

NIAGARA AX to NX MIGRATION

Contents

About this Guide.....	5
Document change log.....	5
Related documentation	5
Chapter 1 Migration overview	6
Why a migration tool?.....	6
Platform compatibility.....	7
Platform resource requirements	8
Recommended minimum resources by controller type	8
Checking for minimum resource requirements	9
Driver and application support.....	11
Wire compatibility	12
Supported connections.....	12
Features and compatibility.....	12
File locations	13
System home.....	14
Controller user home.....	15
Migration workflow diagram.....	20
Migration workflow process.....	21
Backup stations and migrate databases.....	23
Upgrade the Supervisor.....	23
Convert controllers and install new stations	23
Perform system cleanup	23
Chapter 2 Migration tasks	24
Preparing platforms for migration	24
Export certificate Trust Store and Private Key Stores	24
Configuring a station to migrate users.....	24
Kerberos keytab file notes.....	25
Making source backups	26
Secure license files	29
Run the migration tool	29
About the migration tool.....	29
Running the migration tool	32
Migration tool command usage.....	33
Migrating modules made with the Program Modulecomponent	35
Migration details	40
Migration execution.....	40
Migration report (log file).....	41
Post-migration database tasks.....	41
Modifying Program objects	41
Considerations if a station functions as an oBIX server	43
Preliminary checkout of migrated stations.....	44

Upgrade/convert platforms and install stations.....	44
Copying the Supervisor station	45
Performing station cleanup.....	47
Controller conversion	47
Downgrading controllers	52
Chapter 3 System verification	54
Non-default environment files	54
Verify station-to-station communications.....	54
Provisioning notes	55
Verify Supervisor to controller platform daemon communications	55
Provisioning considerations when JACE controllers remain.....	56
Station User notes in migrated stations.....	57
Permissions moved to Roles	58
Authentication is user-specific.....	59
Chapter 4 Migration reference.....	60
Imports.....	60
API changes.....	60
Primitives	60
Security Manager.....	61
File reading/writing	61
Runtime.exec().....	61
Example Program object fix.....	61

About this Guide

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

Product Documentation

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

Document Content

This guide provides information about migration of NiagaraAX system running AX-3.8 to Niagara 4 system running N4.0, and has the following main sections.

- [Chapter 1 Migration overview](#), page 6
Provides requirements, restrictions, compatibilities between NiagaraAX and Niagara 4, background on the need for migration (including major differences), and a general migration workflow process.
- [Chapter 2 Migration tasks](#), page 24
Provides tasks to start migrating stations, including collecting AX-3.8 station archives, running the N4 migration tool, migrated database cleanup, preliminary N4 station checkout, and installation of Niagara 4 software and controller upgrades.
- [Chapter 3 System verification](#), page 54
Explains system areas that may need attention and possible configuration immediately following the installation of Niagara 4 upgrades and migrated stations.
- [Chapter 4 Migration reference](#), page 60
Contains some details on Program object (component) changes in N4 , along with an example "Program object fix".

Document change log

Updated May 6, 2016

- One change was made in the topic, [File Locations](#), page 13: in the table under the heading *Homes in a Supervisor*, corrected the path for the protected_station_home

Updated February 22, 2016 as follows:

- Added information about setting the configuration flag before migrating users. Changes were made to two topics:
 - [Features and compatibility](#) , page 12, and [Assess and prepare for migration](#), page 22.
 - The following topic was added: [Configuring a station to migrate users](#), page 24
- Edited the entire document to streamline the way software names are used and to convert task material to task steps. For example, the table in [Features and compatibility](#) , page 12 was redesigned to simplify software name usage.

Initial release document, August 7, 2015.

Related documentation

Additional information about Niagara platforms, installation and operation Workbench is available in the following documents.

- *Niagara 4 Platform Guide*
- *Getting Started with Niagara*

Chapter 1 Migration overview

Topics covered in this chapter

- ◆ CentraLine Specifics
- ◆ Why a migration tool?
- ◆ Platform compatibility
- ◆ Driver and application support
- ◆ Wire compatibility
- ◆ File locations
- ◆ Migration workflow diagram
- ◆ Migration workflow process

The Niagara Framework® has evolved to the point where existing NiagaraAX platforms and stations, if upgradable, must be *migrated* to Niagara 4. Therefore, after installing Workbench, you cannot simply use stations saved in NiagaraAX.

Migration involves several concepts:

- Conversion of AX artifacts to the corresponding N4 format; for example, station .bog files, Px files. The N4 migration tool handles this conversion.
- “Wire compatibility” (communications) between AX and N4, with the possibility that some migrated jobs must retain some number of AX controllers. Note that some platform types cannot be migrated, for example, JACE-2 controllers.
- For any platforms to be migrated that use custom AX software modules, the necessary “refactoring” of those modules as N4-compatible modules.

CentraLine Specifics

This user guide is for advanced users that are able to execute a migration of CentraLine stations manually by applying the appropriate steps correctly.

As the most efficient alternative, ARENA AX / NX provides the CI Station Upgrade Tool which is partly based on the major steps of the Niagara migration process described in this user guide but allows a convenient wizard-based migration of CentraLine stations in a simple user-friendly way.

We strongly recommend to use the CI Station Upgrade Tool since only then CentraLine-module-specific changes are considered during the migration process. For details on the CI Station Upgrade Tool usage, please refer to the CentraLine NX BACnet Utilities Driver User Guide, EN2Z-1020GE51.

Why a migration tool?

Niagara 4 has different platform and resource requirements than NiagaraAX, and there are other considerations that affect the suitability for a job to migrate from AX to Niagara 4. An understanding of these is essential before beginning migration work for any existing NiagaraAX job. As a result of the fundamental changes and security improvements in Niagara 4, a migration process is required. Simple reuse of any saved AX station is not possible. Following, are a few of the major reasons:

- Directory structure changes

In NiagaraAX, 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: `D:\niagara\niagara-3.8.38`. This directory includes runtime files and executables (modules, JRE, bin) along with configuration data files (stations, Workbench options and registry).

To improve security, these files and folders are relocated in Niagara 4. See [File locations](#), page 13.

- Software modules were refactored

In NiagaraAX, each software module typically contains multiple content levels: “runtime”, “user interface” (ui), and sometimes “doc”. When installing modules in JACE controllers that are unable to utilize all levels, AX platform tools “filter” out unneeded content levels at install time. However, Niagara 4 uses the

Java Security Manager, which does not support this model. So, software modules in Niagara 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 N4 migration tool addresses refactoring when converting an 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 Niagara 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 AuthenticationService 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 Niagara 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 Niagara 4, the station process in all JACE 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 NiagaraAX Platform Guide.

There are many other differences and numerous new features in Niagara 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 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	N4 supported / OS / notes	N4 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 un-supported.</i> Linux OS support is unavailable. Prior editions of Windows Server, such as Windows Server 2008, are unsupported.
JACE-8000	Supported	Initially supports <i>only</i> N4.0. At some future point, an update release of AX-3.8 <i>may</i> be available that supports most features of this newest controller.
JACE-7 (JVLN)	Supported	WiFi option is not supported. Some other option cards are also not supported in (see NPM below).
JACE-6E, JACE-603, JACE-645 (NPM6E)	Supported	Unsupported options include: GPRS, Sedona wireless, and dialup modem. Minimum resources are required (fully loaded units in AX may not run in N4.0). For details, see Platform resource requirements , page 8.
JACE-6, JACE-602 Express (NPM6)		
JACE-3E (NPM3E)		

Host Model	N4 supported / OS / notes	N4 compatibility notes
JACE-2, JACE-202 Express (NPM2)	Not supported	Support is available for data exchange with N4 platforms. For example, these stations can be under the NiagaraNetwork of N4 Supervisor. For related details, see Features and compatibility , page 13.
JACE-403		
JACE-545		
JACE-NXT		
JACE-NXS		
SoftJACE	Supported	Support is available for data exchange.

If you first upgrade a system to AX-3.8 in preparation for migration to N4, be sure to verify proper operation of the system (including NiagaraNetwork communications between stations), before making station backups to use in the migration process.

Niagara 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 8.

Platform resource requirements

N4 requires more disk space, more Java heap, and more system RAM to run than a similar AX station. Therefore, AX-3.8 JACE running near capacity will probably not fit the same station when converted to N4. A brief summary of reasons why are as follows.

- Disk space (Flash)
N4 requires between 14-16 MB more space than an equivalent 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, N4 requires more Java heap (a baseline of 10MB) due to new features and functionality, and more system memory when running on JACE controller.
- History RAM disk size (NPM6/NPM6E only)
To free additional system memory, the maximum history RAM disk size on JACE-6 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 JACE-6 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 N4.

JACE type/model	Minimum Free Disk Space, in MB	Java Max Heap, in MB	Minimum Free Heap, in MB (if maxHeap feature licensed (1))
NPM3 (JACE-3E)	22	94	38
NPM6 (JACE-6)	20	94	38

JACE type/model	Minimum Free Disk Space, in MB	Java Max Heap, in MB	Minimum Free Heap, in MB (if maxHeap feature licensed (1))
NMP6E (JACE-603, JACE-645)	24	94	38
JACE-7	22	94	90

(1) If an JACE is NOT currently licensed with the "maxHeap" feature, then heap usage will not be an issue in N4.

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

- 4MB free reserve for future N4 expansion
- 4MB for Data Recovery (NPM3E and NPM6E)
- 14MB for N4increase
- 2MB for JACE-6 boot partition increase

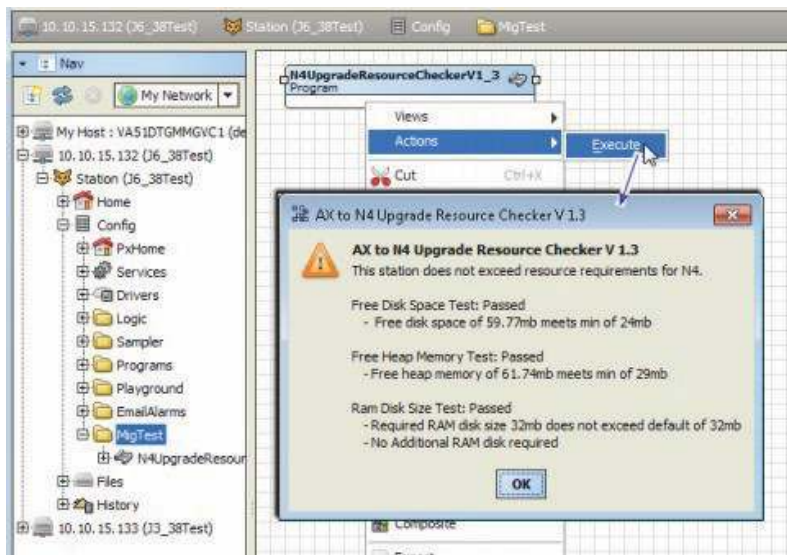
The Minimum Free Heap recommendation is based on the following:

(20% total 86 NB heap size = 17MB) + (13MB additional N4 overhead + 8 MB smaller heap (J3/J6 only) = 38 MB

Checking for minimum resource requirements

There is a special BOG file with the 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 AX-3.8 JACE station



Alternatively, you can manually check resources, and compare to the recommended resource minimums given in [Table 2 on page 6](#).

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, Workbench, 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 AX-3.8 system home folder.
- Step 2 Using Workbench, 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 AX-3.8 and running the station to be migrated.

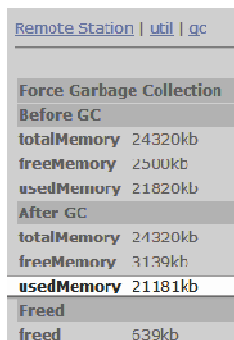
- Step 1 Using Workbench, open the station (Fox/Foxs)..
- Step 2 Expand Config→Services and double-click PlatformServices.
The Platform Service Container Plugin view opens.
- Step 3 In the Filesystem entry for /ffs0, compare the **Free** value with the value in [Table 2](#).
The JACE value should be 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 AX-3.8 and running the station to be migrated.

- Step 1 Using Workbench, 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	539kb

This represents the low point of Java heap usage.

