform daemon started.
- Total Physical Memory — Total KB of physical RAM on the station’s host.
- Free Physical Memory — KB of available physical RAM on the station’s host.
- File System Attributes — Free space statistics for each file system on station’s host.

In addition to these properties, the StationProxy extension has a number of [Actions](#).

Actions

As shown in [Figure 4-7](#), each [StationProxy](#) provisioning extension provides a number of actions, available as right-click commands.

Figure 4-7 Action menu for StationProxy extension



Many of these actions are also available in the [Provisioning Station Director](#) view, as well as in views using a direct platform connection. When invoked, each action performs as follows:

- Poll — Causes Supervisor to poll the host’s platform daemon for current data.
- Dump Threads — Supervisor requests that the station send a thread dump to its console output.
- Save Station — Supervisor requests that the station save its current state to its own (local) `config.bog` file.
- Stop Station — Supervisor requests that the station shuts down gracefully.
- Start Station — (applicable only if current station status is idle). Supervisor requests the platform daemon to start the station.
- Kill Station — Supervisor requests for the station to terminate immediately, without graceful shutdown.
- Restart Station — Depending on “Is Restart Enabled” value, causes one of the following:
 - If “Is Restart Enabled” is `false` — Station’s host is rebooted.
 - If “Is Restart Enabled” is `true` — Station is stopped gracefully, then restarted again.
- Reboot Host — Depending on “Is Reboot Enabled” value, causes one of the following:
 - If “Is Reboot Enabled” is `false` — Nothing happens.
 - If “Is Reboot Enabled” is `true` — Supervisor requests for the platform daemon to shut down gracefully, then reboot the host.

Views

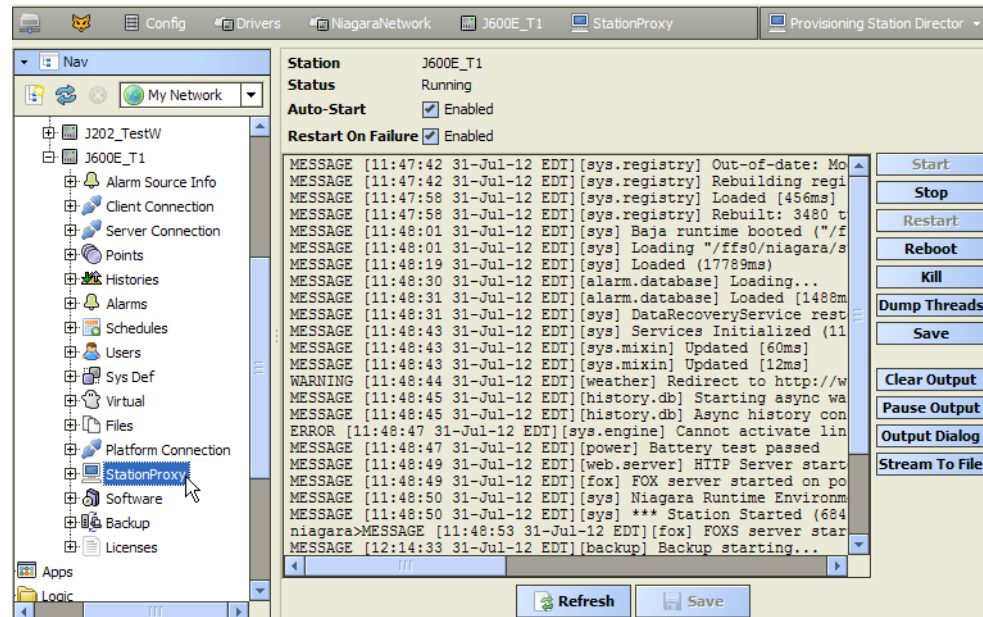
Each **StationProxy** provisioning extension has two special views, summarized as follows:

- **Provisioning Station Director** — The *default* view for the **StationProxy** extension, this view closely resembles the “**Station Director**” view in a direct platform connection.
- **Station Job List** — Provides a list summarizing provisioning job steps that have been executed against this particular station, with additional details available.

Provisioning Station Director

The **Provisioning Station Director** is the default view on the **StationProxy** provisioning extension of a NiagaraStation. As shown in **Figure 4-8**, this view closely resembles the “Application Director” view available in a direct platform connection to a host.

Figure 4-8 Provisioning Station Director is default view on StationProxy extension



Note: Refer to the “Application Director” section in the NiagaraAX Platform Guide for descriptions of most elements in this view. Only elements that differ from that view are explained here.

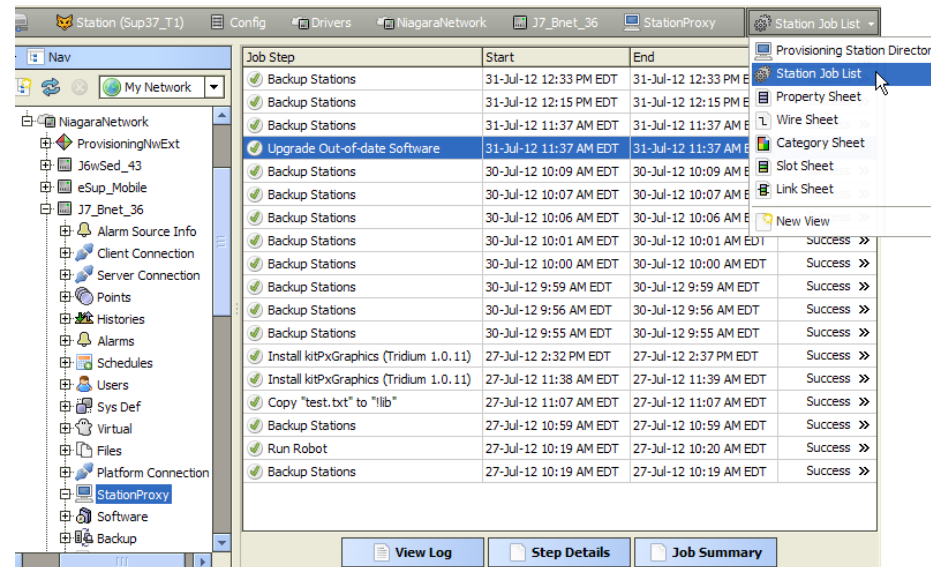
Differences from the **Application Director** view are summarized as follows:

- Since the **Provisioning Station Director** only shows information for *one* station, it does not show the station name and status within a table (at the top of the view), but instead shows this data at the top using simple text labels.
- Where the **Application Director** updates the “Auto-Start” and “Restart on Failure” settings immediately when changed, the Provisioning Station Director works more like a normal view, meaning you must click the **Save** button after making any changes.
- Although the appearances of the two views are similar, their implementations are different. The **Application Director** connects the Workbench view directly to the station’s platform daemon, and is *best for extended troubleshooting*. Whereas, the **Provisioning Station Director** uses the Supervisor station as an intermediary, and as a result is *not as responsive*, and is *less efficient* (uses additional Supervisor resources).

