Technical Document

AX to N4 Migration Guide

August 7, 2015

niagara4

AX to N4 Migration Guide

Tridium, Inc.
3951 Westerre Parkway, Suite 350
Richmond, Virginia 23233
U.S.A

Confidentiality

The information contained in this document is confidential information of Tridium, Inc., a Delaware corporation ("Tridium"). Such information and the software described herein, is furnished under a license agreement and may be used only in accordance with that agreement.

The information contained in this document is provided solely for use by Tridium employees, licensees, and system owners; and, except as permitted under the below copyright notice, is not to be released to, or reproduced for, anyone else.

While every effort has been made to assure the accuracy of this document, Tridium is not responsible for damages of any kind, including without limitation consequential damages, arising from the application of the information contained herein. Information and specifications published here are current as of the date of this publication and are subject to change without notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia.

Trademark notice

BACnet and ASHRAE are registered trademarks of American Society of Heating, Refrigerating and Air-Conditioning Engineers. Microsoft, Excel, Internet Explorer, Windows, Windows Vista, Windows Server, and SQL Server are registered trademarks of Microsoft Corporation. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Mozilla and Firefox are trademarks of the Mozilla Foundation. Echelon, LON, LonMark, LonTalk, and LonWorks are registered trademarks of Echelon Corporation. Tridium, JACE, Niagara Framework, NiagaraAX Framework, and Sedona Framework are registered trademarks, and Workbench, WorkPlaceAX, and AXSupervisor, are trademarks of Tridium Inc. All other product names and services mentioned in this publication that is known to be trademarks, registered trademarks, or service marks are the property of their respective owners.

Copyright and patent notice

The product(s) described herein may be covered by one or more U.S or foreign patents of Tridium.

Contents

About this document

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

- Chapter 1 AX to N4 migration overview, page 7

  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 AX to N4 migration tasks, page 25

  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 JACE controller upgrades.

- Chapter 3 Niagara 4 system verification, page 55

  Explains system areas that may need attention and possible configuration immediately following the installation of Niagara 4 upgrades and migrated stations.

- A Migration details, page 61

  Contains some details on Program object (component) changes in N4, along with an example "Program object fix".

Document change log

Initial release document, August 7, 2015.

Related documentation

Additional information on the Niagara platforms and installation is available in the following documents.

- *Niagara 4 Platform Guide*
- *Niagara 4.0 Installation Guide*

Chapter 1 AX to N4 migration overview

Topics covered in this chapter

The Niagara 4 Framework is similar to the NiagaraAX Framework, but has evolved enough such that existing NiagaraAX (AX) platforms and stations, if upgradeable, must be *migrated* to Niagara 4 (N4). Therefore, after installing Niagara 4 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 jobs migrated to N4 must retain some number of AX JACE controllers. Note that some platform types cannot be migrated to N4 from AX, 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.

Migration requirements and restrictions

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 N4. An understanding of these is essential before beginning migration work for any existing NiagaraAX job.

- "Niagara 4 platform compatibility" on page 6, page 7
- "Niagara 4 driver and application support" on page 8, page 11

Niagara 4 platform compatibility

The initial release of Niagara 4 (N4.0) is supported by PCs with certain Windows operating systems and by many existing JACE controller platforms, as listed in the following table.

NOTE:

To qualify as a candidate for migration from AX to N4, any of the supported JACE platforms listed below should be running a released version of NiagaraAX-3.8, with working communications to other AX-3.8 platforms at the job site. Earlier NiagaraAX versions, e.g. AX-3.7 and AX-3.6 are not supported for migration to N4.

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

Niagara 4 platform compatibility

Niagara platform (“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).	*Windows Vista (all) and Windows XP (all) are unsupported.* Linux OS support is unavailable in N4.0. Prior editions of Windows Server, such as Windows Server 2008, are unsupported.
JACE-8000	Supported	Initially supports *only* N4.0. At some future point, an update release of AX-3.8 *may* be available that supports most features of this newest JACE controller.
JACE-7 (JVLN)	Supported	JACE-7 WiFi option is not supported in N4. Some other JACE option cards are also not supported in N4 (see NPM below).
JACE-6E, JACE-603, JACE-645 (NPM6E)	Supported	JACE options not supported in N4 include GPRS, Sedona wireless, dialup modem. In addition, there are minimum resource requirements (fully loaded units in AX may not be able to run in N4.0). For details, see “JACE platform resource requirements” on page 7, page 8.
JACE-6, JACE-602 Express (NPM6)		
JACE-3E (NPM3E)		
JACE-2, JACE-202 Express (NPM2)	Not supported	Although these JACE platforms cannot run N4, if running AX-3.8 there is support for data exchange with N4 platforms. For example, these stations can be under the NiagaraNetwork of an N4 Supervisor. For related details, see “Features and compatibility between N4 and AX” on page 10, page 13.
JACE-4 (JACE-403)		
JACE-5 (JACE–404)		
JACE-NXT (JACE-NXT)		
JACE-NXS (JACE-NXS)		
SoftJACE (Jsoft)	Not supported	An N4 SoftJACE is not available. However, any AX-3.8 SoftJACE can exchange data with N4 platforms, as described above.

For any supported AX-3.8 JACE to be migrated to N4, be sure it has the necessary resources available. See “JACE platform resource requirements” on page 7, page 8.

JACE platform resource requirements

N4 requires more disk space, more Java heap, and more system RAM to run than a similar AX station. Therefore, an 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 due to new features and functionality. The baseline heap increase is about 10MB. Additionally, there is an increase in the system memory required to run N4 on a JACE controller.

- History RAM disk size (NPM6/NPM6E only)

  The maximum history RAM disk size on a JACE-6 has been reduced from 64MB to 32MB to free additional system memory. The default RAM disk size is 32MB, so this should not affect most units. If the RAM disk on an AX JACE-6 has been increased, then this increase in the RAM disk size comes at the expense of Java heap. Note that maximum history count has not changed.

See the following for more details.

- "Recommended minimum resources by JACE controller type", page 9
- "Checking AX JACE for N4 minimum resource requirements", page 9

Recommended minimum resources by JACE controller type

The following table summarizes recommended resource requirements in AX, before converting to N4.

Recommended JACE resource requirements in AX, before conversion 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	29
NPM6 (JACE-6)	20	94	29
NPM6E (JACE-6E, JACE-603, JACE-645)	24	94	29
JACE-7	22	94	87

(1) If an AX 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 (NPM3 and NPM6E)
- 14MB for N4 increase
- 2MB for JACE-6 boot partition increase

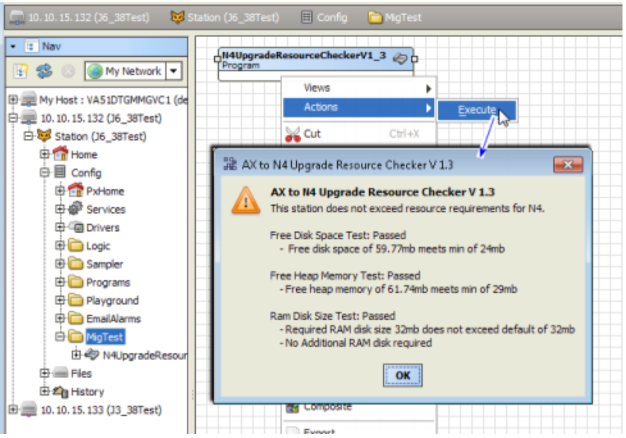
The Minimum Free Heap recommendation is based on the following:

```
(20% total heap size) + 10MB for additional N4 overhead
```

Checking AX JACE for N4 minimum resource requirements

There is a special BOG file with an AX-compatible Program component that you can use to determine if a JACE station meets the minimum resource requirements for and upgrade (conversion to) N4. When copied into an AX-3.8 JACE station and executed, the Program produces a popup notification to state if the JACE station has adequate available resources. 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 a AX-3.8 JACE station



AX to N4 Upgrade Resource Checker V 1.3
This station does not exceed resource requirements for N4.

Free Disk Space Test: Passed
- Free disk space of 59.77mb meets min of 24mb

Free Heap Memory Test: Passed
- Free heap memory of 61.74mb meets min of 29mb

Ram Disk Size Test: Passed
- Required RAM disk size 32mb does not exceed default of 32mb
- No Additional RAM disk required

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

Using the N4 resource checker Program

Prerequisite: You have the BOG file with the AX-compatible Program, AX-3.8 Workbench, and can open a station connection (Fox or Foxs) to the AX-3.8 JACE controller.

Step 1 Copy the BOG file with Program component under your Niagara AX-3.8 system home folder.

Step 2 Using AX-3.8 Workbench, open the AX-3.8 JACE 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 Program (may be named "`N4UpgradeResourceCheckerV1_n`") and copy it into the JACE station. You should be able to immediately use the Program.

Step 4 Right-click the Program in the JACE station, and select **Actions**→ →**Execute**.

After a few seconds, a popup **AX to N4 Upgrade Resource Checker** dialog displays the test results, as either "Passed" or "Failed".

Step 5 Click **OK** to close the popup dialog.

Manually checking free disk space in AX JACE

Prerequisite: JACE platform is at AX-3.8 and running the station to be migrated to N4.0.

Step 1 Using AX-3.8 Workbench, open the station (Fox/Foxs), and expand **Config** > **Services** > **PlatformServices**. Double-click it for the **Platform Service Container Plugin view**.

Step 2 In the "**Filesystem**" entry for **/ffs0**, look at the "Free" value and compare to the value in Table 2. The JACE value should be equal or greater than this minimum value.

