



EC-Net^{AX} to EC-Net 4 Migration Guide

User Guide

EC-Net^{AX} to EC-Net 4 Migration Guide

Distech Controls, Inc.
Brossard, Quebec,
Canada

Legal Notice

©, Distech Controls Inc., 2023. 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.....	5
Document change log	5
Related documentation	6
Chapter 1 Migration overview.....	7
Why a migration tool?.....	7
Platform compatibility	8
Platform resource requirements	9
Recommended minimum resources by controller type	9
Checking for minimum resource requirements	10
Driver and application support.....	11
Wire compatibility	12
Platform and station file systems	14
Controller homes	15
Migration workflow process.....	23
Chapter 2 Migration tasks.....	27
Preparing platforms for migration	27
Certificate export	27
Exporting the certificates	27
Configuring a station to migrate users.....	28
Relocating the Kerberos keytab file	28
Station backups	29
Secure license files	32
Run the migration tool	33
About the migration tool.....	33
Running the migration tool	36
Migration tool command usage	37
Migrating modules made with the ProgramModule component	39
Migration details	44
Migration execution.....	44
Migration report (log file).....	45
Post-migration database tasks.....	45
Modifying Program objects	45
Considerations if a station functions as an oBIX server	47
Preliminary checkout of migrated stations	48
Upgrade/convert platforms and install stations	48
Copying the Supervisor station	49
Performing station cleanup	51
Controller conversion	51
Downgrading controllers.....	56
Chapter 3 System verification	57
Non-default environment files	57
Verify station-to-station communications.....	57

Provisioning notes	58
Verify Supervisor to controller platform daemon communications	58
Provisioning considerations when EC-BOS controllers remain	59
Station User notes in migrated stations.....	60
Permissions moved to Roles.....	61
Authentication is user-specific.....	62
Chapter 4 Migrating EC-Net Access stations.....	63
Upgrading devDriver-based video drivers	64
Mapping Supervisor cameras to EC-Net ^{AX} controllers before migration to EC-Net 4.....	65
Preparing the BackupService.....	67
Backing up by creating a distribution file	68
Migrating each station	69
Copying the migrated files	69
Setting up the Supervisor's database password	70
Creating and updating the HSQL database password	71
Configuring network device passwords.....	72
Deleting the old certificates.....	73
Reconnecting the Supervisor and remote stations	74
Restoring the BackupService schedules.....	75
1. Configuring the EC-Net Access-4.8 Supervisor station connection	77
2. Adding a remote controller 2.3 station	79
3. Logging in to the Supervisor from the controller	80
4. Connecting from the Supervisor to the EC-Net ^{AX} -3.8 legacy security station.....	81
Chapter 5 Migration reference	83
Imports	83
API changes.....	83
Primitives	83
Security Manager	84
File reading/writing.....	84
Runtime.exec().....	84
Example Program object fix	84

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-NetAX™ software.

Document Content

This guide provides information about migration of EC-NetAX system running EC-NetAX-3.8 to EC-Net 4 system running EC-Net 4 v4.0, and has the following main sections.

- [Chapter 1 Migration overview, page 7](#)
Provides requirements, restrictions, compatibilities between EC-NetAX and EC-Net 4, background on the need for migration (including major differences), and a general migration workflow process.
- [Chapter 2 Migration tasks, page 27](#)
Provides tasks to start migrating stations, including collecting EC-NetAX-3.8 station archives, running the N4 migration tool, migrated database cleanup, preliminary EC-Net 4 station checkout, and installation of EC-Net 4 software and controller upgrades.
- [Chapter 3 System verification, page 57](#)
Explains system areas that may need attention and possible configuration immediately following the installation of EC-Net 4 upgrades and migrated stations.
- [Chapter 5 Migration reference, page 83](#)
Contains some details on Program object (component) changes in EC-Net 4 , along with an example “Program object fix”.
- [Chapter 4 Migrating EC-Net Access stations, page 63](#)
Describes how to migrate your EC-NetAX-3.8 based security Supervisor and controller stations to EC-Net 4 v4.8 based security stations.
- [4. Connecting from the Supervisor to the EC-NetAX-3.8 legacy security station, page 81](#)
Describes how you can connect and join EC-Net Access-2.4 stations to EC-NetAX-2.3 Security EC-BOS-6AX controllers.

Document change log

May 29, 2023

- Updated “Communications support between EC-Net 4 and EC-NetAX” table in Wire Compatibility topic
- Added procedure for “N4 existing station” to the “Creating and updating the HSQL database password” topic.

February 6, 2023

Added procedure for “N4 existing station” to the “Creating and updating the HSQL database password” topic.

December 10, 2020

Rewrote the last sentence of “Connecting from the Supervisor to the EC-NetAX-3.8 legacy security station” topic.

November 2, 2020

Combined “Migration workflow diagram” topic with “Migration workflow process” topic and “Migration workflow diagram” topic deleted from book map.

June 1, 2020

- Edited certificate descriptions and procedures.
- Confirmed upgrade procedures for EC-Net Access stations.
- Corrected variables used for document branding.

December 16, 2019

- Edited procedures for exporting certificate key information prior to migration.
- Removed procedure for configuring new Photo ID device in Chapter 5. This information is in the *EC-Net Access Installation and Maintenance Guide*.

August 7, 2019

Edited procedures related to migrating EC-Net Access stations from EC-Net^{AX} to EC-Net 4.

April 19, 2019

Added two chapters related to station migration and legacy controller connections.

November 13, 2017

Removed a reference to tagged categories from the procedure, “Tips for Configuring oBIX Authentication”.

November 19, 2016

Removed text in “Driver and Application Support” topic indicating EC-Net 4 support for video drivers.

May 6, 2016

One change was made in the topic, “File Locations”: in the table under the heading *Homes in a Supervisor*, corrected the path for the protected_station_home

February 22, 2016

- Added information about setting the configuration flag before migrating users. Changes were made to two topics:
 - “Features and compatibility between EC-Net 4 and EC-Net^{AX},” and “Assess and prepare for EC-Net 4 migration.”
 - The following topic was added: “Configuring an EC-Net^{AX} station to migrate users”
- Edited the entire document to streamline the way software names are used and to convert task material to task steps.

August 7, 2015

Initial release document.

Related documentation

Additional information about EC-Net platforms, installation and operation EC-Net 4 Pro is available in the following documents.

- *EC-Net 4 Platform Guide*
- *Getting Started with EC-Net 4*

Chapter 1 Migration overview

Topics covered in this chapter

- ◆ Why a migration tool?
- ◆ Platform compatibility
- ◆ Driver and application support
- ◆ Wire compatibility
- ◆ Platform and station file systems

The Niagara Framework® has evolved to the point where existing EC-Net^{AX} platforms and stations, if upgradable, must be *migrated* to EC-Net 4. Therefore, after installing EC-Net 4 Pro, you cannot simply use stations saved in EC-Net^{AX}.

Migration involves several concepts:

- Conversion of EC-Net^{AX} artifacts to the corresponding EC-Net 4 format; for example, station .bog files, Px files. The EC-Net 4 migration tool handles this conversion.
- “Wire compatibility” (communications) between EC-Net^{AX} and EC-Net 4, with the possibility that some migrated jobs must retain some number of EC-Net^{AX} controllers. Some platform types cannot be migrated, for example, EC-BOS-2^{AX} controllers.
- For platforms to be migrated that use custom EC-Net^{AX} software modules, the custom modules must be refactored as EC-Net 4-compatible modules.

Why a migration tool?

EC-Net 4 has different platform and resource requirements than EC-Net^{AX}, and there are other considerations that affect the suitability for a job to migrate from EC-Net^{AX} to EC-Net 4. An understanding of these is essential before beginning migration work for any existing EC-Net^{AX} job.

As a result of the fundamental changes and security improvements in EC-Net 4, a migration process is required. Simple reuse of any saved EC-Net^{AX} station is not possible. Following, are a few of the major reasons:

- Directory structure changes

In EC-Net^{AX}, all folders and files are installed and created/maintained under a single Sys Home directory. For example, for a Windows PC platform that directory is: C:\niagara\EC-Net-AX-3.8.xxx. This directory includes runtime files and executables (modules, JRE, bin) along with configuration data files (stations, EC-Net 4 Pro options and registry).

To improve security, these files and folders are relocated in EC-Net 4. See [Platform and station file systems, page 14](#).

- Software modules were refactored

In EC-Net^{AX}, each software module typically contains multiple content levels: “runtime”, “user interface” (ui), and sometimes “doc”. When installing modules in EC-BOS controllers that are unable to utilize all levels, EC-Net^{AX} platform tools “filter” out unneeded content levels at install time. However, EC-Net 4 uses the **Java Security Manager**, which does not support this model. So, software modules in EC-Net 4 were refactored, essentially split into two or more separate modules JAR files that differ by runtime profile (-rt, -ux, -wb, -se, -doc). This also simplifies future module updates.

The EC-Net 4 migration tool addresses refactoring when converting an EC-Net^{AX} station. However, if you have stations that use custom or third-party modules, those modules must be refactored the same way before starting a migration.

- Users, Categories, and permissions were overhauled

A number of changes to a station's UserService and child User components occurred in EC-Net 4, which provide easier ways to manage large groups of station users. Coupled with new "Role" components and "Tagged Categories", permissions management is more logical and flexible. A related new Authentication-Service architecture allows the flexibility to specify the "authentication scheme" to use for station access at the *user level*. Among other things, this lets you integrate LDAP users into a station's standard UserService, (no special LDAP or Active Directory user service required).

- Fox Service has moved

The **FoxService** component in EC-Net 4 was relocated to each station's main **Services** container, rather than existing as a frozen slot under the **NiagaraNetwork** component. Other related authentication changes were made as well.

- Ports in QNX platforms are sometimes redirected

As part of increased security in EC-Net 4, the station process in all EC-BOS controllers runs as an "unprivileged process", where such processes cannot bind to TCP/UDP ports less than 1024. This affected some services and network types operating as servers using standard ports in this range.

To accommodate this, a new "Server Port" component architecture is used, where client requests to the standard privileged ports are automatically redirected to other (higher) unprivileged ports. Most notably, the default HTTP and HTTPS ports of 80 and 443 for a station's WebService are now automatically redirected to ports 8080 and 8443. For related details, see "Server Port model" in the EC-Net^{AX} Platform Guide.

There are many other differences and numerous new features in EC-Net 4, however ones like the above are examples of "breaking changes" that require a database migration tool, and not just the simple addition of new Java classes and methods.

Platform compatibility

The PC or host controller you wish to upgrade must meet hardware and software requirements.

To qualify as a candidate for migration, any of the supported platforms listed below should be running a released version of EC-Net^{AX}-3.8, with working communications to other platforms at the job site running the same software version. Earlier versions are not supported for migration.

Host Model	EC-Net 4 supported / OS / notes	EC-Net 4 compatibility notes
Supervisor or Engineering Workstation ("Workstation")	Supported OS is Professional or Enterprise versions of Windows 7, Windows 8.1, or Windows Server 2012 or Windows Server 2012 R2 (either 64-bit or 32-bit editions of any of these).	<i>Windows Vista (all) and Windows XP (all) are unsupported.</i> Prior editions of Windows Server, such as Windows Server 2008, are unsupported.
EC-BOS-8	Supported	Initially supports <i>only</i> EC-Net 4 v4.0. At some future point, an update release of EC-Net ^{AX} -3.8 <i>may</i> be available that supports most features of this newest controller.
EC-BOS-7 ^{AX} (JVLN)	Supported with EC-Net 4 v4.4 and earlier.	WiFi option is not supported. Some other option cards are also not supported in (see NPM below).
EC-BOS-6 ^{AX} , EC-BOS-603 ^{AX} , EC-BOS-645 ^{AX} (NPM6E)	Supported with EC-Net 4 v4.4 and earlier.	Unsupported options include: GPRS, and dialup modem. Minimum resources are required (fully loaded units in EC-Net ^{AX} may not run in EC-Net 4 v4.0). For details, refer to Platform resource requirements, page 9 .
EC-BOS-6 ^{AX} , (NPM6)		
EC-BOS-3 ^{AX} (NPM3E)		
EC-BOS-2 ^{AX} , (NPM2)	Not supported	Support is available for data exchange with EC-Net 4 platforms. For example, these stations can be under the NiagaraNetwork of EC-Net 4 Supervisor. For related details, see .
EC-BOS-403 ^{AX}		

Host Model	EC-Net 4 supported / OS / notes	EC-Net 4 compatibility notes
EC-BOS-545 ^{AX}		
EC-BOS-NXT-FL ^{AX}		
EC-BOS-NXS-FL ^{AX}		
Soft EC-BOS ^{AX}	Supported	Support is available for data exchange.

If you first upgrade a system to EC-Net^{AX}-3.8 in preparation for migration to EC-Net 4, be sure to verify proper operation of the system (including NiagaraNetwork communications between stations), before making station backups to use in the migration process.

EC-Net 4 platform compatibility

For any supported PC or host to be migrated, be sure it has the necessary resources available. See [Platform resource requirements, page 9](#).

Platform resource requirements

EC-Net 4 requires more disk space, more Java heap, and more system RAM to run than a similar EC-Net^{AX} station. Therefore, EC-Net^{AX}-3.8 EC-BOS running near capacity will probably not fit the same station when converted to EC-Net 4. A brief summary of reasons why are as follows.

- Disk space (Flash)

EC-Net 4 requires between 14-16 MB more space than an equivalent EC-Net^{AX} installation. Additional space is required by the new Java 8 VM (vs. Java 5), and new UX features enabling a rich browser experience.

- Heap space

For an equivalent station, EC-Net 4 requires more Java heap (a baseline of 10MB) due to new features and functionality, and more system memory when running on EC-BOS controller.

- History RAM disk size (NPM6/NPM6E only)

To free additional system memory, the maximum history RAM disk size on EC-BOS-6^{AX} has been reduced from 64MB to 32MB. This should not affect most units because the default RAM disk size is 32MB. If the RAM disk on EC-BOS-6^{AX} has been increased, then this increase in the RAM disk size comes at the expense of Java heap. Maximum history count has not changed.

Recommended minimum resources by controller type

Each platform you migrate requires a minimum of resources in preparation to run the new system.

The following table summarizes recommended resource requirements before converting to EC-Net 4.

EC-BOS type/model	Minimum Free Disk Space, in MB	Java Max Heap, in MB	Minimum Free Heap, in MB (if maxHeap feature licensed (1))
NPM3 (EC-BOS-3 ^{AX})	22	94	38
NPM6 (EC-BOS-6 ^{AX})	20	94	38
NMP6E (EC-BOS-603 ^{AX} , EC-BOS-645 ^{AX})	24	94	38
EC-BOS-7 ^{AX}	22	94	90

(1) If an EC-BOS is NOT currently licensed with the "maxHeap" feature, then heap usage will not be an issue in EC-Net 4.

The Minimum Free Disk Space recommendation is based on the following:

- 4MB free reserve for future EC-Net 4 expansion
- 4MB for Data Recovery (NPM3E and NPM6E)
- 14MB for EC-Net 4 increase
- 2MB for EC-BOS-6^{AX} boot partition increase

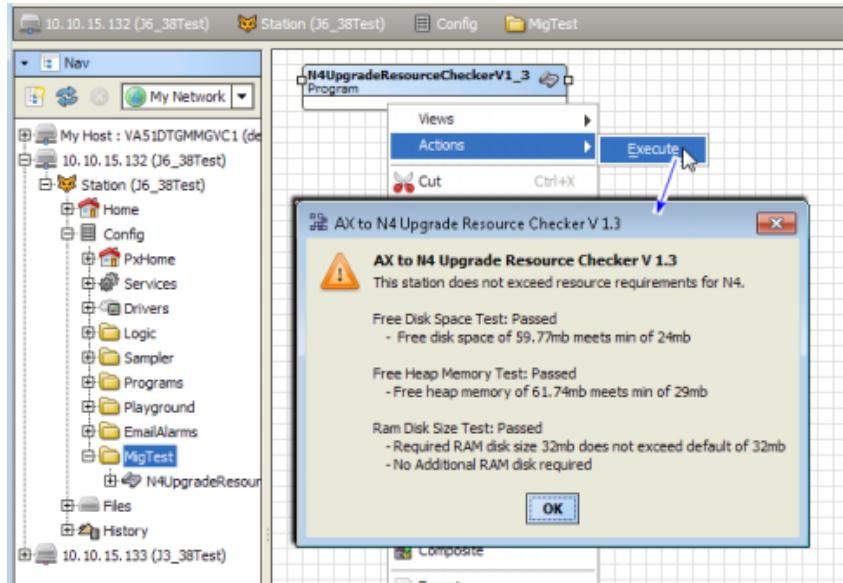
The Minimum Free Heap recommendation is based on the following:

(20% total 86 NB heap size = 17MB) + (13MB additional EC-Net 4 overhead + 8 MB smaller heap (J3/J6 only) = 38 MB

Checking for minimum resource requirements

There is a special BOG file with the EC-Net^{AX}-compatible Program component that you can use to determine if a platform meets the minimum resource requirements for migration. Test results are provided for free disk space, free heap memory, and ram disk size.

Figure 1 Example resources check notification using the Program in EC-Net^{AX}-3.8 EC-BOS station



Alternatively, you can manually check resources, and compare to the recommended resource minimums given in [Table 2 on page 7](#).

Using the resource checker Program

This program provides a binary result: Failed or Passed. Only hosts that pass can be migrated.

Prerequisites: You have the BOG file with the resource checker program, EC-Net^{AX} Pro, and can open a station connection (Fox or Foxs) to the host controller.

- Step 1 Copy the BOG file with the resource checker program component to your EC-Net^{AX}-3.8 system home folder.
- Step 2 Using EC-Net^{AX} Pro, open the station (Fox or Foxs), and expand **Config** to find a folder to copy the Program into, or else add a new folder. You can locate it anywhere under **Config**.
- Step 3 In the Nav tree, locate the BOG file with the resource checker program (it may be named "N4UpgradeResourceCheckerV1_n") and copy it into the station. You should be able to immediately use the program.
- Step 4 Right-click the program in the station, and select **Actions** → → **Execute**.

After a few seconds, a popup **AX to N4 Upgrade Resource Checker** window opens with the test results, as either “Passed” or “Failed”.

- Step 5 Click **OK** to close the window.

Manually checking for free disk space

A minimum amount of free disk space is required.

Prerequisites: The host platform is at EC-Net^{AX}-3.8 and running the station to be migrated.

- Step 1 Using EC-Net^{AX} Pro, open the station (Fox/Foxs)..

- Step 2 Expand **Config→Services** and double-click **PlatformServices**.

The **Platform Service Container Plugin view** opens.

- Step 3 In the **Filesystem** entry for **/ffs0**, compare the **Free** value with the value in [Recommended minimum resources by controller type, page 9](#).

The EC-BOS value should be equal or greater than this minimum value.

Manually checking heap usage

A minimum amount of Java heap is required.

Prerequisites: The host platform is at EC-Net^{AX}-3.8 and running the station to be migrated.

- Step 1 Using EC-Net^{AX} Pro, open the station (Fox/Foxs), and access the **Spy** page (in Nav tree, right-click **Station** node and select **Spy**).

- Step 2 Click **util→gc**

- Step 3 Refresh the page several times, observing the **usedMemory** value under the **After GC** heading.

[Remote Station](#) | [util](#) | [gc](#)

Force Garbage Collection	
Before GC	
totalMemory	24320kb
freeMemory	2500kb
usedMemory	21820kb
After GC	
totalMemory	24320kb
freeMemory	3139kb
usedMemory	21181kb
Freed	
freed	639kb

This represents the low point of Java heap usage.

Driver and application support

The initial release of EC-Net 4 supports most of the commonly used EC-Net^{AX} drivers, including BACnet, Lon-Works, Modbus, SNMP, and some legacy serial-based drivers. Support was dropped for a few drivers.

The following table lists a few drivers and applications that are no longer supported.

Unsupported driver / application	Removed modules / notes
802.1x wired authentication support for QNX-based EC-BOS controllers	Module “platIEEE8021X”
OpenADR client drivers (simple, smart)	Modules “adr” and “dras”

Unsupported driver / application	Removed modules / notes
Alarms to relational database support	Module “alarmRdb”. Superseded by module “alarmOrion”. Note if a third-party application accesses alarms, it will be affected by this (change in the exported schema).
Original crypto/security module, using “CryptoToService” in station	Module “crypto”—replaced by “platCrypto” since EC-Net ^{AX} -3.7 (should only be used by non-Hotspot EC-BOSs, for example, EC-BOS-2 ^{AX}).
Dust wireless mesh network	Module “dust”
Energy Application System (EAS), also known by the branded name VES	Module “eas”
Support for FIPS 140-2 (Federal Information Processing Standard)	Special FIPS distribution (.dist) file
Floating license repository	Module “flr”
Fox tunneling, HTTP tunneling, and platform tunneling	Improved security in EC-Net 4 prohibits most tunneling. Only serial tunneling is supported in EC-Net 4, via the “tunnel” module on the station (server) side and a separate client-side tunnel application.
IBM DB2 relational database support	Module “rdbDb2”
Lon tunneling	Module “lontunnel”. Improved security in EC-Net 4 prohibits this tunneling.
Older video drivers based on the “devDriver”, superseded in EC-Net ^{AX} by “nDriver” based videoDriver framework	Modules “video” (base), “axis”, “dedicatedMicros”, “milestone”, and “rapideye”, superseded in EC-Net ^{AX} by modules “nvideo”, “naxisVideo”, “ndedicatedMicros”, “nmilestone”, and “nrapideye”.
Platform support for EC-Net R2 on any QNX-based EC-BOS controller	EC-Net R2 platform support on EC-BOS-603 ^{AX} /EC-BOS-645 ^{AX} controllers requires EC-Net ^{AX} Pro (3.6, 3.7, or 3.8).

NOTE: This is a partial list. Other rarely-used or limited-distribution software modules were not brought forward to the new version. For more information, check with your support channel to verify availability.

Additionally, any OEM-specific or custom (non-Distech Controls) software modules used on any platform to be migrated must be refactored for use in EC-Net 4. This is a separate process from using the migration tool on a saved EC-Net^{AX} station, and must be completed by the custom module developer before you migrate the station.

Wire compatibility

Supported communications, or “wire compatibility” between platforms and tools can be a factor in different job scenarios. Version compatibility is also relevant when it comes to feature support between EC-Net 4 and EC-Net^{AX}. The following sections describe compatibility between the two major versions of EC-Net with regard to communications and features.

Supported connections

EC-Net 4 Pro can connect to stations running EC-Net^{AX} as well as stations running EC-Net 4. The following table summarizes the possible connections and indicates which are supported in EC-Net 4.

Table 1 Communications support between EC-Net 4 and EC-Net^{AX}

From	To	Supported	Notes
EC-Net ^{AX} EC-Net ^{AX} Pro	EC-Net 4 station, EC-Net 4 Supervisor	No	Prohibited in software.
EC-Net ^{AX} EC-Net ^{AX} Pro	EC-Net 4 platform	No	Prohibited in software.
EC-Net 4 EC-Net 4 Pro	EC-Net ^{AX} platform	Yes	Provides support necessary for EC-Net ^{AX} to EC-Net 4 upgrades. A few functions from the EC-Net ^{AX} Pro remote controller Platform Administration view are unavailable, including the Commissioning Wizard function.
EC-Net ^{AX} EC-Net 4 Pro	EC-Net ^{AX} station	Yes	Supported in software. EC-Net ^{AX} EC-Net 4 Pro is required to configure EC-Net ^{AX} station.
EC-Net ^{AX} station	EC-Net 4 station, EC-Net 4 Supervisor	Yes	Discovery of points, schedules, and histories in EC-Net 4 stations is unavailable. However, you can manually add points. Refer to the following section (<i>Features and compatibility</i>) for more details.
EC-Net 4 station, EC-Net 4 Supervisor	EC-Net ^{AX} station	Yes	Refer to the following section (<i>Features and compatibility</i>) for more details.
EC-Net 4 station	EC-Net ^{AX} Supervisor	No	Not prohibited in software, but unsupported.

Features and compatibility

A limited number of features are supported when communicating between EC-Net 4 and EC-Net^{AX}-3.8.

This table indicates which features are supported when communicating between stations.

Table 2 Feature compatibility table

Feature	From EC-Net 4 to EC-Net ^{AX}	From EC-Net ^{AX} to EC-Net 4	Additional comments
Point discovery	No (EC-Net ^{AX} Supervisor)	Yes (EC-Net 4 Supervisor)	EC-Net ^{AX} stations cannot discover points in EC-Net 4 stations. The alternative is to manually add and configure the points.
Point data	Yes	Yes	EC-Net 4 supports virtual points in EC-Net ^{AX} stations. EC-Net ^{AX} supports virtual points in EC-Net 4 stations.
Export Histories	No	Yes	EC-Net 4 uses different history encoding, and so detects and converts history records as they come in.
Import Histories	No	Yes	
Alarms routed to and acknowledged	No	Yes	
Export Schedules	Yes	No	
Import Schedules	Yes	No	
Synchronize users out (sync out)	Yes, but only users EC-Net ^{AX} is capable of supporting	No	Supported users are those assigned an authentication scheme with a password that can be ported to the EC-Net ^{AX} station. The synchronization process is supported in only one direction.
Synchronize users in (sync in)	No	Yes, but only users EC-Net ^{AX} is capable of supporting	LDAP is a good alternative for network user synchronization. Although EC-Net ^{AX} stations do not support the HTML5 user prototype, you can synchronize users configured with the HTML5 user prototype to EC-Net ^{AX} stations by enabling the <code>UserDefined1</code> configuration flag on the <code>WebProfileConfig</code> property of the user prototype slot in the EC-Net ^{AX} station.

Feature	From EC-Net 4 to EC-Net ^{AX}	From EC-Net ^{AX} to EC-Net 4	Additional comments
Virtual points	Yes	Yes	
Tunneling	n/a	n/a	As Fox, HTTP, and platform communications provide insufficient security, they are not supported in EC-Net 4. Serial tunneling is supported.
Export tags	No, stations cannot join as subordinates	Yes, stations can join as subordinates	<p>For the EC-Net^{AX} station (subordinate) to join an EC-Net 4 station (Supervisor), prior to executing the export tag join, the two-way Fox connection in each station's NiagaraNetwork must be established between the stations. Failure to preconfigure the two-way connection results in an error during the export tag join.</p> <p>Using Px files with ExportTags may be problematic as all Px files must be converted for use with EC-Net 4.</p>
File transfer using Fox (<code>NiagaraDeviceFileExt</code>) is supported.	Yes	Yes	File permission checks are enforced.
Provisioning	Yes	No	<p>Supported operations for EC-Net^{AX} subordinates include:</p> <ul style="list-style-type: none"> • Update licenses • Backup stations • Copy Supervisor file • Install valid EC-Net^{AX} software • Upgrade out-of-date EC-Net^{AX} software • Restore prior station backup, for example, from the step history view of a backup job. <p>EC-Net 4 provisioning robots will not run on EC-Net^{AX} stations.</p>

NOTE: For EC-Net 4 Supervisor to provision EC-Net^{AX} stations in addition to its own stations, you must import the software database from the prior Supervisor into the new Supervisor's software database. However, EC-Net 4 Supervisor cannot directly upgrade platforms to EC-Net 4.

Platform and station file systems

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

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

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

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

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

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

Controller homes

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

Figure 2 Controller System Home (`niagara_home`) and User Home (`niagara_user_home`) locations

The screenshot shows the Platform Administration view for a controller. On the left, the Nav tree shows the 'Platform' section expanded, with 'Platform Administration' selected. Below it, 'Remote File System' is expanded, showing 'Sys Home (Read Only)' and 'User Home (Read Only)'. 'Sys Home (Read Only)' contains directories like bin, defaults, etc, jre, lib, modules, and security. 'User Home (Read Only)' contains daemon, etc, logging, registry, and security. A blue box highlights these two sections. On the right, the main content area displays system configuration details. A blue box highlights the 'System Home' entry in the list, which is described as containing operating system data. Another blue box highlights the 'User Home' entry, which is described as containing configuration data and the installed and running station. A blue arrow points from the 'User Home' entry in the list to the 'User Home (Read Only)' section in the Nav tree.