Station Job List

Another view on the [StationProxy](#) extension, access it via the view selector ([Figure 4-9](#)), or by right-clicking the extension and selecting **Views > Station Job List**.

Figure 4-9 Station Job List is available view on StationProxy extension



The main elements in this view are the [Step Table](#), and the [buttons](#) near the bottom.

Step Table

This main area of the [Station Job List](#) view shows a row for each step that has been executed against the station. Note that no record is available for a step's execution unless it has started. For this reason, the following steps *do not appear* in this view:

- steps for jobs not yet started.
- steps for jobs that are running, but are still running prior steps.
- steps that come after any earlier steps (for any station) that were canceled.
- steps that would have executed after another step, but the other step failed for this station.

Note: Because of this, the [Station Job List](#) is not the appropriate view to find the answer for questions like “why did the backup scheduled for Tuesday on this station not run?”. For this type of information, you should look in the [Niagara Network Job List](#) of the [ProvisioningNwExt](#).

The step table includes [columns](#) for various data. You can do any of the following within the table:

- Click the “>>” (Details) button to the right of the status for any step to view its **Step Log File View** (same as using the **Step Details** button at the bottom of the view).
- Double-click any step row to view its **Job Log**, a series of log messages about the step stored in its job log file (same as using the **View Log** button at the bottom of the view).
- Right-click a step for a popup menu, providing same functions as [buttons](#) (at bottom of view).

Step table columns

Step table columns in the [Station Job List](#) include the following:

- **Job Step** — the type of job step, such as “Backup Stations” and so on (for more details, see “[Niagara provisioning job steps](#)” on page 3-25).
- **Start** — time the step started executing for the station.
- **End** — time the step completed for the station.
- **Status** — the status of the step for the station (note that the overall status for the job, and the status value for the step in other stations, may be different) as one of the following:
 - Running — step is executing.
 - Canceling — request to cancel the step was sent, but has not been processed yet, and the step is still executing.
 - Success — step finished successfully, and is no longer running.
 - Canceled — step was canceled before it completed, and is no longer running.
 - Failed — the step failed and is no longer running.

The status column ends with a “>>” (Details) button for each job, this function the same as the **Step**

Details button at the bottom of this view.

Buttons

Buttons near the bottom of the [Station Job List](#) become enabled when you have a step row selected in the [Step Table](#). Buttons are described as follows:

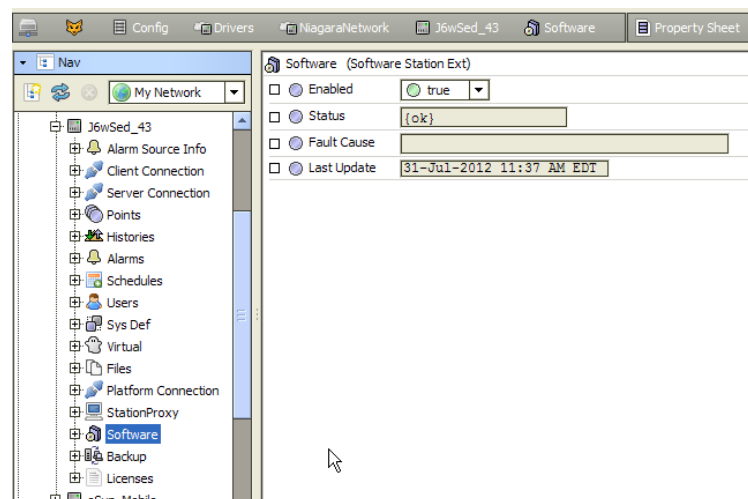
- **View Log** — For a popup **Job Log** dialog, showing log messages output when executing this step in the job. For more details, see “[Job Log notes](#)” on page 3-6.
- **Step Detail** — Switches the view to the **Batch Job Step Log File View**. For more details, see “[Batch Job Step Log File View](#)” on page 3-9.
- **Job Summary** — To change to the **Batch Job Log File View** (for the job that contained this step). For more details, see the “[Batch Job Log File View](#)” on page 3-9.

Software

The Software extension (SoftwareStationExt component), among the [provisioning extensions](#), permits any of the following steps in provisioning jobs to be processed for a station running on this host: [Install Software step](#), [Upgrade Out-of-date Software step](#), [Copy Supervisor File step](#), [Reboot step](#).

By default, this extension is enabled, to allow these provisioning steps for the station. [Figure 4-10](#) shows the property sheet for the station’s Software extension. Note that other functions are also provided by this extension, using its default [Station Software View](#) (a special view).

Figure 4-10 Software Station Ext property sheet



See the following sections for more details:

- [Properties](#)
- [Station Software View](#)

Properties

Properties of the [Software](#) station provisioning extension are described as follows:

- **Enabled** — Defaults to `true`, set to `false` if you wish to prevent steps listed above to process against this station in any provisioning jobs. Also, note this must be `true` to use the [Station Software View](#), or use the [Supervisor Software Manager](#) with this station.
- **Status** — Current status of extension, where value is `ok`, `disabled`, or `fault`.
The platform connection will be in `fault` if any of the following occurs:
 - Supervisor has no `ProvisioningNwExt` under its `NiagaraNetwork` (e.g., it has been deleted).
 - Supervisor is not licensed for provisioning.
 - `NiagaraStation` is in fault.
 - Host’s platform daemon rejects the platform connection’s credentials.
 The extension will be disabled if its **Enabled** property is set to `false`, or if the `ProvisioningNwExt` is disabled.
- **Fault Cause** — If in fault, gives brief text explanation.
- **Last Update** — Date/timestamp of when a “platform snapshot” was last updated (is “null” if a platform snapshot has never occurred).

Station Software View

The **Station Software View** is the default view on the **Software** provisioning extension of a NiagaraStation. As shown in [Figure 4-11](#), this view closely resembles the “**Software Manager**” view available in a direct platform connection to a host.