Driver and application support

The initial release of Niagara 4 supports most of the commonly used NiagaraAX drivers, including BACnet, LonWorks, 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 JACE controllers	Module "platIEEE8021X"
OpenADR client drivers (simple, smart)	Modules "adr" and "dras"
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 "CryptoService" in station	Module "crypto"—replaced by "platCrypto" since AX-3.7 (should only be used by non-Hotspot JACEs, for example, JACE-2).
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 N4 prohibits most tunneling. Only serial tunneling is supported in , 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 N4 prohibits this tunneling.
Sedona (SedonaNetwork, SedonaJen6lpNetwork)	Module "sedonaInstaller" in AX builds, which, from a downloaded Sedona "bundle," would install Sedona-specific modules for integration. These are not supported.
Older video drivers based on the "dev-Driver", superseded in AX by "nDriver" based videoDriverframework	Modules "video" (base), "axis", "dedicatedMicros", "milestone", and "rapideye", superseded in AX (and in N4) by modules "nvideo", "naxisVideo", "ndedicatedMicros", "nmilestone", and "nrapideye".
Platform support for Niagara R2 on any QNX-based JACE controller	Niagara R2 platform support on JACE-603/JACE-645 controllers requires Workbench (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-Tridium) software modules used on any platform to be migrated must be refactored for use in Niagara 4. This is a separate process from using the migration tool on a saved 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.

Supported connections

Workbench can connect to stations running NiagaraAX-3 as well as stations running Niagara 4.

The following table summarizes the possible connections and indicates which are supported in Niagara 4.

From	To	Supported	Notes
Workbench	N4 station, N4 Supervisor	No	Prohibited in software.
Workbench	N4 platform	No	Prohibited in software.
Workbench	AX platform	Yes	Provides support necessary for AX to N4 upgrades. A few functions from the Workbench remote controller Platform Administration view are unavailable, including the Commissioning Wizard function.
Workbench	AX station	No	Prohibited in software. Note that ongoing AX station configuration requires Workbench.
AX station	N4 station, N4 Supervisor	Yes	Discovery of points, schedules, and histories in N4 stations is unavailable. However, you can manually add points. See Features and compatibility , page 12.
N4 station, N4 Supervisor	AX station	Yes	See Features and compatibility , page 12.
N4 station	AX Supervisor	No	Not prohibited in software, but unsupported.

Features and compatibility

A limited number of features are supported when communicating between Niagara 4 and AX-3.8.

Feature compatibility table

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

Feature	From N4 to AX	From AX to N4	Additional comments
Point discovery	No (AX Supervisor)	Yes (N4 Supervisor)	AX stations cannot discover points in N4 stations. The alternative is to manually add and configure the points.
Point data	Yes	Yes	N4 supports virtual points in AX stations. AX supports virtual points in N4 stations.
Export Histories	No	Yes	N4 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 AX is capable of supporting	No	Supported users are those assigned an authentication scheme with a password that can be ported to the AX station. The synchronization process is supported in only one direction.
Synchronize users in (sync in)	No	Yes, but only users AX is	LDAP is a good alternative for network user synchronization.

Feature	From N4 to AX	From AX to N4	Additional comments
		capable of supporting	Although AX stations do not support the HTML5 user prototype, you can synchronize users configured with the HTML5 user prototype to AX stations by enabling the <code>UserDefined1</code> configuration flag on the <code>web_WebProfileConfig</code> property of the user prototype slot in the AX station.
Virtual points	Yes	Yes	
Tunneling	n/a	n/a	As Fox, HTTP, and platform communications provide insufficient security, they are not supported in N4. Serial tunneling is supported.
Export tags	No, stations cannot join as subordinates	Yes, stations can join as subordinates	For the AX station (subordinate) to join an N4 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 pre-configure the two-way connection results in an error during the export tag join. Using Px files with ExportTags may be problematic as all Px files must be converted for use with N4.
File transfer using Fox (<code>NiagaraDeviceFileExt</code>) is supported.	Yes	Yes	File permission checks are enforced.
Provisioning	Yes	No	Supported operations for AX subordinates include: <ul style="list-style-type: none"> • Update licenses • Backup stations • Copy Supervisor file • Install valid AX software • Upgrade out-of-date AX software • Restore prior station backup, for example, from the step history view of a backup job. N4 provisioning robots will not run on AX stations.

NOTE: For Niagara 4 Supervisor to provision 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, Niagara 4 Supervisor cannot directly upgrade platforms to Niagara 4.

File locations

During the Workbench installation and platform commissioning processes, the software differentiates between two types of files based upon the content of the files: *configuration* and *runtime* data. Files and folders that contain configuration data reside in separate locations from files and folders that contain runtime data. This separation enhances security by denying general access to the runtime files and allowing each user access to only their personal configuration files.

As a result of separating configuration and runtime data, the system supports multiple home directories on the Supervisor or engineering workstation. These homes may be identified as:

- The system home contains runtime files, such as core software modules, the JRE, and binary executables.
- Workbench user home for each user contains configuration data, including option files, and registries.
- A platform daemon user home for the Supervisor or engineering workstation contains platform configuration data.
- Two station homes called, *protected station home* and *station home*, are part of each user home.

Homes on a Supervisor

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

Home in the Workbench Nav tree	Home in the Platform Administration view	Niagara 4 alias	Windows folder location and contents	File ORD shortcut
My Host→My File System→Sys Home	System Home	niagara_home	C:\niagara\niagara-4.x.xx Executables and software files	! (as in NiagaraAX)
My Host→My File System→User Home	N/A	niagara_user_home	C:\Users\userName\Niagara4.x\<brand> Workbench user home for each human user contains that user's unique configuration files.	~ (unique to N4)
sharedfolder	N/A	station_home	C:\Users\userName\Niagara4.x\tridium\shared	^ (as in NiagaraAX)
stations folder	N/A	protected_station_home	C:\ProgramData\Niagara4.x\tridium\stations\<stationName>	^^ (unique to N4)
N/A	User Home	niagara_user_home	C:\ProgramData\Niagara4.x\<brand> 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 N4)

Homes on a controller

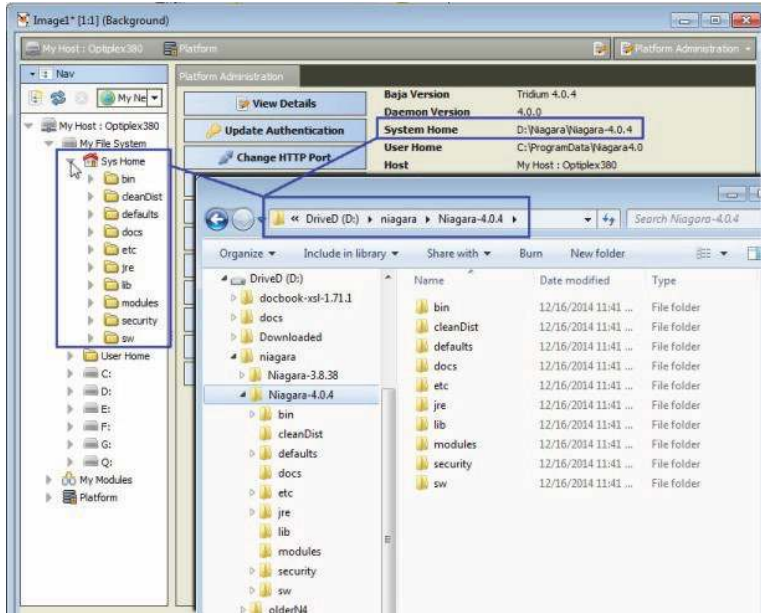
On a controller there are two homes.

Home in the Platform Administration view	Home in the Platform Administration view	Niagara 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 NiagaraAX)
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 N4)

System home

Sometimes identified by its alias, `niagara_home`, the system home is the sole location to which the Workbench installation wizard and platform commissioning wizard install Niagara runtime components, such as core software modules, the JRE, and binary executables. License files and license certificates reside in the Workbench system home, under the `security` subfolder. A system home contains no configuration files that can be changed by a user. Except when it is time to upgrade, these runtime files are read-only.

Figure 2 Example Sys Home (niagara_home) on Supervisor platform



The example above shows the file system for Niagara 4 Supervisor running on a Windows PC. In the Nav tree, the system home folder is referred to as its Sys Home. In the Platform Administration view, the same system home is referred to as System Home.

In the example, the actual location of the system home folder on this PC is: `C:\Niagara\Niagara-4.0.15`.

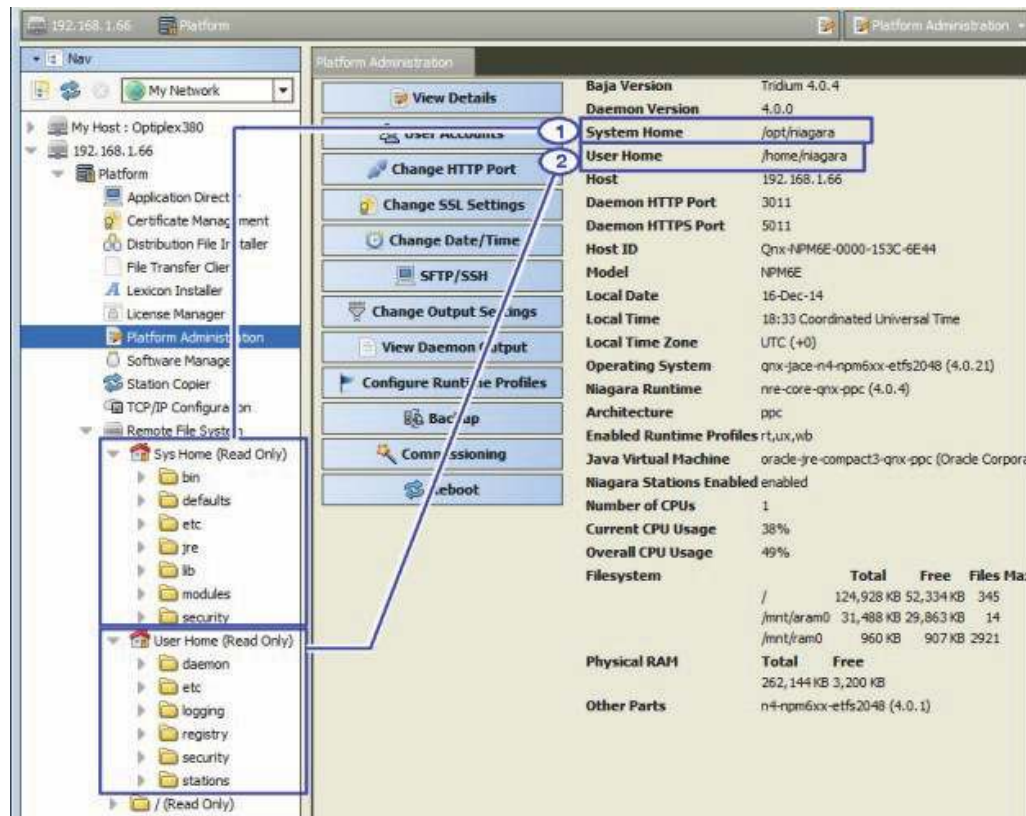
The system home on a controller appears as System Home in the Platform Administration view. The actual location of the system home folder for a controller is: `/opt/niagara/opt/niagara`.

Controller user home

The user homes are the locations under which all configurable data reside. Included are stations, templates, registry, logs, and other 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 Niagara 4.

The JACE controller has one system home and one user home.

Figure 3 JACE SysHome (niagara_home) and User Home (niagara_user_home) locations



Callout 1 above identifies a controller's system home (alias: niagara_home) in both the Nav tree and the Platform Administration view. In the Nav tree, you can find the controller's system home by expanding Platform→Remote File System. The actual folder for the system home is /home/niagara.

Callout 2 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.

Windows platform 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: Workbench User Home (for people), and a platform daemon User Home (for the daemon server processes).

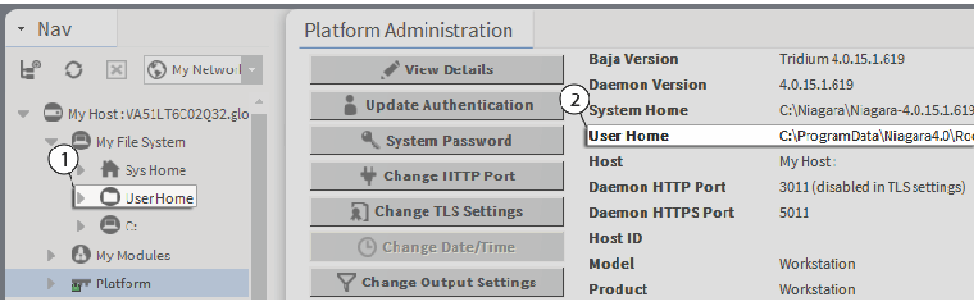
The Supervisor or engineering workstation that is licensed to run a station has a daemon user home. The daemon is a server process and represents a (non-human) user that manages the Supervisor's running station. The Supervisor's daemon user home contains daemon-specific configuration information. The actual location of the Supervisor's daemon user home is C:\ProgramData\Niagara4.0\distech. The platform daemon is installed to this location and started from this location as a Windows service.

In addition to this daemon user home, a Windows host has a separate Workbench user home for each person (operator, administrator) who logs on with credentials to a Windows-based platform licensed for Workbench, meaning a Supervisor or engineering workstation. Any given PC or workstation has at least one, and may contain multiple Workbench user homes.