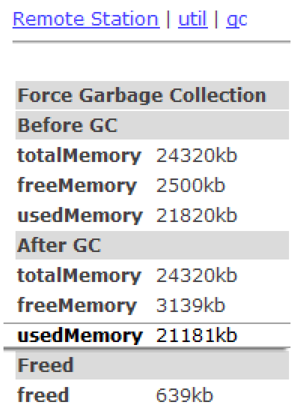
Manually checking heap usage in AX JACE

Prerequisite: JACE platform is at AX-3.8 and running the station to be migrated to N4.0.

Step 1 Using AX-3.8 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	639kb

This represents the low point of Java heap usage.

Niagara 4 driver and application support

The initial release of Niagara 4 (N4.0) supports most of the commonly used drivers in NiagaraAX, for example BACnet, LonWorks, Modbus, SNMP, and some legacy serial based drivers. However, support in N4 was dropped for a few drivers and some applications, with a few notable ones listed in Table 3.

Unsupported drivers and features in Niagara 4 (N4.0)

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, e.g. 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. Only serial tunneling is supported in N4, 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.

Unsupported driver / application	Removed modules / notes
Sedona (SedonaNetwork, SedonaJen6lpNetwork)	Module “sedonaInstaller” in AX builds, which from a downloaded Sedona “bundle” would install Sedona-specific modules for Niagara integration.
“Older type” video drivers based on the “devDriver”, superseded in AX by “nDriver” based videoDriver framework	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	R2 platform support on JACE-603/JACE-645 controllers requires NiagaraAX Workbench (3.6, 3.7, or 3.8).

Note the list above is partial; there are various other rarely-used or limited-distribution software modules in AX that were not brought forward to Niagara 4. If using any such modules, we recommend you check with your Niagara 4 support channel to verify they are available in N4.0.

Additionally, any OEM-specific or custom (non-Tridium) software modules used on any AX-3.8 platform to be migrated must be refactored for use in Niagara 4. Note this is a separate process from using the Niagara 4 migration tool on a saved AX station, and must be completed by the custom module developer before you can migrate a station.

Wire compatibility between Niagara 4 and NiagaraAX

Supported communications, or “wire compatibility” between platforms and tools in Niagara 4 (N4) and NiagaraAX (AX-3.8) can be a factor in different job scenarios.

- “Supported connections between N4 and AX” on page 10, page 12
- “Features and compatibility between N4 and AX” on page 10, page 13

Supported connections between N4 and AX

Among possible connections between Niagara Workbench and Niagara stations, the following matrix summarizes supported and unsupported connections between Niagara 4 (N4) and NiagaraAX-3.8 (AX).

Supported connections between Niagara 4 (N4) and NiagaraAX -3.8 (AX)

From	To	Supported	Notes
AX Workbench	N4 station, N4 Supervisor	No	Prohibited in software.
AX Workbench	N4 platform	No	Prohibited in software.
N4 Workbench	AX platform	Yes	Provides support necessary for AX to N4 upgrades. A few functions from the **Platform Administration** view for AX JACE controller platforms are unavailable, including the AX **Commissioning Wizard**.
N4 Workbench	AX station	No	Prohibited in software. Note that ongoing AX *station configuration* requires AX 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 between N4 and AX”, page 13
N4 station, N4 Supervisor	AX station	Yes	See “Features and compatibility between N4 and AX”, page 13
N4 station	AX Supervisor	No	Not prohibited in software, but unsupported.

Features and compatibility between N4 and AX

Within supported station connections between Niagara 4 and NiagaraAX (AX-3.8), the following matrix summarizes features and compatibilities. This applies mostly to NiagaraNetwork functions between stations, for example Niagara proxy points, imported or exported histories and schedules, and so on.

Features and compatibility between Niagara 4 (N4) and NiagaraAX -3.8 (AX)

Data Type	Support	Notes
Points	Niagara 4 supports point discovery and import from AX stations. Point data can be shared from AX to N4, or N4 to AX. Note that AX stations are *unable to discover* points in N4 stations; the alternative is to manually add and configure points.	
Histories	AX stations are able to export histories to Niagara 4 station. AX stations cannot import histories from N4 stations. N4 histories cannot be exported to AX stations. N4 stations can import histories from AX stations.	Niagara 4 uses a different history encoding, and so detects and converts AX history records.
Alarms	AX alarms can be routed to and acknowledged in N4 stations. Niagara 4 alarms cannot be routed to AX stations.	
Schedules	Niagara 4 schedules can be exported to AX stations. Niagara 4 schedules can be imported by AX stations. AX schedules cannot be exported to N4 stations. AXAX schedules cannot be imported by N4 stations.	
Users	Niagara 4 can only "sync out" users to AX stations for supported "network users". LDAP is a good alternative to Niagara "network user" synchronization.	Niagara 4 supports user synchronization to AX stations for supported users (users that use an authentication scheme that involves a password portable to AX stations). The sync process is supported in one direction only: N4 station to AX station, but not AX station to N4 station. Therefore, the N4 station supports only "sync out" of users to the AX station, and the AX station supports only "sync in" of users from the N4 station.
Virtual Points	Niagara 4 supports Niagara virtual points in AX stations. AX supports Niagara virtual points in N4 stations.	
Tunneling	Fox, HTTP, and platform tunneling are *not supported* in Niagara 4, as they provide insufficient security.	Serial tunneling is supported in Niagara 4.
Export Tags	AX stations can join Niagara 4 stations as subordinates. Niagara 4 stations cannot join AX stations as subordinates.	In order for an AX station (subordinate) to join a Niagara 4 station (Supervisor), prior to executing the export tag join, the two-way fox connection must be established between the Niagara stations in each station's NiagaraNetwork. Failure to preconfigure the two-way connection will result in an error during the export tag join. Px files used with ExportTags may be problematic, as all Px files must be converted for use with N4.
Files	File transfer using Fox (NiagaraDeviceFileExt) is supported.	File permission checks are enforced.
Provisioning	Niagara 4 Supervisors can provision AX stations.	Supported operations for AX subordinates include: • Update licenses • Backup stations • Copy Supervisor file • Install valid AX software • Upgrade out-of-date AX software • Restore prior station backup (e.g., from the step history view of backup job). Note that Niagara 4 provisioning robots will not run on AX stations.

Note for a Niagara 4 Supervisor to provision AX stations in addition to N4 stations, you must import the "software database" from the prior AX-3.8 Supervisor into the N4 Supervisor's software database. However, note an N4 Supervisor station cannot directly upgrade AX-3.8 platforms to Niagara 4.

Need for migration

Because of the fundamental changes and security improvements in Niagara 4 a migration process is required instead of a simple reuse of any saved AX station. Following, are a few of the major differences:

- Directory structure changes

  In NiagaraAX, all folders and files for an AX build are installed and created/maintained under a single "**Sys Home**" directory, for example for a Windows PC platform:

  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 "Types of Niagara 4 "Homes" (User Homes versus Sys Home)" on page 13, page 15.

- Software modules were refactored

  In NiagaraAX, each software module typically contains multiple content levels: "runtime", "user interface" (ui), and sometimes "doc". When installing AX 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 module 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 User Service 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 Authentication Service architecture allows the flexibility to specify the "authentication scheme" to use for station access at the *user level*. Among other things, this lets you integrate LDAP users into a station's standard User Service, (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 *Niagara 4 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 Niagara Java classes and methods.

File locations on Niagara 4 platforms

During the Workbench installation and platform commissioning processes, Niagara 4 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.
- A Workbench user home for each user contains configuration data, including option files, and Niagara registries.
- A platform daemon user home for the Supervisor or engineering workstation platform contains platform configuration data.
- Two station homes called, *protected station home* and *station home*, are part of each user home.

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

Table 1 Homes on a Supervisor or engineering workstation

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-4.0.xx` Executables and software files	! (as in NiagaraAX)
My Host→My File System→User Home	N/A	niagara_user_home	`C:\Users\userName\Niagara4.0\brand` Workbench user home for each human user contains that user's unique configuration files.	~ (unique to Niagara 4)
`shared` folder	N/A	station_home	`C:\Users\userName\Niagara4.0\brand\shared`	^ (as in NiagaraAX)
`stations` folder	N/A	protected_station_home	`C:\Users\userName\Niagara4.0\brand\stations`	^^ (unique to Niagara 4)
N/A	**User Home**	niagara_user_home	`C:\ProgramData\Niagara4.0\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 Niagara 4)

On a controller there are two homes:

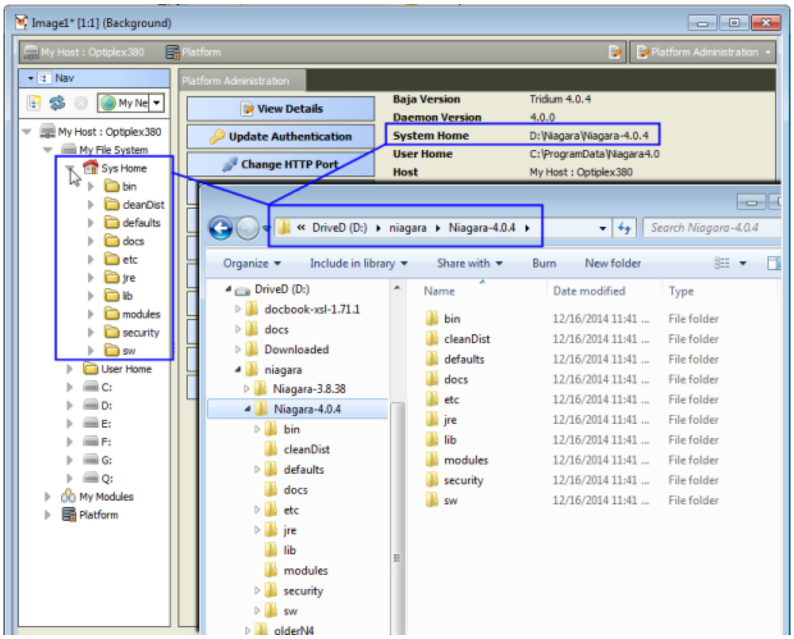
Table 2 Homes on a controller

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 Niagara 4)

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



Baja Version Tridium 4.0.4
Daemon Version 4.0.0
System Home D:\Niagara\Niagara-4.0.4
User Home C:\ProgramData\Niagara4.0
Host My Host : Optiplex380

Name	Date modified	Type
bin	12/16/2014 11:41 ...	File folder
cleanDist	12/16/2014 11:41 ...	File folder
defaults	12/16/2014 11:41 ...	File folder
docs	12/16/2014 11:41 ...	File folder
etc	12/16/2014 11:41 ...	File folder
jre	12/16/2014 11:41 ...	File folder
lib	12/16/2014 11:41 ...	File folder
modules	12/16/2014 11:41 ...	File folder
security	12/16/2014 11:41 ...	File folder
sw	12/16/2014 11:41 ...	File folder

The example above shows the file system for a 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: `D:\niagara\Niagara-4.0.15`.