Baja Version	Tridium 4.7.109.14
Daemon Version	4.7.109.14
System Home	/opt/niagara
User Home	/home/niagara
Host	172.31.66.17 (newSuper2)
Daemon HTTP Port	3011
Daemon HTTPS Port	5011
Host ID	Qnx-TITAN-7E58-A1C7-CC4B-EBF0
Model	TITAN
Product	JACE-8000
Local Date	13-Nov-18
Local Time	8:19 Coordinated Universal Time
Local Time Zone	UTC (+0)
Operating System	qnx-jace-n4-titan-am335x-hs (4.7.109.14)
Niagara Runtime	nre-core-qnx-armle-v7 (4.7.109.14)
Architecture	armle-v7
Enabled Runtime Profiles	rt,ux,wb
Java Virtual Machine	oracle-jre-compact3-qnx-arm (Oracle Corporation 1.8.0)
Niagara Stations Enabled	enabled
Number of CPUs	1
Current CPU Usage	3%
Overall CPU Usage	2%
Filesystem	Total Free Files Max File
/	3,492,848 KB 3,254,942 KB 495 10915
/mnt/aram0	393,215 KB 393,043 KB 0
/mnt/ram0	8,192 KB 8,131 KB 0
Physical RAM	Total Free
	1,048,576 KB 72,272 KB
Other Parts	hsm-ecc508 (Tridium 0.1.50)
	n4-titan-am335x-hs (4.1.0)

- Identifies a controller's System Home (alias: `niagara_home`) in both the Nav tree and the **Platform Administration** view. In the Nav tree, you find the controller's System Home by expanding **Platform**—**Remote File System**. The actual location of the System Home folder for a controller is: `/opt/niagara`.
- Identifies the controller's User Home or daemon User Home (alias: `niagara_user_home`) that contains the installed and running station and other configuration files. The actual folder for the daemon User Home is: `/home/niagara`.

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

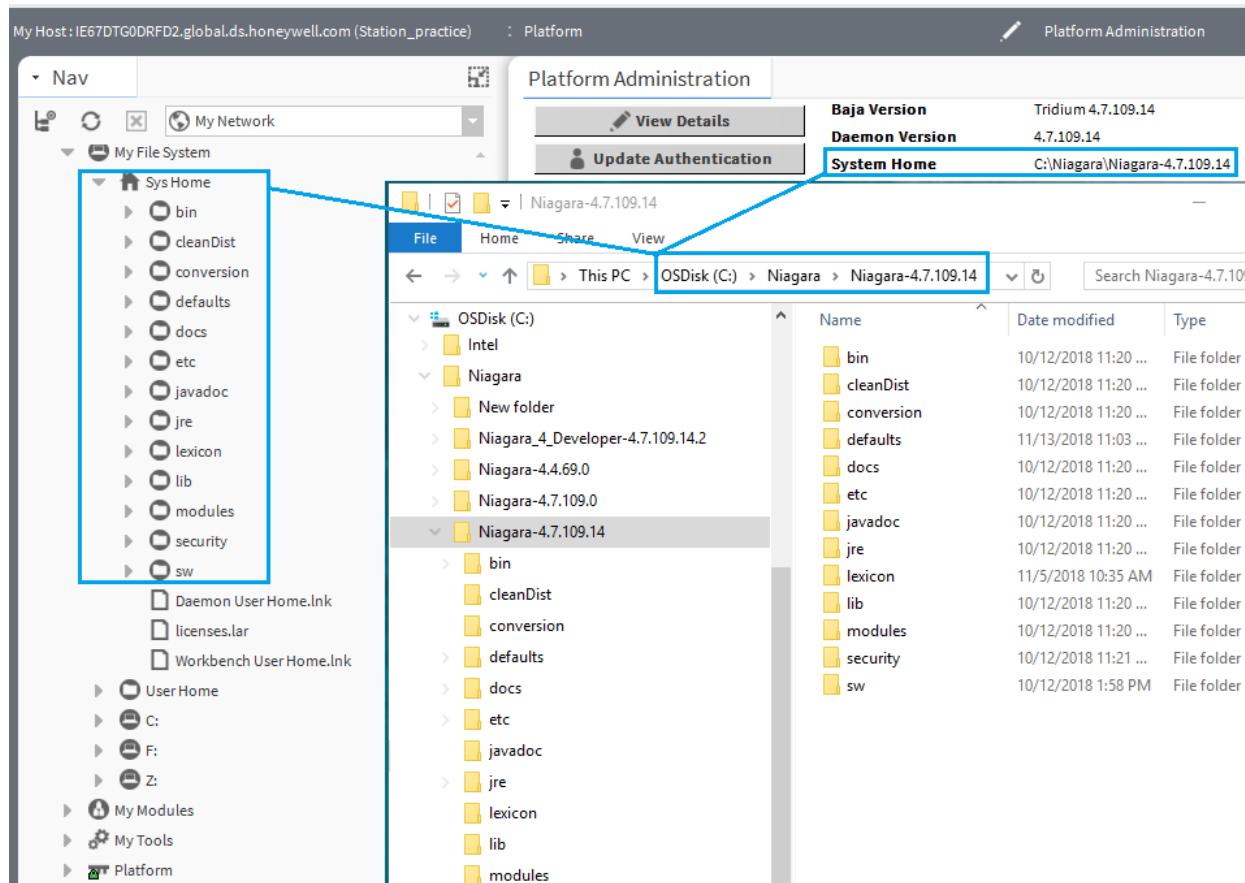
Supervisor homes

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

Supervisor homes include:

- A Sys Home, which contains runtime files, such as core software modules, the JRE, and binary executables.
- A User Home for each user. These are known as EC-Net 4 Pro User Homes. They contain station configuration data, including option files, and registries.
- A platform daemon User Home, which contains the Supervisor station and platform configuration data.
- Two station homes: a Protected Station Home and a Station Home. These are located on the computer's drive C.

Figure 3 Example Sys Home (niagara_home) on a Supervisor platform



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

C:\Niagara\EC-Net4-4.7.109.14.

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

Home in the EC-Net 4 Pro Nav tree	Home in the Platform Administration view	EC-Net 4 alias	Windows folder location and contents	File ORD shortcut
My Host→My File System→Sys Home	System Home	niagara_home	C:\Niagara\EC-Net4-4.x.xx where 4.x.xx is the software version that contains executable and software files.	!(as in Niagara4.x)
My Host→My File System→User Home	N/A	niagara_user_home	C:\Users\username\EC-Net 4-4.x.xx\distech where: username is your name to identify you as the user of your computer. 4.x.xx is a software version. The EC-Net 4 Pro User Home for each human user contains that user's unique configuration files.	~(unique to EC-Net 4)
shared folder	N/A	station_home	C:\Users\username\EC-Net 4-4.x\distech\shared where: username is your name to identify you as the user of your computer. 4.x.xx is a software version.	^(as in Niagara4.x)
stations folder	N/A	protected_station_home	C:\ProgramData\EC-Net 4-4.x\distech\stations\<stationName>	^^ (unique to EC-Net 4)
N/A	User Home	niagara_user_home	C:\ProgramData\Niagara4.x\distech Platform daemon user home (non-human user) holds platform daemon configuration files. Requires a local platform connection to view in the Platform Administration view.	~(unique to EC-Net 4)

Controller homes

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

Figure 4 Controller System Home (`niagara_home`) and User Home (`niagara_user_home`) locations

- Identifies a controller's System Home (alias: `niagara_home`) in both the Nav tree and the **Platform Administration** view. In the Nav tree, you find the controller's System Home by expanding **Platform**→**Remote File System**. The actual location of the System Home folder for a controller is: `/opt/niagara`.
- Identifies the controller's User Home or daemon User Home (alias: `niagara_user_home`) that contains the installed and running station and other configuration files. The actual folder for the daemon User Home is: `/home/niagara`.

Home in the Platform Administration view	Home in the Platform Administration view	EC-Net 4 alias	OFD location and contents	File ORD shortcut
Platform→Remote File System→Sys Home (Read Only)	System Home	niagara_home	/opt/niagara Contains operating system data.	! (as in Niagra4.x)
Platform→Remote File System→User Home (Read Only)	User Home	niagara_user_home	/home/niagara Contains configuration data and the installed and running station.	~ (unique to EC-Net 4)

Windows user homes

For security reasons, each person that is a user of a Windows platform, has their own user home. This means that each Supervisor platform has at least two user home locations: EC-Net 4 Pro User Home (for people), and a platform daemon User Home (for the daemon server processes).

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

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

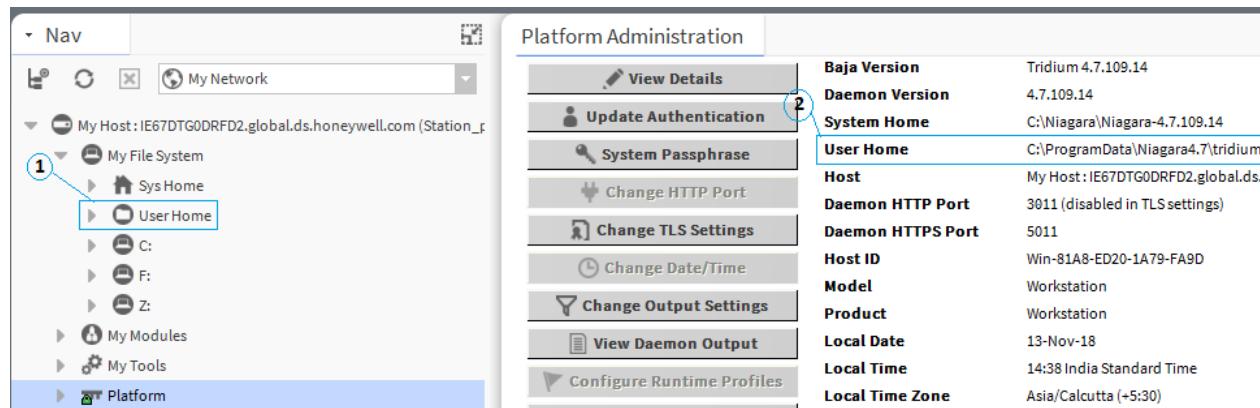
CAUTION: The daemon and EC-Net 4 Pro User Homes are intended to be installed in distinctly separate locations. This separation of homes is for security reasons but it also prevents certain unintended results. For example, when the two homes are installed in the same location the **Station Copier** becomes unavailable, and you will not be able to make a portable copy of the station.

In addition to this daemon User Home, a Windows host has a separate EC-Net 4 Pro User Home for each person (operator, administrator) who logs in with credentials to a Windows-based platform licensed for EC-Net 4 Pro, meaning a Supervisor or engineering workstation. Any given PC or workstation has at least one, and may contain multiple EC-Net 4 Pro User Homes.

Each person's EC-Net 4 Pro User Home is available in the Nav tree as a node under **My Host→My File System** and contains unique configuration information that is not shared. This is where to find any new EC-Net 4 Pro station, as well as any remote station backups, templates and other configuration files. The actual location of each person's User Home is in the **Niagara4.x** folder under your Windows User account.

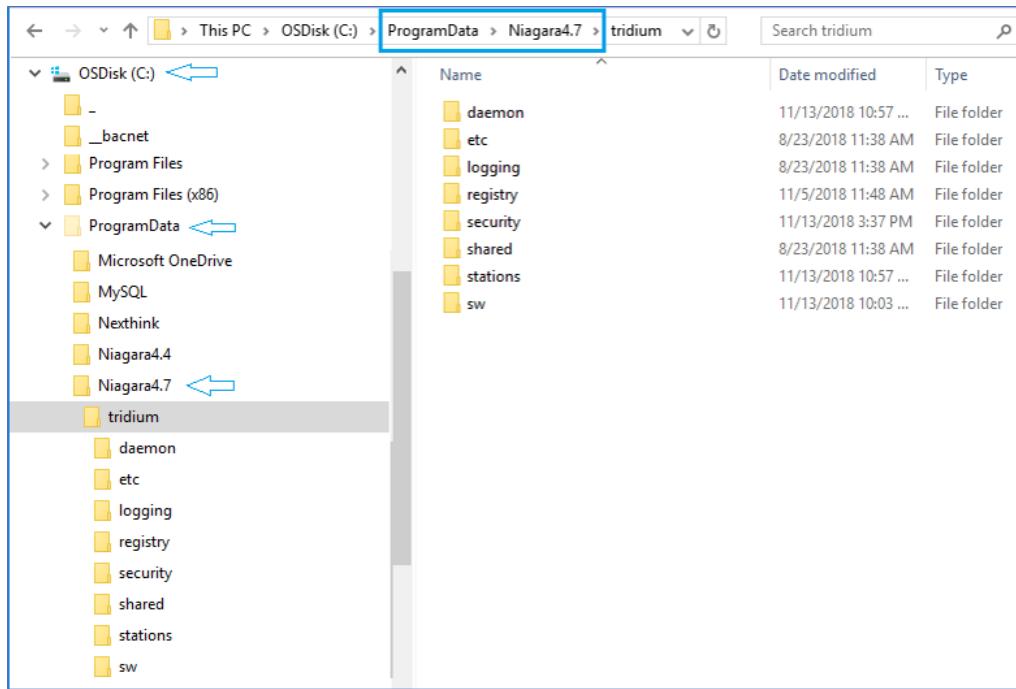
To see both types of User Homes on the same view, open a local platform connection to your Supervisor PC, expand **My File System** in the Nav tree, and double-click on the **Platform Administration** view.

Figure 5 Local platform connection to a Supervisor station with EC-Net 4 Pro and daemon User Homes



- Identifies the EC-Net 4 Pro User Home.
- Identifies the daemon User Home.

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

Figure 6 Example of a daemon User Home location in Windows Explorer

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

The actual location of each user's home folder is under that user's personal Windows account. Some example EC-Net 4 Pro user home locations are:

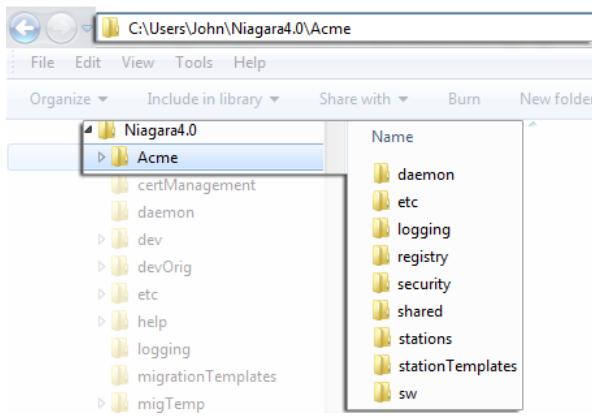
C:\Users\John\Niagara4.x\distech

C:\Users\Mike\Niagara4.x\distech

where "John" and "Mike" are separate Windows user accounts. The first time a Windows user starts EC-Net 4 Pro, the system automatically creates that user's unique User Home folder.

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

The following figure shows an example EC-Net 4 Pro user home location in Windows Explorer.

Figure 7 Example of an automatically-created EC-Net 4 Pro User Home in Windows Explorer

Station homes

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

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

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

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

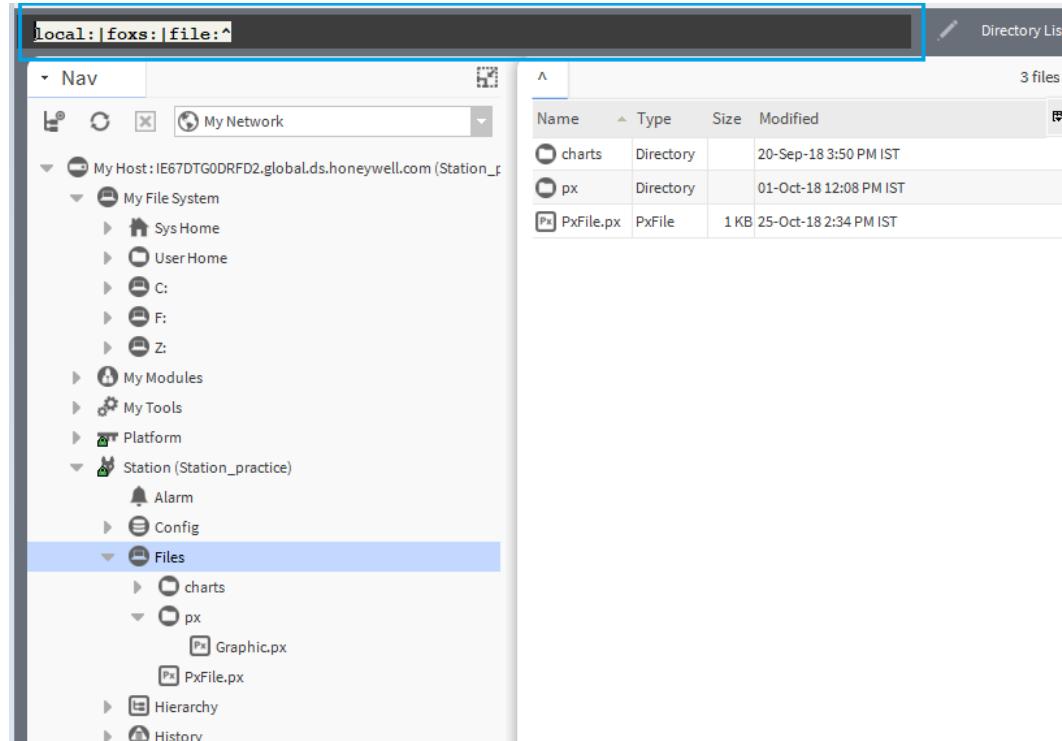
- The `shared` sub-folder, otherwise known as the Station Home (alias: `station_home`), allows all modules to have read, write, and delete permissions.

The alias `station_home` retains the same file ORD shortcut (^) as used in EC-Net^{AX}—only in EC-Net 4 it points to the station's `shared` sub-folder.

Figure 8 Example EC-Net^{AX} station file folders compared to EC-Net 4 station file folders

EC-Net ^{AX} station folder	EC-Net 4 station folder
<ul style="list-style-type: none"> stations demo <ul style="list-style-type: none"> alarm dev frames history html httpd images nav niagaraDriver_nVirtual public px secure tree 	<ul style="list-style-type: none"> stations demo <ul style="list-style-type: none"> alarm history niagaraDriver_nVirtual shared <ul style="list-style-type: none"> dev frames html httpd images nav public px secure tree

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

Figure 9 File ORD for the Station Home in EC-Net 4 now points to the shared folder

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

Copying a new station to the daemon user home

In EC-Net 4, the **New Station Wizard** finishes with an option to copy the station from the station home (the location for each new station) under your EC-Net 4 Pro **User Home** to the **User Home** of the local platform daemon.

Prerequisites: The new station exists in the station home (under User Home).

Step 1 When the **New Station Wizard** prompts you with the option to **Copy station**, select the option and click **Finish**.

Step 2 Make a local platform connection and log on.

The **Station Copier** transfers the station and prompts you with the options to start the station after copying and enabling auto-start.

Step 3 Select the option to start the station.

The **Application Director** opens with the new station present in the daemon **User Home**.

The new station now exists in two locations on your local host: the original location in your EC-Net 4 Pro **User Home**, and also in the platform daemon **User Home**.

Once the station is running in the daemon **User Home**, you can make a backup of the running station, where the backup .dist file goes in the **backups** folder of your EC-Net 4 Pro **User Home**. Or, you can use the platform **Station Copier** to copy the station back to the **stations** folder of your EC-Net 4 Pro **User Home**.

NOTE: Using the **Station Copier** to copy the station back to your EC-Net 4 Pro **User Home** is highly recommended if you made any changes to the station. This is essential if you are installing it (copying it) to any remote platform. Remember, the copy of the station in your EC-Net 4 Pro **User Home** is immediately obsolete as soon as you make changes to the copy of the station running in the daemon **User Home**.

Running a station from the EC-Net 4 Pro User Home

Instead of running a station out of the daemon **User Home**, you can run a station directly from your EC-Net 4 Pro **User Home** (outside of normal platform daemon control).

You do this using the EC-Net 4 console command:

```
station stationName
```

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

Shared file strategy

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

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

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

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

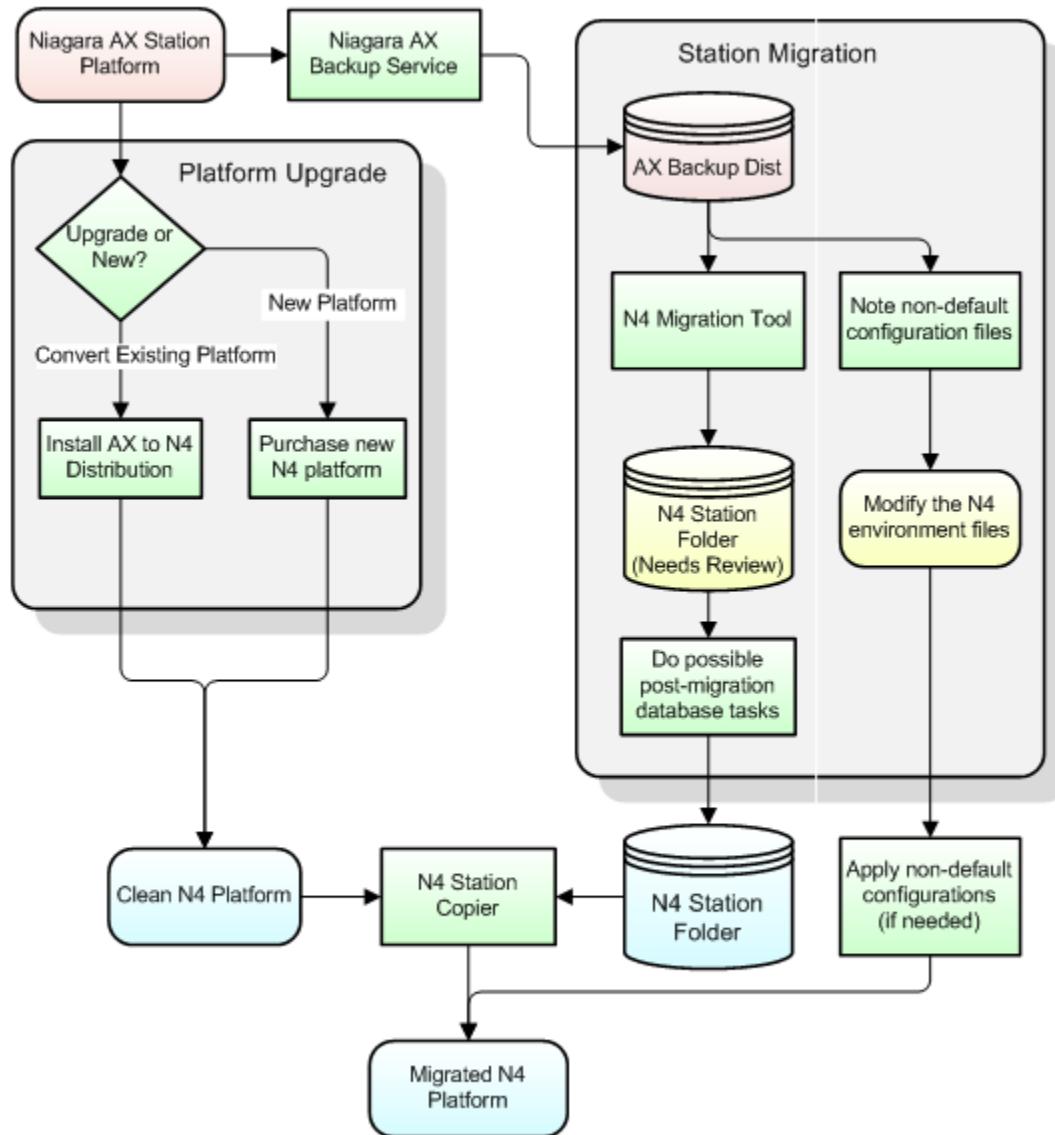
Migration workflow process

Based on an existing EC-Net^{AX} job with a Supervisor and some number of EC-BOS controllers, the workflow process for migration to EC-Net 4 may look similar to below:

Migration workflow diagram

Basic steps in migrating your EC-Net^{AX} installation are included in the following diagram.

Figure 10 Migration steps diagram



Red items indicate EC-Net^{AX} artifacts. Blue items indicate EC-Net 4 artifacts. Green boxes indicate steps you need to take in the migration process.

Environment files are shown in yellow, because they may be generic to all versions of EC-Net. If so, and changed from defaults, they may require individual attention to see if values in them should be re-applied. Examples of such files include modified `!lib/units.xml` or `!lib/colorCoding.properties`.

More detailed migration steps are included in the section [Migration workflow process, page 23](#).

Assess and prepare for migration

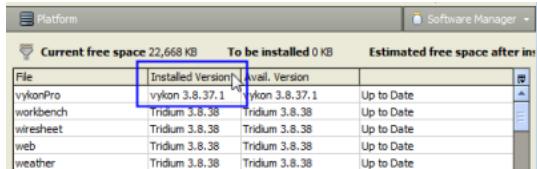
This topic summarizes platform and station-related tasks to complete before migrating.

- Identify those platforms to be migrated. EC-BOS controllers must be running at least EC-Net^{AX} 3.8.nn, and be compatible with EC-Net 4 v4.0. See [Platform compatibility, page 8](#) and [Platform resource requirements, page 9](#) for details.
- Review the current drivers and applications running on the EC-BOS controllers to be converted. See [Driver and application support, page 11](#) for details.
If any drivers or applications are unsupported in EC-Net 4 v4.0, but are critical to retain, you can opt to leave those controllers running EC-Net^{AX}-3.8, and still integrate them with EC-Net 4 Supervisor station. However, you cannot integrate a Supervisor at EC-Net^{AX}-3.8 with EC-Net 4 EC-BOS controllers.
NOTE: For any job that will have a *mix* of EC-Net 4 and EC-Net^{AX}-3.8 platforms, refer to the section [Wire compatibility, page 12](#).
- For each EC-Net^{AX}-3.8 platform to be migrated, identify custom or third-party software modules that it uses. Prior to upgrading, arrange to have those modules refactored as EC-Net 4 modules. Contact the custom module vendor to arrange this.

NOTE:

You cannot migrate a station to EC-Net 4 until all of the modules it uses have been refactored for EC-Net 4.

If you are unsure if an controller has such modules, you can check by making a platform connection to it, then going to the **Software Manager** view.



The screenshot shows a software manager interface with a table of installed modules. The columns are: File, Installed Version, Avail. Version, and Status. The rows include: vykonPro (Installed Version: vykon 3.8.37.1, Avail. Version: vykon 3.8.37.1, Status: Up to Date), workbench (Installed Version: Tridium 3.8.38, Avail. Version: Tridium 3.8.38, Status: Up to Date), wiresheet (Installed Version: Tridium 3.8.38, Avail. Version: Tridium 3.8.38, Status: Up to Date), web (Installed Version: Tridium 3.8.38, Avail. Version: Tridium 3.8.38, Status: Up to Date), and weather (Installed Version: Tridium 3.8.38, Avail. Version: Tridium 3.8.38, Status: Up to Date).

File	Installed Version	Avail. Version	Status
vykonPro	vykon 3.8.37.1	vykon 3.8.37.1	Up to Date
workbench	Tridium 3.8.38	Tridium 3.8.38	Up to Date
wiresheet	Tridium 3.8.38	Tridium 3.8.38	Up to Date
web	Tridium 3.8.38	Tridium 3.8.38	Up to Date
weather	Tridium 3.8.38	Tridium 3.8.38	Up to Date

- Click the “Installed Version” column header to re-sort, as shown above. Scroll down as necessary. Any third-party sourced modules must be refactored by the module vendor before the station can be migrated to EC-Net 4. This assumes that all modules installed in the EC-BOS are required by the controller’s currently running station. Any installed module that is *not used* by the station is inconsequential in for migration.
- If you previously built custom modules of Program components, using the **ProgramModule** feature and its “Program Module Builder” view, you must also refactor these modules. This is something you can do during the migration process.
- If you will synchronize users from the EC-Net 4 Supervisor to the EC-Net^{AX} station that have been assigned an HTML5 prototype, enable the **User Defined 1** configuration flag on the **web_WebProfileConfig** property of each user prototype slot in the EC-Net^{AX} station. (This is necessary because the most likely type of user prototype to be selected is an HTML5 prototype. There is no equivalent prototype in an EC-Net^{AX} station. Consequently, by default, the migration tool assigns the first prototype in the list to each migrated user. This prototype happens to be the Velocity Doc Web Profile.)
- Make an EC-Net 4 v4.0 license request for all platforms to be migrated. You need confirmation that these licenses are ready before migrating the platforms (EC-Net^{AX} licenses do not work in EC-Net 4).

In almost all cases, the calculated host ID remains the same. An exception is if, on a Windows platform, you are moving from a 32-bit install to a 64-bit install (or vice versa), in this case, the host ID differs.

Be sure to archive the old EC-Net^{AX}-3.8 license files as well.

Station backup and database migration

Before migrating, you should back up each station and prepare databases for the new system.

Using EC-Net 4 Pro, make a station *backup* of each EC-BOS controller to be migrated, and do the same for the job's Supervisor station. Be sure to include history and alarm files.

Then run the migration tool against these stations, with output as EC-Net 4 v4.0 station copies.

Sometimes, post-migration database “cleanup” may be needed. You should always do a preliminary checkout of each migrated station.

Supervisor upgrade

You migrate the Supervisor first.

Once migrated, it can start up and communicate with EC-BOS controllers still running EC-Net^{AX}-3.8 (see [Wire compatibility, page 12](#) for related details).

Controller conversion and new station installation

After migrating the controller and Supervisor stations you can convert the controllers and install the new stations.

Before converting EC-BOS controller from EC-Net^{AX}, you must successfully migrate its station to EC-Net 4, including an offline checkout of it using EC-Net 4 Pro. Only then should a conversion to EC-Net 4 be done.

System cleanup

Following the upgrade of all EC-Net^{AX}-3.8 platforms to EC-Net 4 v4.0, including installation of migrated stations, there may be several areas that need attention.

