

Provisioning Guide

User Guide

Provisioning Guide

Distech Controls, Inc.

Brossard, Quebec,
Canada

Legal Notice

©, Distech Controls Inc., 2024. All rights reserved. While all efforts have been made to verify the accuracy of information in this manual, Distech Controls is not responsible for damages or claims arising from the use of this manual. Persons using this manual are assumed to be trained HVAC professionals and are responsible for using the correct wiring procedures, correct override methods for equipment control and maintaining safe working conditions in fail-safe environments. Distech Controls reserves the right to change, delete or add to the information in this manual at any time without notice.

Distech Controls, the Distech Controls logo, EC-Net, and Innovative Solutions for Greener Buildings are registered trademarks of Distech Controls, Inc. LON, LonMark, LonWorks, LNS, and Neuron are registered trademarks of Echelon Corporation registered in the United States and other countries. NiagaraAX and NiagaraAX Framework are registered trademarks of Tridium, Inc. BACnet is a registered trademark of ASHRAE. Windows, Windows XP, Windows Vista, Windows 7, Visual Basic.Net, Visual Basic.Net are registered trademarks of Microsoft Corporation. Intel and Pentium are registered trademark of Intel Corporation in the U.S. and/or other countries. AMD is a registered trademark of Advanced Micro Devices, Inc. EnOcean is a registered trademark of EnOcean GmbH. All other trademarks are property of their respective owners.

Contents

About this Guide	9
Document change log	9
Related documents	10
Chapter 1 Provisioning overview	11
Provisioning FAQs	12
Provisioning Installation.....	12
Adding the BatchJobService	13
Adding the ProvisioningNwExt	13
Configuring platform port and credentials.....	13
Installation troubleshooting	14
Provisioning steps.....	15
Provisioning extensions.....	18
Provisioning-related alarms	19
Provisioning in a mixed AX/N4 or N4.x network	21
Chapter 2 Provisioning environment	23
One-time jobs	23
Creating a one-time job	23
Adding one-time job list steps	25
Updating connections using host name or IP address	26
Updating connections using a provisioning station's DHCP Server.....	26
Setting the reciprocal connection	27
Regularly-scheduled jobs	27
Creating a prototype job	28
Configuring prototype job retention policy	30
Prototype job actions.....	32
Setting up the prototype job schedule	33
Job list step management.....	33
Creating a provisioning robot	33
Adding a robot step to a provisioning job.....	34
Provisioning Job management.....	35
Job execution	35
Provisioning data files	35
Histories related to provisioning	36
Chapter 3 Housekeeping	37
Adding a station user	37
Removing a station user.....	38
Setting station user password	38
Automating host license updates using provisioning	39
Synchronizing with the Supervisor's license database	40
Updating licenses using the Network License Summary.....	40
Copying a file.....	41
Copying a Supervisor file.....	42

Rebooting the system	43
Removing stations	43
Renaming remote stations.....	43
Reordering stations.....	44
Setting a property	44
Removing a property.....	45
Setting the time.....	45
Chapter 4 Station backup	47
Manually triggering a backup	47
Preparing the folder and trigger schedule.....	47
Setting up the backup provisioning job.....	48
Chapter 5 Provisioned software installation.....	51
Synchronizing software databases.....	51
Installing software using the add button	52
Installing software by dragging from the Software container	53
Installing a clean distribution file.....	54
Installing new software modules.....	54
Upgrading out-of-date software.....	55
Chapter 6 Certificates and credentials.....	57
Configuring subordinate stations for SAML IdP Service and Scheme	57
Setting security job steps.....	59
Generating a certificate	60
Setting the certificate alias	61
Signing a certificate.....	61
Installing a certificate.....	63
Exporting a certificate signing request	64
Importing a signed certificate	65
Setting system passphrase.....	66
Setting platform credentials	67
Setting platform user password.....	68
Setting station connection credentials.....	68
Setting the TLS level	69
Chapter 7 Templates.....	71
Adding a template.....	71
Exporting the template spreadsheet	71
Editing the Excel file to configure template instances	73
Copying the Excel file and templates to the Supervisor	73
Setting up and running the bulk provisioning job	74
Upgrading a template	76
Provisioning application templates	77
Exporting configurations for an application template	77
Editing the spreadsheet for an application template	77
Installing an application template	78
Upgrading an application template	79

Updating template or application configurations	80
Chapter 8 Components.....	81
Components in the batchJob module.....	81
batchJob-BatchJobService	81
batchJob-ThreadPoolJobQueue	83
Extension components in the provisioningNiagara module	83
BackupStationExt (provisioningNiagara-BackupStationExt)	83
LicenseStationExt (provisioningNiagara-LicenseStationExt).....	85
ProvisioningNiagaraNetworkExt (provisioningNiagara-ProvisioningNiagaraNetworkExt).....	87
SoftwareStationExt (provisioningNiagara-SoftwareStationExt)	89
TemplateStationExt (provisioningNiagara-TemplateStationExt).....	90
UpdateStationExt (provisioningNiagara-UpdateStationExt)	91
Supporting components in the provisioningNiagara module	91
InstallableSummary (provisioningNiagara-InstallableSummary)	91
InstallableSpec (provisioningNiagara-InstallableSpec).....	92
InstallStep (provisioningNiagara-InstallStep).....	92
NiagaraNetworkJob (provisioningNiagara-NiagaraNetworkJob)	92
NiagaraNetworkJobPrototype (provisioningNiagara-NiagaraNetworkJobPrototype).....	92
PlatformConnection (provisioningNiagara-PlatformConnection)	92
ProvisioningRobot (provisioningNiagara-ProvisioningRobot).....	94
SoftwareContainer (provisioningNiagara-SoftwareContainer).....	94
StationPollScheduler (provisioningNiagara-StationPollScheduler)	95
StationProxy (provisioningNiagara-StationProxy)	95
SupervisorLicenses (provisioningNiagara-SupervisorLicenses)	98
DeviceBootstrap (provisioningNiagara-DeviceBootstrap)	98
Job step components in the provisioningNiagara module	99
Add Station User	99
HTML5 Add Station User.....	99
Backup Stations.....	100
Convert To Perpetual License Mode	100
HTML 5 Convert To Perpetual License Mode	101
HTML5 Backup Stations.....	101
Configure Niagara IdP and SAML Scheme	102
Copy Local File.....	104
Copy Supervisor File.....	104
HTML5 Copy Supervisor File	105
Deploy Template.....	106
HTML5 Deploy Template	106
Enable Bootstrap Mode.....	106

HTML5 Enable Bootstrap Mode	107
Export Application Template Configuration.....	107
HTML5 Export Application Template Configuration	108
Export Certificate Signing Request.....	108
HTML5 Export Certificate Signing Request	109
Update Connections Using Provisioning Station's DHCP Server.....	110
HTML5 Update Connections Using Provisioning Station's DHCP Server.....	110
For Each Station Stage	111
Generate Certificate	112
Import Signed Certificate	114
HTML5 Import Signed Certificate	115
Install Application Template	115
HTML5 Install Application Template.....	116
Install Certificate	117
HTML5 Install Certificate	117
Install Clean Distribution.....	117
Install Software.....	118
HTML5 Install Software	119
Reboot.....	120
HTML5 Reboot	120
Remove Property.....	120
HTML5 Remove Property	121
Remove Station User	121
HTML5 Remove Station User	122
Rename Device Station.....	122
HTML5 Rename Device Station.....	122
Run Robot.....	123
HTML5 Run Robot	123
Security Job Steps	124
HTML5 Security Job Steps	126
Set Certificate Alias.....	126
HTML5 Set Certificate Alias	127
Set Platform Credentials.....	127
HTML5 Set Platform Credentials	128
Set Platform User Password	129
HTML5 Set Platform User Password	129
Set Property	130
HTML5 Set Property	131
Set Station Connection Credentials.....	132
HTML5 Set Station Connection Credentials	133
Set Station User Password	134
HTML5 Set Station User Password	134
Set System Passphrase	135
HTML5 Set System Passphrase	135
Set TLS Level.....	136

HTML5 Set TLS Level	137
Set Time	137
HTML5 Set Time.....	139
Setup Reciprocal Connection.....	140
HTML5 Setup Reciprocal Connection.....	140
Sign Certificate (provisioningNiagara-SignCertificateStep).....	141
HTML5 Sign Certificate (provisioningNiagara-SignCertificateStep).....	142
Update Connections Using Network Discovery.....	143
HTML5 Update Connections Using Network Discovery	143
Update Licenses.....	144
HTML5 Update Licenses	144
Update Template or Application Configuration.....	145
HTML5 Update Template or Application Configuration	145
Upgrade Application Template	145
HTML5 Upgrade Application Template	145
Upgrade Out-of-date Software	146
HTML5 Upgrade Out-of-date Software	146
Upgrade Template	147
HTML5 Upgrade Template.....	147
Chapter 9 Plugins (views).....	149
Plugins in batchJob module	149
batchJob-BatchJobLogFileView	149
batchJob-BatchJobStepLogFileView	151
batchJob-JobStepDetailsView	155
batchJob-PrototypeJobList	158
Plugins in provisioningNiagara module	161
provisioningNiagara-BackupStepDetailsView.....	161
Network License Summary view (provisioningNiagara-NetworkLicenseSummary)	164
provisioningNiagara-NiagaraNetworkJobBuilder	165
HTML5 Niagara Network Job Builder.....	169
provisioningNiagara-NiagaraNetworkJobList	173
HTML5 Niagara Network Job List.....	174
Niagara Network Job view	176
HTML5 Niagara Network Job view	178
provisioningNiagara-NiagaraNetworkPrototypeView.....	180
HTML5 Niagara Network Prototype View.....	185
provisioningNiagara-ProvisioningManager.....	189
provisioningNiagara-ProvisioningRobotEditor	190
provisioningNiagara-ProvisioningStationDirector.....	191
provisioningNiagara-StationJobList	191
provisioningNiagara-StationSoftwareView	193
provisioningNiagara-SupervisorLicenseManager	194
provisioningNiagara-SupervisorSoftwareManager.....	195
Chapter 10 Troubleshooting.....	199

Why the Start Backup action is NOT recommended..... 199

Glossary 201

Index..... 203

About this Guide

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

Product Documentation

This document is part of the EC-Net™ technical documentation library. Released versions of EC-Net software include a complete collection of technical information that is provided in both online help and PDF format. The information in this document is written primarily for Systems Integrators. In order to make the most of the information in this book, readers should have some training or previous experience with EC-Net™ 4 or EC-Net^{AX}™ software.

Document Content

This guide describes how to set up provisioning jobs on a **NiagaraNetwork**. Topics included describe how to install and configure provisioning services and how to setup and manage provisioning jobs across the **NiagaraNetwork**. Also included are FAQs, component and plugin references, and troubleshooting information.

Document change log

This topic summarizes the releases of this document and the changes made at each release.

June 7, 2024

Minor corrections in “Configuring prototype job retention policy” topic.

As of EC-Net 4.14, updated the **Import Signed Certificate** job step by adding the **Use Filename as Alias** checkbox option.

June 20, 2023

Updated the “Update Connections Using Provisioning Station's DHCP Server”, “Provisioning in a mixed AX/N4 or N4.x network” and “Updating connections using a provisioning station's DHCP Server” for EC-BOS-9 reference.

March 10, 2022

- Added retention policy information and how to apply them to “Configuring prototype job retention policy” section.
- Updated the general backwards compatibility table of the 4.12 provisioning step “Set Platform User Password” in chapter “Provisioning in a mixed AX/N4 or N4.x network”.
- Added job step components for “Configure Niagara IdP and SAML Scheme” (4.9), “Copy Supervisor File”, “Remove Property” (4.12), and “Security job steps”. In addition, included “Removing a Property” in the “Housekeeping” chapter.
- Updated components, many screen captures and job steps.
- Added new job step “Set Platform User Password” and component “SetPlatformUserPasswordJobStep”.
- Added new topic “Device Bootstrap” in the “Components” chapter.

August 18, 2021

- Edited multiple topics in the process of documenting client certificate authentication.
- Updated “Job list step management.”
- Updated screen capture and graphics in “provisioningNiagara-NiagaraNetworkJobBuilder”.
- Updated “provisioningNiagara-NiagaraNetworkPrototypeView” with new Set Property job step.
- Added new control to view and edit the details of a selected step.

October 30, 2020

- Many edits throughout related to functional changes for EC-Net 4 v4.10.
- Included additional details on backwards compatibility of provisioning steps for EC-Net 4 v4.9.

December 3, 2019

Many edits throughout related to functional changes for EC-Net 4 v4.9 There are several added steps in Chapter 1 and an added component topic for each new step.

September 6, 2019

For clarification, edited the description of what this step does in the procedure, “Updating connections using provisioning station’s DHCP Server”.

June 11, 2019

Many edits throughout on changes in EC-Net 4 v4.8. These edits include the following:

- Replaced several figures throughout to show latest changes.
- Added topics on the provisioning steps: “Add Station User step”, “Install Clean Distribution step”, and “Remove Station User step”.
- Added new procedures on: “Installing a Clean Distribution File”, “Setting station connection credentials”, “Adding a station user”, Removing a station user”, and “Setting a property”.

April 11, 2019

Many minor edits throughout to reflect functional changes in EC-Net 4 v4.8, including updates to the following:

- Edited “Update License” job step and job progress indicator.
- Replaced many figures shown in Chapter 6: Plugins (views).

December 12, 2018

Updated topics “Software Installation step” and “Installing software using the add button” on the added filtering capability in EC-Net 4 v4.8 and later.

October 9, 2018

Many changes throughout including updating/adding topics describing provisioning steps and functional changes added in EC-Net 4 v4.7.

July 17, 2018

Revised the procedure, “Setting up and running a bulk provisioning job” and removed one other procedure.

April 30, 2018

Updated for template bulk provisioning

September 26, 2016

Minor revisions

August 19, 2015

Initial publication

Related documents

Following is a list of related guides.

- *EC-Net 4 Platform Guide*
- *Drivers Guide*
- *EC-Net Templates Guide*

Chapter 1 Provisioning overview

Topics covered in this chapter

- ◆ Provisioning FAQs
- ◆ Provisioning Installation
- ◆ Provisioning steps
- ◆ Provisioning extensions
- ◆ Provisioning-related alarms
- ◆ Provisioning in a mixed AX/N4 or N4.x network

Provisioning is available as a licensed feature of a station running on a Supervisor. Provisioning automates tasks to perform on remote hosts in the station's **NiagaraNetwork**. For the most part, these are platform tasks—that is—they would otherwise be done using EC-Net 4 Pro. Provisioning robot tasks allow the running of custom program code in the host station (executed by each station's **ProgramService**). The Supervisor station automatically performs these tasks, which are modeled in the station as provisioning jobs.

Outside of provisioning, you would have to perform similar tasks on each station using (full) EC-Net 4 Pro and one of the following methods:

- Making individual platform connections directly to remote hosts, then using the appropriate platform views.
- Making individual tunneled platform connections to remote hosts, then using the appropriate platform views.
- In the case of provisioning robots, by opening station connections and then copying and executing program objects.

For details about the platform user interface, see the *EC-Net 4 Platform Guide*.

Provisioning provides these advantages over individual platform (or station) connections:

- When provisioning, you need only one station connection—to the Supervisor, and no other connections (platform or otherwise). This means that you can run a provisioning job from any location where you can open the Supervisor station. Even using Web EC-Net 4 Pro (ordinary platform tasks cannot be done using Web EC-Net 4 Pro).
- Provisioning allows the same series of tasks (executed as job steps), to be run on any number of target hosts. Job steps execute sequentially on a single host. Provisioning is useful when performing the same tasks on multiple stations, such as when implementing a company-wide software upgrade, or a periodic backup of all hosts' station configurations. Starting in EC-Net 4 v4.7, it is possible to execute these jobs in parallel across stations. The minimum parallel job count depends on the Supervisor's hardware architecture (tied to the number of CPU cores of the host machine). The maximum parallel job count is 10. A specific value can be changed by setting the **Max Provisioning Threads** property on the **BatchJobService** component.
- By default, provisioning provides persistent storage of all jobs on the Supervisor, including all statistics associated with 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, which executes it as another provisioning job.

There are two types of provisioning jobs.

- The first is a job intended to be run only once, usually immediately. You build this type of provisioning job using the **Niagara Network Job Builder**, which is the default view accessed by double-clicking the **Niagara-Network's ProvisioningNwExt** component.
- The second type of job is one that is intended to be run on a regular basis, called a prototype job. This is the type of job you would set up to back up all station configuration data. A prototype job is linked to a **TriggerSchedule**, which specifies when it runs. A prototype job can also be run immediately.

In addition to setting up and running provisioning jobs, provisioning (device) extensions in each station require configuration as well as provide additional provisioning services, under each station modeled in **Drivers**→**NiagaraNetwork**.

Provisioning FAQs

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

What is meant by provisioning?

Provisioning allows an administrator, from a Supervisor station, to configure the automation of one or more pre-defined tasks against one or more (potentially many) stations in the Supervisor's **NiagaraNetwork**. A provisioning job records results, which you can refer to later.

Tasks would otherwise need to be performed manually by a user with EC-Net 4 Pro, often by making an individual platform connection to each host. All provisioning work is performed by the Supervisor station, as one or more provisioning jobs.

What type of pre-defined tasks can provisioning perform?

Among other things, a provisioning job can perform station backups, install software, update licenses, copy files, install and configure certificates, change default credentials, install templates, and configure properties.

Can a provisioning job run a custom program

Yes. You can write custom program code to execute as a provisioning robot. This feature makes it possible to change the station database programmatically.

Is there any other sort of provisioning besides “provisioning EC-Net”?

The information in this document applies only to the **niagaraDriver** for 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 Installation

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

Supervisor requirements:

- EC-Net 4 or later is installed on your Supervisor.
- The Supervisor must be licensed for provisioning. A licensed Supervisor has the following entry in its `[brand].license (license) file`:

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

- The following modules must be installed in the modules folder of the Supervisor PC.
 - `provisioningNiagara`
 - `batchJob`

Host controller requirements:

- Controllers should be running the same software release as the Supervisor. However, hosts running an earlier release than the Supervisor may be supported for most provisioning operations.
- Controllers do not require any provisioning-related module (`provisioningNiagara`, `batchJob`, `provisioning`), nor do they require a license feature for provisioning. They do require the `program` module and the **ProgramService** in their station to process provisioning jobs with Run Robot steps.

Adding the BatchJobService

The **BatchJobService** manages provisioning jobs.

Prerequisites: The Supervisor PC is licensed for provisioning and has the required modules installed.

- Step 1 If it is not already open, in EC-Net 4 Pro, open the Supervisor station.
- Step 2 Open the **provisioningNiagara** palette in the palette side bar.
- Step 3 Expand **Config** and double-click **Services**
The **Service Manager** opens.
- Step 4 Do one of the following:
 - If a **BatchJobService** is already listed, verify that it is enabled (status **ok**).
 - If no **BatchJobService** exists, from the palette, drag the **BatchJobService** onto the station's **Services** folder. In the popup **Name** window, rename the service—or use the default name and click **OK**.

A **BatchJobService** (or whatever you named it), appears under your **Services** folder.
- Step 5 Right-click the **BatchJobService** and click **Views**→**AX Property Sheet**.
- Step 6 Expand **Job Queue** and set the service's **Max Threads** property to a value greater than 1, say 2 or 3 and click **Save**.
This allows more than a single provisioning job to run concurrently.
- Step 7 To monitor provisioning alarms, change **Alarm Class** from the default `Default Alarm Class` to another alarm class

Adding the ProvisioningNwExt

The ProvisioningNwExt (Provisioning Network Extension) sets up the jobs to run on the remote stations.

Prerequisites: The Supervisor PC has been licensed for provisioning and has the required modules installed.

- Step 1 If it is not already open, in EC-Net 4 Pro, open the Supervisor station.
- Step 2 Open the **provisioningNiagara** palette in the EC-Net 4 Pro palette side bar.
- Step 3 Under the station's **Config** space, expand its **NiagaraNetwork**.
- Step 4 If a **ProvisioningNwExt** is already a child of the network, verify that it is enabled (status **ok**) and skip ahead to the next procedure.
- Step 5 From the palette, drag the **ProvisioningNwExt** to the station's **NiagaraNetwork**. In the popup **Name** window, you can rename the **ProvisioningNwExt**—or, use the default name and click **OK**.

You now have a **ProvisioningNwExt** (or whatever you named it), under your **NiagaraNetwork**. When you add this extension to a Supervisor station, every existing remote station under the network automatically receives five new device extensions. As you add more **NiagaraStations** to your network, they too have these extensions.

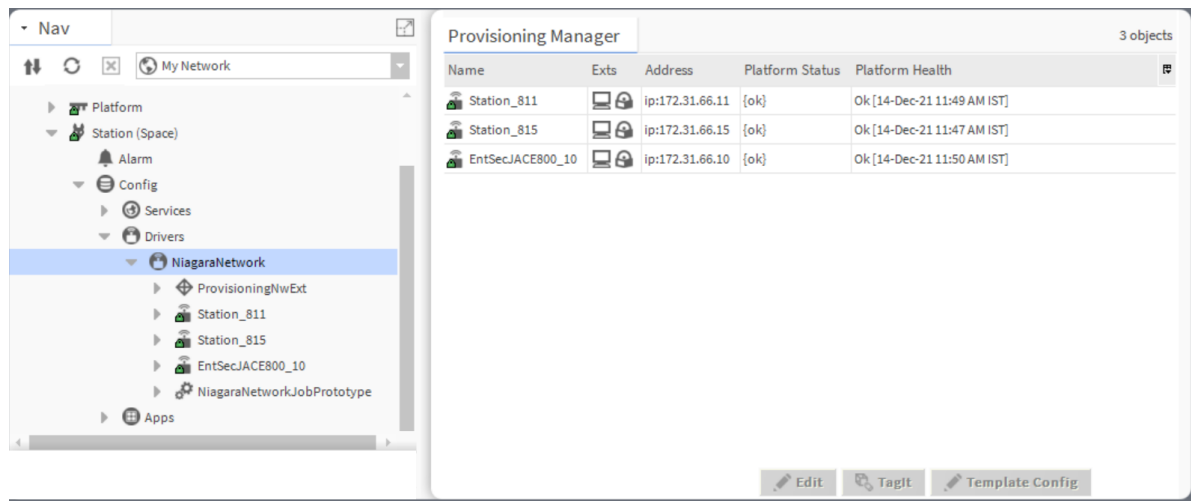
Configuring platform port and credentials

Provisioning requires a connection to each remote platform/station. For each **NiagaraStation** in the client station's **NiagaraNetwork**, you must confirm the port and enter the corresponding platform daemon credentials of its host. The simplest way to do this is by using either the **Station Manager** or **Provisioning Manager** views.

Prerequisites: All platform/stations to be provisioned are running. Each has its own signed server certificate recognized by the client station for server authentication.

- Step 1 In the Nav tree, expand **Config**→**Drivers**, right-click **NiagaraNetwork** and select **Views**→**Provisioning Manager**.

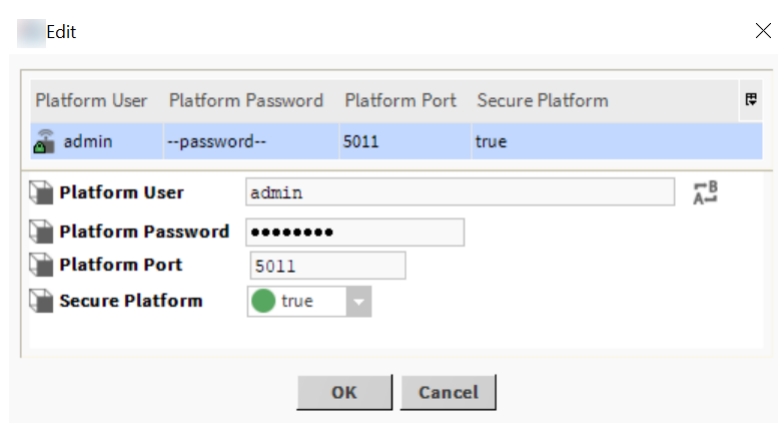
The **Provisioning Manager** opens.



Each row in the table represents a connection to a remote server platform/station.

Step 2 Select one or more (a group) of stations and click **Edit**.

The system opens an **Edit** window.



Step 3 Enter platform connection credentials, and, if using a non-default port address, change the port from the default port 5011 to the port number used and click **OK**.

These platform credentials and port should be the same credentials and port you use when you open a platform connection directly to each remote host.

Upon the next monitor ping of the host's platform daemon, its **Station Run State** should change from **Unknown** to **Running** and the station's **Platform Connection** extension should now have a status of **ok**.

Step 4 To confirm the platform connection immediately, right-click the station's **Platform Connection** extension and click **Action→Ping**.

Step 5 Confirm that the station, as shown in the **Provisioning Manager**, has a platform status of **ok**.

Step 6 To test the validity of the credentials, right-click the station row and click **Action→Ping**.

Repeat this procedure for each **NiagaraStation** listed in the **Station Manager**.

Installation troubleshooting

This topic provides suggestions to resolve error conditions.

Platform Status reports down and the remote station's Health reports Fail with the cause of certificate_unknown

The remote platform/station presented only a self-signed certificate to the client station, which cannot perform server authentication or another certificate error occurred, which was caused by the attempt to securely connect.

As a temporary measure, expand **Services**→**PlatformServices**, double-click **CertManagerService**, click the **Allowed Hosts** tab, select the unapproved certificate (its icon is a white X on a red shield), click **Approve** and click **Yes**. The red shield changes to a white check mark on a green shield.

The remote station's certificate should appear twice in the Supervisor station's **Allowed Hosts** tab, once to connect to the remote platform and the second instance to connect to the remote station.

Approving a certificate using **Allowed Hosts** provides only minimum communication security between client and server. The certificates in **Allowed Hosts** are self-signed by the remote controller station. They encrypt communicated data. But, approving them bypasses the client station's normal client certificate authentication. Rather than rely on approved self-signed certificates, each platform/station's certificate should be signed by a certificate authority recognized by the client station

Provisioning steps

Each provisioning job consists of one or more steps that the Supervisor's **JobService** controls and the remote station's **ProgramService** executes on the remote host. These steps include station backup, update and file copy, among others.

Figure 1 The New Job Step window

Each job runs one or more provisioning steps:

- **Add Station User** (component: `AddStationUserStep`) – adds a new user to the station.

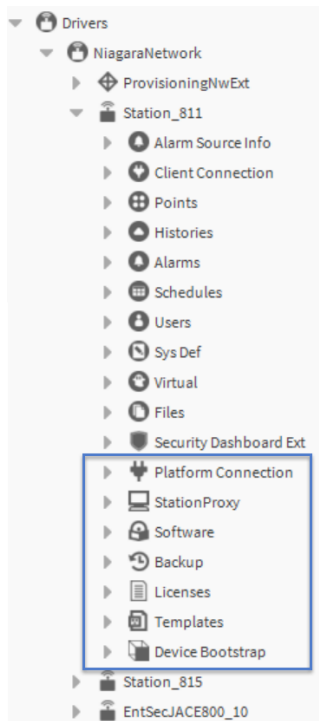
- **Backup Stations** (component: `BackupStationExt`) – makes an online backup of each running station.
- **Configure Niagara IdP and SAML Scheme** configures authentication properties.
- **Copy Local File** (component: `FileCopyStep`) – makes a copy of a local file on your PC.
- **Copy Supervisor File** (component: `FileCopyStep`) – makes a copy of a Supervisor file.
- **Deploy Template** (component: `DeployTemplateStep`) – deploy a component template to each station using an Excel file.
- **Enable Bootstrap Mode** (component: `EnableBootstrapStep`) – creates a one-time provisioning job to connect from the Supervisor to the remote controller using a certificate exemption for the default self-signed certificate. This step is intended for use with other provisioning steps that set up credentials and configure certificates. It is useful for provisioning new out-of-the-box devices.
- **Export Application Template Configuration** exports an application template configuration as an .xlsx file.
- **Export Certificate Signing Request** (component: `ExportCsrJobStep`) - generate and export a CSR for each remote host.
- **Generate Certificate** (component: `GenerateCertJobStep`) – generate a certificate on each remote host.
- **Import Signed Certificate** (component: `ImportSignedCertificateJobStep`) – import a signed certificate for each remote host.
- **Install Application Template** (component: `DeployApplicationStep`) – install an application template on each station using an Excel file.
- **Install Certificate** (component: `InstallCertificateJobStep`) – installs a certificate from the supervisor's user trust store to each remote host's user trust store.
- **Install Clean Distribution** (component: `InstallCleanDistStep`) – installs the selected clean distribution file to the target platforms.
- **Install Software** (component: `InstallBySpecStep`) – installs a software module in one or more remote hosts.
- **Reboot** (component: `RebootJobStep`) – reboots the platform host.
- **Remove Station User** (component: `RemoveStationUserStep`) – remove a named user from the station.
- **Rename Device Station** (component: `RenameStationStep`) – renames the station on the remote host to match the name in the supervisor's **NiagaraNetwork**.
- **Run Robot** (component: `RunRobotStep`) – allows you to create custom code to run on each station.
- **Security Job Steps** is a convenience step that configures and inserts the three credential-related steps in one sequence. The results are the same as if you added the steps for Set Platform Credentials, Set System Passphrase, and Set Station User Password individually.
- **Set Certificate Alias** (component: `SetCertificateAliasJobStep`) – set the certificate alias used for the remote device's platform, web, and fox services.
- **Set Platform Credentials** (component: `SetPlatformCredentialsJobStep`) – adds a new platform user account and remove the default account on the remote device.
- **Set Platform User Password** (component: `SetPlatformUserPasswordJobStep`) – changes password for an existing platform user.
- **Set Property** (component: `SetPropertyJobStep`) – set or add a property.
- **Set Station Connection Credentials** (component: `SetStationConnectionCredentialsStep`) – set the station credentials used to connect to the remote station.
- **Set Station User Password** (component: `SetStationUserPasswordJobStep`) – sets the station user password.
- **Set System Passphrase** (component: `SetSystemPassphraseJobStep`) - sets the system passphrase.

- **Set TLS Level** (component: `SetTlsLevelJobStep`) – set the minimum TLS level for the remote device's platform, web, and fox services.
- **Set Time** (component: `SetTimeJobStep`) - set the time on the remote device.
- **Setup Reciprocal Connection** (component: `SetupReciprocalConnectionStep`) – sets the connection settings for the "reciprocal" station connection, which is the station proxy in the **NiagaraNetwork** on the remote station that points back to the Supervisor.
- **Sign Certificate** (component: `SignCertificateStep`) signs a certificate (typically a server certificate) for each remote platform in the job.
- **Update Connections Using Niagara Network Discovery** (component: `NiagaraNetworkDiscoveryStep`) – updates station connections using **NiagaraNetwork** discovery.
- **Update Connections Using Provisioning Station's DHCP Server** (component: `DhcpDiscoveryStep`) – updates station connections using the provisioning station's DHCP server leases.
- **Upgrade Out-of-Date Software** (component: `SoftwareStationExt`) – installs software, upgrades out-of-date software, supports copy supervisor step, and the reboot step.
- **Update Template or Application Configuration** (component: `UpdateConfigurationStep`) updates the template configuration values of an installed template in each remote station.
- **Update Licenses** job step (component: `UpdateLicensesJobStep`) – updates software license in each remote host.
- **Upgrade Application Template** (`UpgradeApplicationStep`) upgrades the installed application template in each remote host.
- **Upgrade Out-of-Date Software** (`UpgradeOutOfDateStep`) compares module versions on the target hosts with the latest versions in the Supervisor's software database and installs updates as needed.
- **Upgrade Template** (component: `UpgradeTemplateStep`) - upgrades deployed component template instances on each station using an Excel file.

Provisioning extensions

This topic introduces the seven special device extensions that are automatically created under each of the **NiagaraStation** devices contained in its **NiagaraNetwork**. Provided the network has the **ProvisioningNwExt**, the system automatically includes these extensions when you add a new station under the Supervisor's **NiagaraNetwork**.

The provisioning extensions appear under a different area of the Supervisor station from the station's **ProvisioningNwExt**, and from the job prototype components separately copied from the **provisioningNiagara** palette.

Figure 2 Provisioning extensions added to NiagaraStation device

These seven device extensions are in addition to the standard extensions for points, histories, alarms, schedules, users, sys def and files that exist for any **NiagaraStation**.

Some of the extensions have special views. Along with the **ProvisioningNwExt** and its children, all station extension components are required to be present (and enabled) for full provisioning support.

The seven provisioning extensions are:

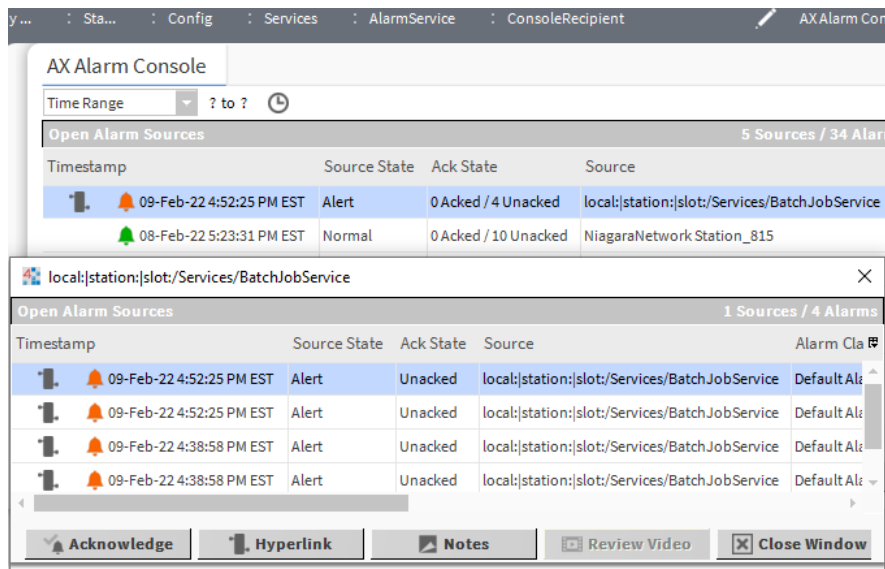
- **Platform Connection** manages the link between the Supervisor and the remote host running the station.
- **StationProxy** polls the station for system statistics.
- **Software** holds a snapshot of the current software versions installed on the remote host.
- **Backup** enables the Supervisor to make backups of the station.
- **Licenses** enables the Supervisor to update licenses on the host that is running the station.
- **Templates** supports the **Details** action, which will show information about component and device templates that have been deployed to the remote station.
- **Device Bootstrap** specifies the target host Id in bootstrap mode.

Provisioning-related alarms

For any provisioning job, you can configure it to generate an alarm either upon job failure, successful job completion, or both.


You configure alarms on the **Niagara Network Job Builder** and **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, which identifies the **BatchJobService** ORD as source (local:|station:|slot:/Services/BatchJobService).

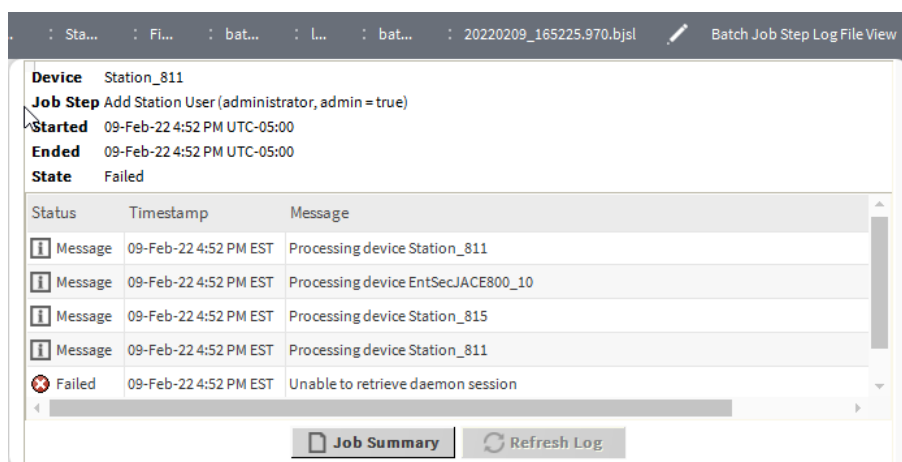
Figure 3 A provisioning alarm is an alert with source as BatchJobService

The single alarm source of the **BatchJobService** applies to all provisioning-related alarms (alerts). This means that only one row is used in the alarm console for all provisioning alerts. Keep this in mind if there are multiple alerts. When you click **Acknowledge** on the row, you are acknowledging all provisioning alerts.

To see multiple provisioning alerts in the **Alarm Details** view for the **BatchJobService**, double-click the row.

Each provisioning alert includes a hyperlink  for more detail. When you select (highlight) the alert and click the **Hyperlink** button, the view changes to either:

- **Batch Job Log File View** — For that job, in cases where the alert is raised for successful job completion.
- **Batch Job Step Log File View** — For a failed job step, in cases where the alert is raised for a job failure.

Figure 4 A Batch Job Step Log File View from the Hyperlink in the Alarm Console

If you disposed of a provisioning job before acknowledging an alarm it generated in the **Alarm Console**, and you click the **Hyperlink** button to view it from an associated Alarm Details view, the system advises, "Cannot Display Page." This happens because disposing of a job removes the batch job log (.bjl) file and all batch job step log (.bjsl) files associated with the job. Therefore, acknowledge any related alarms (alerts) before disposing of provisioning jobs.

Provisioning in a mixed AX/N4 or N4.x network

For a number of reasons, including the increased emphasis on security in EC-Net 4, limitations apply in installations where hosts running EC-Net^{AX}-3.8 and others running EC-Net 4 share the same network. Provisioning job steps continue to be added to EC-Net 4, and may not always be backward-compatible with previous versions.

Be aware that:

- The EC-Net 4 v4.7 Supervisor can provision the EC-Net^{AX}-3.8 station using these steps: Backup Stations, Copy Supervisor File, Install Software, Update Licenses, Set Time, and Reboot.
- The EC-Net^{AX}-3.8 Supervisor cannot provision EC-Net 4 stations.
- EC-Net 4 provisioning robots do not run on EC-Net^{AX}-3.8 stations.
- Component and device templates can be provisioned to stations running EC-Net 4 v4.6 and later
- Application templates can be provisioned to stations running EC-Net 4 v4.7 and later
- In EC-Net 4 v4.7 and later, provisioning no longer runs most job steps against a remote station if default platform credentials or default system passphrase are detected on the station. In that situation, only those steps that set platform credentials or set system passphrase will run.

The table below shows the general backwards compatibility of provisioning steps run on a EC-Net 4 v4.9 Supervisor against stations running prior EC-Net versions. Certain steps have some restrictions listed in the numbered notes following the table.

Job Step	EC-NetVers. Available	AX Support	Legacy EC-Net 4 Support	Fox/Plat
Enable Bootstrap Mode	4.7	Yes	Yes	Both
Update Connections Using Niagara Network Discovery	4.7	N/A	Yes 4.7 (9)	Both
Update Connections Using Provisioning Station's DHCP Server	4.7	N/A	Yes 4.7 (9)	Both
Update Licenses	AX/4.0	No (1)	No 4.4 (1)	Plat
Add Station User	4.8	No	Yes	Both
Backup Stations	AX/4.0	Yes	Yes	Both
Configure Niagara IdP and SAML Scheme	4.9	N/A	Yes 4.4 (6)	Both
Copy Local File	AX/4.0	Yes	Yes	Both
Copy Supervisor File	AX/4.0	Yes	Yes	Both
Deploy Template	4.6	N/A	Yes 4.6	Both
Export Certificate Signing Request	4.7	No	Yes	Both
Generate Certificate	4.6	No	Yes	Plat
Import Signed Certificate	4.6	No	Yes	Both
Install Application Template	4.7	N/A	Yes 4.7	Plat
Install Software	AX/4.0	Yes (2)	Yes (5)	Both
Reboot	AX/4.0	Yes	Yes	Plat
Remove Station User	4.8	Yes	Yes	Both
Rename Device Station	4.7	Yes (4)	Yes	Plat
Run Robot	AX/4.0	No	Yes	Both
Security Job Steps	4.7	N/A	Yes	Both
Set Certificate Alias	4.6	No	Yes	Both

Job Step	EC-NetVers. Available	AX Support	Legacy EC-Net 4 Support	Fox/Plat
Set Platform Credentials	4.6	N/A	Yes	Plat
Set Platform User Password	4.12	No	Yes	Plat
Set Property	4.8/4.9	Yes	Yes	Both
Set Station Connection Credentials	4.8	Yes	Yes	N/A
Set Station User Password	4.7	No	Yes (8)	Both
Set System Passphrase	4.6	N/A	Yes	Plat
Set TLS Level	4.6	No	Yes	Both
Set Time	4.7	Yes (3)	Yes	Both
Setup Reciprocal Connection	4.6	No	Yes	Both
Sign Certificate	4.6	No	Yes	Plat
Update Template or Application Configuration	4.9	N/A	Yes 4.6 (7)	Both
Upgrade Application Template	4.9	N/A	Yes 4.7	Both
Upgrade Out-of-date Software	AX/4.0	Yes (2)	Yes	Both
Upgrade Template	4.6	N/A	Yes 4.6	Both

1. This is prevented from running as expected. This is being addressed.
2. Yes, provided the EC-Net^{AX} software modules are available under the !_{SW} folder of the Supervisor and a reboot of the remote EC-Net^{AX} host is required.
3. NTP Host Mode is not a supported property in EC-Net^{AX} and is ignored.
4. An EC-Net^{AX} station cannot be restarted after being stopped without a host reboot step or manual reboot.
5. Yes, provided the legacy EC-NetTM 4 software modules are available under the !_{SW} folder of the Supervisor.
6. Requires the **saml** and **samlEncryption** modules installed on the remote host. You must add the station to desired the circle of trust under the **SAMLIdPService**.
7. Update Template Configuration is available for EC-Net 4 v4.6 and later. Application Templates are available from version EC-Net 4 v4.7 and later.
8. A EC-Net 4 v4.8 Supervisor works against EC-Net 4 v4.6 and later. A EC-Net 4 v4.9 Supervisor only works against EC-Net 4 v4.6.
9. Intended for use on aEC-BOS-9 acting as a Supervisor for connected Edge devices.

Chapter 2 Provisioning environment

Topics covered in this chapter

- ◆ One-time jobs
- ◆ Regularly-scheduled jobs
- ◆ Job list step management
- ◆ Creating a provisioning robot
- ◆ Provisioning Job management

The benefit of using a provisioning job is that you can run multiple steps against one or more remote hosts. A provisioning environment on a Supervisor provides these provisioning features the: creation of provisioning jobs to run a sequence of steps, monitoring the execution of these jobs on a set of devices, and reviewing job results.

There are two types of provisioning jobs:

- One-time provisioning jobs is for executing special, infrequent tasks, such as updating license files or software modules in selected (or all) remote hosts. A one-time provisioning job uses the **Niagara Network Job Builder** (the default view of the **ProvisioningNwExt**) to configure sequences of provisioning steps against one or more stations in a Supervisor's **NiagaraNetwork**.

Unlike prototype jobs, a one-time provisioning job cannot be saved as a component for reuse.

- Regularly scheduled provisioning jobs are especially useful for large enterprises with hundreds of remote hosts. For this type of job you create and configure a **NiagaraNetworkJobPrototype** component, which can be saved and reused. The job prototype allows you to configure job retention, which is not available when creating a one-time job. For job retention purposes you should run a backup stations step only using a prototype job.

Provisioning jobs run concurrently across multiple stations. By default, the number of parallel executions is derived based on the Supervisor's hardware platform.

One-time jobs

The **Niagara Network Job Builder** is the default view on the **ProvisioningNwExt** component. You start here to specify a one-time provisioning job needed now, meaning that you do not need to save it as a reusable template.

The **ProvisioningNwExt** component is in the **provisioningNiagara** palette.

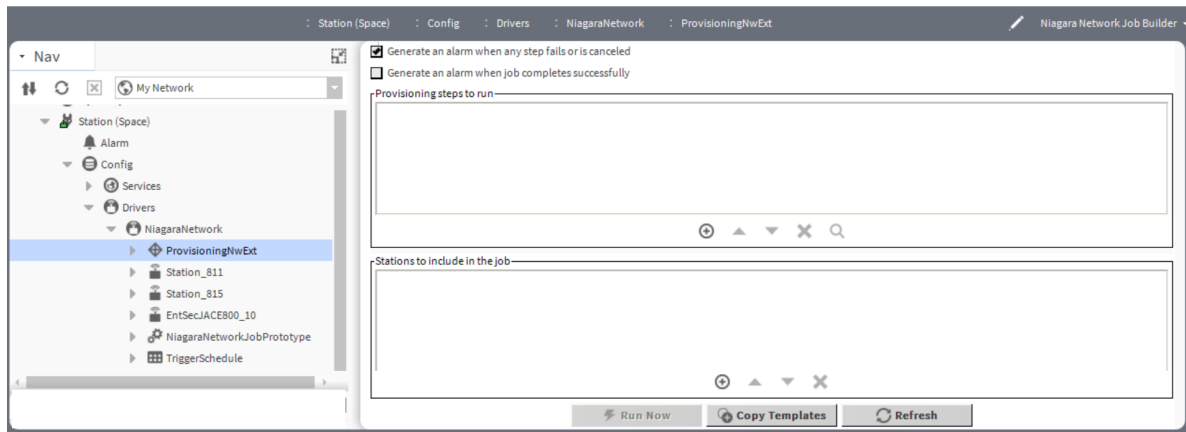
Creating a one-time job

This procedure demonstrates the general procedure used to create one-time provisioning jobs.

Prerequisites: The Supervisor is licensed for **provisioningNiagara**, the **BatchJobService** is available under **Services** folder and **ProvisioningNwExt** component is available under your **NiagaraNetwork**.

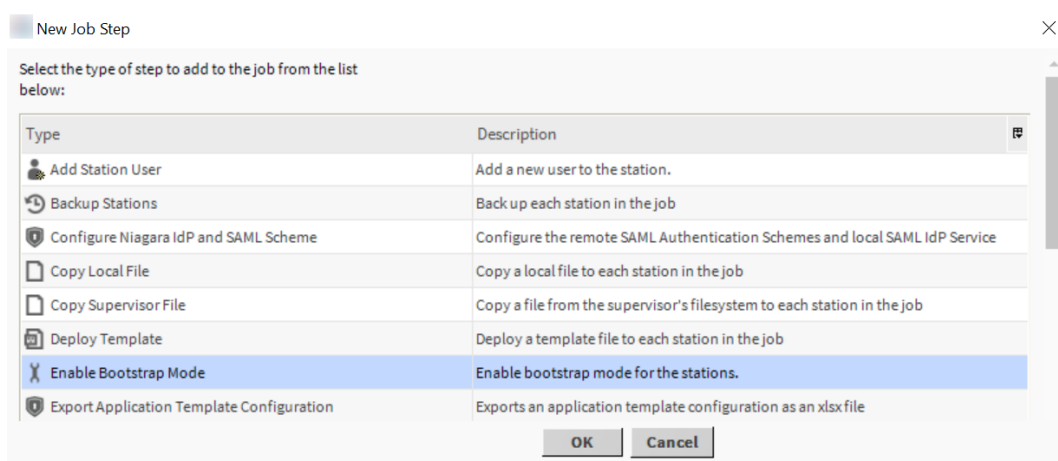
Step 1 Under the Supervisor station's **NiagaraNetwork**, double-click **ProvisioningNwExt**.

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



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

The **New Job Step** window opens.

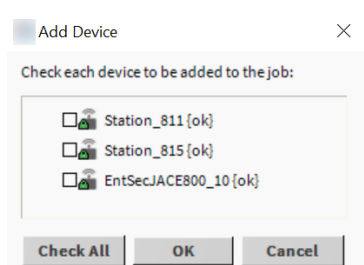


Step 3 Select the job step, for example, **Enable Bootstrap Mode** and click **OK**.

Step 4 To select the stations to provision, do one of the following:

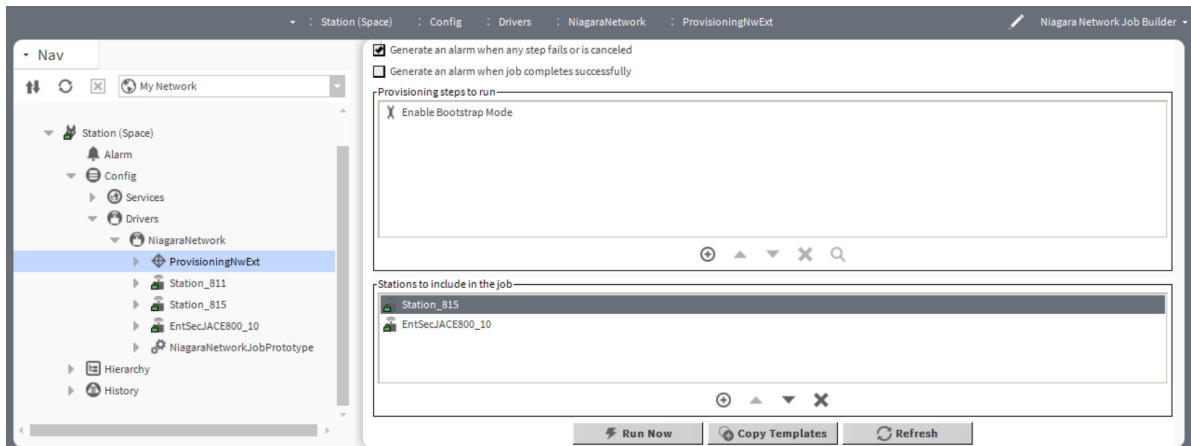
- Drag the stations from the **NiagaraNetwork** in the Nav tree one by one into the bottom pane, **Stations to include in the job**.
- Click add (+) or right-click in the **Stations to include in the job** and click Add.

If you clicked to add stations, the **Add Device** window opens.



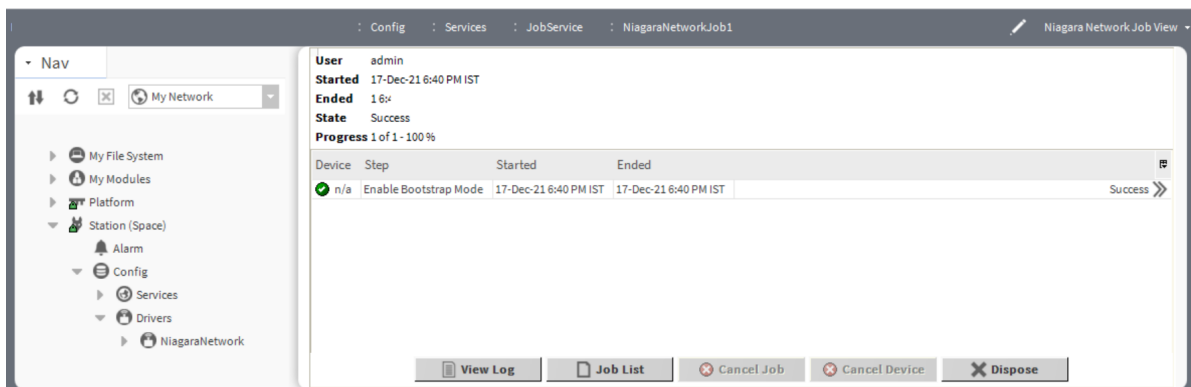
Step 5 Choose the station(s) or, to select them all, click **Check All** and click **OK**.

The system adds the job step and stations to the job builder.



Step 6 Review your choices and click **Run Now** at the bottom of the job builder.

The view changes to the **Niagara Network Job View**, where steps and results appear as the system executes them.



NOTE: Running this step by itself makes no persistent change on the target stations.

Adding one-time job list steps

These job list steps define the one-time tasks to be performed on one or more stations. The **BatchJobService** provisions each station following the sequence defined in the job list steps.

Prerequisites: The Supervisor is licensed for **provisioningNiagara**, the **BatchJobService** is available under **Services** folder and **ProvisioningNwExt** component is available under your **NiagaraNetwork**.

Step 1 Click the add button (+) below the list.

The system displays the **New Job Step** menu.

Step 2 Choose the step and click **OK**.

The same types of job steps are available as in the **Niagara Network Prototype View**.

Step 3 Select the copy action: **Copy Local File** or **Copy Supervisor File**.

Step 4 Right-click in the steps list, select **Add**, and choose the step type from the menu.

Step 5 Drag a file from the Nav tree into the steps list (implicit File Copy step).

Step 6 Drag a software item (module or .dist) from the Nav tree that appears under the **ProvisioningNwExt's Software** container, into the **Job Steps List** (implicit Install Software step).

Step 7 Drag a **ProvisioningRobot** that exists in the station's **Config** (component) architecture into the **Job Steps List** (implicit Run Robot step).

Updating connections using host name or IP address

This procedure creates a one-time provisioning job to update the Supervisor-to-station connections using a **NiagaraNetwork** discovery function. It runs a **Station Discovery** job on the Supervisor. If the stations are not already present under the **NiagaraNetwork**, the job automatically adds them.

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

Step 1 In the Supervisor station's **NiagaraNetwork**, double-click **ProvisioningNwExt**.

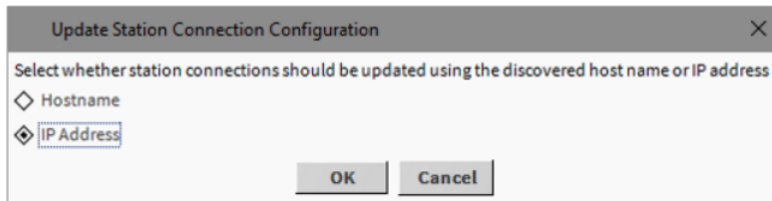
The **Niagara Network Job Builder** window opens.

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

The **New Job Step** window opens

Step 3 Select the **Update Connections Using Niagara Network Discovery** step and click **OK**.

The system prompts for the selection of the **Hostname** or **IP Address** option for connection updates.



Step 4 Select an option and click **OK**.

The system returns to the **Niagara Network Job Builder** window.

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

The **Add Device** window opens.

Step 6 Select the stations and click **OK**.

The **Niagara Network Job Builder** view displays your choices.

Step 7 To initiate the provisioning job, review your choices and click **Run Now** at the bottom of the view.

The view changes to the **Niagara Network Job View**, where steps and results appear as the system executes them.

Updating connections using a provisioning station's DHCP Server

This procedure creates a one-time provisioning job to update the EC-BOS-9-to-station connections using information pulled from the DHCP server running on the EC-BOS-9 provisioning station (the Supervisor). It updates the IP address and port information for Fox and Platform connections based on the discovery results. It is intended to be used in installations where a EC-BOS-9 is acting as a Supervisor for a group of Edge devices connected through the secondary network port.

Prerequisites: The **BatchJobService** is available under the **Services** folder and the **ProvisioningNwExt** component is available under your **NiagaraNetwork**. The **Niagara Network Job Builder** (for a one-time job) or the **Niagara Network Prototype View** (for a prototype job) is open.

NOTE: Only Edge devices respond to this request in a **NiagaraNetwork**. Controller devices do not respond to these requests.

Step 1 In the **Niagara Network Job Builder** window, top pane, **Provisioning steps to run**, click add (+).

The **New Job Step** window opens.

Step 2 Click the **Update Connections Using Provisioning Station's DHCP Server** step and click **OK**.

- Step 3 In the bottom pane, **Stations to include in the job**, click add (+).
- The **Add Device** window opens.
- Step 4 Select the stations and click **OK**.
- The **Niagara Network Job Builder** view reports your choices.
- Step 5 To initiate the provisioning job, review your choices and click **Run Now** at the bottom of the view.
- The view changes to the **Niagara Network Job View**, where steps and results appear as they are executed.

Setting the reciprocal connection

This procedure creates a provisioning job to set up the reciprocal connection to enable the station-to-Supervisor Fox connection under the **NiagaraNetwork** of each remote station.

Prerequisites: The **BatchJobService** is available under the **Services** node and **ProvisioningNwExt** component is available under your **NiagaraNetwork** in the Nav tree. The **Niagara Network Job Builder** (for a one-time job) or the **Niagara Network Prototype View** (for a prototype job) is open.

- Step 1 In the top pane, **Provisioning steps to run**, click add (+).
- The **New Job Step** window opens.
- Step 2 Select the **Set Reciprocal Connection** step and click **OK**.
- The system opens the **Setup Reciprocal Niagara Network Connection** window, which prompts for the Supervisor connection information.

- Step 3 Enter the credentials to use for the station-to-Supervisor connection, and click **OK**.
- Step 4 In the bottom pane, **Stations to include in the job**, click add (+).
- The **Add Device** window opens.
- Step 5 Click to select the stations and click **OK**.
- The **Niagara Network Job Builder** view reports your choices.
- Step 6 To initiate the provisioning job, review your choices and click **Run Now** at the bottom of the view.
- The view changes to the **Niagara Network Job View**, where steps and results appear as they are executed.

Regularly-scheduled jobs

For this type of job, you create and configure a **NiagaraNetworkJobPrototype** component, which can be saved and reused. A prototype job allows you to configure job retention, which is not available when creating a one-time job.

This component is in the **provisioningNiagara** palette. From it you can copy a **NiagaraNetworkJobPrototype** anywhere in the Supervisor station and use it to schedule, save, duplicate and modify prototype jobs.

Creating a prototype job

The benefit of creating a prototype job is that you can specify various provisioning steps against one or more stations in the **NiagaraNetwork** and save the set of steps as normal persisted components. You can duplicate and edit them as needed. Each **NiagaraNetworkJobPrototype** provides a default **Niagara Network Prototype View** for managing job steps. A prototype job is the recommended method for regular, scheduled backups of networked controllers.

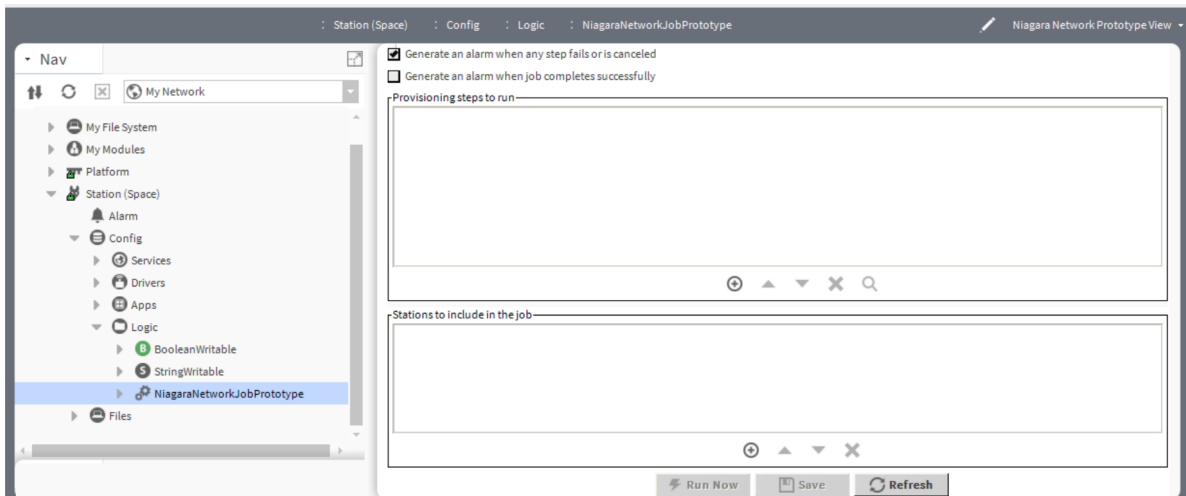
Prerequisites: The Supervisor PC is licensed for **provisioningNiagara**, the **BatchJobService** is available in the **Services** folder and the **provisioningNiagara** palette is open.