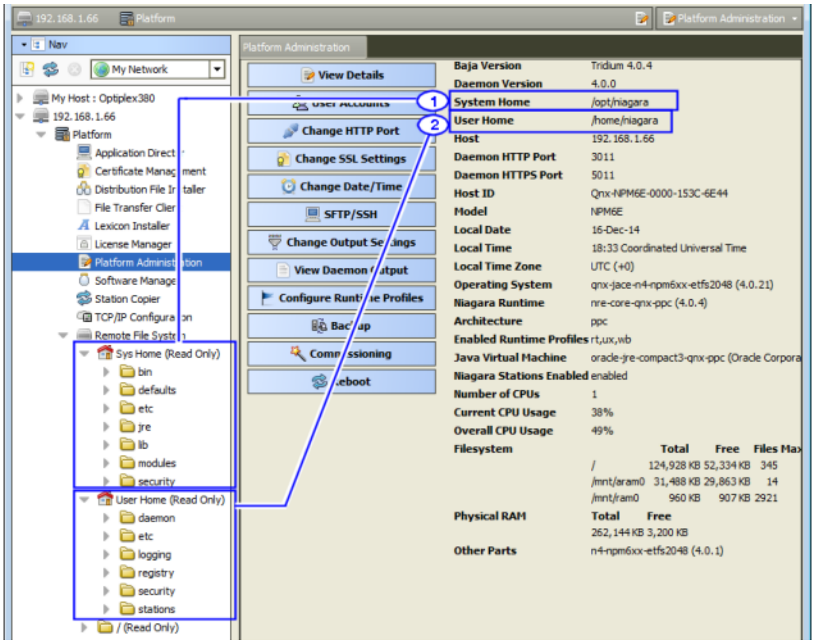
The system home on a controller appears as **System Home** in the **Platform Administration** view. The actual location of the system home folder for a controller is: `/opt/niagara`.

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.

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

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



```
Baja Version                Tridium 4.0.4
Daemon Version              4.0.0
System Home                 /opt/niagara
User Home                   /home/niagara
Host                        192.168.1.66
Daemon HTTP Port            3011
Daemon HTTPS Port           5011
Host ID                     Qnx-NPM6E-0000-153C-6E44
Model                       NPM6E
Local Date                  16-Dec-14
Local Time                  18:33 Coordinated Universal Time
Local Time Zone             UTC (+0)
Operating System            qnx-jace-n4-npm6xx-etfs2048 (4.0.21)
Niagara Runtime             nre-core-qnx-ppc (4.0.4)
Architecture                ppc
Enabled Runtime Profiles    rt,ux,wb
Java Virtual Machine        oracle-jre-compact3-qnx-ppc (Oracle Corpora
Niagara Stations Enabled    enabled
Number of CPUs              1
Current CPU Usage           38%
Overall CPU Usage           49%
Filesystem                             Total      Free   Files Max
                            /          124,928 KB 52,334 KB  345
                            /mnt/aram0  31,488 KB 29,863 KB   14
                            /mnt/ram0      960 KB    907 KB 2921
Physical RAM                Total      Free
                            262,144 KB 3,200 KB
Other Parts                 n4-npm6xx-etfs2048 (4.0.1)
```

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

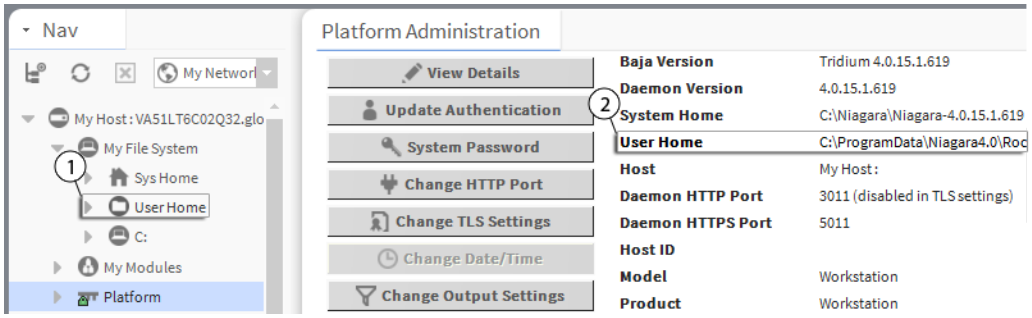
The Supervisor or engineering workstation that is licensed to run a station has a daemon user home. The daemon is a server process and represents a (non-human) user that manages the Supervisor's running station. The Supervisor's daemon user home contains daemon-specific configuration information. The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

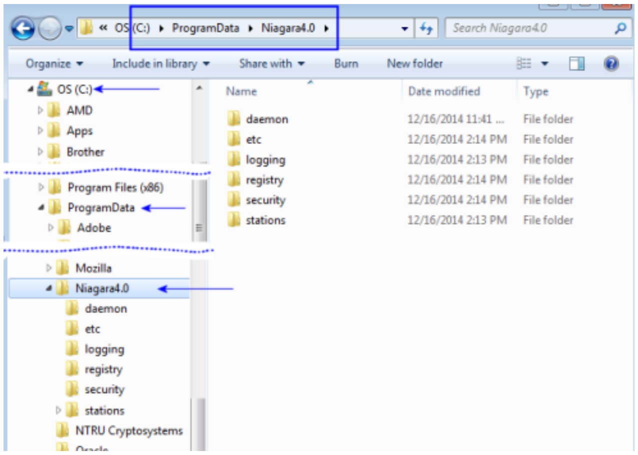


Baja Version	Tridium 4.0.15.1.619
Daemon Version	4.0.15.1.619
System Home	C:\Niagara\Niagara-4.0.15.1.619
User Home	C:\ProgramData\Niagara4.0\Roc
Host	My Host :
Daemon HTTP Port	3011 (disabled in TLS settings)
Daemon HTTPS Port	5011
Host ID	
Model	Workstation
Product	Workstation

- Callout 1 identifies a 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



Name	Date modified	Type
daemon	12/16/2014 11:41 ...	File folder
etc	12/16/2014 2:14 PM	File folder
logging	12/16/2014 2:13 PM	File folder
registry	12/16/2014 2:14 PM	File folder
security	12/16/2014 2:14 PM	File folder
stations	12/16/2014 2:13 PM	File folder

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\brand
```

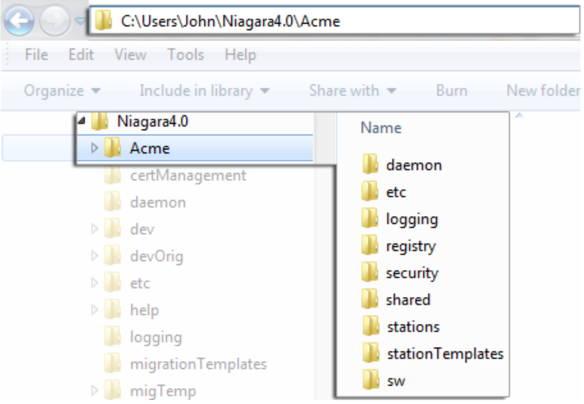
```
C:\Users\Mike\Niagara4.0\brand
```

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

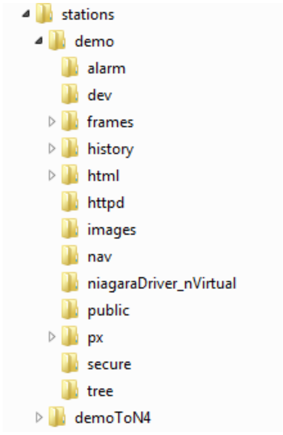
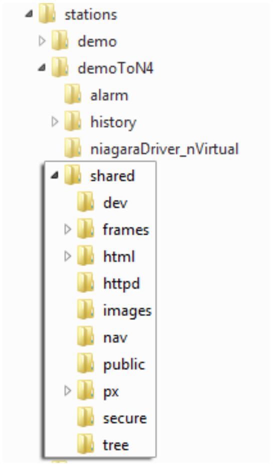
  – dataRecovery

  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.

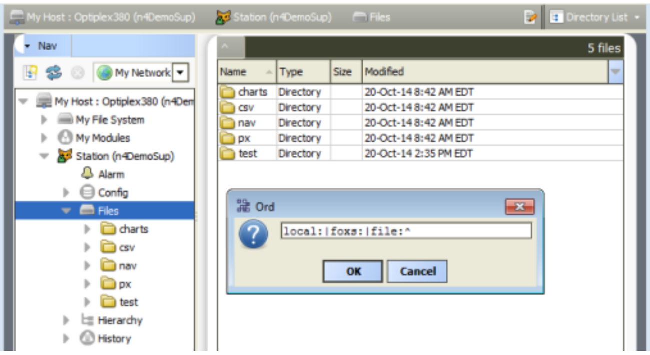
  The alias station_home retains the same file ORD shortcut (^) as used in NiagaraAX—only in Niagara 4 this points to the station's **shared** sub-folder.

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

AX station folder	N4 station folder
 stations demo alarm dev frames history html httpd images nav niagaraDriver_nVirtual public px secure tree demoToN4	 stations demo demoToN4 alarm history niagaraDriver_nVirtual shared 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



Name	Type	Size	Modified
charts	Directory		20-Oct-14 8:42 AM EDT
csv	Directory		20-Oct-14 8:42 AM EDT
nav	Directory		20-Oct-14 8:42 AM EDT
px	Directory		20-Oct-14 8:42 AM EDT
test	Directory		20-Oct-14 2:35 PM EDT