AX to N4 migration tool and code signing

When running the AX to N4 Migration Tool in 4.3 or later, any program object encountered is signed if a code-signing certificate has been configured in EC-Net 4 Pro.

When the migrator encounters the first program object that needs signing, it prompts you for your code-signing certificate password.

Chapter 2 Migration tasks

Topics covered in this chapter

- ◆ Preparing platforms for migration
- ◆ Run the migration tool
- ◆ Migration details
- ◆ Post-migration database tasks
- ◆ Upgrade/convert platforms and install stations

This chapter provides migration tasks, along with reference details as needed.

Preparing platforms for migration

The following sections describe tasks necessary before migrating stations and upgrading platforms.

Certificate export

If your controllers are configured to use secure communications with signed certificates, the best practice is to create new certificates for each controller and install them after migration. However, it is also a good idea to export the current certificates before migration in case you need them later.

To save your existing keys, export them from the User Key Store and the User Trust Store prior to migration. These are deleted from each controller during migration to EC-Net 4. After migration, if you choose to not create new certificates, you can import the exported certificates back using EC-Net 4 Pro and a platform connection.

Exporting the certificates

The User Key Store contains all certificates with their public and private keys. The User Trust Store contains one or more root CA certificates with only their public keys.

Prerequisites: The company serves as its own certificate authority.

- Step 1 Using EC-Net^{AX} Pro, access the controller platform's **Certificate Management** utility. To do this, either open a platform connection and click **Tools**→**Certificate Management** or open the station and double-click on **CertManagerService** (under **Config** > **Services** > **PlatformServices**).
- Step 2 To export the station's server certificate with its private key, click the **User Key Store** tab, select the certificate and click **Export**.
You need to export the unique server certificate from the User Key Store for each controller station.
- Step 3 Enable the **Export the private key** check box and enter an encryption password to protect the private key in this file.
- Step 4 To export a root CA certificate, click the **User Trust Store** tab, select the certificate and click **Export**. Typically, this only needs to be done once because the root CA certificate is the same in every controller station).
NOTE: It is not necessary to export certificates from the System Trust Store as these are default certificates found in every System Trust Store. It is only necessary to export any additional certificates that stored in the User Trust Store.
- Step 5 Repeat the steps above to export the server certificate for each controller station to be migrated.

Make sure exported certificates files are available to the EC-Net 4 Pro PC to use later. For related details, refer to the *Station Security Guide*.

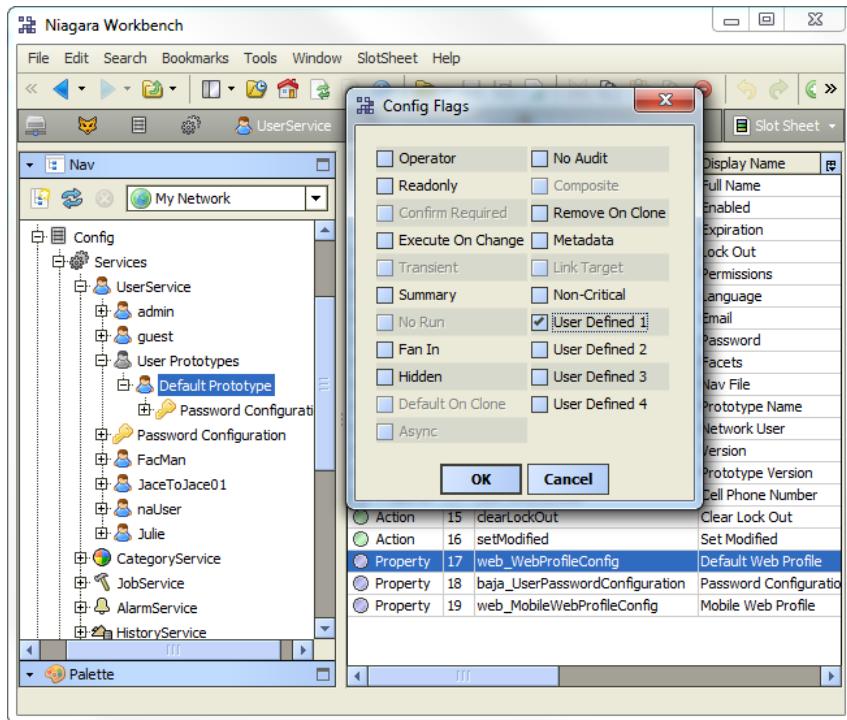
Configuring a station to migrate users

If your EC-Net 4 users will use the HTML5 prototype, which does not exist in EC-Net^{AX}, you must enable the **User Defined 1** configuration flag on the **web_WebProfileConfig** property of each user prototype in your EC-Net^{AX} stations before running the migration tool.

Prerequisites: You have EC-Net 4 Pro access to the remote station.

- Step 1 Expand the **Config**→**Services**→**UserService**→**User Prototypes** node in the station Nav tree.
- Step 2 To view the slot sheet, right-click the user prototype and click **Views**→**Slot Sheet**.
- Step 3 To access the configuration flags, right click the **web_WebProfileConfig** property.

The **Config Flags** window opens.



- Step 4 Enable the **User Defined 1** flag and click **OK**.

Relocating the Kerberos keytab file

If your EC-Net^{AX}-3.8 stations use Kerberos authentication for user services (for example, **LdapV3ADUserService**) and they use a “keytab file” supplied by the site’s Kerberos administrator, the migration tool requires this file to reside under an “ldap” subfolder of each such EC-Net^{AX}-3.8 station. This procedure relocates the Kerberos keytab file prior to migration.

NOTE: If applicable, do this before making source station backups for use with the migration tool.

- Step 1 Using EC-Net^{AX} Pro, open the station (Foxs or Fox), expand the **Config** node, and navigate to the property sheet of the UserService’s authenticator.

For example: **Services** > **LdapV3ADUserService** > **ActiveDirectoryConfig** > **authenticator**

The **Key Tab Location** property ORD value references the current location of the keytab file. Keep this view open.

- Step 2 If it is a *remote station*, open a *new tab* in EC-Net^{AX} Pro, and open a platform connection to the remote host and access the **File Transfer Client** view.

- Step 3 In the (remote) right-side “**Files on IPAddress**” area, navigate to **stations→stationName**, and create a subdirectory titled **ldap**. This is the target location for the **keytab** file.
- Step 4 In the same right-side area, navigate to the **Key Tab Location** noted in [Step 1](#). Select and transfer the **keytab** file from the remote (right-side) to the local (left-side).
- Step 5 In the same right-side area, navigate to the **stations→StationName→ldap** folder created in [Step 3](#), then select and transfer the **keytab** file back from the local (left-side) to the remote (right-side).
The **keytab** file should now be in the **!stations/stationName/ldap** folder.
- Step 6 Switch to the EC-Net^{AX} Pro tab showing the property sheet of the UserService’s authenticator and edit the **Key Tab Location** property ORD value to match the new location of the **keytab** file. Click **Save**.
- Step 7 Save the station (right-click the **Config** node, click **Save**).
- Step 8 Repeat all steps above for each applicable EC-BOS station to be migrated to EC-Net 4.

If this is a local EC-Net^{AX}-3.8 station, such as a Supervisor station, you can use Windows Explorer to make the new **ldap** subdirectory under the **!stations/stationName** folder, and copy and paste the referenced **keytab** file into it. Be sure to update the authenticator’s **Key Tab Location** value with the new location, and save the station database ([steps 6 and 7](#)).

After relocating the Kerberos **keytab** files and updating authenticator properties, test out the operation in EC-Net^{AX}-3.8. A successful operation should continue after the station is migrated to EC-Net 4.

Station backups

For any EC-Net^{AX}-3.8 platform you are migrating, make a station *backup* using EC-Net 4 Pro. You will use this station backup **dist** file as the source when running the migration tool (**n4mig.exe**) from the command prompt of a console window.

CAUTION: Backups for the purpose of station migration should be made only from the BackupService of the running EC-Net^{AX}-3.8 station.

When you migrate a backup **dist** that was made using the **Platform Backup** available from the **Platform Administration** tool, this contains all the stations that are on the platform. For non-embedded platforms, this may be multiple stations, as subordinate stations are often copied to the Supervisor.

You should NOT use the **Station Copier** to make copies for migration.

The proper source for a migration is EC-Net Backup Distribution file from a station’s BackupService, obtained from the host upon which the original station is running. The Backup contains the necessary additional information to correctly migrate the station; a station folder does not. Also, you should use only backup distribution files obtained from the host upon which the original station is running for the source of the migration.

Most of the time there will be one station in the backup that is to be migrated, or at most only a few. If there is more than one station, you will be shown a list of the available stations in the backup. Then you will be prompted to enter a choice selecting which station to migrate. Following, is an example of migration tool output when run on a backup containing nine stations:

```

INFO [09:48:50 29-Apr-15 EDT] [sys] Niagara runtime booted
("C:\Niagara\EC-Net4-4.0.13.3.550") on Win-CE9A-87C6-1526-58EF
[09:48:50 29-Apr-15 EDT] [migration]
-----
[09:48:50 29-Apr-15 EDT] [migration] Migrating station backup
_localhost_demo10_4.dist to backup_localhost_demo10_4.dist
The following station folders were found in this backup:
1 badCop
2 badCop_oldPwds
3 cgAxBac
4 cgOrdsAx
5 demoAppliance
6 goodCop
7 j38
8 nccb11320
9 nccb11602s1_ax
Please select a station from the backup to migrate:
8
Station to be migrated: nccb11320
-----
Migrating station nccb11320 to nccb11320

```

Setting BackupService properties

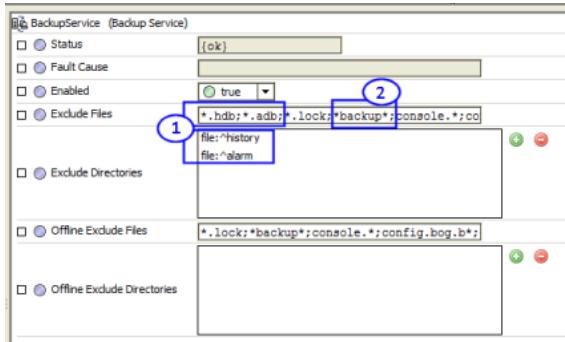
Prior to making a station backup, set properties as needed in the running station's **BackupService**. By default, backups exclude both the history and alarm databases. Other items are also excluded, such as prior saved station database copies, console outputs, and lock files. In most cases, you want to migrate history and alarms databases with your station to EC-Net 4, so you need to modify default properties.

Prerequisites:

Stations are configured and running on a host compatible with EC-Net 4.

In general, this is the recommended backup method for *all platforms*, and the only supported method when making a local backup at the Supervisor PC.

Step 1 In EC-Net^{AX} Pro, open the station and access the **BackupService** property sheet.



Related BackupService properties shown above are:

- a. In the BackupService for each EC-Net^{AX}-3.8 station to be migrated (whether host or Supervisor):

Exclude Files: remove *.hdb; *.adb;

Exclude Directories: remove file:^history and file:^alarm

- b. In the Supervisor station's BackupService, *if configured for provisioning*:

Exclude Files: remove *backup*;

This allows prior station backups to be brought over to the new EC-Net 4 v4.0 Supervisor.

Step 2 Review the **Exclude Files** and **Exclude Directories** settings and make adjustments as necessary. These two properties apply to Fox *online* backups, that is, backups you initiate while the station is running.

Step 3 The bottom two exclude properties apply when you initiate an offline backup with the station *stopped*, using a platform connection and the **Platform Administration** view. By default, alarms and histories *are included* (no **Offline Exclude Directories** entries, also as shown above).

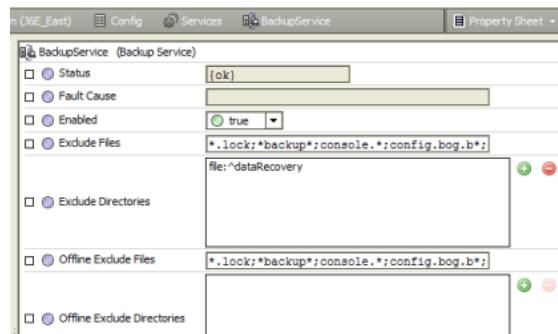
NOTE: Make only *online* Fox backups (with station running) for any local station, such as a Supervisor. Otherwise, if you run an offline backup from the **Platform Administration** view, it includes all stations that may be on that platform. The migration tool rejects such a backup file, as it expects only a single station to be in the backup file. Online Fox backups are more convenient to run than offline backups.

Step 4 **Save** any changes made.

Examples

The following figure shows an example BackupService configured in EC-BOS station.

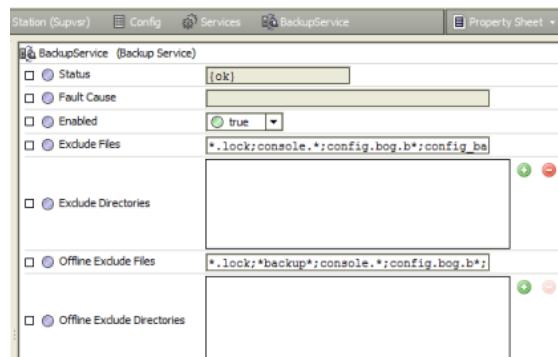
Figure 11 Example BackupService properties in EC-BOS station



Note that **Exclude Directories** still includes `file:^dataRecovery` (not useful in migration).

The following figure shows an example BackupService in a Supervisor station that is configured for provisioning.

Figure 12 Example BackupService properties in Supervisor station configured for provisioning



Note that **Exclude Files** has no entry for: `*backup*`; (to allow prior provisioning backups to be brought over to the EC-Net 4 Supervisor).

Making source station backups

A station backup (.dist file) from a **Backup** command is the preferred archive source when migrating EC-NetAX-3.8 station.

Prerequisites:

The station's BackupService is configured appropriately.

Step 1 Using EC-Net 4 Pro, you can make either a online backup (typical) or an offline backup.

- For an online backup, right-click the  Station node and select  **Backup Station**.
The station continues running while first it saves its configuration locally, then saves a backup .dist file in the backups folder of your niagara_home (install) folder.
- For an *offline* backup, make a platform connection, stop the station from the **Application Director**, then, from the **Platform Administration** view, select **Backup**.

NOTE:

An offline backup applies to EC-BOS stations only. For any local station, such as a Supervisor, you must make an *online* Fox backup (with station running).

After the backup completes, you may *restart* the station so that it can continue working while you run the migration tool in preparation to convert the platform.

Step 2 As a best practice, consider renaming the resulting backup .dist file to a short, meaningful name.

This simplifies typing in the source .dist file name in the command-line migration tool. Do not re-name the .dist extension.

For example, if the file name is backup_stationName_141221_0757.dist, shorten it to: stationName.dist, where stationName is a logical station name.

Step 3 If the new Supervisor or engineering workstation is on a different PC, copy the backup .dist file to a location on that PC.

If the new station is on the same PC, simply leave all the backup .dist files in the default backups folder.

For best performance, do not save backup.dist files in the EC-Net 4 System home or User home directories. The files can be located anywhere else, for example: C:\AxBackups.

Step 4 Repeat this procedure for all EC-Net^{AX}-3.8 stations to be migrated.

Secure license files

Any converted platform (controller, Supervisor, or workstation) requires EC-Net 4 license file, as existing EC-Net^{AX} license files are *incompatible*.

- Before upgrading licenses, back up existing EC-Net^{AX} license files for Windows hosts, and store them in a safe place.
- You can perform migration work on all archived EC-Net^{AX}-3.8 stations without all EC-Net 4 v4.0 licenses secured, provided that you have at least *one* EC-Net 4 v4.0 Supervisor or engineering workstation converted and licensed. However, you should not start upgrading controllers without first securing each license in preparation for the upgrade process.

NOTE: The EC-Net 4 license is not backward compatible with any EC-Net^{AX} install.

- A controller, converted to EC-Net 4 v4.0 uses the same host ID as before the conversion. Provided its operating system remains unchanged, a converted Supervisor PC or engineering workstation also uses the same host ID as before the conversion.

Any *new* installation of EC-Net 4 (EC-Net 4 v4.0) on a PC calculates the host ID for the *first time*. The end of the installation typically generates a "license request form" for web browser submittal, and includes this calculated host ID.

You reference the host ID when you request each EC-Net 4 v4.0 license.

- You need to submit the license request and receive either the license file, or confirmation of it on the licensing server, before you can use EC-Net 4 Pro. For related details, refer to the EC-Net 4 Installation Guide.

- When you receive a new EC-Net 4 v4.0 license file for your Supervisor, do not copy it back to any *EC-Net^{AX}* home's *licenses* folder on the same PC—otherwise EC-Net 4 Pro no longer works. This can also occur if EC-Net^{AX} host downloads the (EC-Net 4) license offered by the licensing server.

In hybrid systems that combine both EC-Net 4 v4.0 and EC-Net^{AX}-3.8 hosts, you need to *retain* EC-Net^{AX} Pro to use for possible station modifications to controllers that remain at EC-Net^{AX}-3.8.

Run the migration tool

Included with EC-Net 4 Supervisor or engineering workstation is a command line utility that you run from a console. This utility is known as the EC-Net 4 migration tool or migrator.

About the migration tool

The migration tool takes a source file, typically EC-Net^{AX}-3.8 station backup *.dist* file, and outputs a migrated EC-Net 4 v4.0-compatible station folder. You can then use EC-Net 4 Pro to open a platform connection to the target platform (for example, a converted controller), and install this station using the EC-Net 4 v4.0 platform **Station Copier**.

Important points to remember are:

- A source backup *.dist* file is *not harmed* in any way, regardless of the migration outcome. Migration output always goes to a different location, along with a text log file that contains execution details.
- The migration tool does not require EC-Net^{AX}-3.8 installed on the same PC, only a path to the source backup *.dist* files.
- You run the migration tool from the standalone EC-Net **Console** window, typically launched from the Windows Start menu.

The following is the log generated while running the migration tool on a very simple Supervisor station backup file. The migrator ran in a standalone EC-Net **Console** window.

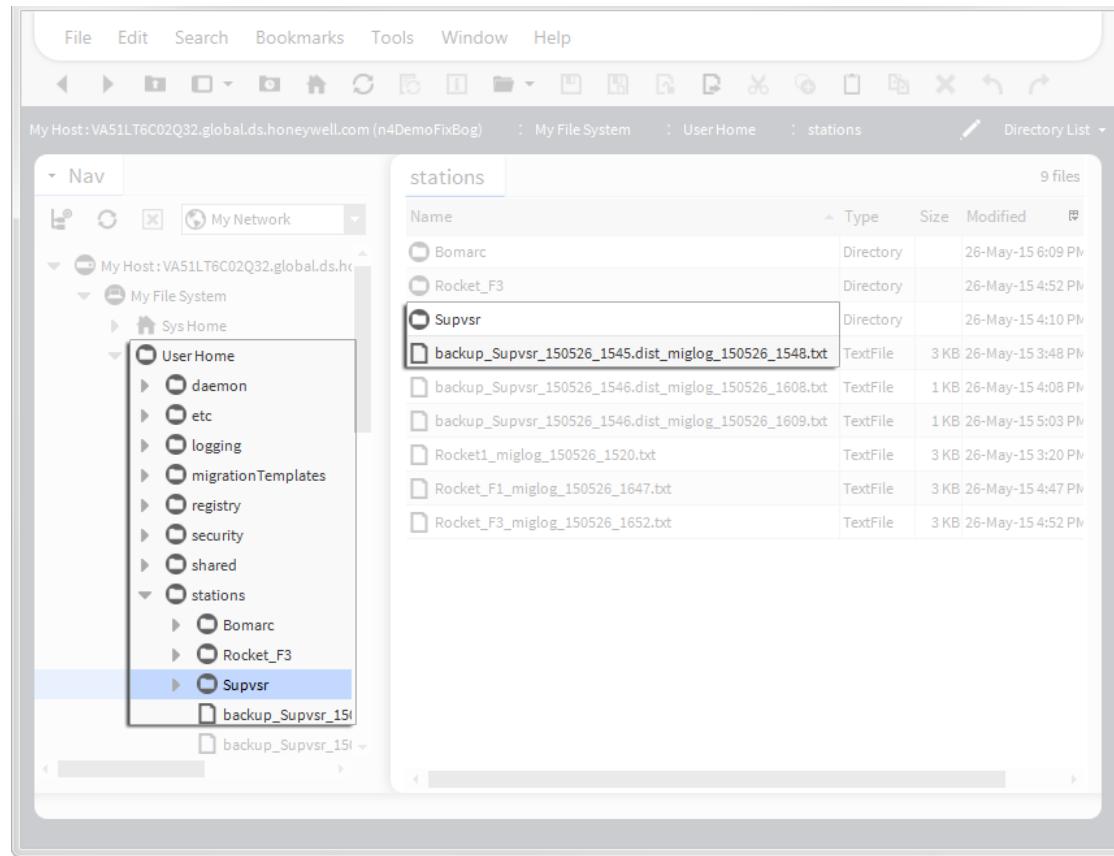
NOTE: Running the migrator in the embedded EC-Net 4 Pro Console is not supported, as it does not properly process the necessary user inputs that may be required if your EC-Net^{AX} station contains passwords that need to be encrypted with a pass phrase.

Figure 13 Example migration tool processing a Supervisor station backup

```
C:\niagara\EC-Net4-4.1.xx.xx>n4mig C:\niagara\EC-Net-AX-3.8.xxx\backups\  
backup_Supvsr_150526_1545.dist  
INFO [15:48:41 26-May-15 EDT] [sys.registry] Up-to-date [92ms]  
INFO [15:48:41 26-May-15 EDT] [sys.registry] Loaded [32ms]  
INFO [15:48:42 26-May-15 EDT] [sys] Niagara runtime booted  
("C:\niagara\EC-Net4-4.1.xx.xx") on Win-23A9-FCD0-B958-4E02  
[15:48:42 26-May-15 EDT] [migration]  
  
[15:48:42 26-May-15 EDT] [migration] Migrating station  
backup_Supvsr_150526_1545.dist to backup_Supvsr_150526_1545.dist  
Station to be migrated: Supvsr  
  
Migrating station Supvsr to Supvsr  
Migration Templates:  
 1: Controller Migration Template (ControllerMigrationTemplate.ntpl)  
 2: Supervisor Migration Template (SupervisorMigrationTemplate.ntpl)  
You can also use 'c' for default Controller and 's' for default Supervisor template.  
Please select a Migration Template: 2  
Using Migration Template SupervisorMigrationTemplate.ntpl  
  
Completed Migration  
Source: C:\Users\John\Niagara4.0\Rocket Industries\migTemp\Supvsr  
Target: C:\Users\John\Niagara4.0\Rocket Industries\stations\Supvsr  
Migration Report: C:\Users\jDoe\Niagara4.0\Rocket Industries\stations\Supvsr_miglog  
_1505261548.txt  
  
Important Migration Notes - see Migration Report for details  
Using Migration Template SupervisorMigrationTemplate.ntpl  
config.bog: REMOVED all uses of type niagaraDriver:NiagaraFoxService in bog
```

In the example above, the migration tool was launched with only the “n4mig” command and the source backup .dist file path, without any options or explicit migration output location. As shown, the migration template was not specified, so the tool paused and prompted for the migration template (1 or 2). After this was entered, the tool continued processing and completed the migration.

The migration tool posts information while processing the backup .dist file. By default, the migration output goes to the stations folder of your EC-Net 4 Pro **User Home**, as shown in the following figure.

Figure 14 Example migration tool output in EC-Net 4 Pro User Home (station folder, migration log file)

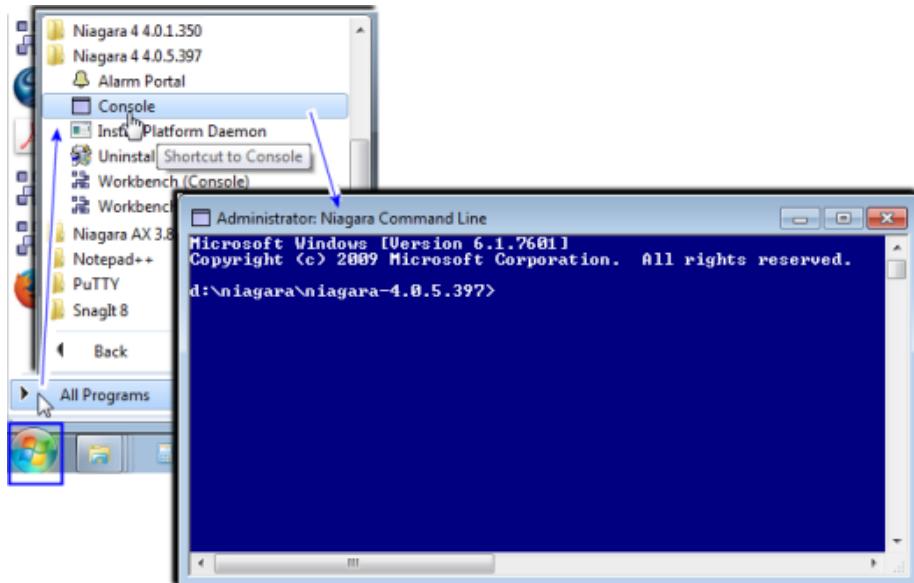
In this example, notice that the migration process created a `Supvsr` folder and an associated log file in the `stations` folder. The target station name is automatically determined from the station name inside the station `backup .dist` file.

Entries in the log file (migration report) display source and target locations for items written, actions performed, and so on. If needed, you can use command line options to create a greater level of log detail.

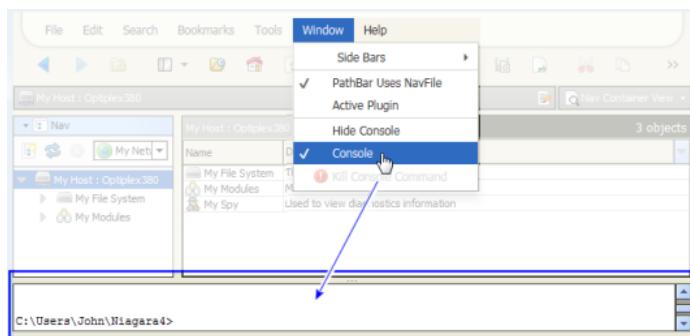
Console choice for migration

You should use only the EC-Net console and not the embedded EC-Net 4 Pro Console to run the migration tool.

For migration , use the standalone EC-Net command line console window, which you typically launch using the shortcut from the Windows Start menu. You may also use a standard Windows command line tool available at [C] :\windows\system32\cmd.exe, for example.



IMPORTANT: The embedded console, that is, the Console prompt available within EC-Net 4 Pro (shown here), cannot be used for migration purposes.



Running the migration tool

With EC-Net 4 installed on your PC, you run the migration tool on the EC-Net^{AX}-3.8 station backup .dist file to migrate a station, or run it on a saved EC-Net^{AX}-3.8 BOG file (.bog) if necessary. EC-Net^{AX} does not have to be installed on this PC—although in many cases, it may be.

NOTE: For any station that uses modules made using the ProgramModule feature, you will need to migrate those modules to EC-Net 4 (refactoring them) before the station can be successfully migrated. See [Migrating modules made with the ProgramModule component, page 39](#).

You must have access to saved EC-Net^{AX}-3.8 station backups, for migrating stations. See [Making source station backups, page 31](#).

Running the migration tool

Step 1 At the PC, open a standalone EC-Net Console window using the shortcut from the Windows Start menu.

NOTE: Do not use the EC-Net 4 Pro embedded Console because it cannot accept keyboard inputs, and thus it hangs on input selections.

If you have a number of station backups in a single folder to migrate, it may be simplest to change the console window's current directory to that location. For example, your console prompt may be similar to: C:\niagara\EC-Net-AX-3.8.xxx\backups>

The above would apply if you have both EC-Net^{AX}-3.8 and EC-Net 4 installed on your PC. Alternatively, enter the full path to source station backup .dist files in the commands.

- Step 2** Enter the migration tool command (n4mig) using syntax that specifies the source backup file, at a minimum. By default (if left unspecified), the migration output targets the stations folder of your EC-Net 4 Pro **User Home**. Generally, this is the recommended method: specify source file only.

If you changed the console's current directory to the location of station backups your command could be:

```
n4mig -t:2 backup_Supvsr_141117_1707.dist
```

(to migrate a backup .dist with that default file name, for a station named "Supvsr").

CAUTION: Do not attempt a station migration using the EC-Net 4 Pro embedded Console. This console cannot accept keyboard inputs, and thus will hang waiting for your input selection. The only workaround at that point is to restart EC-Net 4 Pro.

While the migration tool runs, operation activity is reflected in the console window. When finished, the ending lines in the console window state where the target output results and log file can be found.

For reference details on the n4mig command, see [Migration tool command usage, page 37](#). For general information, see [About the migration tool, page 33](#).