Each person's Workbench 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 Workbench 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 Niagara4.0 folder under your Windows User account.

If you open a local platform connection to your Supervisor PC, expand My File System in the Nav tree, and go to the Platform Administration view, you can see both types of user homes at the same time.

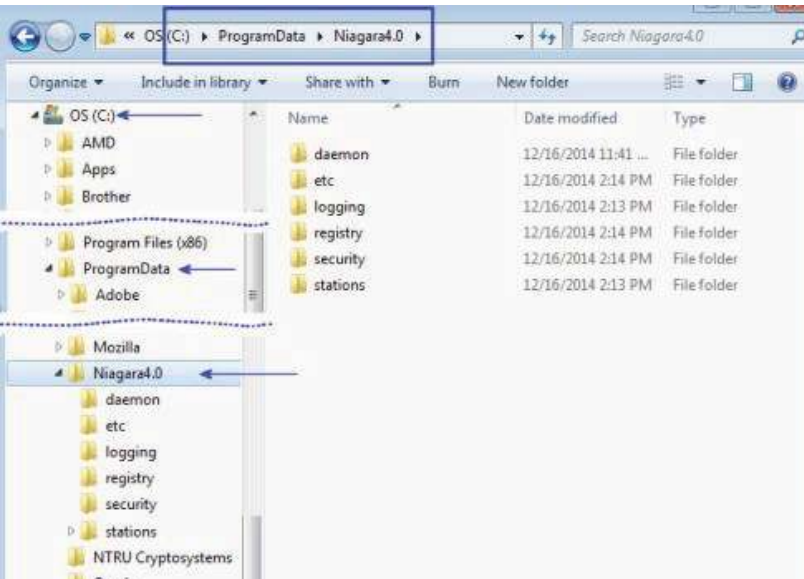
Figure 4 Local platform connection to a Supervisor station with Workbench and daemon user homes



- Callout 1 identifies Workbench User Home.
- Callout 2 identifies the daemon User Home.

When you first install Niagara 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 5 Example of a daemon User Home location in Windows Explorer



You can do this station copy in different ways. In Niagara 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 Workbench user home locations are:

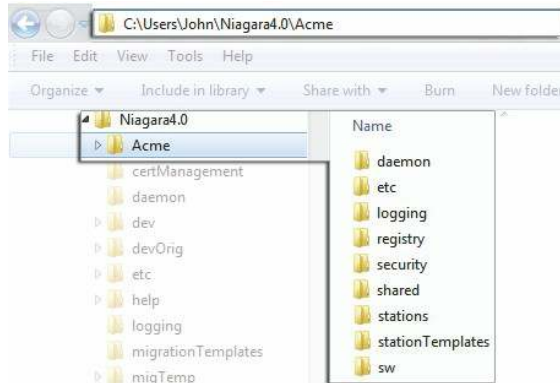
`C:\Users\John\Niagara4.0\distech` `C:\Users\Mike\Niagara4.0\distech`

where "John" and "Mike" are separate Windows user accounts. The first time a Windows user starts Workbench, the system automatically creates that user's unique user home folder.

- The person that installs Niagara 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 Workbench user home on the platform.
- However, if another person logs on to Windows on that computer and starts Workbench, that user also acquires their own Workbench user home.

The following figure shows an example Workbench user home location in Windows Explorer.

Figure 6 Example of an automatically-created Workbench User Home in Windows Explorer



Station homes

Niagara 4 uses the Java Security Manager to protect against malicious from accessing station or platform data and APIs. The Security Manager uses signed policy files that specify the permissions to be granted to code from various sources. Included are tighter controls about which applications have access to parts of the file system. Two folders under the Workbench 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 Niagara 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 protected station home. For this reason, in Niagara 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.

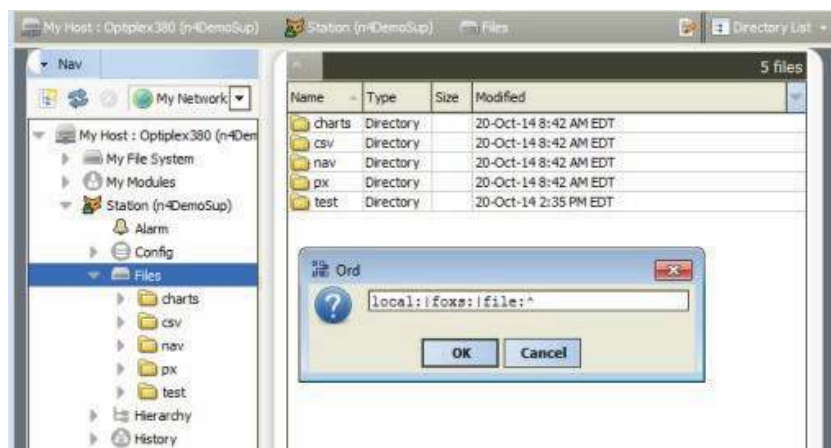
The alias `station_home` retains the same file ORD shortcut (^) as used in NiagaraAX—only in Niagara 4 this points to the station's shared sub-folder.

Figure 7 Example NiagaraAX station file folders compared to Niagara 4 station file folders

AX station folder	N4 station folder
<ul style="list-style-type: none"> stations <ul style="list-style-type: none"> demo <ul style="list-style-type: none"> alarm dev frames history html httpd images nav niagaraDriver_nVirtual public px secure tree demoToN4 	<ul style="list-style-type: none"> stations <ul style="list-style-type: none"> demo <ul style="list-style-type: none"> demoToN4 <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 figure above, comparing an AX station file folder structure (left side) to the same station migrated to Niagara 4, a number of folders are now 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 also write to the shared sub-folder.

Figure 8 File ORD for the station home in Niagara 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 Niagara 4, the New Station Wizard tool finishes with an option to copy the station from the station home (the location for each new station) under your Workbench 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 Workbench 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 Workbench User Home. Or, you can use the platform Station Copier to copy the station back to the `stations` folder of your Workbench User Home.

NOTE: Using the Station Copier to copy the station back to your Workbench 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 Workbench 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 Workbench User Home

Instead of running a station out of the daemon User Home, you can run a station directly from your Workbench User Home (outside of normal platform daemon control).

You do this using the Niagara 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.

If multiple people log on (differently) to Windows on Niagara 4 host and use Workbench, each person has their own separate Workbench User Home.

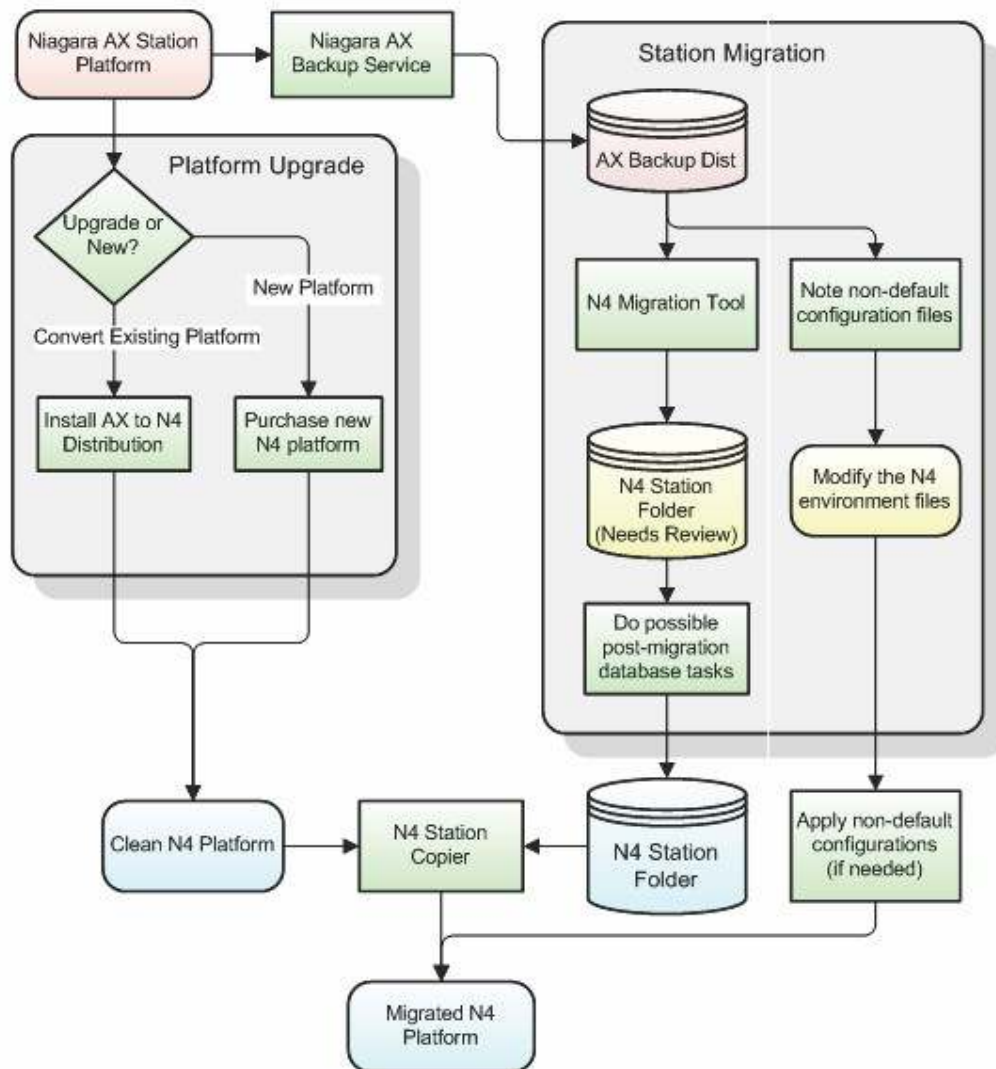
Windows users require permissions to access other users' files; yet it's possible that different users of a system (Supervisor or engineering workstation) may need to share items such as station backups, station copies, saved template files, and so on. Such items may be in multiple Workbench User Home locations in Niagara 4 (unlike in NiagaraAX where a single `!\backups` or `!\stations` folder applies to all users).

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 Niagara 4 Windows PC. For example, you might create a shared folder under the Niagara 4 Sys Home location (Workbench User Home is not shareable).

Migration workflow diagram

Basic steps in migrating your NiagaraAX installation are included in the following diagram.

Figure 9 Migration steps diagram



Red items indicate NiagaraAX artifacts. Blue items indicate Niagara 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 Niagara. 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 21.

Migration workflow process

Based on an existing NiagaraAX job with a Supervisor and some number of JACE controllers, the workflow process for migration to Niagara 4 may look similar to below:

Assess and prepare for migration

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

1. Identify those platforms to be migrated. JACE controllers must be running at least NiagaraAX 3.8.nn, and be compatible with N4.0. See [Platform compatibility](#), page 7 and [Platform resource requirements](#), page 8 for details.
2. Review the current drivers and applications running on the JACE controllers to be converted. See [Driver and application support](#), page 11 for details.

If any drivers or applications are unsupported in N4.0, but are critical to retain, you can opt to leave those controllers running AX-3.8, and still integrate them with Niagara 4 Supervisor station. However, you can- not integrate a Supervisor at AX-3.8 with Niagara 4 JACE controllers.

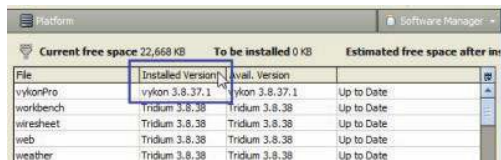
NOTE: For any job that will have a *mix* of Niagara 4 and AX-3.8 platforms, refer to the section [Wire compatibility](#), page 12.

3. For each AX-3.8 platform to be migrated, identify custom or third-party software modules (non-Tridium) that it uses. Prior to upgrading, arrange to have those modules refactored as Niagara 4 modules. Contact the custom module vendor to arrange this.

NOTE:

You cannot migrate a station to N4 until all of the modules it uses have been refactored for Niagara 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.



File	Installed Version	Avail. Version	
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

- a. Click the "Installed Version" column header to re-sort, as shown above. Scroll down as necessary. Any vendor-sourced modules used that do *not* appear listed starting as *Tridium*¹ must be refactored by the module vendor before the station can be migrated to Niagara 4. This assumes that all modules installed in the JACE are required by the controller's currently running station. Any installed module that is *not used* by the station is inconsequential in for migration.
 - b. 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.
4. If you will synchronize users from the N4 Supervisor to the 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 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 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.)
 5. Make an N4.0 license request for all platforms to be migrated. You need confirmation that these licenses are ready before migrating the platforms (AX licenses do not work in Niagara 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 AX-3.8 license files as well.