```
local:|foxs:|file:^
```

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 yourWorkbench **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 a 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 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 a Niagara 4 host and use Workbench, each person has their own separate Workbench **User Home**.

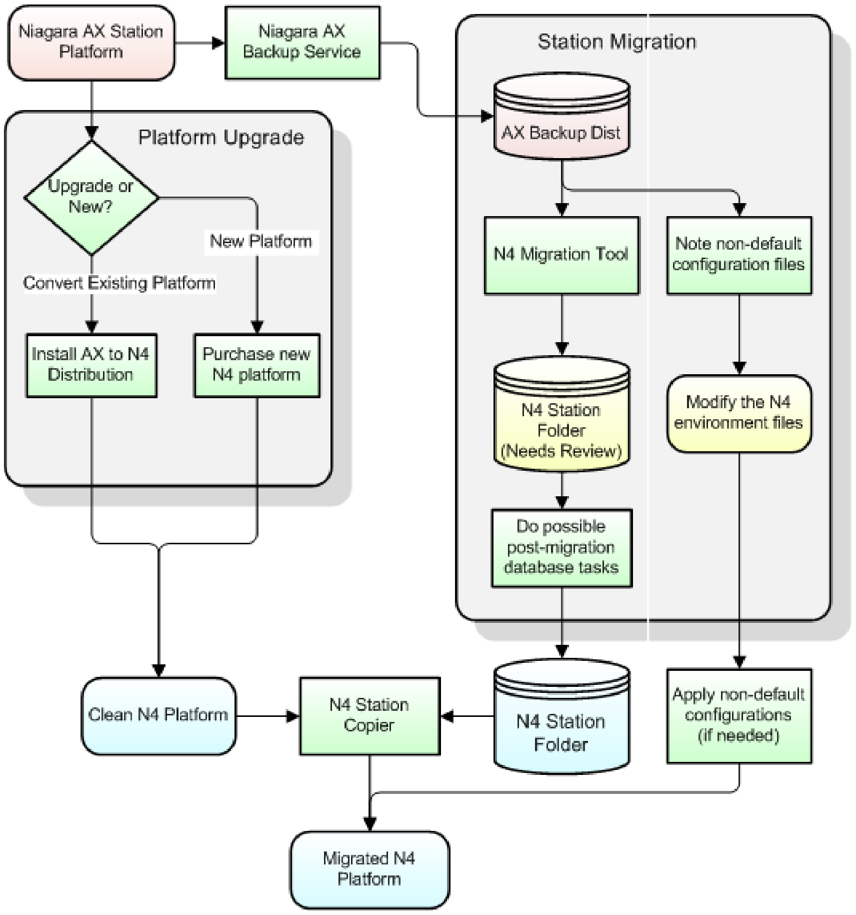
Windows users require permissions to access other users' files; yet it's possible that different users of a system (Supervisor or engineering workstation) may need to share items such as station backups, station copies, saved template files, and so on. Such items may be in multiple Workbench **User Home** 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 (a Workbench **User Home** is not shareable).

Migration workflow diagram

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

Figure 9 Diagram with AX to N4 migration steps


Niagara AX Station Platform
Niagara AX Backup Service
Station Migration
AX Backup Dist
Platform Upgrade
Upgrade or New?
New Platform
Convert Existing Platform
Install AX to N4 Distribution
Purchase new N4 platform
N4 Migration Tool
Note non-default configuration files
N4 Station Folder (Needs Review)
Modify the N4 environment files
Do possible post-migration database tasks
Clean N4 Platform
N4 Station Copier
N4 Station Folder
Apply non-default configurations (if needed)
Migrated N4 Platform

In the diagram above, red items indicate NiagaraAX artifacts, blue items indicate Niagara 4 artifacts, and 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 22.

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 N4 migration", page 23
- "Backup AX stations and migrate databases", page 24
- "Upgrade the Supervisor to Niagara 4", page 24
- "Convert JACE controllers and install N4 stations", page 24
- "Perform system cleanup", page 24

Assess and prepare for N4 migration

1. Identify those platforms to be migrated from NiagaraAX to Niagara 4. Note that JACE controllers must be running at least NiagaraAX 3.8.nn, and be compatible with N4.0. See "Niagara 4 platform compatibility" on page 6, page 7 and "JACE platform resource requirements" on page 7, page 8 for details.
2. Review the current drivers and applications running on the JACE controllers to be converted to N4. See "Niagara 4 driver and application support" on page 8, 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 JACE controllers running AX-3.8, and still integrate them with a Niagara 4 Supervisor station. However, note you cannot integrate a Supervisor at AX-3.8 with Niagara 4 JACE controllers.

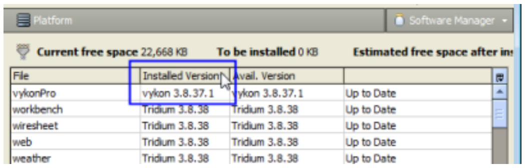
   NOTE:

   For any job that will have a *mix* of Niagara 4 and AX-3.8 platforms, refer to the section "Wire compatibility between Niagara 4 and NiagaraAX" on page 10, 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 the Niagara 4 structure.

   If you are unsure if an AX-3.8 JACE controller has such modules, you can check by making a platform connection to it, then going to the **Software Manager** view.

   Current free space 22,668 KB **To be installed** 0 KB **Estimated free space after ins**

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. Note that 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 the N4 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 N4 migration process..
4. 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).

   Note in almost all cases the calculated Niagara Host ID will remain the same. An exception is if on a Windows platform, if moving from a 32-bit install to a 64-bit install (or vice versa), in this case the Niagara Host ID will be different.

   Be sure to archive the old AX-3.8 license files as well.

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

Backup AX stations and migrate databases

Using AX-3.8 Workbench, make a station *backup* for each JACE controller to be migrated, and do the same for the job's Supervisor station. Then you run the N4 migration tool against these stations, with output as N4.0 station copies. When making AX-3.8 station backups, be sure to include history and alarm files.

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

Upgrade the Supervisor to Niagara 4

You upgrade the AX-3.8 Supervisor to N4.0 first. There it can startup and still communicate to JACE controllers still running AX-3.8 (see "Wire compatibility between Niagara 4 and NiagaraAX" on page 10, page 12 for related details). For the upgrade procedure.

Convert JACE controllers and install N4 stations

You do this after upgrading the Supervisor from AX-3.8 to N4.0—ideally immediately afterwards (providing that you have all the JACE controller stations successfully migrated to N4 stations).

NOTE:

Before converting a JACE controller from AX, you must successfully migrate its station to N4, including an offline checkout of it using Niagara 4 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 AX to N4 migration tasks

Topics covered in this chapter

This section provides AX-3.8 to Niagara 4 (N4.0) migration tasks, along with reference details as needed.

Preparing with NiagaraAX

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

- "Export SSL certificate Trust Store and Private Key Stores", page 26
- "Kerberos keytab file notes", page 26
- "Making source AX-3.8 backups", page 27
- "Secure Niagara 4 license files", page 30

Export SSL certificate Trust Store and Private Key Stores

If AX-3.8 JACE controllers are configured to use SSL (signed certificates), export the keys from the Trust Store and the Private Key store. These are deleted from each JACE during conversion to N4. After conversion, you can import these certificates back using N4 Workbench and a platform connection.

Exporting SSL certificate Trust Store and Key Stores

Step 1 Using AX-3.8 Workbench, access the **Certificate Management** view of the AX-3.8 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 to N4.

Make sure exported certificates files will be available to the N4 Workbench PC to use later. For related details, refer to the *NiagaraAX SSL Connectivity Guide*.

Kerberos keytab file notes

If your AX-3.8 stations use Kerberos authentication for user services (e.g. **LdapV3ADUserService**) *and* use a “keytab file” supplied by the site’s Kerberos administrator, the N4 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 N4 migration tool.

Relocating the Kerberos keytab file in a AX-3.8 station (prior to migration)

Step 1 Using AX-3.8 Workbench, open the station (Foxs or Fox), expand the **Config** node, and navigate to the property sheet of the user service’s authenticator.

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

Note the “Key Tab Location” property ORD value, referencing 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 in it open a platform connection to the remote host. 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 “ldap”. This will be the target spot for relocating 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 user service’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 AX-3.8 backups

For any AX-3.8 platform you are migrating to Niagara 4, make a station *backup* using AX-3.8 Workbench. You will use this station backup dist file as the source when running the Niagara 4 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 will contain 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 a 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
("d:\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
```

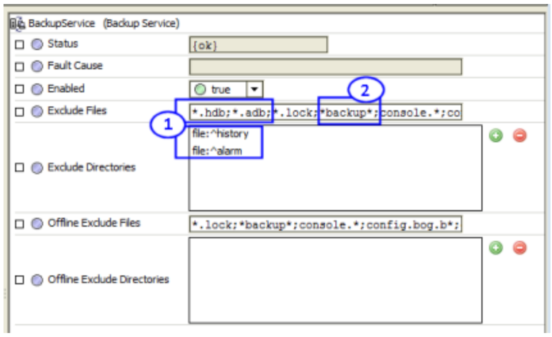
Setting AX-3.8 BackupService properties

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:

Prerequisite: Stations are configured and running on a host compatible with Niagara 4.