Figure 4-11 Station Software View is default view on Software provisioning extension

Module Name	Station Version	Supervisor Version	
aapmp	-	Tridium 3.7.42	Not Installed
aapup	-	Tridium 3.7.42	Not Installed
adr	-	Tridium 3.7.42	Not Installed
ak255	Tridium 3.7.41	Tridium 3.7.42	Out of Date
alarm	Tridium 3.7.42	Tridium 3.7.42	Up to Date
alarmOrion	-	Tridium 3.7.42	Not Installed
alarmRdb	-	Tridium 3.7.42	Not Installed
andoverAC256	-	Tridium 3.7.42	Not Installed
andoverInfinity	-	Tridium 3.7.42	Not Installed
app	Tridium 3.7.42	Tridium 3.7.42	Up to Date
axisVideo	-	Tridium 3.7.42	Not Installed
axvelocity	-	Tridium 3.7.42	Not Installed
backup	Tridium 3.7.42	Tridium 3.7.42	Up to Date
bacnet	Tridium 3.7.42	Tridium 3.7.42	Up to Date
bacnetAws	Tridium 3.7.42	Tridium 3.7.42	Up to Date
bacnetOws	Tridium 3.7.42	Tridium 3.7.42	Up to Date
bacnetws	-	Tridium 3.7.42	Not Installed
baja	Tridium 3.7.42.1	Tridium 3.7.42.1	Up to Date
bajaScript	Tridium 3.7.42	Tridium 3.7.42	Up to Date
bajaui	Tridium 3.7.42	Tridium 3.7.42	Up to Date
baseRtsp	-	Tridium 3.7.42	Not Installed
basicDriver	Tridium 3.7.42	Tridium 3.7.42	Up to Date
batchJob	-	Tridium 3.7.42	Not Installed

Note: Starting in AX-3.5, a number of changes were made to the platform **Software Manager**, including the ability to install an older revision of a selected software module (right-click menu), as well as the ability to install a new module without resulting in an automatic reboot of the target JACE.

Refer to the “**Software Manager**” section in the *NiagaraAX Platform Guide* for descriptions of most elements in this view. Only elements that differ from that view are explained here.

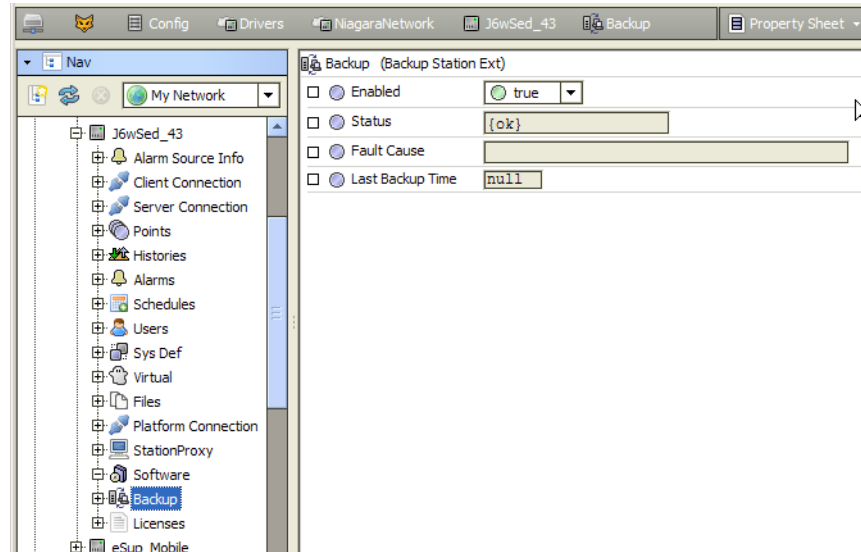
Whenever you access this view, a “snapshot” is taken of that NiagaraStation’s installed software, and this view’s table reflects its current software configuration. Other differences from the **Software Manager** view in a direct platform connection are summarized as follows:

- Where the **Software Manager** has table columns “Installed Version” and “Available Version” that represent the station’s and Workbench’s versions of each software item, the **Station Software View** has equivalent columns “Station Version” and “Supervisor Version”.
- Instead of a **Commit** button that starts the software installation by running it in Workbench, there is an **Execute Job** button, to submit the installation as a provisioning batch job in the Supervisor station. When you click **Execute Job**, the view changes to [Niagara Network Job View](#).

Backup

The Backup extension (BackupStationExt component), among the [provisioning extensions](#), permits [Backup Stations steps](#) in provisioning jobs to be processed for a station running on this host. By default, this extension is enabled, to allow backups in Niagara provisioning jobs. [Figure 4-4](#) shows the [properties](#) for the stations Backup extension (no “special” views exist for this component).

Figure 4-12 Backup Station Ext property sheet



Properties

Properties of the [Backup](#) station provisioning extension are described as follows:

- **Enabled** — Defaults to `true`, set to `false` if you wish to prevent Backup Stations steps to process against this station in any provisioning jobs.
- **Status** — Current status of extension, where value is `ok`, `disabled`, or `fault`. The extension is disabled if its **Enabled** property is set to `false`, or if the ProvisioningNwExt is disabled.
- **Fault Cause** — If in fault, gives brief text explanation.

In addition, there is a single [Start Backup action](#).

Start Backup action

Each [Backup](#) station provisioning extension provides its own **Start Backup** action, which when invoked immediately submits a provisioning job that contains a single Backup Stations step against this station.

Note: *If needed, you can add and link a TriggerSchedule output to this action, to provide specific time(s) or interval for backing up this particular station. An even better practice is to add a “job prototype” component to the station and configure it to do this, using its own TriggerSchedule. This method provides automatic backup job disposition, using configured “job retention” parameters.*

For related details, see “[About the Niagara Network Job Prototype](#)” on page 3-17, including subsections “[Scheduling a job prototype](#)” and “[Job retention](#)”.

Licenses

The Licenses extension (LicenseStationExt component), among the [provisioning extensions](#), permits the [Update Licenses step](#) in Niagara provisioning jobs to be processed for a station running on this host. By default, this extension is enabled.