In EMEA markets, a few "niche drivers" listed as vendor *Tridium* also require module refactoring before migration, meaning Niagara 4 module versions may not be ready at release time. Examples include modules `globalCache`, `protec`, and `resolveBus`.

Backup stations and migrate databases

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

Using Workbench, make a station *backup* of each JACE 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 N4.0 station copies.

Sometimes, post-migration database "cleanup" may be needed. You should always do a preliminary check-out of each migrated station.

Upgrade the Supervisor

You migrate the Supervisor first.

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

Convert controllers and install new stations

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

Before converting JACE controller from AX, you must successfully migrate its station to N4, including an off-line checkout of it using Workbench. Only then should a conversion to N4 be done.

Perform system cleanup

Following the upgrade of all AX-3.8 platforms to N4.0, including installation of migrated stations, there may be several areas that need attention.

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 section provides migration tasks, along with reference details as needed.

Preparing platforms formigration

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

Export certificate Trust Store and Private Key Stores

If JACE controllers are configured to use secure communications with signed certificates, export the keys from the Trust Store and the Private Key store. These are deleted from each JACE during migration to N4. After migration, you can import these certificates back using Workbench and a platform connection.

Exporting the certificate Trust Store and Key Stores

- Step 1 Using Workbench, access the Certificate Management view of the JACE. To do this, either open a platform connection and go to the Certificate Management view, or else open the station and double-click on CertManagerService (under Config > Services > PlatformServices).
- Step 2 On the Key Store tab, select the certificate and click Export. Enable the checkbox to export the private key. Use an encryption password to protect the private keys in this file.
- Step 3 On the Trust Store tab, select the certificate used, and click Export. Typically this only needs to be done once (if used in every JACE).

NOTE: In NiagaraAX, it is not necessary to export all the default Trust store certificates. It is only necessary to export any additional certificates that were imported here.

- Step 4 Repeat steps above for each JACE to be migrated.

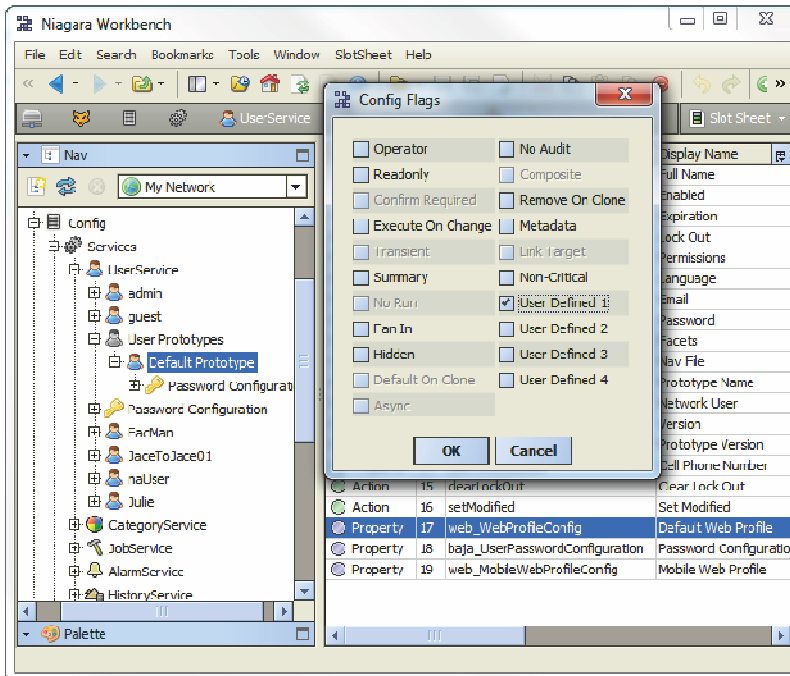
Make sure exported certificates files will be available to the Workbench PC to use later. For related details, refer to the *Station Security Guide*.

Configuring a station to migrate users

If your N4 users will use the HTML5 prototype, which does not exist in AX, enable the `User Defined 1` configuration flag on the `web_WebProfileConfig` property of each user prototype in your AX stations before running the migration tool.

Prerequisites: You have Workbench 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.

Kerberos keytab file notes

If your AX-3.8 stations use Kerberos authentication for user services (for example, `LdapV3ADUserService`) *and* 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 AX-3.8 station.

NOTE: If applicable, do this before making source AX-3.8 station backups for use with the migration tool.

Relocating the Kerberos keytab file

Prior to migration you need to relocate the Kerberos `keytab` file in each AX-3.8 station.

Step 1 Using Workbench, 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 Workbench, 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 Workbench 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 JACE station to be migrated to N4.

If a local AX-3.8 station such as a Supervisor, you can simply 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 operation in AX-3.8. Successful operation should continue after the station is migrated to N4.

Making source backups

For any AX-3.8 platform you are migrating, make a station *backup* using Workbench. 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 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 Niagara 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\Niagara-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

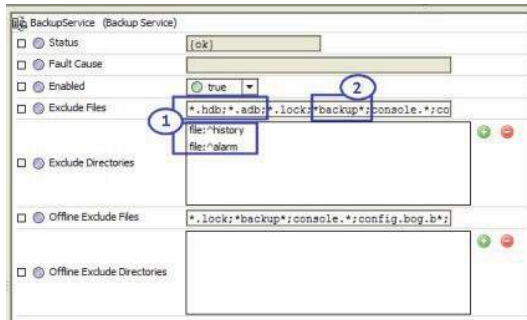
Prior to making a station backup, set properties as needed in the running station's BackupService. By default, backups exclude both the history and alarm databases. Other items are also excluded, such as prior saved station database copies, console outputs, and lock files. In most cases, you want to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

Prerequisites:

Stations are configured and running on a host compatible with Niagara 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 Workbench, open the station and access the BackupService property sheet.



Related BackupService properties shown above are:

a. In the BackupService for each 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 N4.0 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 JACE station.

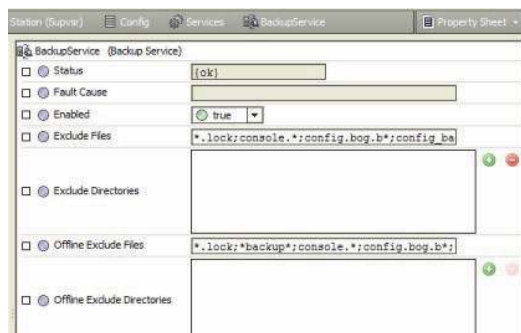
Figure 10 Example BackupService properties in JACE station



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 11 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 N4 Supervisor).

Making source station backups

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

Prerequisites:

The station's BackupService is configured appropriately.

Step 1 Using Workbench, 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 JACE 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 rename 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 Niagara 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 AX-3.8 stations to be migrated.

Secure license files

Any converted platform (controller, Supervisor, or workstation) requires Niagara 4 license file, as existing AX license files are *incompatible*.

- Before upgrading licenses, *back up* existing NiagaraAX license files for Windows hosts, and store them in a safe place.
- You can perform migration work on all archived AX-3.8 stations without all N4.0 licenses secured, provided that you have at least *one* N4.0 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 Niagara 4 license is not backward compatible with any NiagaraAX install.

- A controller, converted to N4.0 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 Niagara 4 (N4.0) 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 N4.0 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 Workbench. For related details, refer to the Niagara 4 Installation Guide.
- When you receive a new N4.0 license file for your Supervisor, do not copy it back to any *NiagaraAX* home's `licenses` folder on the same PC—otherwise Workbench no longer works. This can also occur if NiagaraAX host downloads the (N4) license offered by the licensing server.

In hybrid systems that combine both N4.0 and AX-3.8 hosts, you need to *retain* Workbench to use for possible station modifications to controllers that remain at AX-3.8.

Run the migrationtool

Included with Niagara 4 Supervisor or engineering workstation is a command line utility that you run from a console. This utility is known as the N4 migration tool or migrator.

About the migration tool

The migration tool takes a source file, typically AX-3.8 station backup `.dist` file, and outputs a migrated N4.0-compatible station folder. You can then use Workbench to open a platform connection to the target

platform (for example, a converted controller), and install this station using the N4.0 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 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 Niagara 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 back- up file. The migrator ran in a standalone Niagara Console window.

NOTE: Running the migrator in the embedded Workbench Console is not supported, as it does not properly process the necessary user inputs that may be required if your NiagaraAX station contains passwords that need to be encrypted with a pass phrase.

Figure 12 Example migration tool processing a Supervisor station backup

```
-----
c:\niagara\niagara-4.0.15.0>n4mig C:\niagara\Niagara-3.8.38\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\niagara-4.0.15.0") 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

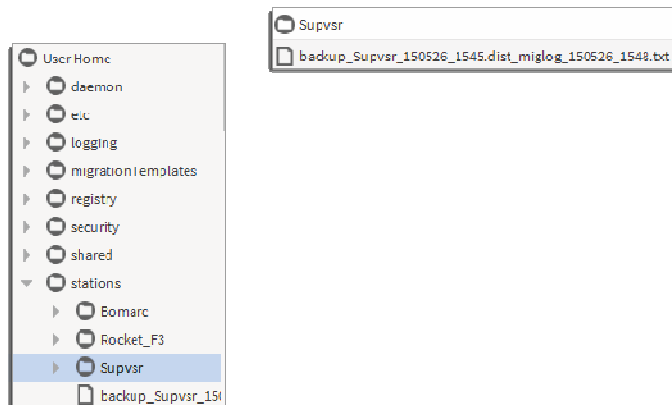
Completed Migration
Source: C:\Users\John\Niagara4.0\Rocket Industries\migTemp\Supvsr
Target: C:\Users\John\Niagara4.0\Rocket Industries\stations\Supvsr
Migration Report: C:\Users\jDoe\Niagara4.0\Rocket Industries\stations\Supvsr_miglog_1505261548.txt

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 Workbench User Home, as shown in the following figure.

Figure 13 Example migration tool output in Workbench User Home (station folder, migration log file)



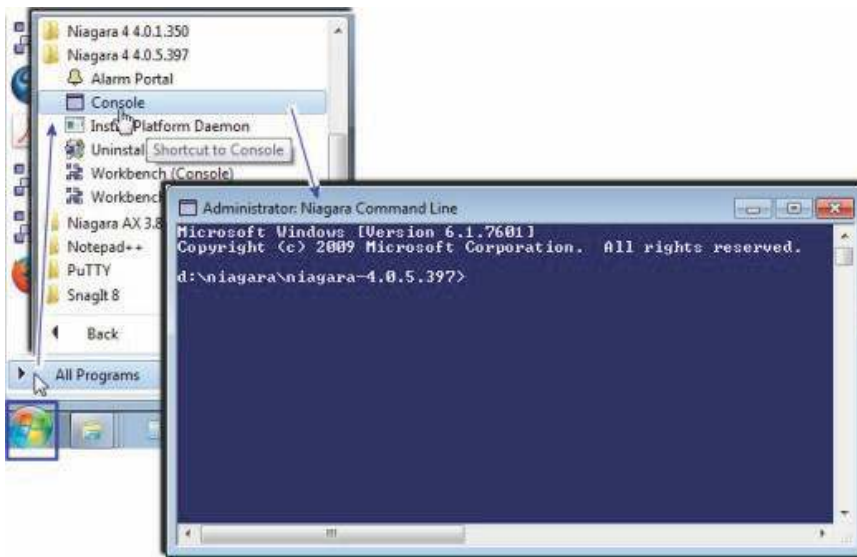
In this example, notice that the migration process created a `Supvsnr` folder and an associated log file in the `stations` folder. The target station name is automatically determined from the station name inside the station backup `.dist` file.

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.

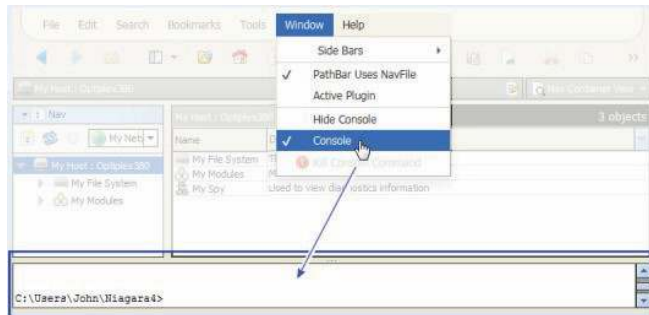
Console choice for migration

You should use only the Niagara console and not the embedded Workbench Console to run the migration tool.

For migration, use the standalone Niagara 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 Workbench (shown here), cannot be used for migration purposes.



Running the migration tool

With Niagara 4 installed on your PC, you run the migration tool on the AX-3.8 station backup `.dist` file to migrate a station, or run it on a saved AX-3.8 BOG file (`.bog`) if necessary. NiagaraAX 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 N4 (refactoring them) before the station can be successfully migrated. See [Migrating modules made with the `ProgramModule` component](#), page 35.

You must have access to saved AX-3.8 station backups, for migrating stations. See [Making source station backups](#), page 28.

Running the migration tool

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

NOTE: Do not use the Workbench 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: `\backups>`

The above would apply if you have both NiagaraAX-3.8 and Niagara 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 Workbench 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 Workbench 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 Workbench.

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 33. For general information, see [About the migration tool](#), page 29.

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 AX-3.8 station backup <code>.dist</code> 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 Niagara 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 <code>stations</code> subfolder of your Workbench User Home, that is <code>C:\Users\userName\Niagara4.x\<brand>\stations</code></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 <code>config.bog</code>. 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 <code>niagara_user_home/stations</code>.</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 35.</p>
Options		Description / Notes
-version		Print NRE version information for this Workbench instance.
-log:level		<p>Specify the log level, where level is one of the following:</p> <p>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 AX station as a result of the migration. This file is created in the root of the <code>stations</code> folder in your Workbench 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 Workbench 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> <p>1 = Controller migration template (for example, ControllerMigrationTemplate.ntpl)</p> <p>2 = Supervisor migration template (for example, SupervisorMigrationTemplate.ntpl)</p> <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 Workbench User Home, to store any customized migration station template files. See the Templates Guide for more details on working with Niagara 4 templates.</p>

Migrator syntax example 1

Here, the current directory of the Niagara console window has been changed to the backups folder of the AX-3.8 installation home, such that the prompt is \backups>

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