Step 1 Copy a **NiagaraNetworkJobPrototype** component from the **provisioningNiagara** palette into the Supervisor station.

You may locate **NiagaraNetworkJobPrototype** component anywhere in the architecture of the station database. It does not need to be under the **NiagaraNetwork**.

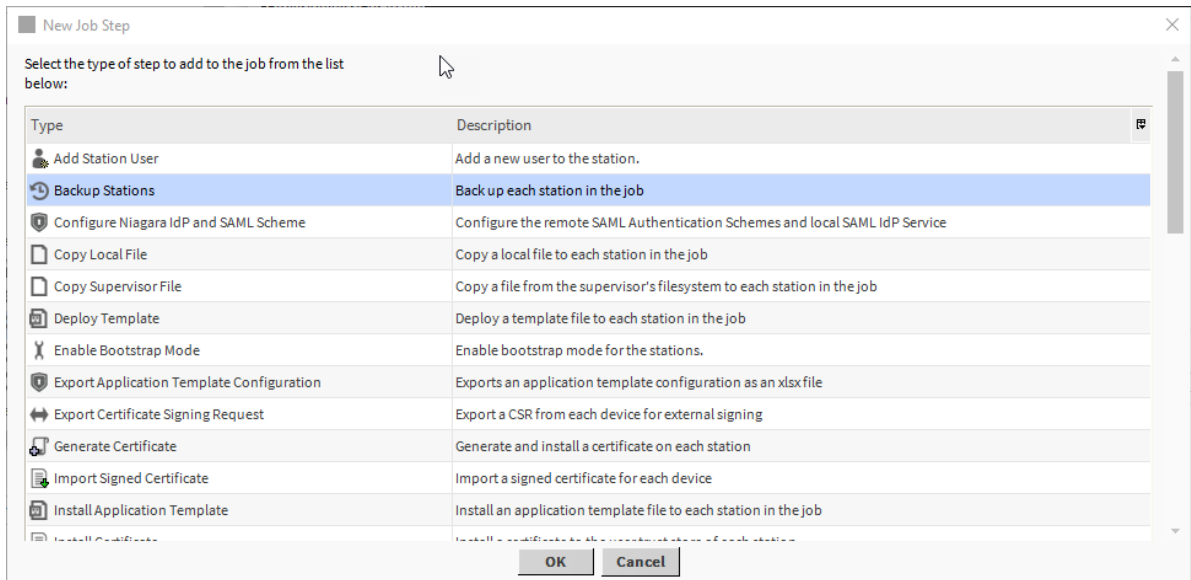
Step 2 To open its **Niagara Network Prototype View**, double-click the component.

The **Niagara Network Prototype View** opens.



Step 3 In the top pane, **Provisioning steps to run**, click add (+) or right-click in the **Provisioning steps to run** pane and click Add.

The **New Job Step** window opens.

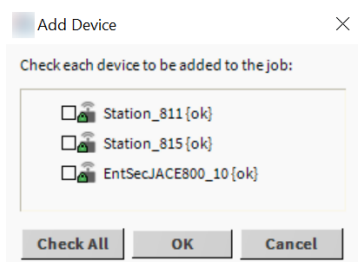


Step 4 Select the job step, for example, Backup Stations and click **OK**.

Step 5 To select the stations to regularly provision, do one of the following:

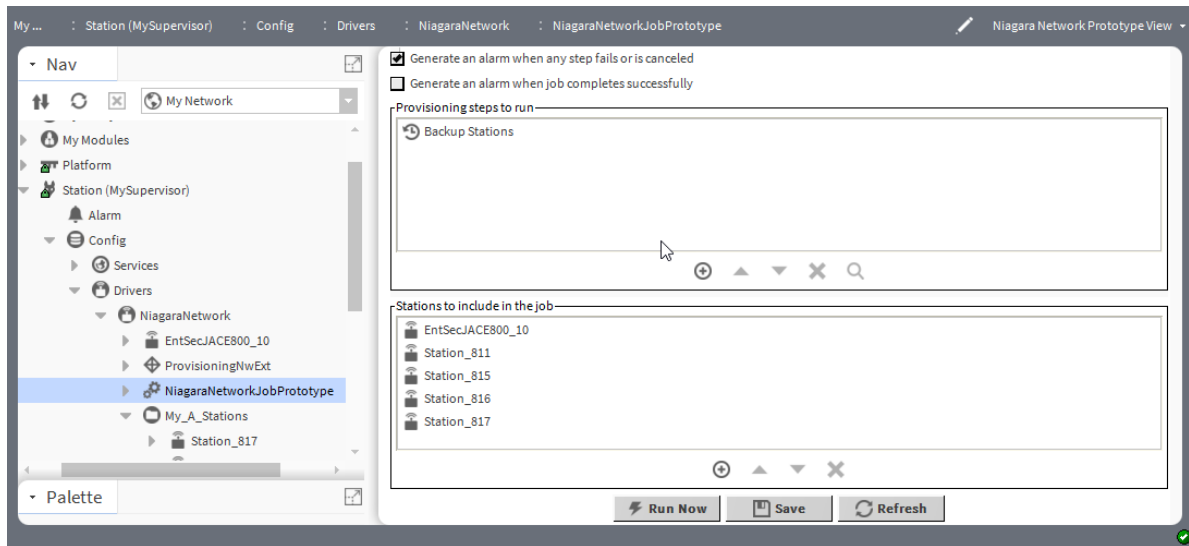
- Drag the stations from the **NiagaraNetwork** in the Nav tree one by one into the bottom pane, **Stations to include in the job**.
- Click add (+) or right-click in the **Stations to include in the job** and click Add.

If you clicked to add stations, the **Add Device** window opens.



Step 6 Choose the station(s) or, to select them all, click **Check All** and click **OK**.

The system adds the job step and stations to the job builder.



The prototype job is ready for the **BatchJobService** to provision each station following the sequence defined in the job list steps.

Configuring prototype job retention policy

Among the various types of batch job components, the ability to set up the disposal of information related to a previously executed job is unique to a job prototype component. When a batch provisioning job is disposed of, any associated files stored on the Supervisor are deleted, and the job removed from the various job list views.

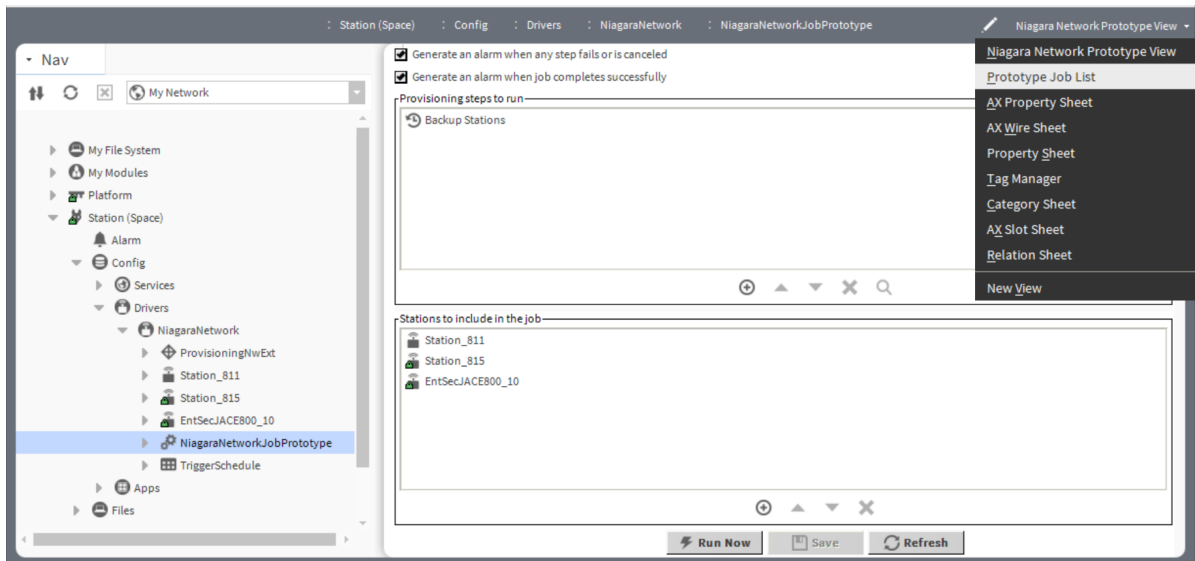
Disposal is especially important for jobs with backup steps, each of which results in an associated backup .dist file to be stored for each station. Although a Supervisor platform has large amounts of hard disk storage, over time, retaining all .dist files can consume huge amounts of file space. As a best practice, you should periodically back up all files to removable media and set the job retention to automatically delete old backups.

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 of). However, you can (and often should) modify a job prototype's retention policy such that some executed jobs are automatically disposed of—depending on either elapsed time or by reaching some number of job executions.

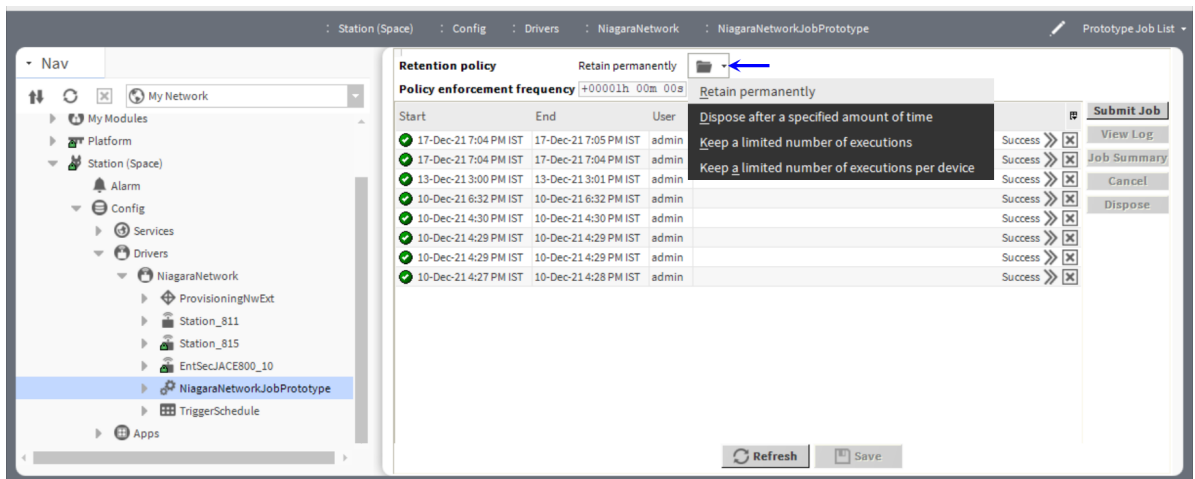
You can access the job retention properties from either the **Prototype Job List** view (this procedure) or from the component's property sheet.

Step 1 Double-click the **NiagaraNetworkPrototypeJob** container.

The system displays the **Niagara Network Prototype View**.



Step 2 Click the drop-down list in the upper right corner of the view and click **Prototype Job List**.
The system displays the **Prototype Job List**.



Step 3 Click the file folder to the right of the **Retention policy** property and select the retention policy.

- Retain permanently
- Dispose after a specified amount of time (default is 7 days)
- Keep a limited number of executions (default is 10) with Count only successful executions **deselected or selected**.
- Keep a limited number of executions per device
Successful executions to keep per device (defaults to 5)
Unsuccessful executions to keep per device (defaults to 1)

To understand how to apply the different retention policies, take a look at the following information:

Retain permanently: All successful and unsuccessful job executions are retained.

Dispose after a specified amount of time: All successful and unsuccessful job executions are disposed of after a defined period of time.

Keep a limited number of executions and **deselect** Count only successful executions: This policy keeps a defined number of job executions regardless of their status, causing the files for the latest successful backups of certain stations to be disposed of when the limit is reached.

Keep a limited number of executions and **select** Count only successful executions. This policy keeps a defined number of job executions with a successful job status, causing files for a large number of job executions to be retained.

Keep a limited number of executions per device and **define the number of Successful** executions to keep per device and **Unsuccessful** executions to keep per device: This policy keeps per station the specified number of successful and unsuccessful job executions. Only when in each station the status limits are exceeded, the job is safe to be disposed of. If the status limit for even one job step is under its limit for its station, the job is retained.

Example: Keep by device, three successful executions and two unsuccessful executions

Order of Retention Policy Processing ↑	Run	Station A			Station B			Station C			Retained or Disposed?
		Step Status	#Success for Station	#Fail for Station	Step Status	#Success for Station	#Fail for Station	Step Status	#Success for Station	#Fail for Station	
	Apr 01	Success	10	3	Success	10	3	Success	4	10	Disposed
	Apr 03	Success	9	3	Success	9	3	Success	3	10	Retained
	Apr 05	Success	8	3	Success	8	3	Success	2	10	Retained
	Apr 07	Success	7	3	Success	7	3	Success	1	10	Retained
	Apr 09	Success	6	3	Success	6	3	Failed	0	10	Disposed
	Apr 11	Success	6	3	Success	6	3	Failed	0	10	Disposed
	Apr 13	Success	6	3	Success	6	3	Failed	0	10	Disposed
	Apr 15	Success	5	3	Success	4	4	Failed	0	8	Disposed
	Apr 17	Success	4	3	Success	3	4	Failed	0	7	Retained
	Apr 19	Failed	3	3	Success	2	4	Failed	0	6	Retained
	Apr 21	Success	3	2	Failed	1	4	Failed	0	5	Retained
	Apr 23	Success	2	2	Success	1	3	Failed	0	4	Retained
	Apr 25	Success	1	2	Failed	0	3	Failed	0	3	Retained
	Apr 27	Failed	0	2	Failed	0	2	Failed	0	2	Retained
	Apr 29	Failed	0	1	Failed	0	1	Failed	0	1	Retained

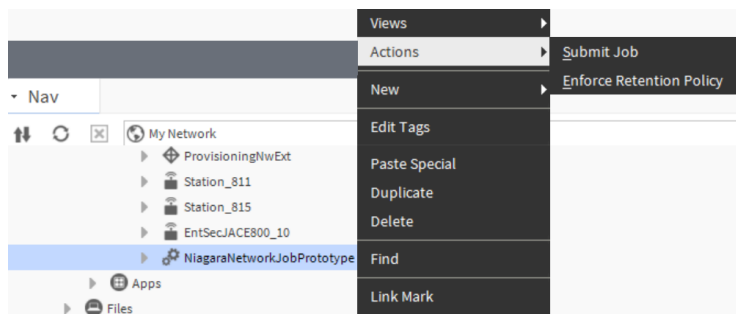
In the right column, the shaded entries indicate that the status limits for all steps in the job are exceeded, and the job is safe to be disposed of (Apr 15). If the status for even one step in a job is at or under its limit for its station, the entire job is retained (Apr 25).

NOTE: Keep in mind that if you configure retention properties and then duplicate the job prototype for the purpose of creating a new provisioning job, all copied job components will have that same retention configuration.

Prototype job actions

Each job prototype (provisioning job) has two available right-click actions.

Figure 5 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.

Setting up the prototype job schedule

Although you can manually run a provisioning job, most provisioning jobs run automatically based on a trigger schedule. This example procedure demonstrates how to configure each of the three job prototypes to backup a total of 12 different stations per provisioning job. This includes adjusting 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.

Prerequisites: All stations are properly configured for platform connections.

- Step 1 To go to the **Trigger Schedule** view, double-click the linked **TriggerSchedule** 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.
- Step 2 Click to **Save** this schedule configuration.
Later, after duplicating this component and re-configuring, you will select a slightly different off-hours time.
- Step 3 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.
- Step 4 Move the copied linked pair to a new spot on the wire sheet, and repeat this procedure again.

Job list step management

You can remove, reorder, review and edit job list steps.

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.

Reordering job list steps


You may reorder a selected job step using either of these two methods.

- Click the (up) or (down) arrow button at the bottom of the list.
- Right-click a job step and click **Move Up** or **Move Down**, as needed.

The **JobService** executes job steps from top to bottom in the order that you define them in the two step lists: initial steps first, then steps for each station.

Reviewing and editing job list steps

You can use any of these three methods to review and edit the details of steps that require additional information:

- Click to select the step, then click the spyglass button () below the list.
- Right-click the step and select **View Step Details** from the popup menu.
- Double-click the step.

NOTE: Only steps that require additional information support review and editing. Other steps report an error message if you attempt to review or edit them.

Creating a provisioning robot

Using a provisioning robot you can customize a provisioning job to perform steps beyond the built-in steps provided. To successfully customize, you should already be familiar with the Baja class structures and methods, as well as the Java syntax used by Baja program code.

Prerequisites: You must have super user permissions to add or edit the Program and Robot components, including ProvisioningRobots. The `niagara.program.requireSuperUser` entry in the Supervisor's `system.properties` file controls this requirement.

Step 1 Copy the **ProvisioningRobot** component from the **provisioningNiagara** palette into the Supervisor station.

NOTE: The program code in the **ProvisioningRobot** includes several helpful programming tips within the remark lines near the top of the code.

Step 2 Modify the robot's program code as appropriate.

Step 3 Click the Compile and Save button (🔧).

The system saves your changes.

Adding a robot step to a provisioning job

The **Run Robot** step adds an existing robot to a provisioning job for running on each remote host.

Prerequisites: A **ProvisioningRobot** customized to perform a task (or tasks) that specifically apply to all stations included in the provisioning job exists. The station user in any remote host used by the Supervisor for client access (fox) must be a super user for such a provisioning job to run successfully on the station.

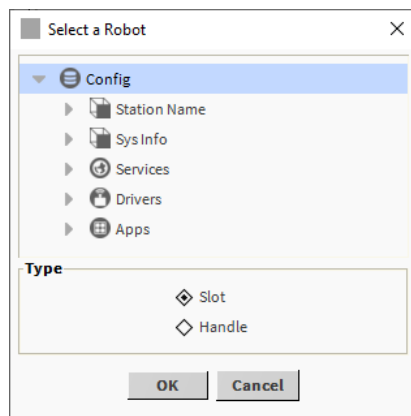
Super user permissions are not required to configure a provisioning job that includes a Run Robot step, where the referenced **ProvisioningRobot** was previously added or edited by a super user.

Step 1 In the top pane, **Provisioning steps to run**, of the **Niagara Network Job Builder** or **Niagara Network Prototype View**, click add (+).

The **New Job Step** window opens.

Step 2 Select the **Run Robot** step and click **OK**.

The **Select a Robot** window opens.



Step 3 Navigate to the robot, select it and click **OK**.

As a simple test, select the **ProvisioningRobot** itself (as copied from the **provisioningNiagara** palette). It should execute without errors on any target station with the **ProgramService**.

Step 4 In the bottom pane, **Stations to include in the job**, click add (+).

The **Add Device** window opens.

Step 5 Select the stations and click **OK**.

Step 6 To initiate the provisioning job, review your choices and click the **Run Now**.

The **ProgramService** of each station in the provisioning job runs the robot. The view changes to the **Niagara Network Job View**, where steps and results appear as they are executed.

Provisioning Job management

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.

Batch jobs are different from other jobs run by the Supervisor's **JobService**, because:

- The 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 deleted.
- 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.

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. **NiagaraNetwork** provisioning this means that the system runs the Update Licenses step (if included) first. 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) to the licensing server. 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 *EC-Net 4 Platform Guide* sections.

Following this, the provisioning batch job executes the remaining steps in sequence for each station, working through its list of stations in sequence. When the job reaches a station in the list, its station state reports **Running**. If any step fails, the station's state reports **Failed**, 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, the station state reports **Success**. If the job is canceled during a station step, the station state and that of all following stations in the list report **Canceled**.

When all steps are complete for all stations without canceling, and all steps complete successfully, the job state reports **Success**. However, if even one step failed, the job state reports **Failed**.

The **Progress Indicator** will show the number of job steps completed, total number of expected steps and the percent complete. Note that the total step count can change while the job runs due to automated step combination (e.g. for software installation steps), automatic step creation (for license updates), and for step failures (where subsequent steps will not be run).

Provisioning data files

Provisioning creates a number of files on the Supervisor PC.

Two types of files back up provisioning jobs:

- Batch job log files and batch step log files are binary records of each completed provisioning batch job and step. The system writes these files under the Supervisor's **Station** directory using the following convention:
`^batchJob/logs/batchJob_type/timestamp.bjl (or .bjsl)`
- Station files are .dist files and software snapshots. The system writes these files under the station's **Files** node.

To see the station files, station users require admin-level Read (R) permissions on a category assigned to the **provisioningNiagara** subfolder.

Backup .dist files are written under the Supervisor's **Station** directory using the following convention:

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

- Snapshots of the software installed on the hosts running the stations are written under the Supervisor's **Station** directory using this convention:

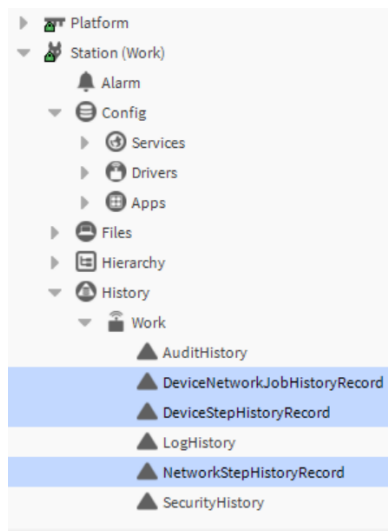
```
^provisioningNiagara/stationData/station/software/snapshot.bog
```

CAUTION: Do not manually delete any provisioning files from the Supervisor PC, for example, by using Windows Explorer, or in EC-Net 4 Pro by using the **My File System** node in the Nav tree. Instead, use the **Dispose** button on various provisioning views. This button is on the **Niagara Network Job List** and **Prototype Job List** view. When you dispose of a job, the system automatically deletes all its related batch job log files and provisioning station data files.

Histories related to provisioning

The system automatically records provisioning jobs within the history space of the Supervisor.

Figure 6 Provisioning related histories in Supervisor's history space



The system automatically creates these histories using the following names:

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

Chapter 3 Housekeeping

Topics covered in this chapter

- ◆ Adding a station user
- ◆ Removing a station user
- ◆ Setting station user password
- ◆ Automating host license updates using provisioning
- ◆ Copying a file
- ◆ Copying a Supervisor file
- ◆ Rebooting the system
- ◆ Removing stations
- ◆ Renaming remote stations
- ◆ Reordering stations
- ◆ Setting a property
- ◆ Removing a property
- ◆ Setting the time

This collection of procedures apply to mostly one-off procedures you may require from time to time.

Adding a station user

This procedure creates a provisioning job to add a new user to the UserService in each station.

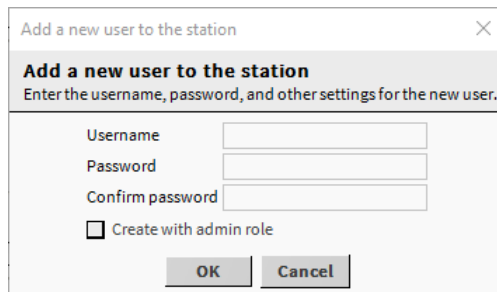
Prerequisites: The **BatchJobService** is available under **Services** and **ProvisioningNwExt** component is available under your **NiagaraNetwork**. Either the **Niagara Network Job Builder View** (for a one-time job) or the **Niagara Network Prototype View** (for a prototype job) is open.

Step 1 In the top pane, **Provisioning steps to run**, click add (+).

The **New Job Step** window opens.

Step 2 Click the default **Add Station User** step and click **OK**.

The **Add a new user to the station** window opens.



A password must be supplied for the new user.

Step 3 Enter the **Username** and **Password** for the new user, if the user will have the administrator role, select the **Create with admin role** check box, and click **OK**.

NOTE: If the check box is not selected, the job creates the user with no role assigned.

Step 4 In the bottom pane, **Stations to include in the job**, click add (+).

The **Add Device** window opens.

Step 5 Select the stations and click **OK**.

- Step 6 Review your choices and click **Run Now** at the bottom of the **Niagara Network Job Builder** view to initiate the provisioning job.

The view changes to the **Niagara Network Job View**, where steps and results appear as they are executed. This step adds a named user to the **UserService** for each station in the provisioning job.

Removing a station user

This procedure creates a provisioning job to remove a user from the User Service on each station in the job.

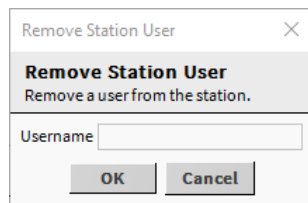
Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under your **NiagaraNetwork**. Either the **Niagara Network Job Builder View** (for a one-time job) or the **Niagara Network Prototype View** (for a prototype job) is open.

- Step 1 In the top pane, **Provisioning steps to run**, click add (+).

The **New Job Step** window opens.

- Step 2 Click the **Remove Station User** step and click **OK**.

The **Remove Station User** window opens.



- Step 3 Enter the **Username** of the user to be removed from the stations, and click **OK**.

- Step 4 In the bottom pane, **Stations to include in the job**, click add (+).

The **Add Device** window opens.

- Step 5 Select the stations and click **OK**.

- Step 6 To initiate the provisioning job, review your choices and click **Run Now** at the bottom of the view.

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

Setting station user password

This procedure creates a provisioning job to set the user credentials for the existing Fox connection on each remote device station in the job. It also updates the user credentials in the corresponding station proxy in the Supervisor's **NiagaraNetwork**.

Prerequisites: The **BatchJobService** is available under the **Services** node and the **ProvisioningNwExt** component is available under your **NiagaraNetwork** in the Nav tree. The **Niagara Network Job Builder** (for a one-time job) or the **Niagara Network Prototype View** (for a prototype job) is open. You have the credentials for the platform connection to the remote device.

- Step 1 In the **Niagara Network Job Builder** window in the top pane, **Provisioning steps to run**, click add (+).

The **New Job Step** window opens.

- Step 2 Click the **Set Station User Password** step and click **OK**.

The **Set Platform Credentials** window prompts for the new username and password for the platform connection to the remote device.



Set Station User Password

Enter the new password for the current station user

☒ Use connection username

Username

Current Password

New Password

Confirm Password

OK Cancel

Step 3 Enter the current and new username and password used for the Fox connection to the remote station, and click **OK**.

NOTE: To change the existing Fox connection user, deselect the check box and enter the new **Username**.

Step 4 In the bottom pane, **Stations to include in the job**, click add (+).
The **Add Device** window opens.

Step 5 Select the stations and click **OK**.

Step 6 To initiate the provisioning job, review your choices and click **Run Now** at the bottom of the **Niagara Network Job Builder** view.
The view changes to the **Niagara Network Job View**, where steps and results appear as they are executed.

Automating host license updates using provisioning

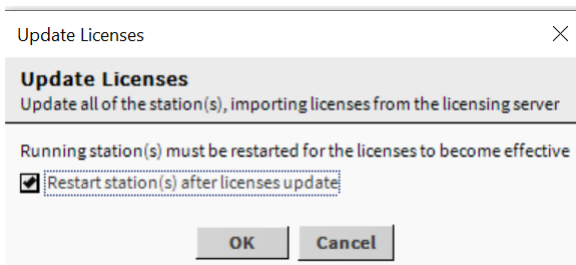
This procedure uses the **Niagara Network Job Builder** to create a one-time provisioning job to automate updating the license in one or more host controllers.

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

Step 1 Double-click **ProvisioningNwExt**.
The system opens the **Niagara Network Job Builder** view.

Step 2 In the top pane, **Provisioning steps to run**, click add (+).
The **New Job Step** window opens.

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



Update Licenses

Update Licenses

Update all of the station(s), importing licenses from the licensing server

Running station(s) must be restarted for the licenses to become effective

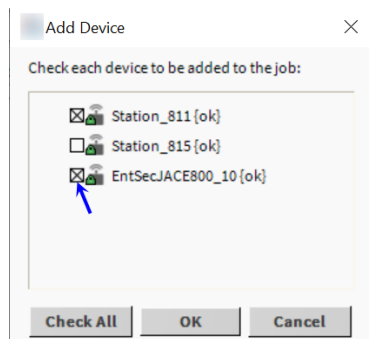
☒ Restart station(s) after licenses update

OK Cancel

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

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

The **Add Device** window opens.



- Step 6 To select the stations to upgrade, click one or more of the boxes next to the station names and click **OK**.
- Step 7 To initiate the provisioning job, review your choices and click **Run Now** at the bottom of the **Niagara Network Job Builder View**.


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

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

Synchronizing with the Supervisor's license database

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

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

- Step 1 Expand the **Config**→**Drivers**→**NiagaraNetwork**→**ProvisioningNwExt** in the Nav tree to see its **Licenses** node. 
- Step 2 Right-click **Licenses** and select **Views**→**Supervisor License Manager**.
The **Supervisor License Manager** window opens.
- Step 3 Click **Synchronize**.
The system prompts with the option to **Synch All Licenses?**
- Step 4 Click **Yes** and, at the **Synchronization Complete** prompt, click **OK**.

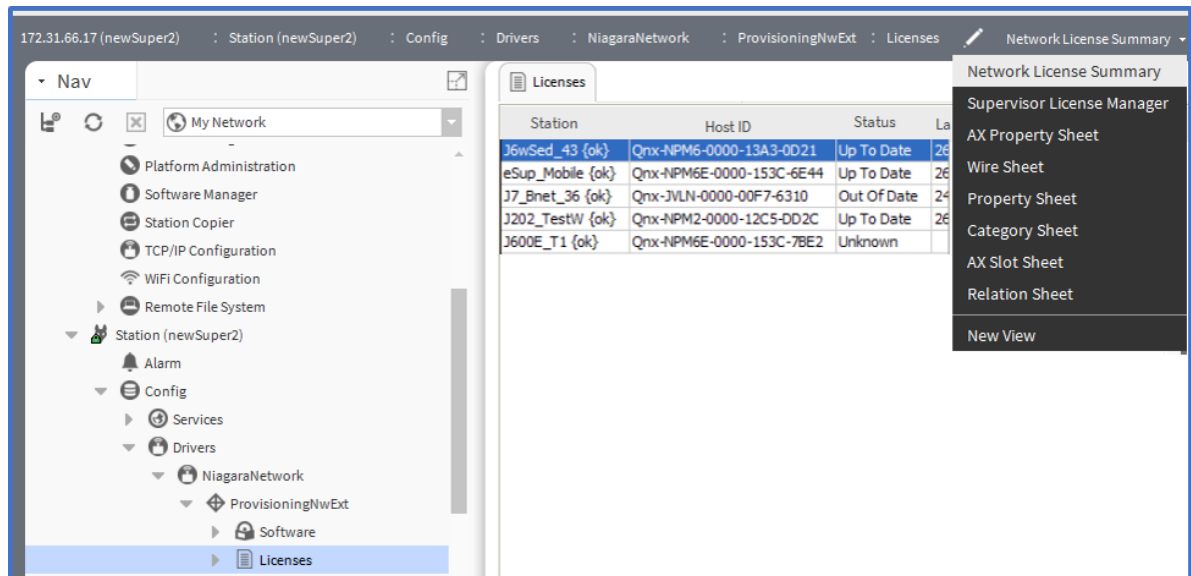
The Supervisor's license database contains the license identifier for each host in the network.

Updating licenses using the Network License Summary

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

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

- Step 1 Select the **Licenses** slot on the **ProvisioningNwExt**.
The system displays the **Network License Summary**.



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

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

Copying a file

You may copy a file from your PC (local file) or a Supervisor file to your remote stations. A local file may be anywhere on your PC. It is a file that is outside of the directory structure that is available from inside your Supervisor station.

Prerequisites: You are working in EC-Net 4 Pro and are connected to your Supervisor station. A ProvisioningNwExt component and a **NiagaraNetworkJobPrototype** component are available under your **NiagaraNetwork**.

Step 1 Expand **Config**→**Drivers** and do one of the following:

- To set up a one-time job that copies a local file from anywhere on your PC to your remote devices, double-click **ProvisioningNwExt**
- To set up a repeating job that copies a file from inside the directory structure that is available from inside your Supervisor station, double-click **NiagaraNetworkJobPrototype**.

For the local file, a **Niagara Network Job Builder** opens. For the Supervisor file, a **Niagara Network Prototype View** opens. They look the same except for the title and buttons at the bottom.

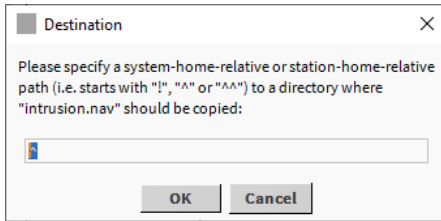
Step 2 In the top pane, **Provisioning steps to run**, click add (+) and select the a copy file step.

The system opens the file chooser.

If you are creating a one-time job using the **Niagara Network Job Builder**, you can select either a Supervisor file or a local file on your PC. If you are creating a job prototype using the **Niagara Network Prototype View**, you can select only a Supervisor file.

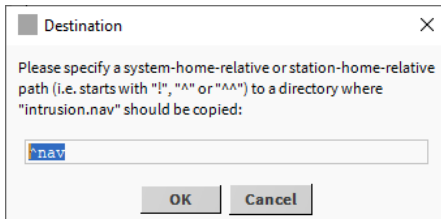
Step 3 Select a local source file to copy.

A **Destination** window prompts you for the target destination to copy this file to.



This destination folder applies to all stations in the job.

Step 4 To edit the destination, click into the entry box and type a destination.



In the example above, the destination was changed to specify a `nav` folder under the station root absolute (^). If a destination folder does not already exist on the target host, the step creates it.

NOTE: The destination string must always begin with the character for either the system-home relative (!) or the station-home relative (^). The system provides no means to modify any files outside of the release directory on any target hosts.

When the job runs, the system makes a temporary copy of the file on the Supervisor unless the file being copied is local to the Supervisor (that is, you are using EC-Net 4 Pro on the Supervisor). Once the job completes, the system deletes this temporary file.

To run the provisioning job more efficiently, the system combines file copy steps with other file copy steps, install software steps, or upgrade out-of-date software steps.

Copying a Supervisor file

A Supervisor file is a file located in the My File System of the Supervisor's Nav tree.

Prerequisites: You have created a new job (one-time, or job prototype) and have either the **Niagara Network Job Builder** or **Niagara Network Schedule View** open.

Step 1 In the top pane, **Provisioning steps to run**, click add (+), and select the **Copy Local File** or **Copy Supervisor File** step.

The system opens the **File Chooser** for the files available as part of the Supervisor station.

Step 2 Using the **File Chooser**, navigate to the source file for the copy operation, select the file, and click **OK**.

If you are creating a one-time job using the **Niagara Network Job Builder**, you can select either a Supervisor file or a local file on your PC. If you are creating a job prototype using the **Niagara Network Prototype View**, you can select only a Supervisor file.

The system prompts you for the target destination.

This destination folder applies to all stations in the job. You may need to edit the destination.

Step 3 Edit the destination system-home-relative or station-home-relative.

The destination string must always begin with the character for either the system-home relative (!) or the station-home relative (^). The system provides no means to modify any files outside of the software release directory on any target host.

The destination is changed to specify a `logos` subfolder under the `images` folder, which is at the station root absolute (^). If a destination folder does not already exist on the target host, the step creates it.

Step 4 Click **Run Now**.

To run more efficiently, the system combines this step with other file copy steps, install software steps, or upgrade out-of-date software steps in the job.

Rebooting the system

This procedure provides a provisioning step to restart the station in each remote device in the job.

Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The **Niagara Network Job Builder** (one-time job) or **Niagara Network Prototype View** (reoccurring job) is open.

Step 1 In the top pane, **Provisioning steps to run**, click add (+).

The **New Job Step** window opens.

Step 2 Select the `Reboot` step and click **OK**.

Step 3 In the bottom pane, **Stations to include in the job**, click add (+).

The **Add Device** window opens.

Step 4 Select the stations to reboot and click **OK**.

Step 5 To initiate the provisioning job, review your choices and click **Run Now** at the bottom of the view.

The step restarts the station's host (platform), then waits until its platform daemon comes back up and is available for connections to begin.

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

Removing stations

You may remove a station from a job in the **Niagara Network Job Builder** or **Niagara Network Prototype View** using either of these 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.

Renaming remote stations

This procedure creates a step to rename the station on the remote device to match the name of the station's proxy name in the Supervisor's **NiagaraNetwork**. This is particularly useful for Edge devices, which are delivered with a default station whose name is unique based on the host ID.

Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The **Niagara Network Job Builder** (one-time job) or the **Niagara Network Prototype View** (reoccurring job) is open. All stations are named as desired under the **NiagaraNetwork** in the Supervisor station.

Step 1 In the top pane, **Provisioning steps to run**, click add (+).

The **New Job Step** window opens.

Step 2 Select the **Rename Device Station** step and click **OK**.

Step 3 In the bottom pane, **Stations to include in the job**, click add (+).

The **Add Device** window opens.

Step 4 Select the stations to rename and click **OK**.

Step 5 To initiate the provisioning job, review your choices and click **Run Now** at the bottom of the view.

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

Reordering stations

Reorder a selected station in the **Niagara Network Job Builder** or **Niagara Network Prototype View** using one of two methods.

- Click the (up) or (down) arrow button at the bottom of the list.
- Right-click a station and select **Move Up** or **Move Down**, as needed.

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

Setting a property

This procedure creates a provisioning job to set a property value of a component on each remote station in the job.

Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The **Niagara Network Job Builder** or **Niagara Network Prototype View** is open.

Step 1 In the top pane, **Provisioning steps to run**, click add (+).

The **New Job Step** window opens.

Step 2 Click the **Set Property** step and click **OK**.

The **Set Property** window opens.

Step 3 Enter the slot path (**Target ORD**) for the target component, the **Property Name** to change, the **Property Type** information, the **Property Value** and click **OK**.

Step 4 In the bottom pane, **Stations to include in the job**, click add (+).

The **Add Device** window opens.

Step 5 Select the stations for which to change the property value and click **OK**.

Step 6 To initiate the provisioning job, review your choices and click **Run Now** at the bottom of the view.

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

Removing a property

Starting in EC-Net 4 v4.7, this procedure creates a provisioning job to remove a property value of a component on each remote station in the job.

Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The **Niagara Network Job Builder** or **Niagara Network Prototype View** is open.

The property that you remove is a dynamic property. The target must be a component and the property must be dynamic.

Step 1 In the top pane, **Provisioning steps to run**, click add (+).

The **New Job Step** window opens.

Step 2 Select the **Remove Property** step and click **OK**.

The **Remove Property** window opens.

Step 3 Enter the slot path (**Target ORD**) for the target component, the **Property Name** to remove, and click **OK**.

In the **Provisioning steps to run** pane, the newly added job appears.

Step 4 In the bottom pane, **Stations to include in the job**, click add (+).

The **Add Device** window opens.

Step 5 Select the stations for which to remove the property value and click **OK**.

Step 6 To initiate the provisioning job, review your choices and click **Run Now** at the bottom of the view.

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

Setting the time

This procedure creates a provisioning job to set the time on each remote device in the job.

Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The **Niagara Network Job Builder** (one-time) or **Niagara Network Prototype View** (reoccurring job) is open.

Step 1 In the top pane, **Provisioning steps to run**, click add (+).

The **New Job Step** window opens.

Step 2 Select the **Set Time** step and click **OK**.

The system opens the **Set Time** window.

Set Time

Set the time for each device

Use supervisor time

Use selected time

Date/Time: 29-Jun-2021 06:54 PM

Time Zone: America/New_York (-5/-4)

Use NTP time

Enabled: true

NTP Host Mode: Server

Sync Local Clock to NTP: true

Sync Time At Boot: false

Use Local Clock as Backup: false

Generate NTP Statistics: false

Address	Peer Mode	Burst	Preferred	Min. Poll Interval	Max. Poll Interval

OK Cancel

Step 3 Select the bullet and configure the time as needed, and click **OK**.

Step 4 In the bottom pane, **Stations to include in the job**, click add (+).

The **Add Device** window opens.

Step 5 Select the stations and click **OK**.

Step 6 To initiate the provisioning job, review your choices and click **Run Now** at the bottom of the view.

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

Chapter 4 Station backup

Topics covered in this chapter

- ◆ Manually triggering a backup
- ◆ Preparing the folder and trigger schedule
- ◆ Setting up the backup provisioning job

Configuring a job prototype to back up all stations provides a good example of how to set up a provisioning job. The provisioning job backs up one or more stations to a Supervisor PC.

Configuring a regularly-scheduled station backup by adding a provisioning job prototype using the **Niagara Network Prototype View** is a better practice than setting up a one-time station backup using the **Niagara Network Job Builder**.

Provisioning backups use either the local **BackupService** or the **CloudBackupService**. The **CloudBackupService** provides all the functions of the local **BackupService**. You can make successful backups using either the **Backup Manager** view to create manual backups, or using the Backup Stations step in a provisioning job. Backups contain the expected files based on the **CloudBackupService** exclusions. Backups created by provisioning job prototype follow the configured job disposition settings and are deleted when a new backup is created.

The topics that follow provide an example procedure for regularly backing up a Supervisor station with 36 device stations in its **NiagaraNetwork**.

Manually triggering a backup

A manual backup is another way to organize a one-time event.

Prerequisites: A ProvisioningNwExt component is configured and in your **NiagaraNetwork**. You have super user permissions in the Supervisor station.