Figure 4-13 License Station Ext property sheet

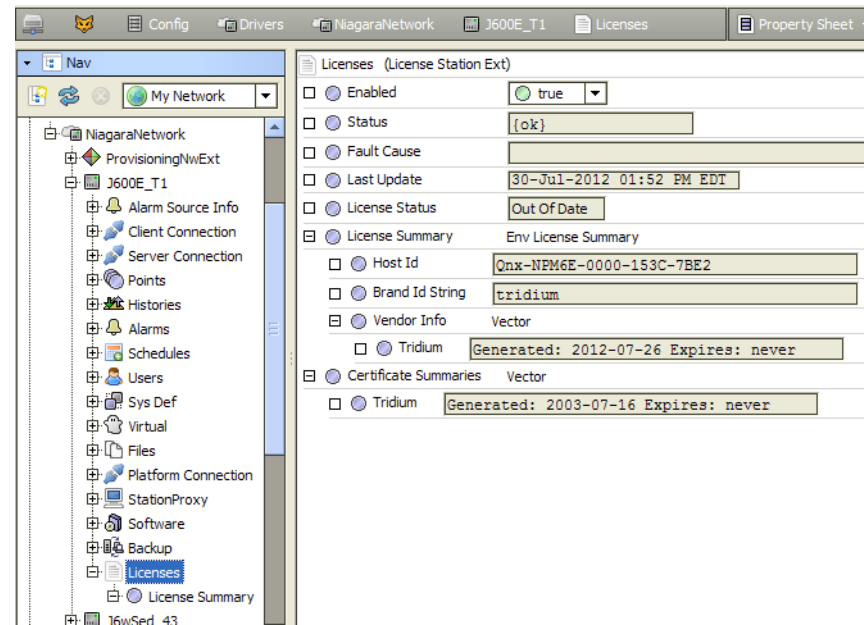


Figure 4-4 shows the [properties](#) for a station's Licenses extension (no “special” views exist for this component). This view appears when you double-click a license row in the “[Network License Summary](#)” view of the ProvisioningNwExt’s “[Licenses](#)” slot, and is used to update the table in that view.

Properties

Properties of the [Licenses](#) station provisioning extension are described as follows:

- **Enabled** — Defaults to `true`, set to `false` if you wish to prevent Update Licenses steps to process against this station in any provisioning jobs.
- **Status** — Current status of extension, where value is `ok`, `disabled`, or `fault`. The extension is `disabled` if its **Enabled** property is set to `false`, or if the ProvisioningNwExt is `disabled`.
- **Fault Cause** — If in `fault`, gives brief text explanation.
- **Last Update** — Timestamp of when the license was last updated by Niagara provisioning, or “null” if it has not yet been updated this way.
- **License Status** — Typically either “Up to Date” or “Unknown” (the latter if a license update (step or action) from Niagara provisioning has never occurred. Other states are possible, such as “Expired”.
- **License Summary** — Container slot for properties describing the license(s), including the unique Host ID for the platform, “brand” identifiers, and other vendor license info including timestamps when each license was originally generated, and date of expiration (if ever).
- **Certificates Summaries** — Container slot for properties describing installed certificate(s), including when each certificate was originally generated, and date of expiration (if ever).

In addition, the Licenses station extension has a single [Poll action](#).

Poll action

Each [Licenses](#) station provisioning extension provides its own **Poll** action, which when invoked immediately submits an “Update” request to retrieve the latest license from the licensing server (or if unavailable, from the Supervisor’s local license database). This action is equivalent to the “**Update**” command issued from the **License Summary View** of the ProvisioningNwExt’s Licenses slot.

CHAPTER 5

Provisioning Plugin Guides

Plugins (views) provide visualizations of components. Summary descriptions of views are available by selecting **Help > On View** (F1) from the menu, or by pressing F1 while the view is open. The following [Plugin Reference Summary](#) includes modules related to provisioning Niagara in AX-3.5 and later.

Plugin Reference Summary

Summary information is provided on views in the following Niagara provisioning-related modules:


- [batchJob](#)
- [provisioningNiagara](#)

Plugins in batchJob module

Provisioning Niagara includes use of the following batchJob views (listed alphabetically), each with a brief summary description:

- [Batch Job Log File View](#)
- [Batch Job Step Log File View](#)
- [Job Step Details View](#)
- [Prototype Job List](#)


batchJob-BatchJobLogFileView

 This view shows the details for a single job execution, and is the default view for any batch job log file (.bjl extension). Typically, you see this from either the [Niagara Network Job List](#) or the [Prototype Job List](#) when you click the “>>” button on any completed job. You also see this view when in the [Batch Job Step Log File View](#) and you click the **Job Summary** button.

The view provides a summary table showing each step in the job, for each station. On any step, you click the “>>” button to see the [Batch Job Step Log File View](#) to see complete details for that step. Or, you can click the **View Log** button to see a popup **Job Log** dialog listing all logged messages for this job.

For more details, see “[Batch Job Log File View](#)” on page 3-9.

batchJob-BatchJobStepLogFileView


 This view shows the details for a provisioning job step executed (and completed) against a single station, and is the default view for any batch job step log file (.bjsl extension). Typically, you see this from either the [Niagara Network Job View](#) or the [Station Job List](#) when you click the “>>” button on any completed step.

The view provides a summary table showing log messages that occurred during step execution. You can double-click rows to show additional **Log Details**, if any.

For more details, see “[Batch Job Step Log File View](#)” on page 3-9.

Note: *If the job step was Backup Station, and it executed successfully, a **Restore** button is available. This allows you to restore the station using the .dist file saved from this provisioning job. If you answer Yes to the confirmation dialog (no undo), an install backup job is immediately executed, and the view changes to the [Niagara Network Job View](#) for that job.*

batchJob-JobStepDetailsView


 This view shows the details for a provisioning job step executed against a single station, while that step is *still running*. Typically, you see this from either the [Niagara Network Job View](#) or the [Station Job List](#) when you click the “>>” button on a running step.

The view provides a summary table showing log messages that occurred during step execution. You can double-click rows to show additional **Log Details**, if any.

For more details, see “[Job Step Details View](#)” on page 3-11.

Note: *If the job step was Backup Station, that step is still running, you see a slightly different view, the [Backup Step Details View](#). In either case, once the step is finished, neither view is accessible—they are replaced by the [Batch Job Step Log File View](#).*

batchJob-PrototypeJobList

 This is an available view on any reusable Niagara provisioning job component ([NiagaraNetworkJob-Prototype](#)) and allows configuration of the following:

- Retention policy — Rules on whether executed jobs be retained permanently, disposed of after a defined time period, or disposed of after a defined number of executions.
- Policy enforcement frequency— Specifies the repeating interval at which the component’s retention policy should be evaluated (processed).

This view also provides a table-based history of this batch job and its results. Included for each job is a **Job Log**, available as a popup dialog via double-clicking a job row, and a [Batch Job Log File View](#), accessed by clicking the “>>” button next to its status descriptor.


For more details, see “[Prototype Job List](#)” on page 3-24.

Plugins in provisioningNiagara module

Provisioning Niagara includes use of the following provisioningNiagara views (listed alphabetically), each with a brief summary description:

- [Backup Step Details View](#)
- [Network License Summary](#)
- [Niagara Network Job Builder](#)
- [Niagara Network Job List](#)
- [Niagara Network Job View](#)
- [Niagara Network Prototype View](#)
- [Provisioning Manager](#)
- [ProvisioningRobotEditor](#)
- [Provisioning Station Director](#)
- [Station Job List](#)
- [Station Software View](#)
- [Supervisor License Manager](#)
- [Supervisor Software Manager](#)

provisioningNiagara-BackupStepDetailsView


 This view shows the details for a Backup Stations step executed against a single station, while that step is *still running*. Typically, you see this from either the [Niagara Network Job View](#) or the [Station Job List](#) when you click the “>>” button on a running step.