Migration tool command usage

A number of commands may be added on the migration tool command line.

```
n4mig [options] <source> [target]
```

Parameter	Required?	Description / Notes
source	Yes	Typically the EC-Net ^{AX} -3.8 station backup .dist file, but it may also be a bog file, palette file, station folder, or a zip archive of a station folder. You can enter an absolute file path, or a path relative to the current directory of the EC-Net 4 console.
target	No	<p>(Optional) The output station folder name to write the migrated station or BOG file. An absolute file path or a relative path can be entered. Any relative path is relative to the stations subfolder of your EC-Net 4 Pro User Home, that is C:\Users\userName\Niagara4.x\distech\stations</p> <p>TIP: Generally, in a station migration you do not specify a target, as you typically want to retain the existing station name in the migrated station. If it is not specified, the migration tool derives it from the name of the source, its parent folder, or the name of a folder within the archive—whatever fits best. The exception is a source named config.bog. In that case, the target is required. If the supplied target is a relative path, the migration tool targets the result relative to the root of niagara_user_home/stations.</p> <p>If you do specify a target folder name, it cannot already exist (by default an existing station is never overwritten); however, any file path above it must exist. For a related example, see Migrator syntax example 2, page 39.</p>

Options	Description / Notes
-version	Print NRE version information for this EC-Net 4 Pro instance.
-log:level	<p>Specify the log level, where level is one of the following: all, finest, finer, fine, config, info, warning, severe, off</p> <p>The default is "info". This produces the text log file migration report, which lists <i>all the changes</i> made to the EC-Net^{AX} station as a result of the migration. This file is created in the root of the stations folder in your EC-Net 4 Pro User Home. Of course the default ("info") also includes any more problematic "warning" or "severe" messages.</p> <p>After becoming familiar with migrations, you could specify the "warning" log level instead—that lists <i>only</i> the more problematic entries, shortening the log report.</p>

Options	Description / Notes
-keepTemp	Do not clean up temp directories after execution. If used, a "migTemp" folder is left in your EC-Net 4 Pro User Home, with various subfolders of intermediate files used in the migration process. Only use -keepTemp option for troubleshooting purposes.
-showSystemLogs	Shows messages from all loggers; does not hide non-migrator logs.
-help, -?	Print the usage information. By default you see this same information if you enter the n4mig command alone, that is without any other parameters or options.
-o	Overwrite the target station folder, if it already exists.
-t:<template>	<p>When migrating a station backup .dist, this is the template type to use, where template is one of the <i>migration station template types</i> available on your system. Choices are:</p> <ul style="list-style-type: none"> 1 = Controller migration template (for example, ControllerMigrationTemplate.ntpl) 2 = Supervisor migration template (for example, SupervisorMigrationTemplate.ntpl) <p>NOTE:</p> <p>You can also use 'c' for the default Controller and 's' for the default Supervisor template.</p> <p>Default migration template files listed above are found in your Sys Home folder under !defaults/workbench/migrationTemplates. An empty migrationTemplates subfolder is also created in your EC-Net 4 Pro User Home, to store any customized migration station template files. See the EC-Net Templates Guide for more details on working with EC-Net 4 templates.</p>

Migrator syntax example 1

Here, the current directory of the EC-Net console window has been changed to the backups folder of the EC-Net^{AX}-3.8 installation home, such that the prompt is C:\niagara\EC-Net-AX-3.8.xxx\backups>

A command is entered specifying only the source, and *not target* (recommended for station migration).

```
C:\niagara\EC-Net-AX-3.8.xxx\backups>n4mig -t:1 backup_J300E_W_150210_1715.dist
```

The migration tool processes the J300E_W backup .dist file in the current directory, as EC-BOS station, and the target output goes under the stations folder of your current EC-Net 4 Pro **User Home**. The standard log file (migration report) is also created in the root of the stations folder.

So here, migration produces these items:

- C:\Users\userName\Niagara4.x\distech\stations\J300E_W (complete station folder)
- C:\Users\userName\Niagara4.x\distech\stations\J300E_W_miglog_150212_0846.txt (log file or "migration report")

The name of the migrated station (folder), in this case "J300E_W", is determined by the station name *inside* the station backup .dist file, and is not affected by the name of the .dist file. Therefore, you could *rename* the source file from backup_J300E_W_150210_1715.dist to, say, J3Wbu.dist before running the migration tool, and run the command (again, specifying only the source):

```
C:\niagara\EC-Net-AX-3.8.xxx\backups>n4mig -t:1 J3Wbu.dist
```

You would have essentially the same results, with these items:

- C:\Users\userName\Niagara4.x\distech\stations\J300E_W (complete station folder)
- C:\Users\userName\Niagara4.x\distech\stations\J3Wbu_miglog_150212_0847.txt (log file or "migration report")

This may be useful to reduce the amount of typing in the console when migrating stations to EC-Net 4.

Migrator syntax example 2

Here, the EC-Net console window has been left at the default EC-Net 4 v4.0 directory (say when you open it from the EC-Net 4 Console Start menu shortcut), and absolute source and target directories are specified.

```
C:\niagara\EC-Net4-4.1.xx.xx>n4mig -t:2 -log:finer C:\niagara\EC-Net-AX-3.8.xxx
\backups\backup_BigSup_150210_1730.dist F:\migratedStations\BigSupN4
```

The migration tool processes the station BigSup backup .dist file from the absolute source path given, as a Supervisor station, and the target output goes to the absolute “F: drive” location given (perhaps a removable USB drive) in folder migratedStations\BigSupN4. The log file (migration report) is also created there, but with finer level detail than the default.

So in this case, migration produces these items:

- F:\migratedStations\BigSupN4 (complete station folder)
- F:\migratedStations\BigSupN4_miglog_150212_0835.txt (log file with finer detail)

The folder F:\migratedStations must already exist, or else the command fails. The specified target station folder BigSupN4 must not already exist, or else the command fails—it never overwrites an existing station folder (unless the –o option switch is included in the command).

The target station folder determines the migrated station’s name, unlike when *not specifying* a target station folder. In this example, the output station folder (and EC-Net 4 station name), will be BigSupN4, which is *different* than the one in the source backup .dist file. Generally this is not recommended when migrating stations, especially if they are part of a NiagaraNetwork of stations, as other stations store remote station names (in configurations of NiagaraStation and StationRecipient components, as two examples).

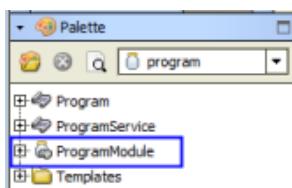
Therefore, when a migrating a station from an EC-Net^{AX}-3.8 station backup .dist file, it is typically best to specify only the source .dist file name, and not specify any target. An example of this is described in [Migrator syntax example 1, page 38](#). After the migration tool completes, if needed, you can then manually copy or move the migrated station folder to another location. For example, you could copy the migrated station folder under a previously created “F:drive” location, say F:\migratedStations.

Migrating modules made with the ProgramModule component

If your installation uses custom modules of Program objects (components) made using the ProgramModule component, they must be refactored for EC-Net 4 before using them with any migrated EC-Net^{AX}-3.8 stations. If you do not migrate the, Program objects they will be deleted in the migrated station.

Prerequisite: You need a working knowledge of the EC-Net^{AX}-3.8 ProgramModule (available in the **program** module’s palette), including its **Program Module Builder** view.

Figure 15 ProgramModule in EC-Net^{AX}-3.8 program palette

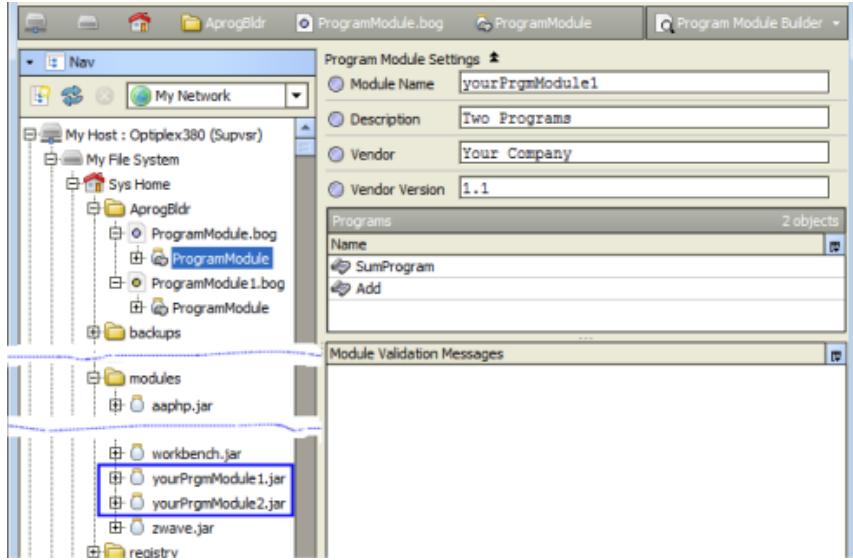


In the EC-Net^{AX} installation, there are different ways the ProgramModule feature can be used to create versioned module JAR files that contain one or more Program objects:

- ProgramModule components copied into a folder under the EC-Net^{AX} Pro file space of the local host, such as a Supervisor, each saved as a BOG file. These ProgramModule components contain Program objects copied from stations, used to build local modules with these Programs.

NOTE: This is the recommended use of ProgramModules.

Figure 16 ProgramModules saved as BOG files in the EC-Net 4 Pro file space



The figure above shows two such ProgramModule BOG files, under an “AprogBldr” folder in the EC-Net^{AX} Pro home. The resultant module JARs are also shown in the local modules folder.

- One or more ProgramModule components copied into a station run on a local host, such as a Supervisor. Typically, this station also has the original Program objects that are now replicated in the station via local modules built by the ProgramModule(s).
- Simply using the EC-Net^{AX} Pro program palette and the ProgramModule in it, pasting in Program objects from stations, then compiling and building local modules with these Programs. In this case, ProgramModules are not persisted—only the modules built using them.

NOTE: Such modules cannot be refactored for EC-Net 4, unless you first use EC-Net 4 Pro to copy ProgramModules from the program palette into a folder in the EC-Net 4 Pro file space, and reconfigure again. Once saved as EC-Net^{AX}-3.8 BOG files, you can run migration on the ProgramModules.

In all but the last case, migration of the modules is straightforward. Migration requires running the source EC-Net^{AX}-3.8 ProgramModule through the migration tool (either as a BOG file, or if within a station, in a station backup .dist file). Next you use EC-Net^{AX} Pro on each migrated ProgramModule to recompile the contained Program(s) and then rebuild the module, which refactors it for EC-Net 4.

Migrating modules from ProgramModules saved as BOG files

If you have saved one or more ProgramModule components in your EC-Net^{AX} Pro file space, you can run the folder (BOG) file containing them through the EC-Net 4 migration tool.

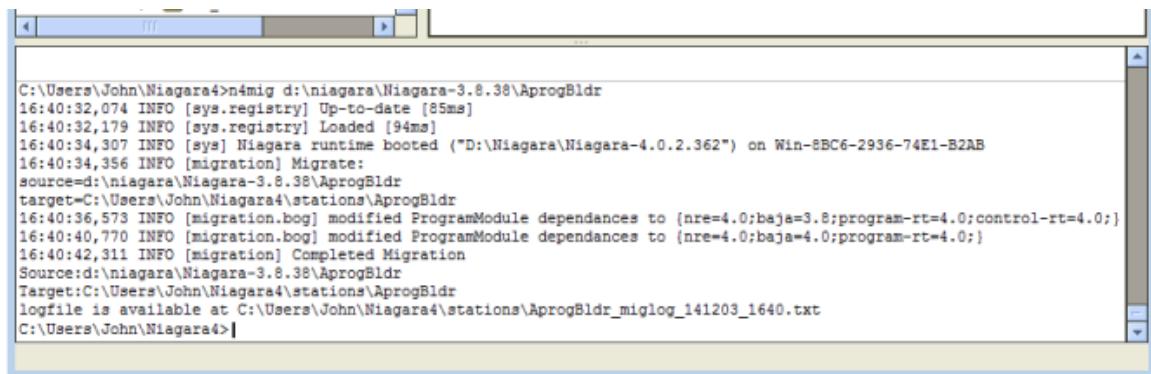
If necessary, you can do this on a PC still running an EC-Net^{AX}-3.8 Supervisor station, but with both EC-Net 4 Pro and EC-Net^{AX} Pro installed.

- Step 1 At the EC-Net 4 Pro PC, open a console. This can be either a standalone Console window or the Console available in EC-Net 4 Pro. See [Console choice for migration, page 35](#).
- Step 2 Enter the migration tool command (n4mig) using syntax that specifies the source folder with ProgramModule BOG files. If using an absolute file path, an example command could look similar to:

```
n4mig C:\niagara\EC-Net-AX-3.8.xxx\AprogBldr
```

for the AprogBldr folder as shown in the previous [Figure 16 Figure 17 on page 34, page 40](#). Leaving the target unspecified (as here), results in a subfolder under stations in your EC-Net 4 Pro **User Home**.

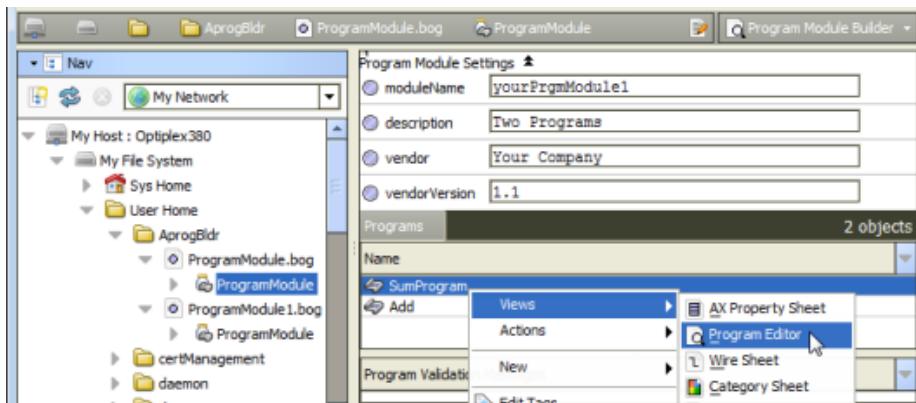
NOTE: After the tool finishes, you can simply cut and paste this “non-station” folder to a better location.



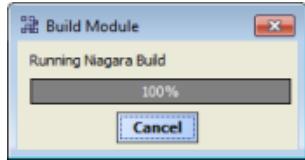
```
C:\Users\John\Niagara4>n4mig d:\niagara\Niagara-3.8.38\AprogBldr
16:40:32,074 INFO [sys.registry] Up-to-date [85ms]
16:40:32,179 INFO [sys.registry] Loaded [94ms]
16:40:34,307 INFO [sys] Niagara runtime booted ("D:\Niagara\Niagara-4.0.2.362") on Win-8BC6-2936-74E1-B2AB
16:40:34,356 INFO [migration] Migrate:
source=d:\niagara\Niagara-3.8.38\AprogBldr
target=C:\Users\John\Niagara4\stations\AprogBldr
16:40:36,573 INFO [migration.bog] modified ProgramModule dependances to {nre=4.0;baja=3.8;program-rt=4.0;control-rt=4.0;}
16:40:40,770 INFO [migration.bog] modified ProgramModule dependances to {nre=4.0;baja=4.0;program-rt=4.0;}
16:40:42,311 INFO [migration] Completed Migration
Source:d:\niagara\Niagara-3.8.38\AprogBldr
Target:C:\Users\John\Niagara4\stations\AprogBldr
logfile is available at C:\Users\John\Niagara4\stations\AprogBldr_miglog_141203_1640.txt
C:\Users\John\Niagara4>
```

While the migration tool runs, operation activity is reflected in the console window. When finished, the ending lines in the console window state where the target output folder and log file can be found. As shown above, you are *not* prompted for a station “type” (unlike if migrating the EC-Net^{AX} backup .dist file).

- Step 3 If the target folder was created in your stations folder, move (cut and paste it) to another location in your EC-Net 4 Pro **User Home**. You can do this in the Nav tree, using right-click commands.
- Step 4 In the Nav tree, expand the target folder to see the BOG file(s) with ProgramModule(s).
- Step 5 Double-click a ProgramModule for its **Program Module Builder** view.

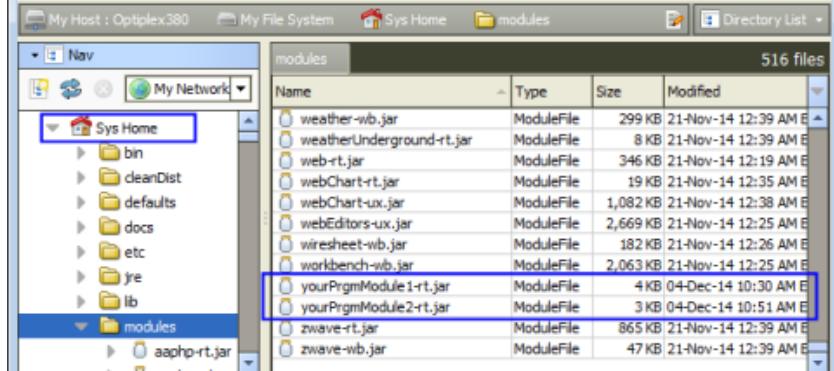


- Step 6 Right-click one of the contained Programs and select **Views**→ **Program Editor** (as above).
- Step 7 In the **Program Editor**, click **Save & Compile** (**ProgramEditor**→ **Save&Compile**). Activity occurs in the console area at the bottom of the EC-Net 4 Pro window.
If successful, the Program status should be up-to-date, and the **Save Bog** icon should be available.
- Step 8 Click **Save Bog**.
- NOTE:** If a Program compile fails, you must make one or more modifications to that Program, working in its **Program Editor**.
- Step 9 Repeat steps 6 and 7 for all Programs in the selected ProgramModule, so that *all* Programs have been successfully compiled and saved.
- Step 10 Double-click the ProgramModule for its **Program Module Builder** view, then click **Build**.



As shown above, a popup dialog appears while the module is created. Completion may take several seconds. Related details should also appear in the console area at the bottom of the EC-Net 4 Pro window.

- Step 11** If multiple ProgramModules are in the folder, repeat steps 5 through 9 for each one, so that a module for each is created. In the example folder shown here, there are two ProgramModules.



As shown above, the modules named `yourPrgmModule1-rt.jar` and `yourPrgmModule2-rt.jar` were created by the two ProgramModules.

- Step 12** When finished migrating modules, close and restart EC-Net 4 Pro. This is necessary to update the registry. Modules should now be ready when running the EC-Net 4 migration tool on a backup .dist file of any EC-Net^{AX}-3.8 station that uses them. See [Running the migration tool, page 36](#).

Migrating modules from ProgramModules in a station database

If ProgramModule components are used within the EC-Net^{AX}-3.8 station (instead of in standalone BOG files in our EC-Net^{AX} Pro file space), you need to migrate that station in the standard way, that is using a station backup .dist file as the source.

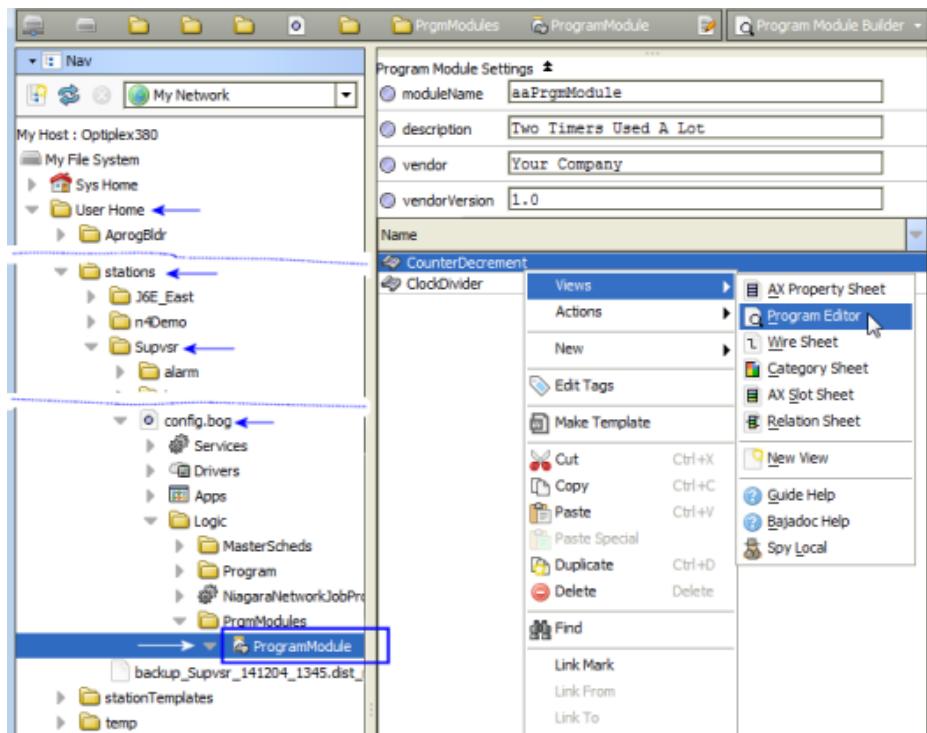
Prerequisites:

However, before installing that migrated station, or any other migrated station that uses modules produced by it, you must use EC-Net 4 Pro to get refactored modules from its ProgramModules. This means recompiling and saving Programs in each ProgramModule, then running a build on each one. Then you must migrate that station *again* (now that the necessary refactored modules are available).

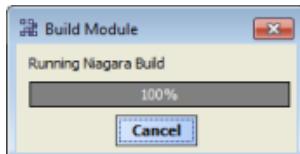
NOTE:

If necessary, you can do this on a PC still running the EC-Net^{AX}-3.8 Supervisor station, but with both EC-Net 4 Pro and EC-Net^{AX} Pro installed.

- Step 1** Run the migration tool on the saved EC-Net^{AX}-3.8 station backup .dist file.
- Step 2** In the EC-Net 4 Pro Nav tree, expand the `stations` folder under your **User Home**, to see this newly-migrated station.
- Step 3** Expand the station's `config.bog` file, and navigate to the first ProgramModule.
- Step 4** Double-click the ProgramModule for its **Program Module Builderni**

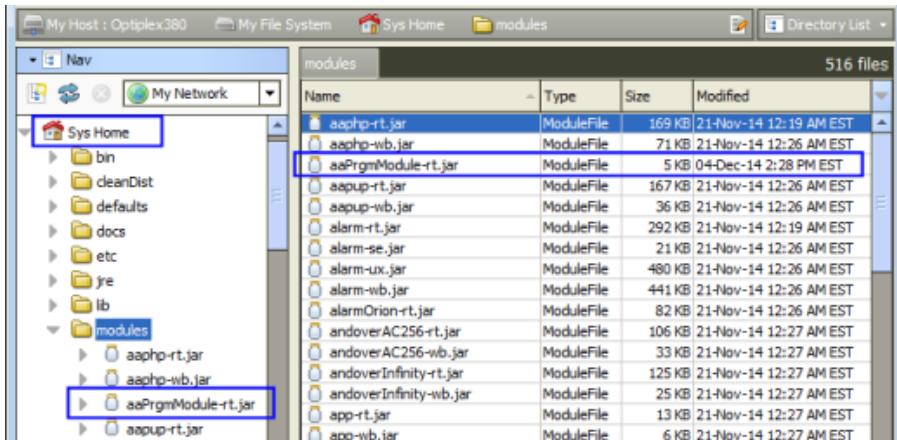


- Step 5 Right-click one of the contained Programs and select **Views**→ **Program Editor** (as above).
- Step 6 In the **Program Editor**, click **Save & Compile** (**ProgramEditor**→ **Save&Compile**). Activity occurs in the console area at the bottom of the EC-Net 4 Pro window.
If successful, the Program status should be up-to-date, and the **Save Bog** icon should be available. Click **Save Bog**.
- NOTE:** If the Program compile fails, you need to make one or more modifications to that Program, working in its **Program Editor**. See [Modifying Program objects, page 45](#).
- Step 7 Repeat steps 5 and 6 for all Programs in the selected ProgramModule, so that *all* Programs have been successfully compiled and saved.
- Step 8 Double-click the ProgramModule for its **Program Module Builder** view, then click **Build**.



As shown above, a popup dialog appears while the module is created. Completion may take several seconds. Related details should also appear in the console area at the bottom of the EC-Net 4 Pro window.

- Step 9 If multiple ProgramModules are in the folder, repeat steps 4 through 8 for each one.



As shown above, the module named aaPrgmModule.jar was created by the ProgramModule.

- Step 10 When done migrating modules, right-click the station's config.bog file and click **Save**.
- Step 11 Close and then restart EC-Net 4 Pro. This is necessary to update the registry. Migrated modules should now be ready when running the migration tool on EC-Net^{AX}-3.8 station backup .dist files, including the station with the source ProgramModules.

Migration details

Details on migration are in the following reference topics:

Migration execution

The migration tool migrates a source artifact to its appropriate EC-Net 4 target. The typical source is EC-Net^{AX}-3.8 station backup .dist file. The output (target) is a station folder, in your EC-Net 4 Pro **User Home**.

At the conclusion of the tool execution, you see these Source and Target locations listed:

```
09:31:37,355 INFO [migration] Completed Migration
Source:C:\niagara\EC-Net-AX-3.8.xxx\backups\backup_Supvsr_141117_1707.dist
Target:C:\Users\John\Niagara4.x\distech\stations\Supvsr
```

You can then install this migrated station into the daemon User Home of EC-Net 4 platform, for example the local platform if a Supervisor, or in a remote platform, such as the migrated station for a controller that has been converted.

Upon execution, the migration tool scans each folder and file in the backup recursively, and migrates each in turn as appropriate. Certain folders, such as the folder tree comprising the history database of a station, are handled separately.

NOTE: Any environment files in the backup .dist that are outside of the station folder, for example, properties files in the EC-Net^{AX} platform's !/lib folder, are not migrated. If you know of any that were edited from default values, and they still apply to EC-Net 4 platforms, you may need to re-apply changed values in the platform after upgrading to EC-Net 4 (however, do not copy entire EC-Net^{AX} files into the EC-Net 4 environment).

If desired, do this after installing the migrated station.

For each migrated artifact, the migration tool determines if a FileMigrator is registered to handle migrating that file type. If no special migrator is registered, the default migration is to simply copy the file as is from the source location to the target location.

In the case of BOG file migration, the BogMigrator opens the .bog file, and walks the XML element tree. For each element it encounters, it checks to see if there is a BogElementConverter registered to handle converting that type.

- If no converter is registered, the type is simply passed through and re-encoded.

- If a converter is registered, the converter is given the element to process.
- If the element cannot be decoded, this is usually because the module that contains the type existed in EC-Net^{AX} but was removed for EC-Net 4. In this case the element will be removed from the station.
- In the case where the EC-Net 4 equivalent module exists, the registry check provides a converter that can convert the obsolete type into the appropriate EC-Net 4 type.

Migration report (log file)

The log file produced each time you run the migration tool answers the question “What did the tool do?” At the default `info` log level, the migration report lists the changes made to migrate the source to the target. If needed, you can have the migration tool generate even more detailed output by specifying another log level (`config`, `fine`, `finer`, `finest`) as an option in your command.

You should *always review the migration report* to make sure that:

- Nothing was removed that is critical to your station’s operation.
- There were no failures, or items that could not be migrated by the tool.
In some cases, issues may require post-migration “cleanup” tasks.

Inspecting migration reports (log files)

We recommend you inspect migration log files for:

- Migration failures

If any migration failure occurs, this will be noted in the report. You should report any migration failures to your support channel. In some cases it may be necessary to modify this part of the station manually, and your support channel should be able to help with that.

- Program compilation failures

The migration tool attempts to recompile all **Program** objects in the station with the new modules. In some cases, this may not be successful. Programs that cannot be compiled will be noted in the report, and you will need to modify the program manually to allow it to compile.

- Removed station components

Station components whose type or module does not exist in EC-Net 4, and for which there is not a type to migrate to, are removed. This should not occur frequently, as the modules that were removed for EC-Net 4 were generally unused or deprecated modules.

If you find that a component has been removed that should not be, please report this to your support channel, as it may be that a converter needs to be developed for that type.

- Any WARNING or SEVERE messages

These messages are included in the migration report prefixed by “WARNING” or “SEVERE”, as appropriate. They indicate a more significant issue, likely a problem with the tool execution. You should inspect items referenced in such messages to make sure migrated artifacts are what you want.

Post-migration database tasks

Sometimes you may need to perform post-migration tasks on migrated station databases, using EC-Net 4 Pro. For example, certain Program objects may need to be changed and recompiled in EC-Net 4, as flagged in the migration report (log file). Other possible station changes may be needed too, in certain circumstances.