```
\backups>n4mig -t:1 backup_J300E_W_150210_1715.dist
```

The migration tool processes the J300E_Wbackup .dist file in the current directory, as JACE station, and the target output goes under the stations folder of your current Workbench 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\<brand>\stations\J300E_W (complete station folder)
- C:\Users\userName\Niagara4.x\<brand>\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 *re-name* 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):

```
\backups>n4mig -t:1 J3Wbu.dist
```

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

- C:\Users\userName\Niagara4.x\<brand>\stations\J300E_W (complete station folder)
- C:\Users\userName\Niagara4.x\<brand>\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 N4.

Migrator syntax example 2

Here, the Niagara console window has been left at the default N4.0 directory (say when you open it from the Niagara 4 Console Start menu shortcut), and absolute source and target directories are specified.

```
C:\Niagara\Niagara-4.1.xx.xx>n4mig -t:2 -log:finer \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 N4 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).

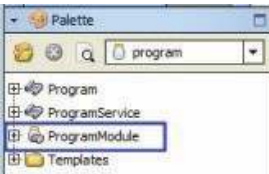
Therefore, when migrating a station from an AX-3.8 station backup .dist file, it is typically best to specify only the source .dist file name, and not specify any target. An example of this is described in [Migrator syntax example 1](#), page 34. After the migration tool completes, if needed, you can 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 Niagara 4 if you are migrating any AX-3.8 station that uses them. Otherwise, Program objects from them will be *deleted* in the migrated station.

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

Figure 14 ProgramModule in AX-3.8 program palette

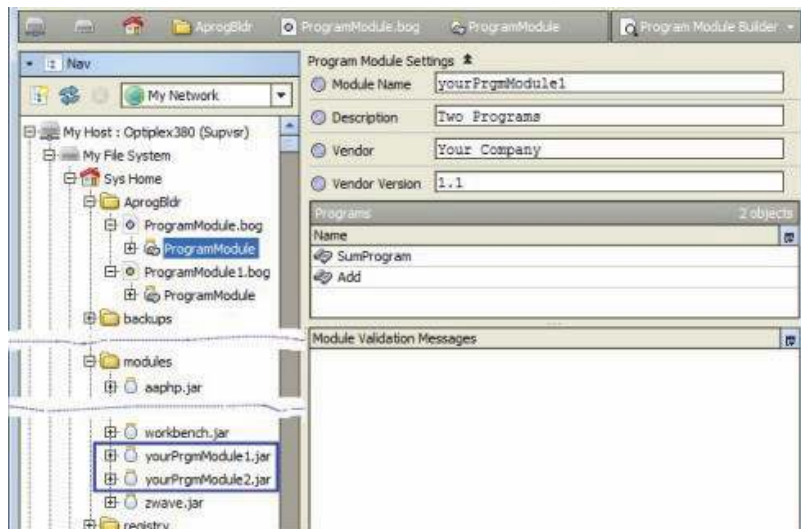


In NiagaraAX installation, there are different ways the ProgramModule feature can be used to create versioned module JAR files that contain one or more Program objects:

- ProgramModule components copied into a folder under the Workbench 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 15 ProgramModules saved as BOG files in the Workbench file space



The figure above shows two such ProgramModule BOG files, under an "AprogBldr" folder in the Workbench 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 Workbench `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 N4, unless you first use Workbench to copy ProgramModules from the `program` palette into a folder in the Workbench file space, and reconfigure again. Once saved as 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 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 Workbench on each migrated ProgramModule to recompile the contained Program(s) and then rebuild the module, which refactors it for Niagara 4.

Migrating modules from ProgramModules saved as BOG files

If you have saved one or more ProgramModule components in your Workbench file space, you can run the folder (BOG) file containing them through the N4 migration tool.

If necessary, you can do this on a PC still running an AX-3.8 Supervisor station, but with both Workbench and Workbench installed.

Step 1 At the Workbench PC, open a console. This can be either a standalone Console window or the Console available in Workbench. See [Console choice for migration](#), page 31.

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 \AprogBldr
```

for the `AprogBldr` folder as shown in the previous [Figure 15](#) [Figure 17](#) on page 33, page 36. Leaving the target unspecified (as here), results in a subfolder under stations in your Workbench User Home".

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

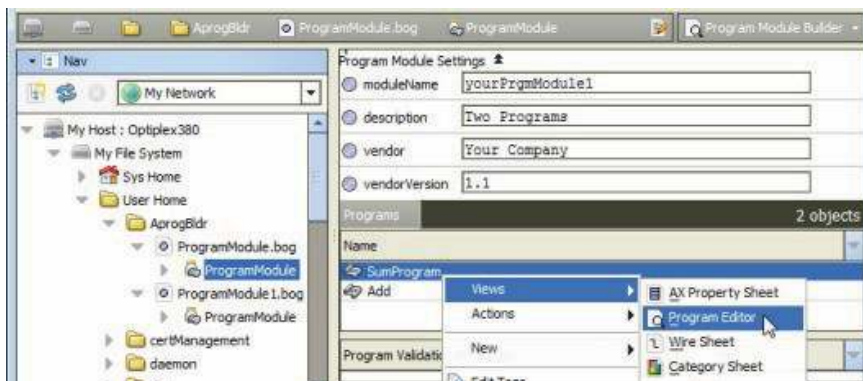
```

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 dependences to {nre=4.0;baja=3.8;program-rt=4.0;control-rt=4.0;}
16:40:40,770 INFO [migration.bog] modified ProgramModule dependences 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>

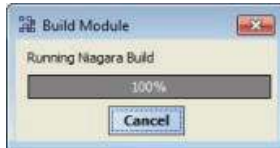
```

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 AX back- up .dist file).

- Step 3 If the target folder was created in your stations folder, move (cut and paste it) to another location in your Workbench 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.

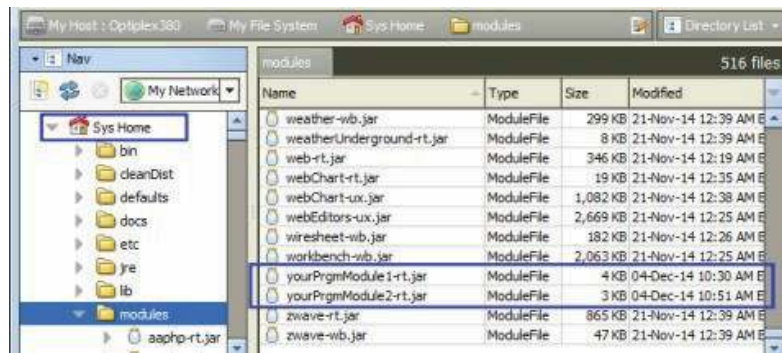


- Step 6 Right-click one of the contained Programs and select Views→ Program Editor (as above).
- Step 7 In the Program Editor, click Save & Compile (ProgramEditor→ Save&Compile). Activity occurs in the console area at the bottom of the Workbench window.
If successful, the Program status should be up-to-date, 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 Workbench 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 `yourPrgrModule1-rt.jar` and `yourPrgrModule2-rt.jar` were created by the two ProgramModules.

- Step 12 When finished migrating modules, close and restart Workbench. This is necessary to update the registry. Modules should now be ready when running the N4 migration tool on a backup `.dist` file of any AX-3.8 station that uses them. See [Running the migration tool](#), page 32.

Migrating modules from ProgramModules in a station database

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

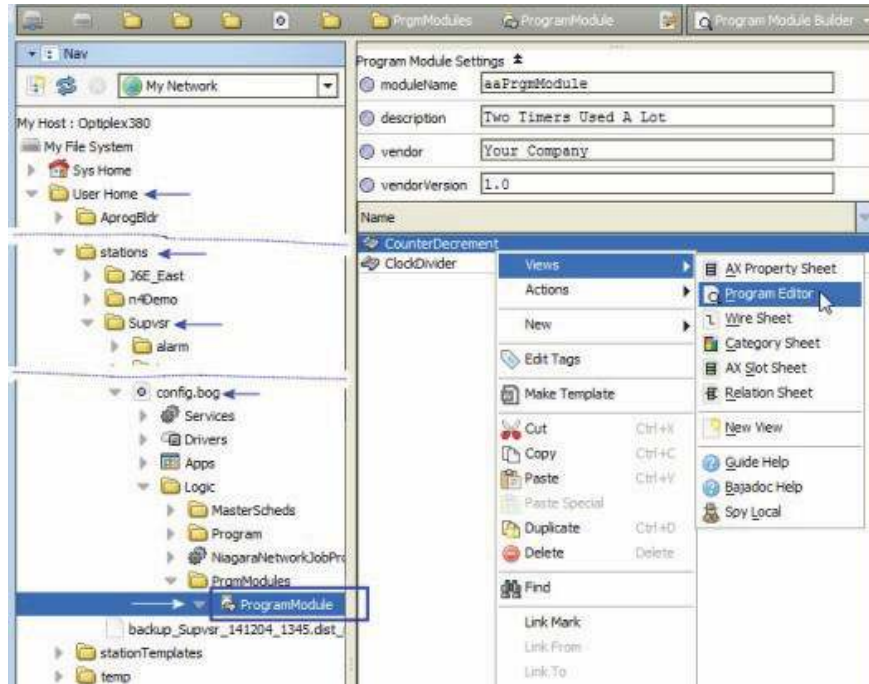
Prerequisites:

However, before installing that migrated station, or any other migrated station that uses modules produced by it, you must use Workbench 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 AX-3.8 Supervisor station, but with both Workbench and Workbench installed.

- Step 1 Run the migration tool on the saved AX-3.8 station backup `.dist` file.
- Step 2 In the Workbench 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 Builder.



Step 5 Right-click one of the contained Programs and select Views→ Program Editor (as above).

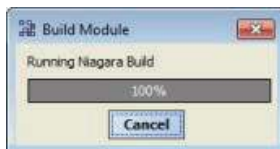
Step 6 In the Program Editor, click Save & Compile (ProgramEditor→ Save&Compile). Activity occurs in the console area at the bottom of the Workbench window.

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 41.

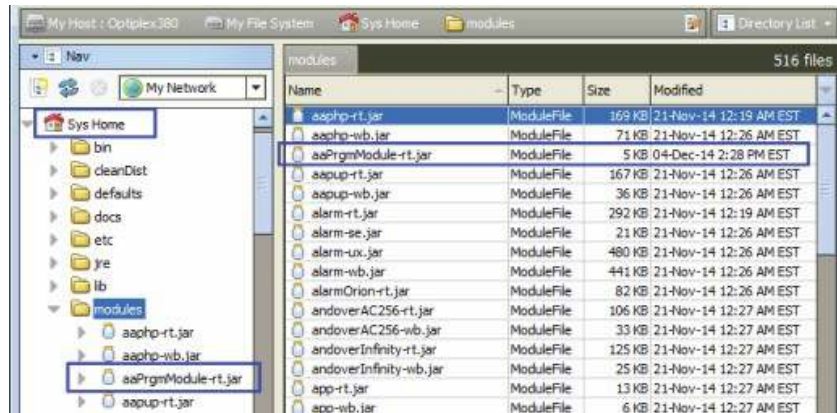
Step 7 Repeat steps 5 and 6 for all Programs in the selected ProgramModule, so that *all* Programs have been successfully compiled and saved.

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 Workbench 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 Workbench. This is necessary to update the registry. Migrated modules should now be ready when running the migration tool on 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 N4 target. The typical source is AX-3.8 station backup `.dist` file. The output (target) is a station folder, in your Workbench 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:\backups\backup_Supvsr_141117_1707.dist
```

```
Target:C:\Users\John\Niagara4.x\<brand>\stations\Supvsr
```

You can then install this migrated station into the daemon User Home of Niagara 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 AX platform's `!lib` folder, are not migrated. If you know of any that were edited from default values, and they still apply to N4 platforms, you may need to re-apply changed values in the platform after upgrading to Niagara 4 (however, do not copy entire AX files into the N4 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 AX but was removed for N4. In this case the element will be removed from the station.
- In the case where the N4 equivalent module exists, the registry check provides a converter that can convert the obsolete type into the appropriate N4 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 N4, 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 N4 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 Workbench. For example, certain Program objects may need to be changed and recompiled in Niagara 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 NiagaraAX Program objects (components) in the station to be compatible with Niagara 4, flagging any that fail to compile properly in the migration report, or log file.