The view provides a summary table showing log messages that occurred during step execution. You can double-click rows to show additional **Log Details**, if any.

For more details, see “[Job Step Details View](#)” on page 3-11


Note: *If the job step type was something other than Backup Stations, and that step is still running, you see a slightly different view, the [Niagara Network Job View](#). In either case, once the step is finished, neither view is accessible—they are replaced by the [Batch Job Step Log File View](#).*

provisioningNiagara-NetworkLicenseSummary

 This is the default view on the **Licenses** slot ([SupervisorLicenses](#)) of the **ProvisioningNwExt** under the NiagaraNetwork. It lists all child NiagaraStations, including their unique host IDs, current license status, and last time of license update. Information in this table is populated by the **Licenses** device extension ([LicenseStationExt](#)) under each NiagaraStation.

For more details, see “[Network License Summary](#)” on page 3-16.


provisioningNiagara-NiagaraNetworkJobBuilder

 The Niagara Network Job Builder view is the default view for the **ProvisioningNwExt** ([ProvisioningNiagaraNetworkExt](#)) of a NiagaraNetwork. It allows you to build a provisioning job for immediate execution (to run now), where you specify steps to be performed in the job, and for which stations. Once you specify a job, you can then submit it for execution.

Access this view by simply double-clicking the **ProvisioningNwExt**, or by right-clicking it and selecting **Views > Niagara Network Job Builder**.

For more details, see “[Niagara Network Job Builder](#)” on page 3-2.


provisioningNiagara-NiagaraNetworkJobList

 The Niagara Network Job List view is an available view for the **ProvisioningNwExt** of a NiagaraNetwork. It provides a Supervisor-wide table-based history of provisioning jobs and their results. Included for each job is a Job Log, available as a popup dialog via double-clicking a job row, and a Job Details view, accessed by clicking the “>>” button next to its status descriptor.

Access this view by right-clicking and selecting **Views > Provisioning Job List**, or selecting it from the ProvisioningNwExt’s view selector.

For more details, see “[Niagara Network Job List](#)” on page 3-5.


provisioningNiagara-NiagaraNetworkJobView

 The Niagara Network Job View shows the details for a single job execution. It is the default view for any **NiagaraNetworkJob**. From the [Niagara Network Job Builder](#) or [Niagara Network Prototype View](#), you change to this view when you specify a job to **Run Now**. You also see this view when in the [Niagara Network Job List](#) and you click the “>>” button on a job that is still running, or from the [Batch Job Step Log File View](#) when you click **Job Summary**.

The view provides a summary table showing each step in the job, for each station. Controls allow you to view a job log, cancel an uncompleted job, or dispose of a completed job.

For more details, see “[Niagara Network Job View](#)” on page 3-7.

provisioningNiagara-NiagaraNetworkPrototypeView


 The Niagara Network Prototype View is the default view for a Niagara provisioning job saved as a reusable component ([NiagaraNetworkJobPrototype](#)). It is nearly identical to the [Niagara Network Job Builder](#), in that you specify job steps, and the stations to be included. However, in addition to being able to run the job immediately, you can also *save* the prototyped job for use later, and/or *schedule* its execution using a separately added (and linked) **TriggerSchedule**.

Note: Use a separately copied **TriggerSchedule** for each **NiagaraNetworkJobPrototype** for linking/scheduling—do not use the existing **TriggerSchedule** (named “*Backup Schedule*”) under the **ProvisioningNwExt** for this.

Access this view by simply double-clicking a [NiagaraNetworkJobPrototype](#) component, or by right-clicking it and selecting **Views > Niagara Network Prototype View**.


For more details, see “[Niagara Network Prototype View](#)” on page 3-21.

provisioningNiagara-ProvisioningManager

 The Provisioning Manager is an available view on the Supervisor’s NiagaraNetwork, providing it has the **ProvisioningNwExt** ([ProvisioningNiagaraNetworkExt](#)). This view provides a central look at the status and health of “platform connectivity” to the various Niagara hosts, as well as quick access to some of the provisioning (device) extensions under each **NiagaraStation**.


For more details, see “[Provisioning Manager](#)” on page 4-2.

provisioningNiagara-ProvisioningRobotEditor

 The Provisioning Robot Editor is the default view of a [ProvisioningRobot](#) in the Supervisor station. It closely resembles the “Edit” tab of the **Program Editor** view for Program components, in that you view, edit, and compile the Baja code represented as the **ProvisioningRobot**.

For more details, see “[Provisioning Robot Editor](#)” on page 3-30.


provisioningNiagara-ProvisioningStationDirector

 The Provisioning Station Director is the default view of the [StationProxy](#) extension under a **NiagaraStation** device. It closely resembles the Station Director view available in a platform connection, where station output can be observed, and other station control is available.

Although the two views operate similarly, the Provisioning Station Director is slightly less responsive, as it connects through the Supervisor station as an intermediary (versus a direct platform connection, as used by the Station Director).

For more details, see “[Provisioning Station Director](#)” on page 4-7.


provisioningNiagara-StationJobList

 The Station Job List view is available for the [StationProxy](#) extension under a NiagaraStation device. It shows a list of provisioning job steps that were executed against this specific station. Included for each job step is a **Job Log**, available as a popup dialog via double-clicking a job row, and a [Batch Job Step Log File View](#), accessed either by clicking the “>” button next to its status descriptor, or by clicking the **Step Details** button.

Access this view by right-clicking the [StationProxy](#) extension and selecting **Views > Station Job List**, or by selecting it from the extension’s view selector.


For more details, see “[Station Job List](#)” on page 4-8.

provisioningNiagara-StationSoftwareView

 Station Software View is the default view of the [SoftwareStationExt](#) under a NiagaraStation device. It closely resembles the Software Manager view available in a platform connection. When this view loads, a “snapshot” of installed software is automatically updated, so that the table reflects the station’s current state.


For more details, see “[Station Software View](#)” on page 4-10.

provisioningNiagara-SupervisorLicenseManager

 This is an available view on the **Licenses** slot ([SupervisorLicenses](#)) of the **ProvisioningNwExt** under the NiagaraNetwork. It provides management access to the Supervisor’s local license database (under its `!licenses/db` subdirectory).