Step 1 Right-click **ProvisioningNwExt** and click **Actions→Start Backup**

The system automatically adds the Backup Stations step (for every station), and stores the backup .dist file for each on the Supervisor, using this path:

```
^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_HH:mm:ss.sss`.

Step 2 To view the .dist files, navigate to the `provisioningNiagara` subfolder on the Supervisor station.

The resulting backup files are encrypted and require credentials to restore.

Preparing the folder and trigger schedule

Setting up a provisioning job begins with setting up the necessary components under the station **Config** folder: **ProvBackups** folder, **NiagaraNetworkJobPrototype** component, **TriggerSchedule** component.

Prerequisites: All stations are properly configured for platform connections. You have super user permissions.

Step 1 With the Supervisor station opened in EC-Net 4 Pro, expand it to reveal its **Config** node.

Step 2 Right-click **Config** and select **New→Folder**.

- Step 3 Name the folder **ProvBackups** (or whatever name you wish. You can create this folder anywhere under **Config**.)
- Step 4 Open the **Palette** side bar, and, in the side bar, open the **provisioningNiagara** palette.
- Step 5 From the **provisioningNiagara** palette, drag a **NiagaraNetworkJobPrototype** component to the new **ProvBackups** folder.
- Step 6 From the **provisioningNiagara** palette, drag a **TriggerSchedule** component to the same **ProvBackups** folder.
- Step 7 To view its Wire Sheet, double-click the **ProvBackups** folder.
- Step 8 In the Wire Sheet, link the **Trigger** slot of the **TriggerSchedule** to the **SubmitJob** slot of the **NiagaraNetworkJobPrototype** component.

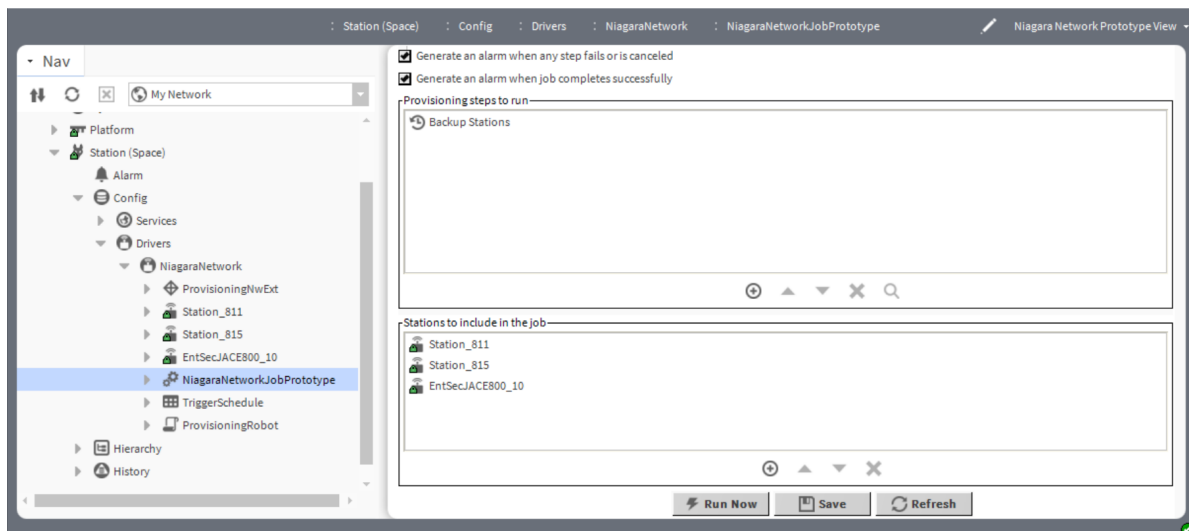
Setting up the backup provisioning job

Each action to be taken is a step in the provisioning job.

Prerequisites: The station Config container contains the necessary folder and components.

- Step 1 To view the provisioning job's **Niagara Network Prototype View**, double-click the **NiagaraNetworkJobPrototype**.

The system opens the provisioning job view.



- Step 2 Configure how you want failed and completed jobs to appear on the alarm console.

NOTE: The alarm check box settings apply to the provisioning job being built, and do not affect other provisioning jobs that may already exist either as other provisioning components, jobs already queued to run, or built in the **NiagaraNetworkJobBuilder** (one-time jobs).

- Step 3 In the top pane, click add (+).
The **New Job Step** window opens.
- Step 4 Select the **Backup Stations** step and click **OK**.
The system adds this job step to the **Provisioning steps to run** pane.
- Step 5 In the bottom pane, click add (+).
The **Add Device** window opens.
- Step 6 Select the stations to back up or click **Check All** and click **OK**.

NOTE: Later, you can duplicate this component and select different stations to back up.

Step 7 To save the prototype, click the **Save** at the bottom of the view.

Chapter 5 Provisioned software installation

Topics covered in this chapter

- ◆ Synchronizing software databases
- ◆ Installing software using the add button
- ◆ Installing software by dragging from the Software container
- ◆ Installing a clean distribution file
- ◆ Installing new software modules
- ◆ Upgrading out-of-date software

Installing software on multiple remote hosts using provisioning involves installing the modules to the Supervisor PC, synchronizing EC-Net 4 Pro and Supervisor databases, setting up a one-time provisioning job, and executing the job.

You need to obtain the necessary license(s) for the software, and download the modules into your Supervisor PC.

Synchronizing software databases

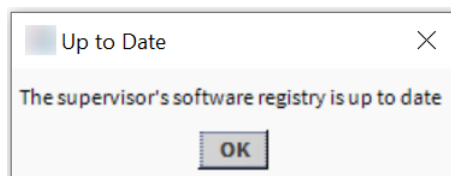
Installing EC-Net 4 Pro on a Supervisor PC or engineering workstation makes all licensed software modules available. To update the modules in a remote host using the Supervisor station, you must first synchronize the EC-Net 4 Pro and Supervisor databases.

Step 1 To open the **Supervisor Software Manager** in the Supervisor station, expand **Config**→**Drivers**→**NiagaraNetwork**→**ProvisioningNwExt** and double-click **Software**.

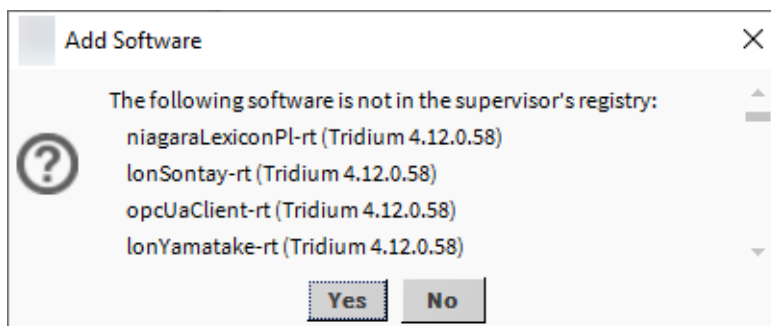
Step 2 Click the **Sync Workbench** button at the bottom of the window.

One of the following happens:

- If the Supervisor already has all the software installable files that are available in the EC-Net 4 Pro database, a display window displays:



- If installable files are available that the Supervisor does not have, an **Add Software** window lists the files, and asks if you wish to transfer them to the Supervisor.




Step 3 Click **Yes** or **No** depending on your needs.

Installing software using the add button

When it is time to install new software on all hosts in a network, you can use provisioning to speed the process of upgrading each remote host. This procedure uses the add button.

Prerequisites: The **Niagara Network Job Builder** (for a one-time job) or the **Niagara Network Prototype View** (for a prototype job) is already open.


Step 1 If you are using a remote PC, connect to the Supervisor station.

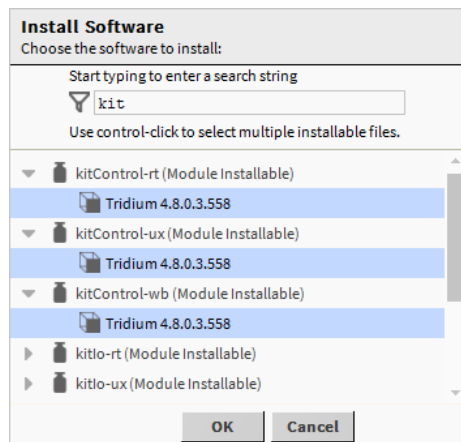
Step 2 In the **Niagara Network Job Prototype** view, click the  (add) button to add the Install Software step to the job.

The system opens the **Install Software** window in which you select the software modules or .dist files to install.

Step 3 Click the triangle to expand an artifact to display available module version numbers. Select the version(s) and click **OK**.

NOTE: In EC-Net 4 v4.8 and later, in the **Install Software** window there is added support for filtering the list of available artifacts for installation (by version or module name), and for making multiple selections of artifacts for installation in a single provisioning step. The Install Software Step will install all modules selected from the **Install Software** window.

- To filter available artifacts, click in the text box next to the  (Filter icon) and start typing the artifact name (as shown) or version. The list of available artifacts will automatically be filtered to match the search string.
- To make multiple selections, press the **Ctrl** key while you click on multiple artifacts.



Once added, the Install Software step appears in the **Job Steps List**.

To satisfy dependencies, if the software has dependencies on one or more modules that are not yet installed on a particular host, and the modules are in the Supervisor's software registry, the step automatically includes the modules in the processing for host (station).

NOTE: It is your (provisioning user's) responsibility to ensure that platform dependencies of the software are met by the 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. However, if a step installs a distribution file specific to one type of controller, note that the dependency check may fail on a different type of device, with no software being installed on that host.

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

To run the provisioning job more efficiently, the system combines Install Software steps with other software install steps, copy file steps, and upgrade out-of-date-software steps.

Installing software by dragging from the Software container

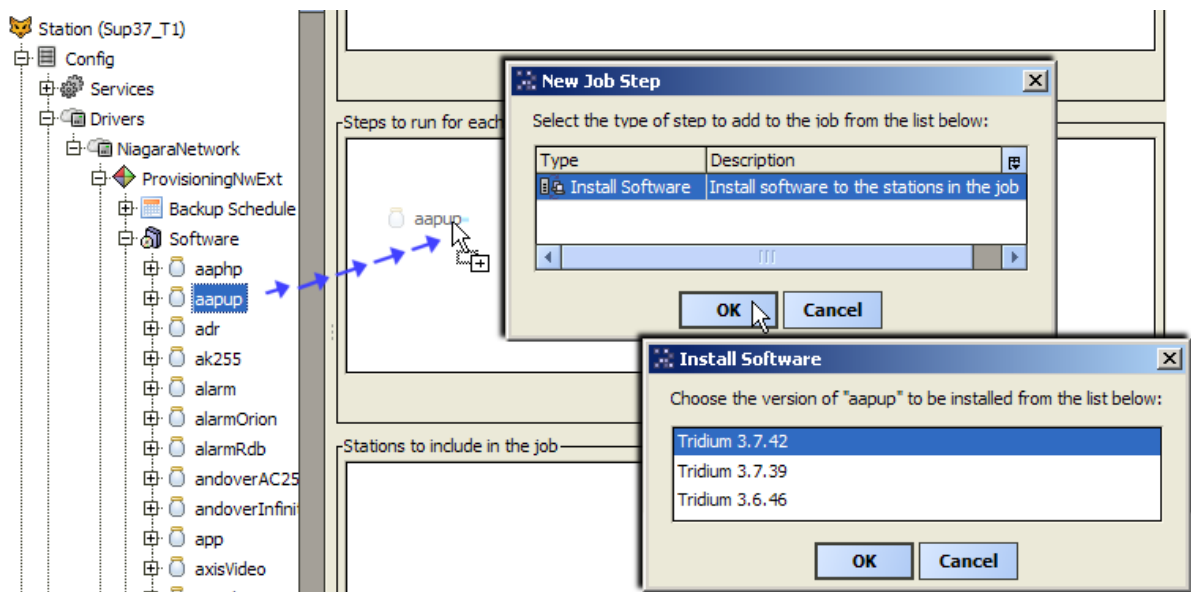
Dragging a file to the **Network Job Builder** or **Niagara Network Prototype View** is an easy way of setting up a copy step.

Prerequisites: The **Niagara Network Job Builder** (for a one-time job) or the **Niagara Network Prototype View** (for a prototype job) is open.

- Step 1 If you are using a remote PC, connect to the Supervisor station.
- Step 2 On the Supervisor station, locate the new software module(s) in the **Software** container.
- Step 3 From the Nav tree, copy or drag a local software module into the **Job Steps List**.

The **JobService** automatically checks to see if the software file and version already exist on the Supervisor. If not, the service downloads the file and registers it with the Supervisor. This download process occurs in the background.

If more than one version (file) for the item is in the Supervisor's software database, a popup window prompts you to select the version.



- Step 4 Select the version and click **OK**.

Once added, the Install Software step appears in the **Job Steps List**.

To satisfy dependencies, if the software has dependencies on one or more modules that are not yet installed on a particular host, and the modules are in the Supervisor's software registry, the step automatically includes the modules in the processing for host (station).

NOTE: It is your (provisioning user's) responsibility to ensure that platform dependencies of the software are met by the 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. However, if a step installs a distribution file specific to one host, note that the dependency check may fail on another type of device, and no software will be installed on that device.

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

To run the provisioning job more efficiently, the system combines Install Software steps with other software install steps, copy file steps, and upgrade out-of-date-software steps.

Installing a clean distribution file

This procedure creates a provisioning job to install a clean distribution file on each station job.

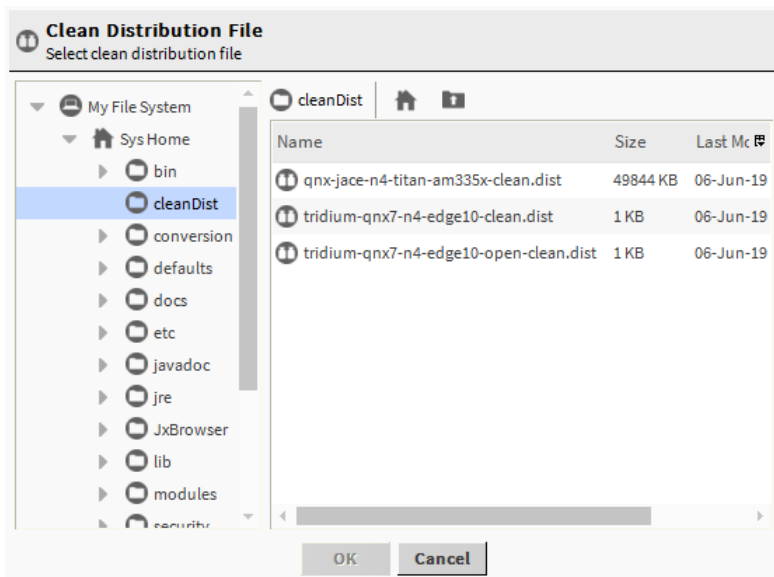
Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The target platform types match the distribution file you intend to provision. Either the **Niagara Network Job Builder** (one-time job) or the **Niagara Network Prototype View** (repeating job) is open.

Step 1 In the top pane, **Provisioning steps to run**, click add (+).

The **New Job Step** window opens.

Step 2 Click the **Install Clean Distribution** step and click **OK**.

The **Clean Distribution File** window opens.



Step 3 Navigate to and select the desired clean distribution file, and click **OK**.

Step 4 In the bottom pane, **Stations to include in the job**, click add (+).

The **Add Device** window opens.

Step 5 Select the stations and click **OK**.

NOTE: Only select stations that match the platform of the selected clean distribution file. The job step fails for any device that does not match the selected clean distribution file.

Step 6 To initiate the provisioning job, review your choices and click the **Run Now**.

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

Installing new software modules

Installing new modules that are not already running on the target device may be required to support all dependencies required by the deployed station services, networks, templates and other components. This is a one-time provisioning procedure.

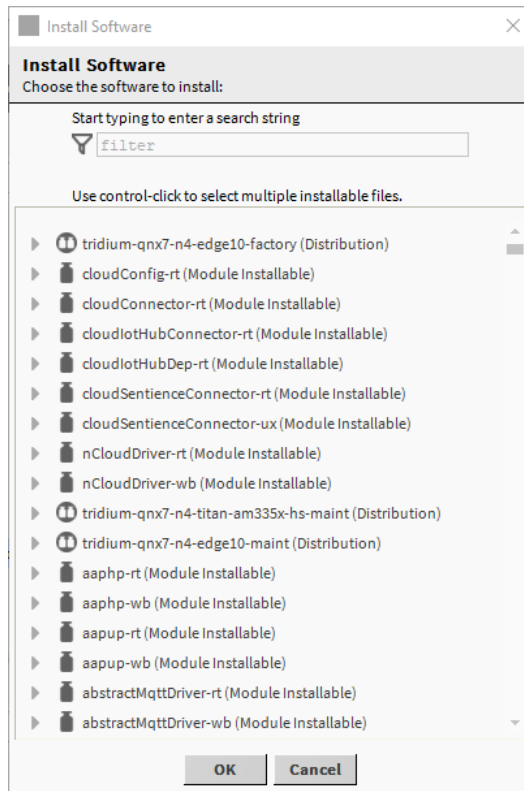
Prerequisites: The **BatchJobService** is under **Services**. The **ProvisioningNwExt** component is under the **NiagaraNetwork**.

Step 1 In the top pane, **Provisioning steps to run**, click add (+).

The **New Job Step** window opens.

Step 2 Select Install Software and click **OK**.

The Install Software window opens.



Step 3 Apply a filter to the list.

Step 4 Select one or more modules and click **OK**.

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

The **Add Device** window opens.

Step 6 Select the stations and click **OK**.

Step 7 To initiate the provisioning job, review your choices and click the **Run Now**.

The view changes to the **Niagara Network Job View**, where steps and results appear as they are executed. The step installs all modules selected from the list.

If needed, the **Sync Workbench** button in the Supervisor **Software Manager** view copies the latest software from the EC-Net 4 Pro installation to the Supervisor station. For more details, refer to “provisioningNiagara-SupervisorSoftwareManager” in the “Components” chapter of this guide.

Upgrading out-of-date software

This procedure creates a provisioning job to upgrade each target device with the latest versions of platform (.dist file) and module (.jar file) software available in the Supervisor's software database.

Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The **Niagara Network Job Builder** (one-time) or **Niagara Network Prototype View** (reoccurring job) is open.

- Step 1 Use the **Sync Workbench** button in the Supervisor **Software Manager** view to copy the latest software from the EC-Net 4 Pro installation to the Supervisor station if needed. For details, refer to the “Software installation” and “Synchronizing software databases” in this guide.
- Step 2 In the **Niagara Network Job Builder**’s top pane, **Provisioning steps to run**, click add (+).
The **New Job Step** window opens.
- Step 3 Select the **Upgrade Out of Date Software** step and click **OK**.
The system adds the job to the list.
- Step 4 In the bottom **Stations to include in the job** pane, click add (+).
The **Add Device** window opens.
- Step 5 Select the stations and click **OK**.
- Step 6 To initiate the provisioning job, review your choices and click **Run Now**.
The **Upgrade Out of Date Software** step 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 the step finds with a higher version on the Supervisor it installs to the station.
The view changes to the **Niagara Network Job View**, where steps and results appear as they are executed.

Normally, a Supervisor has the latest versions of software modules installed. If for some reason it does not, this step always installs the latest version of any software module found under the Supervisor’s software database (under `!sw`) even if the Supervisor itself is using an earlier installed version (as found in its `!modules` directory).

To run more efficiently, the system can combine the upgrade out-of-date-software steps with other software install steps and copy file steps.

Chapter 6 Certificates and credentials

Topics covered in this chapter

- ◆ Configuring subordinate stations for SAML IdP Service and Scheme
- ◆ Setting security job steps
- ◆ Generating a certificate
- ◆ Setting the certificate alias
- ◆ Signing a certificate
- ◆ Installing a certificate
- ◆ Exporting a certificate signing request
- ◆ Importing a signed certificate
- ◆ Setting system passphrase
- ◆ Setting platform credentials
- ◆ Setting platform user password
- ◆ Setting station connection credentials
- ◆ Setting the TLS level

Certificates and credentials are important security elements for every station. Provisioning their deployment across a **NiagaraNetwork** saves substantial configuration time.

All the certificate steps are intended to run as part of a provisioning workflow that establishes valid server certificates on the stations connected to a Supervisor. The typical sequence to run these steps is as follows:

- Generate Certificate
- Create Certificate Signing Request
- Sign Certificate
- Import Signed Certificate
- Set Certificate Alias

The *Station Security Guide* covers all these topics in more detail.

Configuring subordinate stations for SAML IdP Service and Scheme

A provisioning job can configure each remote station's SAML (Security Assertion Markup Language) authentication scheme and local SAML IdP (Identity Provider) service. This procedure covers running a provisioning job on the Supervisor station to configure one or more remote stations with a server certificate (private and public keys) and the SAML authentication scheme configured for the internal IdP.

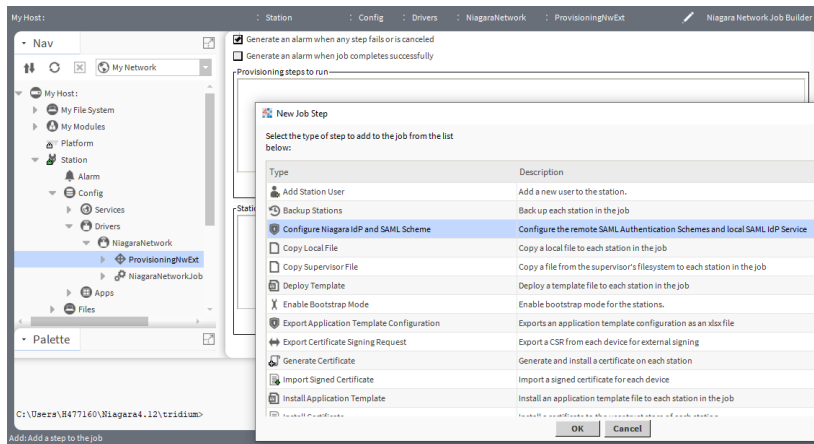
Prerequisites: You are working in a Supervisor station. The **SAMLIdService** is installed and configured on the Supervisor station.


The **NiagaraNetwork** on the Supervisor contains one or more remote stations.

Step 1 Expand **Config**→**Drivers**→**NiagaraNetwork**.

Step 2 To open the **Niagara Network Job Builder** view, double-click on **ProvisioningNwExt**

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



Step 3 In the top pane, **Provisioning steps to run**, click add , click **Configure Niagara IdP and SAML Scheme**, and click **OK**.

The **Configure Niagara IdP and SAML Scheme** window opens.

Step 4 Fill in the following required properties and click **OK**.

- **Login Button Text Format** is the preferred text to display on the login button to access the subordinate station. For example, `Log in to Floor1_%displayName%`, where the system substitutes the name of the Circle of Trust for the BFormat-script `%displayName%`. If multiple Circles of Trust include the subordinate station, the system creates multiple login buttons.
- **IdP Certificate Alias** is the public key of the certificate configured as the **IdP Signing Cert** in the **SAMLIIdPService** that the system imports to the subordinate station's **User Trust Store**. This property configures the alias of the resulting public key as it appears in the subordinate station, for example: `niagaraIdP`. Entering a value in this property activates the **OK** button in the window.
- **SAML Signing Certificate Alias** defines the alias of the certificate to use as the subordinate station's SAML Server Certificate. If not generating a new server certificate (see option below), the certificate should already exist in the subordinate station's **User Key Store**.
- Optionally, you can generate a new server certificate and/or a new remote SAML encryption certificate to use for this purpose. Click the check box to **Generate new remote SAML signing certificate** and/or to **Generate new remote SAML encryption certificate**, fill in the required data in the additional properties, and click **OK**.

The provisioning application adds the step to the job builder.

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

The **Add Device** window opens

Step 6 Select the devices (that is, stations in the **NiagaraNetwork** that are included in a Circle of Trust) to be added to the job, and click **OK**.

The provisioning application adds the job to the step builder.

Step 7 To start the provisioning job, click **Run Now**.

Step 8 To view the job progress open the Job Service **Job Log** view.

The **Job Log** opens.

Status	Timestamp	Message	Details
Message	17-Jan-20 3:01 PM EST	Processing device subStation02	
Message	17-Jan-20 3:01 PM EST	Acquiring public IdP signing certificate	
Message	17-Jan-20 3:01 PM EST	Importing public IdP signing certificate to remote station	
Message	17-Jan-20 3:01 PM EST	Generating remote SAML signing certificate for station	
Message	17-Jan-20 3:01 PM EST	Successfully generated remote SAML signing certificate for station	
Message	17-Jan-20 3:01 PM EST	Getting public SAML signing certificate from remote station	
Message	17-Jan-20 3:01 PM EST	Configuring SAML scheme for circle of trust: westCampus	
Success	17-Jan-20 3:01 PM EST	Step successfully completed for subStation02	
Success	17-Jan-20 3:01 PM EST	Job Success	

This provisioning job exports the public key of the Supervisor's IdP Signing Certificate to the **User Trust Store** of each subordinate station in the job. For each station, it generates a unique SAML Signing Certificate in the station's **User Key Store** (or selects from server certificates already existing in the **User Key Store**). It then as-signs a copy of this certificate's public key to the Station Service Provider under the Circle of Trust in the Supervisor's **SAMLIdPService** using certificate pinning.

NOTE: The signing certificates mentioned in the previous paragraph are actually Server certificates. They should not be confused with Code Signing Certificates which have a different purpose.

For each Circle of Trust that a subordinate station is a part of, the provisioning job creates a SAML Authentication Scheme in the subordinate station's **AuthenticationService**.

For more information, see “About the SAML IdP Service” section of the *Station Security Guide*.

Setting security job steps

This procedure creates three provisioning job steps to set platform credentials, set station user password and set platform user password. It is equivalent to adding these three steps individually.

Prerequisites:

- The **BatchJobService** is available under **Services** and **ProvisioningNwExt** component is available under your **NiagaraNetwork**.
- Either the **Niagara Network Job Builder View** or the **Niagara Network Prototype View** (for a prototype job) is open.

Step 1 In the **Niagara Network Job Builder** view, click (Add) in the top pane, **Provisioning steps to run**.


The **New Job Step** window opens.

Step 2 Select the **Security Job Steps** and click **OK**.

The **Set Platform Credentials** window opens.

Step 3 Enter the new platform username and password, confirm new password, select the **This host will be licensed for FIPS 140–2** check box if this system is intended to be FIPS compliant, and click **OK**.


The **Set Station User Password** window opens.

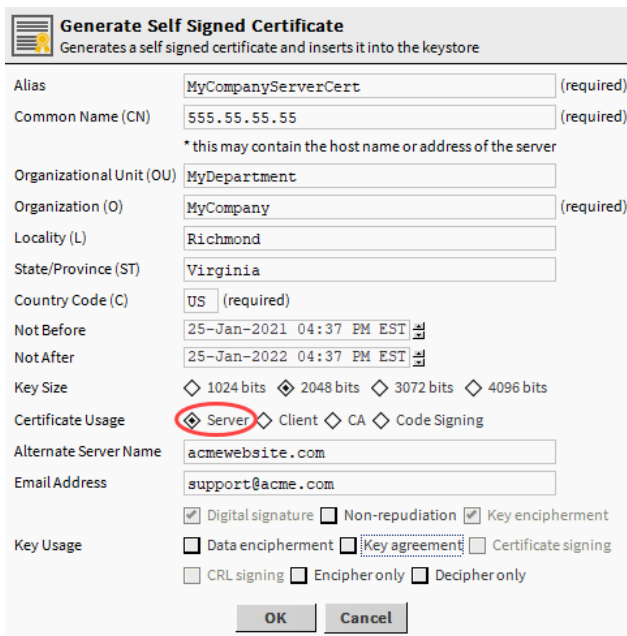
- Step 4 Select the **Use connection username** check box if you intend to use the connection username. Enter the current station user password, new password, confirm new password, and click **OK**.
The **Set Platform User Password** window opens.
- Step 5 To change the password for an existing platform user, enter the current username, current password, new password, confirm password. Select the **This host will be licensed for FIPS 140-2** check box if this system is intended to be FIPS compliant, and click **OK**.
- Step 6 In the bottom **Stations to include in the job**, click  (Add).
The **Add Device** window opens.
- Step 7 Select the stations and click **OK**.
- Step 8 Review your choices and click the **Run Now** button at the bottom of the **Niagara Network Job Builder** view to initiate the provisioning job.
The view changes to the **Niagara Network Job View**, where steps and results appear as they are executed.

Generating a certificate

This procedure creates a provisioning job to generate a server certificate on each remote device in the job.

Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The **Niagara Network Job Builder** or **Niagara Network Prototype View** is open.

- Step 1 In the **Niagara Network Job Builder** window in the top pane, **Provisioning steps to run**, click  (Add).
The **New Job Step** window opens.
- Step 2 Click the **Generate Certificate** step and click **OK**.
The system opens the **Generate Self Signed Certificate** window.



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

Alias: MyCompanyServerCert (required)

Common Name (CN): 555.55.55.55 (required)
* this may contain the host name or address of the server

Organizational Unit (OU): MyDepartment

Organization (O): MyCompany (required)

Locality (L): Richmond

State/Province (ST): Virginia

Country Code (C): US (required)

Not Before: 25-Jan-2021 04:37 PM EST

Not After: 25-Jan-2022 04:37 PM EST

Key Size: ☒ 1024 bits ☐ 2048 bits ☐ 3072 bits ☐ 4096 bits

Certificate Usage: ☒ Server ☐ Client ☐ CA ☐ Code Signing

Alternate Server Name: acmewebsite.com


Email Address: support@acme.com

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

OK Cancel

- Step 3 Enter the certificate information and click **OK**.

Some of the information is optional, but it is generally beneficial to fill in as much information as possible and practical. For **Certificate Usage**, configure **Server**. The image shown here contains an example set of certificate information.


- Step 4 In the bottom pane, **Stations to include in the job**, click  (Add).
- Step 5 In the **Add Device** window, click to select the stations and click **OK**.
- Step 6 To initiate the provisioning job, review your choices and click the **Run Now** button at the bottom of the **Niagara Network Job Builder** view.

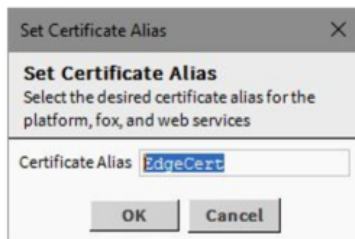
The view changes to the **Niagara Network Job View**, where steps and results appear as they are executed. The job puts the resulting certificate into the **User Key Store** of the remote device platform.


Setting the certificate alias

This procedure creates a provisioning job to set the certificate alias for the certificate to be used for Platform, Fox, and Web services on each remote device in the job. Successfully running this step enables secure connections from the Supervisor to the remote device without having to enable certificate exemptions for the device's Platform and Fox services.

Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The **Niagara Network Job Builder** (one-time) or **Niagara Network Prototype View** (reoccurring job) is open.

- Step 1 In the top pane, **Provisioning steps to run**, click  (Add).
The **New Job Step** window opens.
- Step 2 Click the **Set Certificate Alias** step and click **OK**.
The system opens the **Set Certificate Alias** window.




- Step 3 Enter the **Certificate Alias** on the remote device, and click **OK**.
- Step 4 In the bottom pane, **Stations to include in the job**, click  (Add).
The **Add Device** window opens.
- Step 5 Select the stations and click **OK**.
- Step 6 To initiate the provisioning job, review your choices and click **Run Now**.
The view changes to the **Niagara Network Job View**, where steps and results appear as they are executed.

Signing a certificate

This procedure creates a provisioning job to sign a certificate. This step is intended to run as part of a provisioning workflow to establish server certificates on stations connected to a Supervisor.

Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The **Niagara Network Job Builder** (one-time) or **Niagara Network Prototype View** (reoccurring job) is open.

- Step 1 In the top pane, **Provisioning steps to run for each station**, click  (Add).
The **New Job Step** window opens.
- Step 2 Click the **Sign Certificate** step and click **OK**.
The system opens the **Sign Certificate** window.



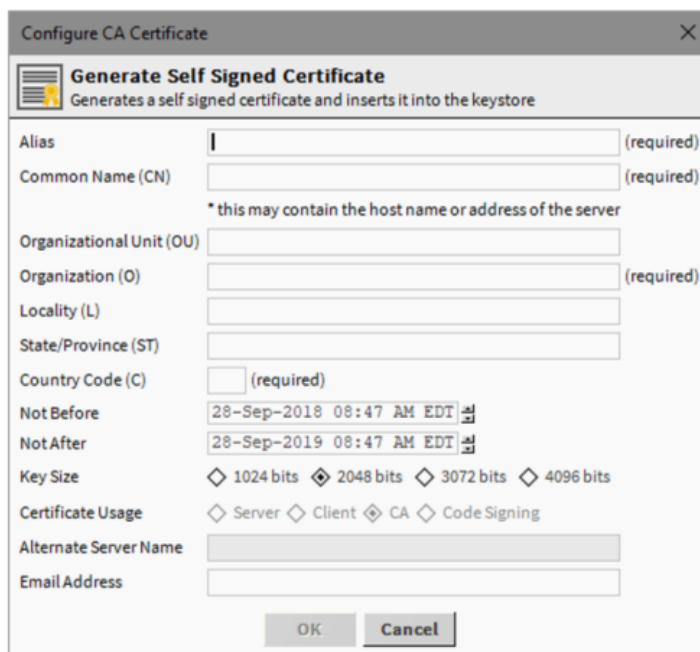
The **Sign Certificate** dialog box is shown. It has a title bar with a close button. Below the title bar is a section titled **Sign Certificate** with the subtitle "Sign a certificate on each station". The form contains the following fields:

- Server Certificate Alias**: Text box with "EdgeCert" entered.
- CA Certificate Alias**: Dropdown menu with "edgesuper" selected.
- CA Certificate Password**: Password field with masked characters.
- Not Before**: Date/time picker showing "01-Jan-2018 12:00 PM EDT".
- Not After**: Date/time picker showing "01-Jan-2019 12:00 PM EDT".

At the bottom are **OK** and **Cancel** buttons.

- Step 3 Enter the appropriate **Server Certificate Alias** on the remote device and **CA Certificate Alias** (as in the example shown above), and click **OK**.

If a CA certificate has not been generated in the Supervisor's **User Key Store**, or if a new CA certificate is desired, select **Generate New** under the **CA Certificate Alias** dropdown. This opens the **Configure CA Certificate** window (shown below) for entering the CA certificate information to be used for the signing CA certificate. When complete, click **OK**.




The **Configure CA Certificate** dialog box is shown. It has a title bar with a close button. Below the title bar is a section titled **Generate Self Signed Certificate** with the subtitle "Generates a self signed certificate and inserts it into the keystore". The form contains the following fields:

- Alias**: Text box with "I" entered, marked as (required).
- Common Name (CN)**: Text box, marked as (required).
- Organizational Unit (OU)**: Text box.
- Organization (O)**: Text box, marked as (required).
- Locality (L)**: Text box.
- State/Province (ST)**: Text box.
- Country Code (C)**: Text box, marked as (required).
- Not Before**: Date/time picker showing "28-Sep-2018 08:47 AM EDT".
- Not After**: Date/time picker showing "28-Sep-2019 08:47 AM EDT".
- Key Size**: Radio buttons for 1024 bits, 2048 bits (selected), 3072 bits, and 4096 bits.
- Certificate Usage**: Radio buttons for Server, Client, CA (selected), and Code Signing.
- Alternate Server Name**: Text box.
- Email Address**: Text box.

At the bottom are **OK** and **Cancel** buttons.

See the "Creating a root CA certificate" section of the *Station Security Guide* for more information on creating a CA certificate used to sign server certificates

- Step 4 In the bottom pane, **Stations to include in the job**, click  (Add).
The **Add Device** window opens.
- Step 5 Select the stations and click **OK**.
- Step 6 To initiate the provisioning job, review your choices and click **Run Now**.

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

Installing a certificate

If the **System Trust Stores** in your remote stations already contain the root CA certificate from the CA (Certificate Authority) that signed your intermediate, server or code-signing certificates, you do not need to run a provisioning job. If your company serves as its own CA, you must install a root CA or intermediate certificate in the **User Trust Stores** of each platform/station that serves as a client. Installing this certificate can be useful before running a signed provisioning robot on several stations.

Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The **Niagara Network Job Builder** (one-time) or **Niagara Network Prototype View** (reoccurring job) is open.

This is a recurring provisioning job that uses the **NiagaraNetworkJobPrototype**.

Step 1 On your Supervisor, expand **Config**→**Services**→**PlatformServices**, double-click **CertManagerService** and click the **User Trust Store** tab.

NOTE: You cannot complete this procedure if you import the certificate into the EC-Net 4 Pro **User Trust Store** of your Supervisor PC.

Step 2 Click the **Import** button, navigate to the location on the thumb drive that contains the root CA certificate and click **Open**.

Step 3 Confirm that the **Subject** of the certificate identifies it as the root CA certificate and click **OK**.

The system imports the certificate in preparation for the provisioning job.

Step 4 Navigate to the location in the station where you manage provisioning jobs.

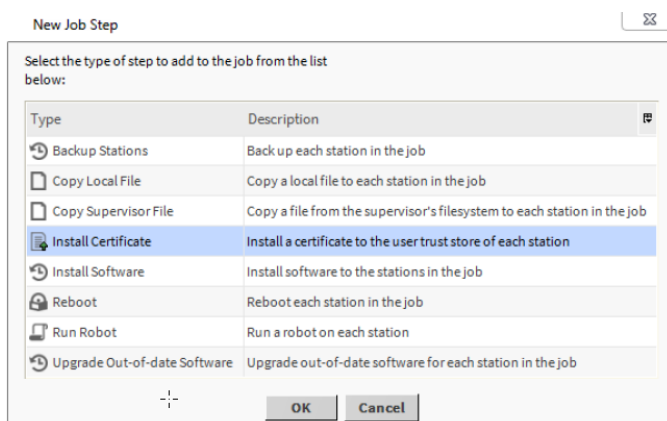
Step 5 Drag a **NiagaraNetworkJobPrototype** component to this location and name the component something like, “Root CA certificate provisioning.”

Step 6 Double-click the new component.

The **Niagara Network Prototype View** opens.

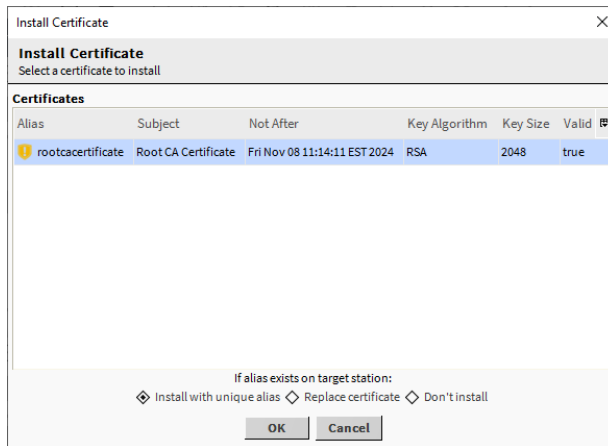
Step 7 In the top pane, **Provisioning steps to run**, click  (Add).

The **New Job Step** window opens.



Step 8 Select **Install Certificate** and click **OK**.

The **Install Certificate** window opens.



Step 9 To complete the installation, select the root CA certificate and click **OK**.

Step 10 Define the stations to include in the job.

The system copies the certificate from the Supervisor station's **User Trust Store** to the **User Trust Stores** of the other clients.

Exporting a certificate signing request

This procedure creates a provisioning job to generate and export a certificate signing request (CSR) file for a certificate installed on each remote device in the job. It stores the resulting files in the `^csrExport` folder on the Supervisor station. This step is intended to run as part of a provisioning workflow to establish server certificates on stations connected to a Supervisor.

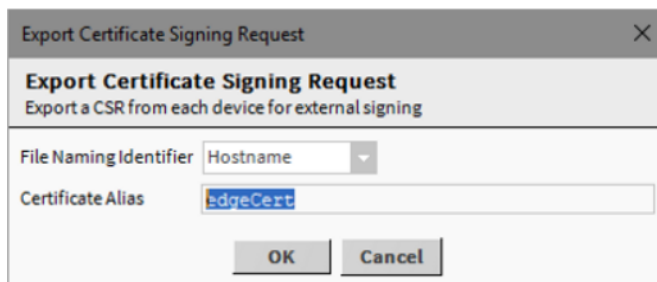
Prerequisites:

- The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The **Niagara Network Job Builder** (one-time) or **Niagara Network Prototype View** (reoccurring job) is open.
- A server certificate, typically created from the Generate Certificate step.

Step 1 In the **Niagara Network Job Builder**'s top pane, **Provisioning steps to run** click  (Add).

Step 2 In the **New Job Step** window, click the **Export Certificate Signing Request** step and click **OK**.

The system opens the **Export Certificate Signing Request** window.



Step 3 Enter the appropriate **File Naming Identifier** and **Certificate Alias** for the certificate on the remote device (as shown in the example), and click **OK**.

NOTE: The alias must be for a server certificate that exists on the target device.

Step 4 In the bottom pane, **Stations to include in the job**, click  (Add).

Step 5 In the **Add Device** window, click to select the stations and click **OK**.

- Step 6 Review your choices, and to initiate the provisioning job, click the **Run Now** button at the bottom of the **Niagara Network Job Builder** view.

The view changes to the **Niagara Network Job View**, where steps and results appear as they are executed. The resulting files will be in the `^csrExport` folder on the Supervisor station.

A CA certificate signs the CSR. The Import Certificate step can then be used to import the signed certificate.

Refer to the “Creating a CSR” topic of the *Station Security Guide* for more information.

Importing a signed certificate

This procedure creates a provisioning job to import a signed certificate for each remote device in the job. The files to import should be located in the `^certImport` folder on the Supervisor station. See the “Importing the signed certificate back into the User Key Store” section of the *Station Security Guide* for more information on importing a certificate.

Prerequisites:

- The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The **Niagara Network Job Builder** (one-time) or **Niagara Network Prototype View** (reoccurring job) is open.
- A signed certificate exists in the `^certImport` folder on the Supervisor station.

- Step 1 In the **Niagara Network Job Builder** window in the top pane, **Provisioning steps to run for each station**, click add (+).

The **New Job Step** window opens.

- Step 2 Select the **Import Signed Certificate** step and click **OK**.

The system opens the **Import Signed Certificate** window.

- Step 3 Select the **File Naming Identifier** and the **Certificate Alias** for the certificate on the remote device. If the certificate contains a private key, enter it into the **Supervisor Password** property.

- Step 4 Alternatively (as of EC-Net 4.14), you have the following options while importing the signed certificate:

- The **File Naming Identifier** is **Hostname**: if you select the **Use Filename as Alias** checkbox and leave the **Certificate Alias** property empty, the **Import Signed Certificate** job step will use the host name as the certificate alias. The file name of the certificate will be `<hostname>.pem`.
- The **File Naming Identifier** is **Host Id**: if you select the **Use Filename as Alias** checkbox and leave the **Certificate Alias** property empty, the **Import Signed Certificate** job step will use the host id as the certificate alias. The file name of the certificate will be `<Hostid>.pem`.
- The **File Naming Identifier** is **Station Name**: if you select the **Use Filename as Alias** checkbox and leave the **Certificate Alias** property empty, the **Import Signed Certificate** job step will use the station name as the certificate alias. The file name of the certificate will be `<StationName>.pem`.

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

The **Add Device** window opens.

Step 6 Click to select the stations and click **OK**.

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

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

Setting system passphrase

This procedure creates a provisioning job to set the system passphrase on each remote device in the job.

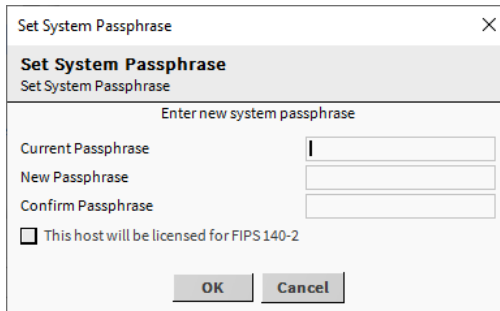
Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The **Niagara Network Job Builder** (one-time) or **Niagara Network Prototype View** (reoccurring job) is open.

Step 1 In the top pane, **Provisioning steps to run**, click add (+).

The **New Job Step** window opens.

Step 2 Click the **Set System Passphrase** step and click **OK**.

The system prompts for the current and new system passphrase for the remote device.



Set System Passphrase

Set System Passphrase
Set System Passphrase

Enter new system passphrase

Current Passphrase

New Passphrase

Confirm Passphrase

☐ This host will be licensed for FIPS 140-2

OK Cancel

Step 3 Enter the current and new system passphrase for the remote device, select the check box if this system is intended to be FIPS compliant, and click **OK**.

Step 4 In the bottom pane, **Stations to include in the job**, click add (+).

The **Add Device** window opens.

Step 5 Select the stations and click **OK**.

Step 6 To initiate the provisioning job, review your choices and click **Run Now**.

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

Setting platform credentials

This procedure creates a provisioning step to set the platform credentials on each remote device in the job. It also updates the platform credentials in the corresponding station proxy in the Supervisor's **NiagaraNetwork**.

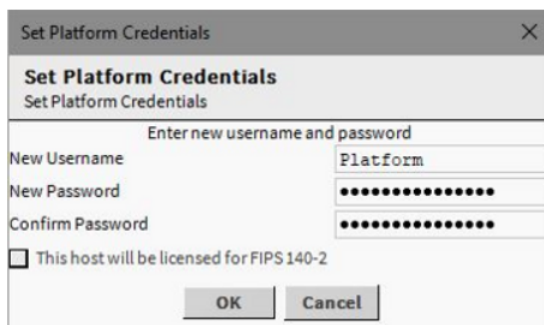
Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The **Niagara Network Job Builder** (one-time) or **Niagara Network Prototype View** (reoccurring job) is open.

Step 1 In the top pane, **Provisioning steps to run**, click add (+).

The **New Job Step** window opens.

Step 2 Click the **Set Platform Credentials** step and click **OK**.

The system prompts for the new username and password for the platform connection to the remote device.



Set Platform Credentials

Set Platform Credentials
Set Platform Credentials

Enter new username and password

New Username

New Password

Confirm Password

☐ This host will be licensed for FIPS 140-2

OK Cancel

Step 3 Enter the **Username** and **Password** for the platform connection to the remote device, select the check box if this system is intended to be FIPS compliant, and click **OK**.

Step 4 In the bottom pane, **Stations to include in the job**, click add (+).

The **Add Device** window opens.

Step 5 Select the stations and click **OK**.

Step 6 To initiate the provisioning job, review your choices and click **Run Now**.

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

Setting platform user password

This procedure helps you create a provisioning step to change the password for an existing platform user.

Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The **Niagara Network Job Builder** (one-time) or **Niagara Network Prototype View** (reoccurring job) is open.

Step 1 In the top pane, **Provisioning steps to run for each station**, click add (+).

The **New Job Step** window opens.

Step 2 Select the **Set Platform User Password** step and click **OK**.

The **Set Platform User Password** window opens.

Step 3 Enter **Current Username**, **Current Password**, **New Password** and **Confirm Password** for the platform connection to the remote device, select the check box if this system is intended to be FIPS compliant, and click **OK**.

Step 4 In the bottom pane, **Stations to include in the job**, click add (+).

The **Add Device** window opens.

Step 5 Select the stations and click **OK**.

Step 6 To initiate the provisioning job, review your choices and click **Run Now**.

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

Setting station connection credentials

This procedure creates a provisioning job to set the username and password for the station (foxs or fox) connection credentials for each station proxy in the job on the Supervisor's **Niagara Network**. No changes are made to the remote station. Existing station connections are reset to use the new credentials as part of this step execution.

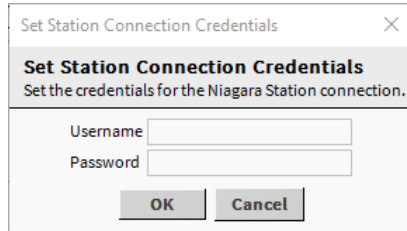
Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The **Niagara Network Job Builder** (one-time) or **Niagara Network Prototype View** (reoccurring job) is open.

Step 1 In the **Niagara Network Job Builder** window, top pane, **Provisioning steps to run**, click add (+).

The **New Job Step** window opens.

Step 2 Click the **Set Station Connection Credentials** step and click **OK**.

The **Set Station Connections Credentials** window prompts for the new username and password for the station connection to the remote device.



The dialog box titled "Set Station Connection Credentials" has a close button (X) in the top right corner. Below the title bar, the text "Set Station Connection Credentials" is followed by "Set the credentials for the Niagara Station connection." There are two input fields: "Username" and "Password". At the bottom, there are "OK" and "Cancel" buttons.

Step 3 Enter a username and password for the station connection and click **OK**.

Step 4 In the bottom pane, **Stations to include in the job**, click add (+).

The **Add Device** window opens.

Step 5 Click to select the stations and click **OK**.

Step 6 To initiate the provisioning job, review your choices and click the **Run Now** button at the bottom of the **Niagara Network Job Builder** view.

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

Setting the TLS level

This procedure creates a provisioning job to set the Transport Layer Security (TLS) level for Platform, Fox, and Web services on each remote device in the job.

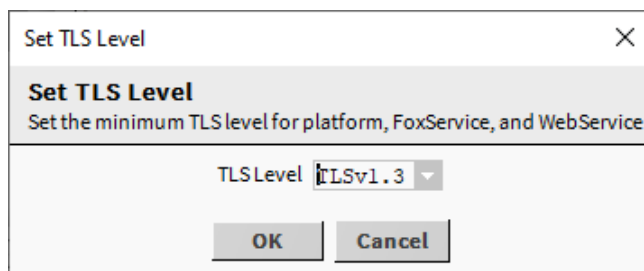
Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The **Niagara Network Job Builder** (one-time) or **Niagara Network Prototype View** (reoccurring job) is open.

Step 1 In the top pane, **Provisioning steps to run**, click add (+).

The **New Job Step** window opens.

Step 2 Click the **Set TLS Level** step and click **OK**.

The system opens the **Set TLS Level** window.



The dialog box titled "Set TLS Level" has a close button (X) in the top right corner. Below the title bar, the text "Set TLS Level" is followed by "Set the minimum TLS level for platform, FoxService, and WebService". There is a drop-down menu labeled "TLSLevel" with "TLSv1.3" selected. At the bottom, there are "OK" and "Cancel" buttons.

Step 3 Select the appropriate level from the drop-down list, and click **OK**.

Step 4 In the bottom pane, **Stations to include in the job**, click add (+).

The **Add Device** window opens.

Step 5 Select the stations and click **OK**.

Step 6 To initiate the provisioning job, review your choices and click **Run Now**.

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

Chapter 7 Templates

Topics covered in this chapter

- ◆ Adding a template
- ◆ Exporting the template spreadsheet
- ◆ Editing the Excel file to configure template instances
- ◆ Copying the Excel and templates to the Supervisor
- ◆ Setting up and running the bulk provisioning job
- ◆ Upgrading a template
- ◆ Provisioning application templates

Provisioning supports deploying standard and application templates.

Adding a template

Component and device templates are created from components in an existing station. One or more templates can be distributed to multiple stations using the **Deploy Template** provisioning step. Bulk template provisioning involves the following primary steps.


Prerequisites: A template created from a station and exported to an Excel file exists. The Excel file contains at least one entry. The template and Excel files were copied to the Supervisor

For details on creating and editing templates, refer to the “Template bulk deployment” topic in the *EC-Net Templates Guide*.

Step 1 If it does not already exist, generate the bulk deployment Excel file for one or more templates to be provisioned.

Step 2 Edit the Excel file to include template instances to be deployed.

Step 3 Copy the Excel and template files to the provisioning Supervisor.

Step 4 Expand **Config**→**Drivers**→**NiagaraNetwork**, double-click **ProvisioningNwExt**, in the top pane, **Provisioning steps to run**, click  (Add), select **Install Application Template** and click **OK**.

This opens an **Application Template Configuration** file chooser.

Step 5 Navigate to the Supervisor’s `^templateCache` folder and select the Excel file to use for this step.

The system adds this step to the job sequence and shows the file path ORD to use.

If more than one Excel file/template file combination is needed to complete an entire application installation, you may export multiple templates to a single Excel file, where each template configuration occupies a separate Excel worksheet tab. Each tab in the Excel workbook configures a separate template instance.

You may include multiple **Deploy Template** steps in a single provisioning job.

Exporting the template spreadsheet

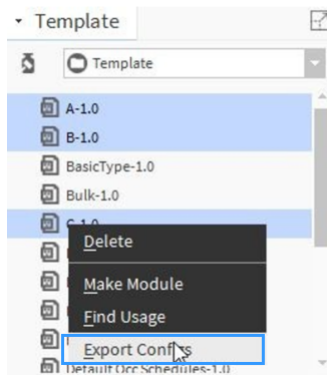
Exporting template configuration properties to an Excel spreadsheet assembles in a single file the information to be deployed to each station.

Prerequisites:

- The **Template** sidebar is open showing the list of templates.
- The templates to include in the bulk deployment (already created) are visible in the **Template** sidebar list.

You may create and deploy more than one Excel file/template file set to complete an entire application installation or include multiple templates as part of the single Excel file. The export function generates a separate Excel worksheet tab for each template.

Step 1 In the **Template** sidebar, select one or more templates, right-click and click **Export Config**.



The **File Chooser** window opens.

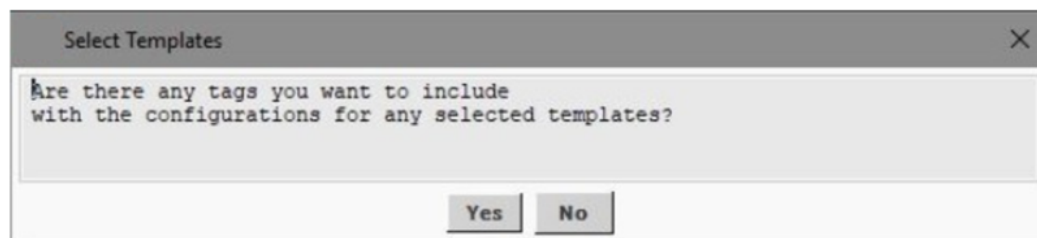
Step 2 In the **File Chooser** window, select a destination and name (extension `.xls` or `.xlsx`) for the exported file, and click **Save**.

An **Encrypt Document** window opens. This document encryption is generally optional, but if the template contains password elements, the exported configurations must be encrypted.

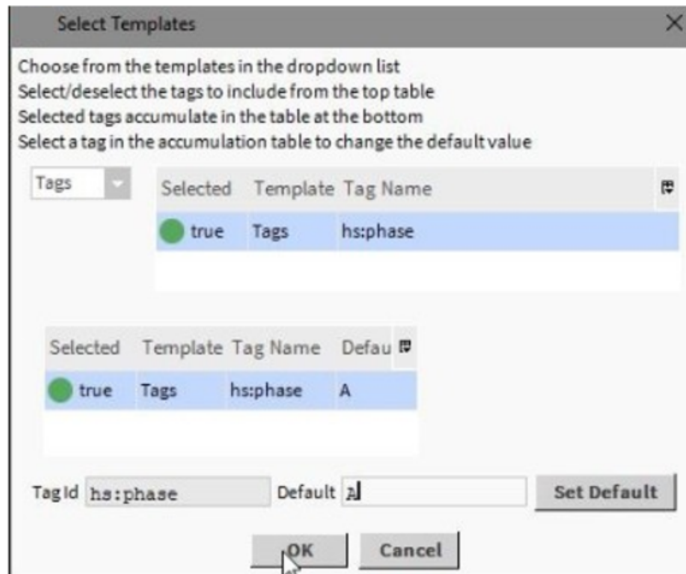


If encryption is desired, enter and confirm the password for unlocking the exported file.

Step 3 A **Select Templates** window opens. Click **Yes** to include string tags as configurations, or **No** to skip (no other tags, such as marker tags, can be included).



Step 4 If you clicked “Yes” in the previous window, in this second **Select Templates** window click the drop-down list in the upper-left corner and select a template. In the example shown here, a template named “Tags” is chosen.



The system populates the table to the right of the dropdown list with the set of string tags that are attached to any component in the chosen template. These are the tags that are available for export. By default, the **Selected** indicator for all tags is initially set to `false`.

Step 5 To include a string tag, click its row in the table. The **Selected** indicator changes to `true`.

The table at the bottom of the window contains the accumulated set of tags for export as well as the default value of the tag, which initially is blank.

Step 6 To change the **Default** value, select the row in the bottom table, and edit the text in the **Tag Id** field.

The system populates the **Default** value in the accumulation table.

NOTE: The system applies the selected tags to the template from which they were selected. If multiple templates have components with the same tag you must select the tag in each applicable template.

Step 7 Once the accumulation table contains all the tags you need, click **OK**.

The system creates an Excel spreadsheet in the `templates` folder.

Editing the Excel file to configure template instances

Adding template instances into the Excel file is exactly the same for provisioning as it is for regular template bulk deployment.

See the “Editing a spreadsheet” section of the *EC-Net Templates Guide* for information on configuring the Excel file for bulk deployment.

Copying the Excel file and templates to the Supervisor

Before running the bulk provisioning job, this procedure copies the Excel file and templates to the `templateCache` folder in the Supervisor station.

Prerequisites: The templates to bulk deploy and Excel file exist. You are working in the Supervisor station.

NOTE: This copying procedure is the same for component/device templates (`.ntp1`) and application templates (`.nap1`), the only difference being the template file extension.

Step 1 Navigate to **Config**→**Drivers**→**NiagaraNetwork** and double-click **ProvisioningNwExt**


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

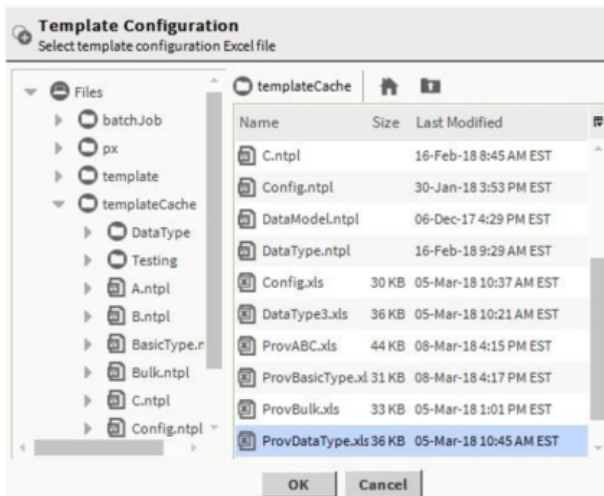
- Step 2 Click the **Copy Templates** button at the bottom of the view.
A File Chooser window opens.
- Step 3 Use the multi-select (**Shift** and **Control**) to select the set of `.xls` and `.ntpl` (or `.napl` as needed) files to be used for provisioning, and click **OK**.
The system transfers the selected files from their source (typically inside the user home `templates` folder) to the Supervisor where it places them in the `templateCache` folder.

Setting up and running the bulk provisioning job