Using Workbench, 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 Workbench. Access to Developer-level Niagara 4 documentation may be needed.

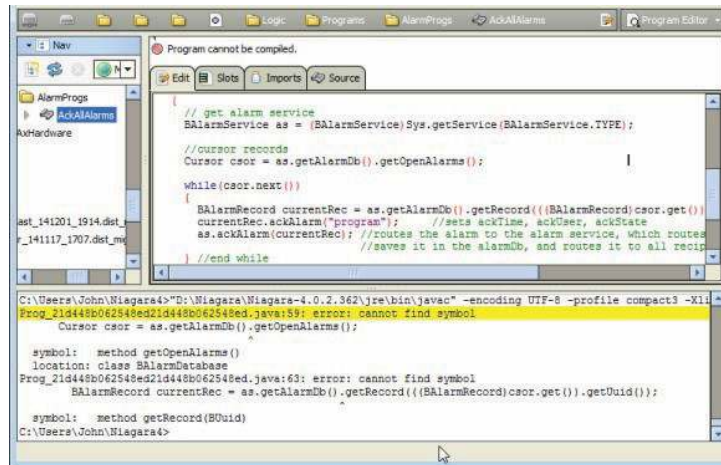
Modifying and recompiling Program objects offline

You can open a station *offline* in Workbench 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 AX-3.8 Supervisor station, but with both Workbench and Workbench installed.

- Step 1 Open Workbench, 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(BOrd)` 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 Niagara 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 Niagara 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 N4 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 in order 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 Workbench, 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 Niagara 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 their 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 Niagara Role is created for each AX User in the station. If the 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 N4 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.
- Tagged Categories
Consider using Tagged Categories for providing access to oBIX client Users; this allows for highly configurable permissions for various station components.

Preliminary checkout of migrated stations

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

As for migrated JACE stations, if you have an extra, available JACE 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 N4.0 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 (N4.0) platform daemon" when first installing the Niagara 4 software. This lets the AX-3.8

platform daemon continue to host the Supervisor station while you use the migration tool. It also lets you become familiar with Workbench, and yet still have Workbench available for interim tasks.

NOTE: When installing Niagara 4 software on the Supervisor PC, be sure to select the option "This instance of Niagara 4 will be used as an installation tool". Refer to the Niagara 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 Niagara 4 on the Supervisor PC and starting Workbench, open a platform connection to the AX-3.8 Supervisor station and stop the station.

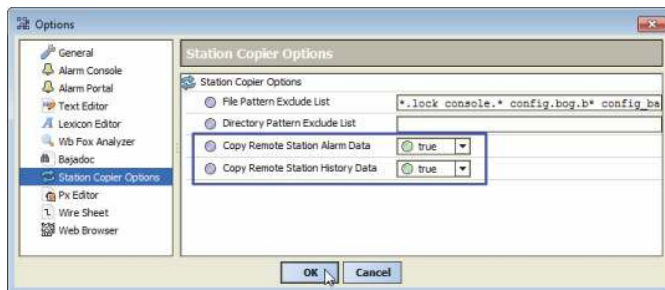
Step 2 At the N4.0 Supervisor PC (with access to N4.0-migrated station folders), install and start the N4.0 platform daemon.

For example, from the Start menu, select All Programs > Niagara 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 Workbench (if not already started) and click Tools→Options→ Station Copier Options.

The Options window opens.



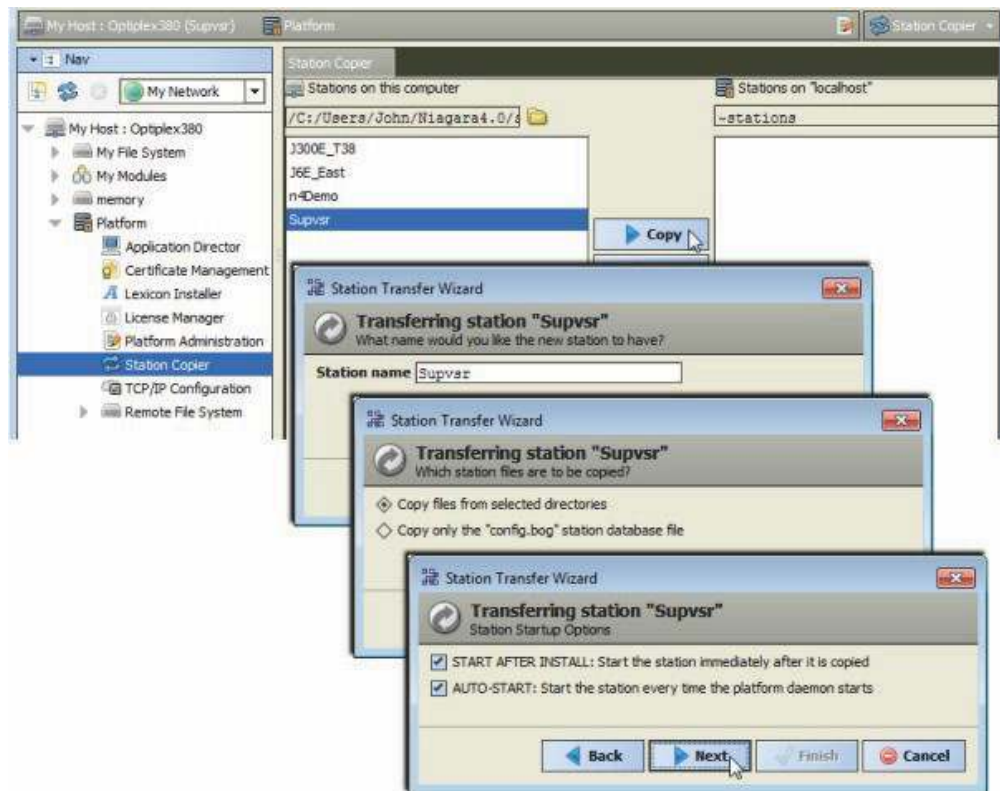
Step 4 Set **Copy Remote Station Alarm Data** and **Copy Remote Station History Data** to `true`, and click **OK**

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

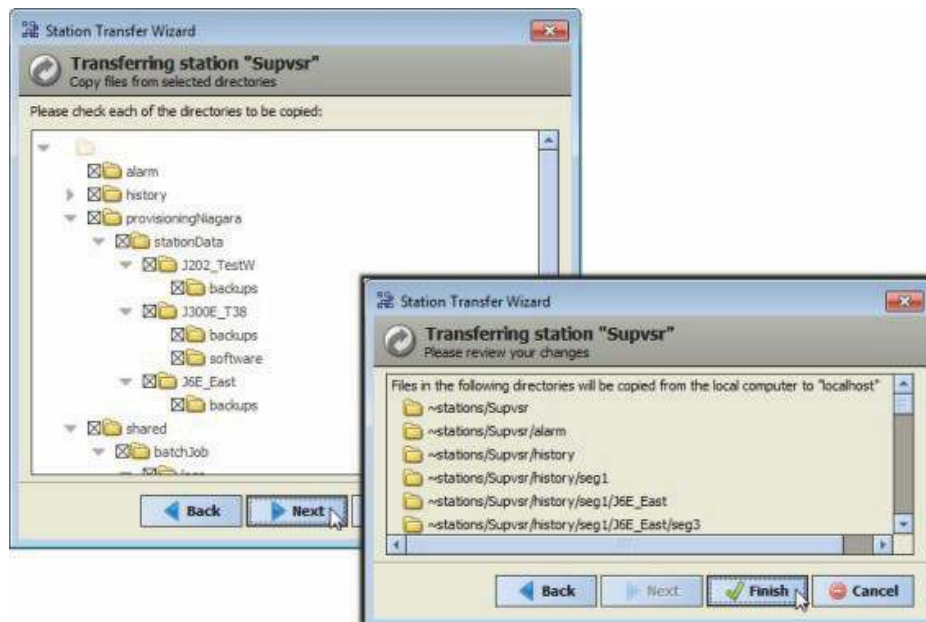
Step 6 After platform login, go to the Station Copier platform view.

Ready-to-install migrated stations are on the left side, in your Workbench 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\user-Name\Niagara4.x\<brand>\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*



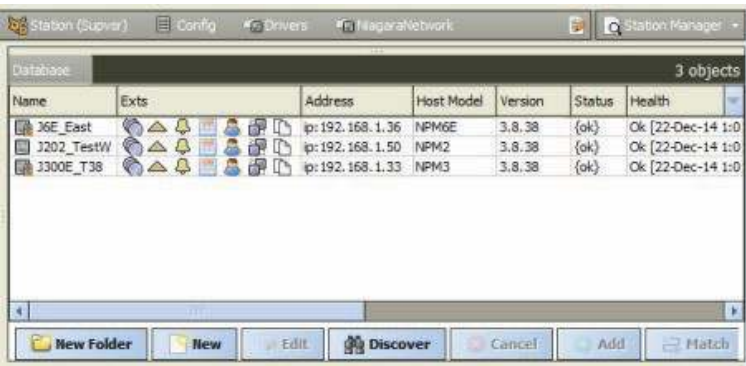
Step 9 Select all station folders to be transferred and click Next→Finish.

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 16 Network Manager view



The figure above shows N4.0 Supervisor's NiagaraNetwork shortly after migration.

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

Save any changes make before upgrading your JACE controllers.

Controller conversion

Typically, after upgrading a Supervisor, you convert JACE controllers to N4.0—or at least you convert all those for which you *successfully migrated a station*. During the N4 commissioning of each JACE, you install this station as one of the steps in the Commissioning Wizard.

CAUTION:

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

Converting a controller

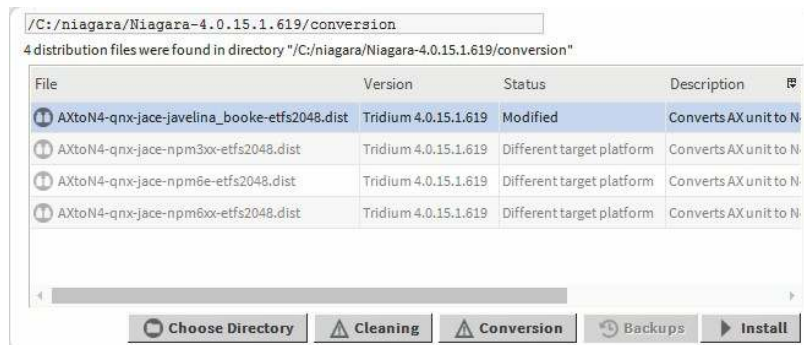
Perform the following procedure for each controller for which you have secured N4.0 license, and have successfully migrated its station database.

Prerequisites:

Workbench 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: N4 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 Niagara 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.