For more details, see “[Supervisor License Manager](#)” on page 3-17.

provisioningNiagara-SupervisorSoftwareManager

 The Supervisor Software Manager is the default view on the [SoftwareContainer](#) of the ProvisioningNwExt in a NiagaraNetwork. It lists all [InstallableSummary](#) components, representing the installable software items under the `!sw` directory on the Supervisor, with each expandable to show [InstallableSpecs](#). Controls at the bottom provide different software management functions.

For more details, see “[Supervisor Software Manager](#)” on page 3-14.

CHAPTER 6

Provisioning Niagara Component Guides

This [Component Reference Summary](#) provides summary descriptions of components related to provisioning Niagara in AX-3.5 and later.

Component Reference Summary

Summary information is available on components in the following provisioning-related modules:

- [batchJob](#)
- [provisioningNiagara](#)

Components in batchJob module

- [BatchJobService](#)
- [ThreadPoolJobQueue](#)

batchJob-BatchJobService

◆ The BatchJobService is responsible for dispatching all provisioning jobs (as batch jobs) on a Supervisor, where batch jobs are run by the station's JobService. The BatchJobService replaces the “job control” functions formerly handled by the ProvisioningService in AX-3.1 or AX-3.2. Other Niagara provisioning functions are handled by the NiagaraNetwork's **ProvisioningNwExt** ([ProvisioningNiagaraNetworkExt](#)).

Note: The BatchJobService requires the Supervisor station to also have the HistoryService and JobService, otherwise the BatchJobService will be in fault. Typically, any Supervisor already has these services. In addition, the BatchJobService requires the Supervisor host platform to be licensed with the “provisioning” feature, or else the service will be in fault.

For more details, see “BatchJobService” on page 3-33.

batchJob-ThreadPoolJobQueue

● The Thread Pool Job Queue is a frozen container slot under the [BatchJobService](#). Its only property, **Max Threads**, specifies the maximum number of concurrent provisioning jobs that can be performed by the Supervisor. By default, this is one (1), and is sometimes left at default, as provisioning threads can be resource intensive in a Supervisor station.


Note: Only after determining the station has available resource overhead, should Max Threads be adjusted up over 2 or 3. Otherwise, other tasks performed by the station may be affected.

Components in provisioningNiagara module

- [BackupStationExt](#)
- [FileCopyStep](#)
- [InstallableSummary](#)
- [InstallableSpec](#)
- [InstallBySpecStep](#)
- [InstallStep](#)
- [LicenseStationExt](#)
- [PlatformConnection](#)
- [ProvisioningBackupStep](#)
- [NiagaraNetworkJob](#)
- [NiagaraNetworkJobPrototype](#)


- [ProvisioningNiagaraNetworkExt](#)
- [RebootJobStep](#)
- [SoftwareContainer](#)
- [SoftwareStationExt](#)
- [Station Poll Scheduler](#)
- [StationProxy](#)
- [UpgradeOutOfDateStep](#)

provisioningNiagara-BackupStationExt

 The BackupStationExt (**Backup**) is one of five device extensions automatically added to every NiagaraStation under the Supervisor's NiagaraNetwork, providing the Supervisor also has the ProvisioningNwExt ([ProvisioningNiagaraNetworkExt](#)) under its NiagaraNetwork. Other NiagaraStation provisioning extensions are the [LicenseStationExt](#), [PlatformConnection](#), [SoftwareStationExt](#), and [StationProxy](#).


For more details, see “[Provisioning Station Extension Concepts](#)” on page 4-1, and “[Backup](#)” on page 4-11.

provisioningNiagara-FileCopyStep


 CopyFile steps can be added in the middle pane (“to run for each station”) in the [Niagara Network Job Builder](#) or the [Niagara Network Prototype View](#), when specifying a provisioning job. This produces the standard File Chooser dialog, in which you specify the file to copy. When working in the Niagara Network Job Builder view, you can select either a Supervisor file or a Local file (on your PC) to copy; if working in the Niagara Network Schedule View you can only select a Supervisor file. After selecting the source file, a Destination dialog lets you specify a system-home-relative or station-home-relative path, to which the file is copied to (in all specified stations). Note that other types of provisioning steps can be also be added.

For more details, see “[Niagara provisioning job steps](#)” on page 3-25, “[Copy Supervisor File step](#)” on page 3-26, and “[Copy Local File step](#)” on page 3-27.


provisioningNiagara-InstallableSummary

 Installable Summaries represent software files under the !sw directory of the Supervisor host, and reside under the [SoftwareContainer](#) of the NiagaraNetwork's **ProvisioningNwExt**. Each contains one or more specification objects for each version in the software registry, represented as [InstallableSpec](#) components. For more details, see “[Supervisor Software container](#)” on page 3-14.

provisioningNiagara-InstallableSpec


 Installable Specs are children of [InstallableSummary](#) components, as the lowest level component under the [SoftwareContainer](#) of the NiagaraNetwork's **ProvisioningNwExt**. Installable Specs are version-specific, and describe the installable software item, including version number, dependencies, and other data. For more details, see “[Supervisor Software container](#)” on page 3-14.

provisioningNiagara-InstallBySpecStep


 InstallSoftware steps can be added in the middle pane (“to run for each station”) in the [Niagara Network Job Builder](#) or the [Niagara Network Prototype View](#), when specifying a provisioning job. When adding a software item (module or dist), you must specify a specific version. Note that other types of provisioning steps can be also be added.

For more details, see “[Niagara provisioning job steps](#)” on page 3-25, and “[Install Software step](#)” on page 3-28.

provisioningNiagara-InstallStep


 Install backupdist steps are created when you copy an existing backup .dist file into the middle job step pane (“to run for each station”) in the [Niagara Network Job Builder](#) or the [Niagara Network Prototype View](#). This step is different from a “version-specific” install step ([InstallBySpecStep](#)) produced by either the “Install Software” step selection (where you must select a particular version), or by copying a specific (versioned) module or .dist file into the job step pane.

provisioningNiagara-LicenseStationExt

 The LicenseStationExt (**Licenses**) is one of five device extensions automatically added to every NiagaraStation under the Supervisor's NiagaraNetwork, providing the Supervisor also has the ProvisioningNwExt ([ProvisioningNiagaraNetworkExt](#)) under its NiagaraNetwork. Other NiagaraStation provisioning extensions are the [BackupStationExt](#), [PlatformConnection](#), [SoftwareStationExt](#), and [StationProxy](#).