These steps configure the bulk provisioning job. A single provisioning job may include multiple Deploy Template steps.

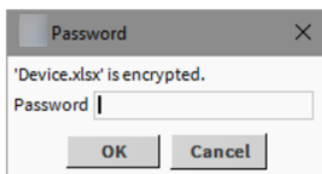
Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The **Niagara Network Job Builder** (one-time) or **Niagara Network Prototype View** (reoccurring job) is open.

- Step 1 In the **Niagara Network Job Builder** window in the top pane, **Provisioning steps to run**, click  (Add).
The **New Job Step** window opens.
- Step 2 Click the **Deploy Template** step and click **OK**.
A **File Chooser** window opens so you can select the Excel file to be used for deploying templates to each remote station in the job.
- Step 3 Navigate to the `templateCache` folder and select the Excel file to use for this step, and click **OK**.



The system adds the use of the Excel file as a step in the job sequence and displays the file path ORD to use.

A **Password** window opens.



- Step 4 If the selected Excel file is encrypted, click the check box, type in the password and click **OK**.
The **Niagara Network Prototype View** opens.

Step 5 To initiate the provisioning job, review your choices and click **Run Now** at the bottom of the view.

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

The screenshot shows the 'Niagara Network Job View' for a job named 'NiagaraNetworkJob3'. The user is 'admin'. The job started and ended on 13-Mar-18 at 2:21 PM EDT, with a state of 'Success'. Below this is a table with columns: Device, Step, Started, Ended, and a status icon. The table lists 18 steps for various devices (transform80, va51tr14a, va51tr14b, va51tr14c). Most steps are successful, but three steps for device 'va51tr14b' failed. At the bottom are buttons: View Log, Job List, Cancel Job, Cancel Device, and Dispose.

Device	Step	Started	Ended	Status
transform80	Deploy Template Step: Excel File Ord Path = "file:*templateCache/ProvABC.xls"	13-Mar-18 2:21 PM EDT	13-Mar-18 2:21 PM EDT	Success
va51tr14a	Deploy Template Step: Excel File Ord Path = "file:*templateCache/ProvABC.xls"	13-Mar-18 2:21 PM EDT	13-Mar-18 2:21 PM EDT	Success
va51tr14b	Deploy Template Step: Excel File Ord Path = "file:*templateCache/ProvABC.xls"	13-Mar-18 2:21 PM EDT	13-Mar-18 2:21 PM EDT	Success
va51tr14c	Deploy Template Step: Excel File Ord Path = "file:*templateCache/ProvABC.xls"	13-Mar-18 2:21 PM EDT	13-Mar-18 2:21 PM EDT	Success
va51tr14b	Deploy Template Step: Excel File Ord Path = "file:*templateCache/ProvBulk.xls"	13-Mar-18 2:21 PM EDT	13-Mar-18 2:21 PM EDT	Success
va51tr14c	Deploy Template Step: Excel File Ord Path = "file:*templateCache/ProvBulk.xls"	13-Mar-18 2:21 PM EDT	13-Mar-18 2:21 PM EDT	Success
va51tr14a	Deploy Template Step: Excel File Ord Path = "file:*templateCache/ProvBulk.xls"	13-Mar-18 2:21 PM EDT	13-Mar-18 2:21 PM EDT	Success
transform80	Deploy Template Step: Excel File Ord Path = "file:*templateCache/ProvBulk.xls"	13-Mar-18 2:21 PM EDT	13-Mar-18 2:21 PM EDT	Success
va51tr14b	Deploy Template Step: Excel File Ord Path = "file:*templateCache/ProvBasicType.xls"	13-Mar-18 2:21 PM EDT	13-Mar-18 2:21 PM EDT	Success
va51tr14c	Deploy Template Step: Excel File Ord Path = "file:*templateCache/ProvBasicType.xls"	13-Mar-18 2:21 PM EDT	13-Mar-18 2:21 PM EDT	Success
va51tr14a	Deploy Template Step: Excel File Ord Path = "file:*templateCache/ProvBasicType.xls"	13-Mar-18 2:21 PM EDT	13-Mar-18 2:21 PM EDT	Success
transform80	Deploy Template Step: Excel File Ord Path = "file:*templateCache/ProvBasicType.xls"	13-Mar-18 2:21 PM EDT	13-Mar-18 2:21 PM EDT	Success
va51tr14b	Deploy Template Step: Excel File Ord Path = "file:*templateCache/ProvDataType.xls"	13-Mar-18 2:21 PM EDT	13-Mar-18 2:21 PM EDT	Failed
va51tr14c	Deploy Template Step: Excel File Ord Path = "file:*templateCache/ProvDataType.xls"	13-Mar-18 2:21 PM EDT	13-Mar-18 2:21 PM EDT	Failed
va51tr14a	Deploy Template Step: Excel File Ord Path = "file:*templateCache/ProvDataType.xls"	13-Mar-18 2:21 PM EDT	13-Mar-18 2:21 PM EDT	Failed
transform80	Deploy Template Step: Excel File Ord Path = "file:*templateCache/ProvDataType.xls"	13-Mar-18 2:21 PM EDT	13-Mar-18 2:21 PM EDT	Success

Each row in the table represents a single step execution on a single station.

Step 6 On completion, for any row in the table double-click the double arrow chevron icon (>>) at the far-right end of the row to see the job execution details.

The **Batch Job Step Log File View** opens.

The screenshot shows the 'Batch Job Step Log File View' for a job step. The device is 'va51tr14a' and the job step is 'Deploy Template Step: Excel File Ord Path = "file:*templateCache/ProvDataType.xls"'. The job started and ended on 13-Mar-18 at 2:21 PM EDT, with a state of 'Failed'. Below this is a table with columns: Status, Timestamp, and Message. The table shows four messages: three successful messages and one failed message. A 'Log Details' dialog box is open, showing the full text of the failed message: 'Failed [14:21:50 13-Mar-18] Missing module dependencies for template DataType: Module: report-rt'. At the bottom are buttons: Job Summary and Refresh Log.

Status	Timestamp	Message
Message	13-Mar-18 2:21 PM EDT	Installing template type = Component, title = DataType, version = 1.0, to target name = DT1, i
Message	13-Mar-18 2:21 PM EDT	Resolving remote component ord ip:172.31.67.14 foxs: station: h:ef9
Message	13-Mar-18 2:21 PM EDT	Copying template file DataType.ntpl
Failed	13-Mar-18 2:21 PM EDT	Missing module dependencies for template DataType:

Log Details

Failed [14:21:50 13-Mar-18] Missing module dependencies for template DataType:
Module: report-rt

Entries in this table display the first line of any progress update or failure messages for that job step.

Step 7 Double-click on a line to see the full text of the log message.


This can be helpful to determine why a step succeeded or failed.

Upgrading a template

This procedure creates a provisioning job to upgrade an existing component or device template on each remote device station in the job.

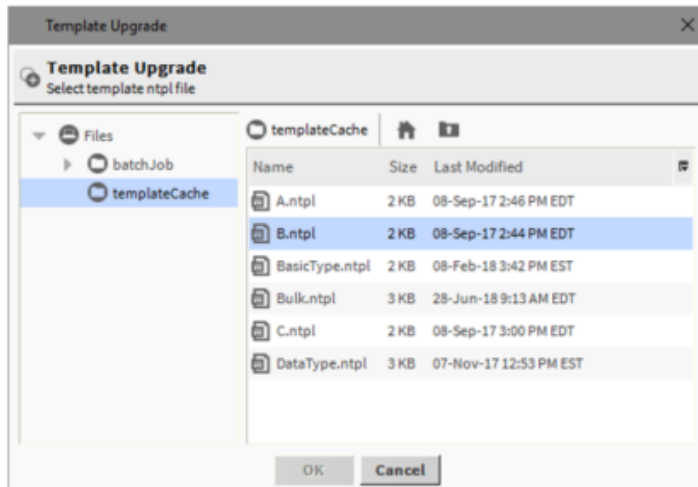
Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **Niagara Network**. The **Niagara Network Job Builder** (one-time) or **Niagara Network Prototype View** (reoccurring job) is open.

Step 1 To copy the upgraded .ntpl file to the Supervisor station, expand **Config**→**Drivers**→**Niagara Network**, double-click **ProvisioningNwExt** and click the **Copy Templates** button at the bottom of the view.

Step 2 In the top pane, **Provisioning steps to run**, click  (Add).
The **New Job Step** window opens.


Step 3 Select the **Upgrade Template** step and click **OK**.

The system opens a **File Chooser** window so you can select the template (*.ntpl) file to be used for deploying templates to each remote station in the job.



Step 4 Navigate to the `templateCache` folder and select the Excel file to use for this step.

The system adds the step in the job sequence and displays the template file path ORD to use.

Step 5 In the bottom pane, **Stations to include in the job**, click  (Add).
The **Add Device** window opens.

Step 6 Click to select the stations and click **OK**.

Step 7 Review your choices and do one of the following:

- Click the **Run Now** button at the bottom of the **Niagara Network Job Builder** view to initiate the provisioning job.
- Click the **Save** button at the bottom of the **Niagara Network Prototype View** and **Run** the prototype to apply the upgraded template.

The results if you save the job are the same as if you upgraded each template individually with each station's **Template Manager**. The job reapplies initial configurations on the template to the upgraded template.

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

You may include multiple **Upgrade Template** job steps in a single provisioning prototype to upgrade multiple templates at the same time.

Provisioning application templates

Application templates can be deployed to multiple stations using provisioning and template bulk deployment. The workflow is similar to component and device templates, with a few key differences that are outlined here.

NOTE: Keep in mind that only one application template can be installed on a station at any one time. Deploying multiple instances of an application template to a station is not meaningful - only the contents of the last application template installed will exist on the station.

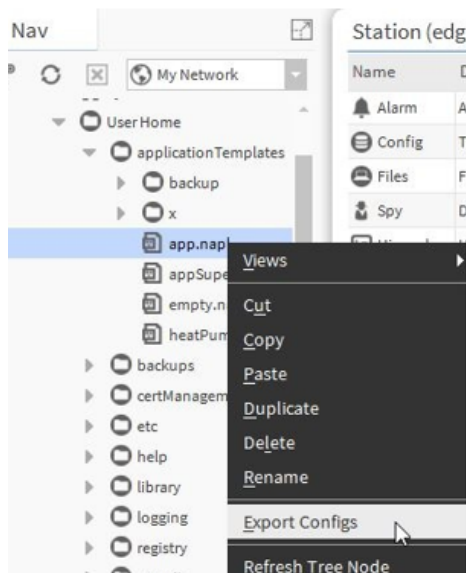
Exporting configurations for an application template

This procedure describes the steps to export the configurations of an application template to an Excel spreadsheet.

Prerequisites: There is an existing application template to be exported.

NOTE: Application templates are not made available from the **Template** sidebar.

- Step 1 Navigate to the EC-Net 4 Pro user home `applicationTemplates` folder.
- Step 2 Right-click on the desired application template file (`.nap1`) and click the **Export Configs** option in the pop-up menu.



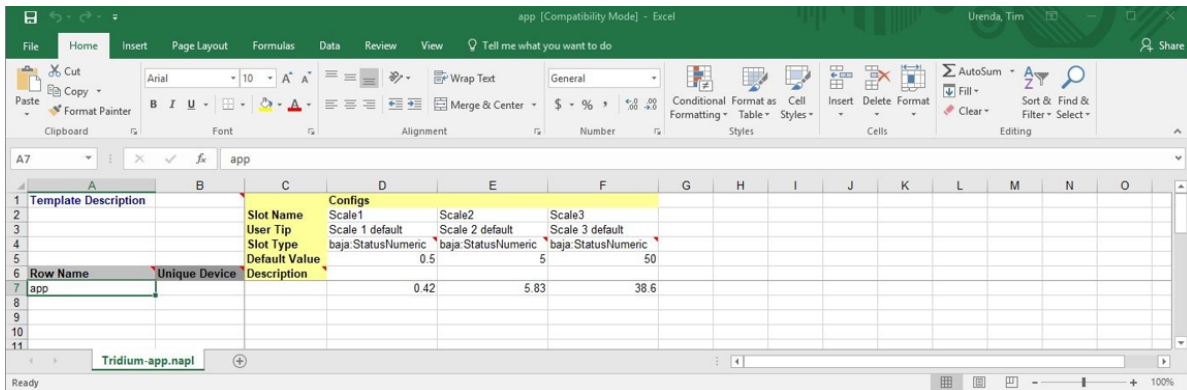
The results are the same as for component templates, the configuration data is exported to an Excel spreadsheet, except that application templates have no input, output, or relation configurations. For details, see [Exporting the template spreadsheet, page 71](#).

Editing the spreadsheet for an application template

This procedure describes how to edit application template data exported to an Excel spreadsheet.

Prerequisites: You have already exported an application template (which creates the template Excel spreadsheet).

- Step 1 Navigate to the spreadsheet file and double-click to open it in Excel.



Step 2 Edit any configuration value as needed, using appropriate entries.

NOTE: Application templates have no input, output, or relation configurations.

NOTE: One entry is required in the Excel spreadsheet to generate an instance of the application template during installation.

Installing an application template

This procedure creates a provisioning job to install the application template in each remote station in the job.

Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The **Niagara Network Job Builder** (one-time) or **Niagara Network Prototype View** (reoccurring job) is open.

Installing an application template is always a one-time job.

NOTE: Only one application template can be installed on a station.

Step 1 To copy the template and Excel files to the Supervisor, click the **Copy Templates** button at the bottom of the **Niagara Network Job Builder**.

A **File Chooser** window opens.

Step 2 Select the application (*.napl) and Excel (*.xls) files to copy to the Supervisor and click **OK**.

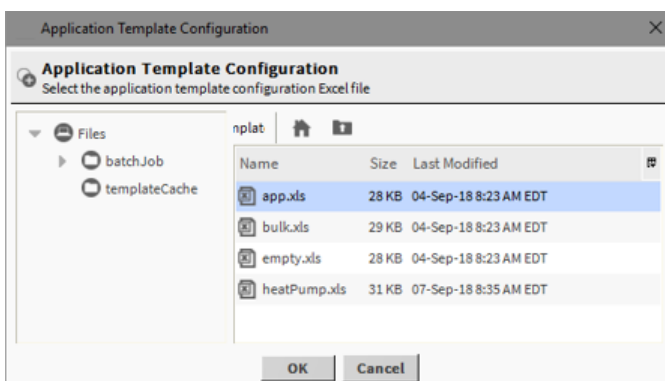
The system puts these files into the ^templateCache folder on the Supervisor.

Step 3 In the top pane, **Provisioning steps to run**, click **(Add)**.

The **New Job Step** window opens.

Step 4 Click the **Install Application Template** step and click **OK**.

A **File Chooser** window opens so you can select the Excel file to be used for deploying templates to each remote station in the job.



- Step 5 Navigate to the `^templateCache` folder on the Supervisor, select the Excel file to use for this step, and click **OK**.

The system adds the use of the Excel file as a step in the job sequence and displays its file path ORD.

- Step 6 In the bottom pane, **Stations to include in the job**, click  (Add).

The **Add Device** window opens.

- Step 7 Select the stations and click **OK**.

- Step 8 Re-initiate the provisioning job, review your choices and click the **Run Now**.

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

Upgrading an application template

This procedure creates a provisioning job to upgrade an existing application template on each remote device station in the job.

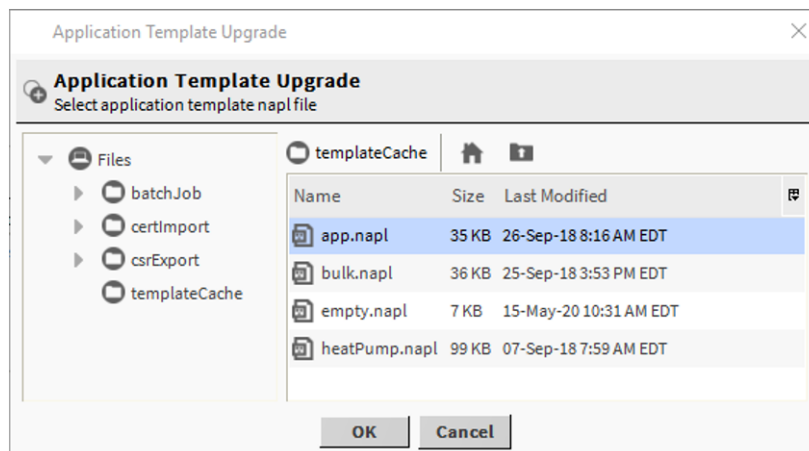
Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The **Niagara Network Job Builder** (one-time) or **Niagara Network Prototype View** (reoccurring job) is open. A new version of an installed application template is copied to the Supervisor's `templateCache` folder.

- Step 1 In the top pane, **Provisioning steps to run**, click  (Add).

The **New Job Step** window opens.

- Step 2 Select the **Upgrade Application Template** step and click **OK**.

The system opens a **File Chooser** window so you can select the application template (*.napl) file to be used for upgrading application templates to each remote station in the job.



- Step 3 Navigate to the `templateCache` folder and select the application template file to use for this step.

The system adds the step in the job sequence and displays the template file path ORD to use.

- Step 4 In the bottom **Stations to include in the job**, click  (Add).

The **Add Device** window opens.

- Step 5 Select the stations and click **OK**.


- Step 6 Review your choices and click **Run Now**.

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

Updating template or application configurations

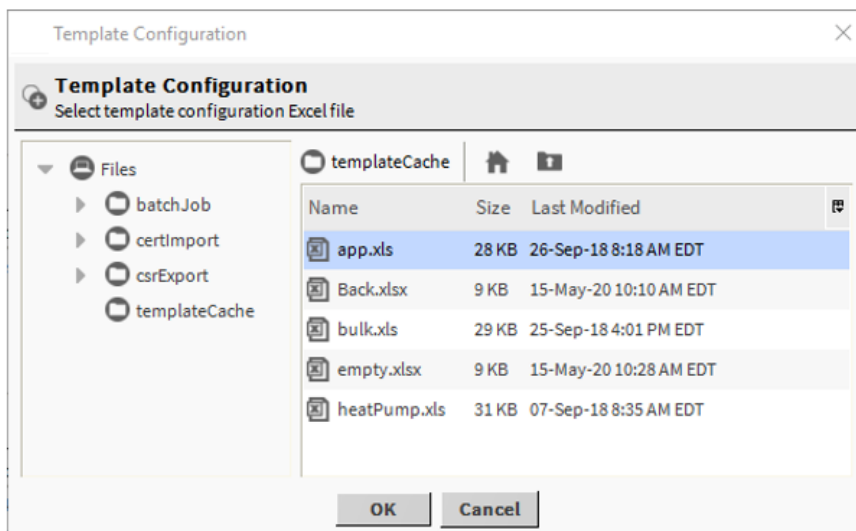
This procedure creates a provisioning job to update the configuration values of an existing component, device, or application template on each remote device station in the job.

Prerequisites: The **BatchJobService** is available under **Services**. The **ProvisioningNwExt** component is available under the **NiagaraNetwork**. The **Niagara Network Job Builder** (one-time) or **Niagara Network Prototype View** (reoccurring job) is open. An updated Excel spreadsheet containing new configuration values is copied to the Supervisor's `templateCache` folder.

Step 1 In the top pane, **Provisioning steps to run**, click  (Add).
the **New Job Step** window opens.


Step 2 Select the **Update Template or Application Configuration** step and click **OK**.

The system opens a **File Chooser** window so you can select the Excel file to be used for updating configurations to each remote station in the job.



Step 3 Navigate to the `templateCache` folder and select the Excel file to use for this step.

The system adds the step in the job sequence and displays the Excel file path ORD to use.

Step 4 In the bottom pane, **Stations to include in the job**, click  (Add).
The **Add Device** window opens.

Step 5 Select the stations and click **OK**.

Step 6 Review your choices and click **Run Now**.

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

Provisioned application templates can be updated at some later time using the Upgrade Application job step. First, use the Copy Template button to copy the upgraded `.napl` file to the Supervisor station. Then set up a provisioning prototype to include the Upgrade Application job step and the stations to be updated. Save and Run the prototype to apply the upgraded application.

The results will be the same as if you upgraded each application individually with each station's Upgrade Application Template menu item. Initial configurations on the application will be reapplied to the upgraded application.

Chapter 8 Components

Topics covered in this chapter

- ◆ Components in the batchJob module
- ◆ Extension components in the provisioningNiagara module
- ◆ Supporting components in the provisioningNiagara module
- ◆ Job step components in the provisioningNiagara module

Components include services, folders and other model building blocks associated with a module. You may drag them to a property or wire sheet from a palette.

Descriptions included in the following topics appear as context-sensitive help topics when accessed by:

- Right-clicking on the object and selecting **Views→Guide Help**
- Clicking **Help→Guide On Target**

Components in the batchJob module

There are two components in the batchJob module: **BatchJobService** and **ThreadPoolJobQueue**.

batchJob-BatchJobService

This service provides job control functions for provisioning jobs, including sending each provisioning job (as a batch job) to be run by the station's **JobService**.

The **BatchJobService** is required in the Supervisor station to facilitate provisioning. After copying it from the **batchJob** or **provisioningNiagara** palettes into the Supervisor's **Services** folder, you do not normally interface with it, apart from specifying the station's alarm class to use for provisioning alarms.

The **BatchJobService** requires the Supervisor station to also have the **HistoryService** and **JobService**, otherwise the **BatchJobService** is 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 is in *fault*.

Figure 7 BatchJobService properties

BatchJobService (Batch Job Service)	
Status	{ok}
Fault Cause	
Enabled	● true
Job Queue	Thread Pool Job Queue
Max Threads	1 [1-max]
Alarm Class	Default Alarm Class
Summary Manager Type	batchJob HistoryJobSummaryManager
Max Provisioning Threads	2

Property	Value	Description
Status	read-only	<p>Reports the condition of the entity or process at last polling.</p> <p><code>{ok}</code> indicates that the component is licensed and polling successfully.</p> <p><code>{down}</code> indicates that the last check was unsuccessful, perhaps because of an incorrect property, or possibly loss of network connection.</p> <p><code>{disabled}</code> indicates that the Enable property is set to false.</p> <p><code>{fault}</code> indicates another problem. Refer to Fault Cause for more information.</p>
Enabled	true (default) or false	<p>Activates (<code>true</code>) and deactivates (<code>false</code>) use of the object (network, device, point, component, table, schedule, descriptor, etc.).</p>
Job Queue	thread pool job queue	<p>Manages the submission of provisioning batch jobs by using a thread pool, to ensure that 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.</p> <p>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.</p>
Alarm Class	drop-down list	<p>Specifies the alarm routing option for the component.</p> <p>Replace provides a selection list of a local alarm classes, from which to select one to use for all alarms received from this device.</p> <p>Use Existing routes alarms from this remote station to any matching alarm class, that is, one with an identical name as that in each alarm record. If the program finds no local matching alarm class, it uses the station's default alarm class.</p> <p>Prepend adds leading text (as specified) to the incoming alarm class string, then routes it to any local matching alarm class in the station.</p> <p>Append adds trailing text (as specified) to the incoming alarm class string, then routes it to any local matching alarm class in the station.</p> <p>The check boxes at the top of the Niagara Network Job Builder and Niagara Network Prototype View configure when to generate an alarm for a provisioning job that failed.</p>

Property	Value	Description
Summary Manager Type	read-only (defaults to batchJob: HistoryJobSummary-Manager)	Reports the type of the class that manages summary logs. Developers who extend the BatchJobService can set this value to a custom manager that implements the BJJobSummaryManager interface.
Max Provisioning Threads	number from 1 through 10 (defaults to 1)	Starting in EC-Net 4 v4.7, provisioning allows job steps to run in parallel across multiple target stations. This property defines the maximum number of parallel jobs that can be run for a single provisioning job step. If there are more target stations than Max Provisioning Threads for a provisioning job, the system queues the remaining station jobs and runs them when the running jobs finish.

batchJob-ThreadPoolJobQueue

This frozen container slot under the **BatchJobService** has only one 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 best left at default, as provisioning threads can be resource intensive in a Supervisor station.

NOTE: Only after determining that 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.

Extension components in the provisioningNiagara module

These components provide a variety of services. They extend the functionality of the components to which they are added.

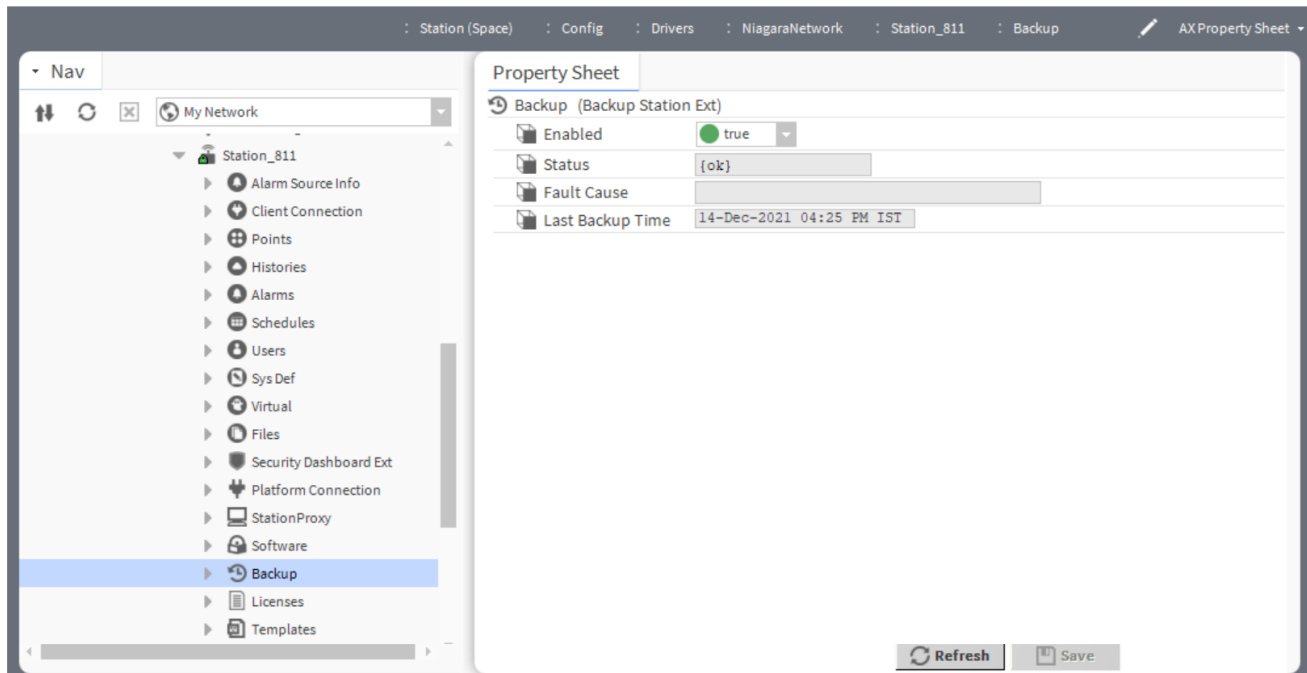
The following extension components are automatically added to every **NiagaraStation** under the Supervisor's **NiagaraNetwork**, providing the Supervisor also has the **ProvisioningNwExt** (**ProvisioningNiagara-NetworkExt**) under its **NiagaraNetwork**. These provisioning extensions are listed here.

- **BackupStationExt**
- **LicenseStationExt**
- **SoftwareStationExt**
- **TemplateStationExt**

BackupStationExt (provisioningNiagara-BackupStationExt)

This component supports the **Backup Stations** step in provisioning jobs. It is one of the seven device extensions that are automatically added to every station under the Supervisor's **NiagaraNetwork**. By default the extension is enabled.

This component requires that the **ProvisioningNwExt** component be resident under its **NiagaraNetwork**. No special views exist for this extension.

Figure 8 Backup Station Ext properties

To access these properties, expand **Config**→**Drivers**→**NiagaraNetwork**→**NiagaraStation** and double-click **Backup**. **NiagaraStation** represents your station name.

Property	Value	Description
Enabled	true or false; defaults to true	Activates (true) and deactivates (false) use of the object (network, device, point, component, table, schedule, descriptor, etc.). To prevent the use of the Backup Station step in any provisioning job created for this station, change this value to false.
Status	read-only	Reports the condition of the entity or process at last polling. {ok} indicates that the component is licensed and polling successfully. {down} indicates that the last check was unsuccessful, perhaps because of an incorrect property, or possibly loss of network connection. {disabled} indicates that the Enable property is set to false. {fault} indicates another problem. Refer to Fault Cause for more information.
Fault Cause	read-only	Indicates the reason why a system object (network, device, component, extension, etc.) is not working (in fault). This property is empty unless a fault exists.
Last Backup Time	read-only	Displays the date and time of last backup occurred.

Start Backup action

When invoked, the Start Backup action immediately submits a provisioning job that contains a single Backup Stations step against this station. To schedule the backup to run at a specific time or at a specific interval for this

particular station, you can link a **TriggerSchedule** output to this action. An even better practice is to add a job prototype component to the station and configure it to perform the backup using its own **TriggerSchedule** to set the time. This method allows you to configure job retention, which provides automatic backup job disposition.

NOTE: Provisioning backups work using either the local BackupService or the CloudBackupService. When the CloudBackupService is installed it provides all the functions of the local BackupService. You can make successful backups using either the **Backup Manager** view to create manual backups or using the **Backup Stations** step in provisioning jobs in the Supervisor. Backups will contain the expected files based on the CloudBackupService exclusions. Backups created by a provisioning job prototype follow the configured job disposition settings and will be deleted when new backups are created.

LicenseStationExt (provisioningNiagara-LicenseStationExt)

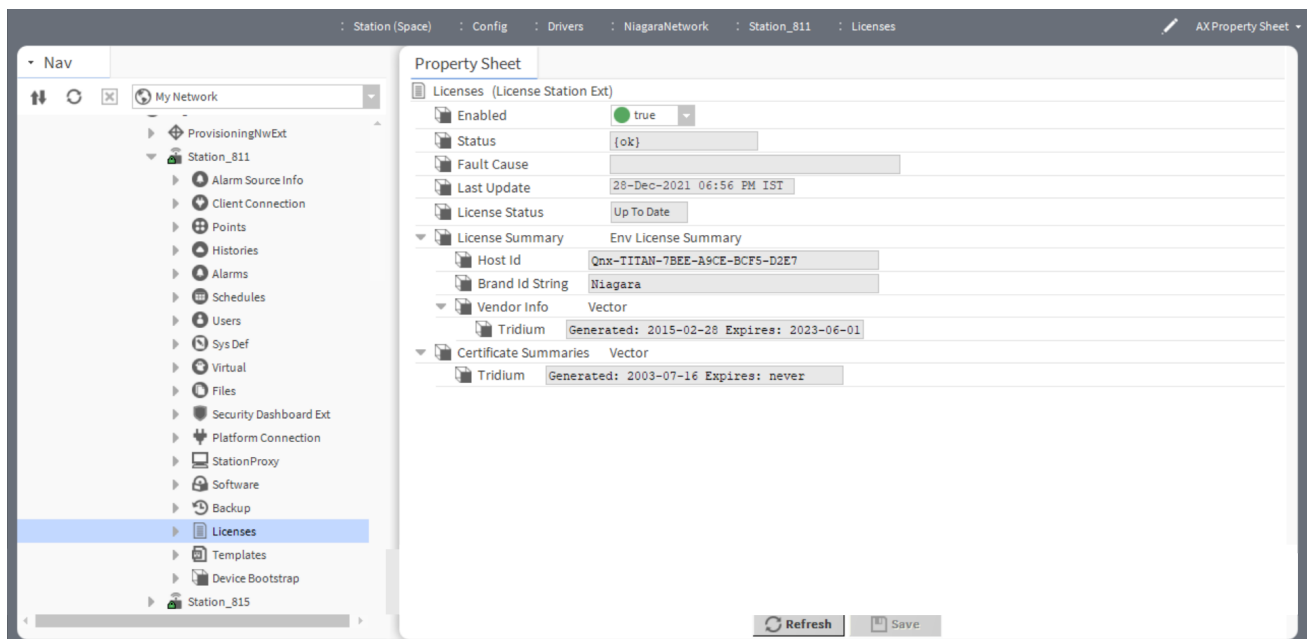
This component provides the **Niagara Network Job Builder** and the **Property Sheet** that configures it. It is one of seven device extensions the system adds automatically to every station under the Supervisor's **NiagaraNetwork**.

By default this component is enabled. The properties you configure on its **Property Sheet** update the values reported in the **ProvisioningNwExt's Licenses** view. No special views exist for this extension. Adding this component to a job triggers a pop-up window for a potential brand change. To change the license brand, you enter the new brand ID.

When initiated by the Supervisor, this component gathers information about the licenses installed on each target host, accesses the online licensing server (in one message) to see if the licenses are up-to-date, and, if a host's license is out-of-date, updates each license in the target station's host, and in the Supervisor's local license database. For brand updates, this component looks for a license matching the new Brand ID in the local license database and the online licensing server for each target platform Host ID in the job. If it finds a license, it deletes the existing license and copies the new license to the platform. It also copies license vendor certificates to the target platforms if needed.

NOTE: If the Supervisor is not configured for Internet connectivity, this component uses only its local license database to compare against licenses installed in the target host(s). If a host's license is out-of-date, the step updates the license in the target host(s).

Figure 9 License Station Ext properties



To access these properties, expand **Config**→**Drivers**→**NiagaraNetwork**→**NiagaraStation**, and double-click **Licenses**. **NiagaraStation** represents your station name.

Properties

Property	Value	Description
Enabled	true (default) or false	Activates (<code>true</code>) and deactivates (<code>false</code>) use of the object (network, device, point, component, table, schedule, descriptor, etc.). To prevent a provisioning job from running the Update Licenses step against this station, set this property to <code>false</code> .
Status	read-only	Reports the condition of the entity or process at last polling. {ok} indicates that the component is licensed and polling successfully. {down} indicates that the last check was unsuccessful, perhaps because of an incorrect property, or possibly loss of network connection. {disabled} indicates that the Enable property is set to <code>false</code> . {fault} indicates another problem. Refer to Fault Cause for more information.
Fault Cause	read-only	Indicates the reason why a system object (network, device, component, extension, etc.) is not working (in fault). This property is empty unless a fault exists.
Last Update	date and time	Reports the date and time that a provisioning job updated the license last. This property reports null if the license has not yet been updated this way.
License Status	Up-to-Date, Unknown, Expired	Indicates the state of the license: <ul style="list-style-type: none"> Up-to-Date reports that the license is the most recent available. Unknown indicates that provisioning has never updated the license in this station. License Summary provides a container slot to hold properties describing the license(s). The information held by this slot includes the unique Host ID for the platform, brand identifiers, and other vendor license information with timestamps to indicate when each license was originally generated, and its expiration date (if ever).
Certificates Summaries	text	Provides a container slot to hold properties that describe installed certificate(s), including when each certificate was originally generated, and its expiration date (if ever).

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.

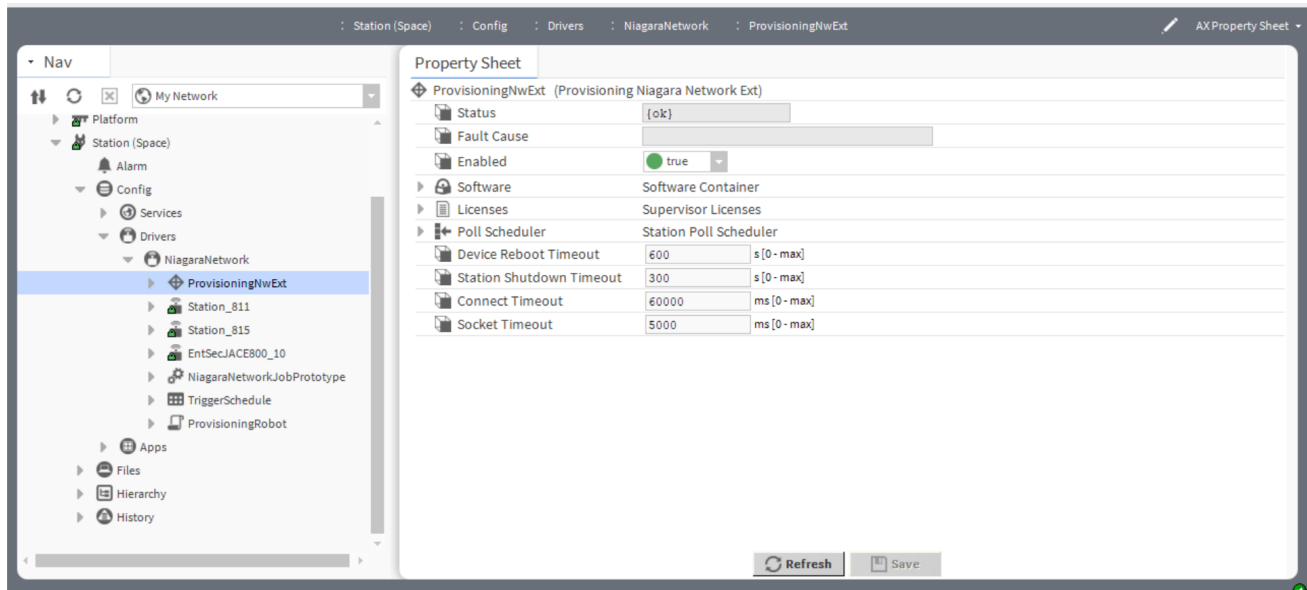
ProvisioningNiagaraNetworkExt (provisioningNiagara-ProvisioningNiagaraNetworkExt)

This component supports the **ProvisioningNwExt** extension that runs in a Supervisor station. It is the core provisioning component and is required to enable the provisioning of hosts represented by stations under the Supervisor's **NiagaraNetwork**.

ProvisioningNiagaraNetworkExt provides a central spot to perform provisioning tasks, which the system dispatches as batch jobs by the station's **BatchJobService**, and executed by its **JobService**.

This component has a number of frozen slots, both of which are containers (some with special views), as well as a few properties. The default view is the **Niagara Network Job Builder**. A **Niagara Network Job List** view is also available.

Figure 10 Provisioning Network Ext properties



To access these properties, expand **Config**→**Drivers**→**NiagaraNetwork**, and right-click the **ProvisioningNwExt**→**Views**→**AX PropertySheet**.

Type	Value	Description
Status	read-only	Activates (<code>true</code>) and deactivates (<code>false</code>) use of the object (network, device, point, component, table, schedule, descriptor, etc.).
Enabled	<code>true</code> or <code>false</code> ; defaults to <code>true</code>	<p>Reports the condition of the entity or process at last polling.</p> <p><code>{ok}</code> indicates that the component is licensed and polling successfully.</p> <p><code>{down}</code> indicates that the last check was unsuccessful, perhaps because of an incorrect property, or possibly loss of network connection.</p> <p><code>{disabled}</code> indicates that the Enable property is set to <code>false</code>.</p> <p><code>{fault}</code> indicates another problem. Refer to Fault Cause for more information.</p>

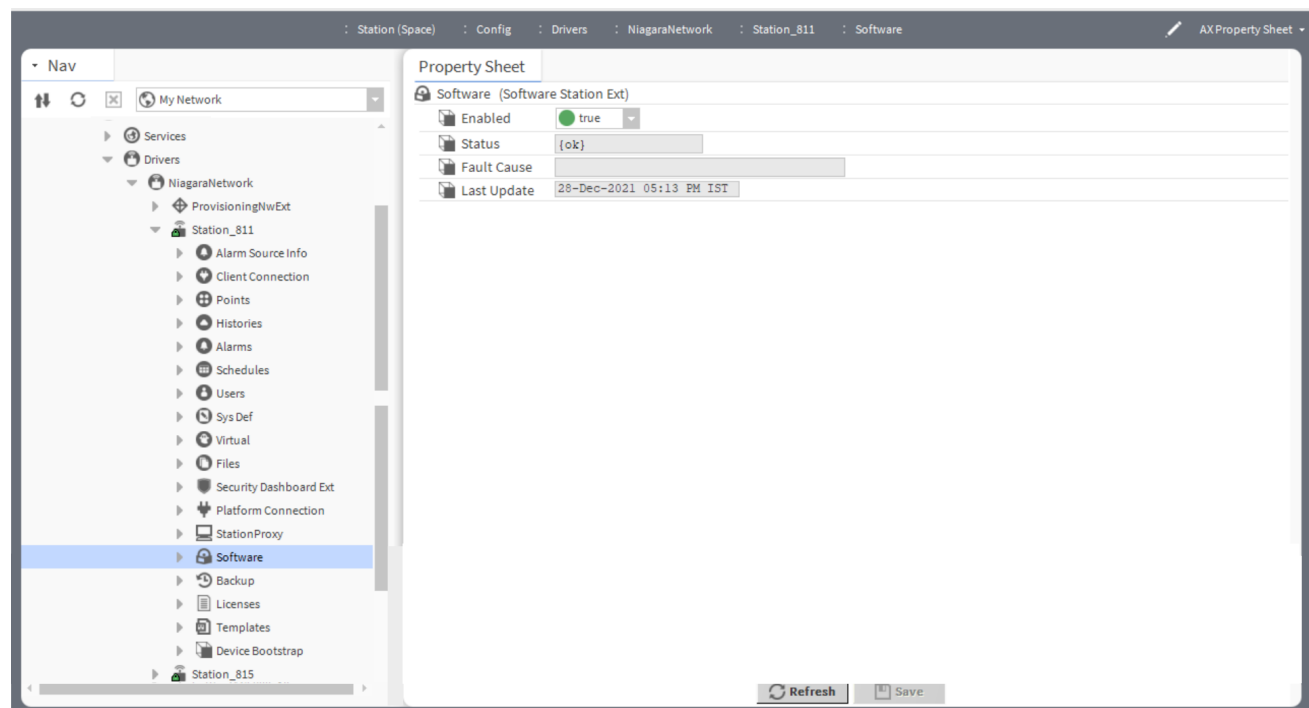
Type	Value	Description
Fault Cause	read-only	Indicates the reason why a system object (network, device, component, extension, etc.) is not working (in fault). This property is empty unless a fault exists.
Device Reboot Timeout	seconds(defaults to 600)	Specifies the maximum time in seconds that a provisioning job step waits while attempting to reconnect to a station that has been rebooted before setting the step status to <code>failed</code> and advancing to the next station. The default time is 600 seconds (10 minutes).
Station Shutdown Timeout	seconds(defaults to 300)	Specifies the maximum time in seconds a provisioning step waits between receiving a shutdown request and setting the step status to <code>failed</code> , followed by advancing to the next station. The default time is 300 seconds (5 minutes).
Connect Timeout	milliseconds (defaults to 60000)	Defines how long the provisioning job waits before going on to the next station when a connection to the current station fails.
Socket Timeout	milliseconds (defaults to 5000)	Defines how long the provisioning job waits before going on to the next station when the current station's socket fails.
Backup Schedule	frozen child component	<p>A standard TriggerSchedule with its trigger output pre-linked to the ProvisioningNwExt's <code>Start Backup</code> 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. To view its Trigger Scheduler view, double-click the component.</p> <p>Use of this Backup Schedule is not advisable on a larger enterprise system, where the Supervisor has many subordinate stations. Instead, use the job prototype components (NiagaraNetworkJobPrototype) copied from the provisioningNiagara palette.</p>
Software	frozen child container	Summarizes the software versions available on the Supervisor to be installed into stations.
Licenses	frozen child component	Provides views into the licenses installed in NiagaraStations under the NiagaraNetwork , as well as the Supervisor's license database.
Poll Scheduler	frozen child component	<p>Controls the polling of the StationProxy extensions by the ProvisioningNwExt component. This poll scheduler operates as in most driver networks. Polling only happens for station proxy objects that are subscribed (linked somewhere, or being viewed or charted, etc.).</p> <p>Often you can leave the poll configuration properties set at their default values. Otherwise, you may need to configure the StationProxy extensions (NiagaraStation provisioning extensions) to something other than the default, which is <code>Normal</code>.</p>
Device Reboot Timeout	seconds (defaults to 600)	Configures how long to wait before rebooting the provisioned station. This time prevents the remote station from hanging an entire provisioning job.
Station Shutdown Timeout	seconds (defaults to 300)	Configures how long to wait before shutting down the provisioned station. This time prevents the remote station from hanging an entire provisioning job.

Type	Value	Description
Connect Timeout	milliseconds (defaults to 60000)	Configures how long to wait before a failed connection attempt times out. This time prevents the remote station from hanging an entire provisioning job.
Socket Timeout	milliseconds (defaults to 5000)	Configures how long to wait before an attempt to connect to a socket times out. This time prevents the remote station from hanging an entire provisioning job.

SoftwareStationExt (provisioningNiagara-SoftwareStationExt)

This component supports the steps that a provisioning job can run on a station. By default this component is always enabled. Other functions provided by the station are available using the **Station Software View**.

Figure 11 Software Station Ext properties



To access these properties, expand **Config**→**Drivers**→**NiagaraNetwork**→**NiagaraStation** and right-click the **Software**→**Views**→**AX Property Sheet**. **NiagaraStation** represents your station name.

Property	Value	Description
Enabled	true (default) or false	Activates (<code>true</code>) and deactivates (<code>false</code>) use of the object (network, device, point, component, table, schedule, descriptor, etc.). Changing this property to <code>false</code> prevents the processing of any steps against this station. Enabled must be set to <code>true</code> to use the Station Software View and Supervisor Software Manager on this station.
Status	read-only	Reports the condition of the entity or process at last polling. <code>{ok}</code> indicates that the component is licensed and polling successfully.

Property	Value	Description
		<p>{down} indicates that the last check was unsuccessful, perhaps because of an incorrect property, or possibly loss of network connection.</p> <p>{disabled} indicates that the Enable property is set to false.</p> <p>{fault} indicates another problem. Refer to Fault Cause for more information.</p>
Fault Cause	read-only	Indicates the reason why a system object (network, device, component, extension, etc.) is not working (in fault). This property is empty unless a fault exists.
Last Update	date and time	Documents the last platform snapshot. This property is null if a platform snapshot has never occurred.

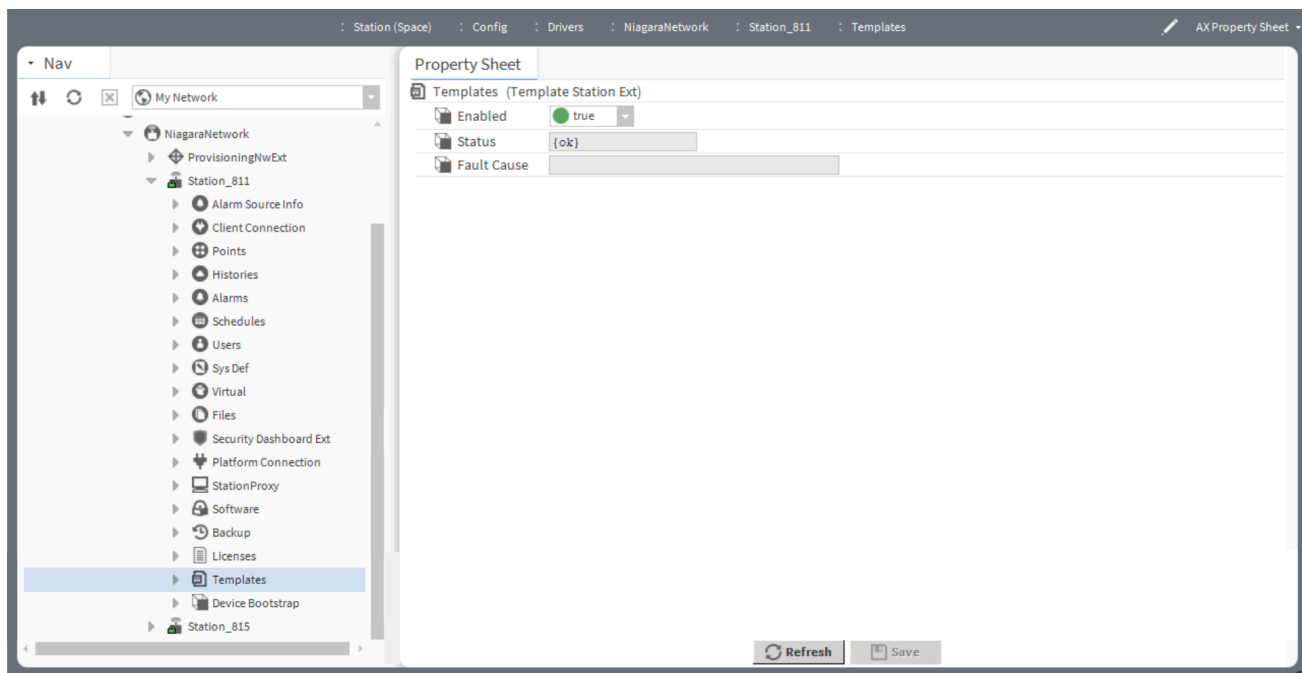
TemplateStationExt (provisioningNiagara-TemplateStationExt)

This component currently only supports a **Details** action, which will show information about component and device templates that have been deployed to the remote station.

These are the only template types managed by a station's Template Service (application and station templates are not). Invoking the **Details** action will print deployed template information to the Supervisor console.

This extension will provide additional functionality in future releases and template management capabilities are expanded.

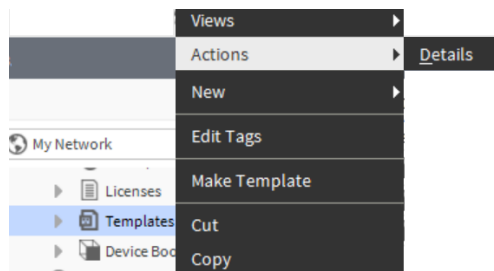
Figure 12 Templates Station Ext Properties



To access these properties, expand **Config**→**Drivers**→**NiagaraNetwork**→**NiagaraStation**, and double-click **Templates**. **NiagaraStation** represents your station name.

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

Figure 13 Action



To access this action, right-click the **Templates**→**Actions**→**Details**. The **Details** action displays information about templates deployed to remote station.

UpdateStationExt (provisioningNiagara-UpdateStationExt)

This component supports the **Update Licenses** step. It updates the software licenses in each remote host.

You add this component in the top pane (**To run for each station**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

Supporting components in the provisioningNiagara module

The purpose of these components is to provide support and services to other components, job steps, and views.

InstallableSummary (provisioningNiagara-InstallableSummary)

This component facilitates the creation of a list of software files under the `!sw` directory of the Supervisor host, and displays the files under the **SoftwareContainer** of the **NiagaraNetwork's ProvisioningNwExt**. This container contains all installed software modules, including modules added to the registry after the initial scan. To manage this container you use the **Supervisor Software Manager**.

Each installable summary contains one or more specification objects for each version in the software registry. Each object is represented as an **InstallableSpec** component. At station startup, this service starts a thread that scans the software registry and populates the **SoftwareContainer**.

Direct children of the **SoftwareContainer** 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 **SoftwareContainer** has but a single frozen property: **loaded**—a boolean slot that indicates if the startup thread has finished scanning the registry (by default, it is hidden).

Your key interface to the **SoftwareContainer** is its default view: the **Supervisor Software Manager**.

InstallableSpec (provisioningNiagara-InstallableSpec)

These components are children of **InstallableSummary** components. They occupy the lowest level under the **SoftwareContainer** in the **NiagaraNetwork**’s **ProvisioningNwExt** component.

InstallableSpecs are version-specific, and describe each software item that can be installed on a remote host, including version number, dependencies and other data.

InstallStep (provisioningNiagara-InstallStep)

This component supports the Install backupdist steps the system creates when you copy an existing backup .dist file into the middle job step pane (To run for each station) in either the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

This step differs from the Install Software steps supported by **InstallBySpecStep**. It does not require selection by version number.

There is no EC-Net 4 Pro factory option for this component, so it is not available in the list of selectable provisioning steps.

NiagaraNetworkJob (provisioningNiagara-NiagaraNetworkJob)

This component facilitates the **Niagara Network Job View**, which lists the job stages with steps that the system performed on each host in the job list.

Jobs are found under the Supervisor station’s **JobService**. The default view of this component is the **Niagara Network Job View**.

NiagaraNetworkJobPrototype (provisioningNiagara-NiagaraNetworkJobPrototype)

This component is a persisted component that represents a provisioning job—meaning a specific set of steps to be performed on a given list of host stations. It differs from a one-time job in that the prototype job’s specific set of steps are to be performed regularly on its given list of host stations.

Because of available job retention policies, **NiagaraNetworkJobPrototype** components 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).

You copy this component from the **provisioningNiagara** palette or duplicate it from another **NiagaraNetworkJobPrototype**. To specify some future (and possibly periodic) execution, you link the out (**Trigger**) slot of a standard **TriggerSchedule** to the component’s **Submit Job** action. To schedule the job to run immediately, you click **Run Now**.

Like other components, you can duplicate, modify, and reuse these jobs as needed in the Supervisor station.

This component’s default view is the **Niagara Network Prototype View**. The **Prototype Job List** view, which is also available on this component, provides job retention options.

PlatformConnection (provisioningNiagara-PlatformConnection)

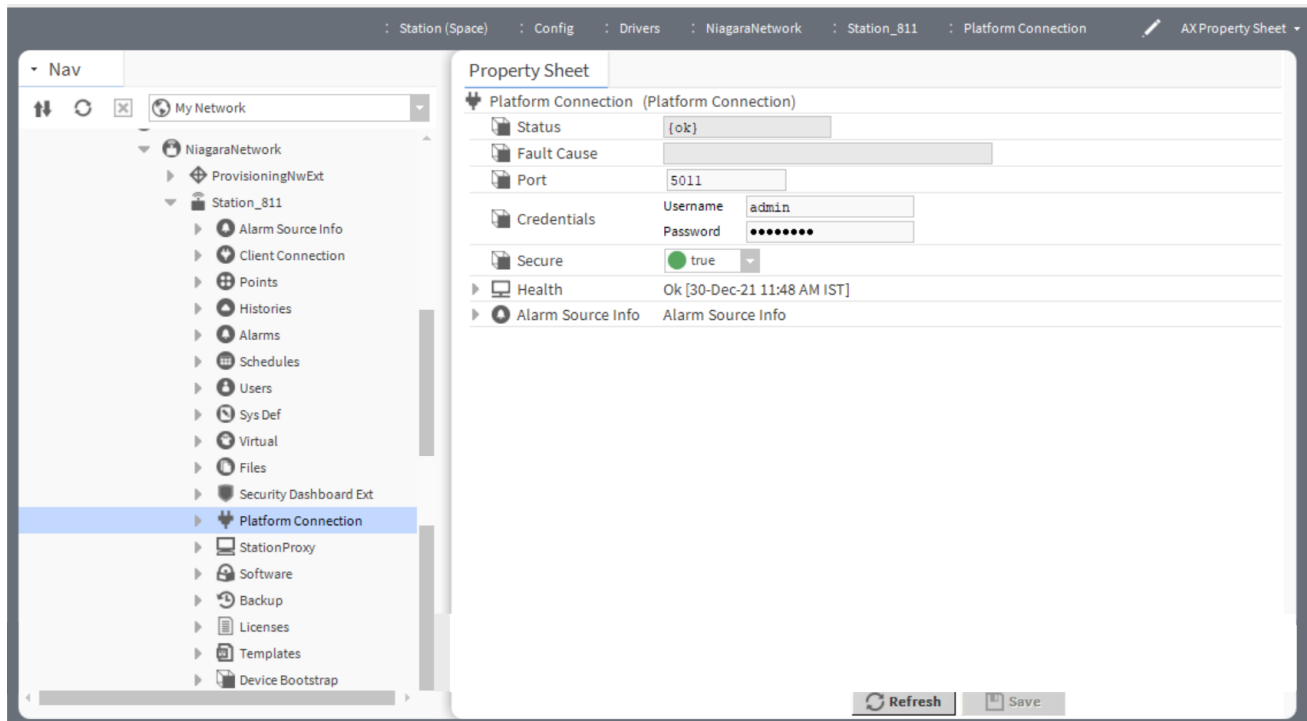
This component is used by the **ProvisioningNwExt** to poll each station.

To configure each host’s newly-created PlatformConnection component, you access the Platform Connection property sheet by right-clicking **ProvisioningNwExt** and clicking **Views→Property Sheet**. Then you specify

both the connection's HTTP port and platform credentials (username and password). No special view, other than the property sheet, exists for this component.

Most provisioning jobs use secure platform connections (platformssl for versions 3.7 and 3.8; platformtls for versions 4.0 and later).

Figure 14 Platform Connection Properties



To access these properties, expand **Config**→**Drivers**→**NiagaraNetwork**→**NiagaraStation**, and double-click **Platform Connection**. **NiagaraStation** represents your station name.

Property	Value	Description
Status	read-only	Reports the condition of the entity or process at last polling. {ok} indicates that the component is licensed and polling successfully. {down} indicates that the last check was unsuccessful, perhaps because of an incorrect property, or possibly loss of network connection. {disabled} indicates that the Enable property is set to false. {fault} indicates another problem. Refer to Fault Cause for more information.
Fault Cause	read-only	Indicates the reason why a system object (network, device, component, extension, etc.) is not working (in fault). This property is empty unless a fault exists.
Port	number (defaults to 3011 for a connection that is not secure; defaults to	Identifies the port on which the platform daemon in the station's host is listening. Displays as <code>Platform Port</code> in the Add or Edit windows when working in the Station Manager view.

Property	Value	Description
	5011 for a secure connection)	
Credentials	Username and password	Specifies the credentials used for a platform connection to the host's running station. Credentials display as <code>Platform User</code> and <code>Platform Password</code> in the Add or Edit windows when working in the Station Manager view.
Secure	<code>true</code> (must be <code>true</code> for EC-Net 4 v4.0 and later) or <code>false</code> (default for EC-Net ^{AX} -3.7 and EC-Net ^{AX} -3.8)	Specifies if a secure connection should be used to the host. Displays as <code>Secure Platform</code> in the Add or Edit dialog in the Station Manager view.
Health	text	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.

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. To test their validity, issue this action after entering port and credentials properties.

ProvisioningRobot (provisioningNiagara-ProvisioningRobot)

This special program component can be customized and saved on the Supervisor station, then selected as the source robot in a **Run Robot** step when configuring a provisioning job.

You can find the **ProvisioningRobot** in the `provisioningNiagara` palette. The default view for this component is the **Provisioning Robot Editor**, which provides the ability to edit and compile the baja code for running in each host.

NOTE: Super user permissions are required to add and edit robot components.

As copied from the `provisioningNiagara` palette, a **ProvisioningRobot** comes pre-configured with some essential wrapper lines of code. Also included is placeholder do-something code that may prove useful in some cases.

The Supervisor's **JobService** runs the job that contains the robot code. When it comes to the robot step, the **ProgramService** in each host station (that is included in the job) executes the robot's program code in the station.

SoftwareContainer (provisioningNiagara-SoftwareContainer)

This component supports the **Software** component under the **ProvisioningNwExt** extension. It provides visibility into the Supervisor's software registry that contain the software files (stored under the `!sw` directory) that can be installed on a remote host.

The children of this container are summary objects for named, typed, software files, for example, modules and platform distributions (dist) represented as `InstallableSummary` components. The default view of this component is the **Supervisor Software Manager**.

StationPollScheduler (provisioningNiagara-StationPollScheduler)

This component controls the polling of **StationProxy** extensions by the **ProvisioningNwExt** of the **Niagara-Network**. As a container slot under this network extension, the poll scheduler operates as in most driver networks.

Polling only happens for **StationProxy** extensions that are subscribed, that is, linked somewhere or being viewed or charted, etc.

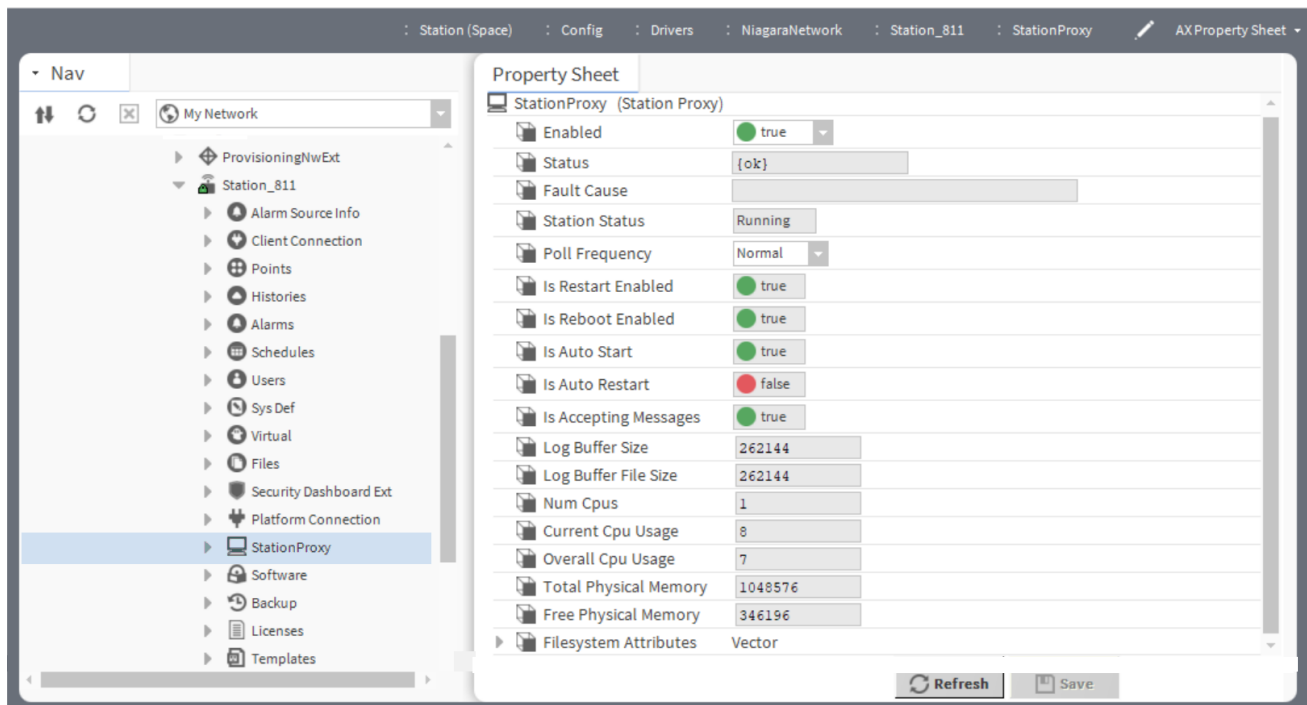
In most cases, you can leave the poll scheduler properties at their default settings. If needed, you may change the **Poll Frequency** property of the **StationProxy** slots.

StationProxy (provisioningNiagara-StationProxy)

This component provides platform administration functions like those available when you open a direct platform connection in EC-Net 4 Pro, using the **Station Director** and **Platform Administration** views. It also provides a number of actions for station control functions.

This component's default view is the **Provisioning Station Director**. The **StationProxy** component provides other special views, including the default view (**Provisioning Station Director**) as well as a **Station Job List**.

Figure 15 Station Proxy Properties



To access these properties, expand **Config**→**Drivers**→**NiagaraNetwork**→**NiagaraStation**, and right-click the **StationProxy**→**Views**→**AX Property Sheet**. **NiagaraStation** represents your station name.

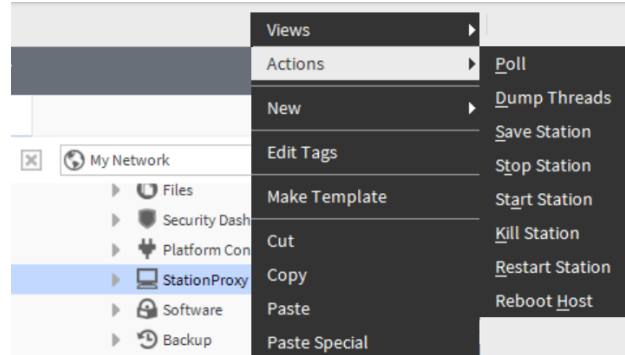
By default, this component is enabled to allow polling from the **ProvisioningNwExt** for values of the extension's properties.

Property	Value	Description
Enabled	true (default) or false	Activates (<code>true</code>) and deactivates (<code>false</code>) use of the object (network, device, point, component, table, schedule, descriptor, etc.). To prevent polling by the ProvisioningNwExt , set this property to <code>false</code> . To use the special views on the Station Proxy extension, namely the Provisioning Station Director and Station Job List , Enabled must be <code>true</code> .
Status	read-only	Reports the condition of the entity or process at last polling. {ok} indicates that the component is licensed and polling successfully. {down} indicates that the last check was unsuccessful, perhaps because of an incorrect property, or possibly loss of network connection. {disabled} indicates that the Enable property is set to <code>false</code> . {fault} indicates another problem. Refer to Fault Cause for more information.
Fault Cause	read-only	Indicates the reason why a system object (network, device, component, extension, etc.) is not working (in fault). This property is empty unless a fault exists.
Station Status	read-only	Reflects one of the following values: <ul style="list-style-type: none"> <code>Idle</code> — Station is not currently running, and can be started without a reboot. <code>Starting</code> — Station process is running, but has not completed its startup sequence. <code>Running</code> — Station is running. <code>Stopping</code> — Station is in process of shutting down, but its process is still alive. <code>Halted</code> — Station is not currently running, and the host must be rebooted before it can start. <code>Unknown</code> — StationProxy status is <code>disabled</code> or <code>fault</code>. As a result, 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).
Is Restart Enabled	read-only	If <code>true</code> , the station you can restart the station without a reboot of its host platform (such as with Win32-based platforms).
Is Reboot Enabled	read-only	If <code>true</code> , the host's platform daemon is capable of (and allows) rebooting of the host.
Is Auto Start	read-only	If <code>true</code> , the station restarts automatically after the host reboots.
Is Auto Restart	read-only	If a station terminates with a failure exit code and this property is <code>true</code> , the host restarts (or reboots) if Is Restart Enabled = <code>false</code>).

Property	Value	Description
Is Accepting Messages	read-only	If <code>false</code> (unlikely), thread dumps, station saves, and graceful shutdown are not possible using the platform daemon.
Log Buffer Size	read-only	Configures the size (in bytes) of the buffer used by the platform daemon to hold the console output.
Log Buffer File Size	read-only	The maximum size of the console.txt file (in bytes) that the platform daemon saves console output to when the station stops.
Num Cpus	read-only	The number of CPUs on the host running the station.
Current Cpu Usage	read-only	The percentage of time the CPU(s) have been in use in the last second.
Overall Cpu Usage	read-only	The percentage of time the CPU(s) have been in use since the platform daemon started.
Total Physical Memory	read-only	The total KB of physical RAM on the station's host.
Free Physical Memory	read-only	The KB of available physical RAM on the station's host.
File System Attributes	read-only	Free space statistics for each file system on station's host.

Actions