Modifying Program objects

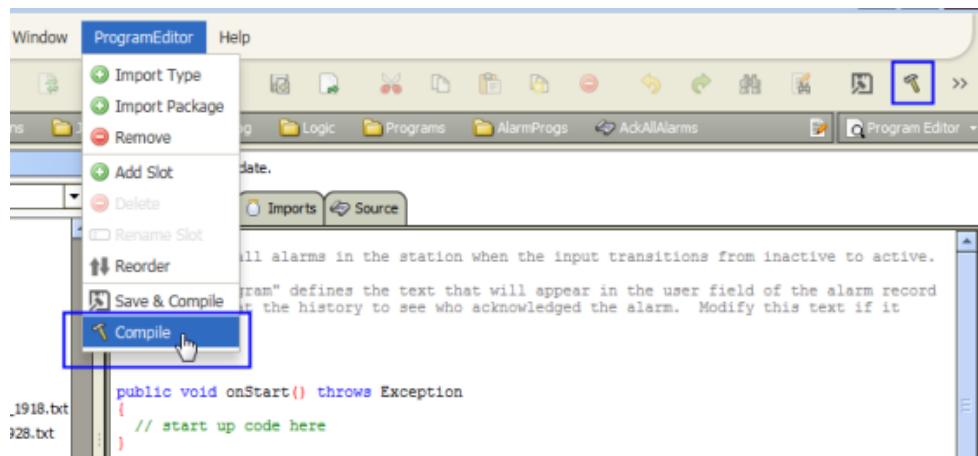
The migration tool attempts to compile all EC-Net^{AX} Program objects (components) in the station to be compatible with EC-Net 4, flagging any that fail to compile properly in the migration report, or log file. Using EC-Net 4 Pro, you should modify any such Program objects until they recompile successfully, before installing the station.

Prerequisite: You must have a working knowledge of Java programming, Program objects, and the **Program Editor** view in EC-Net 4 Pro. Access to Developer-level EC-Net 4 documentation may be needed.

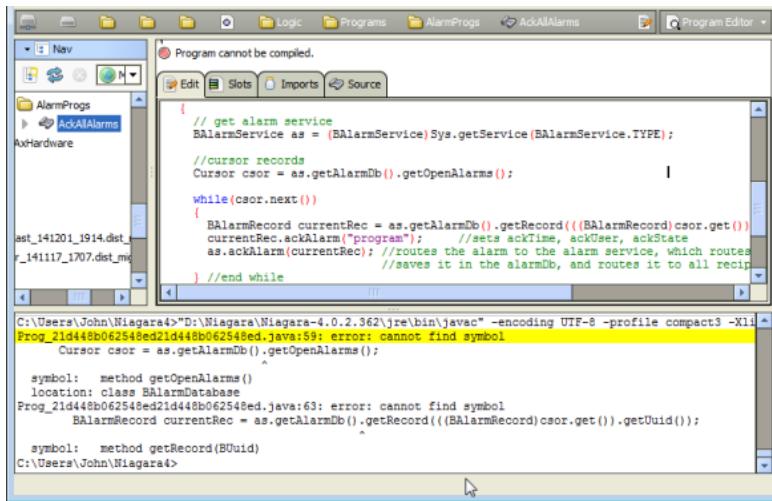
Modifying and recompiling Program objects offline

You can open a station *offline* in EC-Net 4 Pro to modify and recompile Program objects if necessary, resaving the station (config.bog file) when done. If necessary, you can do this on a PC still running an EC-Net^{AX}-3.8 Supervisor station, but with both EC-Net^{AX} Pro and EC-Net 4 Pro installed.

- Step 1 Open EC-Net 4 Pro, if not already open.
- Step 2 In the Nav tree, expand **My File System** and navigate to the station folder of the migrated station that had Program compilation failures.
Typically, such stations are under your **User Home**, under the **stations** folder.
- Step 3 Expand the station folder, and expand the config.bog file under it.
- Step 4 In the station database, navigate to a Program object identified to have a “compilation failure” in the migration report (log file).
- Step 5 Open the **Program Editor** view for the component. (One way is from the Nav tree, right-click the Program and select **Program Editor**).
- Step 6 In the **ProgramEditor** menu or toolbar, click  **Compile**.



One or more errors result in the Console area at the bottom. Often all possible errors are not included, as the compiler stops on failure and outputs an error. Therefore, once you correct one or two errors, subsequent compile attempts may uncover other errors.



In this example, errors shown relate to the methods `getOpenAlarms()` and `getRecord(B0rd)` having moved from the `BAlarmService` class to a new `AlarmSpaceConnection` class, located in the `javax.baja.alarm` package of the `alarm` module. To fix this, you would need Developer-level documentation on “breaking changes” in EC-Net 4 Alarm API, and make the required changes in the program code (further details are outside the scope of this document).

Depending on the Program object, other possible compile errors may require changes in its package import definitions (**Imports** tab).

- Step 7 Continue with changes until the Program compiles successfully, and click **Save & Compile** and also **Save Bog**.
- Step 8 Repeat steps 4 through 7 for each Program object identified to have a compilation failure.
- Step 9 When finished, in the Nav tree right-click the station’s config.bog file, and ensure **Save** is dimmed. If available (not dimmed), click **Save**.

Considerations if a station functions as an oBIX server

With the EC-Net 4 addition of the new **AuthenticationService**, any oBIX request to a station must be authenticated using the “HTTP Basic Scheme” (HTTP Basic Authentication Scheme). If your station has an **ObixNetwork** that operates as an oBIX server, note that it is not possible for the EC-Net 4 migration tool to determine which station users were used for oBIX authentication.

Since the HTTP Basic Authentication Scheme is assigned on a per-user basis, the migration tool updates the authentication by completing the following steps:

1. Adds an HTTP Basic Authentication Scheme from the `baja` Palette to Services -> Authentication Service -> Authentication Schemes
2. Creates a new user (the default name is “obix”). If a User with the name “obix” already exists, the Migration Tool will try “obix1”, “obix2”, etc. until a free username is found.
3. Sets the Authentication Scheme Name of the User created in Step 2 to the HTTP Basic Authentication Scheme added in step 1.

NOTE: The password assigned to this user is the default password. It cannot be used by oBIX clients until you manually assign a password.

oBIX authentication configuration

After migration, you MUST perform the following additional configuration to allow OBIX clients the proper access to the station’s oBIX server. These requirements apply to the default “obix” User created during migration or any other User that you are using to allow oBIX clients access.

- Step 1 Open EC-Net 4 Pro, if not already open.
- Step 2 In the Nav tree of the migrated oBIX server station, expand **My File System** and navigate to the station folder.
Typically, such stations are under your **User Home**, under the stations folder.
- Step 3 Expand the station folder, and expand the config.bog file under it.
- Step 4 Expand **Services > UserService** and double-click on the newly created “obix” user to open the **Edit User** dialog box. The user that the migration tool created is a user name beginning with the letters “obix”, such as: “obix”, “obix1”, “obix2”, etc.
- Step 5 In the **Edit User** dialog box, make the following changes:
- change the user password from the existing default value to a password that complies with the EC-Net 4 password configuration requirements.
 - assign a Role to “obix” user that allows access to the necessary station components that you desire to be accessible to oBIX clients.
 - verify that the user permissions include read access to the ObixNetwork in order to provide access to the ObixNetwork in order to provide access to the oBIX Lobby of the server.
- Step 6 Click the **OK** button to close the dialog box and save the station changes (e.g. in the Nav tree, right-click **config.bog** and click **Save**).

Tips for Configuring oBIX Authentication

The following list includes tips and considerations for configuring oBIX stations after migration.

- Assign Role from a previous user

During migration, the EC-Net Role is created for each EC-Net^{AX} User in the station. If the EC-Net^{AX} station was being accessed by oBIX clients, you can probably pick the User that clients were using, and simply select the Role with that name as a starting point for EC-Net 4 oBIX client access.

- Separate Users for each Client

Consider using a separate User for each oBIX client (or client type) that accesses the station's oBIX server. This is another way to increase the security of your installation by encapsulating access and giving each user the minimum access necessary to accomplish the needed function.

Preliminary checkout of migrated stations

Prior to installing stations, we recommend that you use EC-Net 4 Pro to open each of the migrated stations offline, that is in the stations folder under your EC-Net 4 Pro **User Home**, and check it out carefully. If necessary, make and save any changes.

As for migrated EC-BOS stations, if you have an extra, available EC-BOS controller (of the same model type) that has been converted, you can install a migrated station and observe startup behavior. However, expect many errors from driver-sourced items—as you would for any unit disconnected from device networks.

Upgrade/convert platforms and install stations

Once you have stations migrated, you can upgrade/convert platforms to EC-Net 4 v4.0 and install the migrated stations. Typically you upgrade a Supervisor before upgrading any of its controllers.

Often, you use the same PC to run the migration tool, on the Supervisorstation archives, as well as the station archives of all controllers being migrated. When this is the case, you usually *do not choose* to “Install and start the (EC-Net 4 v4.0) platform daemon” when first installing the EC-Net 4 software. This lets the EC-Net^{AX}-3.8 platform daemon continue to host the Supervisor station while you use the migration tool. It also lets you become familiar with EC-Net 4 Pro, and yet still have EC-Net^{AX} Pro available for interim tasks.

NOTE: When installing EC-Net 4 software on the Supervisor PC, be sure to select the option “This instance of EC-Net 4 will be used as an installation tool”. Refer to the EC-Net 4 Installation Guide for details.

Alternatively, you may have used a *different* PC to run the migration tool, and perhaps even to (eventually) use as the new Supervisor host for the job.

As different combinations are possible in a transition such as this, use the procedures and steps provided to serve as a general guide.

Copying the Supervisor station

The first step to preparing the Supervisor station is to copy the migrated station.

Prerequisites: You have secured the required licenses and the station databases are ready to install.

Step 1 After installing EC-Net 4 on the Supervisor PC and starting EC-Net 4 Pro, open a platform connection to the EC-Net^{AX}-3.8 Supervisor station and stop the station.

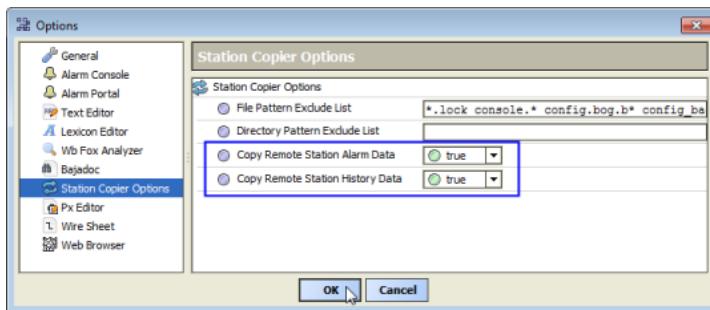
Step 2 At the EC-Net 4 v4.0 Supervisor PC (with access to EC-Net 4 v4.0-migrated station folders), install and start the EC-Net 4 v4.0 platform daemon.

For example, from the **Start** menu, select **All Programs** >EC-Net 4 4.0.3 → **Install Platform Daemon**

In some cases, this daemon may already be running; however re-installing causes no issues.

Step 3 Start EC-Net 4 Pro (if not already started) and click **Tools**→**Options**→ **Station Copier Options**.

The Options window opens.



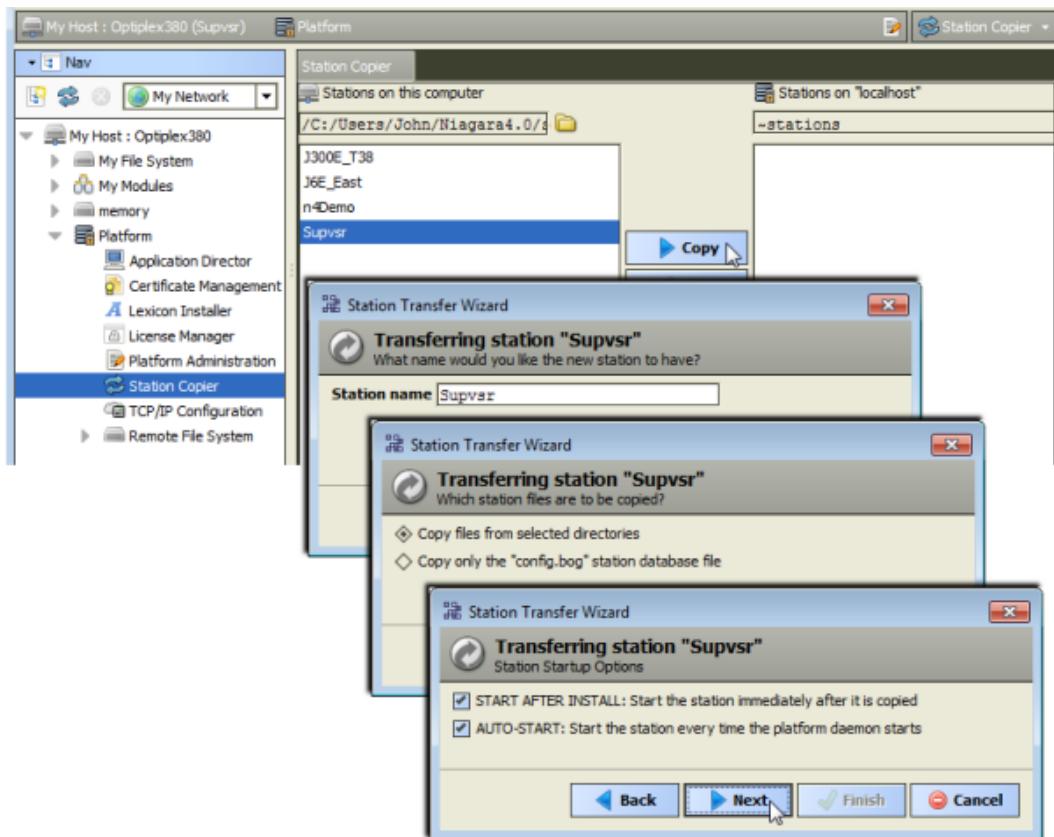
Step 4 Set **Copy Remote Station Alarm Data** and **Copy Remote Station History Data** to **true**, and click **OK**

Step 5 Open a local platform connection
(in the Nav tree, right-click **My Host**, and choose **Open Platform**).

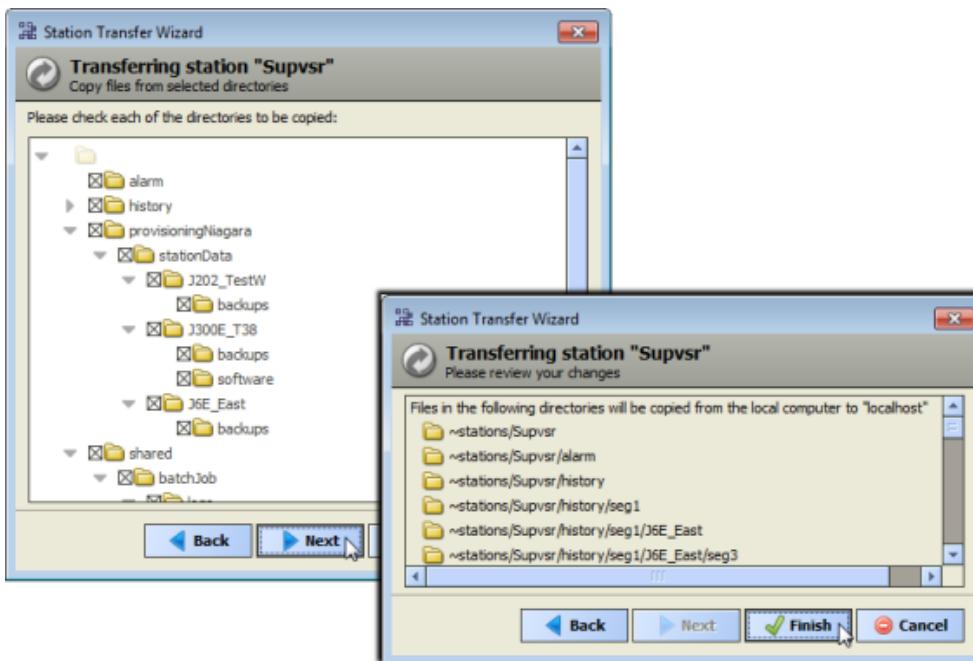
Step 6 After platform login, go to the **Station Copier** platform view.

Ready-to-install migrated stations are on the left side, in your EC-Net 4 Pro User Home, and there is *no station* installed on the right side, in the daemon user home (localhost). If your migrated station folders are not on the left side, copy them into your User Home. For example in: C:\Users\userName\Niagara4.x\distech\stations

Step 7 Select the migrated Supervisor station, and copy (install) to the daemon user home.



- Step 8 To transfer in this station copy, choose to **Copy files from selected directories**, and **START AFTER INSTALL...** and **AUTO-START...** and click **Next**. As shown below, select *all station folders*



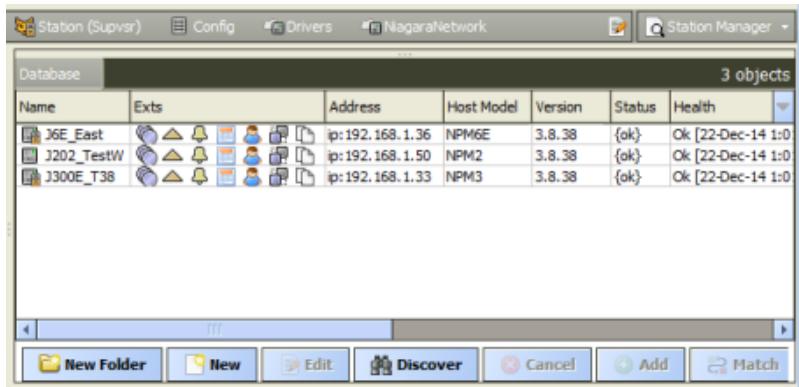
- Step 9 Select all station folders to be transferred and click **Next→Finish**.

Performing station cleanup

After copying the Supervisor take a moment to confirm that the copy was successful and make any needed changes to properties.

When the station is copied, open the **Application Director** and observe startup messages. Then open a local station (Fox or Foxs) connection to the Supervisor when it has started, and begin preliminary checkout.

Figure 17 Network Manager view



The figure above shows EC-Net 4 v4.0 Supervisor's **NiagaraNetwork** shortly after migration.

Stations that represent EC-BOS controllers in the Supervisor's **NiagaraNetwork** should be communicating, where all may still be at EC-Net^{AX}-3.8 (yet to be converted). For any that must remain at EC-Net^{AX}-3.8 (EC-BOS-2^{AX}s for example), review, and if necessary, adjust client connection properties.

Save any changes make before upgrading your EC-BOS controllers.

Controller conversion

Typically, after upgrading a Supervisor, you convert EC-BOS controllers to EC-Net 4 v4.0—or at least you convert all those for which you *successfully migrated a station*. During the EC-Net 4 commissioning of each EC-BOS, you install this station as one of the steps in the Commissioning Wizard.

CAUTION:

You must migrate the EC-Net^{AX} station from the EC-BOS and confirm that the migration was successful first (all Programs report {ok}, for example), *before installing* the .dist file.

Converting a controller

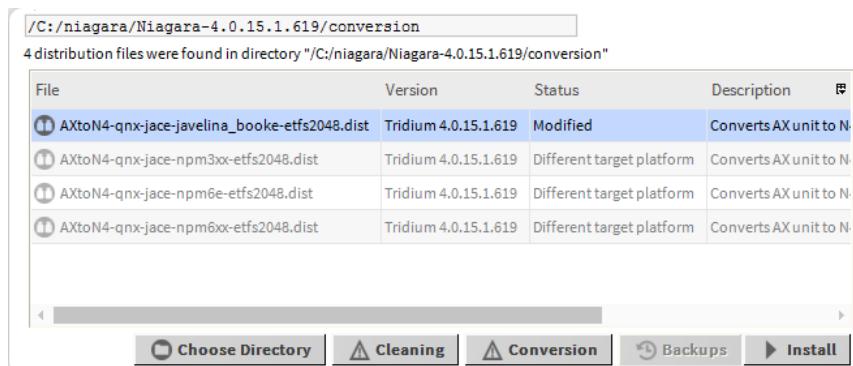
Perform the following procedure for each controller for which you have secured EC-Net 4 v4.0 license, and have successfully migrated its station database.

Prerequisites:

EC-Net 4 Pro is open, you have used it to check the migrated station for this controller, and confirmed that the migration was successful.

Step 1 Open a platform connection to the controller to be converted, and access the **Distribution File Installer**.

The list of distribution files opens.



NOTE: EC-Net 4 may display the `cleanDist` or other files for other controllers, but you can select only the file that is appropriate for your controller. This file is only for a unit already converted to EC-Net 4. If you do not see the appropriate `*.dist` file, verify that you are in the conversion folder. If necessary, use the buttons at the bottom of the view to navigate and browse for the file.

Point the installer to the `!/sw/N4buildNumber` folder, for example: `!/conversion/4.0.15.619`.

Only *one* of several listed should be selectable. This is an example of the file for converting a controller:

`aAXtoN4-qnx-jace-npm6e-etfs2048.dist`

Step 2 Select the appropriate `.dist` file and click the **Install** button.

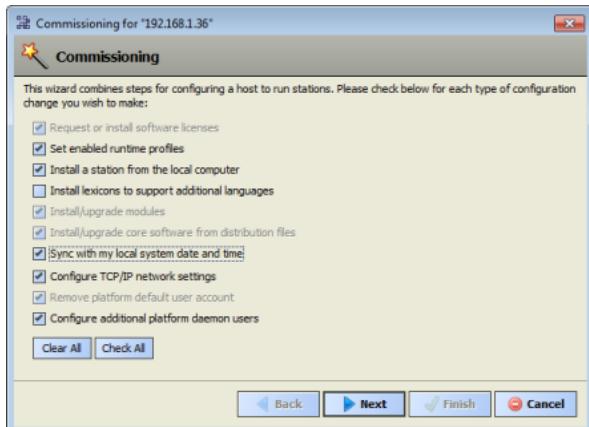
NOTE: Installation clears *all data* in the controller, including EC-Net^{AX} licenses and certificates, environment and properties files, SSL certificates and key files, along with all station data and EC-Net^{AX} software. The controller's IP configuration is retained. You must use the default platform credentials to regain access.

Step 3 After this `.dist` file installs and the controller reboots, re-open a platform connection (port 3011) using the **factory default** credentials.

Step 4 If you previously exported SSL certificate stores from this controller, access the platform **Certificate Manager** view, and do the following:

- On the **User Trust Store** tab, click **Import**, and navigate to select the previously saved public key certificate (typically common among all platforms at the job).
- On the **User Key Store** tab, click **Import**, and navigate to select the previously saved private key certificate unique to this platform.
- Access the platform **Platform Administration** view, click the **Change SSL Settings** button, and select the new server certificate you just imported.
- Run the platform **Commissioning Wizard** (right-click the **Platform** node, and select **Commissioning Wizard**).

In this initial commissioning of the controller, you typically use *all default step selections*. This includes most steps except **Install lexicons** (but if you use text-based lexicons, select this too).



In most cases you select to:

- Install the licenses from the licensing server (unless you already have these as files).
- Use runtime profiles RUNTIME and UX, or else RUNTIME, UX, and WB.
- Install the specific station that you migrated for this *controller*, choosing to:

Copy files from selected directories.

NOTE:

You typically *select all directories* shown in the migrated station, including the alarm and history subfolder (to migrate prior alarms and histories). Apart from the migration, this is an unusual practice. If you are installing the same station in different/multiple platforms, be sure to deselect (clear) the alarm and history subfolders.

- Install all pre-selected software modules required for this station.
- Install all listed core software distribution files.
- Review, and if necessary change, the controller's TCP/IP configuration.
- You must add at least *one new* platform admin user in the "Remove platform default user account" step (initial **Platform Daemon Authentication** dialog). This *replaces* the factory-default platform user. Note that in EC-Net 4, platform credentials require a stronger password than in EC-Net^{AX}. The password must be at least 8 characters, with at least 1 digit (numeral). If the password does not meet this requirement, an error dialog prompts you to create a valid one.

Make note of these platform credentials; you should use a login as such a platform user upon next access.

NOTE:

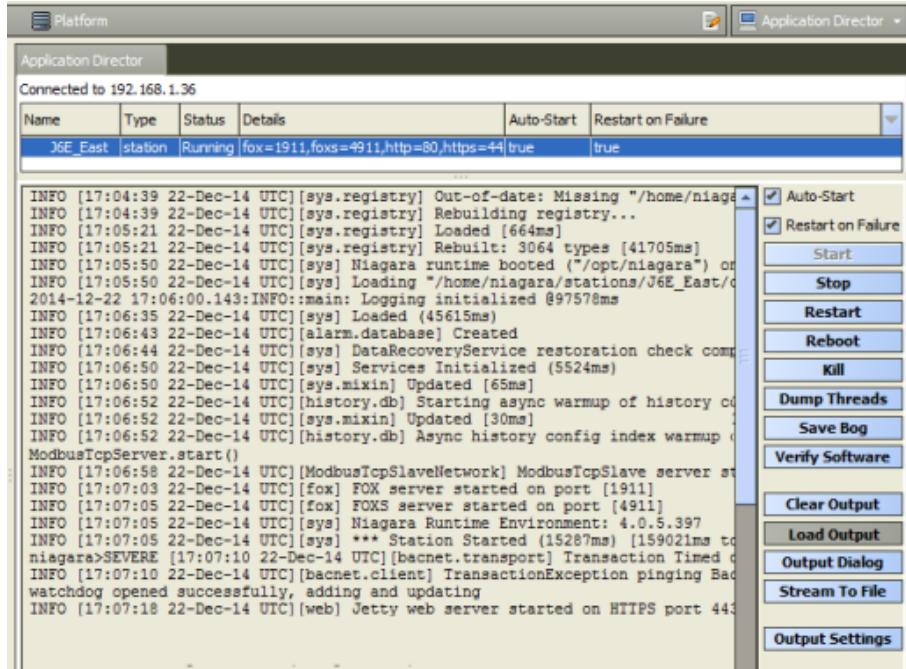
For this platform session, EC-Net 4 Pro remembers the first platform admin user you created to replace the factory default platform user. This simplifies re-connection after the controller reboots.

For related details, see the EC-BOS Install and Startup Guide for EC-Net 4. Example screen captures of a few of these **Commissioning Wizard** steps are shown in the [Controller conversion examples, page 54](#)

Step 5 Review and finish the commissioning wizard, to install all selected items and reboot the controller. After several minutes, the conversion completes, and platform access should be possible.

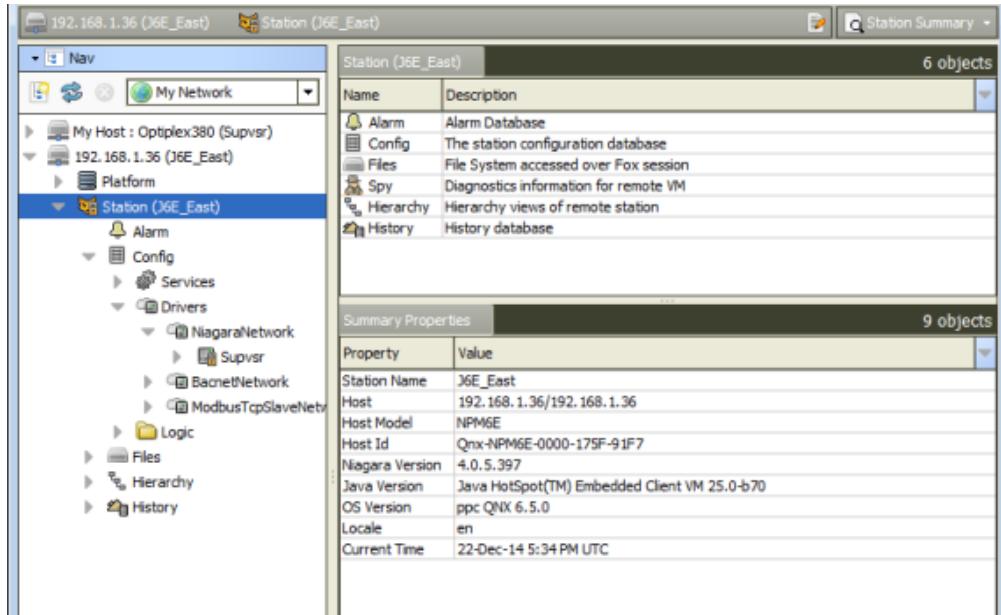
CAUTION: Do not remove power from the controller during this reboot, which may take up to 7 or more minutes to complete. Removing power could easily cause the unit to become *unrecoverable*. If desired (and convenient), you can use a serial shell connection to the controller to monitor progress as files are installed, the platform is rebooted, and the unit is prepared.

Step 6 Re-open a platform connection to the converted controller, using a *new platform account* you entered in the wizard. Go to the **Application Director** and view the station standard output.



Again, there may be warnings and messages, but its station status should change to “running”.

- Step 7** In EC-Net 4 Pro, open a station connection (Fox or Foxs) to the converted controller, typically using the admin (superuser) login. Verify the station opens.



- Step 8** For each remainin controller to be converted, repeat the above steps

When done with this, all converted controllers should be running stations as EC-Net 4 v4.0 platforms, as well as the Supervisor station.