Figure 10 Entries typically removed in BackupService properties



BackupService (Backup Service)

□ Status: {ok}

□ Fault Cause:

□ Enabled: true

□ Exclude Files: *.hdb;*.adb;*.lock;*backup*;console.*;co

□ Exclude Directories: file:^history
file:^alarm

□ Offline Exclude Files: *.lock;*backup*;console.*;config.bog.b*;

□ Offline Exclude Directories:

Related BackupService properties shown above are:

1. In the BackupService for each AX-3.8 station to be migrated (whether JACE or Supervisor):

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

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

2. 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 1 In AX-3.8 Workbench, open the station and access the **BackupService** property sheet, as above.

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.

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 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).

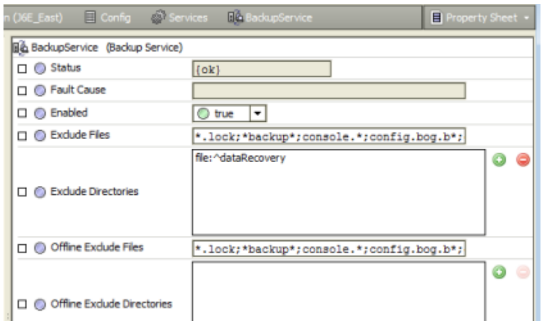
NOTE:

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 will include 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. Note also that online Fox backups are more convenient to run than offline backups.

Step 4 **Save** any changes made.

The following figure shows an example BackupService configured in a JACE station.

Figure 11 Example BackupService properties in JACE station



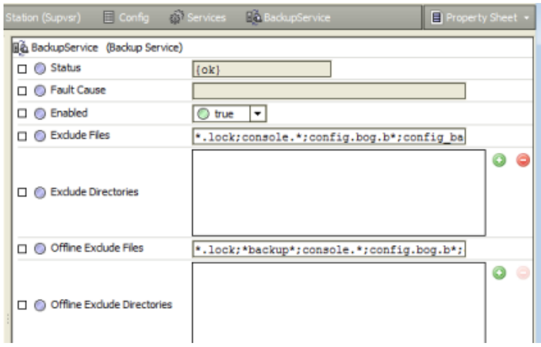
BackupService (Backup Service)

□ Status	{ok}
□ Fault Cause	
□ Enabled	true
□ Exclude Files	*.lock;*backup*;console.*;config.bog.b*;
□ Exclude Directories	file:^dataRecovery
□ Offline Exclude Files	*.lock;*backup*;console.*;config.bog.b*;
□ Offline Exclude Directories	

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

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

Figure 12 Example BackupService properties in Supervisor station configured for provisioning



BackupService (Backup Service)

□ Status	{ok}
□ Fault Cause	
□ Enabled	true
□ Exclude Files	*.lock;console.*;config.bog.b*;config_ba
□ Exclude Directories	
□ Offline Exclude Files	*.lock;*backup*;console.*;config.bog.b*;
□ Offline Exclude Directories	

Note that “Exclude Files” has no entry for: `*backup*;` (to allow prior Niagara provisioning backups to be brought over to the N4 Supervisor).

Making source AX-3.8 station backups

A station backup (.dist file) from a **Backup** command is the preferred archive source when migrating a AX-3.8 station to Niagara 4.

Prerequisites:

Prerequisite: The station’s BackupService is configured appropriately.
See “Setting AX-3.8 BackupService properties” on page 25, page 28.

NOTE:

If desired, you can *shorten default file names* of the station backup .dist files. This simplifies typing in the source .dist file name in the command-line migration tool. Do this either when making the station backups, or afterwards by *renaming* saved backup .dist files (leave the .dist extension intact).

For example, if: backup_stationName_141221_0757.dist, shorten to: stationName.dist. Also for best performance, do not save these backup.dists into the Niagara 4 System home or User home directories. The files can be located anywhere else, for example: `C:\AxBackups`.

Step 1 Using AX-3.8 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 it first locally saves is configuration, and then a backup dist file is saved in the backups folder of your Niagara home (install) folder.

- For an *offline* backup, you must make a platform connection and first *stop* the station, from the **Application Director**. Then from the **Platform Administration** view, select **Backup**.

  NOTE:

  This alternative method 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 AX station if desired, so that it can continue working while you run the Niagara 4 migration tool and are ready to convert that platform.

Step 2 Repeat this for all AX-3.8 stations to be migrated to Niagara 4.

If working from an AX-3.8 Supervisor PC to be upgraded to an N4.0 Supervisor, or an AX-3.8 engineering workstation that is also a N4.0 workstation, you can simply leave all the backup dist files in the default backups folder.

If the new N4.0 Supervisor or engineering workstation is on a different PC, copy the backup dist files to a location on that PC.

Secure Niagara 4 license files

Any platform converted from NiagaraAX to Niagara 4 requires a Niagara 4 license file, as existing AX licenses files are *incompatible*. Without an N4.0 license file, any N4.0 JACE controller or Supervisor is unable to run a station. An N4.0 engineering workstation without an N4.0 license file is also unable to run Niagara 4 Workbench.

NOTE:

You can perform N4 migration work on all archived AX-3.8 stations without all N4.0 licenses secured, providing you have at least *one* N4.0 Supervisor or engineering workstation installed and licensed. However, you should not start upgrading JACE controllers from AX-3.8 to N4.0 without first having their licenses secured and ready to install.

- Note that any JACE controller converted to N4.0 uses the same Niagara host ID as it did NiagaraAX. Reference this host ID when you request N4.0 licenses.
- Typically any Supervisor PC or engineering workstation upgraded from AX-3.8 to N4.0 uses the identical Niagara host ID as well, providing its operating system remains unchanged. However, note that a Niagara 4 (N4.0) license is not “backward compatible” with any NiagaraAX install.

  Note that in “hybrid systems” that have both N4.0 and AX-3.8 hosts, you need to *retain* AX-3.8 Workbench, to use for possible station modifications to JACE controllers that remain at AX-3.8.

  Be sure to *back up* existing NiagaraAX license files for Windows hosts, and store somewhere safe. In other words, when requesting and receiving an N4.0 license file for your N4.0 Supervisor, do not also copy it back to any *NiagaraAX* home’s licenses folder on that same PC—otherwise AX Workbench will no longer work. This can also occur if a NiagaraAX host downloads the (N4) license offered by the licensing server.

Note any *new* installation of Niagara 4 (N4.0) on a PC will calculate the Niagara 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 need to submit the license request and receive either the license file, or confirmation of it on the licensing server, before you can use Niagara 4 Workbench. For related details, refer to the *Niagara 4 Framework Install Guide*.

Run the N4 migration tool

Included with a Niagara 4 (N4.0) Supervisor or engineering workstation is a command line utility that you run from a Niagara console. This utility is known as the “N4 migration tool”, or sometimes simply the “N4 migrator”.

- “About the N4 migration tool”, page 31
  - “Console choices in migration”, page 33

- "Running the N4 migration tool", page 34
- "N4 migration tool command usage", page 35
- "N4 migration details", page 42
  - "N4 migration execution", page 42
  - "N4 migration report (log file)", page 42

About the N4 migration tool

The N4 migration tool takes a source file, typically an AX-3.8 station backup .dist file, and outputs a "migrated" N4.0-compatible station folder. You can then use Niagara 4 Workbench to open a platform connection to the target N4.0 platform (for example, a converted controller), and install this station using the N4.0 platform Station Copier.

NOTE:

Important points to remember are:

- A source AX 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 with execution details.
- The N4 migration tool does not require NiagaraAX (AX-3.8) installed on the same PC, only a path to the source AX-3.8 backup .dist files.
- Run the migration tool from the standalone Niagara**Console** window, typically launched from the Windows Start menu.

The following figure shows output from the migration tool processing a backup of a very simple Supervisor station, using the standalone Niagara**Console** window.