Figure 16 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

Action	Description
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. 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	Supervisor requests the platform daemon to start the station. This action is applicable only if current station status is idle.
Kill Station	Supervisor requests for the station to terminate immediately, without graceful shutdown.

Action	Description
Restart Station	Depending on Is Restart Enabled value, causes one of the following: <ul style="list-style-type: none"> If Is Restart Enabled is <code>false</code> the station's host is rebooted. If Is Restart Enabled is <code>true</code> the station is stopped gracefully, then restarted again.
Reboot Host	Depending on Is Reboot Enabled value, causes one of the following: <ul style="list-style-type: none"> If Is Reboot Enabled is <code>false</code> nothing happens. If Is Reboot Enabled is <code>true</code>, the Supervisor requests for the platform daemon to shut down gracefully, then reboot the host.

SupervisorLicenses (provisioningNiagara-SupervisorLicenses)

This component supports the **Licenses** slot under the **NiagaraNetwork**'s **ProvisioningNwExt** component. Among the **ProvisioningNwExt** slots, the **Licenses** slot provides two views of the licenses related to the **NiagaraNetwork**.

- The **Network License Summary** is the default view for the slot. It lists NiagaraStations and their host IDs.
- The **Supervisor License Manager** provides a window into the Supervisor's local license database (under its `!licenses/db` subdirectory).

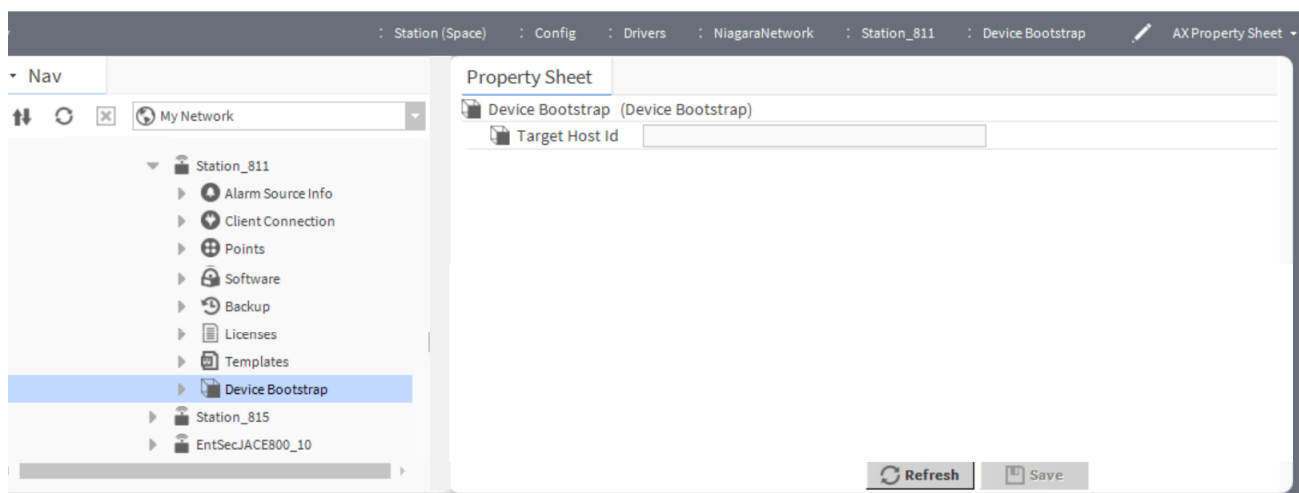
This slot has no direct component children and only a single property: **Allow License Server Access**, which is `true` by default.

DeviceBootstrap (provisioningNiagara-DeviceBootstrap)

This component sets up the target host Id was the bootstrap mode.

It is one of the seven device extensions that are automatically added to every station under the Supervisor's **NiagaraNetwork**. By default the extension is enabled.

Figure 17 Device Bootstrap property



To access these properties, expand **Config**→**Drivers**→**NiagaraNetwork**→**NiagaraStation**, and double-click the **Device Bootstrap**. **NiagaraStation** represents your station name.

Property	Value	Description
Target Host Id	Alpha numeric code	Sets up the target host Id was the bootstrap mode.

Job step components in the provisioningNiagara module

These components are job steps that you can run as part of a provisioning job using the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

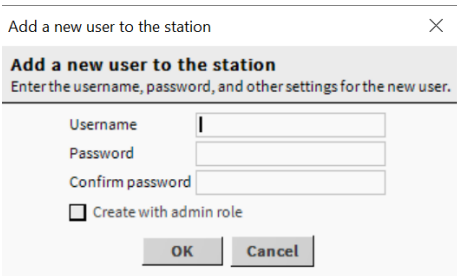
You add this job step components in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

Add Station User

This component supports the **Add Station User** step. This step adds a named user to the **UserService** for each station in the provisioning job.

You add this step to the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

Figure 18 Add a new user to the station



Add a new user to the station

Add a new user to the station

Enter the username, password, and other settings for the new user.

Username

Password

Confirm password

☐ Create with admin role

OK

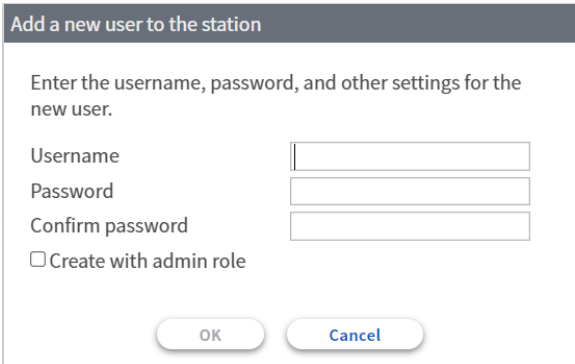
Cancel

Property	Value	Description
Username	text	Defines a Username .
Password	text	Defines a Password for the Username .
Confirm Password	text	Verifies the entered password for the Username
Create with admin role	check box	Adds new admin role to new station user.

HTML5 Add Station User

In EC-Net 4.14 and later, there is added browser support for **Add Station User** step. This step adds a named user to the **UserService** for each station in the provisioning job. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 19 Add a new user to the station



Add a new user to the station

Enter the username, password, and other settings for the new user.

Username

Password

Confirm password

☐ Create with admin role

OK

Cancel

To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click  and double-click **Add Station User** or select **Add Station User** and click **OK**.

You add this step to the top pane (**To run for each station**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

Property	Value	Description
Username	text	Defines a Username .
Password	text	Defines a Password for the Username .
Confirm Password	text	Verifies the entered password for the Username
Create with admin role	check box	Adds new admin role to new station user.

Backup Stations

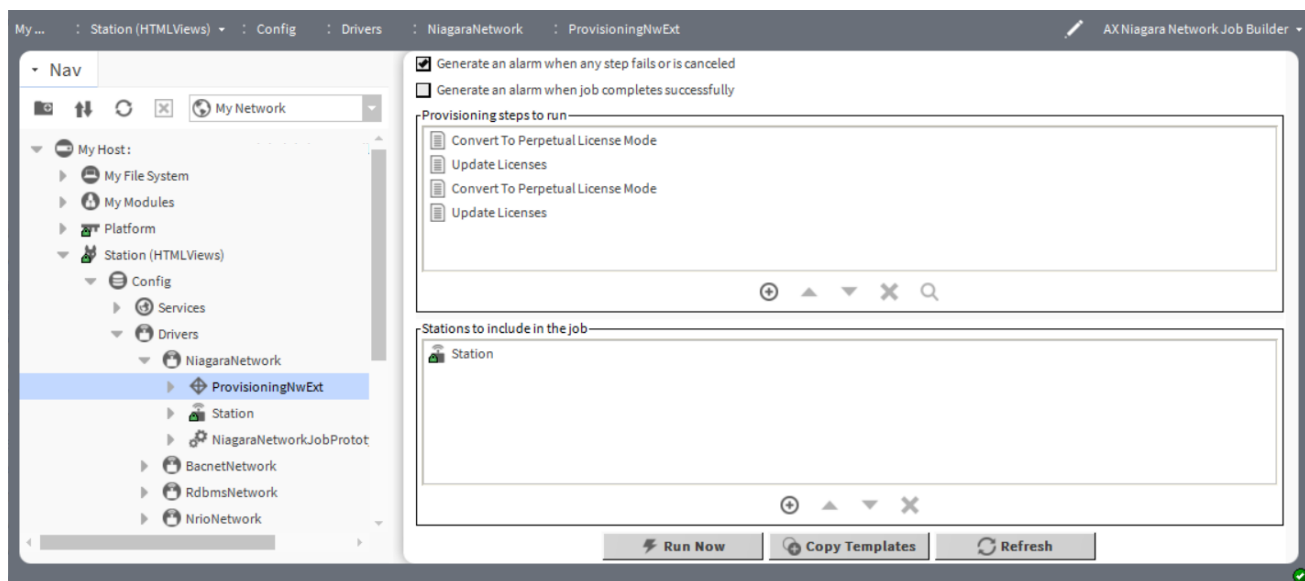
This component enables the **Backup Station** steps used to make an online backup for each specified station (provided the station is running), or if it is not running, an offline backup. In either case, the backup .dist file for each station is stored on the Supervisor



You add this step to the top pane (**To run for each station**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

Convert To Perpetual License Mode

In EC-Net 4.14 and later, there is added browser support that converts all the stations' licensing mode from subscription to perpetual. The HTML 5 version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 20 Convert To Perpetual License Mode



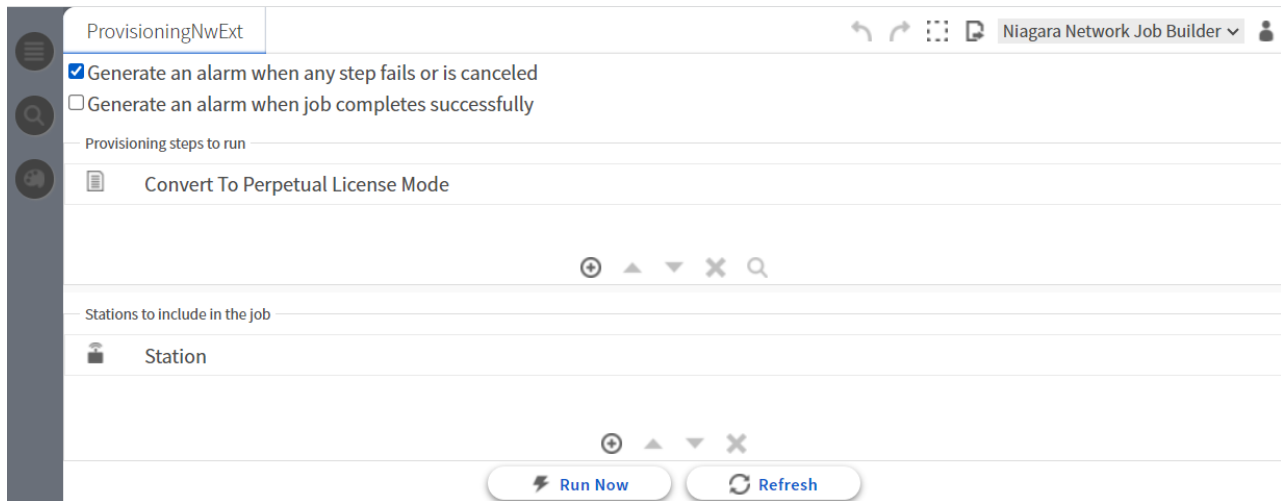
To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click  and double-click **Convert To Perpetual License Mode** or select **Convert To Perpetual License Mode** and click **OK**. This adds the job step **Convert To Perpetual License Mode** and **Update Licenses** to the in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Click  to add the appropriate station in the **Stations to include in this job**. Running this job converts all the stations' licensing mode from subscription to perpetual.

NOTE: Restart the station to get the changes effective.

HTML 5 Convert To Perpetual License Mode

In EC-Net 4.14 and later, there is added browser support that converts all the stations' licensing mode from subscription to perpetual. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 21 Convert To Perpetual License Mode

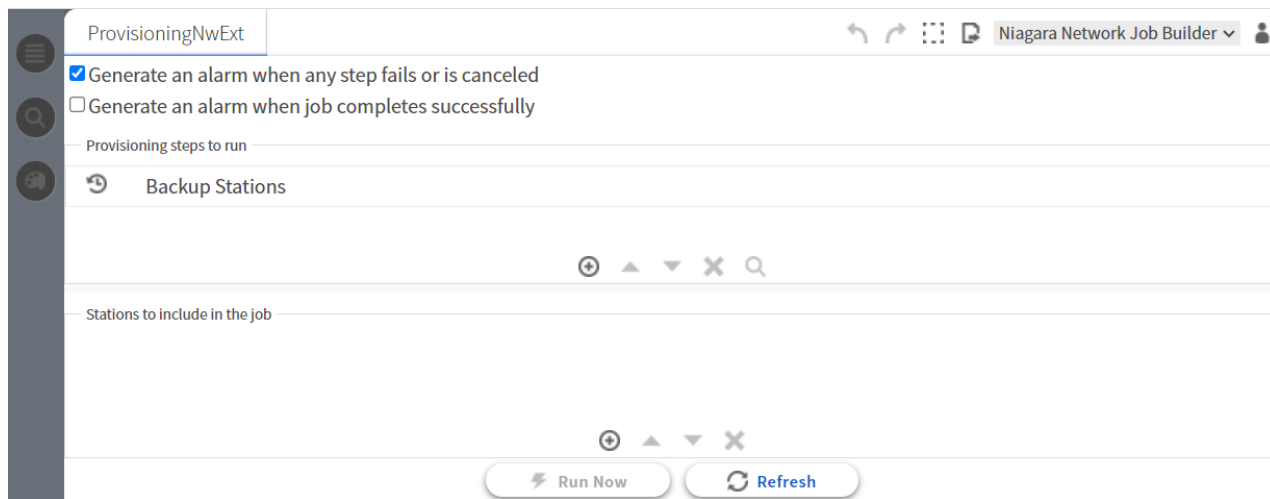


To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click ⊕ and double-click **Convert To Perpetual License Mode** or select **Convert To Perpetual License Mode** and click **OK**. This adds the job step **Convert To Perpetual License Mode** to the in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Click ⊕ to add the appropriate station in the **Stations to include in this job**. Running this job converts all the stations' licensing mode from subscription to perpetual.

NOTE: Restart the station to get the changes effective.

HTML5 Backup Stations

In EC-Net 4.14 and later, there is added browser support that enables the **Backup Station** steps used to make an online backup for each specified station (provided the station is running), or if it is not running, an offline backup. In either case, the backup .dist file for each station is stored on the Supervisor. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 22 Backup Stations

To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click ⊕ and double-click **Backup Stations** or select **Backup Stations** and click **OK**.

You add this step to the top pane (**To run for each station**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

Configure Niagara IdP and SAML Scheme

This component sets up a provisioning job on the Supervisor station to configure one or more remote stations with a server certificate (private and public keys) and the SAML (Security Assertion Markup Language) authentication scheme configured for the internal IdP (Identity Provider). First this provisioning job configures SAML authentication schemes on the remote station based on the Circle of Trust for which the station is configured in the **SAMLIdpService**. Then, it updates the **Station Service Provider** objects in the Circle of Trust with the correct values from the remote station.

You add this job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

Figure 23 Niagara IdP and SAML Scheme Properties

Configure Niagara IdP and SAML Scheme

Configure Niagara IdP and SAML Scheme
Enter the desired aliases for the remote IdP and SAML Server certificates, and optionally generate new SAML Server certificates

Login Button Text Format (required)

IdP Certificate Alias ☒ Use SAMLIdPService Certificate Alias

SAML Signing Certificate Alias (required)

SAML Encryption Certificate Alias (required)

☒ Generate new remote SAML signing certificate
Common Name (CN) (required)
* this may contain the host name or address of the server

☒ Generate new remote SAML encryption certificate
Common Name (CN) (required)
* this may contain the host name or address of the server

Organizational Unit (OU)

Organization (O) (required)

Locality (L)

State/Province (ST)

Country Code (C) (required)

Not Before

Not After

Key Size ☒ 1024 bits ☒ 2048 bits ☐ 3072 bits ☐ 4096 bits

Certificate Usage ☒ Server ☐ Client ☐ CA ☐ Code Signing

Alternate Server Name

Alternate Server URI

Email Address

☒ Digital signature ☐ Non-repudiation ☒ Key encipherment

Key Usage ☐ Data encipherment ☐ Key agreement ☐ Certificate signing

☐ CRL signing ☐ Encipher only ☐ Decipher only

OK Cancel

Property	Value	Description
Login Button Text Format	text	Displays the text on the login button to access the subordinate station.
IdP Certificate Alias	text	Specifies the alias for the certificate (with public key).
SAML Signing Certificate Alias	text	Specifies the alias of the certificate to use as the subordinate station's SAML Server Certificate.
SAML Encryption Certificate Alias	text	Specifies the alias of the encryption certificate.
Generate new remote SAML signing certificate	check box	Generates a new server certificate to use as the subordinate station's SAML Server Certificate.
Common Name (CN)	text	Specifies the distinguished name of the host or address of the server.
Generate new remote SAML encryption certificate	text	Generates a new server encryption certificate to use as the subordinate station's SAML Server Certificate.
Common Name (CN)	text	Specifies the distinguished name of the host or address of the server.

Property	Value	Description
Organizational Unit (OU)	text	The name of a department within the organization or a Doing-Business-As (DBA entry).
Organization (O)	text	The legally registered name of your company or organization. Do not abbreviate this name.
Locality (L)	text	The city in which the organization for which you are creating the certificate is located.
State/Province (ST)	text	The complete name of the state or province in which your organization is located. This property is optional.
Country Code (C)	two-character ISO-format country code	If you do not know your country's two-character code, check www.countrycode.org .
Not Before	date	Specifies the date before which the certificate is not valid.
Not After	date	Specifies the expiration date for the certificate.
Key Size	number	Specifies the size of the keys in bits. Four key sizes are allowed: 1024 bits, 2048 bits (this is the default), 3072 bits, and 4096 bits. Larger keys take longer to generate but offer greater security.
Certificate Usage:	text	Specifies the purpose of the certificate.
Alternative Server Name	text	This property provides a name other than the Subject (Common Name) that the system can use to connect to the server.
Email Address	email address	The contact address for this certificate. It may also be the address to which your signed certificate (.pem file) will be sent.
Key Usage	check box	Indicates the business scenario that requires authentication, encryption, and digital signing. The public and private keys associated with each certificate may be used to provide these secure features.

Copy Local File

This component supports the copy file steps (**Copy Supervisor File** step and **Copy Local File** step).

When creating a one-time job using the **Niagara Network Job Builder**, you can select either a Supervisor file or a local file on your PC to copy. A Copy Local File step is only available in the **Niagara Network Job Builder**. It is not available in the **Niagara Network Prototype View**.

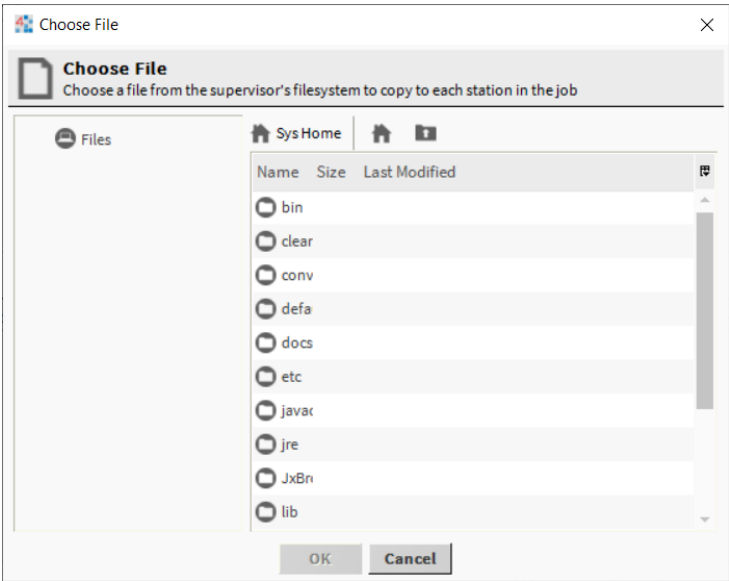
If you are creating a job prototype using the **Niagara Network Schedule View**, you can select only a Supervisor file.

Copy Supervisor File

This component copies a single file from your Supervisor station's file system to a location in each remote station in the job. The system opens the **File Chooser** for the files available as part of the Supervisor station.

You add this job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

Figure 24 Copy Supervisor File

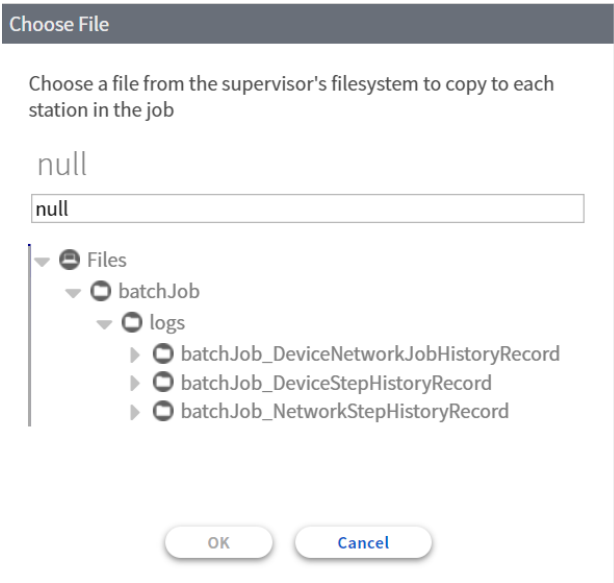


Property	Value	Description
File Chooser	File path	Prompts you to navigate to a source file from the file system of the Supervisor to copy to each station within the job.

HTML5 Copy Supervisor File

In EC-Net 4.14 and later, there is added browser support that allows to copy a single file from your Supervisor station's file system to a location in each remote station in the job. The system opens the **File Chooser** for the files available as part of the Supervisor station. The "HTML5" version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 25 Copy Supervisor File



To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click  and double-click **Copy Supervisor File** or select **Copy Supervisor File** and click **OK**. The **Choose File** window opens.

You add this job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

Property	Value	Description
File Chooser	File path	Prompts you to navigate to a source file from the file system of the Supervisor to copy to each station within the job.

Deploy Template

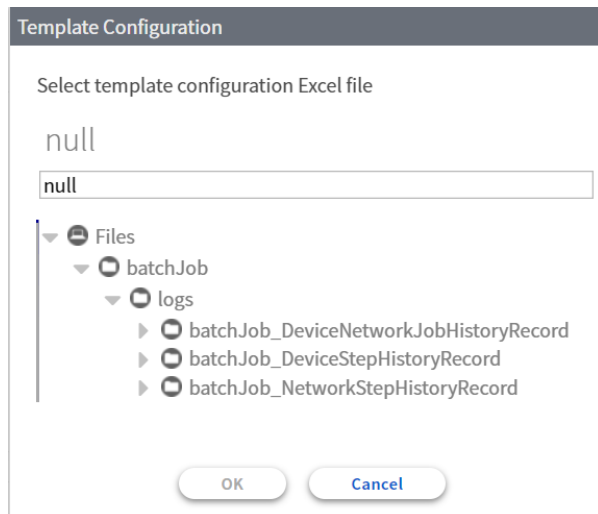
This component supports the **Deploy Template** step that can be added in the top pane of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

Running this step deploys a template file (*.ntpl) to run for each remote station from an Excel workbook containing template configurations.

HTML5 Deploy Template

In EC-Net 4.14 and later, there is added browser support for the **Deploy Template** step that can be added in the top pane of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 26 Deploy Template



To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click  and double-click **Deploy Template** or select **Deploy Template** and click **OK**. The **Template Configuration** window opens, select the required file and click **OK**. You add this job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

Running this step deploys a template file (*.ntpl) to run for each remote station from an Excel workbook containing template configurations.

Enable Bootstrap Mode

This component triggers a Supervisor-to-device connection using a certificate exemption for self-signed certificate.

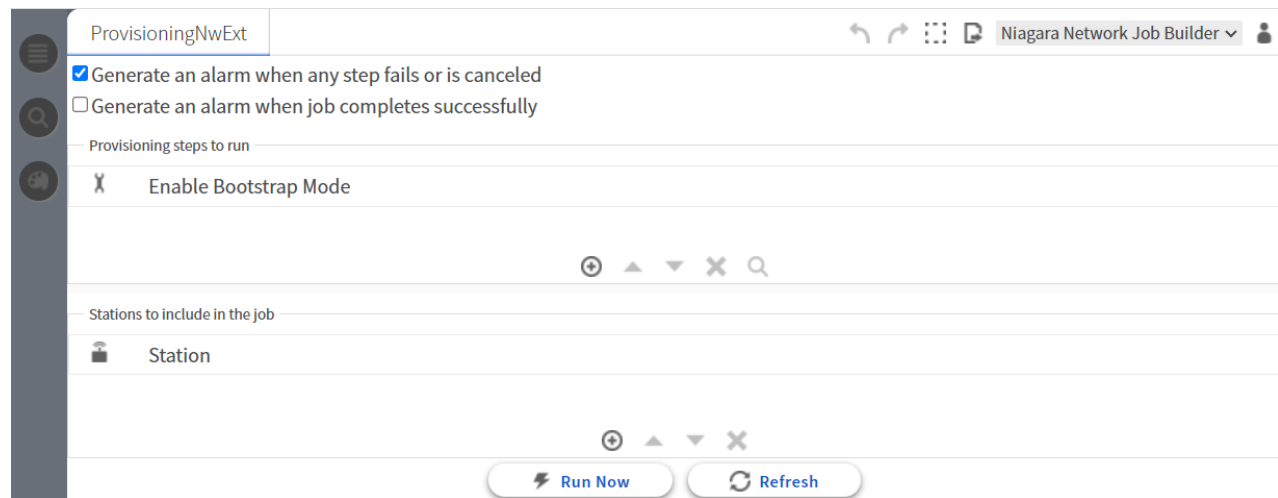
To access these job step components, double-click **ProvisioningNwExt** or **NiagaraNetworkJobPrototype**. The **Niagara Network Job Builder** or the **Niagara Network Prototype View** view opens. In the top pane **Provisioning steps to run** click the (+) symbol and click **Enable Bootstrap Mode**.

NOTE: Running this step by itself makes no persistent change in the target stations.

HTML5 Enable Bootstrap Mode

In EC-Net 4.14 and later, there is added browser support to trigger the Supervisor-to-device connection using a certificate exemption for self-signed certificate. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 27 Enable Bootstrap Mode



To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **ProvisioningNwExt** or **NiagaraNetworkJobPrototype**. The **Niagara Network Job Builder** view opens. In the top pane **Provisioning steps to run** click the (+) symbol and double-click **Enable Bootstrap Mode** or select **Enable Bootstrap Mode** and click **OK**.

NOTE: Running this step by itself makes no persistent change in the target stations.

Export Application Template Configuration

This step exports an application template configuration as an Excel (*.xlsx) file on each station in the job. 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**.

A prerequisite for adding this step to a provisioning job is that there is an existing application template to be exported.

See the “Template bulk deployment” section of the *EC-Net Templates Guide* for details on creating, configuring, and deploying templates.

NOTE: Application templates are not made available from the **Template** sidebar. Instead, the files (*.napl) can be found in the Windows User Home `applicationTemplates` folder.

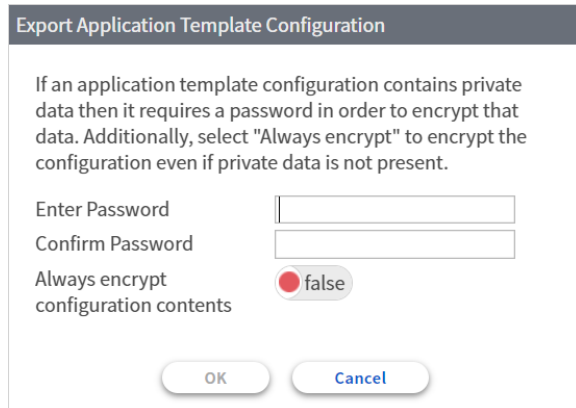
This opens a **File Chooser** window where you navigate to the Supervisor’s `^applicationTemplates` folder and select the application template file (*.napl) to use for this step and click the **Export Configs** option. The step will be added to the job sequence and will show the file path ORD to use.

For more details, see “Exporting the template spreadsheet” in this guide.

HTML5 Export Application Template Configuration

In EC-Net 4.14 and later, there is added browser support to export an application template configuration as an Excel (*.xlsx) file on each station in the job. 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**. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 28 Export Application Template Configuration



The dialog box titled "Export Application Template Configuration" contains the following text: "If an application template configuration contains private data then it requires a password in order to encrypt that data. Additionally, select 'Always encrypt' to encrypt the configuration even if private data is not present." Below this text are two password input fields labeled "Enter Password" and "Confirm Password". There is also a radio button labeled "Always encrypt configuration contents" which is currently set to "false". At the bottom are "OK" and "Cancel" buttons.

To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **ProvisioningNwExt**. To open the job steps window, click  and double-click **Export Application Template Configuration** or select **Export Application Template Configuration** and click **OK**.

The **Export Application Template Configuration** window opens. Enter the station password and click **OK**. This adds the job step in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

A prerequisite for adding this step to a provisioning job is that there is an existing application template to be exported.

See the “Template bulk deployment” section of the *EC-Net Templates Guide* for details on creating, configuring, and deploying templates.

NOTE: Application templates are not made available from the **Template** sidebar. Instead, the files (*.napl) can be found in the Windows User Home `applicationTemplates` folder.

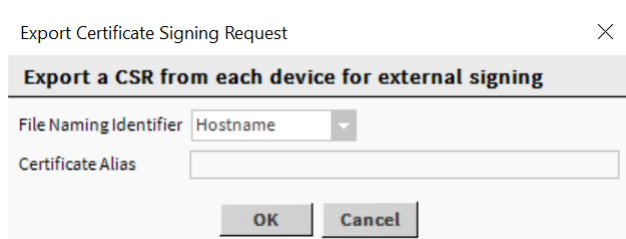
This opens a **File Chooser** window where you navigate to the Supervisor’s `^applicationTemplates` folder and select the application template file (*.napl) to use for this step and click the **Export Configs** option. The step will be added to the job sequence and will show the file path ORD to use.

For more details, see “Exporting the template spreadsheet” in this guide.

Export Certificate Signing Request

This component supports the **Export Certificate Signing Request** step. Running this step exports a Certificate Signing Request (CSR) from each remote device for external signing. Certificates installed for use on a controller are typically Server certificates.

Figure 29 Export Certificate Signing Request Job Step



The dialog box titled "Export Certificate Signing Request" has a close button (X) in the top right corner. Below the title bar is a header bar that says "Export a CSR from each device for external signing". Below this are two input fields: "File Naming Identifier" with a dropdown menu showing "Hostname" and "Certificate Alias" with an empty text box. At the bottom are "OK" and "Cancel" buttons.

To access these job step components, double-click **ProvisioningNwExt** or **NiagaraNetworkJobPrototype**. The **Niagara Network Job Builder** or the **Niagara Network Prototype View** view opens. In the top pane **Provisioning steps to run** click the (+) symbol and click **Export Certificate Signing Request**.

Property	Value	Description
File Naming Identifier	drop-down	Defines the file Name. <ul style="list-style-type: none"> <code>Hostname</code> represents the name of the server. <code>Host id</code> represent the alphanumeric code unique to the specific host. <code>Station Name</code> represents the remote station name.
Certificate Alias	text	Defines the certificate name.

HTML5 Export Certificate Signing Request

In EC-Net 4.14 and later, there is added browser supports the **Export Certificate Signing Request** step. Running this step exports a Certificate Signing Request (CSR) from each remote device for external signing. Certificates installed for use on a controller are typically Server certificates. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 30 Export Certificate Signing Request

To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **ProvisioningNwExt**. To open the job steps window, click ⊕ and double-click **Export Certificate Signing Request** or select **Export Certificate Signing Request** and click **OK**.

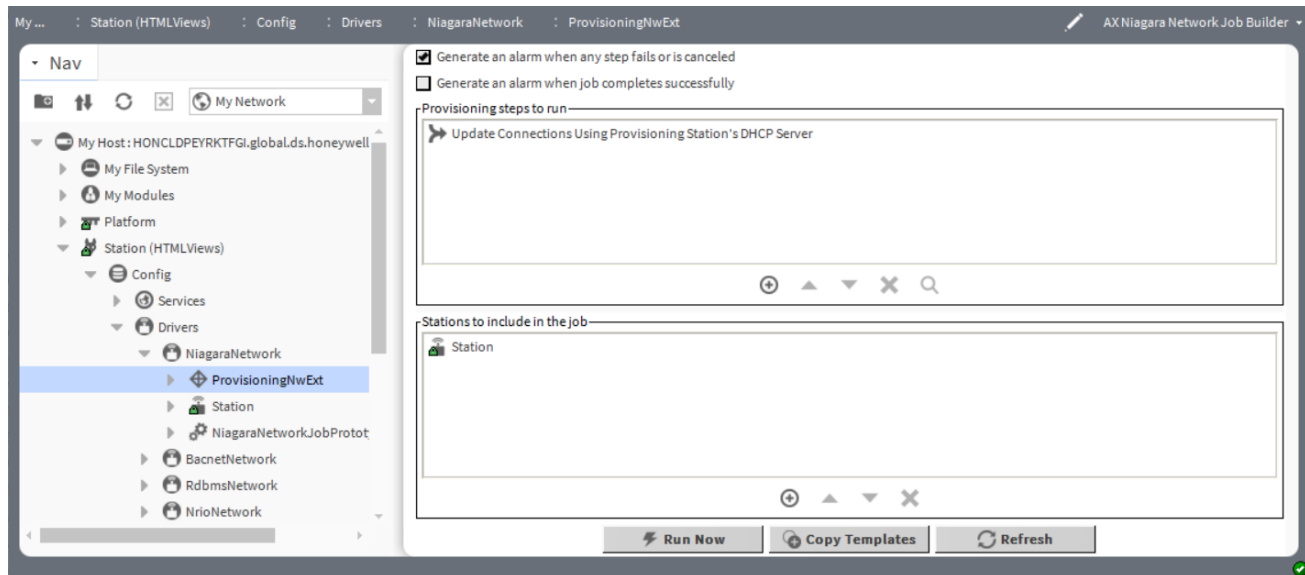
After configuring the properties the job step gets added to the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**

Property	Value	Description
File Naming Identifier	drop-down	Defines the file Name. <ul style="list-style-type: none"> <code>Hostname</code> represents the name of the server. <code>Host id</code> represent the alphanumeric code unique to the specific host. <code>Station Name</code> represents the remote station name.
Certificate Alias	text	Defines the certificate name.
Password	text	Defines the password for retrieving the private key. Click on Use global certificate password to use the global password.

Update Connections Using Provisioning Station's DHCP Server

This component updates the Supervisor-to-station connections when a EC-BOS-9 is acting as a Supervisor for a group of Edge devices connected through the secondary network port. It uses information pulled from the DHCP (Dynamic Host Configuration Protocol) server running on the EC-BOS-9. This information includes IP address and port information for Fox and Platform connections.

Figure 31 Update Connections Using Provisioning Station's DHCP Server

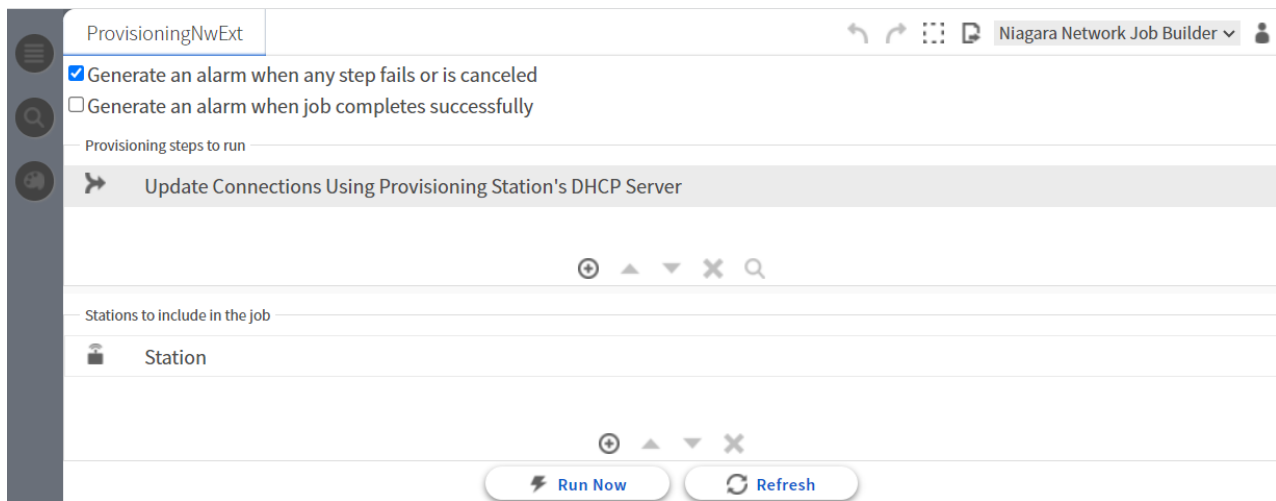


To access these job step component, expand **Config**→**Drivers**→**NiagaraNetwork** and double click the **ProvisioningNwExt** or **NiagaraNetworkJobPrototype**. The **Niagara Network Job Builder** or the **Niagara Network Prototype View** view opens. In the top pane **Provisioning steps to run** click the (+) icon and click **Update Connections Using Provisioning Station's DHCP Server**.

NOTE: Only Edge devices respond to this request in a **NiagaraNetwork**. EC-BOS controllers do not respond to these requests.

HTML5 Update Connections Using Provisioning Station's DHCP Server

In EC-Net 4.14 and later, there is added browser support that updates the Supervisor-to-station connections when a EC-BOS-9 is acting as a Supervisor for a group of Edge devices connected through the secondary network port. It uses information pulled from the DHCP (Dynamic Host Configuration Protocol) server running on the EC-BOS-9. This information includes IP address and port information for Fox and Platform connections.. The "HTML5" version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 32 Update Connections Using Provisioning Station's DHCP Server

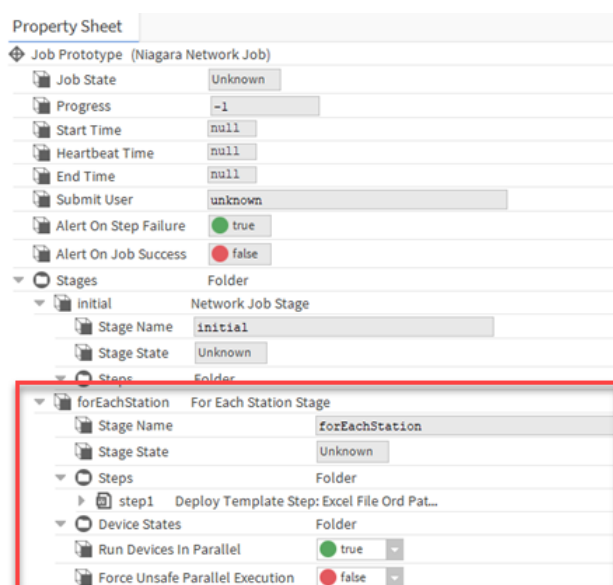
To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click **+** and double-click **Update Connections Using Provisioning Station's DHCP Server** or select **Update Connections Using Provisioning Station's DHCP Server** and click **OK**.

You add this job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

NOTE: Only Edge devices respond to this request in a **NiagaraNetwork**. EC-BOS controllers do not respond to these requests.

For Each Station Stage

This component contains the sequence of provisioning steps that can be added in the middle pane (**To run for each station**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

Figure 33 For Each Station Stage properties

This component has properties that control the execution of provisioning jobs executed across a set of remote devices. You can set the following properties through the **Property Sheet** view on the **For Each Station Stage** component.

Property	Value	Description
Stage Name	read-only	Specifies whether a provisioning job step is added to the “initial” job stage or to “for each station” stage.
Stage State	read-only	Specifies the state of the job stage.
Steps	folder	This folder contains the various added job steps.
Device States	folder	This folder contains the various added device stages.
Run Devices In Parallel	true (default) or false	To run provisioning jobs for each remote device serially (one at a time), change this value to <code>false</code> . When set to <code>false</code> forces all job steps to complete for one remote device before proceeding to run the same job steps on the next remote device. This can take considerably longer to run, but uses less system resources on the Supervisor.
Force Unsafe Parallel Execution	true or false (default)	<p>To force parallel execution of job step that are not always considered “safe”, change this value to <code>true</code>. When set to <code>true</code>, this allows parallel execution of job steps considered “unsafe”. Examples of these steps include those that might cause the system to reboot, like installing or updating software, or renaming the station.</p> <p>CAUTION: This is particularly important for IP devices configured in a daisy chained network topology, where rebooting the system causes the network segment to be disabled for some time. This will likely cause communication disruptions between the Supervisor and remote devices, which would make some provisioning jobs fail to complete.</p>

Generate Certificate

This component supports the **Generate Certificate** step. This step generates and installs a self-signed certificate on the remote device platform.

Figure 34 Generate Certificate

Generate Self Signed Certificate

Generate Certificate
Generate a self signed certificate in the user keystore

Alias (required)

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

Organizational Unit (OU)

Organization (O) (required)

Locality (L)

State/Province (ST)

Country Code (C) (required)

Not Before

Not After

Key Size ☒ 1024 bits ☒ 2048 bits ☐ 3072 bits ☐ 4096 bits

Certificate Usage ☒ Server ☐ Client ☐ CA ☐ Code Signing

Alternate Server Name

Alternate Server URI

Email Address

☒ Digital signature ☐ Non-repudiation ☒ Key encipherment

Key Usage ☐ Data encipherment ☐ Key agreement ☐ Certificate signing
☐ CRL signing ☐ Encipher only ☐ Decipher only

OK Cancel

To access these job step components, double-click **ProvisioningNwExt** or **NiagaraNetworkJobPrototype**. The **Niagara Network Job Builder** or the **Niagara Network Prototype View** view opens. In the top pane **Provisioning steps to run** click the (+) symbol and click **Generate Certificate**.

Property	Value	Description
Alias	text	Provides a short name used to distinguish certificates.
Common Name (CN)	text	Defines the host name or the address of the server.
Organizational Unit (OU)	text	The name of a department within the organization or a Doing-Business-As (DBA entry).
Organization (O)	text	The legally registered name of your company or organization. Do not abbreviate this name.
Locality (L)	text	The city in which the organization for which you are creating the certificate is located.
State/Province (ST)	text	The complete name of the state or province in which your organization is located. This property is optional.
Country Code (C)	two-character ISO-format country code	If you do not know your country's two-character code, check www.countrycode.org .
Not Before	date	Specifies the date before which the certificate is not valid.
Not After	date	Specifies the expiration date for the certificate.
Key Size	number	Specifies the size of the keys in bits. Four key sizes are allowed: 1024 bits, 2048 bits (this is the default), 3072 bits, and 4096 bits. Larger keys take longer to generate but offer greater security.

Property	Value	Description
Certificate Usage:	text	Specifies the purpose of the certificate.
Alternate Server Name	text	This property provides a name other than the Subject (Common Name) that the system can use to connect to the server.
Alternate Server URI	text	Provides an alternate server URI that the system can use to connect the server.
Email Address	email address	The contact address for this certificate. It may also be the address to which your signed certificate (.pem file) will be sent.
Email Address	check box	Indicates the business scenario that requires authentication, encryption, and digital signing. The public and private keys associated with each certificate may be used to provide these secure features.

Import Signed Certificate

This component supports the **Import Signed Certificate** step that can be added in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

Running this step imports a signed certificate into the **User Key Store** on the remote device platform in the job.

Figure 35 Import Signed Certificate

To access these job step components, double-click **ProvisioningNwExt** or **NiagaraNetworkJobPrototype**. The **Niagara Network Job Builder** or the **Niagara Network Prototype View** view opens. In the top pane **Provisioning steps to run** click the (+) symbol and click **Import Signed Certificate**.

Property	Value	Description
File Naming Identifier	text	Defines the file name. <ul style="list-style-type: none"> <code>Hostname</code> represents the name of the server. <code>Host Id</code> represent the Alphanumeric code unique to the specific host. <code>Station Name</code> represents the remote station name.
Use Filename as Alias	checkbox	If selected, the chosen File Naming Identifier (<code>Hostname</code> , <code>Host Id</code> , <code>Station Name</code>) will be used as the certificate alias.

Property	Value	Description
Certificate Alias	text	Defines the certificate name.
Supervisor Password	text	Defines the Password required to access the Supervisor station.

HTML5 Import Signed Certificate

In EC-Net 4.14 and later, there is added browser support to **Import Signed Certificate** step that can be added in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 36 Import Signed Certificate

To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click **+** and double-click **Import Signed Certificate** or select **Import Signed Certificate** and click **OK**.

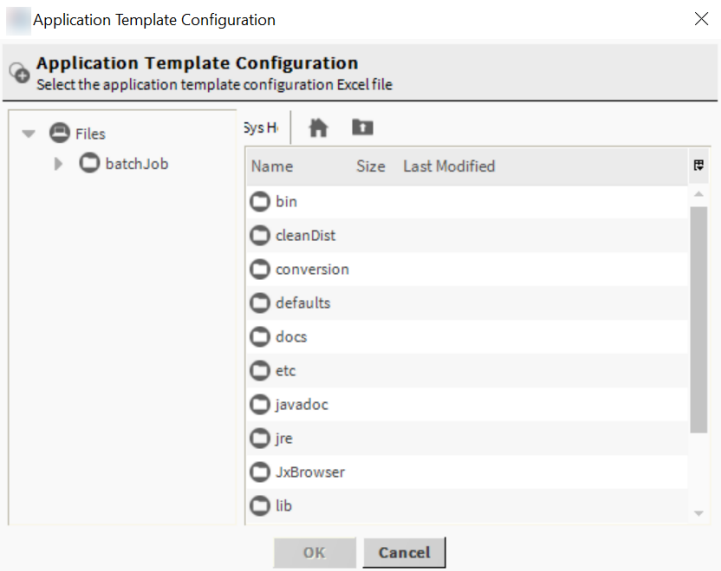
The **Import Signed Certificate** window opens. Enter the properties mentioned below and click **OK**. This adds the job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Running this step imports a signed certificate into the **User Key Store** on the remote device platform in the job.

Property	Value	Description
File Naming Identifier	text	Defines the file name. <ul style="list-style-type: none"> <code>Hostname</code> represents the name of the server. <code>Host Id</code> represent the Alphanumeric code unique to the specific host. <code>Station Name</code> represents the remote station name.
Use Filename as Alias	checkbox	If selected, the chosen File Naming Identifier (<code>Hostname</code> , <code>Host Id</code> , <code>Station Name</code>) will be used as the certificate alias.
Certificate Alias	text	Defines the certificate name.
Supervisor Password	text	Defines the Password required to access the Supervisor station.

Install Application Template

This component supports the **Install Application Template** step. Running this step installs an Application Template file (*.napl) on each station in the job.

Figure 37 Install Application Template



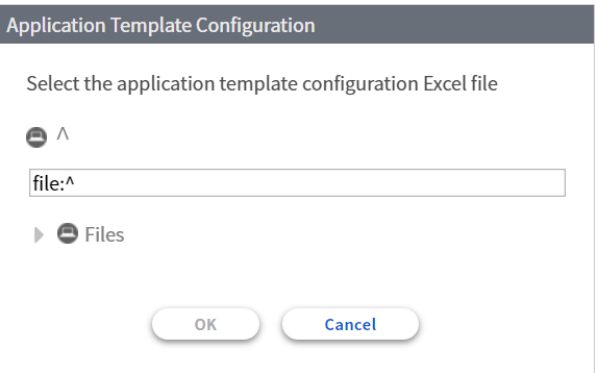
To access these job step component, double click the **ProvisioningNwExt** or **NiagaraNetworkJobPrototype**. The **Niagara Network Job Builder** or the **Niagara Network Prototype View** view opens. In the top pane **Provisioning steps to run** click the (+) symbol and click **Install Application Template**.

Property	Value	Description
Select Application Template	File path	Prompts you to navigate the application template in the Supervisor station using an Excel file

HTML5 Install Application Template

In EC-Net 4.14 and later, there is added browser supports the **Install Application Template** step. Running this step installs an Application Template file (*.nap1) on each station in the job. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 38 Install Application Template



To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **ProvisioningNwExt**. To open the job steps window, click ⊕ and double-click **Install Application Template** or select **Install Application Template** and click **OK**.

The **Application Template Configuration** window opens. Select the application template in the Supervisor station using an Excel file. This adds the job step in the top pane (**Provisioning steps to run**) of the **Niagara**

Network Job Builder or the **Niagara Network Prototype View**. Running this step imports the application template on the remote device platform in the job.

Install Certificate

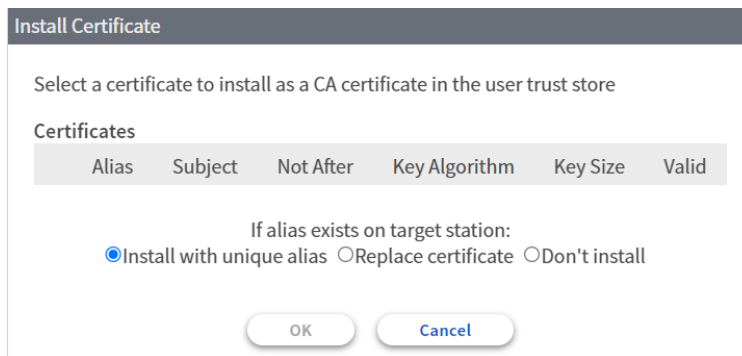
This component supports the **Install Certificate** step. Running this step installs a certificate to the **User Trust Store** of each remote station in the job.

You add this step in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

HTML5 Install Certificate

In EC-Net 4.14 and later, there is added browser supports the **Install Certificate** step. Running this step installs a certificate to the **User Trust Store** of each remote station in the job. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 39 Install Certificate



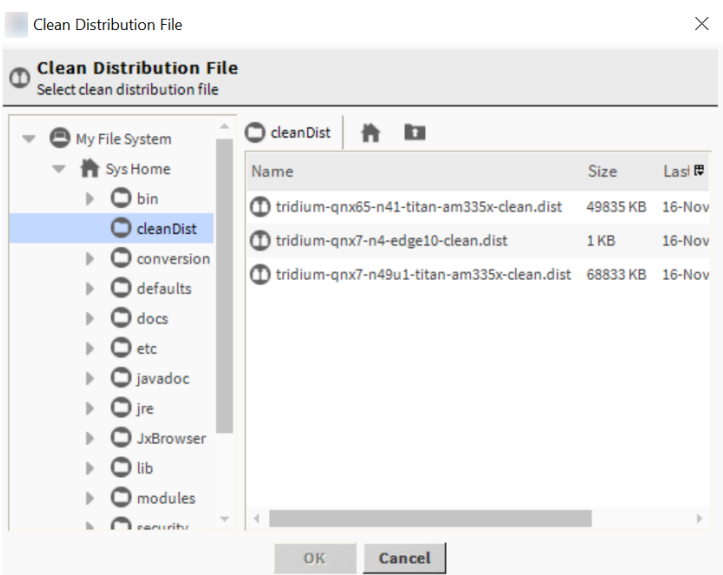
To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click **+** and double-click **Install Certificate** or select **Install Certificate** and click **OK**.

The **Install Certificate** window opens. Select the certificate and select one of the options **Install with unique alias** or **Replace certificate** or **Don't install** and click **OK**. This adds the job step in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Running this step installs the certificate on the remote device platform in the job.

Install Clean Distribution

This component supports the **Install Clean Distribution** step. It installs the selected clean distribution file to the target platforms. This effectively returns the device to the factory default state.

Figure 40 Install Clean Distribution



To access these job step components, double-click **ProvisioningNwExt** or **NiagaraNetworkJobPrototype**. The **Niagara Network Job Builder** or the **Niagara Network Prototype View** view opens. In the top pane **Provisioning steps to run** click the (+) symbol and click **Install Clean Distribution**.

NOTE: Clean distribution files are specific to a platform type, so only include target platforms that support the specific clean distribution in the list of target devices. The step fails for any mismatched platform.

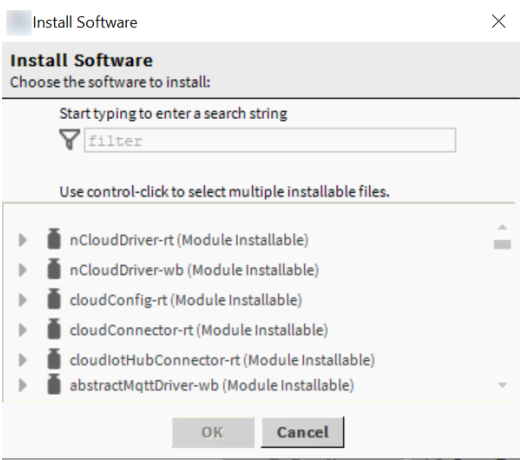
Property	Value	Description
Clean Distribution File	alpha-numeric file name	Installs the selected clean distribution file to the target platforms.

Install Software

This component supports the **Install Software** step, which allows you to install a versioned software module or dist to the target host(s) from the Supervisor’s software database (files under its `!sw` folder).

When adding a software item (module or .dist), you must specify a specific version.

Figure 41 Install Software



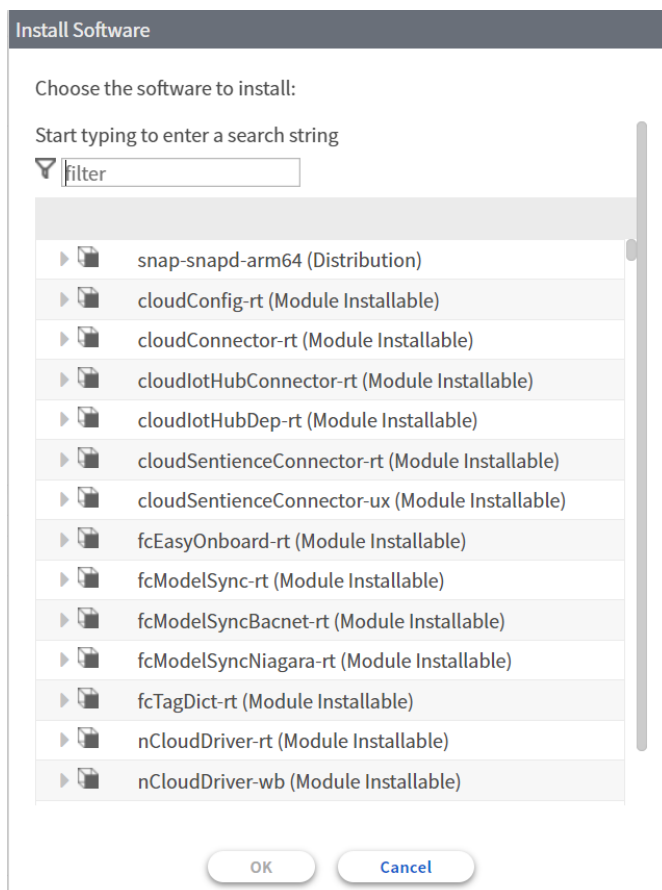
To access these job step components, double-click **ProvisioningNwExt** or **NiagaraNetworkJobPrototype**. The **Niagara Network Job Builder** or the **Niagara Network Prototype View** view opens. In the top pane **Provisioning steps to run** click the (+) symbol and click **Install Software**.

Property	Value	Description
Install Software	text	Installs a software module in one or more remote hosts.

HTML5 Install Software

In EC-Net 4.14 and later, there is added browser supports the **Install Software** step, which allows you to install a versioned software module or dist to the target host(s) from the Supervisor's software database (files under its !_{sw} folder). The "HTML5" version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 42 Install Software



To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **ProvisioningNwExt**. To open the job steps window, click ⊕ and double-click **Install Software** or select **Install Software** and click **OK**.

The **Install Software** window opens. Select the software and click **OK**. This adds the job step in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Running this step installs a software module in one or more remote hosts. When adding a software item (module or .dist), you must specify a specific version.

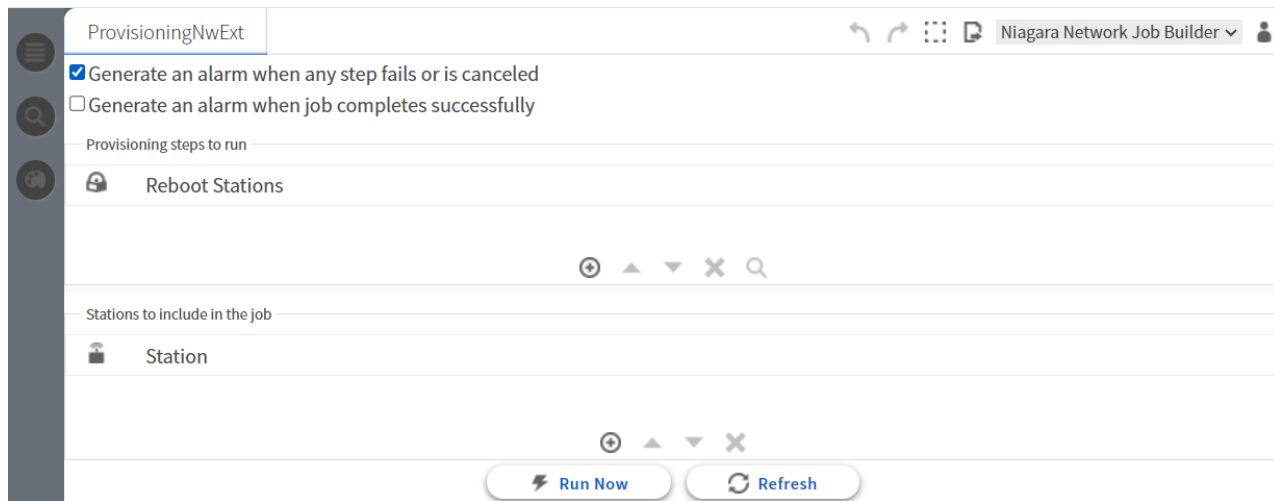
Reboot

This component supports the **Reboot** step. Running this step restarts the station's host (platform), then waits until its platform daemon comes back up and is available for connections to begin.

HTML5 Reboot

In EC-Net 4.14 and later, there is added browser support to the **Reboot** step. Running this step restarts the station's host (platform), then waits until its platform daemon comes back up and is available for connections to begin. The "HTML5" version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 43 Reboot



To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **ProvisioningNwExt**. To open the job steps window, click **+** and double-click **Reboot Stations** or select **Reboot Stations** and click **OK**.

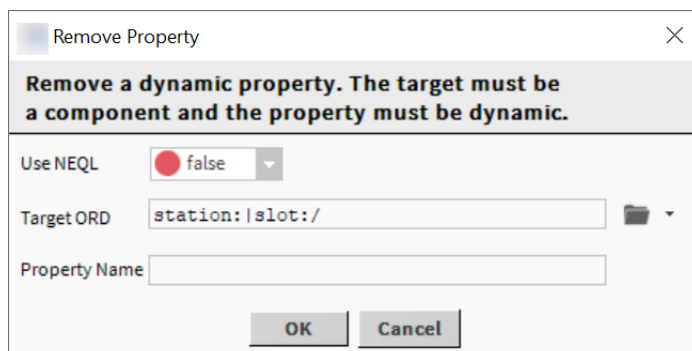
You add this job step in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Running this step restarts the station's host (platform), then waits until its platform daemon comes back up and is available for connections to begin.

Remove Property

Starting in EC-Net 4 v4.12, this component supports the Remove Property job step. Running this step removes a property from a component at a defined slot path for each station in the provisioning job.

You add this job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

Figure 44 Remove Property



Property	Value	Description
Use NEQL	true or false (default)	Turns on (<i>true</i>) and off (<i>false</i>) the use of a NEQL query to identify the property to remove.
Target ORD	ORD	Selects one of seven choosers to identify the property.
Property Name	text	Specifies the name of the selected property.

HTML5 Remove Property

In EC-Net 4.14 and later, there is added browser supports the **Remove Property** job step. Running this step removes a property from a component at a defined slot path for each station in the provisioning job. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 45 Remove Property

Remove a dynamic property. The target must be a component and the property must be dynamic.

Use NEQL ☒ false

Target ORD

Property Name

To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click and double-click **Remove Property** or select **Remove Property** and click **OK**.

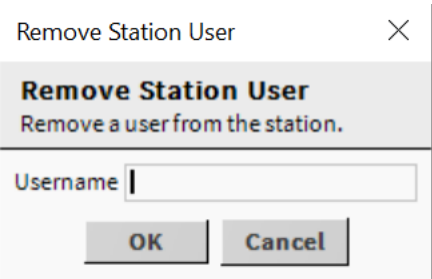
The **Remove Property** window opens. Enter the properties mentioned below and click **OK**. This adds the job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Running this step removes a property from a component at a defined slot path for each station in the provisioning job.

Property	Value	Description
Use NEQL	true or false (default)	Turns on (<i>true</i>) and off (<i>false</i>) the use of a NEQL query to identify the property to remove.
Target ORD	ORD	Selects one of seven choosers to identify the property.
Property Name	text	Specifies the name of the selected property.

Remove Station User

This component supports the Remove Station User step. Running this step removes a named user from the User Service for each station in the provisioning job.

Figure 46 Remove Station User



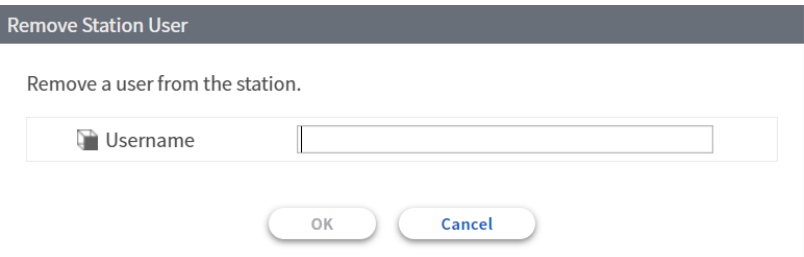
To access these job step components, double-click **ProvisioningNwExt** or **NiagaraNetworkJobPrototype**. The **Niagara Network Job Builder** or the **Niagara Network Prototype View** view opens. In the top pane **Provisioning steps to run** click the (+) symbol and click **Remove Station User**.

Property	Value	Description
Username	text	Identifies the <code>Username</code> of the user to be removed from the station.

HTML5 Remove Station User

In EC-Net 4.14 and later, there is added browser support to **Remove Station User** step. Running this step removes a named user from the **User Service** for each station in the provisioning job. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 47 Remove Station User



To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **ProvisioningNwExt**. To open the job steps window, click ⊕ and double-click **Remove Station User** or select **Remove Station User** and click **OK**.

The **Remove Station User** window opens. Enter the `Username` of the user to be removed from the station job step in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Running this step removes a named user from the **User Service** for each station in the provisioning job.

Rename Device Station

This component supports the **Rename Device Station** step. Running this step renames the station on the remote device to match the station proxy name in the Supervisor’s **NiagaraNetwork**. This is done for each station in the provisioning job.

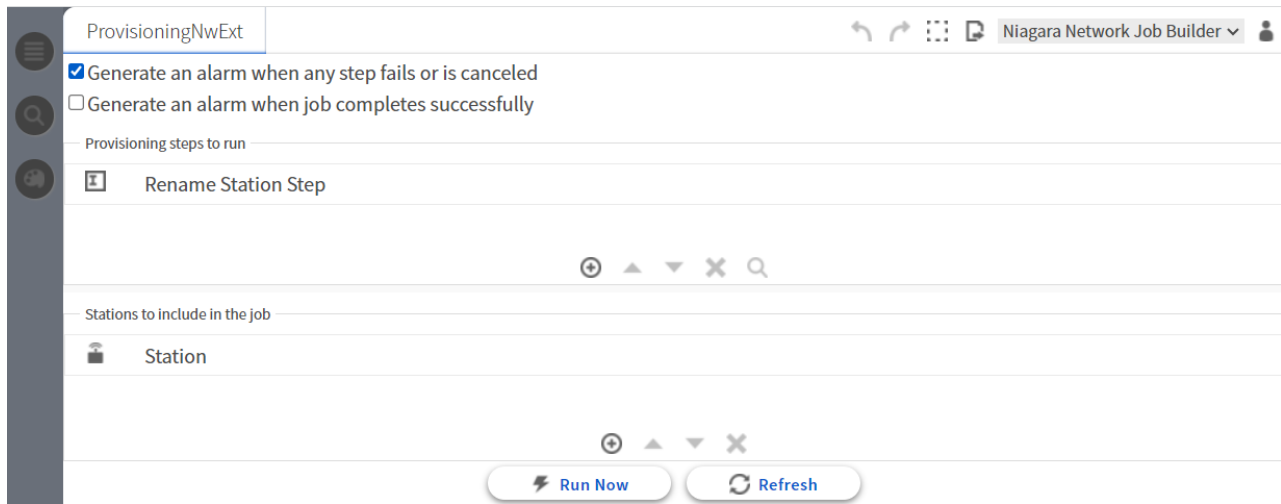
You add this component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**

HTML5 Rename Device Station

In EC-Net 4.14 and later, there is added browser supports the **Rename Device Station** step. Running this step renames the station on the remote device to match the station proxy name in the Supervisor’s

NiagaraNetwork. This is done for each station in the provisioning job.. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 48 Rename Device Station



To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click and double-click **Rename Device Station** or select **Rename Device Station** and click **OK**. Running this step renames the station on the remote device.

Run Robot

This component supports the **Run Robot** step. This step passes the specified **ProvisioningRobot** (custom-created program code) located in the Supervisor station to the **ProgramService** on each station in the job. The station's **ProgramService** runs the custom code in the remote station.

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**.

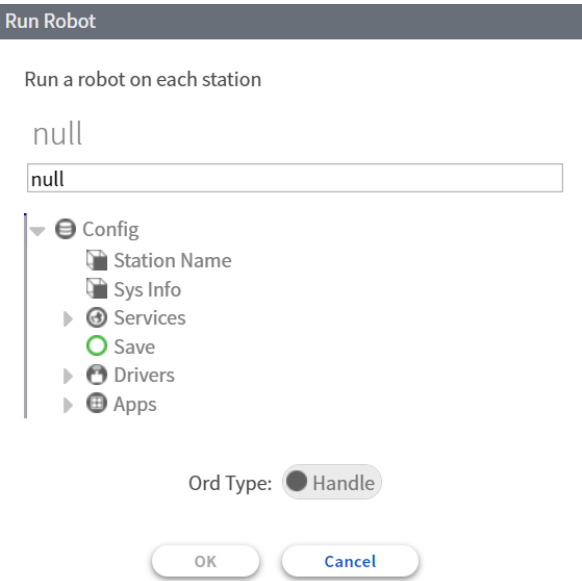
For any **Run Robot** step, you use the **Select a Robot** window to specify a provisioning robot on the Supervisor. The **ProgramService** of each station in the provisioning job runs the robot.

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

HTML5 Run Robot

In EC-Net 4.14 and later, there is added browser supports the **Run Robot** step. This step passes the specified **ProvisioningRobot** (custom-created program code) located in the Supervisor station to the **ProgramService** on each station in the job. The station's **ProgramService** runs the custom code in the remote station. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 49 Run Robot



To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click and double-click **Run Robot** or select **Run Robot** and click **OK**.

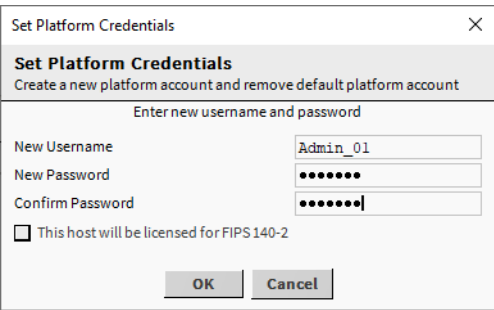
The **Run Robot** window opens. Specify a provisioning robot on the Supervisor. This adds the job step in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. The **ProgramService** of each station in the provisioning job runs the robot. Usage is expected to be infrequent, perhaps as a temporary measure for some misbehaving third-party software module. This job step fails if a station's **Software** device extension is disabled or in fault.

Security Job Steps

This component is a convenience step that generates three separate sets of credentials in one place and in one sequence. The individual steps are **Set Platform Credentials**, **Set Station User Password** and **Set Platform User Password**.

You add this component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or **Niagara Network Prototype View**.

Figure 50 Security Job Steps — Set Platform Credentials



Property	Value	Description
New Username	text	Defines the New Username to access an existing platform.
New Password	text	Defines the New Password to access an existing platform.

Property	Value	Description
Confirm Password	text	Confirms the changed password to access an existing platform.
This host will be licensed for FIPS 140–2	check box	Enables the Federal Information Processing Standard, version 140–2.

Figure 51 Security Job Steps — Set Station User Password

Property	Value	Description
Use connection username	check box	Enables the use of the same connection username.
Current Password	text	Defines the Current Password to access an existing station.
New Password	text	Defines the New Password to access an existing station.
Confirm Password	text	Confirms the changed password to access an existing station.

Figure 52 Security Job Steps — Set Platform User Password

Properties	Value	Description
Current Username	text	Defines the Current Username to access an existing platform.
Current Password	text	Defines the Current Password to access an existing platform.
New Password	text	Defines the New Password to access an existing platform.
Confirm Password	text	Confirms the changed Password to access an existing platform.
This host will be licensed for FIPS 140–2	check box	Enables the Federal Information Processing Standard, version 140–2.

HTML5 Security Job Steps

In EC-Net 4.14 and later, there is added browser support to generate three separate sets of credentials in one place and in one sequence. The individual steps are **Set Platform Credentials**, **Set Station User Password** and **Set Platform User Password**. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 53 Security Job Steps

Set Platform Credentials

Create a new platform account and remove default platform account

Enter new username and password

New Username

New Password

Confirm Password

This host will be licensed for FIPS 140-2

false

OK

Cancel

To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click  and double-click **Security Job Steps** or select **Security Job Steps** and click **OK**.

The **Set Platform Credentials** window opens. Enter the properties mentioned below and click **OK**. This adds the job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Running this step generates three separate sets of credentials in one place and in one sequence.

Property	Value	Description
New Username	text	Defines the New Username to access an existing platform.
New Password	text	Defines the New Password to access an existing platform.
Confirm Password	text	Confirms the changed password to access an existing platform.
This host will be licensed for FIPS 140-2	toggle button	Enables the Federal Information Processing Standard, version 140-2.

Set Certificate Alias

This component supports the **Set Certificate Alias** step. Running this step sets the certificate alias for the platform, Fox and Web services on each remote device in the job.

Figure 54 Set Certificate Alias

Set Certificate Alias

Set Certificate Alias

Set the certificate alias for platform, FoxService, and WebService

Certificate Alias

OK

Cancel

To access these job step components, double-click **ProvisioningNwExt** or **NiagaraNetworkJobPrototype**. The **Niagara Network Job Builder** or the **Niagara Network Prototype View** view opens. In the top pane **Provisioning steps to run** click the (+) symbol and click **Set Certificate Alias**.

Property	Value	Description
Certificate Alias	text	Sets the certificate alias for the platform, Fox and Web services on each remote device in the job.

HTML5 Set Certificate Alias