Controller conversion examples

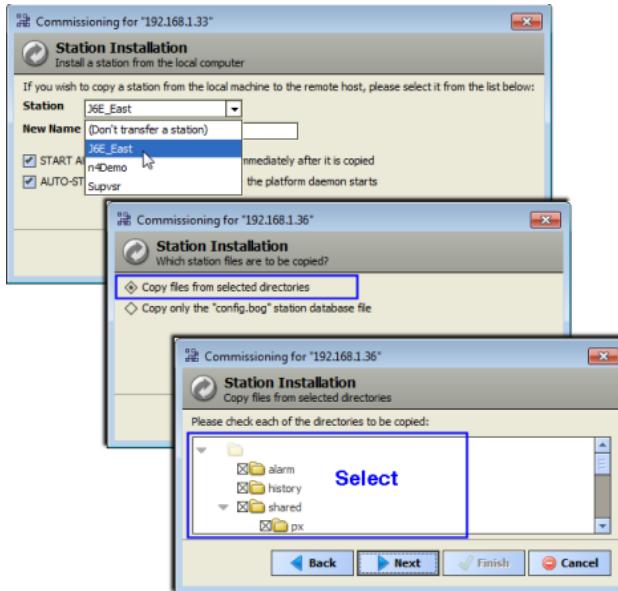
Following, are example Commissioning Wizard windows when converting EC-BOS controller to EC-Net 4.

Example conversion steps

Figure 18 Example “Set enable runtime profiles” step

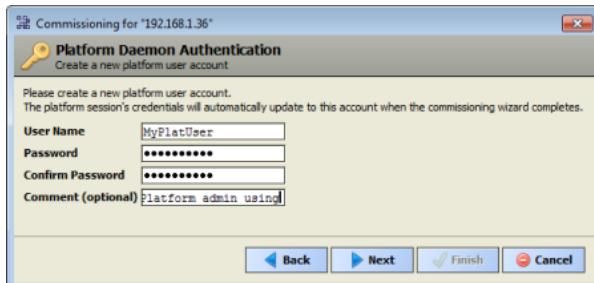


Figure 19 Example “Install station” step (pick migrated station)



As shown above, by default all items are selected. If necessary you can de-select any items not wanted.

Figure 20 Example platform daemon authentication step (to replace factory platform user)



If this controller was previously provisioned by the AX Supervisor, you may wish to have at least one platform admin user with the *prior* user name, and possibly prior password (if it was “strong” before). Otherwise, you must update the platform connection Credentials property for this station back in the Supervisor station’s NiagaraNetwork.

An even better practice might be to add another platform user in the EC-BOS that *no person uses* to make platform connections, but is only referenced back in the EC-Net 4 Supervisor.

Downgrading controllers

In some cases it may be desirable or necessary to convert a controller from EC-Net 4 to EC-Net^{AX}. During the commissioning of EC-BOS controller, all station data are deleted. This includes SSL certificates and key files. EC-Net 4 stations cannot be converted to EC-Net^{AX} stations. But, you can install a previously saved EC-Net^{AX} station during the downgrade commissioning process as one of the steps in the Commissioning Wizard.

NOTE: Controller downgrade from EC-Net 4 removes all data except the controller's Ethernet TCP/IP configuration. Platform credentials revert to factory defaults. The EC-Net 4 license is not retained and EC-Net^{AX} license is required. All station data is deleted.

Downgrade procedure

Perform the following for each controller to be downgraded, and for which you have secured EC-Net^{AX} license.

Prerequisites: To restore an existing station, make sure you have access to the station file, or a backup distribution file.

- Step 1 At the Supervisor, using EC-Net 4 Pro, open a platform connection to the controller to be downgraded, then open the **Application Director** and stop any running station.
- Step 2 With the platform still selected, choose the **Distribution File Installer** from the view selector, navigate to the location of the downgrade distribution file, and click **Conversion**.

NOTE: Some non-applicable files, such as EC-Net 4 `cleanDist`, `.dist` or other files may be selectable but, in this case, do not select these files. Click the **Choose Directory** button to browse to the location of the downgrade `.dist` file, for example: `N4toAX-qnx-jace-<npm-model>etfs2048-clean.dist`.

- Step 3 Select the appropriate `.dist` file and click the **Install** button.

NOTE: Installation clears *all data* in the controller, including EC-Net^{AX} licenses and certificates, environment and properties files, SSL certificates and key files, along with all station data and software. The controller's IP configuration is retained. You must use the default platform credentials to regain access.
- Step 4 After this `.dist` file installs and the controller reboots, re-open a platform connection using EC-Net 4 Pro (port 3011) and the *factory default* credentials.

NOTE: At this point, the EC-BOS will be at 3.8.38 version. If you need an earlier release EC-Net^{AX} version, use the `clean` `.dist` files in the EC-Net^{AX}-3.8 **Distribution File Installer**. This will revert the EC-BOS back to the base-factory version.
- Step 5 Run the platform **Commissioning Wizard** (right-click the **Platform** node, and select **Commissioning Wizard**). During the commissioning process you can choose to reinstall a station, if you have one ready, perhaps from a previous backup, or you can restore from a backup as described in the next step.

CAUTION: Do not remove power from the controller during this reboot, which may take up to 7 or more minutes. Removing power could cause the unit to become *unrecoverable*. If desired (and convenient), you can use a serial shell connection to the controller to monitor progress as files are installed, the platform is rebooted, and the unit is prepared.

For more details on commissioning the EC-Net^{AX}-3.8 controller, see the *EC-BOS Install and Startup Guide for EC-Net 4*.
- Step 6 If you did not restore a station with the previous commissioning step, make sure you have a backup `.dist` file and a platform connection to the rebooted EC-Net^{AX}-3.8 controller. Select the **Distribution File Installer** view and click on the **Backups** button or **Choose Directory** button to navigate to the appropriate dist file location.
- Step 7 Select the `.dist` file and click the **Install** button. Follow the prompts in the **Distribution File Installer** to complete the installation.

Chapter 3 System verification

Topics covered in this chapter

- ◆ Non-default environment files
- ◆ Verify station-to-station communications
- ◆ Provisioning notes
- ◆ Station User notes in migrated stations

Following the migration and initial upgrade of a system from EC-Net^{AX}-3.8, there may be additional considerations and tasks that you need to verify operation.

The following sections include related topics.

Non-default environment files

The migration tool *does not* migrate all parts of the station backup `.dist` file. For any EC-Net^{AX}-3.8 host where you made changes in properties files or xml configuration files, these settings may need to be re-applied to equivalent files in the EC-Net 4 host.

Some examples of files that may have been modified include:

- `system.properties`
- `nre.properties`
- `daemon.properties`
- `units.xml`
- `unitConversion.xml`
- `facetKeys.properties`

In most cases the station should at least run successfully, and these settings can be re-applied once the station is running under EC-Net 4. It may not even be necessary to make the same changes, so you should evaluate once you have the station running in EC-Net 4.

NOTE: Do not copy entire files from the EC-Net^{AX} platform to the EC-Net 4 platform, but instead only those values that were modified from defaults in the equivalent EC-Net^{AX} file(s).

Verify station-to-station communications

Ideally, station-to-station communications between the platforms converted to EC-Net 4 v4.0 and all others resume after the upgrade. That is, communications between the Supervisor and controller stations, via the **NiagaraNetwork**. Supervisor communications between any EC-BOS stations which remain at EC-Net^{AX}-3.8 should also resume. (Note this assumes all station names remained unchanged.)

However, in some cases you may need to take additional steps to restore communications, working in the EC-Net 4 v4.0 Supervisor stations and/or controller stations (EC-Net 4 v4.0 EC-BOSs, and if applicable, EC-Net^{AX}-3.8 too).

NOTE: If configuration of any EC-Net^{AX}-3.8 EC-BOS station is required, say to edit “client connection” properties of the NiagaraStation that represents the EC-Net 4 v4.0 Supervisor, need to use EC-Net^{AX} Pro for this.

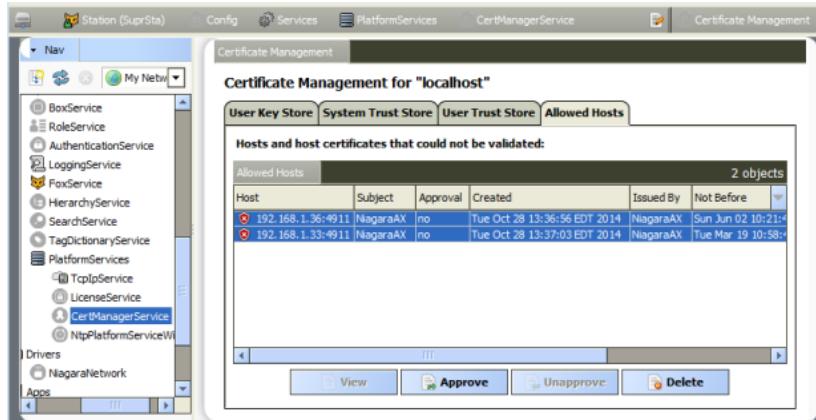
Some variation of the following sequence may be required:

1. In EC-Net 4 Pro, re-open the Supervisor station and go to the **Station Manager** view of its **NiagaraNetwork**.

If new EC-BOS controllers were added (not migrated) there will not be an existing station child for them—you need to select and add them.

2. For each NiagaraStation in the Supervisor station's **NiagaraNetwork**, verify that communications report {ok}. The status of all stations in the network should be {ok}.

In cases where secure communication (Foxs) is being used with the default self-signed certificates, it may be necessary to go to the Supervisor's **CertManagerService** platform service, and perform other actions.



The figure above shows allowed hosts exceptions for migrated controllers that need approval. This may not be necessary, if previously-exported TLS/SSL certificates were imported before commissioning.

3. In a station connection to each EC-BOS controller, go to the **Station Manager** view of its **NiagaraNetwork**, and verify communications to the Supervisor are {ok}.

Again, if Foxs (TLS/SSL) is being used by a Hotspot EC-BOS station back to the Supervisor (and possibly other EC-BOS stations) you may need to go to the EC-BOS station's **CertManagerService** platform service, and perform other actions.

4. While working in remote EC-BOS stations, after verifying **NiagaraNetwork** communications back to the Supervisor are {ok}, verify other "inter-network" configuration is valid. For example, verify the correct **NiagaraStation** component (representing the Supervisor) is referenced in the appropriate **StationRecipient** component, under the station's **AlarmService**. Typically, this is necessary only if station names were changed when migrating.

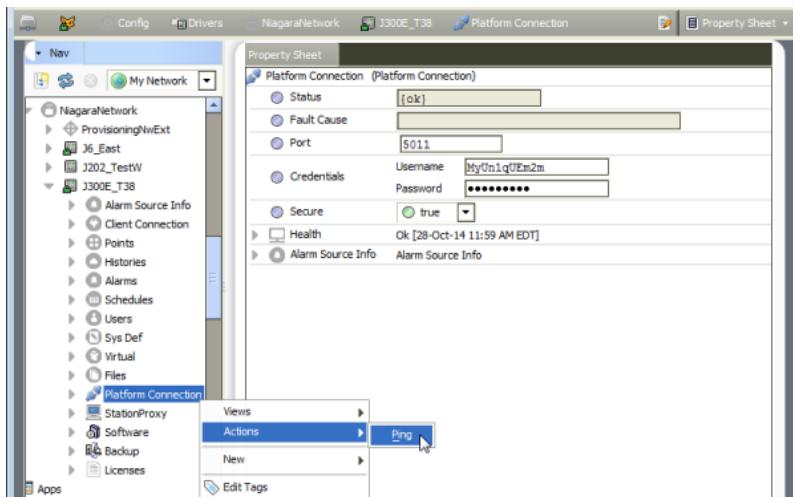
Provisioning notes

For any Supervisor migrated to EC-Net 4 v4.0 that was configured for provisioning stations in its **NiagaraNetwork**, that is, using the `provisioningNiagara` components (for example, **ProvisioningNwExt**, also **BatchJobService**), there may be related post-migration measures needed.

Verify Supervisor to controller platform daemon communications

You need to verify that communications from the Supervisor to the platform daemon of each controller are {ok}.

Expand the Supervisor's **NiagaraNetwork**, and issuing a ping from the **PlatformConnection** device extension of each child **NiagaraStation**.



The figure above shows successful communications to the EC-BOS platform daemon, using platformssl. However, you may first need to edit the **Credentials** property to match a new strong password for the platform admin account in an migrated controller. And/or, perhaps (again) access the Supervisor's **CertManagerService** platform service, and perform other actions.

This station-to-platform provisioning communications is one-way only, from the Supervisor to each controller. Therefore, no reciprocal configuration in controller stations is needed.

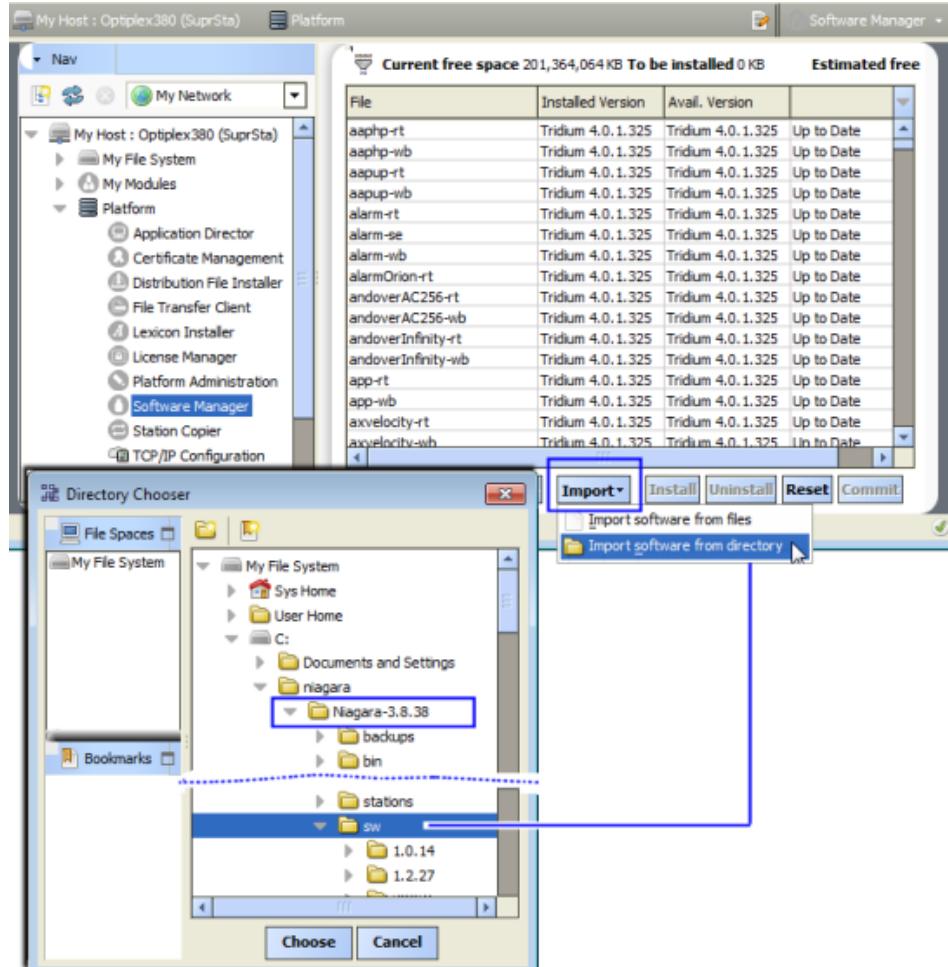
Provisioning considerations when EC-BOS controllers remain

Sometimes one or more EC-BOS controllers may need to remain running EC-Net^{AX}-3.8, yet continue to be in the migrated Supervisor's **NiagaraNetwork**. This is supported by the EC-Net 4 v4.0 Supervisor, including *many* provisioning operations. For example, the backup of EC-Net^{AX}-3.8 stations and installation of EC-Net^{AX}-3.8 software.

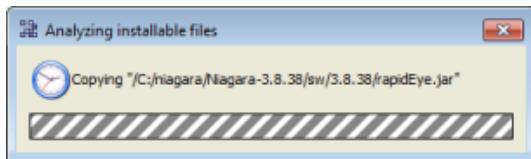
For proper support, you should *import* the software database of the former EC-Net^{AX}-3.8 Supervisor into the software database of the EC-Net 4 v4.0 Supervisor. This copies over all the versioned EC-Net^{AX} software modules and .dist files to the new Supervisor, and makes them available for provisioning of the older platforms.

To do this when the EC-Net 4 v4.0 Supervisor is installed on the same PC as the former EC-Net^{AX}-3.8 Supervisor:

1. Use EC-Net 4 Pro to open a *platform connection* to the EC-Net 4 v4.0 Supervisor, then access the **Software Manager** view.
2. Click the **Import** button at the bottom, and then Import software from directory.



- As shown above, in the **Directory Chooser** popup, navigate to the EC-Net^{AX}-3.8 Supervisor's installation folder, then expand to find its `sw` subfolder. Select it and click **Choose**.



Progress is displayed in a popup dialog similar to the one above, while software files are analyzed for versions and then copied into appropriate subfolders in the EC-Net 4 v4.0 Supervisor's software database. When finished copying, the normal **Software Manager** view opens.

If the former EC-Net^{AX}-3.8 Supervisor is on a *different* PC than the EC-Net 4 v4.0 Supervisor, you perform the same procedure. However, first copy the entire software database (`!\sw` folder) from the EC-Net^{AX}-3.8 Supervisor to a location accessible from the EC-Net 4 v4.0 Supervisor, such as on a USB flash drive, etc. Then navigate to that location from the **Software Manager** while platform-connected to the EC-Net 4 v4.0 Supervisor.

Station User notes in migrated stations

A number of EC-Net 4 changes were made affecting station users, meaning User components that are children of a station's **UserService**.

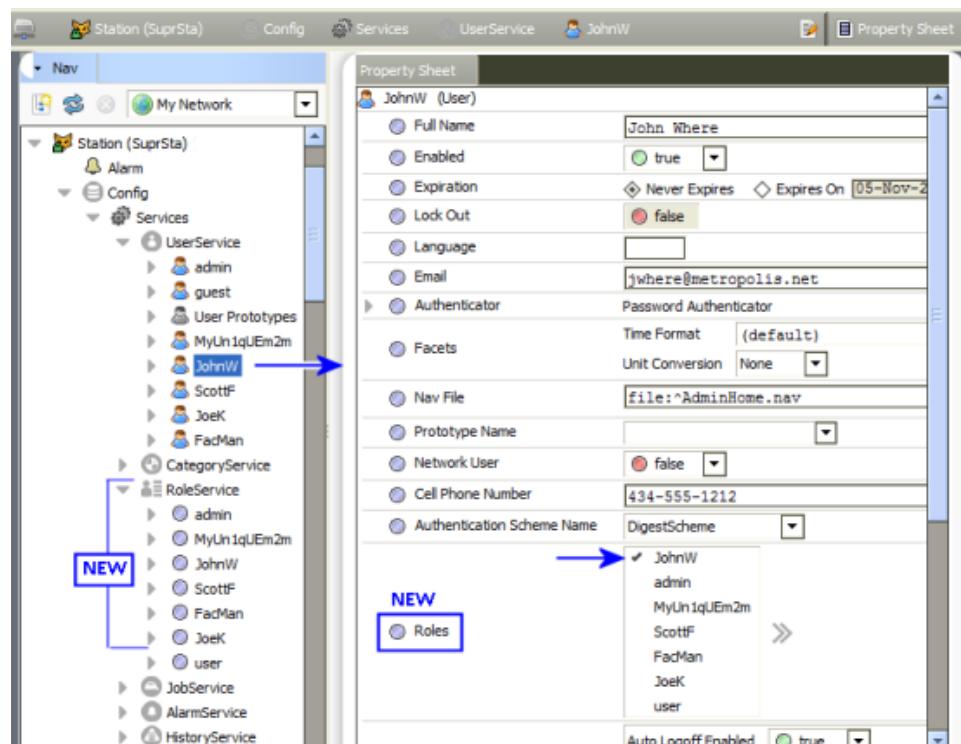
Permissions moved to Roles

The permissions map that every User had (Permissions property) was *removed*. The permissions map identifies what operator-level and admin-level privileges are in place for different *Category* components in the station.

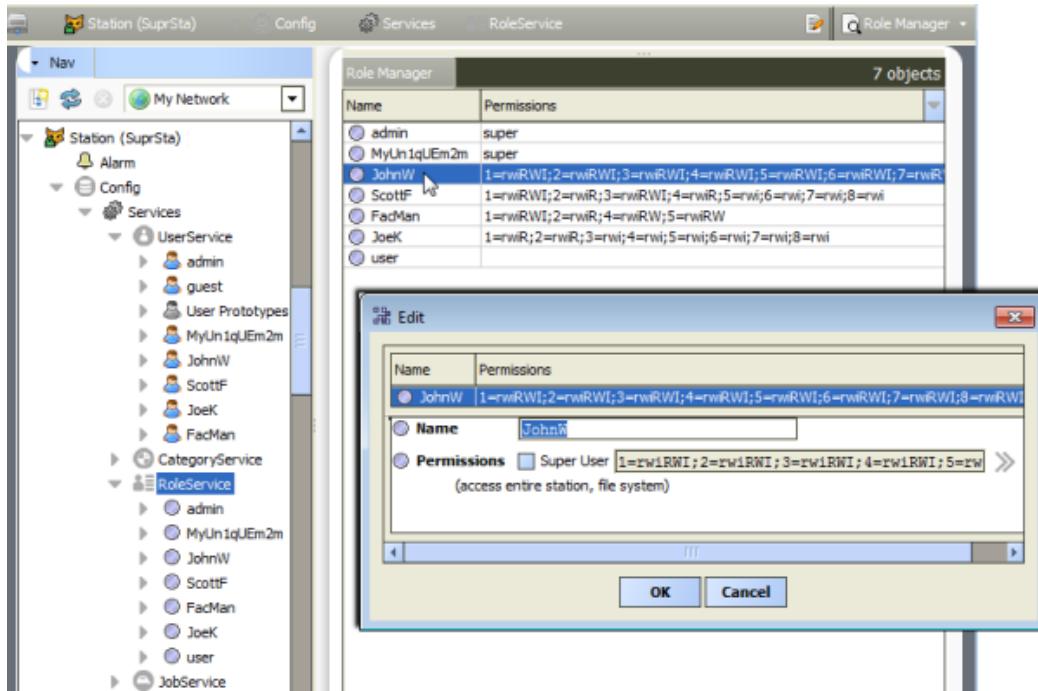
The Permissions property was *relocated* to **Role** components (*new* in EC-Net 4), which are children of a *new* **RoleService** in the station. Instead of a User being assigned permissions directly, now you assign each user to one or more *Roles*. Permissions of those roles are “OR’ed” together to specify what privileges a user has on the categories in the station.

In order to migrate Users from EC-Net^{AX} to EC-Net 4 v4.0, while preserving their existing category permissions, note an EC-Net 4 migration results in a *separate Role created for each User—and named identically to that User*.

Figure 21 Station migrated to EC-Net 4 v4.0 has default “one-to-one” mapping of new Roles to existing Users



This one-to-one mapping of created roles to users is shown above, where user **JohnW** has (by default), only *one* role assigned: also named **JohnW**. (Note additional roles *could* be assigned, but are not here.) Effectively, nothing changed here: the permissions map that used to be in user **JohnW**’s permissions property is now in the “same named” **Role** component.

Figure 22 Permissions map of User relocated to Role created with identical name

As shown above, you can see this in the popup **Edit** window for each Role (in the **Role Manager** view on the **RoleService**). The role now has the permissions map.

In summary, to make better use of roles, you should consider this a workaround effect of migration. For example, you could create more duty specific roles and/or rename or delete roles, later reassigning to users in various combinations. When adding or editing users, this can simplify how permissions are assigned.

The **Role Manager** lets you do all these things, except the assignment of one or more roles to users. That you do from editing (or adding new) users in the **User Manager**, or from User property sheets.

Authentication is user-specific

When looking at a **User** in EC-Net 4 station, notice that the **Password** property is now inside a new **Authenticator** container, along with **Password Config** properties. In addition, each User has a specified **Authentication Scheme Name**, where the default is typically **DigestScheme**.

This is part of a new authentication architecture in EC-Net 4. Along with a new (and required) **Authentication Service** and child **AuthenticationScheme** components, station security is improved, providing finer control. For the migration of existing station users, typically nothing remains to be adjusted. However, when creating new users, this architecture may provide added flexibility.

For related details on authentication changes, refer to the Station Security Guide.

Chapter 4 Migrating EC-Net Access stations

Topics covered in this chapter

- ◆ Upgrading devDriver-based video drivers
- ◆ Mapping Supervisor cameras to EC-Net^{AX} controllers before migration to EC-Net 4
- ◆ Preparing the BackupService
- ◆ Backing up by creating a distribution file
- ◆ Migrating each station
- ◆ Copying the migrated files
- ◆ Setting up the Supervisor's database password
- ◆ Creating and updating the HSQL database password
- ◆ Configuring network device passwords
- ◆ Deleting the old certificates
- ◆ Reconnecting the Supervisor and remote stations
- ◆ Restoring the BackupService schedules

EC-Net Access stations are EC-Net stations that have some additional unique migration requirements and options. The following topics describe the specific steps required to migrate your EC-Net^{AX}-3.8 security Supervisor and controller stations to EC-Net 4 v4.8 (and later) security stations. This topic provides an overview of the process.

To reduce downtime, migrate the Supervisor station first and then the individual security stations.

A typical Supervisor station includes persons, badges, access rights, schedules, threat level groups, intrusion PIN, intrusion zones, NiagaraIntegration ID, tenants, new users, new roles, new categories, a PhotoIDNetwork, and ObixNetwork, PxViews, additional personnel information, an Ldap network, alarm class, custom Wiegand format, and keypad configuration.

The following list summarizes the aspects of the EC-Net^{AX} security system that need special migration attention:

- **Passphrase:** Be sure to have the station passphrase available before starting the migration process. Depending on the migration scenario you are using, you may need to re-enter an existing passphrase or create a new passphrase during the migration process. Refer to “System Passphrase” and related topics in the *EC-Net 4 Platform Guide* for more details about passphrase usage.
- **devDriver based video drivers:** EC-Net 4 does not support RapidEye and Dedicated Micros legacy devices. During migration, the migrator deletes devDriver-based devices, including these two. If your installation includes one or more of these devices, please upgrade to the supported devices: Axis, Milestone and Max-Pro video drivers.
If your installation uses the devDrivers for AxisvideoNetwork and Milestone, a procedure in this chapter documents how to use the **Video Driver Upgrade tool** to upgrade these drivers to nVideo-based version of the drivers.
- **Remote Video Camera mapping:** As part of EC-Net^{AX} to EC-Net 4 compatibility restrictions, Supervisor cameras cannot be discovered under the NiagaraNetwork from an EC-Net^{AX} controller. So, to migrate cameras to the new Supervisor station, map them in a EC-Net^{AX}-2.3 Security EC-BOS before doing the station migration. Then, during migration, the new station retains that mapping without additional changes after migration.
- **Backup:** Before migration, you will make a backup of the Supervisor station and all subordinate stations. By default, the BackupService does not include alarms and histories. To retain these database records, you must specifically include alarms and histories in the backup.
- **Backup schedules:** Before migration you must use the web UI to unassign any schedules in the BackupService and re-assign them after completing the migration.

- **Photo ID station AsureID Legacy Device:** If you use AsureID to create photo ID badges, EC-Net 4 v4.8 (and later) does not support the AsureID Legacy Device. Before migrating, add the newer AsureID Client Device to your Photo ID network.
Refer to the “Manually adding a Photo ID device (not legacy)” topic in the *EC-Net Access Installation and Maintenance Guide* for guidance on how to configure the newer Asure ID Client Device.
- **Supervisor database support:** EC-Net 4 v4.8 (and later) supports MySQL 5.6 (and later) and SQL Server 2008 (and later). These are the minimum requirements for the Supervisor database.
NOTE: There is no special procedure for preparing the Supervisor database. If your database is older than the supported versions, contact the database supplier for software upgrade procedures.
- **Remote station database password:** The migrator clears the Hsql password in each remote station. After migration it sets up the default password to access the Hsql database. While the database will work with the default password, you should change it to a more secure credential after migration.
- **Roles:** Roles are handled differently in EC-Net 4. The migrator removes all EC-Net^{AX} security roles from the UserService and moves them to the EC-Net 4 Role Service.
- **Custom modules:** If your installation includes custom modules made using the ProgramModule feature, you will need to refactor them before migrating the Supervisor and subordinate stations. Refer to “Migrating modules made with the ProgramModule component” in this guide.

Upgrading devDriver-based video drivers

If you are using one or more devDriver-based video drivers, a conversion tool is available to upgrade these drivers in each controller station to the newer nVideo-based drivers. You must upgrade these drivers before you migrate each controller station.

Prerequisites: Your laptop or PC is connected to the same network as your controller(s) and is running the EC-Net^{AX} version of EC-Net 4 Pro. The driver upgrade module is available.

- Step 1 From EC-Net^{AX} Pro, open a connection to the desired controller station (**File**→**Station**→**Open Station**) and enter credentials with admin privileges.
- Step 2 From the main menu, click **Tools**→**Driver Upgrade Tool**.