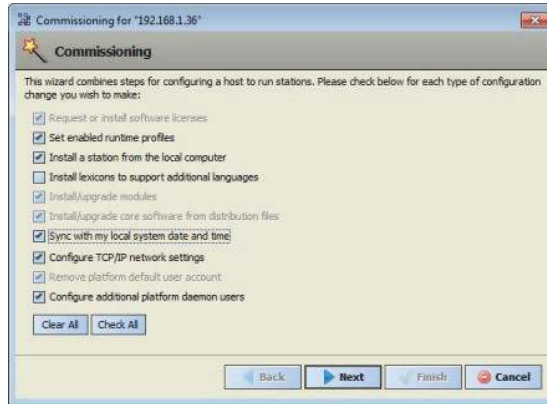
NOTE: Installation clears *all data* in the controller, including AX licenses and certificates, environment and properties files, SSL certificates and key files, along with all station data and NiagaraAX software. The controller's IP configuration is retained. You must use the default platform credentials 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 N4, platform credentials require a stronger password than in 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:

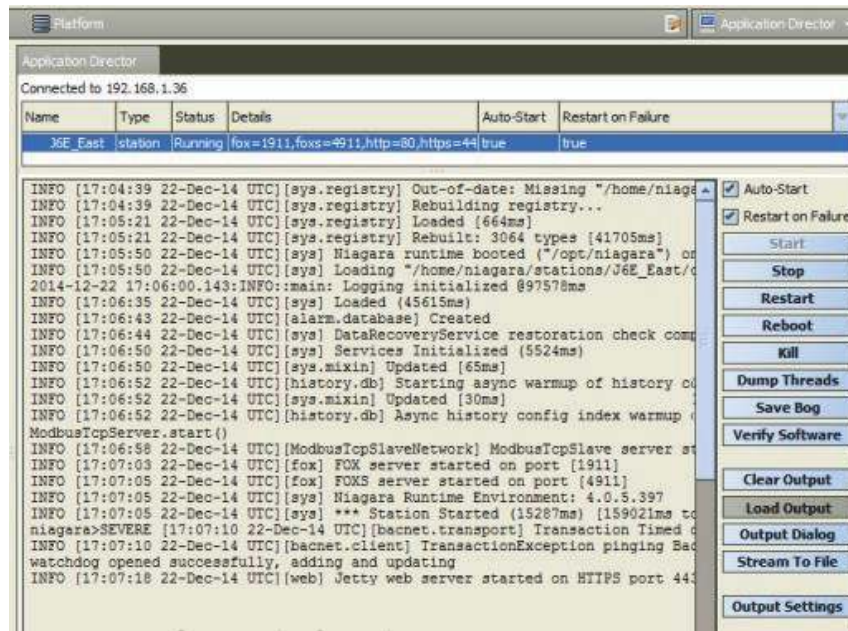
For this platform session, Workbench remembers the first platform admin user you created to replace the factory default platform user. This simplifies re-connection after the controller reboots.

For related details, see the JACE Niagara 4 Install and Startup Guide. Example screen captures of a *few* of these Commissioning Wizard steps are shown in the [Controller conversion examples](#), page 51

- 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.

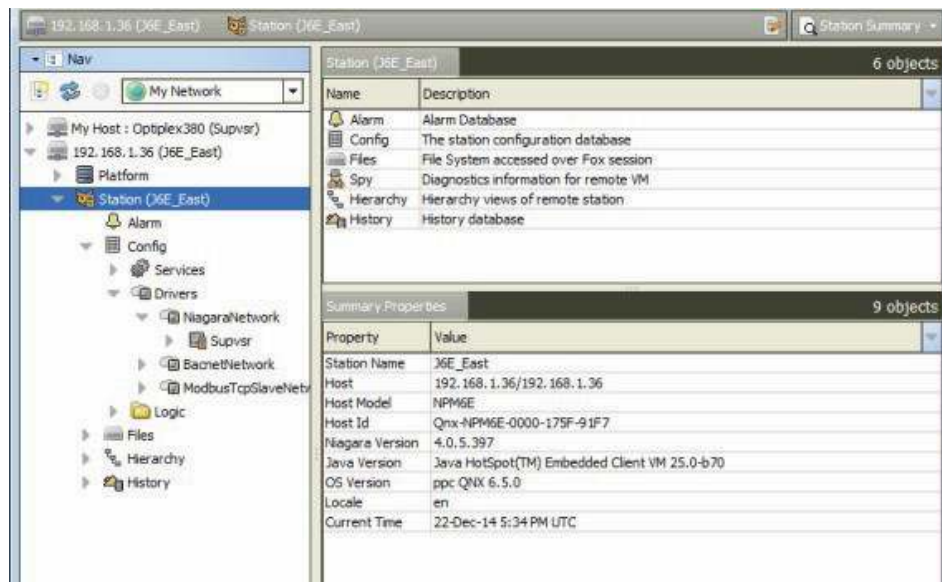
CAUTION: Do not remove power from the controller during this reboot, which may take up to 7 or more minutes to complete. Removing power could easily cause the unit to become *unrecoverable*. If desired (and convenient), you can use a serial shell connection to the controller to monitor progress as files 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 Workbench, 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 N4.0 platforms, as well as the Supervisor station.

Controller conversion examples

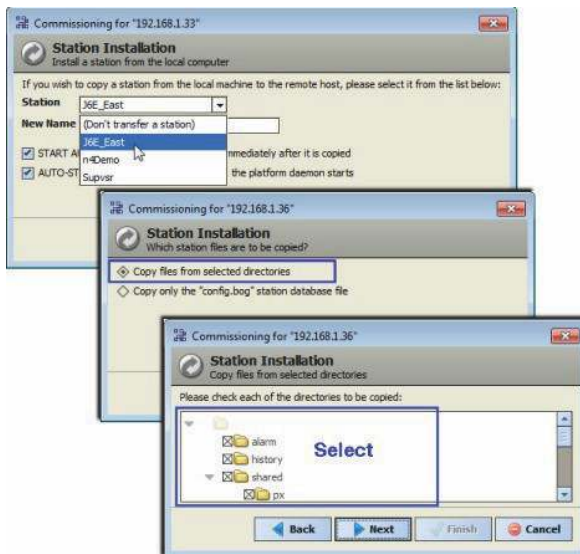
Following, are example Commissioning Wizard windows when converting JACE controller to Niagara 4.

Example conversion steps

Figure 17 Example “Set enable runtime profiles” step



Figure 18 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 19 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 JACE that *no person uses* to make platform connections, but is only referenced back in the N4 Supervisor.

Downgrading controllers

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

NOTE: Controller downgrade from Niagara 4 removes all data except the controller's Ethernet TCP/IP configuration. Platform credentials revert to factory defaults. The Niagara 4 license is not be retained and NiagaraAX 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 NiagaraAX 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 Workbench, 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 N4 `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 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 Workbench (port 3011) and the *factory default* credentials.

NOTE: At this point, the JACE will be at 3.8.38 version. If you need an earlier release AX version, use the clean `.dist` files in the AX-3.8 Distribution File Installer. This will revert the JACE 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 AX-3.8 controller, see the JACE Niagara 4 Install and Start-up Guide.

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 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 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 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 Niagara 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 Niagara 4. It may not even be necessary to make the same changes, so you should evaluate once you have the station running in Niagara 4.

NOTE: Do not copy entire files from the NiagaraAX platform to the Niagara 4 platform, but instead only those values that were modified from defaults in the equivalent NiagaraAX file(s).

Verify station-to-station communications

Ideally, station-to-station communications between the platforms converted to N4.0 and all others resume after the upgrade. That is, communications between the Supervisor and controller stations, via the Niagara-Network. Supervisor communications between any JACE stations which remain at 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 N4.0 Supervisor stations and/or controller stations (N4.0 JACEs, and if applicable, AX-3.8 too).

NOTE: If configuration of any AX-3.8 JACE station is required, say to edit "client connection" properties of the NiagaraStation that represents the N4.0 Supervisor, need to use Workbench for this.

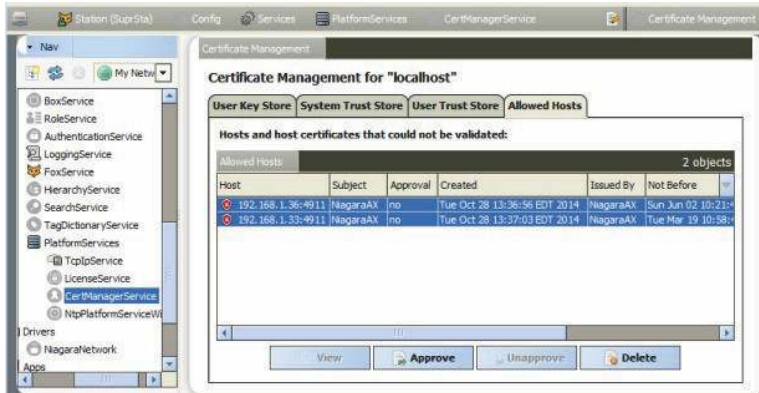
Some variation of the following sequence may be required:

1. In Workbench, re-open the Supervisor station and go to the Station Manager view of its NiagaraNetwork.

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

- 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.

- In a station connection to each JACE 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 JACE station back to the Supervisor (and possibly other JACE stations) you may need to go to the JACE station's CertManagerService platform service, and perform other actions.

- While working in remote JACE 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.

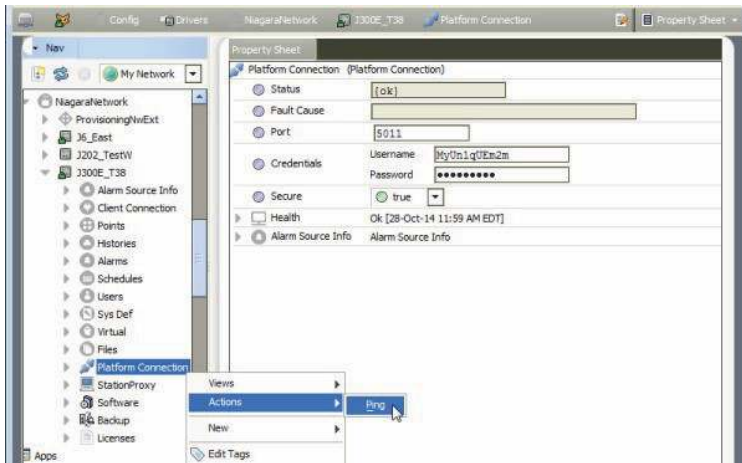
Provisioning notes

For any Supervisor migrated to N4.0 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 JACE 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.

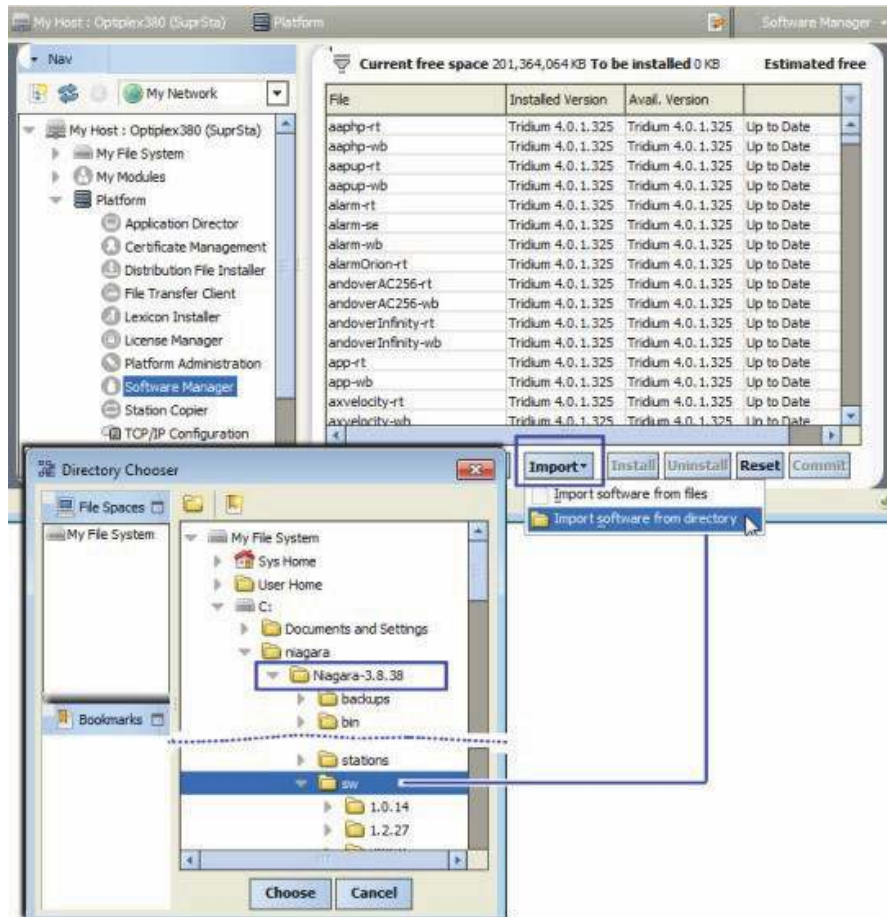
Provisioning considerations when JACE controllers remain

Sometimes one or more JACE controllers may need to remain running AX-3.8, yet continue to be in the migrated Supervisor's NiagaraNetwork. This is supported by the N4.0 Supervisor, including *many* provisioning operations. For example, the backup of AX-3.8 stations and installation of AX-3.8 software.

For proper support, you should *import* the software database of the former AX-3.8 Supervisor into the software database of the N4.0 Supervisor. This copies over all the versioned NiagaraAX software modules and .dist files to the new Supervisor, and makes them available for provisioning of the older platforms.