In EC-Net 4.14 and later, there is added browser support for the **Set Certificate Alias** step. Running this step sets the certificate alias for the platform, Fox and Web services on each remote device in the job. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 55 Set Certificate Alias

Set Certificate Alias

Set the certificate alias for platform, FoxService, and WebService

Certificate Alias

Please enter the password for retrieving the private key.

Password

☐ Use global certificate password

OK

Cancel

To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **ProvisioningNwExt**. To open the job steps window, click ⊕ and double-click **Set Certificate Alias** or select **Set Certificate Alias** and click **OK**.

This adds the job step in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Running this step sets the certificate alias for the platform, Fox and Web services on each remote device in the job.

Property	Value	Description
Certificate Alias	text	Sets the certificate alias for the platform, Fox and Web services on each remote device in the job.
Password	text	Enter the password or click Use global certificate password for retrieving the private key.

Set Platform Credentials

This component supports the **Set Platform Credentials** step. It creates a new platform account and sets the credentials for that account on each remote device in the job. It also removes the default platform account.

Figure 56 Set Platform Credentials

Set Platform Credentials

Set Platform Credentials

Create a new platform account and remove default platform account

Enter new username and password

New Username

New Password

Confirm Password

☐ This host will be licensed for FIPS 140-2

OK

Cancel

To access these job step components, double-click **ProvisioningNwExt** or **NiagaraNetworkJobPrototype**. The **Niagara Network Job Builder** or the **Niagara Network Prototype View** view opens. In the top pane **Provisioning steps to run** click the (+) symbol and click **Set Platform Credentials**.

Property	Value	Description
New Username	text	Defines the New Username for the new platform account on each remote device in the job.
New Password	text	Defines the New Password for the new platform account on each remote device in the job.
Confirm Password	text	Confirms the new password for the new platform account on each remote device in the job.
This host will be licensed for FIPS 140-2	check box	Enables the Federal Information Processing Standard, version 140-2.

HTML5 Set Platform Credentials

In EC-Net 4.14 and later, there is added browser support to the **Set Platform Credentials** step. It creates a new platform account and sets the credentials for that account on each remote device in the job. It also removes the default platform account. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 57 Set Platform Credentials

Set Platform Credentials

Create a new platform account and remove default platform account

Enter new username and password

New Username

New Password

Confirm Password

This host will be licensed for FIPS 140-2 ☐ false

OK

Cancel

To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **ProvisioningNwExt**. To open the job steps window, click ⊕ and double-click **Set Platform Credentials** or select **Set Platform Credentials** and click **OK**. The **Set Platform Credentials** window opens. Enter the properties mentioned below and click **OK**. This adds the job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job**

Builder or the **Niagara Network Prototype View**. Running this step creates a new platform account and sets the credentials for that account on each remote device in the job.

Property	Value	Description
New Username	text	Defines the New Username for the new platform account on each remote device in the job.
New Password	text	Defines the New Password for the new platform account on each remote device in the job.
Confirm Password	text	Confirms the new password for the new platform account on each remote device in the job.
This host will be licensed for FIPS 140–2	check box	Enables the Federal Information Processing Standard, version 140–2.

Set Platform User Password

This component supports the **Set Platform User Password** step. It changes the password for an existing platform user on each remote device in the job.

Figure 58 Set Platform User Password window

You add this component in the top pane (**To run for each station**) of the **Niagara Network Job Builder** or **Niagara Network Prototype View**.

Type	Value	Description
Current Username	text	Defines the Current Username to access an existing platform.
Current Password	text	Defines the Current Password to access an existing platform.
New Password	text	Defines the New Password to access an existing platform.
Confirm Password	text	Confirms the changed Password to access an existing platform.
This host will be licensed for FIPS 140–2	check box	Enables the Federal Information Processing Standard, version 140–2.

HTML5 Set Platform User Password

In EC-Net 4.14 and later, there is added browser supports the **Set Platform User Password** step. It changes the password for an existing platform user on each remote device in the job. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 59 Set Platform User Password

Set Platform User Password

Change password for an existing platform user

Enter username, current password and new password

Current Username

Current Password

New Password

Confirm Password

This host will be licensed for FIPS 140-2

false

OK

Cancel

To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click  and double-click **Set Platform User Password** or select **Set Platform User Password** and click **OK**. The **Set Platform User Password** window opens. Enter the properties mentioned below and click **OK**. This adds the job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Running this step changes the password for an existing platform user on each remote device in the job.

Type	Value	Description
Current Username	text	Defines the Current Username to access an existing platform.
Current Password	text	Defines the Current Password to access an existing platform.
New Password	text	Defines the New Password to access an existing platform.
Confirm Password	text	Confirms the changed Password to access an existing platform.
This host will be licensed for FIPS 140–2	toggle button	Enables the Federal Information Processing Standard, version 140–2.

Set Property

This component supports the **Set Property Job** step. Running this step sets or adds a property to a component at a defined slot path for each station in the provisioning job.

You add this component in the top pane (Provisioning steps to run) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

Figure 60 Set Property properties

Set Property

Set or add a property

Use NEQL

false

Target ORD

station:|slot:/

Property Name

Property Type

baja

String

Add If Missing

false

Unique Value per Station

false

Property Value:

OK

Cancel

Property	Value	Description
Use NEQL	true or false (default)	Turns on (true) and off (false) the use of a NEQL query to identify the property to update.
Target ORD	ORD	Selects one of seven choosers to identify the property.
Property Name	read-only	Reports the name of the selected property.
Property Type	two drop-down lists	Select any type registered with the Supervisor.
Add if Missing	true or false (default)	Adds the property at the slot path.
Unique Value per Station	true or false (default)	Indicates if the updated property in each station should have the same value false or a unique value (true).
Property Value	text	<div>Contains the value. The editor for this value changes based on the selected Property Type.</div> <div>You can set individual values if the Unique Value per Station is set to true. The property Value pane will contain an entry for each available station in the NiagaraNetwork.</div>

HTML5 Set Property

In EC-Net 4.14 and later, there is added browser supports the **Set Property Job** step. Running this step sets or adds a property to a component at a defined slot path for each station in the provisioning job. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 61 Set Property properties

Set Property

Set or add a property

Use NEQL

false

Target ORD

station:|slot:/

Property Name

Property Type

baja

String

Add If Missing

false

Unique Value per Station

false

Property Value:

OK

Cancel

To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click  and double-click **Backup Stations** or select **Backup Stations** and click **OK**. The **Set Property** window opens. Enter the properties mentioned below and click **OK**. This adds the job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Running this step creates a new platform account and remove default platform account.

Property	Value	Description
Use NEQL	true or false (default)	Turns on (true) and off (false) the use of a NEQL query to identify the property to update.
Target ORD	ORD	Selects one of seven choosers to identify the property.
Property Name	read-only	Reports the name of the selected property.
Property Type	two drop-down lists	Select any type registered with the Supervisor.
Add if Missing	true or false (default)	Adds the property at the slot path.
Unique Value per Station	true or false (default)	Indicates if the updated property in each station should have the same value false or a unique value (true).
Property Value	text	Contains the value. The editor for this value changes based on the selected Property Type . You can set individual values if the Unique Value per Station is set to true. The property Value pane will contain an entry for each available station in the NiagaraNetwork .

Set Station Connection Credentials

This component supports the **Set Station Connection Credentials** step. Running this step sets the **Username** and **Password** for the client connection in the station proxy in the Supervisor's **NiagaraNetwork**.

Figure 62 Set Station Connection Credentials

To access these job step components, double-click **ProvisioningNwExt** or **NiagaraNetworkJobPrototype**. The **Niagara Network Job Builder** or the **Niagara Network Prototype View** view opens. In the top pane **Provisioning steps to run** click the (+) symbol and click **Set Station Connection Credentials**.

NOTE: This step resets any existing connection to use the new credentials when the step is executed.

Properties	Value	Description
Username	text	Defines the Username to connect the remote station.
Password	text	Defines the Password for the Username .

HTML5 Set Station Connection Credentials

In EC-Net 4.14 and later, there is added browser supports the **Set Station Connection Credentials** step. Running this step sets the **Username** and **Password** for the client connection in the station proxy in the Supervisor's **NiagaraNetwork**. The "HTML5" version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 63 Set Station Connection Credentials

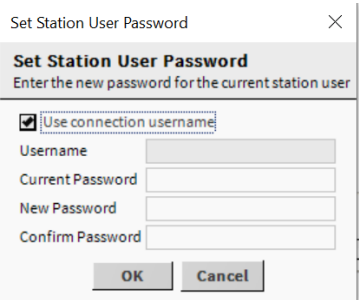
To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **ProvisioningNwExt**. To open the job steps window, click ⊕ and double-click **Set Station Connection Credentials** or select **Set Station Connection Credentials** and click **OK**. The **Set Station Connection Credentials** window opens. Enter the properties mentioned below and click **OK**. This adds the job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Running this step sets the **Username** and **Password** for the client connection.

Properties	Value	Description
Username	text	Defines the Username to connect the remote station.
Password	text	Defines the Password for the Username .

Set Station User Password

This component supports the **Set Station User Password** step. Running this step sets the password of a station user on each remote device in the job. It provides the option of changing the password for the established **NiagaraNetwork** Fox user, or changing the password of another user account on the station.

Figure 64 Set Station User Password



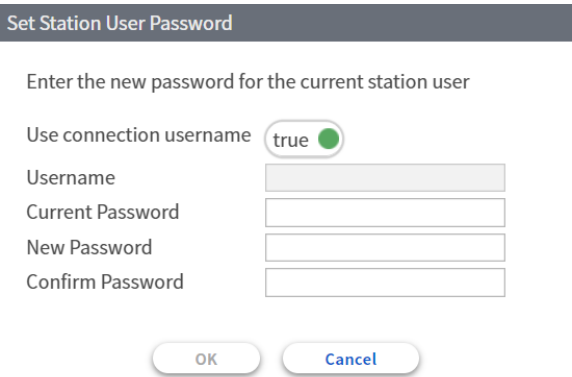
To access these job step component, double click the **ProvisioningNwExt** or **NiagaraNetworkJobPrototype**. The **Niagara Network Job Builder** or the **Niagara Network Prototype View** view opens. In the top pane **Provisioning steps to run** click the (+) symbol and click **Set Station User Password**.

Property	Value	Description
Use connection username	check box	Enables the use of the same connection username.
Current Password	text	Defines the Current Password to access an existing station.
New Password	text	Defines the New Password to access an existing station.
Confirm Password	text	Confirms the changed password to access an existing station.

HTML5 Set Station User Password

In EC-Net 4.14 and later, there is added browser support for the **Set Station User Password** step. Running this step sets the password of a station user on each remote device in the job. It provides the option of changing the password for the established **NiagaraNetwork** Fox user, or changing the password of another user account on the station.. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 65 Set Station User Password



To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **ProvisioningNwExt**. To open the job steps window, click ⊕ and double-click **Set Station User Password** or select **Set Station User Password** and click **OK**. The **Set Station User Password** window opens. Enter the properties mentioned below and click

OK. This adds the job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Running this step sets the password of a station user on each remote device in the job.

Property	Value	Description
Use connection username	toggle button	Enables the use of the same connection username.
Current Password	text	Defines the Current Password to access an existing station.
New Password	text	Defines the New Password to access an existing station.
Confirm Password	text	Confirms the changed password to access an existing station.

Set System Passphrase

This component supports the **Set System Passphrase** step. Running this step sets the system passphrase on each remote device in the job.

Figure 66 Set System Passphrase properties

You add this component in the top pane (**To run for each station**) of the **Niagara Network Job Builder** or **Niagara Network Prototype View**.

Property	Value	Description
Current Passphrase	text	Defines the current passphrase.
New Passphrase	text	Creates a new passphrase.
Confirm Passphrase	text	Confirms the new passphrase.
This host will be licensed for FIPS 140–2	check box	Enables the Federal Information Processing Standard, version 140–2.

HTML5 Set System Passphrase

In EC-Net 4.14 and later, there is added browser support for the **Set System Passphrase** step. Running this step sets the system passphrase on each remote device in the job. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 67 Set System Passphrase

Set System Passphrase

Enter new system passphrase

Current Passphrase

New Passphrase

Confirm Passphrase

This host will be licensed for FIPS 140-2

false

OK

Cancel

To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click **+** and double-click **Set System Passphrase** or select **Set System Passphrase** and click **OK**. The **Set System Passphrase** window opens. Enter the properties mentioned below and click **OK**. This adds the job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Running this step sets the system passphrase on each remote device in the job.

Property	Value	Description
Current Passphrase	text	Defines the current passphrase.
New Passphrase	text	Creates a new passphrase.
Confirm Passphrase	text	Confirms the new passphrase.
This host will be licensed for FIPS 140-2	toggle button	Enables the Federal Information Processing Standard, version 140-2.

Set TLS Level

This component supports the **Set TLS Level** step. Running this step sets the minimum TLS level for the platform, Fox, and Web services for each remote device in the job.

Figure 68 Set TLS Level properties

Set TLS Level

Set TLS Level

Set the minimum TLS level for platform, FoxService, and WebService

TLSLevel

TLSv1.3

OK

Cancel

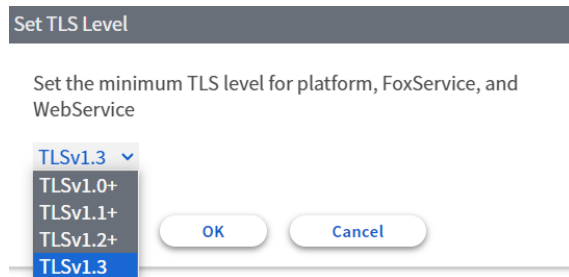
To access these job step components, double-click **ProvisioningNwExt** or **NiagaraNetworkJobPrototype**. The **Niagara Network Job Builder** or the **Niagara Network Prototype View** view opens. In the top pane **Provisioning steps to run** click the **(+)** symbol and click **Set TLS Level**.

Property	Value	Description
TLS Level	drop-down list	Selects the version of TLS required to support secure communication.

HTML5 Set TLS Level

In EC-Net 4.14 and later, there is added browser supports the **Set TLS Level** step. Running this step sets the minimum TLS level for the platform, Fox, and Web services for each remote device in the job. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 69 Set TLS Level



To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **ProvisioningNwExt**. To open the job steps window, click and double-click **Set TLS Level** or select **Set TLS Level** and click **OK**. The **Set TLS Level** window opens. Enter the properties mentioned below and click **OK**. This adds the job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Running this step sets the minimum TLS level for the platform.

Property	Value	Description
TLS Level	drop-down list	Selects the version of TLS required to support secure communication.

Set Time

This component supports the **Set Time** step. Running this step sets the time for each remote device in the job.

Figure 70 Set Time properties

You add this component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or **Niagara Network Prototype View**.

Property	Value	Description
Use supervisor time	bullet	Enables use of the time as set in the Supervisor PC to replicate in the remote stations.
Use selected time	bullet	Enables the Date/Time and Time Zone properties.
Date/Time	calendar	Sets the date and time.
Use NTP time	bullet	Enables the use of the time as supplied by the Network Time Protocol. This protocol synchronizes clocks among computer systems using a packet-switched, variable-latency data network.
Enabled	true (default) and false	Enables (true) and disables (false) use of NTP time.
NTP Host Mode	drop-down list	Selects the NTP host. Options are <code>Server</code> , <code>Server (Time Only)</code> and <code>Client</code> .
Sync Local Clock to NTP	true (default) and false	Enables (true) and disables (false) local synchronization immediately.
Sync Time At Boot	true and false (default)	Enables (true) and disables (false) synchronization when the platform restarts.
Use Local Clock as Backup	true and false (default)	
Generate NTP Statistics	true and false (default)	Enables (true) and disables (false) the tracking of information about NTP accuracy.
table	read-only	Provides NTP statistics.

HTML5 Set Time

In EC-Net 4.14 and later, there is added browser support to the **Set Time** step. Running this step sets the time for each remote device in the job. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 71 Set Time properties

Set Time

Set the time for each device

☒ Use supervisor time

☐ Use selected time

11/28/2023 12:13:28 PM
Asia/Calcutta (+5:30)

☐ Use NTP time

Enabled ☒ true

NTP Host Mode Server

Sync Local Clock to NTP ☒ true

Sync Time At Boot ☐ false

Use Local Clock as Backup ☐ false

Generate NTP Statistics ☐ false

Address	Peer Mode	Burst	Preferred	Min. Poll Interval	Max. Poll Interval
<div> + × ✎ ▲ ▼ </div>					

OK Cancel

To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **ProvisioningNwExt**. To open the job steps window, click ⊕ and double-click **Set Time** or select **Set Time** and click **OK**. The **Set Time** window opens. Enter the properties mentioned below and click **OK**. This adds the job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Running this step sets the time for each remote device in the job.

Property	Value	Description
Use supervisor time	bullet	Enables use of the time as set in the Supervisor PC to replicate in the remote stations.
Use selected time	bullet	Enables the Date/Time and Time Zone properties.
Date/Time	calendar	Sets the date and time.
Use NTP time	bullet	Enables the use of the time as supplied by the Network Time Protocol. This protocol synchronizes clocks among computer systems using a packet-switched, variable-latency data network.
Enabled	true (default) and false	Enables (true) and disables (false) use of NTP time.
NTP Host Mode	drop-down list	Selects the NTP host. Options are <i>Server</i> , <i>Server (Time Only)</i> and <i>Client</i> .

Property	Value	Description
Sync Local Clock to NTP	true (default) and false	Enables (true) and disables (false) local synchronization immediately.
Sync Time At Boot	true and false (default)	Enables (true) and disables (false) synchronization when the platform restarts.
Use Local Clock as Backup	true and false (default)	
Generate NTP Statistics	true and false (default)	Enables (true) and disables (false) the tracking of information about NTP accuracy.
table	read-only	Provides NTP statistics.

Setup Reciprocal Connection

This component supports the **Set Reciprocal Connection** step. Running this step sets up the **NiagaraNetwork** connection from the remote station to the Supervisor station for each station in the job.

Figure 72 Setup Reciprocal Niagara Network Connection properties

You add this component in the top pane (To run for each station) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

Property	Value	Description
Supervisor Username	text	Defines the Username required to access the Supervisor station.
Supervisor Password	text	Defines the Password required to access the Supervisor station.
Supervisor IP Address/Hostname	text	Defines the IP address or host name for the Supervisor station.

HTML5 Setup Reciprocal Connection

In EC-Net 4.14 and later, there is added browser support for the **Set Reciprocal Connection** step. Running this step sets up the **NiagaraNetwork** connection from the remote station to the Supervisor station for each station in the job. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 73 Setup Reciprocal Niagara Network Connection properties

Setup the Niagara Network connection from remote station to supervisor station

Supervisor Username (required)

Supervisor Password (required)

Supervisor IP Address/Hostname

OK Cancel

To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click and double-click **Set Reciprocal Connection** or select **Set Reciprocal Connection** and click **OK**. The **Set Reciprocal Connection** window opens. Enter the properties mentioned below and click **OK**. This adds the job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

Property	Value	Description
Supervisor Username	text	Defines the Username required to access the Supervisor station.
Supervisor Password	text	Defines the Password required to access the Supervisor station.
Supervisor IP Address/Hostname	text	Defines the IP address or host name for the Supervisor station.

Sign Certificate (provisioningNiagara-SignCertificateStep)

This component supports the **Sign Certificate** step. Running this step signs a certificate (typically a server certificate) for each remote platform in the job.

You will enter the alias of the certificate on the remote platform, and select the CA certificate alias for the signing certificate on the Supervisor platform.

The “Signing a certificate” section of the *Station Security Guide* contains more information about signing a server certificate.

Figure 74 Sign Certificate properties

Sign Certificate

Sign a certificate on each station

Server Certificate Alias

CA Certificate Alias

CA Certificate Password

Not Before

Not After

OK Cancel

You add this component in the top pane (**To run for each station**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

Property	Value	Description
Server Certificate Alias	text	Defines the certificate name.
CA Certificate Alias	text	Defines the CA (Certificate Authority) name.
CA Certificate Password	text	Defines the CA's password.
Not Before	date and time	Configures when the certificate becomes valid.
Not After	date and time	Configures when the certificate expires.

HTML5 Sign Certificate (provisioningNiagara-SignCertificateStep)

In EC-Net 4.14 and later, there is added browser support for the **Sign Certificate** step. Running this step signs a certificate (typically a server certificate) for each remote platform in the job. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

You will enter the alias of the certificate on the remote platform, and select the CA certificate alias for the signing certificate on the Supervisor platform.

The “Signing a certificate” section of the *Station Security Guide* contains more information about signing a server certificate.

Figure 75 Sign Certificate properties

Sign a certificate on each station

Server Certificate Alias

Server Certificate Password

Server Certificate To Use Global Password ☒ false

CA Certificate Alias Generate New ▾

CA Certificate Password

Not Before 28-Nov-23 2:12 PM IST

Not After 27-Nov-24 2:12 PM IST

OK Cancel

To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **ProvisioningNwExt**. To open the job steps window, click ⊕ and double-click **Sign Certificate** or select **Sign Certificate** and click **OK**. The **Sign Certificate** window opens. Enter the properties mentioned below and click **OK**. This adds the job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

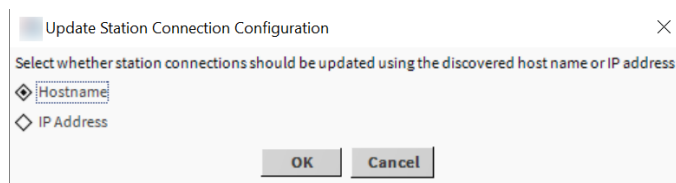
Property	Value	Description
Server Certificate Alias	text	Defines the certificate name.
Server Certificate Password	text	Defines the certificate password.
Server Certificate To Use Global Password	toggle button	Allows to use the global password for server certificate.
CA Certificate Alias	text	Defines the CA (Certificate Authority) name.

Property	Value	Description
CA Certificate Password	text	Defines the CA's password.
Not Before	date and time	Configures when the certificate becomes valid.
Not After	date and time	Configures when the certificate expires.

Update Connections Using Network Discovery

This component updates each Supervisor-to-station connection using **NiagaraNetwork** discovery.

Figure 76 Update Station Connection Configuration



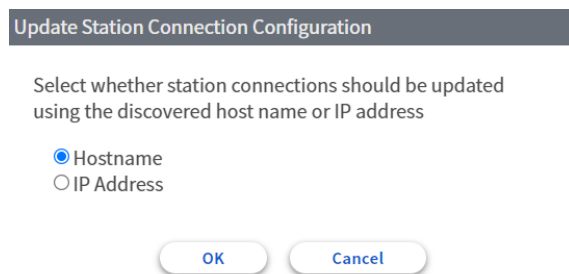
To access these job step components, double-click **ProvisioningNwExt** or **NiagaraNetworkJobPrototype**. The **Niagara Network Job Builder** or the **Niagara Network Prototype View** view opens. In the top pane **Provisioning steps to run** click the (+) symbol and click **Update Connections Using Niagara Network Discovery**.

Property	Value	Description
Hostname	text	Identifies the name of the remote station server.
IP Address	range	Defines the IP address of the remote station.

HTML5 Update Connections Using Network Discovery

In EC-Net 4.14 and later, there is added browser support for updating each Supervisor-to-station connection using **NiagaraNetwork** discovery. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 77 Update Station Connection Configuration



To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **ProvisioningNwExt**. To open the job steps window, click ⊕ and double-click **Update Station Connection Configuration** or select **Update Station Connection Configuration** and click **OK**. The **Update Station Connection Configuration** window opens. Enter the properties mentioned below and click **OK**. This adds the job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Running this step updates Supervisor to station connection.

Property	Value	Description
Hostname	text	Identifies the name of the remote station server.
IP Address	range	Defines the IP address of the remote station.

Update Licenses

This component updates the software license in each remote host.

You add this component in the top pane (Provisioning steps to run) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. This is the only step you can add as an initial step. When processed, the Supervisor gathers information about the licenses installed in each target host. Then, using a single message, it accesses the license server to determine if the licenses are up-to-date. If it finds updated licenses in the server, the system installs the license in each host, and updates the Supervisor's local license database.

If the Supervisor is not configured for Internet connectivity, the component uses only its local license database to compare against licenses installed on the target host(s). If it finds an updated license, the system installs the license in each host.

Updating the license may include changing the license brand. When you add this step to the **Niagara Network Job Builder** or **Niagara Network Prototype View**, a window prompts you to change the brand. If you change the brand and the job step finds the new brand in the license server it deletes the old brand and installs the new brand in each host.

HTML5 Update Licenses

In EC-Net 4.14 and later, there is added browser support for updating the software license in each remote host. The "HTML5" version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 78 Update Licenses

Update Licenses

Update all of the station(s), importing licenses from the licensing server

Running station(s) must be restarted for the licenses to become effective

☒ Restart station(s) after licenses update true

OK Cancel

To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click and double-click **Update Licences** or select **Update Licences** and click **OK**. The **Update Licences** window opens. Select **Restart Station(s) after license update** and click **OK**.

This adds the job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. This is the only step you can add as an initial step. When processed, the Supervisor gathers information about the licenses installed in each target host. Then, using a single message, it accesses the license server to determine if the licenses are up-to-date. If it finds updated licenses in the server, the system installs the license in each host, and updates the Supervisor's local license database.

If the Supervisor is not configured for Internet connectivity, the component uses only its local license database to compare against licenses installed on the target host(s). If it finds an updated license, the system installs the license in each host.

Update Template or Application Configuration

This component supports the **Update Template or Application Configuration** step. It updates the template configuration values of an installed template on each station.

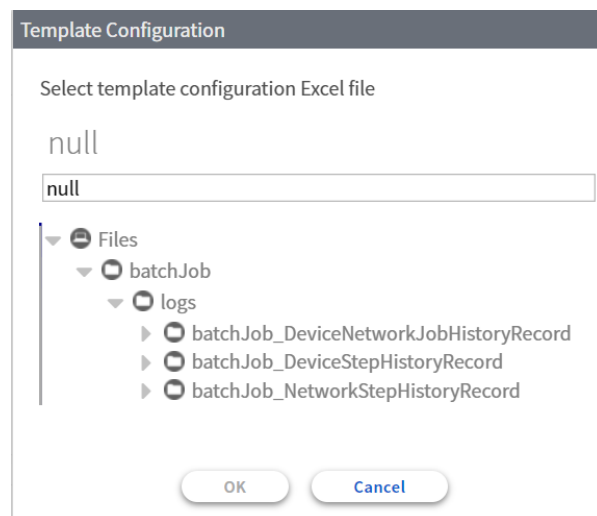
This step signs a certificate (typically a server certificate) on each remote device platform in the job. You enter the alias of the certificate on the remote platform, and select the CA certificate alias for the signing certificate on the Supervisor's platform.

You add this component in the top pane (**To run for each station**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

HTML5 Update Template or Application Configuration

In EC-Net 4.14 and later, there is added browser support the **Update Template or Application Configuration** step. It updates the template configuration values of an installed template on each station. The "HTML5" version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 79 Update Template or Application Configuration



To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click and double-click **Update Template or Application Configuration** or select **Update Template or Application Configuration** and click **OK**.

The **Template Configuration** window opens. Select the application template in the Supervisor station using an Excel file. This adds the job step in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Running this step updates the application template on the remote device platform in the job.

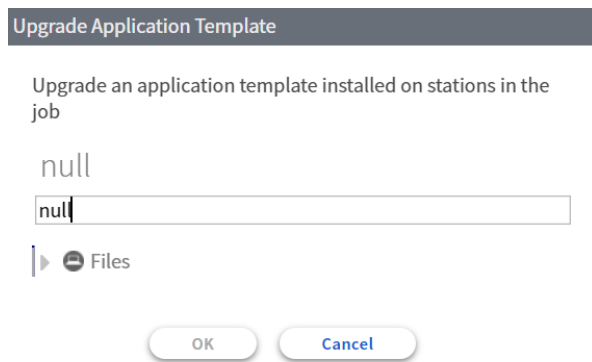
Upgrade Application Template

This component supports the **Upgrade Application Template** step. It upgrades the installed application template on each station.

You add this component in the top pane (**To run for each station**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

HTML5 Upgrade Application Template

In EC-Net 4.14 and later, there is added browser support supports the **Upgrade Application Template** step. It upgrades the installed application template on each station. The "HTML5" version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 80 Upgrade Application Template

To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click **+** and double-click **Upgrade Application Template** or select **Upgrade Application Template** and click **OK**. The **Upgrade Application Template** window opens. Select the template from the file chooser and click **OK**. This adds the job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Running this step upgrades the installed application template.

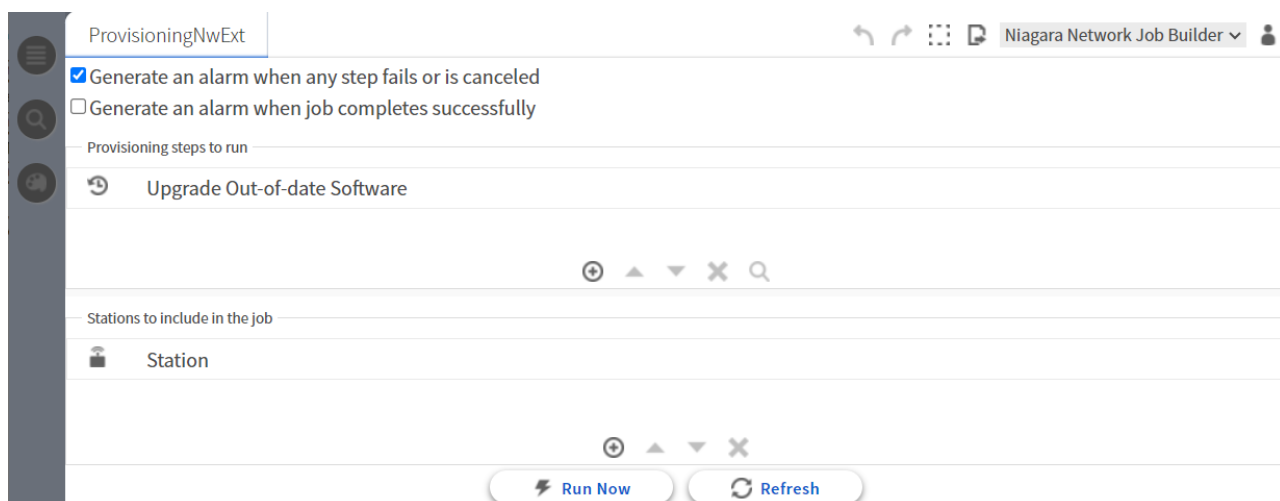
Upgrade Out-of-date Software


This component supports the **Upgrade Out-of-Date Software** step. When processed, the versions of software on the target host(s) are compared with the latest versions in the Supervisor's software database. If the system finds newer versions, this step installs them to the target host(s).

You add this component to the top pane (**To run for each station**) in the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

HTML5 Upgrade Out-of-date Software

In EC-Net 4.14 and later, there is added browser support the **Upgrade Out-of-Date Software** step. When processed, the versions of software on the target host(s) are compared with the latest versions in the Supervisor's software database. If the system finds newer versions, this step installs them to the target host(s). The "HTML5" version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 81 Upgrade Out-of-date Software

To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click  and double-click **Upgrade Out-of-Date Software** or select **Upgrade Out-of-Date Software** and click **OK**. This adds the job step in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Running this step upgrades the software with the latest version to the target host.

Upgrade Template

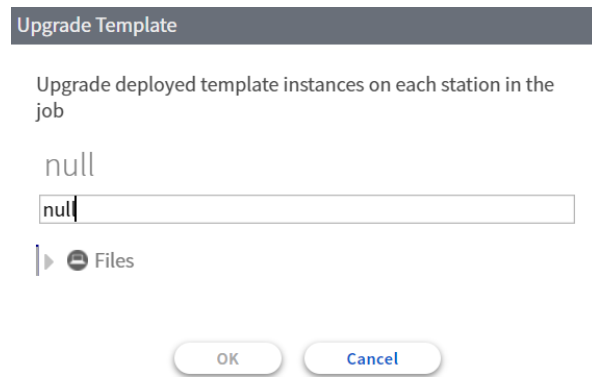
This component supports the **Upgrade Template** step. Running this step upgrades the deployed template instances on each station.


You add this component in the top pane (**To run for each station**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**.

HTML5 Upgrade Template

In EC-Net 4.14 and later, there is added browser support for the **Upgrade Template** step. Running this step upgrades the deployed template instances on each station. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view.

Figure 82 Upgrade Template



To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **.ProvisioningNwExt**. To open the job steps window, click  and double-click **Upgrade Template** or select **Upgrade Template** and click **OK**. The **Upgrade Template** window opens. Select the template from the file chooser and click **OK**. This adds the job step component in the top pane (**Provisioning steps to run**) of the **Niagara Network Job Builder** or the **Niagara Network Prototype View**. Running this step upgrades the deployed templates.

Chapter 9 Plugins (views)

Topics covered in this chapter

- ◆ Plugins in batchJob module
- ◆ Plugins in provisioningNiagara module

Plugins provide views of components and can be accessed in many ways. For example, double-click a component in the Nav tree to see its default view. In addition, you can right-click on a component and select from its **Views** menu.

Plugins in batchJob module

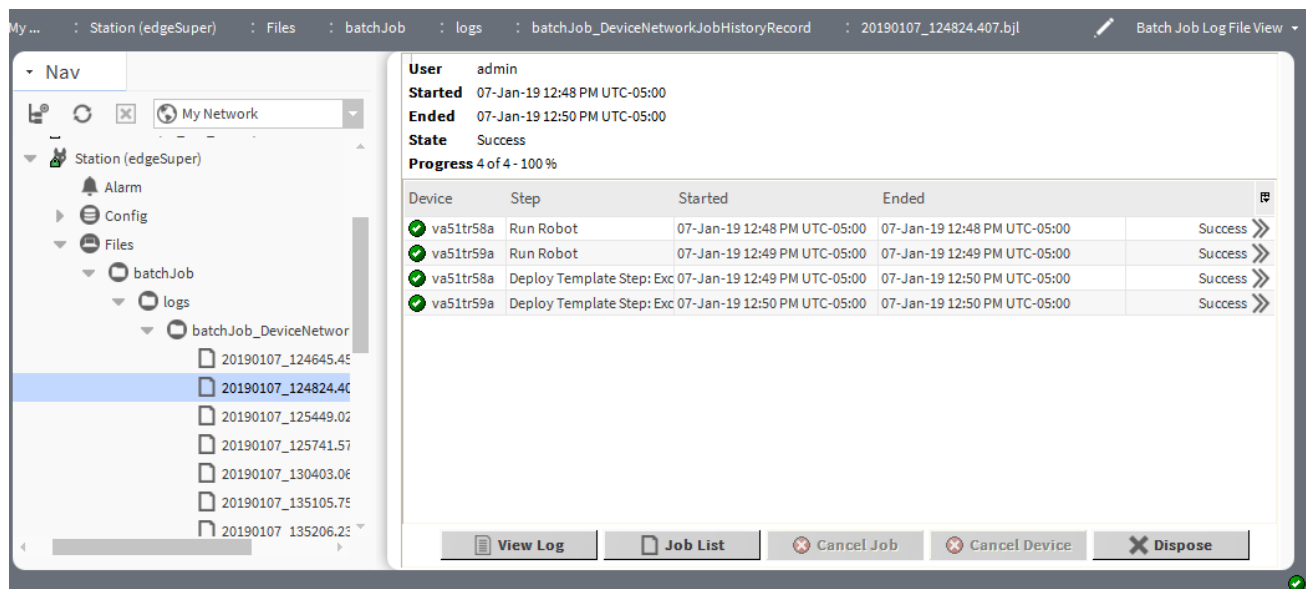
The provisioning batchJob module views list jobs and steps.

batchJob-BatchJobLogFileView

The **Batch Job Log File View** displays the details for the execution of a single provisioning job that has completed. It is the default view for any batch job log file (.bjl extension). You access this view by clicking the details (>>) button at the end of each row on the **Niagara Network Job List** or the **Prototype Job List**. You can also access this view when in the **Batch Job Step Log File View** by clicking the **Job Summary** button.

This view provides a summary table showing each step for each station in the job. To view additional **Log Details**, if any, for a specific step, double-click its row.

Figure 83 Batch Job Log File View lists the steps in one job



This view is nearly identical to the **Niagara Network Job View** (for a provisioning job that is still under the station's **JobService**). From top to bottom, this view has three areas:

- Job elements, which are read-only.
- A summary table of the steps in the job.
- A series of buttons at the bottom.

Elements

Element	Value	Description
User [provisioning]	text	Defines the station user that requested the job. This column displays <code>unknown</code> if job was triggered by a linked schedule.
Start or Started	date and time (read-only)	Displays the date and time that the system submitted the job to the job queue.
End or Ended [job]	date and time	Displays the date and time when the job stopped running. This property is blank if the job is still running.
Status or State	read-only text:	<p>Displays the current or final state of the job. The first three states appear on the Device Network Job view.</p> <ul style="list-style-type: none"> <code>Unknown</code> —the job is pending execution. <code>Running</code> — the job is executing. <code>Canceling</code> — request to cancel the job was sent, but has not been processed yet, and the job is still executing. <code>Success</code> — job finished successfully, with all steps completed for all stations. <code>Canceled</code> — job was canceled before it completed, and is no longer running. <code>Failed</code> — at least one step failed in one station; job is no longer running. <p>Each row in the table ends with a details button (>>) and a dispose button (X) . Clicking this button changes the view to the Niagara Network Job view or the Batch Job Step Log File view, which shows all logged messages that are related to this single job.</p> <p>The overall status for the job in other stations, may be different).</p>
Progress	job steps complete, total step count, percent complete	As the job executes, the progress updates as steps are started and completed. Note that the total step count can change while the job runs due to automated step combination (e.g. for software installation steps), automatic step creation (for license updates), and for step failures (where subsequent steps will not be run).

Table columns

Column	Value	Description
Device	element	Identifies the station that is being processed or has been processed.
Job Step	element	Identifies the type of provisioning step: Backup Stations, File Copy, Install Software, etc.
Start or Started	date and time (read-only)	Displays the date and time that the system submitted the job to the job queue.
End or Ended [step]	date and time	Displays the date and time when the step stopped running. This property is blank if the job is still running.

Buttons

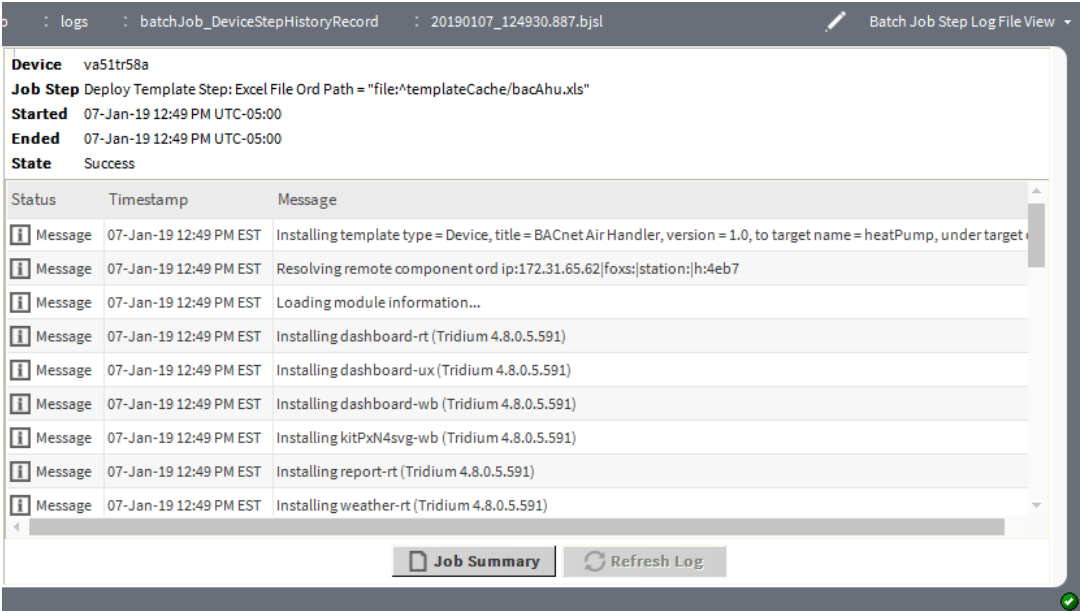
View Log	button enabled when a job row is selected	Opens a popup Job Log window that displays the messages output by the selected job or step.
Job List	button always enabled	Opens the Niagara Network Job List .
Cancel or Cancel Job	button enabled only if a job is running	Clicking this button notifies the system to cancel the job when it is safe to do so. Not all job steps can be canceled.
Cancel Device	button enabled only if a job is running and a step row is selected	For the selected station only, clicking this button notifies the system to cancel the job when it is safe to do so. The system begins processing the job for the next station.
Dispose	button enabled when a job is finished	<p>Clicking this button prompts you to confirm that you want to delete the selected job(s). The deletion includes all associated job files. If you confirm the deletion:</p> <ol style="list-style-type: none"> 1. The system deletes the job from the JobService (if it is still there and not rolled off as the 11th job, or station restart) 2. The system deletes all associated job files including the batch job log file, batch log step log files, and other files if applicable. For a backup job, this includes deleting the backup .dist file(s). 3. The system removes the job from the Jobs Table in the ProvisioningNwExt's Niagara Network Job List. <p>You can select multiple jobs to dispose of at the same time.</p>

batchJob-BatchJobStepLogFileView

The **Batch Job Step Log File View** shows details for one provisioning job step executed and completed against a single station. It is the default view for any batch job step log file (.bjsl extension). You access this view on a completed step by clicking the details button (>>) at the end of a step row in the **Niagara Network Job View** or **Station Job List**.

This view provides a summary table, which lists the log messages that occurred during step execution. To view additional **Log Details**, if any, for a specific step, double-click a step row.

Figure 84 Batch Job Step Log File View lists details on one step for a station.



From top-to-bottom this view has three areas:

- Step elements, which are read-only
- Step log table
- Buttons

Step elements

Elements	Value	Description
Device	element	Identifies the station that is being processed or has been processed.
Job Step	element	Identifies the type of provisioning step: Backup Stations, File Copy, Install Software, etc.
Start or Started	date and time (read-only)	Displays the date and time that the system submitted the job to the job queue.

Elements	Value	Description
End or Ended [job]	date and time	Displays the date and time when the job stopped running. This property is blank if the job is still running.
Status or State	read-only text:	<p>Displays the current or final state of the job. The first three states appear on the Device Network Job view.</p> <ul style="list-style-type: none"> Unknown — the job is pending execution. Running — the 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. <p>Each row in the table ends with a details button (>>) and a dispose button (X) . Clicking this button changes the view to the Niagara Network Job view or the Batch Job Step Log File view, which shows all logged messages that are related to this single job.</p> <p>The overall status for the job in other stations, may be different).</p>

Columns

Elements, columns and buttons	Value	Description
Status or State	read-only text	<p>Displays the current or final state of each step:</p> <ul style="list-style-type: none"> Running — the step is executing. Canceling — the request to cancel the step was sent, but has not been processed yet. The step is still running. Success — the step finished successfully. Canceled — the step was canceled before it completed and is no longer running. Failed — the step did not complete <p>Each row in the table ends with a details button (>>) and a dispose button (X) . This button functions the same as the Step Details button at the bottom of the view.</p> <p>The overall status for the step in other stations, may be different).</p>
File	text (visible only if a Backup Stations step)	Identifies the file path and name on the Supervisor for the saved backup .dist file. It uses the convention: ^provisioningNiagara/stationData/stationName/backups/backup_stationName_yymmdd_hhmm.dist
Timestamp	date and time	Displays the date and time when the log message was written.

Elements, columns and buttons	Value	Description
Message	text	Displays the log message.
Details	text	Provides additional information (or the beginning of such information) related to the step.

Batch Job Step File Log View buttons

Button	Value	Description
Job Summary	button always enabled	Changes to the Batch Job Log File View for the job that contains this step.
Refresh Log	button enabled when information needs to be refreshed	Recreates the log.
Restore	button available if the job step is Backup Station and the backup completed successfully	Restores the station using the .dist file saved from this provisioning job. If you answer Yes to the confirmation window (no undo), an install backup job executes immediately and the view changes to the Niagara Network Job View .

Run Robot buttons

Selecting a Run Robot step provides additional robot-specific buttons near the bottom of the view.

Figure 85 Run Robot step in Batch Job Step Log File View

Device	va51tr58a
Job Step	Run Robot
Started	26-Feb-19 12:49 PM EST
Ended	26-Feb-19 12:49 PM EST
State	Success

Status	Timestamp	Message	Details
Success	26-Feb-19 12:49 PM EST	Step successfully completed for va51tr58a	

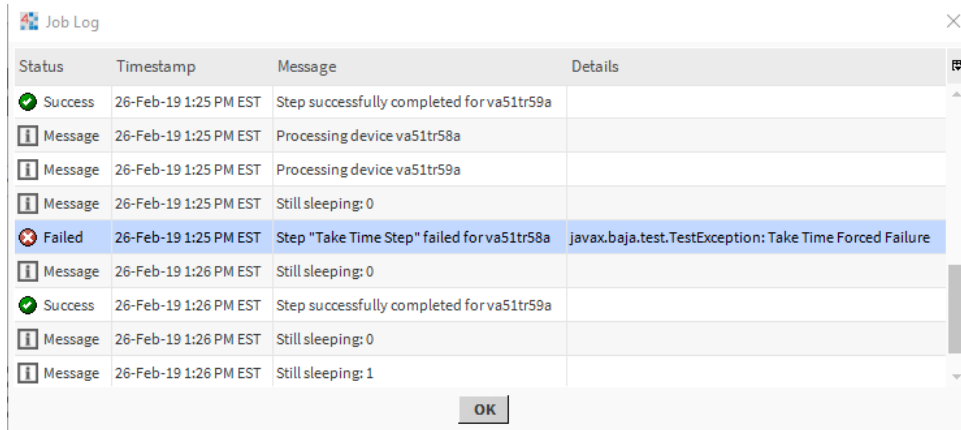
[Job Summary](#)
[Refresh Log](#)
[Robot Source](#)
[Robot Log](#)

Button	Value	Description
Job Summary	button always enabled	Changes to the Batch Job Log File View for the job that contains this step.
Refresh Log	button enabled when information needs to be refreshed	Recreates the log.
Robot Source	button always enabled	Displays a read-only listing of the ProvisioningRobot code used in the Run Robot step.
Robot Log	button always enabled	Displays any text written by the system to the robot's log Print-Writer during the execution of the step.

Job Log

The popup **Job Log** table opens a running log of messages output with the execution of the job. Each row includes a column for status, timestamp, message, and details.

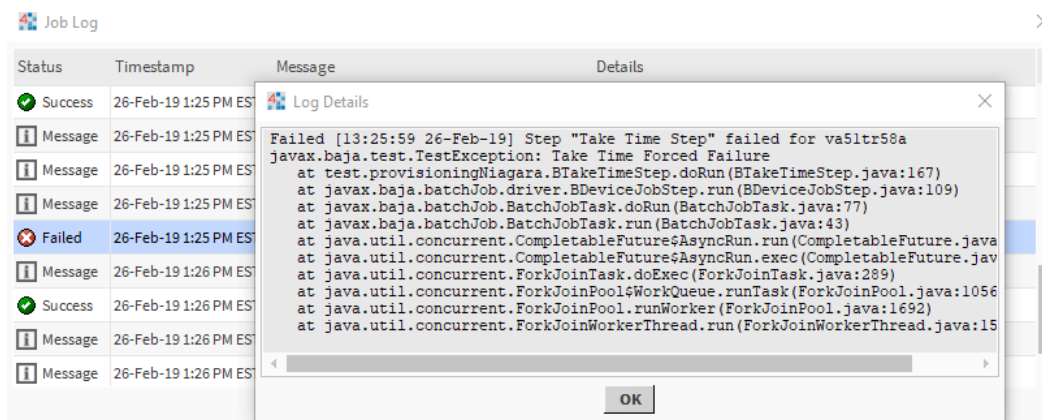
Figure 86 Example of a Job Log



Status	Timestamp	Message	Details
Success	26-Feb-19 1:25 PM EST	Step successfully completed for va51tr59a	
Message	26-Feb-19 1:25 PM EST	Processing device va51tr58a	
Message	26-Feb-19 1:25 PM EST	Processing device va51tr59a	
Message	26-Feb-19 1:25 PM EST	Still sleeping: 0	
Failed	26-Feb-19 1:25 PM EST	Step "Take Time Step" failed for va51tr58a	javax.baja.test.TestException: Take Time Forced Failure
Message	26-Feb-19 1:26 PM EST	Still sleeping: 0	
Success	26-Feb-19 1:26 PM EST	Step successfully completed for va51tr59a	
Message	26-Feb-19 1:26 PM EST	Still sleeping: 0	
Message	26-Feb-19 1:26 PM EST	Still sleeping: 1	

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 below.

Figure 87 Log Details example from Job Log

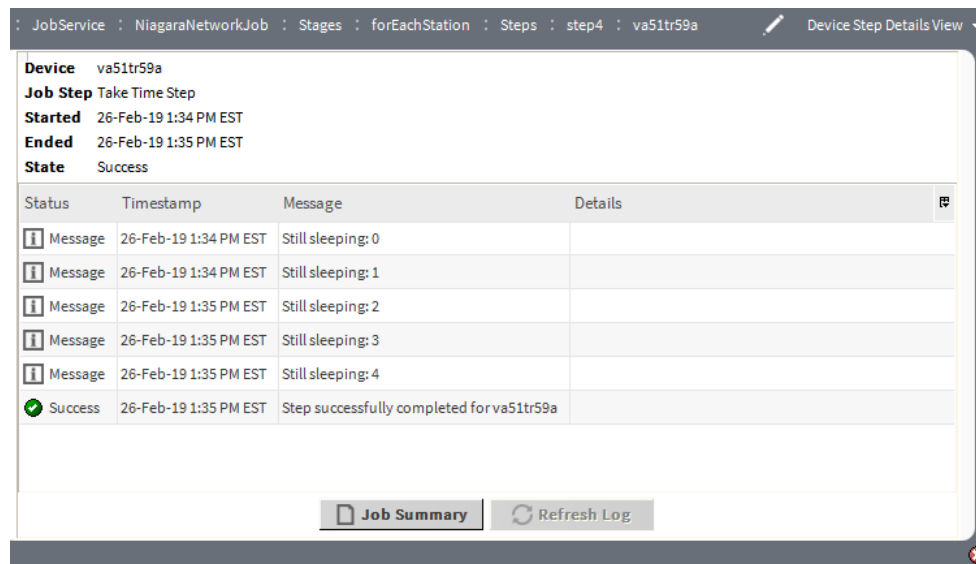


Status	Timestamp	Message	Details
Success	26-Feb-19 1:25 PM EST	Step successfully completed for va51tr59a	
Message	26-Feb-19 1:25 PM EST	Processing device va51tr58a	
Message	26-Feb-19 1:25 PM EST	Processing device va51tr59a	
Message	26-Feb-19 1:25 PM EST	Still sleeping: 0	
Failed	26-Feb-19 1:25 PM EST	Step "Take Time Step" failed for va51tr58a	<div data-bbox="435 1050 1182 1383"> <p>Log Details</p> <p>Failed [13:25:59 26-Feb-19] Step "Take Time Step" failed for va51tr58a javax.baja.test.TestException: Take Time Forced Failure at test.provisioningNiagara.BTakeTimeStep.doRun(BTakeTimeStep.java:167) at javax.baja.batchJob.driver.BDeviceJobStep.run(BDeviceJobStep.java:109) at javax.baja.batchJob.BatchJobTask.doRun(BatchJobTask.java:77) at javax.baja.batchJob.BatchJobTask.run(BatchJobTask.java:43) at java.util.concurrent.CompletableFuture\$AsyncRun.run(CompletableFuture.java:1660) at java.util.concurrent.CompletableFuture\$AsyncRun.exec(CompletableFuture.java:1674) at java.util.concurrent.ForkJoinTask.doExec(ForkJoinTask.java:289) at java.util.concurrent.ForkJoinPool\$WorkQueue.runTask(ForkJoinPool.java:1056) at java.util.concurrent.ForkJoinPool.runWorker(ForkJoinPool.java:1692) at java.util.concurrent.ForkJoinWorkerThread.run(ForkJoinWorkerThread.java:158)</p> </div>
Message	26-Feb-19 1:26 PM EST	Still sleeping: 0	
Success	26-Feb-19 1:26 PM EST	Step successfully completed for va51tr59a	
Message	26-Feb-19 1:26 PM EST	Still sleeping: 0	
Message	26-Feb-19 1:26 PM EST	Still sleeping: 1	

batchJob-JobStepDetailsView

The **Job Step Details View** shows the details for the provisioning job step that is currently running on a single station. You access this view from either the **Niagara Network Job View** or the **Station Job List** when you click the "<<" button to the right of the running step. After the job completes and you ask for step details, instead of this view the system opens the **Batch Job Step Log File View**.

This view provides a summary table showing the log messages that occurred during step execution. To show additional **Log Details**, if any, double-click a step row in the table.

Figure 88 Details for the currently-running step on a single station