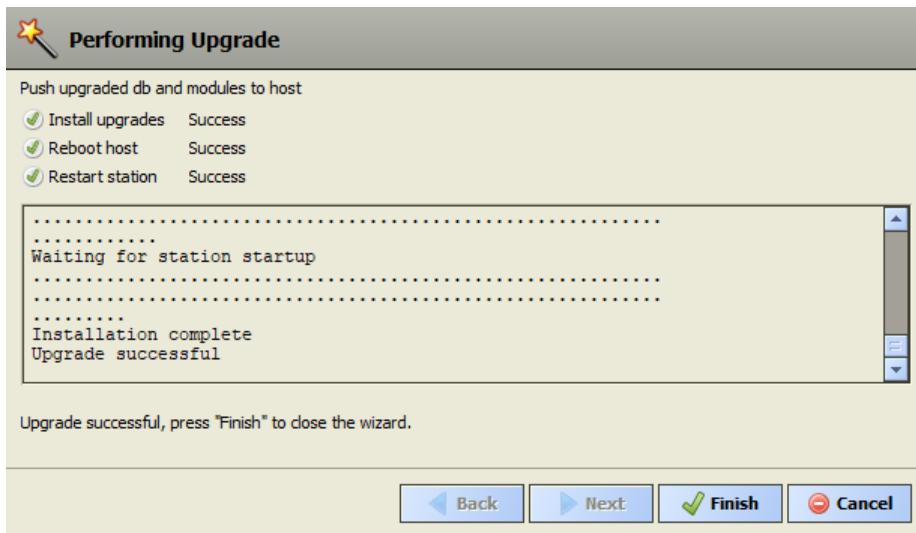
The **Driver Upgrade Tool** window opens.



- Step 3 Enter the host address, platform and station addresses and click **Next**.

The wizard reads the current station and scans its .bog (station) file. If a driver can be upgraded, the wizard changes the .bog file and downloads any dependent modules. Be sure that you have the required modules or an upgrade is not possible.

When it successfully completes the upgrade, the wizard indicates that it was successful.

**NOTE:**

The wizard does not delete the old devDriver based video driver modules.

- Step 4** To conserve space in the controller, delete the old devDriver based modules if they are no longer needed. devDriver modules that you can delete, include the following:
- devDriver.jar
 - devHttpDriver.jar
 - devIpDriver.jar
 - devVideoDriver.jar

NOTE: The devDriver.jar may be required as a dependency for other drivers. If so, you cannot delete it while that dependency exists.

- Step 5** Delete any older driver modules, such as those for Closed-Circuit TV (CCTV) equipment associated with the older drivers.

Mapping Supervisor cameras to EC-Net^{AX} controllers before migration to EC-Net 4

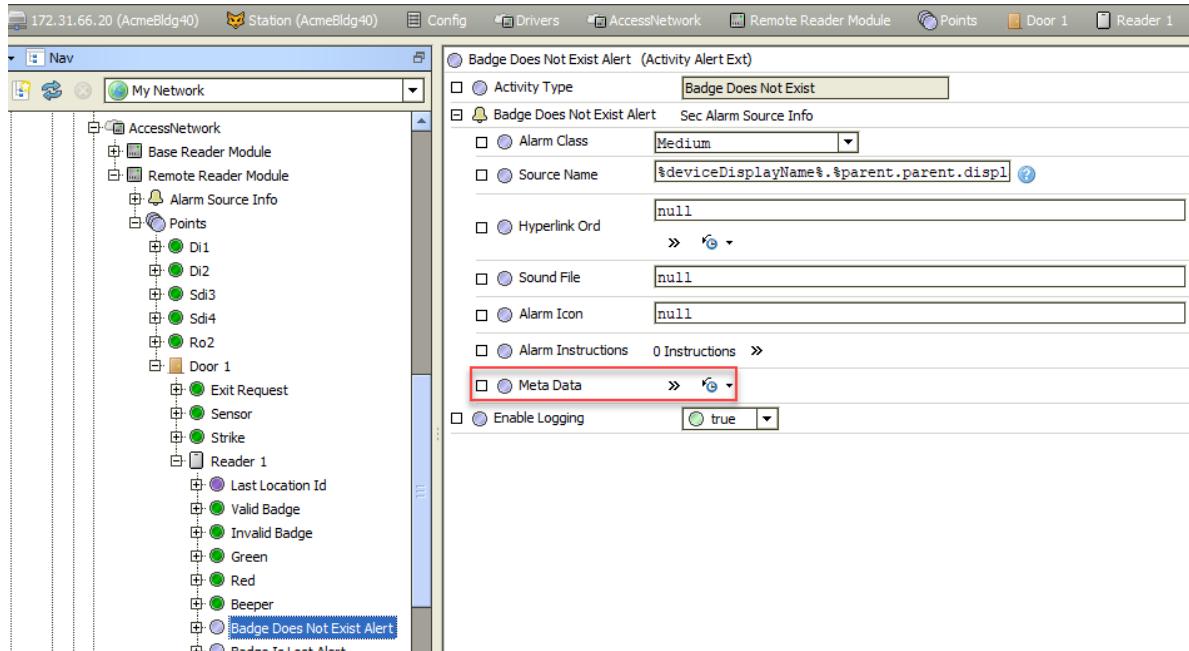
As part of EC-Net^{AX} to EC-Net 4 compatibility restrictions, Supervisor cameras cannot be discovered under the NiagaraNetwork from an EC-Net^{AX} controller. This procedure describes a way to pre-configure your station before migration in order to retain the Supervisor camera to remote controller connection after migration.

Prerequisites: The camera device address is available. The Supervisor IP address is known.

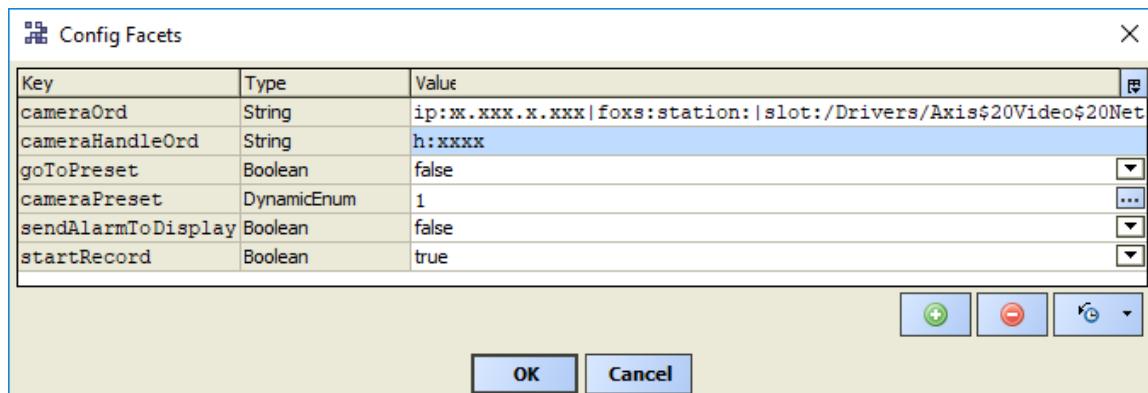
If you map the desired Supervisor camera to the remoter controller before migration (using this procedure), the system maintains the camera relationship through the migration process and functions after migration for use with readers and alarms.

- Step 1** Using the EC-Net^{AX}-2.3 Security EC-Net^{AX} Pro, navigate to the following location: **Config→Drivers→AccessNetwork→Remote Reader Module→Points→Door 1→Reader 1→Badge Does Not Exist Alert**

The property sheet view opens.



Step 2 To open the **ConfigFacets** window, click the facets icon (>>) next to the **Meta Data** property.



Step 3 Add facets as shown in the image above (also, refer to details in table below), where xxx.xxx.x.xxx is the IP address of the Supervisor platform and xxxx is the actual **cameraHandleOrd** for the camera you are mapping.

The following table indicates the key names, type, and values that go in the facets window.

Key	Type	Value
cameraOrd	String	ip:xxx.xxx.x.xxx foxs:station: slot:/Drivers/Axis\$20Video\$20Net
cameraHandleOrd	String	h:xxxx
goToPreset	Boolean	false
cameraPreset	dynamicEnum	1
sendAlarmToDisplay	Boolean	false
startRecording	Boolean	true

NOTE: The **startRecording** property works only for **Milestone** cameras but not for **Axis Video** cameras. Set the **startRecording** property to **false** for an **AxisVideoCamera**.

When this task is completed the station is ready for backup and migration as described in the tasks included with [Chapter 4 Migrating EC-Net Access stations, page 63](#) topic.

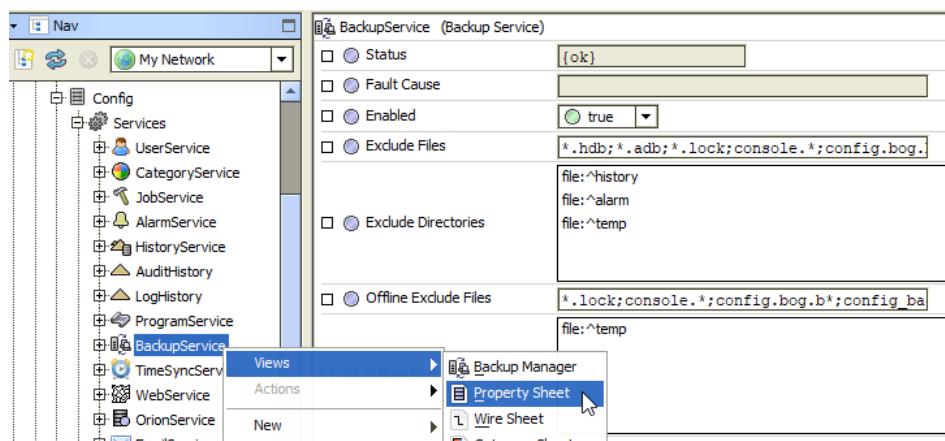
Preparing the BackupService

By default, backups do not include alarm and history records. To preserve alarms and histories you need to edit the excluded file values.

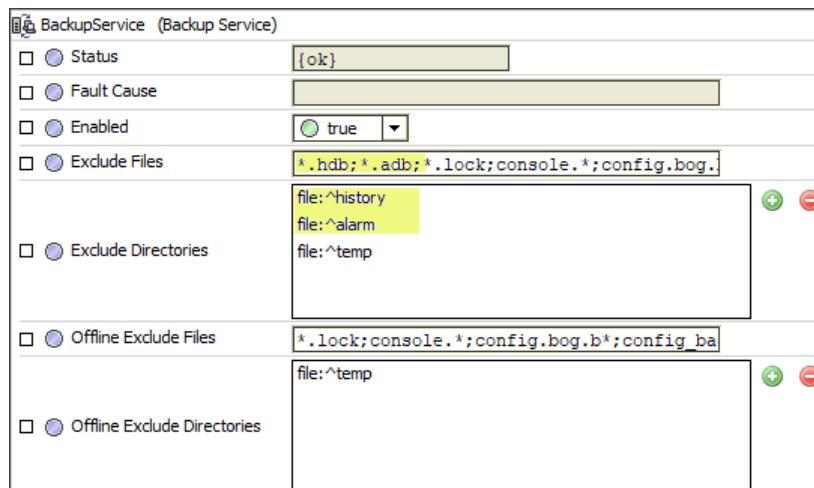
Prerequisites: You are logged in to your EC-Net^{AX} Supervisor station with admin privileges.

- Step 1 Navigate to the **Config→Services** node in the Nav tree, right-click the **BackupServices** node and select **AX Property Sheet** from the popup menu.

The **BackupServices** property sheet opens.



- Step 2 In the **Exclude Files** text field, select and delete the following: *** .hdb;** and *** .adb;** using your keyboard and mouse.



This includes (not excludes) your database files.

- Step 3 For the **Exclude Directories** property, select the following files and delete them using the remove icon (⊖): **file:^history** and **file:^alarm**

This includes (not excludes) alarm and history files.

Backing up by creating a distribution file

This procedure backs up a Supervisor and any subordinate stations. Creating a single distribution (.dist) file is the most complete way to back up a Supervisor station and its subordinate remote stations. The backup file includes all station data and module-dependency information.

Prerequisites: Your PC is connected to the network with any remote controller subordinate stations. You are using a browser (the web UI) to connect to the Supervisor station.

Step 1 Open a browser and log in to the EC-Net^{AX} Supervisor station that you want to back up.

Step 2 From the Home view, click **System Setup** → **Backups**.

The **Backups** view opens.

File Name	Timestamp	Backup Type
file:~backups/backup_entSecSupTest_01_181212_1349.dist	12-Dec-18 1:49 PM EST	Local
file:~backups/backup_entsec_181212_1346.zip	12-Dec-18 1:47 PM EST	System

Step 3 Click the **System Backup** button.

The **System Backup** window opens with two option boxes selected.



Selecting the first option (checkbox) enables a backup of the Supervisor station and all its subordinate stations. The second option (checkbox), when selected, automatically downloads the backup to your local **Downloads** folder.

Step 4 To continue, click **Ok**.

The **Job** window displays a progress bar while the backup job runs. When the job finishes, another **Local Backup** window opens with a “Show Details” link and a **Download** button.



Step 5 To display the history of this backup job, click the **Show Details** link.

Step 6 To save the backup *.dist file to a designated location, click the **Download** button. In this case, the browser saves the file to the temporary folder and prompts you to open or save the file.

NOTE: You must save the file within five minutes of creating it or it is not available.

Step 7 Give the file a short name that identifies the Supervisor and subordinate stations, and save the file to a location other than the System Home or User Home.

- Step 8** Finally, make a note of the date and where you saved the downloaded file for reference when migrating the file.

Migrating each station

The migrator is a command line utility. Using the backup .dist file, it creates the EC-Net 4 Supervisor and subordinate station folders. Then, using EC-Net 4 Pro, you open a platform connection to each new platform and install each new station using the Station Copier.

Prerequisites: EC-Net 4 is installed on your PC. You know the name of the .dist file and where it is located. Zip files must be “unzipped” to expose the “*.dist” file for the migrator to process.

- Step 1** Open a standalone Console window using the shortcut from the Windows Start menu.

NOTE: Do not use the EC-Net 4 Pro embedded Console. It cannot accept the required keyboard inputs during the migration process.

- Step 2** Change directories to the location of the backup .dist file.

NOTE: If you change directories to the location of the backup .dist file you do not need to specify the path to this file.

- Step 3** Enter: n4mig distributionfilename.dist, where distributionfilename.dist is the name of the backup .dist file you created.

If the current directory does not contain the .dist file, include the correct file path in the command line to precede the .dist file name. For example: c:\path\distributionfilename.dist.

- Step 4** When the process pauses, respond to the prompts in the console window, as follows:

- Please select a migration template:1: Controller Migration Template or 2: Supervisor Migration Template
- Enter the passphrase used to protect EC-Net 4 BOG files.

If you know the passphrase of the system that you are targeting with the migrated station(s), enter it here. If you do not know or have access to the system passphrase, create one now and remember it. You will have to enter it when you copy the migrated station to the EC-Net 4 host system. For more details about passphrase refer to the *EC-Net 4 Platform Guide* topics: “System Passphrase” and “System passphrase usage in backups and station copies”.

While the migration utility runs, it displays the log in the console window. The final message displays the source, target, and Migration Report locations for reference, as shown below:

```
Completed Migration
Source: backup_EntSec24Supervisor_190411_1415.dist
Target: C:\Users\User123\Niagara4.8\brand\stations\EntSec24Supervisor
Migration Report: C:\Users\User123\Niagara4.8\brand\stations\
backup_EntSec24Supervisor_190411_1415.dist_miglog_190411_1423.html
```

- Step 5** Make a note of these locations.

Copying the migrated files

The migrator places all migrated files in the same location. You need to copy these files to the proper location for installation.

Prerequisites: The migrator ran successfully. You have installed EC-Net 4 on your PC.

- Step 1** Using the operating system’s File Explorer, copy the new, migrated, stations to the user_home/Niagara 4.x/brand/stations directory.

- Step 2** Launch the EC-Net 4 Pro and make a platform connection to the Supervisor station.

Step 3 Do one of the following:

- If the PC you are using also runs the Supervisor station, use Station Copier to copy Supervisor station from the user_home to the ProgramData directory.
- If the Supervisor station runs on a separate PC, export the tables from the MySQL or SQL Server database and copy the ProgramData directory with the exported tables to the other PC.

Setting up the Supervisor's database password

When you upgrade a Supervisor station from EC-Net^{AX} to EC-Net 4, the migrator preserves the password for the Supervisor's third-party database (MySQL, Oracle, MS SqlServer).

Prerequisites: The third-party database exists in the Supervisor station.

If no database exists in the Supervisor station, create the new database schema and import to the new schema the database tables that you previously exported.

Step 1 Using a browser (web UI), make a connection to the station.

Step 2 Click **System Setup**→**Miscellaneous**→**Configure Database**.

The **Configure Database** view opens to the Database Services tab.

Step 3 Select the Database tab, for example, MySQLDatabase or SqlServerDatabase.

The selected database (MySQLDatabase or SqlServerDatabase.) tab opens.

Configure Database	
<input type="button" value="Save"/> <input type="button" value="Manage Databases"/> <input type="button" value="Set Orion Database"/> <input type="button" value="Restart Station"/>	
<input checked="" type="radio"/> Database Services <input type="radio"/> MySQLDatabase	
Status	{ok}
Enabled	true
Fault Cause	
Health	Ok [27-Jun-22 5:09 PM EDT]
Alarm Source Info	Alarm Source Info >
Host Address	ip:localhost
User Name	MySQLUserName
Password
Database Name	myentsec
Port	3306
Extra Connection Properties	
Max Active Connections	30
Peak Active Connections	0
Active Connections	0

Step 4 Confirm that:

- **Enabled** is set to true.
- **Host Address** is localhost (for the Supervisor station).
- **User Name** is correct for the password.

- **Database Name** matches the name of the database.

Step 5 To change the **Password** type a new, strong password and click **Save**.

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

After making any database change, the framework restarts the station. After the station restarts, the database should connect using the new credentials, providing data communication from the newly-connected database.

If **Status** is {down} and **Fault Cause** reports that EC-Net 4 Pro cannot find the connector even though a connector is in the C:\Niagara\Niagara-home\jre\lib\ext folder, make sure that the connector you are using is compatible with the MySQL software version. For example, version 5 connector will not work with MySQL version 8.

The User Name to define on the MySQLDatabase tab is the database User Name, not your station's User name.

Creating and updating the HSQL database password

Each controller ships from the factory with a default HSQL database password and upgrading an EC-Net^{AX} station to an EC-Net 4 station resets the HSQL database password back to the factory default. While the default password initially works, as soon as you install or upgrade a controller and its software, you should change this password to a unique and strong string.

Prerequisites: You have just installed a new controller or upgraded an existing controller from EC-Net^{AX} to EC-Net 4. You are using EC-Net 4 Pro running on a PC that is connected to the network that services the controller.

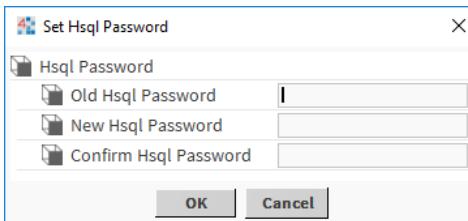
The password must contain a minimum of 10 alphanumeric-only characters along with these special characters: @ # ! \$ & + > <] [() . The password strength check requires at least: 1 digit, 1 upper case character and 1 lower case character.

Step 1 Connect to the controller station using EC-Net 4 Pro.

Step 2 Expand **Config**→**Drivers**→**RdbmsNetwork**.

Step 3 Right-click **HsqlDbDatabase** and click **Actions**→**Set Hsql Password**.

The **Password** window opens.



Step 4 Enter the Old Hsql Password, configure a strong new password and click **OK**.

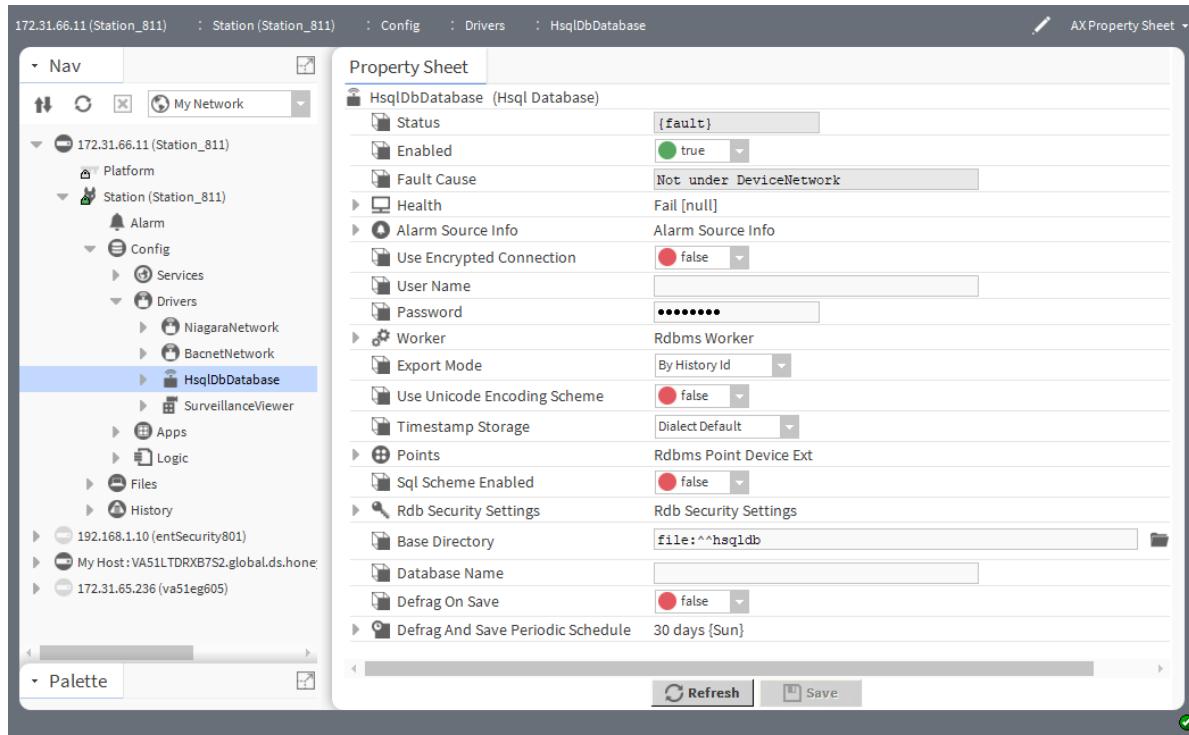
If you are changing the password after installing a new controller or after upgrading from EC-Net^{AX} to EC-Net 4 the default password is EC-Net123.

Step 5 Make a note of the new password and store it in a safe place.

As a best practice, consider changing this password on a regular basis.

Step 6 Follow these steps, if you are using an existing EC-Net 4 station. Open the device's **AX Property Sheet** by right-clicking on **HsqlDbDatabase** followed by clicking **Views**→**AX Property Sheet**.

The **AX Property Sheet** view opens.



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

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

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

Configuring network device passwords

The devices on the station network may require password reentry in the migrated stations. For example, a camera or DVR may have a unique device password that needs to be re-entered because it was cleared during migration process.

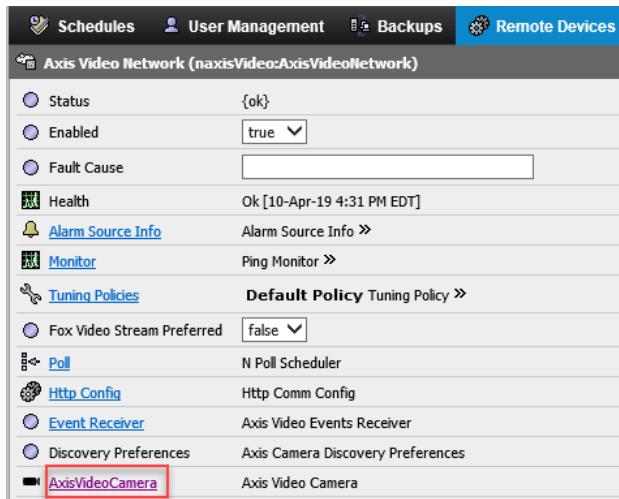
Prerequisites: Your PC is connected to the network with the remote controllers.

Step 1 From a browser (web UI), make a connection to the remote controller station.

Step 2 Navigate to the **Remote Drivers** view (**Controller Setup**→**Remote Devices**→**Remote Drivers**).

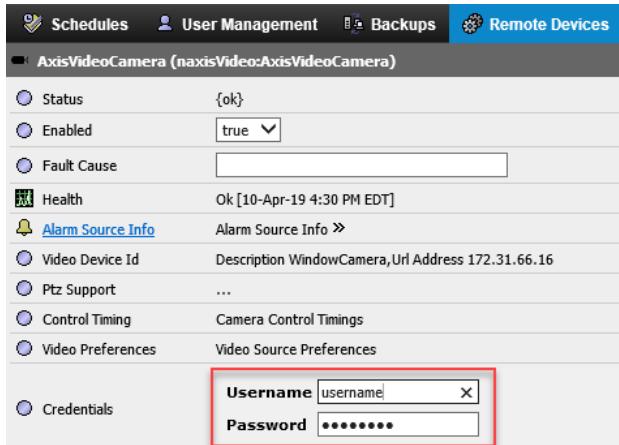
NOTE: The following steps can be for any network device that requires a password. The steps below show an example for a video camera device.

Step 3 For each network driver that may require a password, select the driver, click the **Hyperlink** button () on the menu and navigate to the device in the network (for example a Camera).



The screen capture shows this link for an Axis camera.

- Step 4** To display and update the device properties, click the device link.



- Step 5** Update the device password and click **Save**

The device password is updated.

Deleting the old certificates

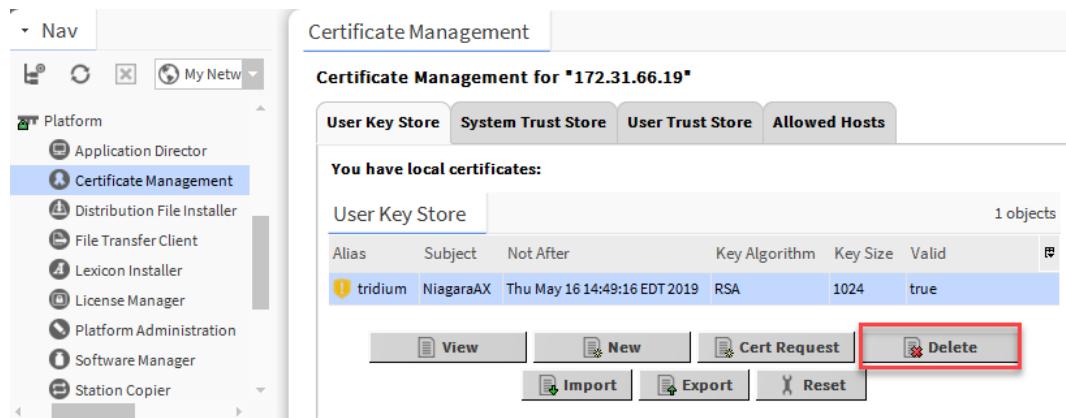
To promote secure communication in EC-Net 4 it is recommended that you install new TLS certificates. This procedure deletes the old certificates. Perform these steps for each station, Supervisor and all remote stations.

Prerequisites: You are using the Supervisor PC running EC-Net 4 Pro. All stations are idle.

NOTE: Subsequent procedures assume that this installation uses default self-signed certificates. If the company uses signed certificates (the most secure configuration), you will need to create a new root CA certificate, create a new server certificate for each platform/station, use the root CA certificate to sign each server certificate and install each signed server certificate after completing the migration.

While you may delete certificates using the browser (web UI), this procedure manages all certificate processes using EC-Net 4 Pro.

- Step 1** Open a platform connection to the Supervisor PC or remote controller (**File**→**Open**→**Open Platform**).
Step 2 Expand the **Platform** node and double-click **Certificate Management** in the Nav tree.



- Step 3 In the User Key Store, select the certificate, click **Delete** (as shown above) and click **Yes** to confirm the deletion.
- Step 4 Select the Allowed Hosts tab, select an allowed self-signed certificate, click **Delete** and click **Yes** to confirm the deletion.
This action deletes the certificate.
- Step 5 Restart the station.
- Step 6 Perform these steps in the Supervisor and each remote station.

Reconnecting the Supervisor and remote stations

This procedure reconnects the remote stations to the Supervisor station.

Prerequisites: You have deleted all old EC-Net^{AX} certificates from all stations.

- Step 1 Using the web UI, make a connection to the Supervisor station and accept the self-signed certificate.
- Step 2 Click **System Setup** → **Remote Devices** → **Station Manager**.
The **Edit** window opens.

The 'Edit' window has a title bar 'Edit'. The main area contains the following configuration options:

- Station Name:** entSecurity602
- Enabled:** true
- Address:** 172.31.66.19
- Credential Store:** UsernameAndPassword
- Username:** admin
- Password:** (Redacted)
- Use Foxs:** true
- Fox Port:** 4911

At the bottom are two buttons: 'Continue to Remote Configuration...' and 'Cancel'.

- Step 3 To re-establish the connection between the Supervisor and each subordinate station, enter the remote station credentials.
- Step 4 Approve each station's self-signed certificate.
- Step 5 For each subordinate station, do a **Join**, as described in "Joining a controller station to the Supervisor station" in the *EC-Net Access Installation and Maintenance Guide*

- Step 6** To set up data in each remote station, do a **Replicate**, as described in the “Replicating a configuration” topic in the *EC-Net Access Installation and Maintenance Guide*.