NOTE: The embedded Workbench Console is not supported, as it does not properly process the necessary user inputs that may be required if your Niagara AX station contains passwords that need to be encrypted with a pass phrase for Niagara 4

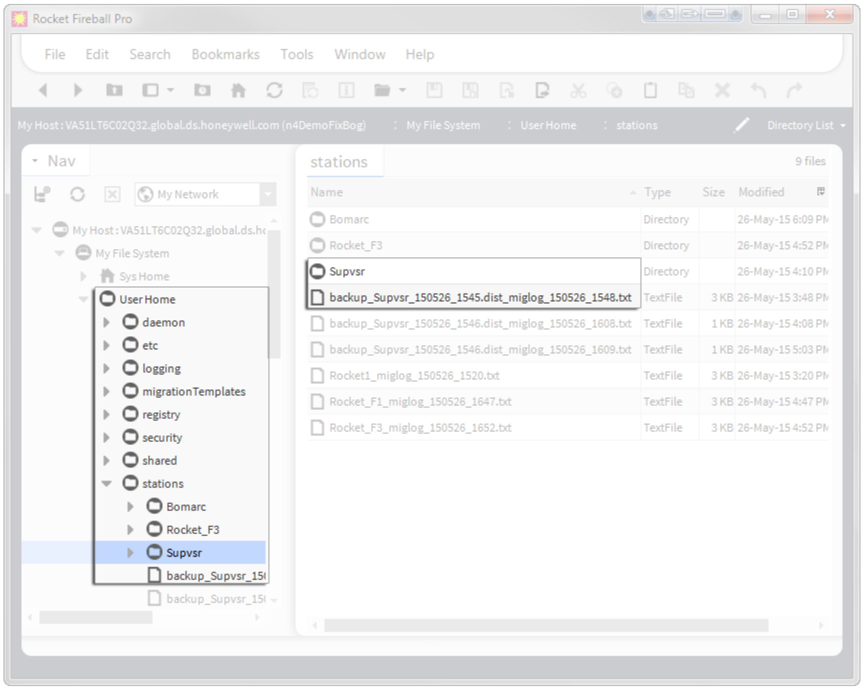
Figure 13 Example N4 migration tool processing a Supervisor station backup

```
-------------------------------------------------------------------------------
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 pauses and prompts for migration template (1 or 2). After you enter this, the tool continues processing and completes the migration.

The migration tool posts information while processing the AX-3.8 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 14 Example migration tool output in Workbench User Home (station folder, migration log file)



Name	Type	Size	Modified
Bomarc	Directory		26-May-15 6:09 PM
Rocket_F3	Directory		26-May-15 4:52 PM
Supvsr	Directory		26-May-15 4:10 PM
backup_Supvsr_150526_1545.dist_miglog_150526_1548.txt	TextFile	3 KB	26-May-15 3:48 PM
backup_Supvsr_150526_1546.dist_miglog_150526_1608.txt	TextFile	1 KB	26-May-15 4:08 PM
backup_Supvsr_150526_1546.dist_miglog_150526_1609.txt	TextFile	1 KB	26-May-15 5:03 PM
Rocket1_miglog_150526_1520.txt	TextFile	3 KB	26-May-15 3:20 PM
Rocket_F1_miglog_150526_1647.txt	TextFile	3 KB	26-May-15 4:47 PM
Rocket_F3_miglog_150526_1652.txt	TextFile	3 KB	26-May-15 4:52 PM

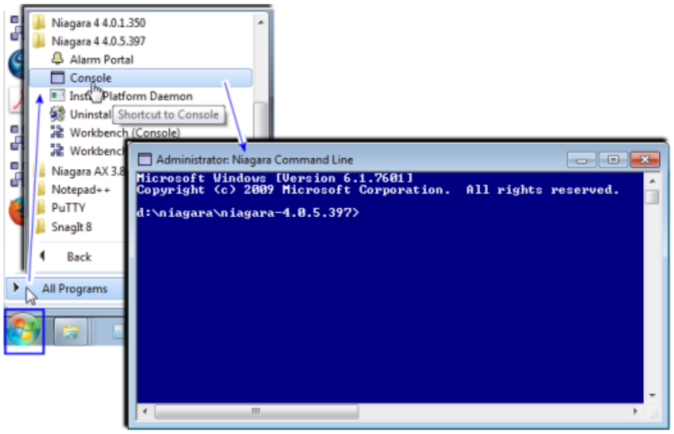
In the example figure shown above, notice that the migration processed created a "Supvsr" folder and an associated log file in the stations folder. The target stationname is automatically determined from the station name inside the station backup .dist file.

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

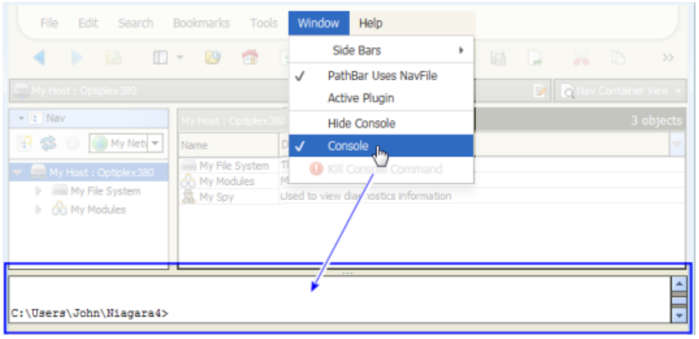
You should use only the Niagara console and not the embedded Workbench Console to run the migration tool.

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.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

d:\niagara\niagara-4.0.5.397>
```

IMPORTANT: The “ Workbench embedded console”, that is, the Console prompt available within Workbench (shown here), cannot be used for migration purposes.



```
C:\Users\John\Niagara4>
```

Running the N4 migration tool

With Niagara 4 installed on your PC, you can run the N4 migration tool on an 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” on page 34, page 37.

Prerequisite: Access to saved AX-3.8 station backups, for migrating stations. See “Making source AX-3.8 backups” on page 25, page 27.

Running the N4 migration tool

Step 1 At the Niagara 4 Workbench 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:`C:\niagara\Niagara-3.8.38\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 AX-3.8 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 “target” is the stations folder of your Niagara 4 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 theWorkbench 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 the Niagara 4 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 "N4 migration tool command usage" on page 32, page 35. For general information, see "About the N4 migration tool" on page 28, page 31.

N4 migration tool command usage

Command line usage for the N4 migration tool is:

n4mig [options] <source> [target]

Parameter	Required?	Description / Notes
source	Yes	Typically an AX-3.8 station backup .dist file, but in 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 else a path relative to the current directory of the Niagara 4 console.
target	No	(Optional) The output station folder name to write the migrated station or BOG file. An absolute file path or a relative path can be entered. Any relative path is relative to the stations subfolder of your " Workbench **User Home**", i.e. `C:\Users\userName\Niagara4.0\<brand>\stations` 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, then it will be derived from the name of the source, its parent folder, or the name of a folder within the archive - whatever fits best. The exception is a source named config.bog. In that case, target is required. If the supplied target is a relative path, then the result will be placed relative to the root of `niagara_user_home/stations`. If you do specify a target folder name, it cannot already exist (by default an existing station is never overwritten); however, any filepath above it must exist. For a related example, see "Migrator syntax example 2" on page 33, page 36.

Options	Description / Notes
-version	Print NRE version information for this Workbench instance.
-log:level	Specify the log level, where level is one of the following: all, finest, finer, fine, config, info, warning, severe, off The default is "info". This produces the text log file "migration report", listing *all the changes* made to the AX station in the N4 migration. This file is created in the root of the stations folder in your Workbench **User Home**. Of course the default ("info") also includes any more problematic "warning" or "severe" messages. Note after becoming familiar with migrations, you could specify the "warning" log level instead—that lists *only* the more problematic entries, shortening the log report.
-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. Note (by default) you see this same information if you enter the n4mig command alone, that is without any other parameters or options.

Options	Description / Notes
-o	Overwrite the target station folder, if it already exists.
-t:<template>	When migrating a station backup .dist, this is the template type to use, where template is one of the *migration station template types* available on your system. Choices are: 1 = Controller migration template (e.g. ControllerMigrationTemplate.ntpl) 2 = Supervisor migration templage (e.g. SupervisorMigrationTemplate.ntpl) **NOTE:** You can also use 'c' for the default Controller and 's' for the default Supervisor template. 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 *Niagara 4.0 Templates* document for more details on working with Niagara 4 templates.

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 `D:\niagara\Niagara-3.8.38\backups>`

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

```
D:\niagara\Niagara-3.8.38\backups>n4mig -t:1 backup_J300E_W_150210_1715.dist
```

The migration tool processes the `J300E_W` `backup .dist` file in the current directory, as a JACE station, and the target output goes under the stations folder of your current N4.0 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\Niagara4.0\<brand>\stations\J300E_W (complete station folder)
- C:\Users\userName\Niagara4.0\<brand>\stations\J300E_W_miglog_150212_0846.txt (log file or "migration report)

Note that 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, if you had *renamed* the source file from backup_J300E_W_150210_1715.dist to say J3Wbu.dist before running the migration, and run the command (again, specifying only the source):

```
D:\niagara\Niagara-3.8.38\backups>n4mig -t:1 J3Wbu.dist
```