If the currently-running station step is a backup of the station, instead of this view, the system opens a slightly different view: the **Backup Step Details View**. The view for a **Run Robot step** provides additional buttons.

Elements

Elements, columns and buttons	Value	Description
Device	element	Identifies the station that is being processed or has been processed.
Job Step	element	Identifies the type of provisioning step: Backup Stations, File Copy, Install Software, etc.
Start or Started	date and time (read-only)	Displays the date and time that the system submitted the job to the job queue.
End or Ended [job]	date and time	Displays the date and time when the job stopped running. This property is blank if the job is still running.
Status or State	read-only text:	<p>Displays the current or final state of the job. The first three states appear on the Device Network Job view.</p> <ul style="list-style-type: none"> Unknown —the job is pending execution. Running — the 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. <p>Each row in the table ends with a details button (>>) and a dispose button (X) . Clicking this button changes the view to the Niagara Network Job view or the Batch Job Step Log File view,</p>

Elements, columns and buttons	Value	Description
		<p>which shows all logged messages that are related to this single job.</p> <p>The overall status for the job in other stations, may be different).</p>
File	text (visible only if a Backup Stations step)	Identifies the file path and name on the Supervisor for the saved backup .dist file. It uses the convention: ^provisioningNiagara/stationData/stationName/backups/backup_stationName_yymmdd_hhmm.dist
Status or State	read-only text	<p>Displays the current or final state of each step:</p> <ul style="list-style-type: none"> Running — the step is executing. Canceling — the request to cancel the step was sent, but has not been processed yet. The step is still running. Success — the step finished successfully. Canceled — the step was canceled before it completed and is no longer running. Failed — the step did not complete <p>Each row in the table ends with a details button (>>) and a dispose button (X) . This button functions the same as the Step Details button at the bottom of the view.</p> <p>The overall status for the step in other stations, may be different).</p>
Timestamp	date and time	Displays the date and time when the log message was written.
Message	text	Displays the log message.
Details	text	Provides additional information (or the beginning of such information) related to the step.

Columns

Elements, columns and buttons	Value	Description
Device	element	Identifies the station that is being processed or has been processed.
Job Step	element	Identifies the type of provisioning step: Backup Stations, File Copy, Install Software, etc.
Start or Started	date and time (read-only)	Displays the date and time that the system submitted the job to the job queue.
End or Ended [job]	date and time	Displays the date and time when the job stopped running. This property is blank if the job is still running.
Status or State	read-only text	<p>Displays the current or final state of each step:</p> <ul style="list-style-type: none"> Running — the step is executing. Canceling — the request to cancel the step was sent, but has not been processed yet. The step is still running. Success — the step finished successfully.

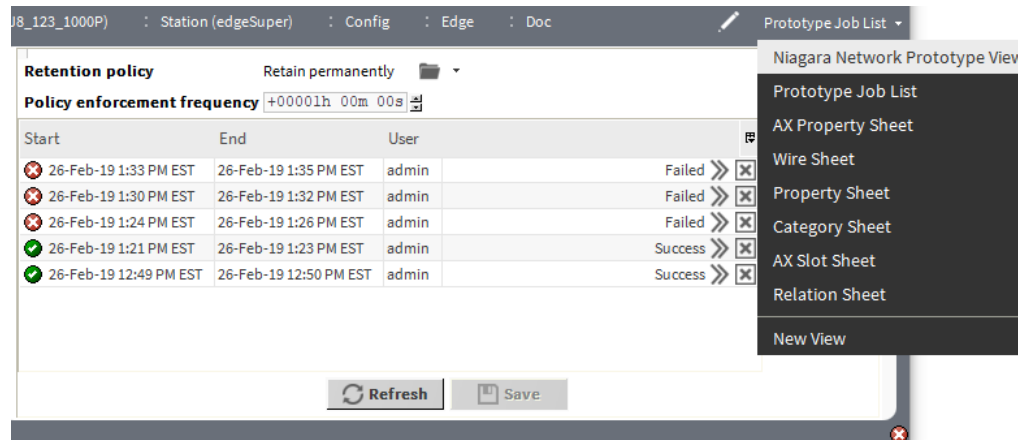
Elements, columns and buttons	Value	Description
		<ul style="list-style-type: none"> Canceled — the step was canceled before it completed and is no longer running. Failed — the step did not complete <p>Each row in the table ends with a details button (>>) and a dispose button (X) . This button functions the same as the Step Details button at the bottom of the view.</p> <p>The overall status for the step in other stations, may be different).</p>
File	text (visible only if a Backup Stations step)	Identifies the file path and name on the Supervisor for the saved backup .dist file. It uses the convention: ^provisioningNiagara/stationData/stationName/backups/backup_stationName_yymmdd_hhmm.dist
Status or State	read-only text	<p>Displays the current or final state of each step:</p> <ul style="list-style-type: none"> Running — the step is executing. Canceling — the request to cancel the step was sent, but has not been processed yet. The step is still running. Success — the step finished successfully. Canceled — the step was canceled before it completed and is no longer running. Failed — the step did not complete <p>Each row in the table ends with a details button (>>) and a dispose button (X) . This button functions the same as the Step Details button at the bottom of the view.</p> <p>The overall status for the step in other stations, may be different).</p>
Timestamp	date and time	Displays the date and time when the log message was written.
Message	text	Displays the log message.
Details	text	Provides additional information (or the beginning of such information) related to the step.

Buttons

Button	Value	Description
Job Summary	button always enabled	Changes to the Batch Job Log File View for the job that contains this step.
Refresh Log	button enabled when information needs to be refreshed	Recreates the log.

batchJob-PrototypeJobList

This view provides a table-based history of this batch job and its results. It is available on any reusable provisioning job component (**NiagaraNetworkJobPrototype**), and is accessed by using the view selector, or by right-clicking the component and selecting **Views**→ **Prototype Job List**.

Figure 89 Prototype Job List is another view of a NiagaraNetworkJobPrototype component

This view has just one list area, a table with buttons on the right and the bottom. It differs from the **Niagara Network Job List** (**NiagaraNetwork's ProvisioningNewExt** 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. Setting retention policy is recommended for any job that includes ongoing periodic backups.
- The table shows only jobs from the (one) associated job prototype (provisioning job), whereas the **Niagara Network Job List** table shows all retained provisioning jobs.

Retention Policy

These rules determine what happens to the completed provisioning job.

Retention policy rule	Value	Description
Retain permanently	n/a	Causes all executed batch jobs to remain in the Supervisor's station job management system until manually disposed of.
Dispose after a specified amount of time	days, hours, and minutes (default 7 days) <small>Retention policy Dispose after 00:07:00</small>	Causes executed batch jobs to be deleted after a period of time relative to the job's end timestamp.
Keep a limited number of executions	number and check box <small>Retention policy Executions to retain 10 [x] Count only successful executions</small> The check box defaults to count only successful executions.	Causes the system to save only the most recent number of executed batch jobs, after which older jobs are deleted. Removing the selection causes the system to include all job attempts.
Policy enforcement frequency	hours, minutes, seconds (default is one hour)	Establishes the periodic frequency at which the job's retention policy is evaluated and enforced. You can manually invoke the Enforce Retention Policy action on the job component.

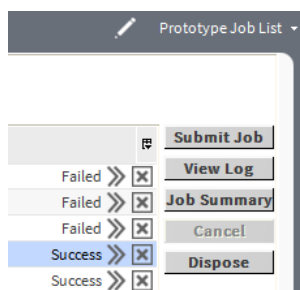
Columns

This main area of the **Prototype Jobs List** shows provisioning jobs that have been sent to run, are running, or are have completed. Any pending jobs do not appear until the job prototype's linked trigger schedule actually fires.

Column	Value	Description
Start or Started	date and time (read-only)	Displays the date and time that the system submitted the job to the job queue.
End or Ended [step]	date and time	Displays the date and time when the step stopped running. This property is blank if the job is still running.
User	text	Defines the station user that requested the job. This column displays <code>unknown</code> if job was triggered by a linked schedule.
Status or State	read-only text:	<p>Displays the current or final state of the job. The first three states appear on the Device Network Job view.</p> <ul style="list-style-type: none"> <code>Unknown</code> —the job is pending execution. <code>Running</code> — the job is executing. <code>Canceling</code> — request to cancel the job was sent, but has not been processed yet, and the job is still executing. <code>Success</code> — job finished successfully, with all steps completed for all stations. <code>Canceled</code> — job was canceled before it completed, and is no longer running. <code>Failed</code> — at least one step failed in one station; job is no longer running. <p>Each row in the table ends with a details button (>>) and a dispose button (X) . Clicking this button changes the view to the Niagara Network Job view or the Batch Job Step Log File view, which shows all logged messages that are related to this single job.</p> <p>The overall status for the job in other stations, may be different).</p>

Buttons

Figure 90 Buttons along right side of Prototype Job List view.



Button	Value	Description
Submit Job	always enabled	Lets you request that this provisioning job run now. The system adds a new job row at the top of the jobs table, dynamically updating the job status to indicate the job's progress.
View Log	button enabled when a job row is selected	Opens a popup Job Log window that displays the messages output by the selected job or step.

Button	Value	Description
Job Summary	button always enabled	Changes to the Batch Job Log File View for the job that contains this step.
Cancel or Cancel Job	button enabled only if a job is running	Clicking this button notifies the system to cancel the job when it is safe to do so. Not all job steps can be canceled.
Dispose	button enabled when a job is finished	<p>Clicking this button prompts you to confirm that you want to delete the selected job(s). The deletion includes all associated job files. If you confirm the deletion:</p> <ol style="list-style-type: none"> 1. The system deletes the job from the JobService (if it is still there and not rolled off as the 11th job, or station restart) 2. The system deletes all associated job files including the batch job log file, batch log step log files, and other files if applicable. For a backup job, this includes deleting the backup .dist file(s). 3. The system removes the job from the Jobs Table in the ProvisioningNwExt's Niagara Network Job List. <p>You can select multiple jobs to dispose of at the same time.</p>

Plugins in provisioningNiagara module

Provisioning includes use of the following **provisioningNiagara** views (listed alphabetically).

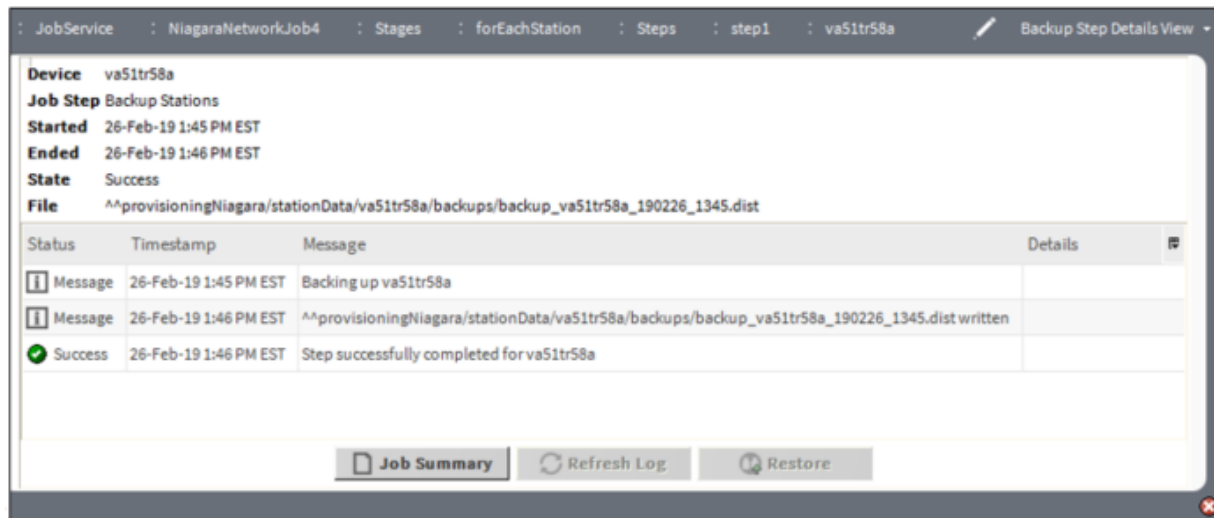
- Backup Step Details View
- Network License Summary
- Niagara Network Job Builder
- Niagara Network Job List
- Niagara Network Job View
- Niagara Network Prototype View
- Provisioning Manager
- ProvisioningRobotEditor
- Provisioning Station Director
- Station Job List
- Station Software View
- Supervisor 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. You access this from the **Niagara Network Job View** or the **Station Job List** by clicking ">>" next to the running step.

If the job step is other than a Backup Stations step, and the step is still running, you see a slightly different view, the **Niagara Network Job View**. In either case, once the step finishes, neither view is accessible. Both are replaced by the **Batch Job Step Log File View**.

This view provides a summary table of log messages that occur during step execution. To view additional **Log Details**, if any, double-click an individual row.

Figure 91 Backup Step Details View with details on one running backup step for a station

NOTE: Provisioning backups work using either the local BackupService or the CloudBackupService. When the CloudBackupService is installed it provides all the functions of the local BackupService. You can make successful backups using either the **Backup Manager** view to create manual backups or using the **Backup Stations** step in provisioning jobs in the Supervisor. Backups will contain the expected files based on the CloudBackupService exclusions. Backups created by a provisioning job prototype follow the configured job disposition settings and will be deleted when new backups are created.

Step elements

Elements, columns and buttons	Value	Description
Device	element	Identifies the station that is being processed or has been processed.
Job Step	element	Identifies the type of provisioning step: Backup Stations, File Copy, Install Software, etc.
Start or Started	date and time	Displays the date and time that the step began processing.
End or Ended [job]	date and time	Displays the date and time when the job stopped running. This property is blank if the job is still running.

Elements, columns and buttons	Value	Description
Status or State	read-only text	<p>Displays the current or final state of each step:</p> <ul style="list-style-type: none"> Running — the step is executing. Canceling — the request to cancel the step was sent, but has not been processed yet. The step is still running. Success — the step finished successfully. Canceled — the step was canceled before it completed and is no longer running. Failed — the step did not complete <p>Each row in the table ends with a details button (>>) and a dispose button (X) . This button functions the same as the Step Details button at the bottom of the view.</p> <p>The overall status for the step in other stations, may be different).</p>
File	text (visible only if a Backup Stations step)	Identifies the file path and name on the Supervisor for the saved backup .dist file. It uses the convention: ^provisioningNiagara/stationData/stationName/backups/backup_stationName_yymmdd_hhmm.dist

Columns

Columns	Value	Description
Status or State	read-only text	<p>Displays the current or final state of each step:</p> <ul style="list-style-type: none"> Running — the step is executing. Canceling — the request to cancel the step was sent, but has not been processed yet. The step is still running. Success — the step finished successfully. Canceled — the step was canceled before it completed and is no longer running. Failed — the step did not complete <p>Each row in the table ends with a details button (>>) and a dispose button (X) . This button functions the same as the Step Details button at the bottom of the view.</p> <p>The overall status for the step in other stations, may be different).</p>
Timestamp	date and time	Displays the date and time when the log message was written.
Message	text	Displays the log message.

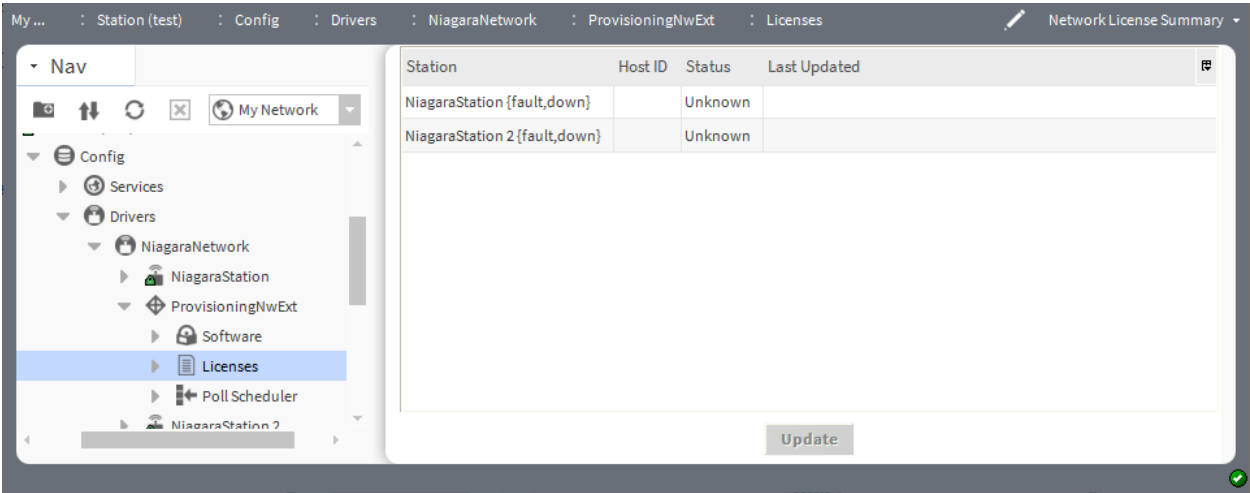
Buttons

Button	Value	Description
Job Summary	button always enabled	Changes to the Batch Job Log File View for the job that contains this step.
Refresh Log	button enabled when information needs to be refreshed	Recreates the log.
Restore	button available if the job step is Backup Station and the backup completed successfully	Restores the station using the .dist file saved from this provisioning job. If you answer Yes to the confirmation window (no undo), an install backup job executes immediately and the view changes to the Niagara Network Job View .

Network License Summary view (provisioningNiagara-NetworkLicenseSummary)

This view provides lists the currently-known license information for each **NiagaraStation** in the network.

Figure 92 Network Licenses Summary view



Each row contains the license information for a host running a station.

To access this view expand **Config**→**Drivers**→**NiagaraNetwork**→**ProvisioningNwExt** and double-click **Licenses**.

Column	Value	Description
Station	text	Identifies the name of the station.
Host ID	text	Provides a 20-character identifier that uniquely identifies each host.
Status	Up-To-Date	A status of Up To Date means that the license on the remote host agrees with the license that theSupervisor has for it in its (own) local license database. It may be possible that a more recent license is available for it on the licensing server.
Last Updated	date and time	The timestamp when the station’s license was last updated.

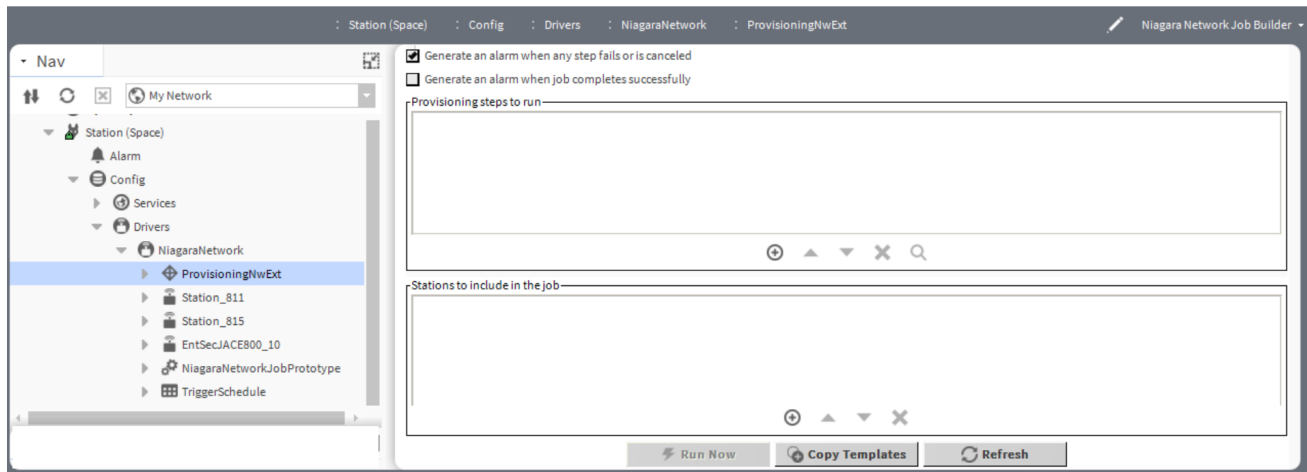
provisioningNiagara-NiagaraNetworkJobBuilder

This view creates a one-time provisioning job for immediate execution (to run now). It is the default view on the **ProvisioningNwExt** component.

Each provisioning job consists of one or more steps that the Supervisor's **JobService** controls and the remote station's **ProgramService** executes on the remote host. These steps include station backup, update and file copy, among others. Using this view, you specify the steps to be performed by the one-time job, and for which stations. Then you submit the job without saving the job as a reusable component.

NOTE: To build a job that you can schedule or run at a later time, save, duplicate and modify, use a **Niagara-NetworkJobPrototype** component copied anywhere in the Supervisor's station (each has its own equivalent view). You can find them in the **provisioningNiagara** palette. In general, those are the components you should use to create regularly scheduled station backup jobs.

Figure 93 Niagara Network Job Builder is the default view of ProvisioningNwExt



To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and double-click **ProvisioningNwExt**.






Alarm check boxes

Alarm check box	Value	Description
Generate an alarm when any step fails or is canceled	check box	Determines if the BatchJobService is to issue alarms for this provisioning job, and under what circumstances. Alarms use the alarm class specified in the BatchJobService and appear in the alarm console as alerts. When selected, the BatchJobService raises an alarm whenever a job step fails or is canceled.
Generate an alarm when job completes successfully	check box	Configures the BatchJobService too raise an alarm when a job completes with no step failures.

Steps and stations

List area	Value	Description
Provisioning steps to run	text	Provides 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. For specific steps, refer to Steps, page 167 .
Stations to include in the job	text	Lists all the stations to be processed by the job. This means each station processes all steps in the (middle pane) Steps for Each Station List . Only stations in the Supervisor's Niagara-Network 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).

Controls

Control	Description
	Adds a step.
	Moves the selected step up in the sequence of steps.
	Moves the selected step down in the sequence of steps.
	Removes the selected step.
	View and edit the details of the selected step.

Buttons

- **Run Now** dispatches the job to the batch job queue for immediate execution. The view automatically changes to the **Niagara Network Job** view. This button is enabled when there is at least one job step in either the **Initial Steps To Run Once List** or **Steps for Each Station List**, and one station in the **Stations List**.
- **Copy Templates** opens a **Copy Templates** chooser.
- **Refresh** removes all entries from the three lists.

Steps

Figure 94 The New Job Step window



The screen captures shows all provisioning job steps to run for each station. This list includes all steps:

- **Add Station User** (component: **AddStationUserStep**) – adds a new user to the station.
- **Backup Stations** (component: **BackupStationExt**) – makes an online backup of each running station.
- **Convert To Perpetual Licence Mode** — Converts all the stations' licensing mode from subscription to perpetual
- **Copy Local File** (component: **FileCopyStep**) – makes a copy of a local file on your PC.
- **Copy Supervisor File** (component: **FileCopyStep**) – makes a copy of a Supervisor file.
- **Deploy Template** (component: **DeployTemplateStep**) – deploys a component template to each station using and Excel file.
- **Enable Bootstrap Mode** (component: **EnableBootstrapStep**) – creates a one-time provisioning job to connect from the Supervisor to the remote controller using a certificate exemption for the default self-signed certificate. This step is intended for use with other provisioning steps that set up credentials and configure certificates. It is useful for provisioning new out-of-the-box devices.
- **Export Application Template Configuration** — Exports an application template configuration as an xlsx file
- **Export Certificate Signing Request** (component: **ExportCsrJobStep**) - generates and exports a CSR for each remote host.
- **Import Signed Certificate** (component: **ImportSignedCertificateJobStep**) – imports a signed certificate for each remote host.
- **Install a Signed Certificate for an Individual Service (Fox, Web, etc)** – Install an Individual Signed Cert Config to onboard/renew a signed certificate for a particular service, like the **FoxService** or **WebService**.
- **Install a Signed Certificate for Fox/Web/Platform to share** – Install a Combined Signed Cert Config to onboard/renew a signed certificate for Fox, Web, and/or Platform to share.
- **Install Application Template** (component: **DeployApplicationStep**) – installs an application template on each station using an Excel file.
- **Install Certificate** (component: **InstallCertificateJobStep**) – installs a certificate from the supervisor's user trust store to each remote host's user trust store.
- **Install Clean Distribution** (component: **InstallCleanDistStep**) – installs the selected clean distribution file to the target platforms.
- **Install Software** (component: **InstallBySpecStep**) – installs a software module in one or more remote hosts.
- **Reboot** (component: **RebootJobStep**) – reboots the platform host.
- **Remove Property** (component: **RemovePropertyStep**) – removes a property from a component at a defined slot path for each station in the provisioning job.
- **Remove Station User** (component: **RemoveStationUserStep**) – removes a named user from the station.
- **Rename Device Station** (component: **RenameStationStep**) – renames the station on the remote host to match the name in the supervisor's **NiagaraNetwork**.
- **Run Robot** (component: **RunRobotStep**) – allows you to create custom code to run on each station.
- **Security Job Steps** is a convenience step that generates three separate sets of credentials in one place and in one sequence. The individual steps are **Set Platform Credentials**, **Set Station User Password** and **Set Platform User Password**.
- **Set Certificate Alias** (component: **SetCertificateAliasJobStep**) – sets the certificate alias used for the remote device's platform, web, and fox services.
- **Set Platform Credentials** (component: **SetPlatformCredentialsJobStep**) – adds a new platform user account and removes the default account on the remote device.

- **Set Platform User Password** (component: **SetPlatformUserPasswordJobStep**) – changes password for an existing platform user.
- **Set Property** (component: **SetPropertyJobStep**) – set or add a property.
- **Set Station Connection Credentials** (component: **SetStationConnectionCredentialsStep**) – sets the station credentials used to connect to the remote station.
- **Set Station User Password** (component: **SetStationUserPasswordJobStep**) – sets the station user password.
- **Set System Passphrase** (component: **SetSystemPassphraseJobStep**) - sets the system passphrase.
- **Set TLS Level** (component: **SetTlsLevelJobStep**) – sets the minimum TLS level for the remote device's platform, web, and fox services.
- **Set Time** (component: **SetTimeJobStep**) - sets the time on the remote device.
- **Setup Reciprocal Connection** (component: **SetupReciprocalConnectionStep**) – sets the connection settings for the "reciprocal" station connection, which is the station proxy in the **NiagaraNetwork** on the remote station that points back to the Supervisor.
- **Sign Certificate** (component: **SignCertificateStep**) signs a certificate (typically a server certificate) for each remote platform in the job.
- **Update Connections Using Niagara Network Discovery** (component: **NiagaraNetworkDiscoveryStep**) – updates station connections using **NiagaraNetwork** discovery.
- **Update Connections Using Provisioning Station's DHCP Server** (component: **DhcpDiscoveryStep**) – updates station connections using the provisioning station's DHCP server leases.
- **Upgrade Out-of-Date Software** (component: **SoftwareStationExt**) – installs software, upgrades out-of-date software, supports copy supervisor step, and the reboot step.
- **Update Template or Application Configuration** (component: **UpdateConfigurationStep**) updates the template configuration values of an installed template in each remote station.
- **Update Licenses** job step (component: **UpdateLicensesJobStep**) – updates software license in each remote host.
- **Upgrade Application Template** (**UpgradeApplicationStep**) upgrades the installed application template in each remote host.
- **Upgrade Out-of-Date Software** (**UpgradeOutOfDateStep**) compares module versions on the target hosts with the latest versions in the Supervisor's software database and installs updates as needed.
- **Upgrade Template** (component: **UpgradeTemplateStep**) - upgrades deployed component template instances on each station using and Excel file.

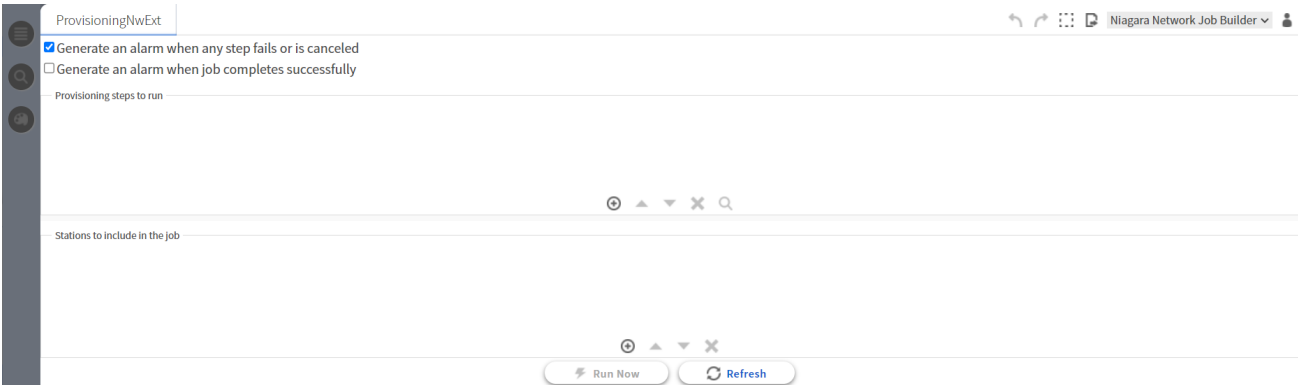
HTML5 Niagara Network Job Builder

In EC-Net 4.14 and later, there is added browser support for the Lon Network Router Manager View. The "HTML5" version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view. This view creates a one-time provisioning job for immediate execution (to run now). It is the default view on the **ProvisioningNwExt** component.

Each provisioning job consists of one or more steps that the Supervisor's **JobService** controls and the remote station's **ProgramService** executes on the remote host. These steps include station backup, update and file copy, among others. Using this view, you specify the steps to be performed by the one-time job, and for which stations. Then you submit the job without saving the job as a reusable component.

NOTE: To build a job that you can schedule or run at a later time, save, duplicate and modify, use a **Niagara-NetworkJobPrototype** component copied anywhere in the Supervisor's station (each has its own equivalent view). You can find them in the **provisioningNiagara** palette. In general, those are the components you should use to create regularly scheduled station backup jobs.

Figure 95 HTML5 Niagara Network Job Builder



To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click **ProvisioningNwExt**.

Alarm check boxes

Alarm check box	Description
Generate an alarm when any step fails or is canceled	Determines if the BatchJobService is to issue alarms for this provisioning job, and under what circumstances. Alarms use the alarm class specified in the BatchJobService and appear in the alarm console as alerts. When selected, the BatchJobService raises an alarm whenever a job step fails or is canceled.
Generate an alarm when job completes successfully	Configures the BatchJobService too raise an alarm when a job completes with no step failures.

Steps and Stations

List Area	Description
Provisioning steps to run	Provides 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. For specific steps, refer to . Steps, page 171
Stations to include in the job	Lists all the stations to be processed by the job. 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).

Controls

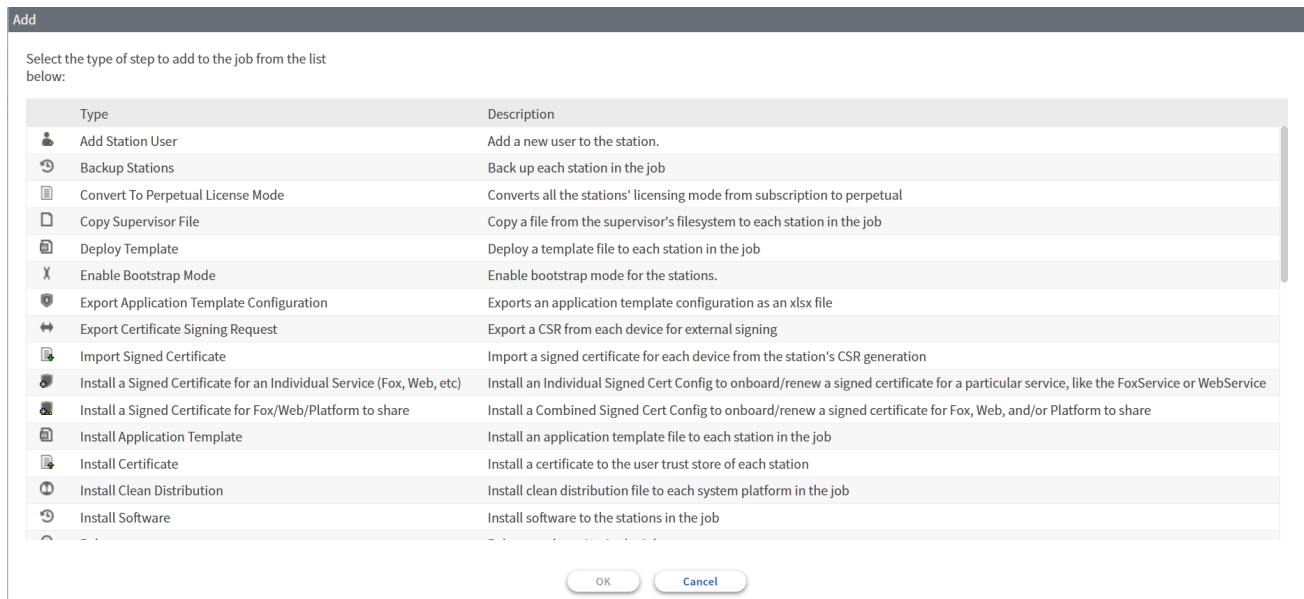
Control	Description
⊕	Adds a step.
▲	Moves the selected step up in the sequence of steps.
▼	Moves the selected step down in the sequence of steps.
✕	Removes the selected step.
🔍	View and edit the details of the selected step.

Buttons

- **Run Now** dispatches the job to the batch job queue for immediate execution. The view automatically changes to the **Niagara Network Job** view. This button is enabled when there is at least one job step in either the **Initial Steps To Run Once List** or **Steps for Each Station List**, and one station in the **Stations List**.
- **Refresh** removes all entries from the three lists.

Steps

Figure 96 The Add Steps Window







The screen captures shows all provisioning job steps to run for each station. This list includes all steps:

- **Add Station User** (component: **AddStationUserStep**) – adds a new user to the station.
- **Backup Stations** (component: **BackupStationExt**) – makes an online backup of each running station.
- **Convert To Perpetual Licence Mode** — Converts all the stations' licensing mode from subscription to perpetual
- **Copy Local File** (component: **FileCopyStep**) – makes a copy of a local file on your PC.
- **Copy Supervisor File** (component: **FileCopyStep**) – makes a copy of a Supervisor file.
- **Deploy Template** (component: **DeployTemplateStep**) – deploys a component template to each station using and Excel file.
- **Enable Bootstrap Mode** (component: **EnableBootstrapStep**) – creates a one-time provisioning job to connect from the Supervisor to the remote controller using a certificate exemption for the default self-signed certificate. This step is intended for use with other provisioning steps that set up credentials and configure certificates. It is useful for provisioning new out-of-the-box devices.
- **Export Application Template Configuration** — Exports an application template configuration as an excel file
- **Export Certificate Signing Request** (component: **ExportCsrJobStep**) - generates and exports a CSR for each remote host.
- **Import Signed Certificate** (component: **ImportSignedCertificateJobStep**) – imports a signed certificate for each remote host.

- **Install a Signed Certificate for an Individual Service (Fox, Web, etc)** – Install an Individual Signed Cert Config to onboard/renew a signed certificate for a particular service, like the **FoxService** or **WebService**.
- **Install a Signed Certificate for Fox/Web/Platform to share** – Install a Combined Signed Cert Config to onboard/renew a signed certificate for Fox, Web, and/or Platform to share.
- **Install Application Template** (component: **DeployApplicationStep**) – installs an application template on each station using an Excel file.
- **Install Certificate** (component: **InstallCertificateJobStep**) – installs a certificate from the supervisor's user trust store to each remote host's user trust store.
- **Install Clean Distribution** (component: **InstallCleanDistStep**) – installs the selected clean distribution file to the target platforms.
- **Install Software** (component: **InstallBySpecStep**) – installs a software module in one or more remote hosts.
- **Reboot** (component: **RebootJobStep**) – reboots the platform host.
- **Remove Property** (component: **RemovePropertyStep**) – removes a property from a component at a defined slot path for each station in the provisioning job.
- **Remove Station User** (component: **RemoveStationUserStep**) – removes a named user from the station.
- **Rename Device Station** (component: **RenameStationStep**) – renames the station on the remote host to match the name in the supervisor's **NiagaraNetwork**.
- **Run Robot** (component: **RunRobotStep**) – allows you to create custom code to run on each station.
- **Security Job Steps** is a convenience step that generates three separate sets of credentials in one place and in one sequence. The individual steps are **Set Platform Credentials**, **Set Station User Password** and **Set Platform User Password**.
- **Set Certificate Alias** (component: **SetCertificateAliasJobStep**) – sets the certificate alias used for the remote device's platform, web, and fox services.
- **Set Platform Credentials** (component: **SetPlatformCredentialsJobStep**) – adds a new platform user account and removes the default account on the remote device.
- **Set Platform User Password** (component: **SetPlatformUserPasswordJobStep**) – changes password for an existing platform user.
- **Set Property** (component: **SetPropertyJobStep**) – set or add a property.
- **Set Station Connection Credentials** (component: **SetStationConnectionCredentialsStep**) – sets the station credentials used to connect to the remote station.
- **Set Station User Password** (component: **SetStationUserPasswordJobStep**) – sets the station user password.
- **Set System Passphrase** (component: **SetSystemPassphraseJobStep**) - sets the system passphrase.
- **Set TLS Level** (component: **SetTlsLevelJobStep**) – sets the minimum TLS level for the remote device's platform, web, and fox services.
- **Set Time** (component: **SetTimeJobStep**) - sets the time on the remote device.
- **Setup Reciprocal Connection** (component: **SetupReciprocalConnectionStep**) – sets the connection settings for the "reciprocal" station connection, which is the station proxy in the **NiagaraNetwork** on the remote station that points back to the Supervisor.
- **Sign Certificate** (component: **SignCertificateStep**) signs a certificate (typically a server certificate) for each remote platform in the job.
- **Update Connections Using Niagara Network Discovery** (component: **NiagaraNetworkDiscoveryStep**) – updates station connections using **NiagaraNetwork** discovery.

- **Update Connections Using Provisioning Station's DHCP Server** (component: `DhcpDiscoveryStep`) – updates station connections using the provisioning station's DHCP server leases.
- **Upgrade Out-of-Date Software** (component: `SoftwareStationExt`) – installs software, upgrades out-of-date software, supports copy supervisor step, and the reboot step.
- **Update Template or Application Configuration** (component: `UpdateConfigurationStep`) updates the template configuration values of an installed template in each remote station.
- **Update Licenses** job step (component: `UpdateLicensesJobStep`) – updates software license in each remote host.
- **Upgrade Application Template** (`UpgradeApplicationStep`) upgrades the installed application template in each remote host.
- **Upgrade Out-of-Date Software** (`UpgradeOutOfDateStep`) compares module versions on the target hosts with the latest versions in the Supervisor's software database and installs updates as needed.
- **Upgrade Template** (component: `UpgradeTemplateStep`) - upgrades deployed component template instances on each station using and Excel file.

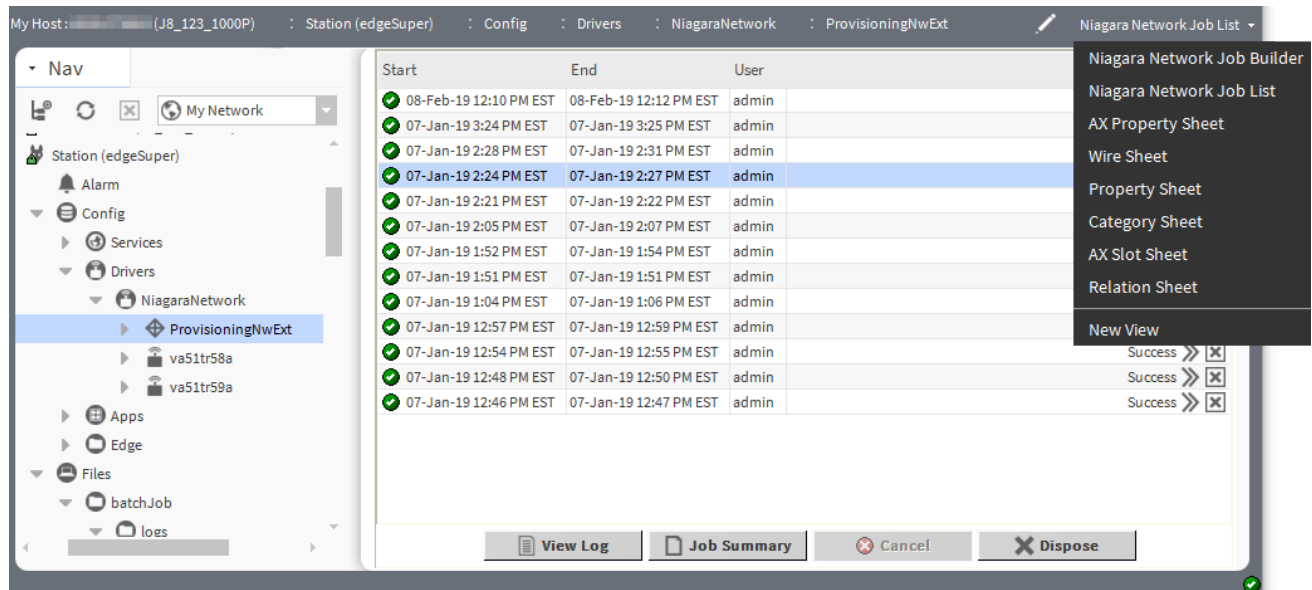
Toolbar for Niagara Network Job Builder

Toolbar Icon and Name	Description
 Undo	Reverses the previous command.
 Redo	Restores a command action after the Undo command has removed it.
 Multi-selection Mode	Enables you to individually select multiple points without holding down the ctrl key.
 Export	Exports the current view or object.

provisioningNiagara-NiagaraNetworkJobList

This view provides a supervisor-wide, table-based history of provisioning jobs and their results. It is an available view for the `ProvisioningNwExt` component of a `NiagaraNetwork`.

Figure 97 Niagara Network Job List is another view of the ProvisioningNwExt



Start	End	User
08-Feb-19 12:10 PM EST	08-Feb-19 12:12 PM EST	admin
07-Jan-19 3:24 PM EST	07-Jan-19 3:25 PM EST	admin
07-Jan-19 2:28 PM EST	07-Jan-19 2:31 PM EST	admin
07-Jan-19 2:24 PM EST	07-Jan-19 2:27 PM EST	admin
07-Jan-19 2:21 PM EST	07-Jan-19 2:22 PM EST	admin
07-Jan-19 2:05 PM EST	07-Jan-19 2:07 PM EST	admin
07-Jan-19 1:52 PM EST	07-Jan-19 1:54 PM EST	admin
07-Jan-19 1:51 PM EST	07-Jan-19 1:51 PM EST	admin
07-Jan-19 1:04 PM EST	07-Jan-19 1:06 PM EST	admin
07-Jan-19 12:57 PM EST	07-Jan-19 12:59 PM EST	admin
07-Jan-19 12:54 PM EST	07-Jan-19 12:55 PM EST	admin
07-Jan-19 12:48 PM EST	07-Jan-19 12:50 PM EST	admin
07-Jan-19 12:46 PM EST	07-Jan-19 12:47 PM EST	admin

To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and right-click **ProvisioningNwExt**→**Views**→**NiagaraJobList**.

This main area shows a Supervisor-wide list of provisioning jobs that have been sent to run, are running, or have completed. Note that any pending jobs do not appear until the linked trigger schedule actually fires.

Columns

Column	Value	Description
Start or Started	date and time (read-only)	Displays the date and time that the system submitted the job to the job queue.
End or Ended [job]	date and time	Displays the date and time when the job stopped running. This property is blank if the job is still running.
User	text	Defines the station user that requested the job. This column displays <code>unknown</code> if job was triggered by a linked schedule.
Status or State	read-only text:	<p>Displays the current or final state of the job. The first three states appear on the Device Network Job view.</p> <ul style="list-style-type: none"> <code>Unknown</code> —the job is pending execution. <code>Running</code> — the job is executing. <code>Canceling</code> — request to cancel the job was sent, but has not been processed yet, and the job is still executing. <code>Success</code> — job finished successfully, with all steps completed for all stations. <code>Canceled</code> — job was canceled before it completed, and is no longer running. <code>Failed</code> — at least one step failed in one station; job is no longer running. <p>Each row in the table ends with a details button (>>) and a dispose button (X) . Clicking this button changes the view to the Niagara Network Job view or the Batch Job Step Log File view, which shows all logged messages that are related to this single job.</p> <p>The overall status for the job in other stations, may be different).</p>

Buttons

The system enables most buttons near the bottom of the **Niagara Network Job List** when you select a job in the **Jobs Table**.

- **View Log** opens a popup **Job Log** window that displays the messages output by the selected job or step.
-
- **Cancel Job** clicking this button notifies the system to cancel the job when it is safe to do so. Not all job steps can be canceled.
- **Dispose**

HTML5 Niagara Network Job List

In EC-Net 4.14 and later, there is added browser support for the Lon Network Router Manager View. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view. This view provides a supervisor-wide, table-based history of provisioning jobs and their results. It is an available view for the **ProvisioningNwExt** component of a **NiagaraNetwork**.

Figure 98 Niagara Network Job List

Start	End	User	Status
02-Nov-23 1:43 PM IST	02-Nov-23 1:43 PM IST	admin	Success
02-Nov-23 1:43 PM IST	02-Nov-23 1:43 PM IST	admin	Failed
02-Nov-23 1:41 PM IST	02-Nov-23 1:42 PM IST	admin	Success
02-Nov-23 1:36 PM IST	02-Nov-23 1:36 PM IST	admin	Success
02-Nov-23 1:35 PM IST	02-Nov-23 1:35 PM IST	admin	Success
01-Nov-23 12:07 PM IST	01-Nov-23 12:07 PM IST	admin	Success

Buttons: View Log, Job Summary, Cancel Job, Dispose

To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and right-click **ProvisioningNwExt**→**Views**→**NiagaraJobList**.

This main area shows a Supervisor-wide list of provisioning jobs that have been sent to run, are running, or have completed. Note that any pending jobs do not appear until the linked trigger schedule actually fires.

Columns





Column Name	Description
Start	Displays the date and time that the system submitted the job to the job queue.
End	Displays the date and time when the job stopped running. This property is blank if the job is still running.
User	Defines the station user that requested the job. This column displays <code>unknown</code> if job was triggered by a linked schedule.
Status	<p>Displays the current or final state of the job. The first three states appear on the Device Network Job view.</p> <ul style="list-style-type: none"> Unknown —the job is pending execution. Running — the 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. <p>Each row in the table ends with a details button (>>) and a dispose button (X) . Clicking this button changes the view to the Niagara Network Job view or the Batch Job Step Log File view, which shows all logged messages that are related to this single job.</p> <p>The overall status for the job in other stations, may be different).</p>

Buttons

The system enables most buttons near the bottom of the **Niagara Network Job List** when you select a job in the **Jobs Table**.

- **View Log** opens a popup **Job Log** window that displays the messages output by the selected job or step.
- **Job Summary** changes to the **Batch Job Log File View** for the job that contains this step.
- **Cancel Job** clicking this button notifies the system to cancel the job when it is safe to do so. Not all job steps can be canceled.
- **Dispose**

Toolbar for Niagara Network Job List View

Toolbar Icon and Name	Description
 Undo	Reverses the previous command.
 Redo	Restores a command action after the Undo command has removed it.
 Multi-selection Mode	Enables you to individually select multiple points without holding down the ctrl key.
 Export	Exports the current view or object.

Niagara Network Job view

This view shows the details for the execution of a single provisioning job and is the default view for any **NiagaraNetworkJob** component.

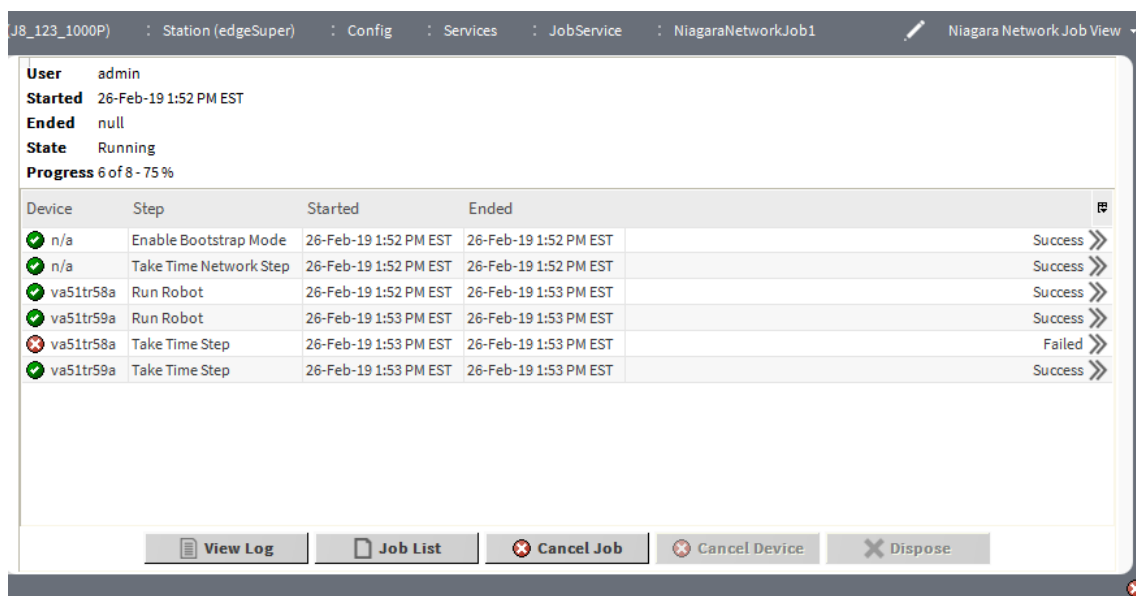
You can access this view several ways:

- By clicking **Run Now** in the **Niagara Network Job Builder** or **Niagara Network Prototype View**.
- By clicking the ">>" button in the **Niagara Network Job List** or **Prototype Job List** on any provisioning job that is still running.

If the job has completed and no longer appears in the **Job Service Manager** view, the system displays the **Batch Job Log File** view instead of this view. The **Batch Job Log File** functions the same as the **Niagara Network Job View**.

- By clicking **Job Summary** on the **Batch Job Step Log File View**.

Figure 99 Niagara Network Job View lists steps in one job



From top to bottom, this view has three areas:

- Job elements, which are read-only.
- A summary table of the steps in the job.
- A series of buttons at the bottom.

Job elements

Element	Value	Description
User	text	Defines the station user that requested the job. This column displays <code>unknown</code> if job was triggered by a linked schedule.
Start or Started	date and time (read-only)	Displays the date and time that the system submitted the job to the job queue.
End or Ended [job]	date and time	Displays the date and time when the job stopped running. This property is blank if the job is still running.
Status or State	read-only text:	<p>Displays the current or final state of the job. The first three states appear on the Device Network Job view.</p> <ul style="list-style-type: none"> • <code>Unknown</code> —the job is pending execution. • <code>Running</code> — the job is executing. • <code>Canceling</code> — request to cancel the job was sent, but has not been processed yet, and the job is still executing. • <code>Success</code> — job finished successfully, with all steps completed for all stations. • <code>Canceled</code> — job was canceled before it completed, and is no longer running. • <code>Failed</code> — at least one step failed in one station; job is no longer running. <p>Each row in the table ends with a details button (>>) and a dispose button (X) . Clicking this button changes the view to the Niagara Network Job view or the Batch Job Step Log File view, which shows all logged messages that are related to this single job.</p> <p>The overall status for the job in other stations, may be different).</p>

Columns

Column	Value	Description
Device	element	Identifies the station that is being processed or has been processed.
Job Step	element	Identifies the type of provisioning step: Backup Stations, File Copy, Install Software, etc.
Start or Started	date and time	Displays the date and time that the step began processing.

Column	Value	Description
End or Ended [step]	date and time	Displays the date and time when the step stopped running. This property is blank if the job is still running.
Status or State	read-only text	<p>Displays the current or final state of each step:</p> <ul style="list-style-type: none"> • Running — the step is executing. • Canceling — the request to cancel the step was sent, but has not been processed yet. The step is still running. • Success — the step finished successfully. • Canceled — the step was canceled before it completed and is no longer running. • Failed — the step did not complete <p>Each row in the table ends with a details button (>>) and a dispose button (X) . This button functions the same as the Step Details button at the bottom of the view.</p> <p>The overall status for the step in other stations, may be different).</p>

Buttons

- **View Log** opens a popup **Job Log** window that displays the messages output by the selected job or step.
-
- **Cancel Job** clicking this button notifies the system to cancel the job when it is safe to do so. Not all job steps can be canceled.
- **Cancel Device** for the selected station only, clicking this button notifies the system to cancel the job when it is safe to do so. The system begins processing the job for the next station.
- **Dispose** Clicking this button prompts you to confirm that you want to delete the selected job(s). The deletion includes all associated job files. If you confirm the deletion.

HTML5 Niagara Network Job view

In EC-Net 4.14 and later, there is added browser support for the Lon Network Router Manager View. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view. This view shows the details for the execution of a single provisioning job and is the default view for any **NiagaraNetworkJob** component.

You can access this view several ways:

- By clicking **Run Now** in the **Niagara Network Job Builder** or **Niagara Network Prototype View**.
- By clicking the “>>” button in the **Niagara Network Job List** or **Prototype Job List** on any provisioning job that is still running.

If the job has completed and no longer appears in the **Job Service Manager** view, the system displays the **Batch Job Log File** view instead of this view. The **Batch Job Log File** functions the same as the **Niagara Network Job View**.

- By clicking **Job Summary** on the **Batch Job Step Log File View**.

Figure 100 Batch Job Log File View

The screenshot shows the 'Batch Job Log File View' interface. At the top, there's a header bar with 'bajaux Widget' and a dropdown menu. Below the header, job details are listed: User (admin), Started (07-Nov-23 10:17 AM IST), Ended (07-Nov-23 10:17 AM IST), State (Success), and Progress (1 of 1 - 100 %). A table follows with columns: Device, Step, Started, Ended, and a details button. The table contains one row: Device (n/a), Step (Enable Bootstrap Mode), Started (07-Nov-23 10:17 AM IST), Ended (07-Nov-23 10:17 AM IST), and State (Success). At the bottom, there are five buttons: View Log, Job List, Cancel Job, Cancel Device, and Dispose.

From top to bottom, this view has three areas:

- Job elements, which are read-only.
- A summary table of the steps in the job.
- A series of buttons at the bottom.

Job elements

Element	Description
User	Defines the station user that requested the job. This column displays <code>unknown</code> if job was triggered by a linked schedule.
Started	Displays the date and time that the system submitted the job to the job queue.
Ended	Displays the date and time when the job stopped running. This property is blank if the job is still running.
State	<p>Displays the current or final state of the job. The first three states appear on the Device Network Job view.</p> <ul style="list-style-type: none"> • <code>Unknown</code> —the job is pending execution. • <code>Running</code> — the job is executing. • <code>Canceling</code> — request to cancel the job was sent, but has not been processed yet, and the job is still executing. • <code>Success</code> — job finished successfully, with all steps completed for all stations. • <code>Canceled</code> — job was canceled before it completed, and is no longer running. • <code>Failed</code> — at least one step failed in one station; job is no longer running. <p>Each row in the table ends with a details button (>>) and a dispose button (X) . Clicking this button changes the view to the Niagara Network Job view or the Batch Job Step Log File view, which shows all logged messages that are related to this single job.</p> <p>The overall status for the job in other stations, may be different).</p>





Columns

Column	Description
Device	Identifies the station that is being processed or has been processed.
Job Step	Identifies the type of provisioning step: Backup Stations, File Copy, Install Software, etc.