To do this when the N4.0 Supervisor is installed on the same PC as the former AX-3.8 Supervisor:

1. Use Workbench to open a *platform connection* to the N4.0 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 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 N4.0 Supervisor's software database. When finished copying, the normal Software Manager view opens.

If the former AX-3.8 Supervisor is on a *different* PC than the N4.0 Supervisor, you perform the same procedure. However, first copy the entire software database (`! \sw` folder) from the AX-3.8 Supervisor to a location accessible from the N4.0 Supervisor, such as on a USB flash drive, etc. Then navigate to that location from the Software Manager while platform-connected to the N4.0 Supervisor.

Station User notes in migrated stations

A number of Niagara 4 changes were made affecting station users, meaning User components that are children of a station's `UserService`.

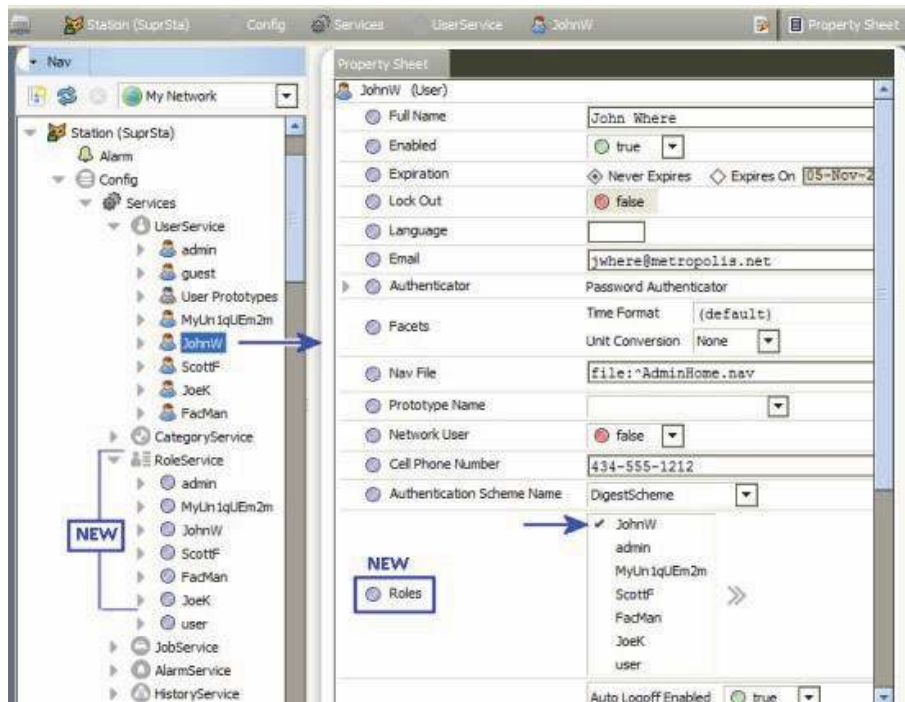
Permissions moved to Roles

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 Niagara 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.

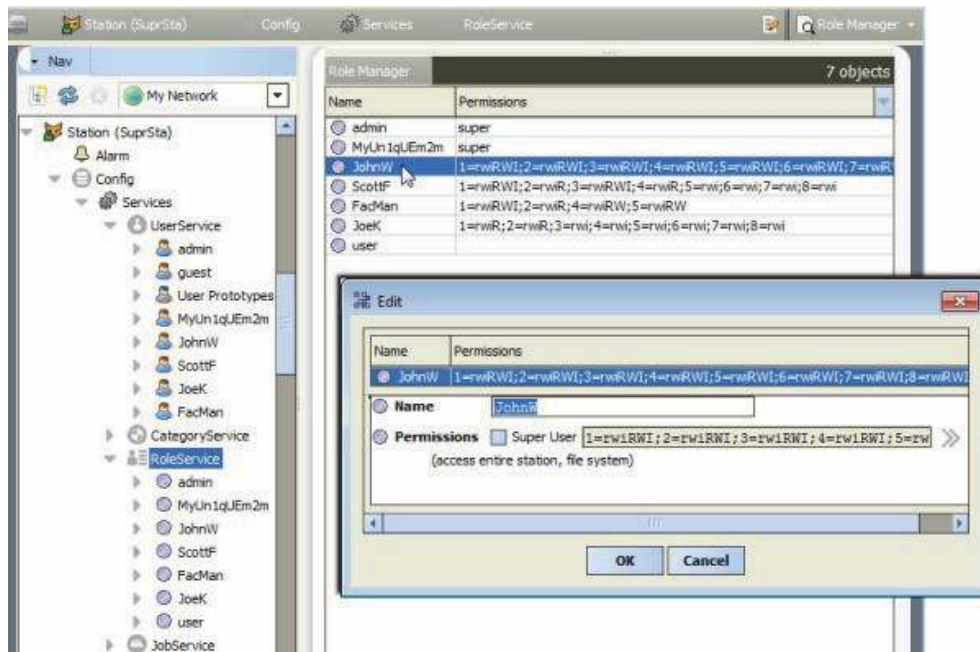
In order to migrate Users from AX to N4.0, while preserving their existing category permissions, note an N4 migration results in a *separate Role created for each User—and named identically to that User*.

Figure 20 Station migrated to N4.0 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 21 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 Niagara 4 station, notice that the **Password** property is now inside a new Authenticatorcontainer, 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 Niagara 4. Along with a new (and required) AuthenticationService 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 Migration reference

Topics covered in this chapter

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

A number of changes to NiagaraAX Program objects (components) need to be made to allow them to run under Niagara 4. The N4 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 Niagara 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 N4 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 Niagara 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 AX-3.5 in order to support compiling Programs into Program Modules. It has been removed for N4.0. 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`

NiagaraAX Program object:

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

Niagara 4 Program object:

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

Security Manager

Niagara 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 NiagaraAX, there was a flag/entry to change this behavior in `system.properties`. In Niagara 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

NiagaraAX Program object:

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

Niagara 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 NiagaraAX demo station. It is one of several Programs flagged with a "WARNING unable to compile" entry when AX-3.8 station backup `.dist` file for the demo station is used as source in the N4 migration tool.

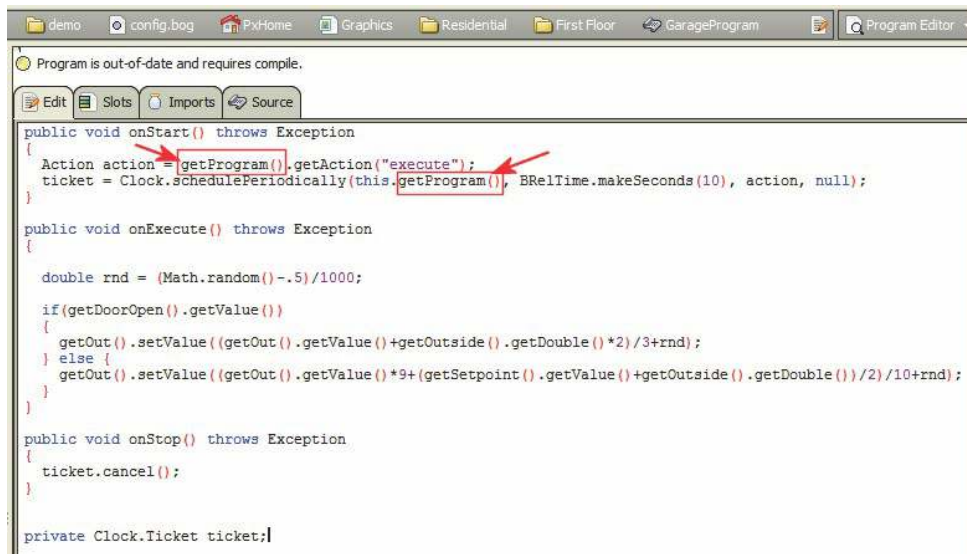
A related snippet from the migration report (log file) looks like below.

```
WARNING unable to compile Program object PxHome.Graphics.Residential.First Floor.
GarageProgram
C:\Users\e333988\Niagara4.0\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 60.

Opening the migrated "demo" station in your N4 Workbench, 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.

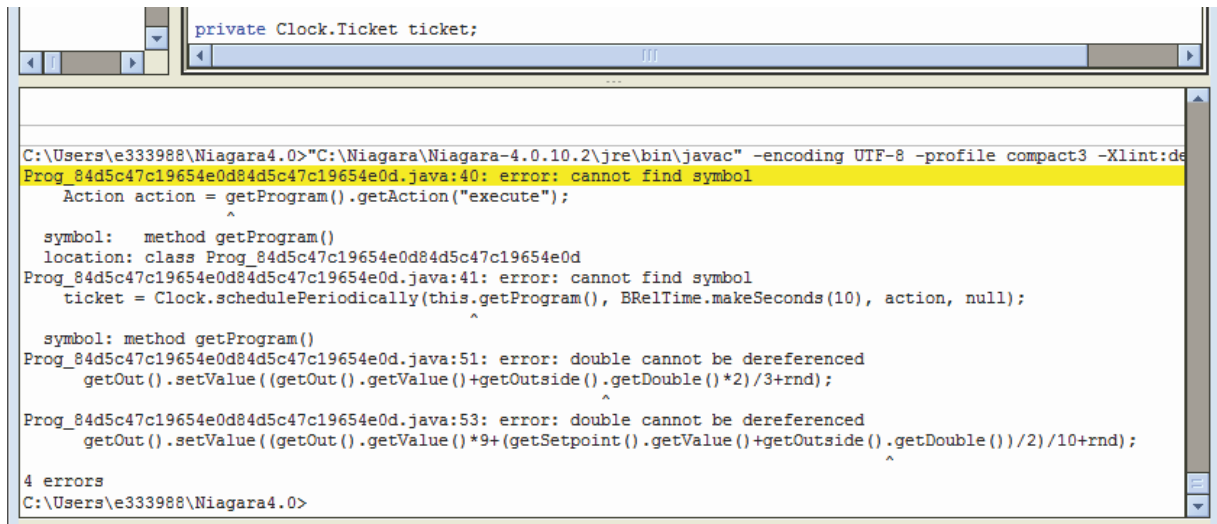
Figure 22 Example Program with issues from migrated “demo” station



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 Workbench window, as shown below.

Figure 23 Example errors from a compile before making any Program changes



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 24 Figure 28, page 63).

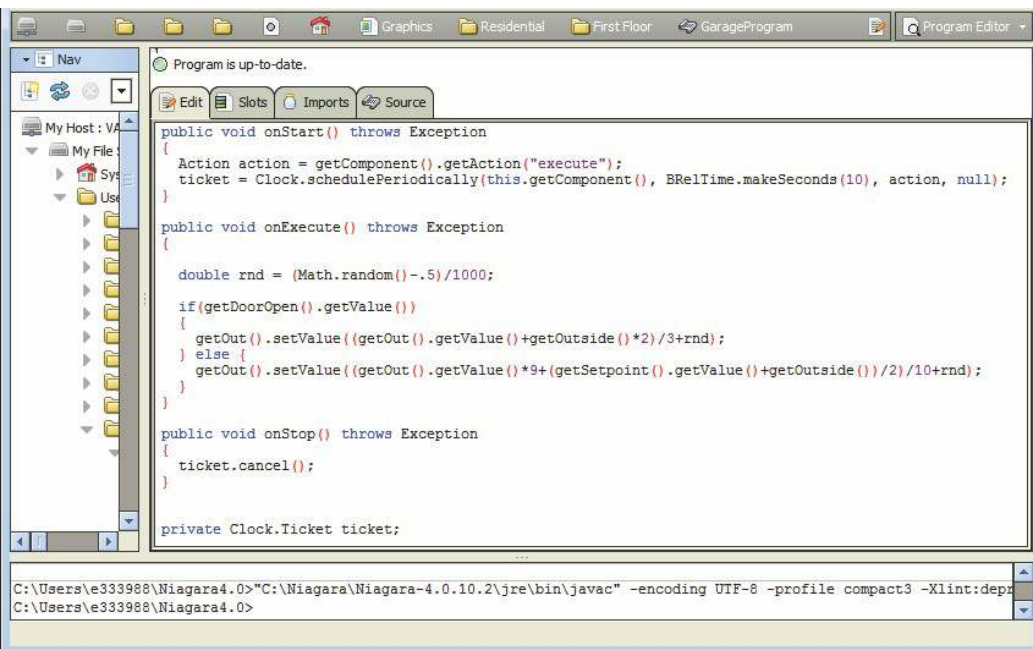
Figure 24 Example Program with remaining issues from primitive syntax errors



This is no longer necessary, as explained in the section [Primitives](#), page 62. To fix, simply remove the text shown marked above.

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

Figure 25 Example Program object after fixing all issues and recompiling.



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