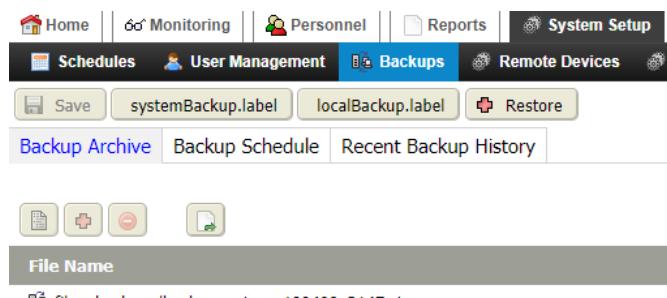
Restoring the BackupService schedules

This procedure describes how to restore the BackupService schedules.

Prerequisites: You are connected to the Supervisor station using a browser (the web UI).

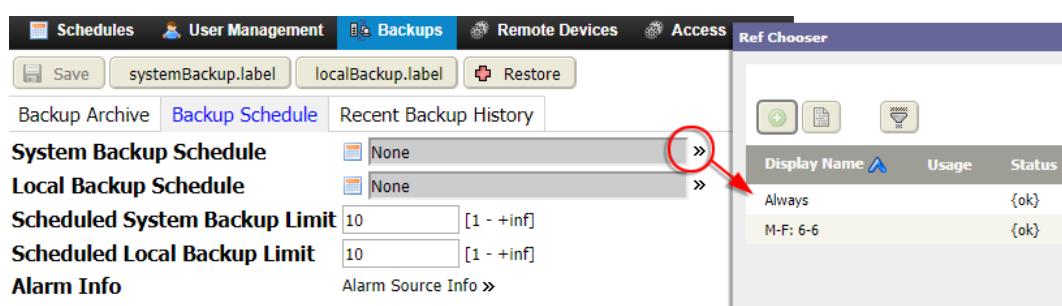
- Step 1** Click **System Setup**→**Backups**.

The **Backups** view opens.



- Step 2** Click the **Backup Schedule** tab.

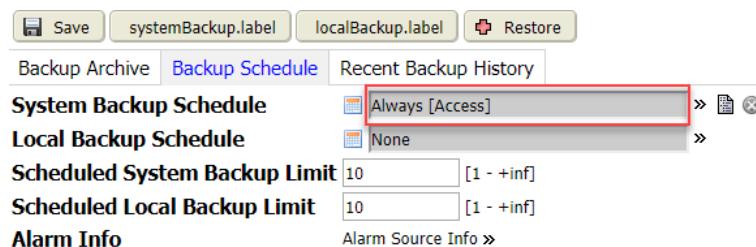
The **Backup Schedule** view appears.



- Step 3** To select the backup schedule, click the Ref Chooser chevron (>>)

The **Ref Chooser** window opens.

- Step 4** To assign a schedule to the backup, select the desired schedule from the table and click **Ok**.



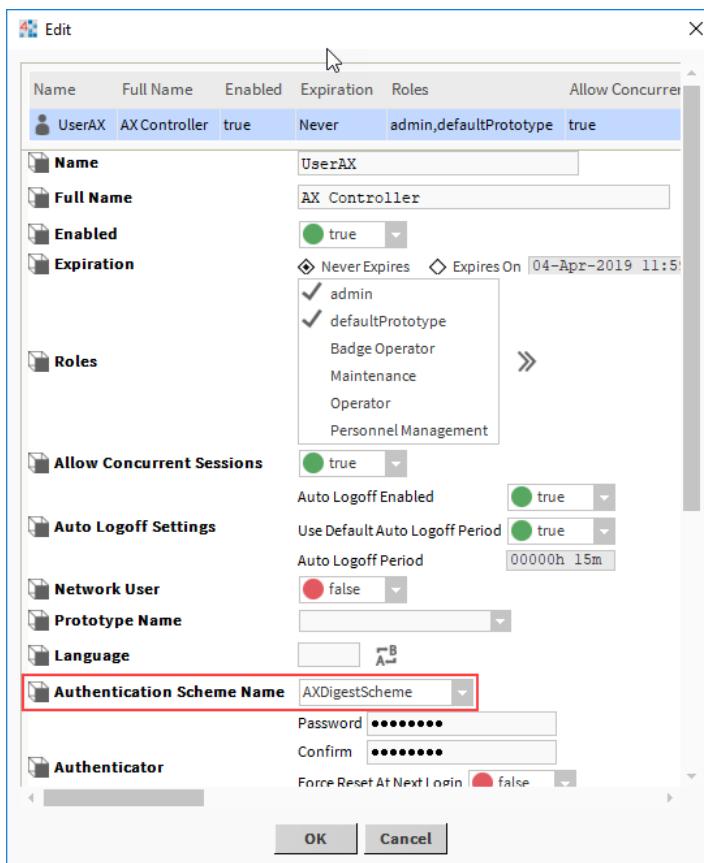
Repeat this procedure for all the migrated stations.

1. Configuring the EC-Net Access–4.8 Supervisor station connection

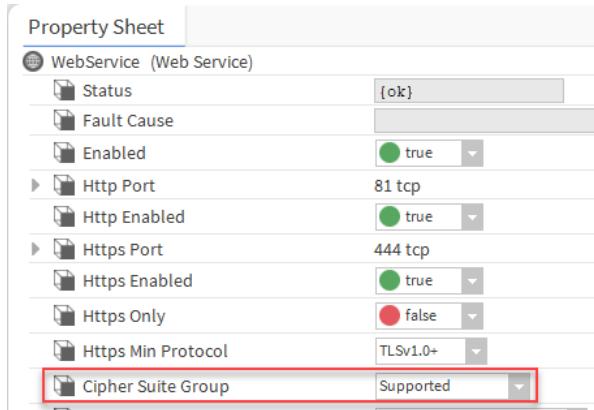
This procedure describes how to setup the EC-Net Access–4.8 station so that a legacy EC-BOS-6^{AX} controller (running a station with EC-Net^{AX}-3.8) can connect and authenticate a user with the newer station.

Prerequisites: EC-Net Access–4.8 EC-Net 4 Pro application is installed and running.

- Step 1 Using EC-Net Access–4.8 EC-Net 4 Pro, connect to the security Supervisor station that will establish a connection with the legacy controller.
- Step 2 In the Supervisor station, navigate to **Config→Services** and double-click on the UserService node. The **User Manager** view opens.



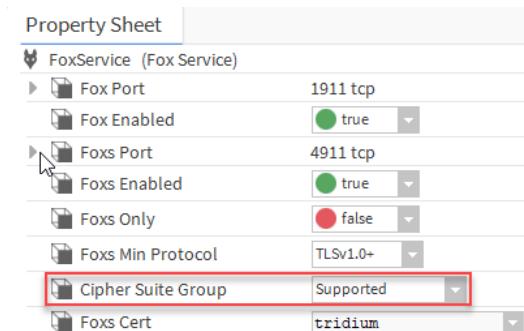
- Step 3 In this view, create a new user that you will use later to authenticate the EC-Net^{AX} controller station connection back to this Supervisor station.
- Step 4 Name the User and set the property values including these two properties:
 - For **Authentication Scheme Name**, choose: AX digest scheme.
 - For **Roles**, choose: admin.
- Step 5 Navigate to **Config→Services→WebService**. The WebService **Property Sheet** opens.



Step 6 Set the **Cipher Suite Group** property value to the **Supported** option and click **Save**.

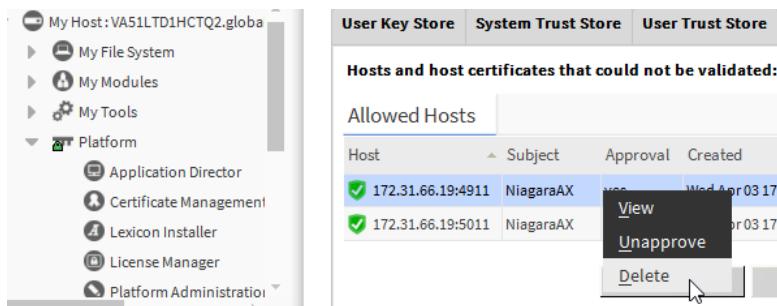
Step 7 Navigate to **Config→Services→FoxService**.

The FoxService **Property Sheet** opens.



Step 8 Set **Cipher Suite Group** to the **Supported** option and click **Save**.

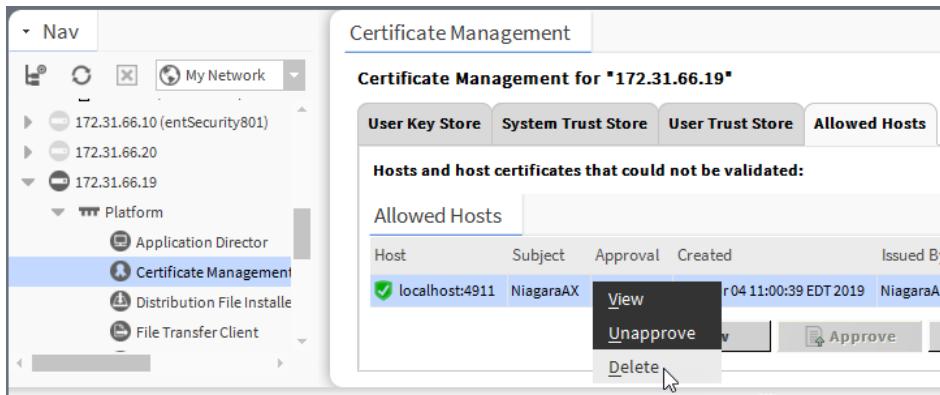
Step 9 Navigate to the Supervisor Certificate Management view (**Platform→Certificate Management**) and delete the remote controller certificate from the **Allowed Hosts** table.



Step 10 Within EC-Net 4 EC-Net 4 Pro, open a platform connection to your EC-Net^{AX} controller.

While you cannot connect to an EC-Net^{AX} station from EC-Net 4, you can connect to the platform.

Step 11 Navigate to the remote controller **Certificate Management** view (**Platform→Certificate Management**) and delete the Supervisor certificate from the **Allowed Hosts** table.



2. Adding a remote controller 2.3 station

Use this procedure to initiate the connection between EC-Net Access-4.8 stations and EC-Net^{AX}-3.8 stations. Authentication fails on the initial attempt to connect but may complete in a subsequent procedure.

Prerequisites: The Supervisor station is configured to accept the EC-Net^{AX}-3.8 user connection.

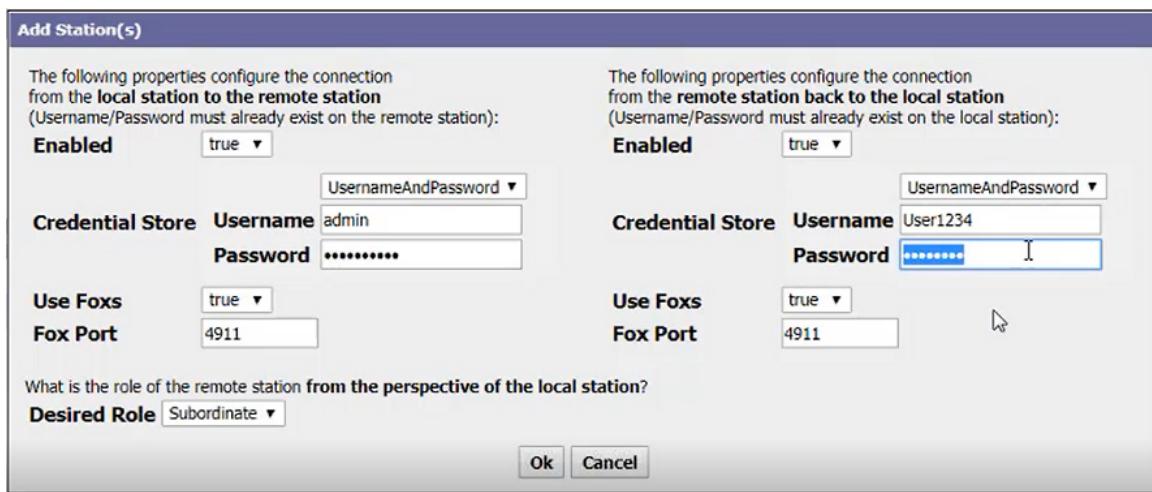
- Step 1 Using the browser of your choice and the Java web launcher app, open a station connection to your Supervisor and controller stations using the user with admin rights.
- Step 2 To find the EC-Net^{AX} Stations to connect to from the EC-Net 4 Supervisor, navigate to **Remote Devices** and run a discovery job.

The station appears in the **Discovered** pane.

NOTE: If discovery does not find the station, manually add the station (click the **Add** button, and provide the EC-Net^{AX} controller's IP address and credentials).

- Step 3 Select the station to add and click the add (+) button.

The **Add Station(s)** window opens.



- Step 4 Starting with the left side of the window, enter the **Username** and **Password** for Supervisor to login to the remote station (these credentials must already exist on the remote controller).
- Step 5 On the right side of the window, enter the **Username** and **Password** for the newly created user so that the remote EC-Net^{AX}-3.8 controller can log in to the Supervisor (these credentials must already exist on the Supervisor).
- Step 6 Make sure that the **Desired Role** property is set to **Subordinate** and click **Ok**.

The Supervisor runs the add-station job and adds the remote station to the Supervisor database.

NOTE:

An **Authentication Failure** message displays at the completion of the job. This is because of the different credential storage capabilities in the two different versions. The following steps provide a way to resolve this from the remote controller and complete the connection.

The screenshot shows the Niagara Framework interface with the title 'System Setup' and 'Remote Devices' selected. A table lists a single remote device entry:

Station Name	Host Name	Scheme	Fox Port	Status	Actual Role	Role Status
entSecurity602	172.31.66.19	foxs	4911	{ok}	Peer	{fault} [entSecurity602]: SysDef extension not in a valid state to change roles: {down}

Below the table, there is a section titled 'Discovered' with icons for new stations. A cursor points to the 'Host Name' column header in the table.

- Step 7 Navigate to **System Setup**→**Remote Devices**→**Certificate Management** and approve the remote controller certificate.

3. Logging in to the Supervisor from the controller

These steps connect from the EC-Net^{AX}-3.8 legacy security station to the security Supervisor and authenticate the connection using the newly-created credentials for a user with **AXDigestScheme** authentication.

Prerequisites: The record for the remote legacy controller is added to the EC-Net Access-4.8 Supervisor station database.

- Step 1 In the browser, connect to the remote EC-Net^{AX} controller: from the main menu, select **Controller Setup**→**Remote Devices** and click on the **Certificate Management** view, approve the Supervisor certificate.
- Step 2 From the main menu, select **Controller Setup**→**Remote Devices** and click on **Station Manager** to display the **Station Manager Database** view.

The Supervisor station displays as a row in the table view.

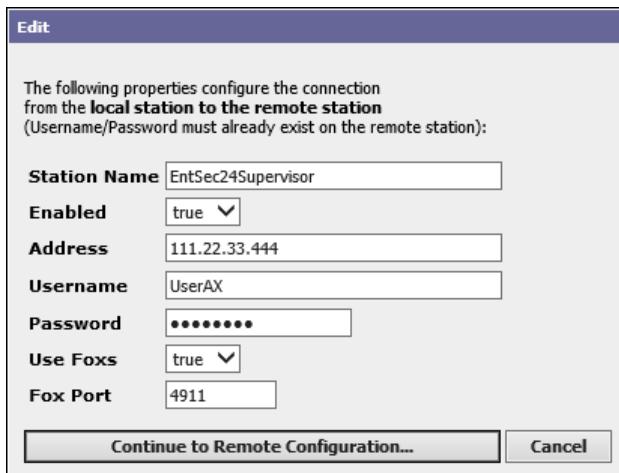
The screenshot shows the Niagara Framework interface with the title 'Controller Setup' and 'Remote Devices' selected. A table lists a single supervisor station entry:

Station Name	Host Name	Scheme	Fox Port	Status	Actual Role	Role Status
EntSec24Supervisor	137.19.60.123	foxs	4911	{down}	Peer	{ok}

A cursor points to the 'Edit' icon (pencil) in the first column of the table.

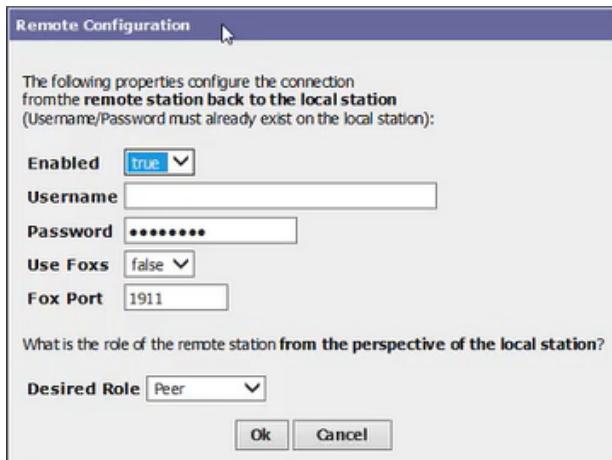
- Step 3 Select the Supervisor station and click the Edit icon (>Edit).

The **Edit** window opens.



- Step 4 Enter credentials for the newly created EC-Net^{AX} user, verify that the other properties are correct for the connection and click **Continue to Remote Configuration**.

The connection to the remote station (Supervisor) continues and the **Remote Configuration** window appears.



- Step 5 Click **Cancel** in the **Remote Configuration** window.

The system closes the window. To authenticate the connection, you must complete the process with the following task.

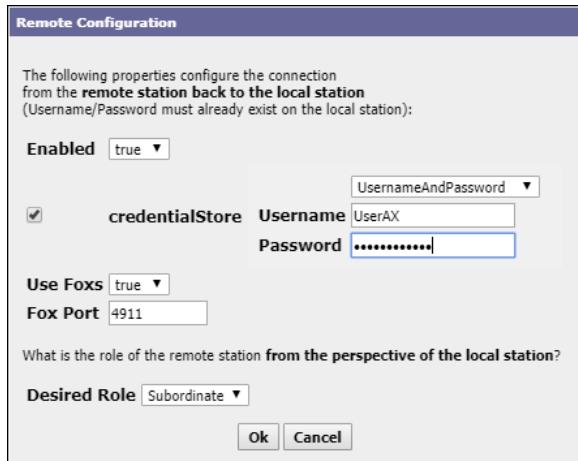
4. Connecting from the Supervisor to the EC-Net^{AX}-3.8 legacy security station

The following task describes how to connect from a configured legacy controller station (running EC-Net^{AX}-3.8) to a properly configured EC-Net Access-2.4 Supervisor.

Prerequisites: The legacy controller station has completed a user login to the EC-Net Access-2.4 Supervisor station.

- Step 1 In the browser, connect to the Supervisor and navigate to the station database view (**System Setup** → **Remote Devices** → **Station Management**)
- Step 2 Select the remote EC-Net^{AX}-3.8 legacy security station and click the Edit icon (edit icon). These fields should already be populated. Verify the fields and click **Continue to Remote Configuration**.

The **Remote Configuration** window opens.



Step 3 Enter the newly created user credentials and click **Ok**.

The **Remote Configuration** window closes, the configure job runs and establishes the connection with the correct relationship between Supervisor and subordinate. You are ready to proceed with station **Join** and **Replicate** procedures.

Once the EC-Net Access-4.8 Supervisor station connects to the EC-Net^{AX}-3.8 legacy station, no further migration of the legacy station is required. The legacy station cannot be migrated to EC-Net Access-4.8 until the EC-BOS-6^{AX} is replaced with newer hardware.

Chapter 5 Migration reference

Topics covered in this chapter

- ◆ Imports
- ◆ API changes
- ◆ Primitives
- ◆ Security Manager
- ◆ Example Program object fix

A number of changes to EC-Net^{AX} Program objects (components) need to be made to allow them to run under EC-Net 4. The EC-Net 4 migration tool can make some of these changes for Programs as part of the station migration process.

However, other changes often require post-migration work on certain Programs in the (offline) migrated station.

Imports

The first step in converting Program objects to run under EC-Net 4 is to update imports to declare dependencies on the appropriate runtime profile JAR files. In most cases, this will be the `rt` runtime profile JAR file for the module (baja and nre do not have a runtime profile extension).

If your Program object was run through the EC-Net 4 migration tool as part of a station migration, this change will have been made for you.

API changes

If your Program objects use certain APIs that have had breaking changes, they will need to have their Program code updated and recompiled using the new EC-Net APIs.

Most notable are the “Collection API/Cursor”, “Alarm API”, “History API”, and “Logging API”. For more information on these changes and how to update your code, see the *Niagara Developer Guide* for each of these API changes.

Also, note `getProgram()` was deprecated starting in EC-Net^{AX}-3.5 in order to support compiling Programs into Program Modules. It has been removed for EC-Net 4 v4.0. While in the past, Program objects using this method would still compile and run, they now require conversion to use `getComponent()` instead.

Primitives

Slots that are defined as `baja:Boolean`, `baja:Double`, `baja:Float`, `baja:Integer`, `baja:Long` and `baja:String` now return their Java primitive types.

This means that you no longer need to call `getBoolean()`, `getDouble()`, `getFloat()`, `getInt()`, `getLong()` or `getString()` to cast the values of these slots to primitive values.

For example, for a slot defined on a Program as: `Name=temperature, Type=baja:Double`
EC-Net^{AX} Program object:

```
double temp = getTemperature().getDouble();
```

EC-Net 4 Program object:

```
double temp = getTemperature();
```

Security Manager

EC-Net 4 utilizes the Java Security Manager, which puts restrictions on what Program objects may do. Program objects interacting with the station's component tree should be largely unaffected by the Security Manager.

File reading/writing

A restriction has been put on file reading and writing to only allow read-from and write-to the station_home directory (file:^).

Runtime.exec()

The ability to call external executables is extremely useful in certain situations, however the use of `Runtime.exec()` can allow potentially malicious code to be executed. As such, several restrictions are placed on it to help protect your system.

Only station super users can add and edit Program objects. In EC-Net^{AX}, there was a flag/entry to change this behavior in `system.properties`. In EC-Net 4, this ability has been removed.

Program objects can no longer directly call `Runtime.getRuntime().exec(command)`. Instead, a wrapper called `ProgramRuntime` behaves the same way as `java.lang.Runtime`, except that it takes the Program object as an additional parameter. Commands executed by `ProgramRuntime` are logged and audited.

Additionally, to enable the use of `ProgramRuntime`, the hidden slot `allowProgramRuntimeExec` on the station's **ProgramService** must be set to true.

NOTE: Only standalone Program objects can use `ProgramRuntime.exec()`. Program objects that have been compiled into Program Modules cannot call this function.

An example of this in Program code is below

EC-Net^{AX} Program object:

```
Runtime.getRuntime().exec("notepad.exe");
```

EC-Net 4 Program object:

```
ProgramRuntime.getRuntime().exec(this, "notepad.exe");
```

Example Program object fix

The following is a simple example fix for a Program object included in the widely-distributed EC-Net^{AX} demo station. It is one of several Programs flagged with a “WARNING unable to compile” entry when EC-Net^{AX}-3.8 station backup .dist file for the demo station is used as source in the EC-Net 4 migration tool.

A related snippet from the migration report (log file) looks like below.

```
WARNING unable to compile Program object PxHome.Graphics.Residential.First Floor.
GarageProgram
C:\Users\e333988\Niagara4.x\temp\Prog_ea44c5d3b904631ea44c5d3b904631.java:40: error:
cannot find symbol
    Action action = getProgram().getAction("execute");
                           ^
symbol:   method getProgram()
location: class Prog_ea44c5d3b904631ea44c5d3b904631
```

This error relates to use of `getProgram()`, as described in [API changes, page 83](#).

Opening the migrated “demo” station in your EC-Net 4 EC-Net 4 Pro, you can follow the ORD given in the warning, and open the **Program Editor** view for that Program (“GarageProgram”), as shown in the following figure.

Figure 23 Example Program with issues from migrated “demo” station

```

public void onStart() throws Exception
{
    Action action = getProgram().getAction("execute");
    ticket = Clock.schedulePeriodically(this.getProgram(), BRelTime.makeSeconds(10), action, null);
}

public void onExecute() throws Exception
{
    double rnd = (Math.random()-.5)/1000;
    if(getDoorOpen().getValue())
    {
        getOut().setValue((getOut().getValue()+getOutside().getDouble()*2)/3+rnd);
    } else {
        getOut().setValue((getOut().getValue()*9+(getSetpoint().getValue()+getOutside().getDouble())/2)/10+rnd);
    }
}

public void onStop() throws Exception
{
    ticket.cancel();
}

private Clock.Ticket ticket;

```

The WARNING snippet resulted from use of the now-obsolete `getProgram()`, which must be replaced by `getComponent()`. Also note the Program status: “Program is out of date and requires compile.”

If you compile the Program now without making any changes, various errors (4) appear in the console area at the bottom of the EC-Net 4 Pro window, as shown below.

Figure 24 Example errors from a compile before making any Program changes

```

private Clock.Ticket ticket;

C:\Users\e333988\Niagara4.0>"C:\Niagara\Niagara-4.0.10.2\jre\bin\javac" -encoding UTF-8 -profile compact3 -Xlint:de
Prog_84d5c47c19654e0d84d5c47c19654e0d.java:40: error: cannot find symbol
    Action action = getProgram().getAction("execute");
                           ^
symbol:   method getProgram()
location: class Prog_84d5c47c19654e0d84d5c47c19654e0d
Prog_84d5c47c19654e0d84d5c47c19654e0d.java:41: error: cannot find symbol
    ticket = Clock.schedulePeriodically(this.getProgram(), BRelTime.makeSeconds(10), action, null);
                           ^
symbol:   method getProgram()
Prog_84d5c47c19654e0d84d5c47c19654e0d.java:51: error: double cannot be dereferenced
    getOut().setValue((getOut().getValue()+getOutside().getDouble()*2)/3+rnd);
                           ^
Prog_84d5c47c19654e0d84d5c47c19654e0d.java:53: error: double cannot be dereferenced
    getOut().setValue((getOut().getValue()*9+(getSetpoint().getValue()+getOutside().getDouble())/2)/10+rnd);
                           ^
4 errors
C:\Users\e333988\Niagara4.0>

```

After replacing the two `getProgram()` instances to `getComponent()`, saving and then recompiling again, errors are reduced to two—related to use of `getDouble()` for primitives (Figure 25 Figure 28, page 86).

Figure 25 Example Program with remaining issues from primitive syntax errors

```

Program cannot be compiled.

Edit Slots Imports Source

public void onStart() throws Exception
{
    Action action = getComponent().getAction("execute");
    ticket = Clock.schedulePeriodically(this.getComponent(), BRelTime.makeSeconds(10), action, null);
}

public void onExecute() throws Exception
{

    double rnd = (Math.random()-.5)/1000;

    if(getDoorOpen().getValue())
    {
        getOut().setValue((getOut().getValue() +getOutside().getDouble()*2)/3+rnd);
    } else {
        getOut().setValue((getOut().getValue()*9+(getSetpoint().getValue() +getOutside().getDouble())/2)/10+rnd);
    }
}

public void onStop() throws Exception
{
    ticket.cancel();
}

```

The screenshot shows the Niagara software's program editor. A red arrow points to the line `getOut().setDouble()*2)`, and another red arrow points to the line `getDouble())/2)`. Both lines contain a closing parenthesis that is unnecessary according to the migration guide.

This is no longer necessary, as explained in the section [Primitives, page 83](#). To fix, simply remove the text shown marked above.

As shown below, now after resaving the Program and recompiling, all errors are gone.

Figure 26 Example Program object after fixing all issues and recompiling.

```

Nav Edit Slots Imports Source

My Host : VA My File : Sys User ...
public void onStart() throws Exception
{
    Action action = getComponent().getAction("execute");
    ticket = Clock.schedulePeriodically(this.getComponent(), BRelTime.makeSeconds(10), action, null);
}

public void onExecute() throws Exception
{
    double rnd = (Math.random()-.5)/1000;
    if(getDoorOpen().getValue())
    {
        getOut().setValue((getOut().getValue() +getOutside().getDouble()*2)/3+rnd);
    } else {
        getOut().setValue((getOut().getValue()*9+(getSetpoint().getValue() +getOutside().getDouble())/2)/10+rnd);
    }
}

public void onStop() throws Exception
{
    ticket.cancel();
}

private Clock.Ticket ticket;

```

The screenshot shows the Niagara software's program editor after fixing the issues. The status bar at the top says "Program is up-to-date". The bottom console area shows the command-line output of the Java compiler: `C:\Users\...\Niagara4.0>C:\Niagara\Niagara-4.0.10.2\jre\bin\javac" -encoding UTF-8 -profile compact3 -Xlint:dep C:\Users\...\Niagara4.0>`.

Now note the status of the Program is: "Program is up-to-date", and there are no errors reported in the bottom console area after the last compile and save.

DISTECH
CONTROLS™

AX to EC-Net4_UG_15_EN