Started	Displays the date and time that the step began processing.
Ended	Displays the date and time when the step stopped running. This property is blank if the job is still running.
State	<p>Displays the current or final state of each step:</p> <ul style="list-style-type: none"> Running — the step is executing. Canceling — the request to cancel the step was sent, but has not been processed yet. The step is still running. Success — the step finished successfully. Canceled — the step was canceled before it completed and is no longer running. Failed — the step did not complete <p>Each row in the table ends with a details button (>>) and a dispose button (X) . This button functions the same as the Step Details button at the bottom of the view.</p> <p>The overall status for the step in other stations, may be different).</p>

Buttons

- **View Log** opens a popup **Job Log** window that displays the messages output by the selected job or step.
- **Cancel Job** clicking this button notifies the system to cancel the job when it is safe to do so. Not all job steps can be canceled.
- **Cancel Device** for the selected station only, clicking this button notifies the system to cancel the job when it is safe to do so. The system begins processing the job for the next station.
- **Dispose** Clicking this button prompts you to confirm that you want to delete the selected job(s). The deletion includes all associated job files. If you confirm the deletion.

Toolbar for Niagara Network Job Builder

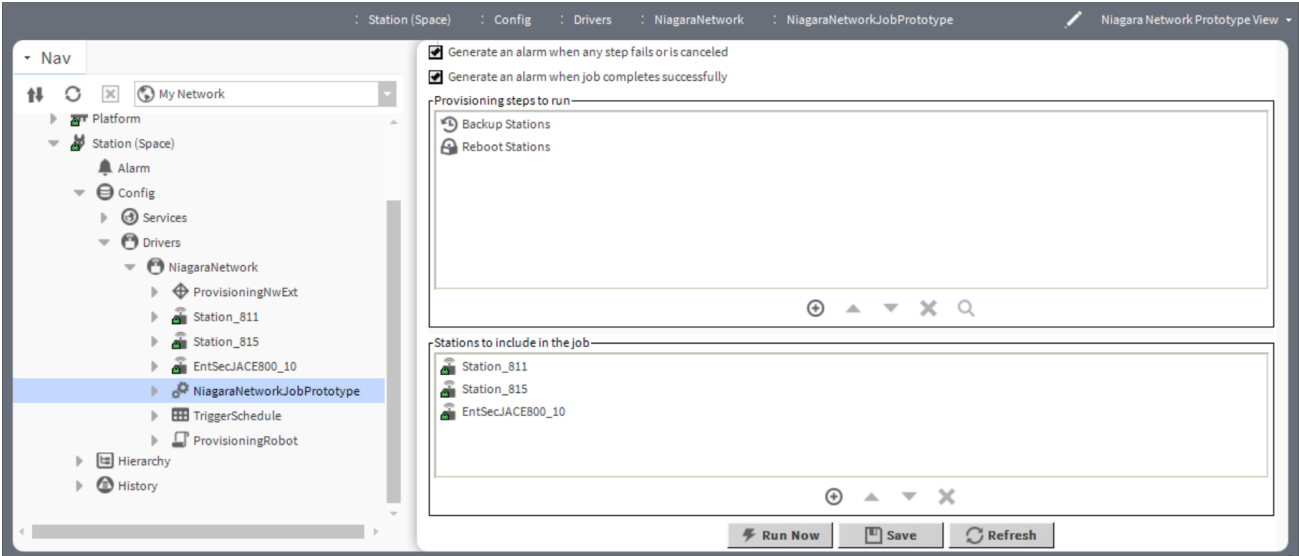
Toolbar Icon and Name	Description
 Undo	Reverses the previous command.
 Redo	Restores a command action after the Undo command has removed it.
 Multi-selection Mode	Enables you to individually select multiple points without holding down the ctrl key.
 Export	Exports the current view or object.

provisioningNiagara-NiagaraNetworkPrototypeView

The **Niagara Network Prototype View**, which is nearly identical to the **ProvisioningNwExt's Niagara Network Job Builder**, is the default view on each reusable job prototype component (**NiagaraNetworkJobPrototype**).

You use this view to specify and edit a specific provisioning job.

Figure 101 Niagara Network Prototype View example



To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and double-click the **NiagaraNetworkJob-Prototype** , or right-click **NiagaraNetworkJobPrototype**→**Views**→**Niagara Network Prototype View**

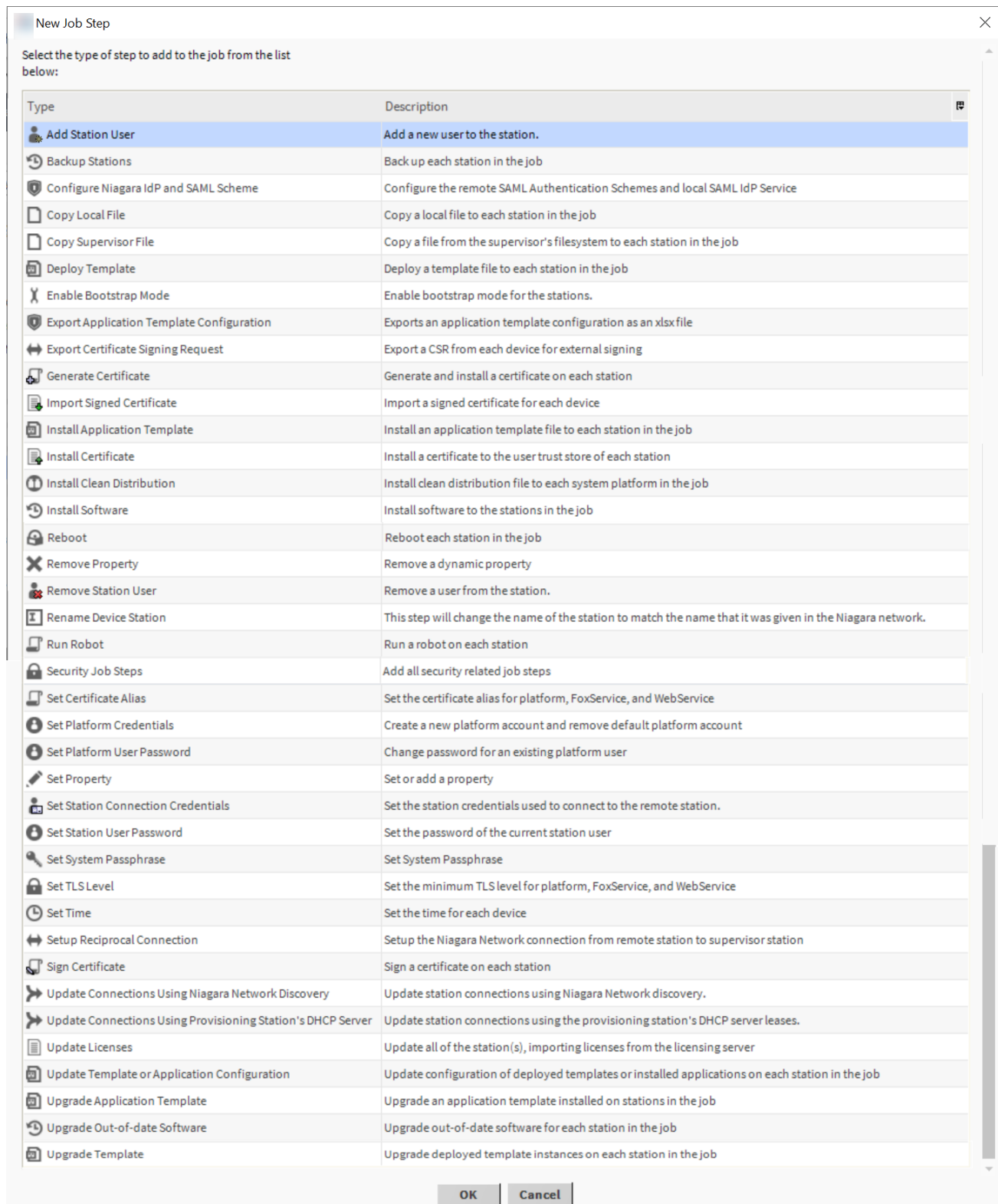
As on the **Niagara Network Job Builder** view, this view has alarm check boxes, three list areas with controls, and buttons at the bottom of the view.

Alarm check boxes

Alarm check box	Value	Description
Generate an alarm when any step fails or is canceled	check box	Determines if the BatchJobService is to issue alarms for this provisioning job, and under what circumstances. Alarms use the alarm class specified in the BatchJobService and appear in the alarm console as alerts. When selected, the BatchJobService raises an alarm whenever a job step fails or is canceled.
Generate an alarm when job completes successfully	check box	Configures the BatchJobService too raise an alarm when a job completes with no step failures.

Steps and stations

List area	Value	Description
Provisioning steps to run	text	Provides 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.
Stations to include in the job	text	Lists all the stations to be processed by the job. This means each station processes all steps in the (middle pane) Steps for Each Station List . Only stations in the Supervisor's Niagara-Network 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).






Figure 102 New Job Step windows

The upper left window opens when you click the plus (+) in the **Provisioning steps to run** pane. The windows provide a brief description for each step. Additional details on each step are available in the "Provisioning steps" section of Chapter 1, as well as in the following table.

Step	Use this job step to
Enable Bootstrap Mode	<p>Enable the bootstrap mode for Supervisor-to-remote device connections. It creates a one-time provisioning job to set the Supervisor-to-device Platform and Fox connections to bypass any certificate checks during the execution of the provisioning job. This step is intended for use with other provisioning steps that set credentials and configure certificates.</p> <p>NOTE: Running this step by itself makes no persistent change on the target stations.</p>
Update Connections Using Niagara Network Discovery	Update station connections using the NiagaraNetwork Discovery function.
Update Connections Using Provisioning Station's DHCP Server	Update the station connections using the provisioning station's DHCP server leases.
Update Licenses	<p>Update the licenses on all remote hosts included in the job.</p> <p>When processed by the Supervisor, it gathers information about the licenses installed on each target host, then accesses the online licensing server (in one message) to see if the licenses are up-to-date. If a host's license is out-of-date, the step updates each licence in the target station's host, and in the Supervisor's local license database.</p> <p>NOTE: If the Supervisor is not configured for Internet connectivity, only its local license database is used to compare against licenses installed in the target host(s). If a host's license is out-of-date, the step updates the license in the target host(s).</p> <p>The Brand of a license can also be changed. Existing license files will be deleted and the new license will be copied to the platform. License vendor certificates may also be copied to the target platform if needed.</p>
Backup Stations	<p>Make an online station backup if the station is running, or an offline backup if it is idle.</p> <p>NOTE: Configuring a regularly-scheduled station backup by adding a provisioning job prototype (using the Niagara Network Prototype View) is a better practice than setting up a one-time station backup (using the Niagara Network Job Builder).</p>
Copy Local File	<p>Copy a single file from your (local) PC to a given location on each remote station in the job. You access this step in the New Job Step menu when adding a step in the Niagara Network Job Builder view.</p> <p>This is not available in the Niagara Network Prototype View.</p>
Copy Supervisor File	Copy a single file from your Supervisor station's filesystem to a location in each remote station in the job.
Deploy Template	Deploy a component or device template file to each remote station in the job. Each template instance can have custom configurations, inputs, outputs, and relations using the Bulk Deploy Excel spreadsheet.
Export Certificate Signing Request	Export a Certificate Signing Request (CSR) from each remote host for external signing. CSRs are part of the certificate setup for establishing trusted secure communications between a supervisor and remote devices.
Generate Certificate	Generate and installs a certificate into the User Trust Store on each remote station. This is part of the certificate setup when the Supervisor will be used to sign certificates.
Import Signed Certificate	Import a signed certificate for each remote device Use this to import certificates signed externally (initiated from a CSR) into devices containing the original certificate with the matching private key.
Install Application Template	Install an application template file to each remote station in the job. Application templates are useful for establishing a complete station in a single deployable container. Provisioning application templates enables installation across a set of stations. Each installation can have custom configurations applied to it.
Install Certificate	Install a certificate to the User Trust Store of each remote station in the job so that it can connect to the Supervisor. This is a step you would perform when you initially bring up your NiagaraNetwork , or after a certificate expires and a new one needs to be installed in each station.
Install Software	<p>Install software to each remote station in the job. This step installs new modules not already running on the target device.</p> <p>NOTE: All software MUST be installed on the device to support all dependencies required by the deployed station services, networks, templates, and any other components.</p>
Reboot	Restart each remote station in the job. Typically this step is not explicitly needed, but can be used to ensure a system restart after changes have been made to remote devices.

Step	Use this job step to
Rename Device Station	Rename each device station to match station proxy name.
Run Robot	Passes the specified ProvisioningRobot (custom-created program code) located in the Supervisor station to the ProgramService on each remote station in the job. The remote station's ProgramService runs the custom code in the station. For more details, see "Run Robot step" in Chapter 1.
Security Job Steps	Add security-related job steps This is a convenience step that adds the Platform Credentials, Station User Password, and System Passphrase steps all at once.
Set Certificate Alias	Set the certificate alias for the Platform, Fox, and Web services This is typically done after signed certificates have been installed or configured in the User Trust Store on each remote device.
Set Platform Credentials	Assign a user name and password for the Platform connection to each remote device in the job. This is used to replace the default Platform credentials.
Set Property	Sets or adds a property to an existing component for each station in the job. You can define the component using a slot path ORD or a NEQL query. If necessary, you can set a unique value for each station in the job.
Set Station User Password	Set the password of the current station user for each remote station in the job. This is typically used to replace the default Fox user credentials for new devices, but can be used to replace any user password for the user associated with the current Fox connection.
Set System Passphrase	Set the system passphrase for each remote station in the job. This is used to replace the default system passphrase for new devices.
Set TLS Level	Set the TLS level for the Platform, Fox, and Web services. The TLS level can be customized across a set of remote devices.
Set Time	Set the time for each device in the job. System time for a set of remote devices can be updated to match the Supervisor, or a specific time can be pushed down.
Setup Reciprocal Connection	Set up the NiagaraNetwork connection from each remote station to the Supervisor. Certain station functionality (alarm reporting, etc.) requires a Fox connection from the remote station to the Supervisor.
Sign Certificate	Sign a certificate on each remote station in the job. This is done as part of the certificate setup, where the Supervisor is acting as the signing authority.
Upgrade Out-of-date Software	Upgrade out-of-date software on each remote station in the job. This step 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. If an older version is found on the station, it installs the latest software there. Common software versions across remote devices reduces the risk of errors caused by version incompatibility.
Upgrade Templates	Upgrade deployed template instances on each remote station in the job. For more details, see "Upgrade Template step" in Chapter 1.

Controls

Control	Description
	Adds a step.
	Moves the selected step up in the sequence of steps.
	Moves the selected step down in the sequence of steps.
	Removes the selected step.
	View and edit the details of the selected step.

Buttons

- **Run Now** dispatches the job to the batch job queue for immediate execution. The view automatically changes to the **Niagara Network Job** view. This button is enabled when there is at least one job step in either the **Initial Steps To Run Once List** or **Steps for Each Station List**, and one station in the **Stations List**.

- **Refresh** removes all entries from the three lists.
- **Save** saves changes to the corresponding job prototype component.

HTML5 Niagara Network Prototype View

In EC-Net 4.14 and later, there is added browser support for the Lon Network Router Manager View. The “HTML5” version of this view is a web-browser-based implementation and it provides the same functions as the EC-Net 4 Pro view. The **Niagara Network Prototype View**, which is nearly identical to the **ProvisioningN-wExt’s Niagara Network Job Builder**, is the default view on each reusable job prototype component (**NiagaraNetworkJobPrototype**).

Figure 103 Niagara Network Prototype View

To access this view, expand **Config**→**Drivers**→**NiagaraNetwork** and click the **NiagaraNetworkJobPrototype**, or right-click **NiagaraNetworkJobPrototype**→**Views**→**Niagara Network Prototype View**.

As on the **Niagara Network Job Builder** view, this view has alarm check boxes, three list areas with controls, and buttons at the bottom of the view.






Alarm check boxes

Alarm check box	Description
Generate an alarm when any step fails or is canceled	Determines if the BatchJobService is to issue alarms for this provisioning job, and under what circumstances. Alarms use the alarm class specified in the BatchJobService and appear in the alarm console as alerts. When selected, the BatchJobService raises an alarm whenever a job step fails or is canceled.
Generate an alarm when job completes successfully	Configures the BatchJobService too raise an alarm when a job completes with no step failures.

Steps and Stations

List Area	Description
Provisioning steps to run	Provides 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. For specific steps, refer to . Steps, page 187
Stations to include in the job	Lists all the stations to be processed by the job. 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).

Controls

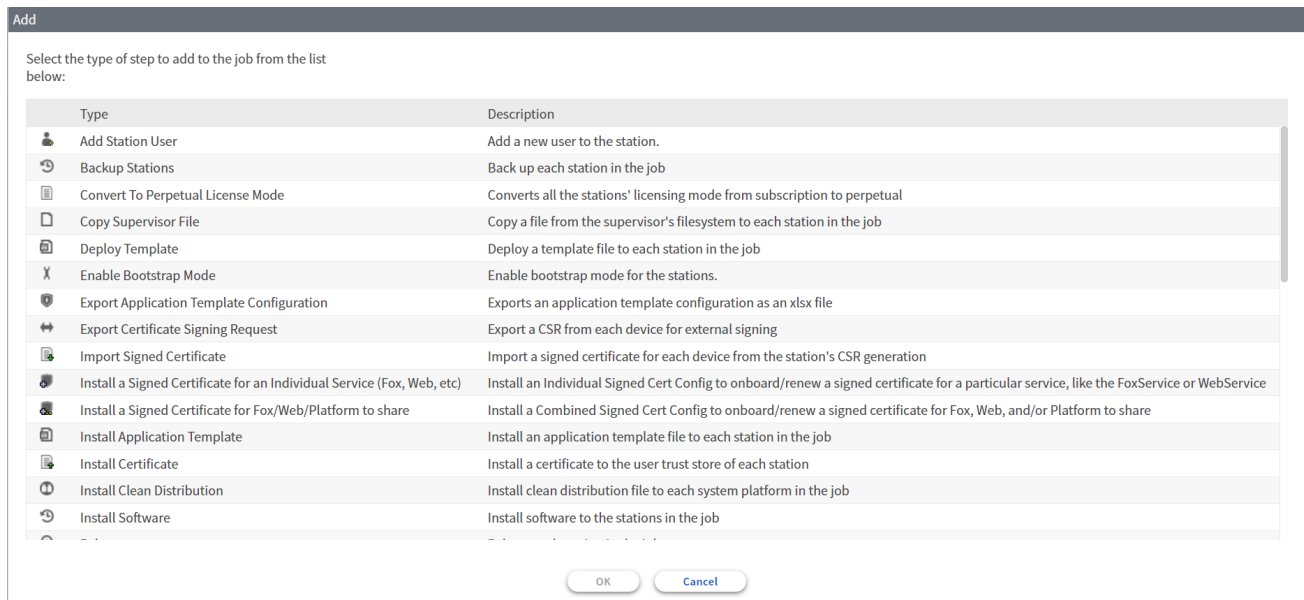
Control	Description
	Adds a step.
	Moves the selected step up in the sequence of steps.
	Moves the selected step down in the sequence of steps.
	Removes the selected step.
	View and edit the details of the selected step.

Buttons

- **Run Now** dispatches the job to the batch job queue for immediate execution. The view automatically changes to the **Niagara Network Job** view. This button is enabled when there is at least one job step in either the **Initial Steps To Run Once List** or **Steps for Each Station List**, and one station in the **Stations List**.
- **Refresh** removes all entries from the three lists.

Steps

Figure 104 The Add Steps Window








The screen captures shows all provisioning job steps to run for each station. This list includes all steps:

- **Add Station User** (component: **AddStationUserStep**) – adds a new user to the station.
- **Backup Stations** (component: **BackupStationExt**) – makes an online backup of each running station.
- **Convert To Perpetual Licence Mode** — Converts all the stations' licensing mode from subscription to perpetual
- **Copy Supervisor File** (component: **FileCopyStep**) – makes a copy of a Supervisor file.
- **Deploy Template** (component: **DeployTemplateStep**) – deploys a component template to each station using and Excel file.
- **Enable Bootstrap Mode** (component: **EnableBootstrapStep**) – creates a one-time provisioning job to connect from the Supervisor to the remote controller using a certificate exemption for the default self-signed certificate. This step is intended for use with other provisioning steps that set up credentials and configure certificates. It is useful for provisioning new out-of-the-box devices.
- **Export Application Template Configuration** — Exports an application template configuration as an xlsx file
- **Export Certificate Signing Request** (component: **ExportCsrJobStep**) - generates and exports a CSR for each remote host.
- **Import Signed Certificate** (component: **ImportSignedCertificateJobStep**) – imports a signed certificate for each remote host.
- **Install a Signed Certificate for an Individual Service (Fox, Web, etc)** – Install an Individual Signed Cert Config to onboard/renew a signed certificate for a particular service, like the **FoxService** or **WebService**.
- **Install a Signed Certificate for Fox/Web/Platform to share** – Install a Combined Signed Cert Config to onboard/renew a signed certificate for Fox, Web, and/or Platform to share.
- **Install Application Template** (component: **DeployApplicationStep**) – installs an application template on each station using an Excel file.
- **Install Certificate** (component: **InstallCertificateJobStep**) – installs a certificate from the supervisor's user trust store to each remote host's user trust store.

- **Install Clean Distribution** (component: **InstallCleanDistStep**) – installs the selected clean distribution file to the target platforms.
- **Install Software** (component: **InstallBySpecStep**) – installs a software module in one or more remote hosts.
- **Reboot** (component: **RebootJobStep**) – reboots the platform host.
- **Remove Platform User** Remove a user from the platform.
- **Remove Property** (component: **RemovePropertyStep**) – removes a property from a component at a defined slot path for each station in the provisioning job.
- **Remove Station User** (component: **RemoveStationUserStep**) – removes a named user from the station.
- **Rename Device Station** (component: **RenameStationStep**) – renames the station on the remote host to match the name in the supervisor's **NiagaraNetwork**.
- **Run Robot** (component: **RunRobotStep**) – allows you to create custom code to run on each station.
- **Security Job Steps** is a convenience step that generates three separate sets of credentials in one place and in one sequence. The individual steps are **Set Platform Credentials**, **Set Station User Password** and **Set Platform User Password**.
- **Set Certificate Alias** (component: **SetCertificateAliasJobStep**) – sets the certificate alias used for the remote device's platform, web, and fox services.
- **Set Platform Credentials** (component: **SetPlatformCredentialsJobStep**) – adds a new platform user account and removes the default account on the remote device.
- **Set Platform User Password** (component: **SetPlatformUserPasswordJobStep**) – changes password for an existing platform user.
- **Set Property** (component: **SetPropertyJobStep**) – set or add a property.
- **Set Station Connection Credentials** (component: **SetStationConnectionCredentialsStep**) – sets the station credentials used to connect to the remote station.
- **Set Station User Password** (component: **SetStationUserPasswordJobStep**) – sets the station user password.
- **Set System Passphrase** (component: **SetSystemPassphraseJobStep**) - sets the system passphrase.
- **Set TLS Level** (component: **SetTlsLevelJobStep**) – sets the minimum TLS level for the remote device's platform, web, and fox services.
- **Set Time** (component: **SetTimeJobStep**) - sets the time on the remote device.
- **Setup Reciprocal Connection** (component: **SetupReciprocalConnectionStep**) – sets the connection settings for the "reciprocal" station connection, which is the station proxy in the **NiagaraNetwork** on the remote station that points back to the Supervisor.
- **Sign Certificate** (component: **SignCertificateStep**) signs a certificate (typically a server certificate) for each remote platform in the job.
- **Update Connections Using Niagara Network Discovery** (component: **NiagaraNetworkDiscoveryStep**) – updates station connections using **NiagaraNetwork** discovery.
- **Update Connections Using Provisioning Station's DHCP Server** (component: **DhcpDiscoveryStep**) – updates station connections using the provisioning station's DHCP server leases.
- **Upgrade Out-of-Date Software** (component: **SoftwareStationExt**) – installs software, upgrades out-of-date software, supports copy supervisor step, and the reboot step.
- **Update Template or Application Configuration** (component: **UpdateConfigurationStep**) updates the template configuration values of an installed template in each remote station.
- **Update Licenses** job step (component: **UpdateLicensesJobStep**) – updates software license in each remote host.

- **Upgrade Application Template (UpgradeApplicationStep)** upgrades the installed application template in each remote host.
- **Upgrade Out-of-Date Software (UpgradeOutOfDateStep)** compares module versions on the target hosts with the latest versions in the Supervisor's software database and installs updates as needed.
- **Upgrade Template** (component: **UpgradeTemplateStep**) - upgrades deployed component template instances on each station using and Excel file.

Toolbar for Niagara Network Prototype

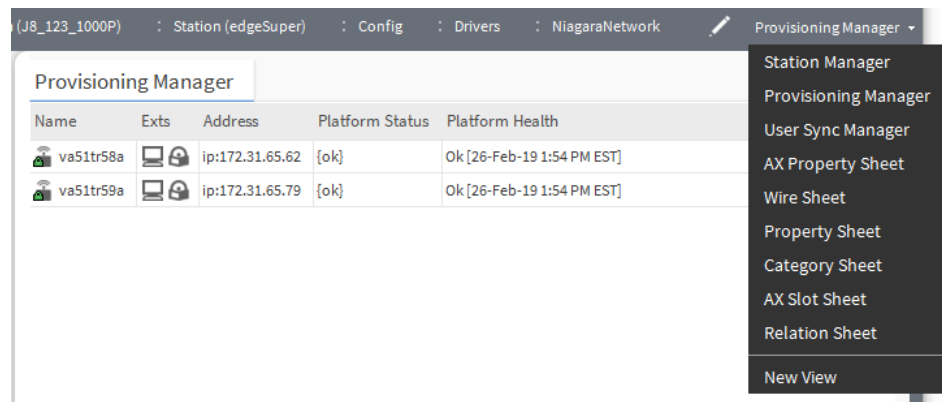
Toolbar Icon and Name	Description
 Save	Saves the changes.
 Undo	Reverses the previous command.
 Redo	Restores a command action after the Undo command has removed it.
 Multi-selection Mode	Enables you to individually select multiple points without holding down the ctrl key.
 Export	Exports the current view or object.

provisioningNiagara-ProvisioningManager

This view provides a central look at the status and health of platform connectivity to the various remote hosts, as well as quick access to some of the provisioning (device) extensions under each **NiagaraStation**. It is an available view on the Supervisor's **NiagaraNetwork**, provided that the Supervisor is licensed for provisioning and has the **ProvisioningNwExt** installed.

The view is based on a table, where each row represents a **NiagaraStation** component (similar to the network's default **Station Manager** view).

Figure 105 Provisioning Manager view of the Supervisor's **NiagaraNetwork**



Columns

Column	Value	Description
Name	text	The identifying name of the station.
Exts	n/a	Provides double-click access to two provisioning device extensions views: <ul style="list-style-type: none"> • Provisioning Station Director for StationProxy • Station Software Manager for Software

Column	Value	Description
		To access the other provisioning extensions, expand a NiagaraStation in the Nav tree or use the property sheet of a NiagaraStation .
Address	IP format	The IP address of the station.
Platform Status		Displays the current condition of the provisioning extension.
Platform Health		This information is updated by the ongoing ping monitor to the platform daemon. For any station (row), you can also right-click it and manually issue a Ping action.

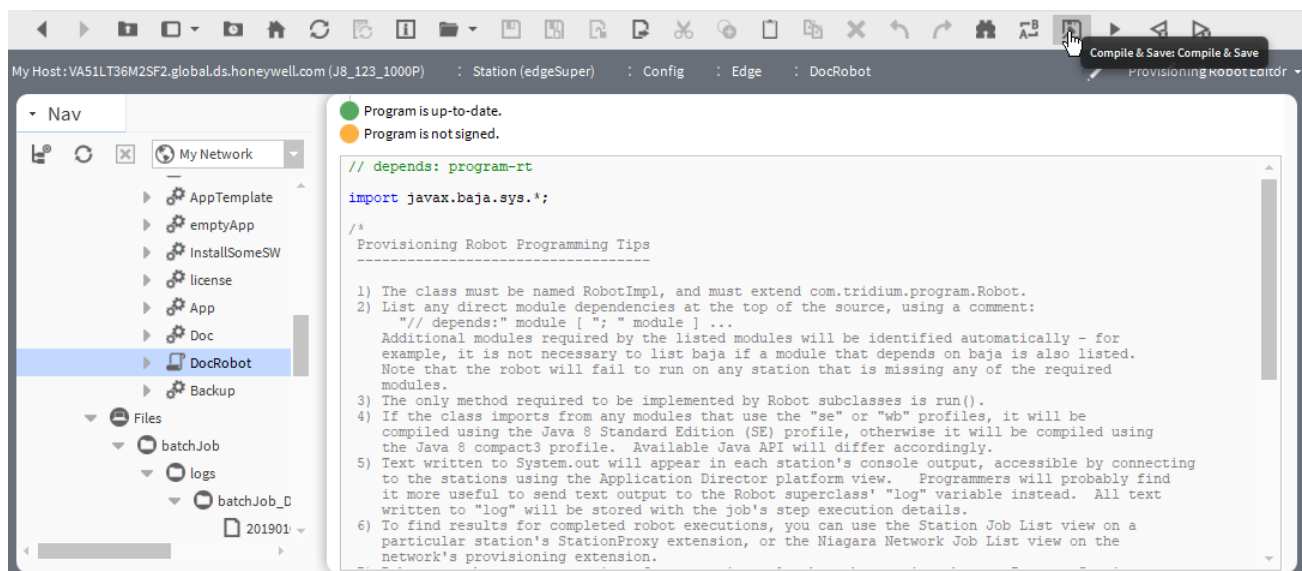
Buttons

Button	Value	Description
Edit	enabled when one or more rows is selected.	Provides a way to modify the platform connection credentials and port. These credentials provide provisioning access to the Supervisor station. These credentials also may be modified using each NiagaraStation 's Platform Connection (device extension) property sheet.

provisioningNiagara-ProvisioningRobotEditor

This view provides a program editing window that 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 a **ProvisioningRobot**. This view is the main view of the **ProvisioningRobot** component.

Figure 106 Provisioning Robot Editor is default view for a ProvisioningRobot



This view operates like the **Robot Editor** view of a station's **ProgramService**.

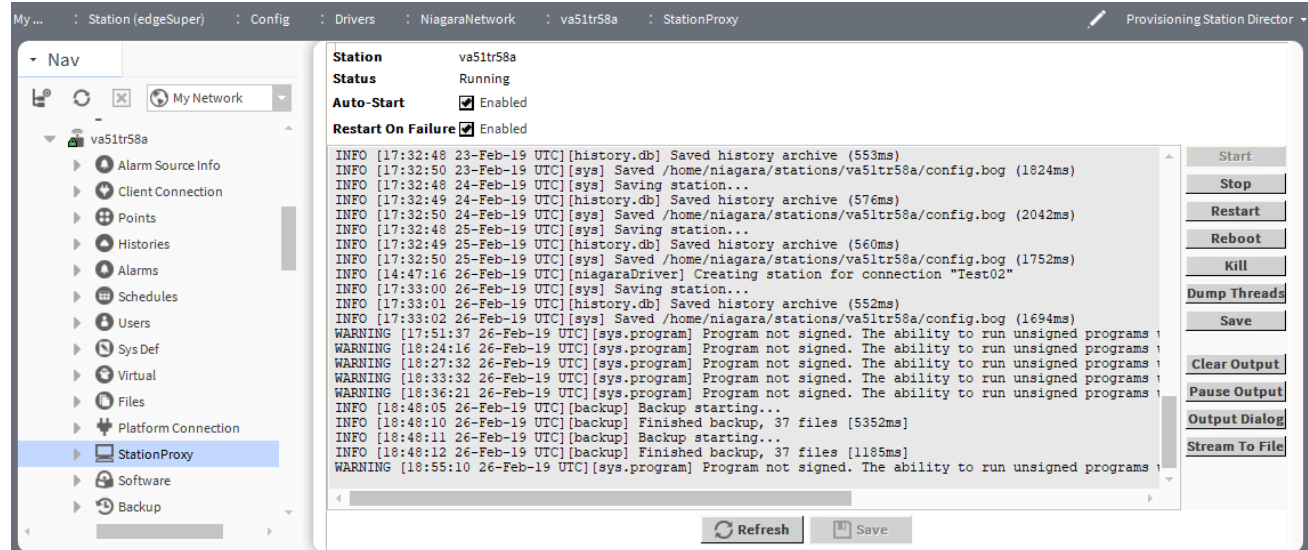
Button	Icon	Description
Compile and save		Compiles and saves the code.
Run now		Opens the Choose Stations window.

provisioningNiagara-ProvisioningStationDirector

This view is the default view on the **StationProxy** provisioning extension of a **NiagaraStation**. This view closely resembles the **Station Director** view in a direct platform connection.

This view closely resembles the **Application Director** view available in a direct platform connection to a host.

Figure 107 Provisioning Station Director is default view on StationProxy extension

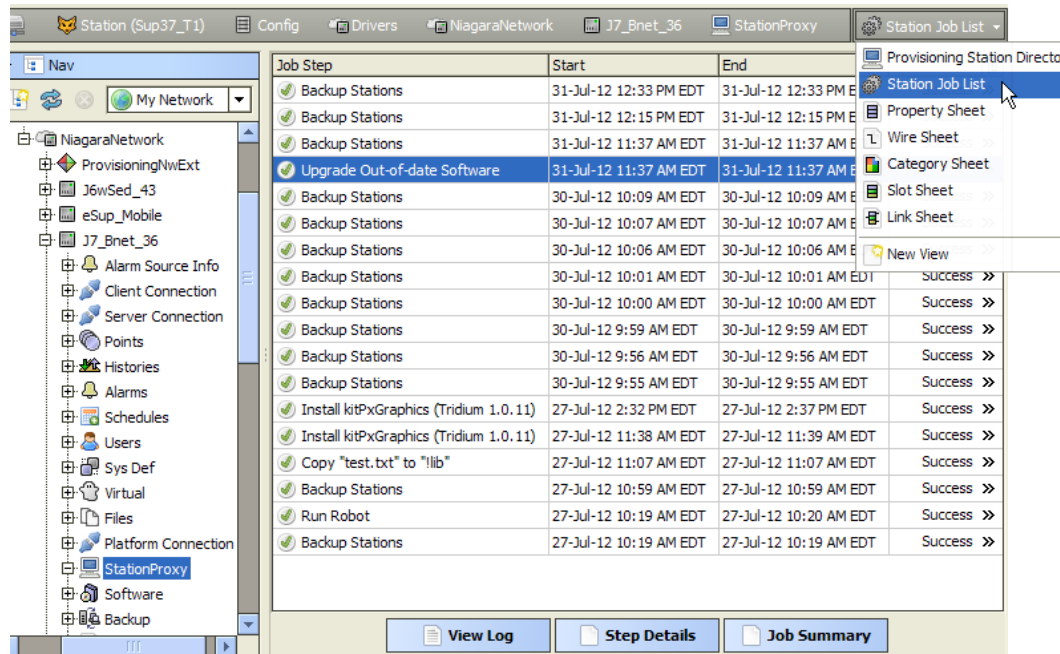


Refer to the “Application Director” section in the *Drivers Guide* for descriptions of most elements in this view. Only elements that differ from that view are explained here.

- 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 EC-Net 4 Pro 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).

provisioningNiagara-StationJobList

This view summarizes provisioning job steps that have been executed against this particular station, with additional details available. It is an additional view of the **StationProxy** extension under a **NiagaraStation** device. You access this view using the view selector or by right-clicking the **StationProxy** extension and selecting **Views**→ **Station Job List** or by selecting it from the extension’s view selector.

Figure 108 Station Job List is available view on StationProxy extension

Step table

This main area of the **Station Job List** view shows a row for each step that has been executed against the station. 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.

Because of this, the **Station Job List** is not the appropriate view to use to find the answer for questions like, "Why did the backup scheduled for Tuesday on this station not run?" For this type of information, 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:

- To view any step's **Step Log File View**, which is the same as using the **Step Details** button at the bottom of the view, click the ">>" (Details) button to the right of the status
- To view the **Job Log** for any job, double-click any step row. The Job Log lists a series of messages about the step. This is the same as using the **View Log** button at the bottom of the view.
- Right-click a step for a popup menu, providing the same functions as those provided by the buttons at bottom of view.

Column	Value	Description
Job Step	element	Identifies the type of provisioning step: Backup Stations, File Copy, Install Software, etc.
Start or Started	date and time (read-only)	Displays the date and time that the system submitted the job to the job queue.

Column	Value	Description
End or Ended [step]	date and time	Displays the date and time when the step stopped running. This property is blank if the job is still running.
Status or State	read-only text:	<p>Displays the current or final state of the job. The first three states appear on the Device Network Job view.</p> <ul style="list-style-type: none"> Unknown —the job is pending execution. Running — the 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. <p>Each row in the table ends with a details button (>>) and a dispose button (X) . Clicking this button changes the view to the Niagara Network Job view or the Batch Job Step Log File view, which shows all logged messages that are related to this single job.</p> <p>The overall status for the job in other stations, may be different).</p>

Buttons

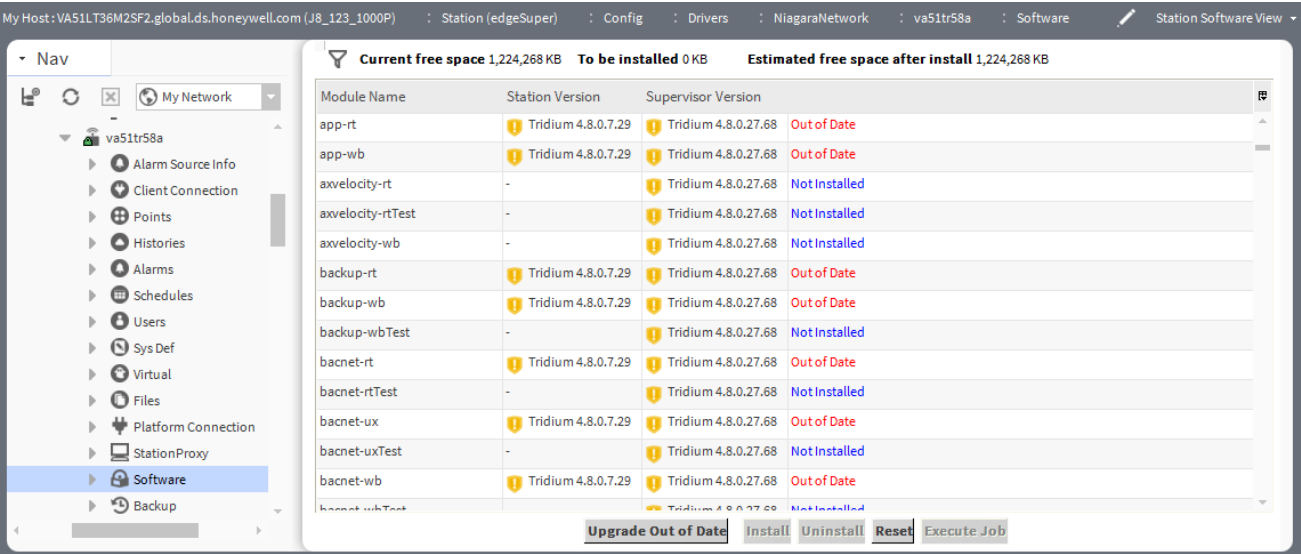
Button	Value	Description
View Log	button enabled when a job row is selected	Opens a popup Job Log window that displays the messages output by the selected job or step.
Step Detail	button always available	Switches the view to the Batch Job Step Log File View .
Job Summary	button always enabled	Changes to the Batch Job Log File View for the job that contains this step.

provisioningNiagara-StationSoftwareView

This displays the current state of the station's software modules. It is the default view for the **Software** provisioning extension.

The **Station Software Manager** closely resembles the **Software Manager** that is available with a direct platform connection to a host. For more information about this view, refer to the Software Manager section in the *EC-Net 4 Platform Guide*. Only elements that differ from that view are explained here.

Figure 109 Station Software View is default view on Software provisioning extension



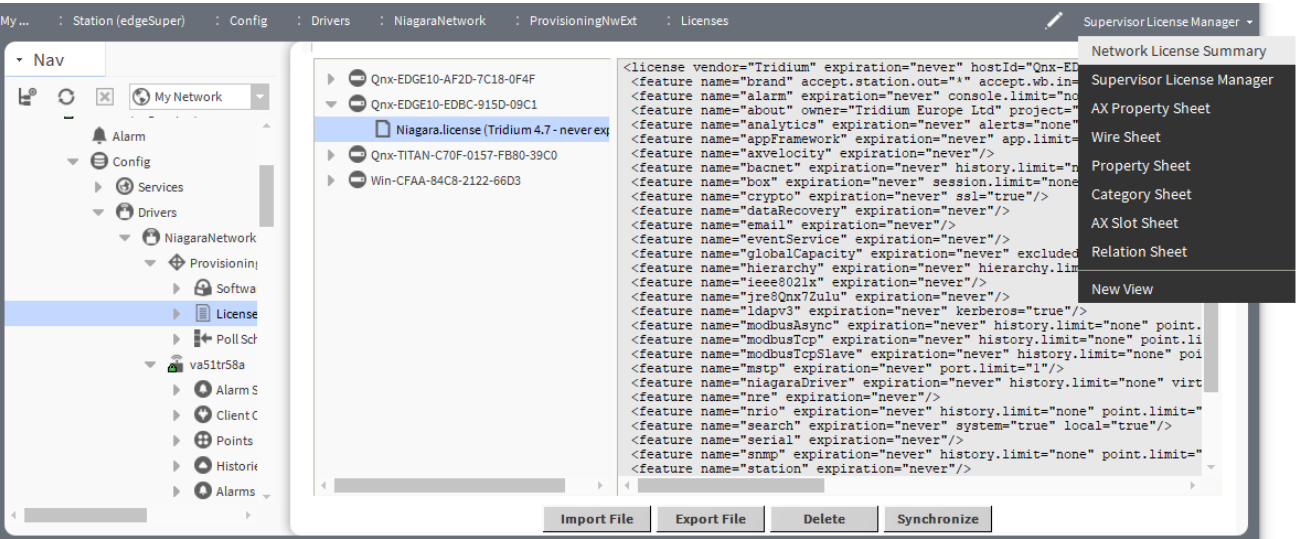
When you access this view, the system takes a snapshot of the station’s current software configuration and displays it as a table. Other differences from the **Software Manager** view in a direct platform connection are summarized as follows:

- The **Software Manager** provides columns for Installed Version and Available Version. These identify the version of each module installed on the station and available in EC-Net 4 Pro. The **Station Software View** has equivalent columns labeled Station Version and Supervisor Version.
- Instead of a **Commit** button that starts the software installation by running it in EC-Net 4 Pro, the **Station Software View** has an **Execute Job** button. To submit the installation as a provisioning batch job in the Supervisor station, you click **Execute Job**, which opens the **Niagara Network Job View**.

provisioningNiagara-SupervisorLicenseManager

This view provides management access to the Supervisor’s local license database, which is located under its !licenses/db subdirectory. The **Supervisor License Manager** is an available view on the **Licenses** slot of the **ProvisioningNwExt** under the **NiagaraNetwork**. To access this view, double-click the **Licenses** slot.

Figure 110 Supervisor License Manager view is available view on ProvisioningNwExt’s License slot



The Supervisor's local license database is the structured organization of "host ID-named" sub-folders under the Supervisor's `!/licenses` folder that contain license files. As in the equivalent EC-Net 4 Pro **License Manager** view, this view provides a two-pane window into all the license files and parent host ID folders, where the:

- 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.

Button	Value	Description
Import File	always enabled	Adds license file(s) from a local license file or license archive (.lar) file.
Export File	always enabled	Allows you to save all licenses (or any selected licenses) locally as a license archive file.
Delete	enabled when a row is selected	Allows you to remove licenses from the Supervisor's local license database.
Synchronize	enabled if you have Internet connectivity	Allows you to update all licenses (or any selected licenses) in the Supervisor's local license database with the most current versions on the online licensing server. For more details and related information, refer to the <i>EC-Net 4 Platform Guide</i> .

provisioningNiagara-SupervisorSoftwareManager

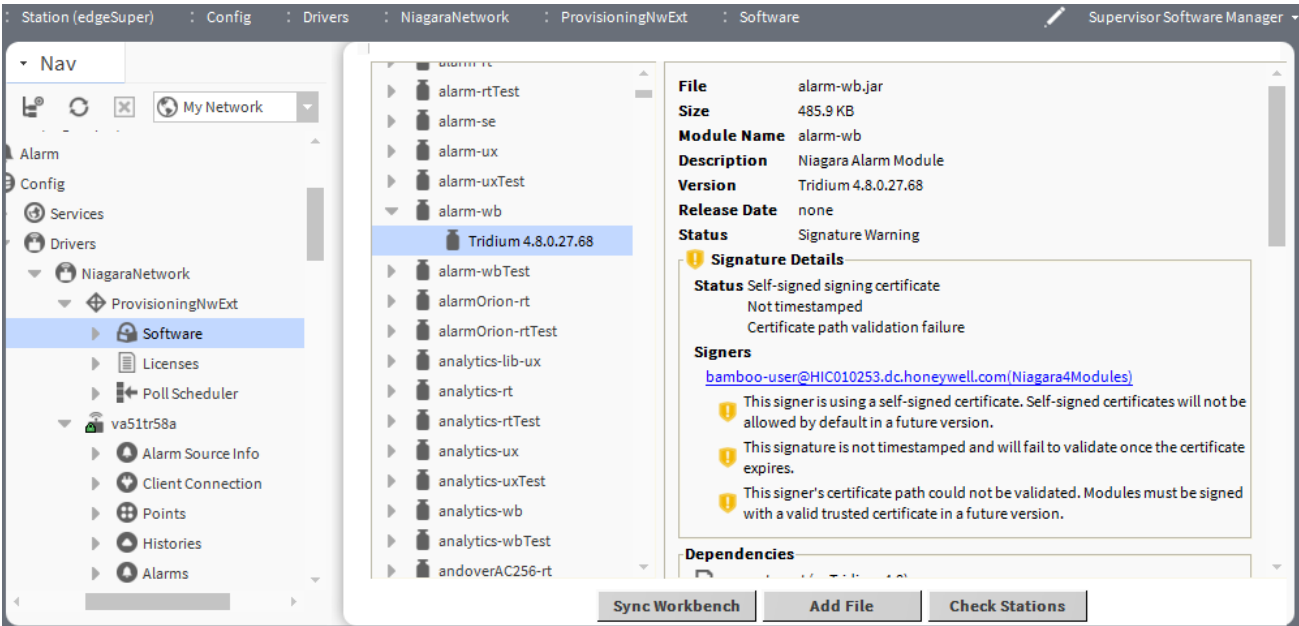
This view lists all components found in the `!sw` directory on the Supervisor PC that may be installed in a station. You access this view by double-clicking the **Software** container under the **ProvisioningNwExt** extension.

When the **SoftwareService** starts (at station startup), it starts a thread that scans the software registry and populates this container. The **Software** container includes software files added to the registry after the initial scan.



Direct children of the **Software** container are summary objects 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 that indicates if the startup thread has finished scanning the registry (by default, this property is hidden). Your key interface to the **Software** container is this default view.

Figure 111 Supervisor Software Manager is default view of Software container



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 EC-Net 4 Pro's Nav tree into this software tree. If the file you drop is an acceptable file (that is readable and correctly formatted), its version is checked against the software database of the Supervisor. If not a duplicate, the file is transferred to the Supervisor for registry in the Supervisor's software database.

Details pane

This right pane of the **Supervisor Software Manager** shows details about any version node selected in the software tree.

Information	Description
File	Displays the name of the module .jar.
Size	Displays the file size in thousand bytes.
Module Name	Displays the logical name of the module.
Description	Briefly identifies what the module does.

Information	Description
Version	Displays the software version number
Status	<p>Indicates the state of the module as follows:</p> <p>Out of Date identifies a module that is older than the one available in the Supervisor's software database.</p> <p>Not Installed identifies a module that is available in the Supervisor's software database, but has not been installed on the platform.</p> <p>Signature Warning indicates a self-signed certificate is being used. These certificates encrypt transmitted data, but cannot validate a server.</p> <p>Up to Date identifies modules that have the same version or possibly a newer version) than that in the Supervisor's software database.</p>

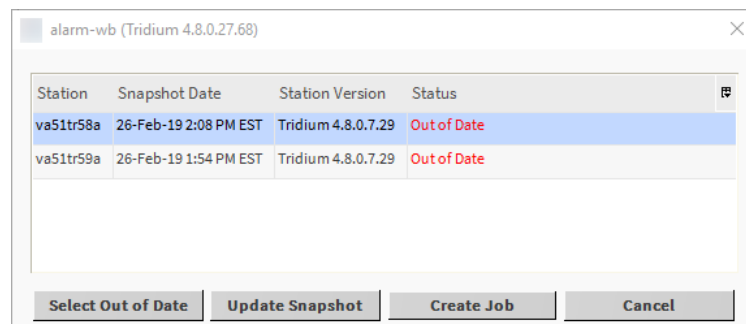
Buttons

- **Sync Workbench** updates the Supervisor station's software database to include all the files that can be installed from the software installation's database (under the !*sw* directory).
- **Add File** adds a software module or distribution file manually using a standard **File Chooser** window.
- **Check Stations** reports the status of an individual module in each station. This component enables this button when you select a software node in the software tree.

Check stations list

Clicking the **Check Stations** button at the bottom of the **Supervisor Software Manager** window opens the check stations list.

Figure 112 Example window for installable from Check Stations function



The window shows one row for each **NiagaraStation**. The last column in the table shows the status of the platform snapshot for each station.

Columns

Columns	Description
Station	The station identifier.
Snapshot Date	The date the most recent state of the station was captured.

Columns	Description
Station version	The version of the software in the platform snapshot (viewable in its Software extension).
Status	<p>Up to Date indicates that the station version is equal to or greater than the software file in the equivalent file in the Supervisor station</p> <p>Out of Date indicates that the version of the file in the Supervisor station is greater than the version installed on the station's host.</p> <p>Not Installed indicates that this file is not installed on the station's host.</p> <p>No Snapshot indicates that no platform snapshot has been taken for this station. There is no basis for comparison. Click to select the station row and click Update Snapshot.</p> <p>Bad Remote file indicates that the version of the module on the station's host is corrupt or otherwise unusable.</p>

Buttons

Buttons	Description
Select Out of Date	Causes the system to display the information for only stations whose software is not current.
Update Snapshot	Captures the current state of the software modules and stores it in the Supervisor database
Create Job	Creates a new provisioning job.
Cancel	Closes the window.

Chapter 10 Troubleshooting

Topics covered in this chapter

- ♦ Why the Start Backup action is NOT recommended

Use these notes to resolve provisioning problems.

Provisioning steps fail when applied to my remote device.

Check to be sure the remote station is not using default credentials/passphrase. In EC-Net 4 v4.7 and later, provisioning no longer runs most job steps against a remote station if default platform credentials or a default system passphrase are detected on the station. Only those provisioning steps that alter default platform credentials or default system passphrase will run.

The Supervisor station's ProvisioningNwExt (under its NiagaraNetwork) and all provisioning-related extensions for NiagaraStations under its NiagaraNetwork have a `fault` status.

A license for the provisioningNiagara feature is missing or expired. Only a Supervisor can be (or needs to be) licensed for provisioningNiagara.

I attempted to drag the BatchJobService folder to my station Config container and got the message, "The following missing modules are required for root targets to be transefered: 'batchJob'."

Your Supervisor station is not licensed for provisioning.

When comparing the module databases on my Supervisor with those on each host station one of the station module files appears to be corrupted.

Back up the station data, commission the station again, and restore the backup.

I clicked the Hyperlink button on the Alarm Console in an attempt to view the details of a provisioning alert and got the message, "Cannot Display Page." What is going on?

You or someone else disposed of the related provisioning job before it was acknowledged. Disposing of a job removes the batch job log (*.bll) file and all batch job step log (*.bisl) files associated with the provisioning job. Provisioning alarms should be acknowledged before disposing of them.

Why the Start Backup action is NOT recommended

The `Start Backup` action works well for systems with only a few controllers. As controllers multiply, there is a better way to back up all stations.

In a large system with many controllers connected to a single Supervisor, invoking the action to backup all stations (or equivalent steps listed) creates a provisioning job that can take an excessive amount of time to complete and can put an undue load on the system, especially if an administrator invokes it at a peak time. Finally, unlike a provisioning job from a **NiagaraNetworkJobPrototype**, which is the preferred backup method, the accumulated backup .dist files remain stored on the Supervisor until manually deleted. These files are not controlled by job retention policies. Without manual intervention, over a long period of time this could lead to a disk-full condition on a Supervisor.

For a large enterprise system where a Supervisor has many controllers, backing up is better accomplished by adding multiple **NiagaraNetworkJobPrototype** components in the Supervisor's station. You copy the components from the **provisioningNiagara** palette. Then you can configure each one for a custom backup job, selecting some of the system's host stations in each component.

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

Additionally, each **NiagaraNetworkJobPrototype** component has job retention policies, which you can configure via its **Prototype Job List** view. You can (and should) configure them to provide automatic disposal of older saved backup .dist files, based on age or some number of earlier saved backup .dist files. For an example procedure including **NiagaraNetworkJobPrototypes**, see “Prototype jobs” in Chapter 3.

Glossary

batch job	A station job that is managed (sent to the station's job service) by the station's BatchJobService. Provisioning functions using batch jobs.
job prototype	Refers to the NiagaraNetworkJobPrototype component found in the provisioningNiagara palette. This persisted component defines a provisioning job by containing one or more processing steps to be performed on a given list of host stations. To set up the regular performance of a task, such as backing up multiple stations, you drag this container to the wire sheet and connect the out slot (Trigger slot) of a standard TriggerSchedule to the component's Submit Job action.
job	The mechanism used to manage a task that a station performs. Jobs run asynchronously in the background while providing user visibility regarding what is going on in the station. See also, provisioning job.
platform snapshot	A list of installed software on a remote host that is running a station. When you access the Software extension under a NiagaraStation component, the system builds or updates this list. Provisioning uses this list when performing queries and installing software.
provisioning	<p>Automating the tasks required to maintain host controllers in a station's Niagara-Network. For the most part, these are platform tasks—that is, they would otherwise done using EC-Net 4 Pro and an individual platform connection directly to a remote host.</p> <p>Using a provisioning robot you can run custom program code in the station running on each host. The stations' ProgramServices run the custom code. The Supervisor station performs all these tasks, modeled in the Supervisor station as provisioning jobs.</p> <p>Outside of provisioning, you must perform similar tasks manually using full EC-Net 4 Pro.</p>
provisioning job	A sequence of steps to perform on one or more host stations. The steps provide a way to perform the same task on multiple stations, such as update licenses, back up, install software, copy files, etc.
software registry	The catalog of available software files, such as modules (.jars) or distribution files (.dists), that are located under the Supervisor's ! <i>sw</i> directory and can be installed in a remote host.

Index

A

Add Station User.....	99
AddStationUserStep.....	99
alarms.....	19
application template provisioning.....	77
application template spreadsheet	
editing.....	77
exporting.....	77

B

backup	
manual.....	47
Backup Stations.....	101
Backup Step Details View.....	161
backup steps.....	48
BackupStationExt.....	83
Batch Job Log File View.....	149
batch job log files.....	35
Batch Job Step Log File View.....	151
batch jobs.....	35
batchJob	
plugins.....	149
batchJob module	
components.....	81
BatchJobService.....	81
installing.....	13
bulk provisioning job	
set up and /run.....	74
bulk template provisioning.....	71

C

certificate	
exporting CSR.....	64
generating a server certificate on remote	
devices.....	60
importing signed certificate.....	65
installing in a remote platform/station.....	63
setting certificate alias.....	61
signing.....	61
certificates.....	57
check stations list.....	197
component	
extensions.....	83
components.....	81
AddStationUserStep.....	99
BackupStationExt.....	83
Copy Supervisor File.....	104
DeployTemplateStep.....	106
DeviceBootstrap.....	98
DhcpDiscoveryStep.....	110
EnableBootstrapModeStep.....	106
ExportCertificateSigningRequestStep.....	108, 111

FileCopyStep.....	104
GenerateCertificateStep.....	112
ImportSignedCertificateStep.....	114
in batchJob module.....	81
InstallableSpec.....	92
InstallableSummary.....	91
InstallApplicationTemplateStep.....	115
InstallBySpecStep.....	118
InstallCertificateStep.....	117
InstallCleanDistStep.....	117
InstallStep.....	92
job steps.....	99
LicenseStationExt.....	85
NiagaraNetworkDiscoveryStep.....	143
NiagaraNetworkJob.....	92
NiagaraNetworkJobPrototype.....	92
PlatformConnection.....	92
ProvisioningBackupStep.....	100
ProvisioningNiagaraNetworkExt.....	87
ProvisioningRobot.....	94
RebootJobStep.....	120
RemoveStationUserStep.....	121
RenameStationStep.....	122
RunRobotStep.....	123
SetCertificateStep.....	126
SetPlatformCredentialsStep.....	127
SetPlatformUserPasswordJobStep.....	129
SetPropertyJobStep.....	130
SetReciprocalConnectionStep.....	140
SetStationConnectionCredentialsStep.....	132
SetStationUserPasswordStep.....	134
SetSystemPassphraseStep.....	135
SetTimeStep.....	137
SetTLSLevelStep.....	136
SignCertificateStep.....	141
SoftwareContainer.....	94
SoftwareStationExt.....	89
StationPollSehcduler.....	95
StationProxy.....	95
TemplateStationExt.....	90
UpdateConfigurationStep.....	145
UpdateLicensesJobStep.....	144
UpdateStationExt.....	91
UpgradeApplicationStep.....	145
UpgradeOutOfDateStep.....	146
UpgradeTemplateStep.....	147
Configure Niagara IdP and SAML Scheme.....	102
connections.....	26
Connections Using Provisioning Station's	
DHCP Server.....	110
ConvertToPerpetualLicenseModejobstep.....	100–101
Copy Local File step.....	104
Copy Supervisor File.....	104–105
Copy Supervisor File step.....	104
CopyFile.....	42

credentials 57

D

data files 35
 Deploy Template 106
 DeployTemplateStep 106
 DeviceBootstrap 98
 DHCP server 26
 DhcpDiscoveryStep 110
 discovery 26
 document change log 9

E

edge device 26
 edit application template spreadsheet 77
 edit template spreadsheet 73
 Enable Bootstrap Mode 107
 EnableBootstrapModeStep 106
 enabling bootstrap mode 23
 Excel file 73
 Export Application Template Configuration 108
 Export Application Template Configurations
 step 107
 export application template spreadsheet 77
 Export Certificate Signing Request 109
 export template spreadsheet 71
 ExportCertificateSigningRequestStep 108, 111
 exporting certificate signing request 64
 extensions 18, 83

F

file
 copying a local file 41
 copying a supervisor file 42
 FileCopyStep 104
 files
 station data 35

G

GenerateCertificateStep 112
 generating a certificate 60

H

histories 36
 host
 update license 40
 host license
 updating 39
 “HTML5” Niagara Network Job Builder 169

I

importing signed certificate 65
 ImportSignedCertificateStep 114–115
 install application template 78
 Install backupdist 92
 Install Certificate 117
 InstallableSpec 92
 InstallableSummary 91
 InstallApplicationTemplateStep 115–116
 installation
 of provisioning software 12
 installation troubleshooting 14
 InstallBySpecStep 118–119
 InstallCertificateStep 117
 InstallCleanDistStep 117
 InstallStep 92

J

job
 creating 23
 schedule 33
 job execution 35
 job list steps
 adding 25
 removing 33
 job log notes 155
 job prototype
 actions 32
 job step components
 Configure Niagara IdP and SAML Scheme 102
 Remove Property Step 120
 Security Job Steps 124
 Job Step Details View 155
 jobs
 configuring 23
 provisioning environment 23

L

legal notices 2
 license
 synchronizing 40
 licenses
 updating 39–40
 LicensesStationExt 85
 local file
 copying 41

M

modules
 installing 54

N

Network License Summary	40
Network Licenses Summary view	164
Niagara 4	21
Niagara Network Job Builder	165
Niagara Network Job List	173–174
Niagara Network Job view	178
Niagara Network Job View	176
Niagara Network Prototype View	180
NiagaraAX	21
NiagaraNetworkDiscoveryStep	143
NiagaraNetworkJob	92
NiagaraNetworkJobPrototype	92

O

one-time job	
creating	23

P

platform connection	
configuring	13
PlatformConnection	92
plugins	149
prerequisites	
provisioning	12
prototype job	27
creating	28
Prototype Job List	158
provisioning	
alarms	19
bulk	74
data files	35
histories	36
installing a clean distribution file	54
overview	11
removing a property	45
setting a property	44
setting station connection credentials	68
steps	11
provisioning application templates	77
provisioning component templates	71, 77
provisioning components	
setting up	47
provisioning device templates	71
provisioning extensions	18
provisioning job	
creating a one-time job	165, 169
management	35
Provisioning Manager	189
Provisioning Robot Editor	190
Provisioning Station Director	191
ProvisioningBackupStep	100
provisioningNiagara	
components	
supporting components	91

extension components	83
job step components	99
supporting components	91
ProvisioningNiagaraNetworkExt	87
ProvisioningNwExt	87
installing	13
ProvisioningRobot	94

R

reboot the system	43
RebootJobStep	120
Remove Property Step	120–121
RemoveStationUserStep	121–122
RenameStationStep	122–123
retention policy	30
robot	
adding to a provisioning job	34
creating job step	33
root CA certificate	
installing in remote platform/stations	63
RunRobotStep	123

S

SAML authentication scheme	57
SAML Idp Service	
configure subordinate station(s)	57
Security Job Steps	124, 126
SetCertificateStep	126–127
SetPlatformCredentialsStep	127–128
SetPlatformUserPasswordJobStep	129
SetPropertyJobStep	130–131
SetReciprocalConnectionStep	140
SetStationConnectionCredentialsStep	132–133
SetStationUserPasswordStep	134
SetSystemPassphraseStep	135
SetTimeStep	137, 139
setting certificate alias	61
setting platform credentials	67
setting platform user password	68
setting reciprocal connection	27
setting security job steps	59
setting station user password	38
setting system passphrase	66
setting TLS level	69
SetTLSLevelStep	136–137
SignCertificateStep	141–142
software	
installing	52–53
upgrading out-of-date	55
software installation	12
on remote hosts	51
Software License Manager	194
SoftwareContainer	94
SoftwareStationExt	89
station	

renaming.....	43
station backup	
example	47
station data files	35
Station Job List	191
Station Software View	193
station user	
adding.....	37
removing.....	38
StationPollScheduler	95
StationProxy	95
stations	
removing from the job.....	43
reordering	44
steps	15
Export Application Template Configurations	107
Supervisor Software Manager	195
Supervisor station	
software database	51
SupervisorLicenses	
components	
SupervisorLicenses	98

T

template spreadsheet	
editing.....	73
exporting.....	71
templates	
preparation for bulk deployment.....	73
TemplateStationExt	90
Thread Pool Job Queue.....	83
time	
setting.....	45
trigger schedule	33
troubleshooting	199

U

UpdateConfigurationStep	145
UpdateLicensesJobStep.....	144
UpdateStationExt	91
updating template or application configurations	80
UpgradeApplicationStep	145
UpgradeOutOfDateStep	146
UpgradeTemplateStep.....	147
upgrading an application template	79
upgrading template	76
user	
adding.....	37
removing.....	38

V

views.....	149
------------	-----

W

Workbench	
software database	51