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

- `C:\Users\userName\Niagara4.0\<brand>\stations\J300E_W` (complete station folder)
- `C:\Users\userName\Niagara4.0\<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.

```
D:\niagara\niagara-4.0.10.2>n4mig -t:2 -log:finer D:\niagara\Niagara-3.8.38\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)

Note 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).

Also note that 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).

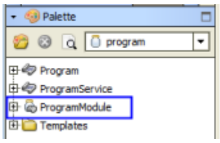
Therefore, when a migrating a station from an AX-3.8 station backup .dist file, it is typically best to specify only the source .dist file name, and not specify any target. An example of this is described in “Migrator syntax example 1” on page 33, page 36. After the migration tool completes, if needed, 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 N4 station.

Prerequisite: Working knowledge of the AX-3.8 ProgramModule (available in the `program` module’s palette), including its **Program Module Builder** view.

Figure 15 ProgramModule in AX-3.8 `program` palette



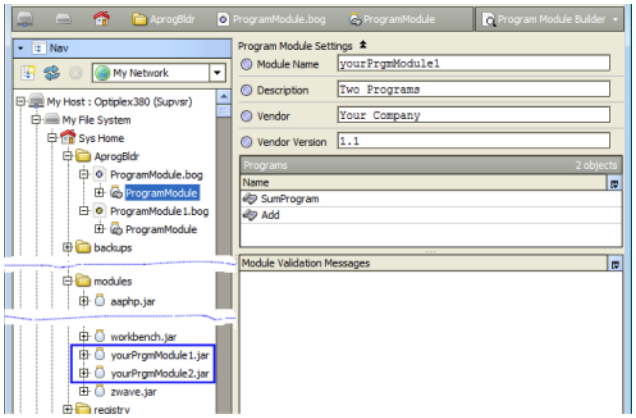
Note in a NiagaraAX installation, there are different ways the ProgramModule feature can be used to create versioned module JAR files that contain one or more Program objects:

- ProgramModule components copied into a folder under the AX-3.8 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 16 ProgramModules saved as BOG files in the AX-3.8 Workbench file space



Program Module Settings

Module Name: yourPrgmModule1

Description: Two Programs

Vendor: Your Company

Vendor Version: 1.1

Programs — 2 objects
Name
SumProgram
Add

The figure above shows two such ProgramModule BOG files, under an "AprogBldr" folder in the AX-3.8 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 AX-3.8 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 AX-3.8 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 N4 migration on the ProgramModules.

In all but the last case, N4 migration of the modules is straightforward. Migration requires running the source AX-3.8 ProgramModule through the N4 migration tool (either as a BOG file, or if within a station, in a station backup .dist file). Next you use N4 Workbench on each migrated ProgramModule to recompile the contained Program(s) and then rebuild the module, which refactors it for Niagara 4.

The following related tasks apply:

- "Migrating modules from ProgramModules saved as AX-3.8 BOG files", page 38
- "Migrating modules from ProgramModules in a station database", page 40

Migrating modules from ProgramModules saved as AX-3.8 BOG files

If you have saved one or more ProgramModule components in your AX-3.8 Workbench file space, you can run the folder (BOG) file containing them through the N4 migration tool.

Prerequisites:

NOTE:

If necessary, you can do this on a PC still running an AX-3.8 Supervisor station, but with both N4.0 Workbench and AX-3.8 Workbench installed.

Step 1 At the Niagara 4 Workbench PC, open a Niagara console. This can be either a standalone Console window or the Console available in Workbench. See "Console choices in migration" on page 30, page 33.

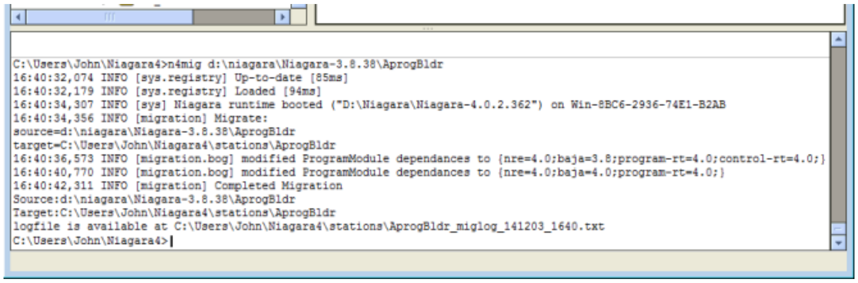
Step 2 Enter the migration tool command (n4mig) using syntax that specifies the source folder with ProgramModule BOG files. If using an absolute file path, an example command could look similar to:

n4mig D:\niagara\Niagara-3.8.38\AprogBldr

for the “AprogBldr” folder as shown in the previous Figure 16 Figure 17 on page 34, page 38. Leaving the target unspeciified (as here), results in a subfolder under stations in your Niagara 4 “ Workbench **User Home**”.

NOTE:

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



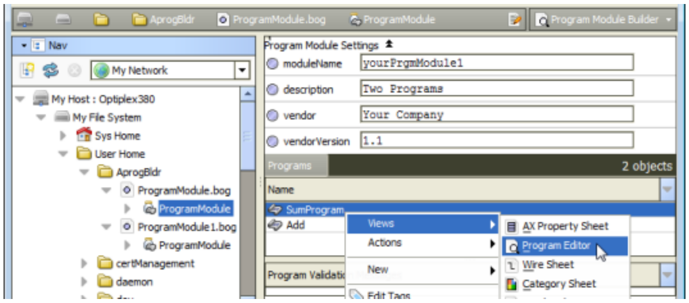
```
C:\Users\John\Niagara4>n4mig d:\niagara\Niagara-3.8.38\AprogBldr
16:40:32,074 INFO [sys.registry] Up-to-date [85ms]
16:40:32,179 INFO [sys.registry] Loaded [94ms]
16:40:34,307 INFO [sys] Niagara runtime booted ("D:\Niagara\Niagara-4.0.2.362") on Win-8BC6-2936-74E1-B2AB
16:40:34,356 INFO [migration]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

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

Step 3 If the target folder was created in your stations folder, move (cut and paste it) to another location in your Workbench **User Home**. You can do this in the Workbench Nav tree, using right-click commands.

Step 4 In the Workbench 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.



Program Module Settings
moduleName yourPrgmModule1
description Two Programs
vendor Your Company
vendorVersion 1.1

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.

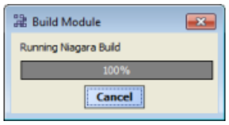
Step 8 Click **Save Bog**.

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.

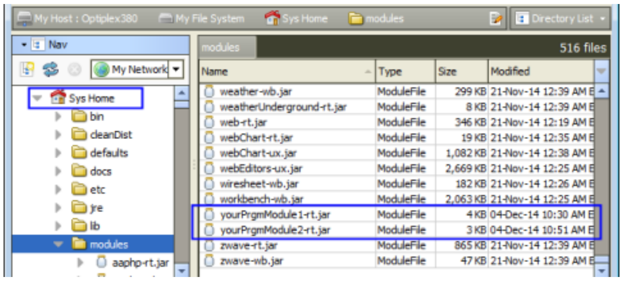
Step 10 Double-click the ProgramModule for its **Program Module Builder** view, then click **Build**.



Running Niagara 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.



Name	Type	Size	Modified
weather-wb.jar	ModuleFile	299 KB	21-Nov-14 12:39 AM E
weatherUnderground-rt.jar	ModuleFile	8 KB	21-Nov-14 12:39 AM E
web-rt.jar	ModuleFile	346 KB	21-Nov-14 12:19 AM E
webChart-rt.jar	ModuleFile	19 KB	21-Nov-14 12:35 AM E
webChart-ux.jar	ModuleFile	1,082 KB	21-Nov-14 12:38 AM E
webEditors-ux.jar	ModuleFile	2,669 KB	21-Nov-14 12:25 AM E
wiresheet-wb.jar	ModuleFile	182 KB	21-Nov-14 12:26 AM E
workbench-wb.jar	ModuleFile	2,063 KB	21-Nov-14 12:25 AM E
yourPrgmModule1-rt.jar	ModuleFile	4 KB	04-Dec-14 10:30 AM E
yourPrgmModule2-rt.jar	ModuleFile	3 KB	04-Dec-14 10:51 AM E
zwave-rt.jar	ModuleFile	865 KB	21-Nov-14 12:39 AM E
zwave-wb.jar	ModuleFile	47 KB	21-Nov-14 12:39 AM E

As shown above, the modules named `yourPrgmModule1-rt.jar` and `yourPrgrmModule2-rt.jar` were created by the two ProgramModules.

Step 12 When finished migrating modules, close and restart Niagara 4 Workbench. This is necessary to update the Workbench 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 N4 migration tool" on page 31, page 34.

Migrating modules from ProgramModules in a station database

If ProgramModule components are used within an AX-3.8 station (instead of in standalone BOG files in our AX-3.8 Workbench file space), you need to migrate that station to N4 in the standard way, that is using an AX-3.8 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 N4 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:

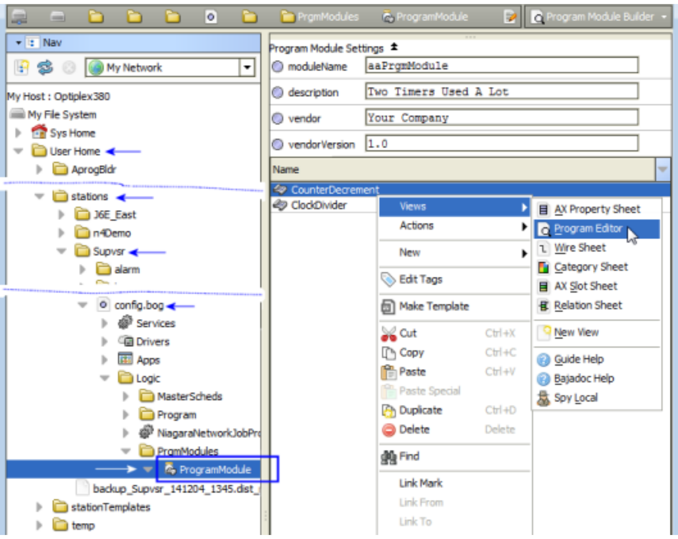
If necessary, you can do this on a PC still running an AX-3.8 Supervisor station, but with both N4.0 Workbench and AX-3.8 Workbench installed.

Step 1 Run the N4 migration tool on the saved AX-3.8 station backup .dist file.

Step 2 In the N4 Workbench Nav tree, expand the `stations` folder under your Workbench **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** view.



moduleName `aaPrgmModule`
description `Two Timers Used A Lot`
vendor `Your Company`
vendorVersion `1.0`

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.

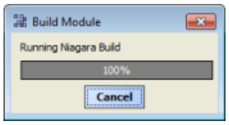
1. If successful, the Program status should be “up-to-date”, and the **Save Bog** icon should be available. Click **Save Bog**.

   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 for Niagara 4” on page 39, page 43.

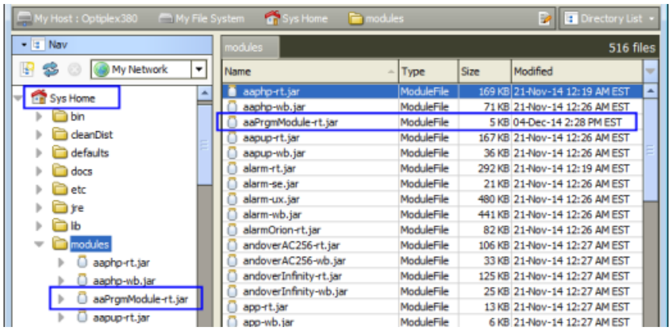
Step 7 Repeat steps 5 and 6 for all Programs in the selected ProgramModule, so that *all* Programs have been successfully compiled and saved.