For more details, see “[Provisioning Station Extension Concepts](#)” on page 4-1, and “[Licenses](#)” on page 4-12.

provisioningNiagara-PlatformConnection


 The PlatformConnection is one of five device extensions automatically added to every NiagaraStation under the Supervisor's NiagaraNetwork, providing the Supervisor also has the ProvisioningNwExt ([ProvisioningNiagaraNetworkExt](#)) under its NiagaraNetwork. Other NiagaraStation provisioning extensions are the [BackupStationExt](#), [LicenseStationExt](#), [SoftwareStationExt](#), and [StationProxy](#).

For more details, see “[Provisioning Station Extension Concepts](#)” on page 4-1, and “[Platform Connection](#)” on page 4-3.

provisioningNiagara-ProvisioningBackupStep


 BackupStation steps can be added in the middle pane (“to run for each station”) in the [Niagara Network Job Builder](#) or the [Niagara Network Prototype View](#), when specifying a provisioning job. When executed, this step makes an online backup for each specified station (providing station is running), or if a specified station is not running, an offline backup. In either case, the backup dist file for each station is stored on the Supervisor. Note that other types of provisioning steps can be also be added. For more details, see “[Niagara provisioning job steps](#)” on page 3-25, and “[Backup Stations step](#)” on page 3-26.

provisioningNiagara-NiagaraNetworkJob

 A NiagaraNetworkJob specifies a list of job stages with steps that were performed on each Niagara station in the list. NiagaraNetworkJobs are found under the Supervisor station's JobService. The default view of a NiagaraNetworkJob is the [Niagara Network Job View](#).


For more details, see “[Niagara Network Job View](#)” on page 3-7.

provisioningNiagara-NiagaraNetworkJobPrototype

 A [NiagaraNetworkJobPrototype](#) is a component representing a potential Niagara provisioning job. A NiagaraNetworkJobPrototype may be copied from the palette (or duplicated from another NiagaraNetworkJobPrototype). You can link a TriggerSchedule to its “Submit Job” action in order to schedule executions of this job, as well as run it immediately, as needed. The default view of this component is the [Niagara Network Prototype View](#). Another [Prototype Job List](#) view is also available, which provides configurable parameters for job retention.

For more details, see “[About the Niagara Network Job Prototype](#)” on page 3-17, including subsections “[Scheduling a job prototype](#)” on page 3-18 and “[Job retention](#)” on page 3-19.


provisioningNiagara-ProvisioningNiagaraNetworkExt

 The ProvisioningNiagaraNetworkExt ([ProvisioningNwExt](#)) is a network extension for the NiagaraNetwork in a Supervisor station. It is required to enable provisioning of hosts represented by NiagaraStations under that network. It provides a central spot to perform Niagara provisioning tasks, dispatched as batch jobs by the station's [BatchJobService](#), and executed by the station's JobService.

The ProvisioningNwExt contains frozen slots that serve different functions, and has special views (apart from standard component views). The default view is the [Niagara Network Job Builder](#), and a [Niagara Network Job List](#) view is also available.

For more details, see “[About the ProvisioningNwExt](#)” on page 3-2.

provisioningNiagara-ProvisioningRobot

 A ProvisioningRobot is a special Program component that can be customized and saved on the Supervisor station, then selected as the source “robot” in a [RunRobotStep](#) when configuring a Niagara Network provisioning job. When queued and run by the Supervisor's [BatchJobService](#), the program code in the ProvisioningRobot is executed by each station included in the provisioning job, by each station's ProgramService.


Note: *Starting in AX-3.7, by default you require “super user” permissions on a station to add/edit Program and Robot components, including ProvisioningRobot components. For related details see “[Provisioning Robot notes](#)” on page 3-30.*

As copied from the provisioningNiagara palette, a ProvisioningRobot is “pre-configured” with some essential “wrapper” lines of code. Also included is placeholder “do something” code that may prove useful in some cases (if modified in a way meaningful when run against the stations included in a job).

Find the ProvisioningRobot in the provisioningNiagara palette. The default view for the ProvisioningRobot is the [ProvisioningRobotEditor](#), where you edit and compile the Baja code intended to be run by all stations in a Niagara provisioning job.


For more details, see “[Run Robot step](#)” on page 3-30 and “[Provisioning Robot notes](#)” on page 3-30.

provisioningNiagara-RebootJobStep

 RebootStations step can be added in the middle pane (“to run for each station”) in the [Niagara Network Job Builder](#) or the [Niagara Network Prototype View](#), when specifying a provisioning job. This results in platform (host) reboot of hosts running the specified stations. Note that other types of provisioning steps can be also be added.


For more details, see “[Niagara provisioning job steps](#)” on page 3-25, and “[Reboot step](#)” on page 3-29.

provisioningNiagara-RunRobotStep

 RunRobot steps can be added in the middle pane (“to run for each station”) in the [Niagara Network Job Builder](#) or the [Niagara Network Prototype View](#), when specifying a Niagara provisioning job. For any Run Robot step, you use the “Select a Robot” dialog to specify a Provisioning Robot on the Supervisor, to be run by the ProgramService of each station in the provisioning job.


For more details, see “[Niagara provisioning job steps](#)” on page 3-25, and “[Run Robot step](#)” on page 3-30.

provisioningNiagara-SoftwareContainer

 The Software Container (**Software**) under the ProvisioningNwExt provides visibility into the Supervisor’s software registry (the installable software files stored under the !sw directory). Children in this container are summary objects for named, typed, software files—for example, modules and platform distributions, represented as [InstallableSummary](#) components. The default view is the [Supervisor Software Manager](#).


For more details, see “[Supervisor Software container](#)” on page 3-14.

provisioningNiagara-SoftwareStationExt

 The SoftwareStationExt (**Software**) is one of five device extensions automatically added to every NiagaraStation under the Supervisor’s NiagaraNetwork, providing the Supervisor also has the ProvisioningNwExt ([ProvisioningNiagaraNetworkExt](#)) under its NiagaraNetwork. Other NiagaraStation provisioning extensions are the [BackupStationExt](#), [LicenseStationExt](#), [PlatformConnection](#), and [StationProxy](#). The default view is the [Station Software View](#).


For more details, see “[Provisioning Station Extension Concepts](#)” on page 4-1, and “[Software](#)” on page 4-9.

provisioningNiagara-StationPollScheduler

 The Station Poll Scheduler controls the polling of StationProxy extensions by the ProvisioningNwExt of the NiagaraNetwork. As a container slot under this network extension, the poll scheduler operates as in most driver networks. See “About poll components” in the *Drivers Guide* for general information. Polling only happens for StationProxy extensions that are subscribed (linked somewhere, or being viewed or charted, etc.).


Note: Often, you can leave configuration at defaults. Or if needed, assign the Poll Frequency property of StationProxy slots (NiagaraStation provisioning extensions) differently than the default “Normal.”

provisioningNiagara-StationProxy

 The **StationProxy** is one of five device extensions automatically added to every NiagaraStation under the Supervisor’s NiagaraNetwork, providing the Supervisor also has the ProvisioningNwExt ([ProvisioningNiagaraNetworkExt](#)) under its NiagaraNetwork. Other NiagaraStation provisioning extensions are the [BackupStationExt](#), [LicenseStationExt](#), [PlatformConnection](#), and [SoftwareStationExt](#).

For more details, see “[Provisioning Station Extension Concepts](#)” on page 4-1, and “[StationProxy](#)” on page 4-5.


provisioningNiagara-SupervisorLicenses

 The SupervisorLicenses (**Licenses**) slot under the NiagaraNetwork’s **ProvisioningNwExt** provides two views:

- Network License Summary — Default view, lists NiagaraStations and their host IDs.
- Supervisor License Manager — Provides a manager view into the Supervisor’s local license database (under its !licenses/db subdirectory).

This component has no other properties or actions. For more details, see “[Licenses slot](#)” on page 3-16.

provisioningNiagara-UpdateLicensesJobStep


 UpdateLicensesJobStep (Update Licenses) can be added in the *top* pane (“initial steps to run only once”) in the [Niagara Network Job Builder](#) or the [Niagara Network Prototype View](#), when specifying a provisioning job. Currently, this is the *only* step type you can add as an initial step. When processed, the Supervisor gathers information on the licenses installed on the target station’s host(s), and then accesses

the license server (in one message) to see if they are up-to-date. If any updated (different) licenses are found, they are installed to the target station's host(s)—and also updated in the Supervisor's local license database.

If the Supervisor is not configured for Internet connectivity, then only its local license database is used to compare against licenses installed on the target station's host(s). If any updated (different) licenses are found, they are installed on the target station's host(s).

For more details, see [“Niagara provisioning job steps”](#) on page 3-25 and [“Update Licenses step”](#) on page 3-26.

provisioningNiagara-UpgradeOutOfDateStep

 UpgradeOutOfDateStep can be added in the middle pane (“to run for each station”) in the [Niagara Network Job Builder](#) or the [Niagara Network Prototype View](#), when specifying a provisioning job. When processed, the versions of software on the target station's host(s) are compared with the latest versions of the same software in the Supervisor's software database. If any newer versions are found, they are installed to the target station's host(s). Note that other types of provisioning steps can be also be added.

For more details, see [“Niagara provisioning job steps”](#) on page 3-25, and [“Upgrade Out-of-date Software step”](#) on page 3-30.

APPENDIX A

Converting from old style (AX-3.1) Provisioning

This appendix provides details on converting the “old style” provisioning used in a Supervisor previously running AX-3.2 or AX-3.1, which has now been upgraded to a newer NiagaraAX release. Such a Supervisor station has a “ProvisioningService” component in its “Services” container.

Note: *It is expected this scenario to rarely occur, which is why this information was moved to this appendix. If an upgraded Supervisor was previously running AX-3.3 or AX-3.4 (and was already configured for provisioning), it is probably already using the “new style” provisioning. The new style uses a “ProvisioningNwExt” as a child of its NiagaraNetwork, and does not have a “ProvisioningService”.*

Conversion- Supervisor previously running AX-3.1/AX-3.2 style provisioning

To convert a Supervisor upgraded to a later release to use the newer-style provisioning, a conversion utility is included in Workbench as a right-click command on the Supervisor’s **Provisioning-Service**. You use this command while the Supervisor station is running.

Conversion applies only if the Supervisor was configured for provisioning when running AX-3.1 or AX-3.2. Otherwise, for “quick start” information see [“Add and configure necessary components”](#) on page 2-1.

Note: *Before converting a Supervisor station to the newer “ProvisioningNwExt-based” Niagara provisioning, please note the following:*

- “Pending jobs” (if any) that are under the Pending Jobs folder under the ProvisioningService are *not* converted—you may wish to examine them first, in order to replicate them after the conversion. Typically, you would use a (new) “job prototype” component to replace a pending job.
- Conversion launches two jobs: first a “Station Save” job for the Supervisor, then the actual “Conversion” job in which: the BatchJobService is added/converted, the ProvisioningService is *deleted*, the new ProvisioningNwExt is added to the NiagaraNetwork, and finally provisioning-related device extensions under NiagaraStation components are “mixed-in/updated.” See the station’s Job Service Manager for details on these jobs after converting.

See the following for more information:

- [Run the conversion to new style Provisioning](#)
- [Post conversion notes](#)

Run the conversion to new style Provisioning

Perform the following only after reading the previous [Note](#) in the previous section.

To convert the Supervisor to use the newer style provisioning

- Step 1 In Workbench, open the running Supervisor station.
- Step 2 Expand its Config space to reveal its **ProvisioningService** in its **Services** folder.
- Step 3 Right-click the **ProvisioningService**, and in the popup menu choose **Upgrade from Provisioning 3.1**.
A second **Confirm** dialog appears, asking whether to convert to a “ProvisioningNiagaraNetworkExt”. Click **Yes**.
- Step 4 The Workbench view changes to the station’s **Job Service Manager**, in which you see two related jobs—first a Station Save, then the Conversion. Both jobs typically complete within a few seconds.
- Step 5 The provisioning conversion is now complete. Notice that the ProvisioningService is now gone, but a BatchJobService was added. Also, the NiagaraNetwork now has a “ProvisioningNwExt”, with similar views as the old ProvisioningService.

Assuming that provisioning was previously configured and operational, it should now be operational for your NiagaraNetwork.

For “quick start” information see [“Run provisioning jobs”](#) on page 2-3.

Post conversion notes

- After conversion, you should be able to immediately run new provisioning jobs. However, because after the conversion there is no ProvisioningService (with associated views), you no longer have the previous Workbench access to older provisioning files, including backups, job log and step log details—these are now effectively orphaned.

Note that these older files *remain* under the Supervisor station’s “^provisioning” directory, including backups in the “^provisioning/stationData/*stationName*/backups” directory, and so these older backups are still restorable.

For example, at the Supervisor PC, using Workbench you could expand the “My File System” in the Nav tree to navigate to the stations’ provisioning backup folders (above), and then drag backup .dist file(s) into the Niagara Network Job Builder view (on the ProvisioningNwExt) as steps for each station list.

- After conversion, it is recommended that you delete the “docProvisioning.jar” file from the Supervisor’s !/modules folder, leaving only the equivalent “docProvisioningNiagara.jar” file (associated with this document) in its place. Otherwise, provisioning-related Help topics viewed at the Supervisor may show incorrect information.