Step 8 Double-click the ProgramModule for its **Program Module Builder** view, then click **Build**.

2. 

   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.

3. 

Name	Type	Size	Modified
aaphp-rt.jar	ModuleFile	169 KB	21-Nov-14 12:19 AM EST
aaphp-wb.jar	ModuleFile	71 KB	21-Nov-14 12:26 AM EST
aaPrgmModule-rt.jar	ModuleFile	5 KB	04-Dec-14 2:28 PM EST
aapup-rt.jar	ModuleFile	167 KB	21-Nov-14 12:26 AM EST
aapup-wb.jar	ModuleFile	36 KB	21-Nov-14 12:26 AM EST
alarm-rt.jar	ModuleFile	292 KB	21-Nov-14 12:19 AM EST
alarm-se.jar	ModuleFile	21 KB	21-Nov-14 12:26 AM EST
alarm-ux.jar	ModuleFile	480 KB	21-Nov-14 12:26 AM EST
alarm-wb.jar	ModuleFile	441 KB	21-Nov-14 12:26 AM EST
alarmOrion-rt.jar	ModuleFile	82 KB	21-Nov-14 12:26 AM EST
andoverAC256-rt.jar	ModuleFile	106 KB	21-Nov-14 12:27 AM EST
andoverAC256-wb.jar	ModuleFile	33 KB	21-Nov-14 12:27 AM EST
andoverInfinity-rt.jar	ModuleFile	125 KB	21-Nov-14 12:27 AM EST
andoverInfinity-wb.jar	ModuleFile	25 KB	21-Nov-14 12:27 AM EST
app-rt.jar	ModuleFile	13 KB	21-Nov-14 12:27 AM EST
app-wb.jar	ModuleFile	6 KB	21-Nov-14 12:27 AM EST

   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 Niagara 4 Workbench. This is necessary to update the Workbench registry. Migrated modules should now be ready when running the N4 migration tool on AX-3.8 station backup .dist files, including the station with the source ProgramModules.

N4 migration details

Details on N4 migration are in the following reference topics:

- "N4 migration execution", page 42
- "N4 migration report (log file)", page 42
  - "Inspecting N4 migration reports (log files)", page 43

N4 migration execution

The Niagara 4 migration tool migrates a source artifact to its appropriate N4 target. The typical source is an 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:d:\niagara\Niagara-3.8.38\backups\backup_Supvsr_141117_1707.dist
Target:C:\Users\John\Niagara4\<brand>\stations\Supvsr
```

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

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, you can 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 an N4 equivalent module exists, the registry check provides a converter that can convert the obsolete type into the appropriate N4 type.

N4 migration report (log file)

The log file produced each time you run the N4 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.

NOTE:

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 N4 migration reports (log files)

We recommend you inspect N4 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 N4 station databases, using N4.0 Workbench. For example, certain Program objects may need to be changed and recompiled in Niagara 4, as flagged in the N4 migration report (log file). Other possible station changes may be needed too, in certain circumstances.

See the following for more details:

- "Modifying Program objects for Niagara 4" on page 39, page 43
- "Considerations if a station functions as an oBIX server" on page 40, page 45
- "Preliminary checkout of migrated stations" on page 41, page 46

Modifying Program objects for Niagara 4

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 N4 migration report, or log file. Using Niagara 4 Workbench, you should modify any such Program objects until they recompile successfully, before installing the station.

Prerequisite: 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 Niagara 4 Workbench to modify and recompile Program objects if necessary, resaving the station (config.bog file) when done.

Prerequisites:

NOTE:

If necessary, you can do this on a PC still running an AX-3.8 Supervisor station, but with both N4.0 Workbench and AX-3.8 Workbench installed.

Step 1 Open Niagara 4 (N4.0) 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.

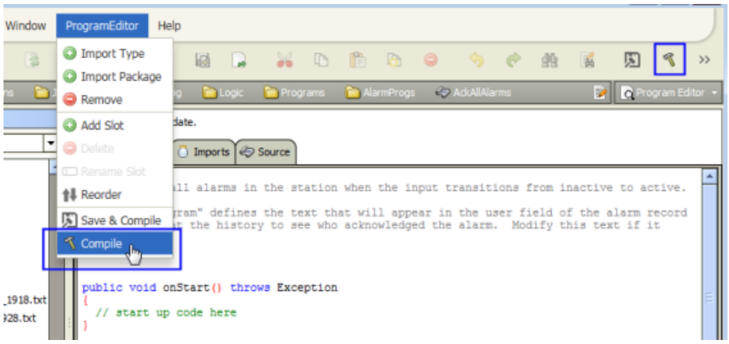
Typically, such stations are under your **User Home**, under the stations folder.

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 N4 migration report (log file).

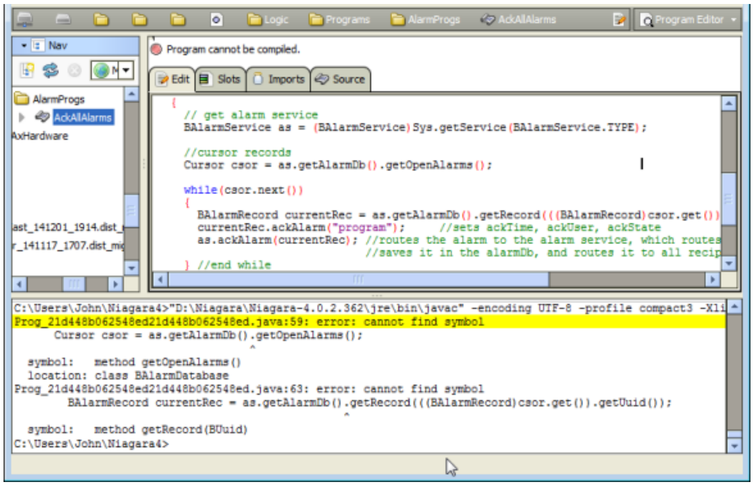
Step 5 Open the **Program Editor** view for the component.
(One way is from the Nav tree, right-click the Program and select **Program Editor**.)

Step 6 In the **ProgramEditor** menu or toolbar, click **Compile**.



```
date.
ll alarms in the station when the input transitions from inactive to active.

ram" defines the text that will appear in the user field of the alarm record
r the history to see who acknowledged the alarm.  Modify this text if it


public void onStart() throws Exception
{
  // start up code here
}
```

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.



Program cannot be compiled.

```
{
  // get alarm service
  BAlarmService as = (BAlarmService)Sys.getService(BAlarmService.TYPE);

  //cursor records
  Cursor csor = as.getAlarmDb().getOpenAlarms();

  while(csor.next())
  {
    BAlarmRecord currentRec = as.getAlarmDb().getRecord(((BAlarmRecord)csor.get())
    currentRec.ackAlarm("program");     //sets ackTime, ackUser, ackState
    as.ackAlarm(currentRec); //routes the alarm to the alarm service, which routes
                             //saves it in the alarmDb, and routes it to all recip
  } //end while
```

```
C:\Users\John\Niagara4>"D:\Niagara\Niagara-4.0.2.362\jre\bin\javac" -encoding UTF-8 -profile compact3 -Xli
Prog_21d448b062548ed21d448b062548ed.java:59: error: cannot find symbol
      Cursor csor = as.getAlarmDb().getOpenAlarms();
                                   ^
  symbol:   method getOpenAlarms()
  location: class BAlarmDatabase
Prog_21d448b062548ed21d448b062548ed.java:63: error: cannot find symbol
        BAlarmRecord currentRec = as.getAlarmDb().getRecord(((BAlarmRecord)csor.get()).getUuid());
                                                 ^
  symbol:   method getRecord(BUuid)
C:\Users\John\Niagara4>
```

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. In order 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 a 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 Niagara 4 (N4.0) 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 begining 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 ther 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 to Niagara 4:

- Assign Role from a previous user

  During migration, a 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 Niagara 4 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 to N4, you can install a migrated JACE station and observe startup behavior. However, expect many errors from driver-sourced items—as you would for any unit disconnected to device networks.

Upgrade/convert platforms and install stations

Once you have AX-3.8 stations migrated to Niagara 4 (N4.0), you can upgrade/convert platforms to N4.0 and install those migrated stations.

Prerequisites:

- You have secured the N4.0 licenses required. Without such licenses, upgraded platforms (including converted JACE controllers) will be unable to run a station. For more details, see "Secure Niagara 4 license files" on page 27, page 30.
- You have N4.0 station databases ready to install. See "Run the N4 migration tool" on page 28, page 30, and "Post-migration database tasks" on page 39, page 43.

Upgrade or install the Niagara 4 Supervisor

Typically you upgrade an AX-3.8 Supervisor to be a Niagara 4 (N4.0) Supervisor before upgrading any of its JACE controllers.

Often, you use this same Supervisor PC to run the N4 migration tool, on both the Supervisor's AX-3.8 station archive, as well as the station archives of all AX-3.8 JACE controllers being migrated to N4.0. In this case, when first installing the Niagara 4 software, you usually *do not choose* to "install and start the (N4.0) platform daemon". This lets the AX-3.8 platform daemon continue hosting the AX-3.8 Supervisor station while you use the N4 migration tool. It also lets you become familiar with N4.0 Workbench, and yet still have AX-3.8 Workbench available for interim AX-3.8 Supervisor tasks.

NOTE:

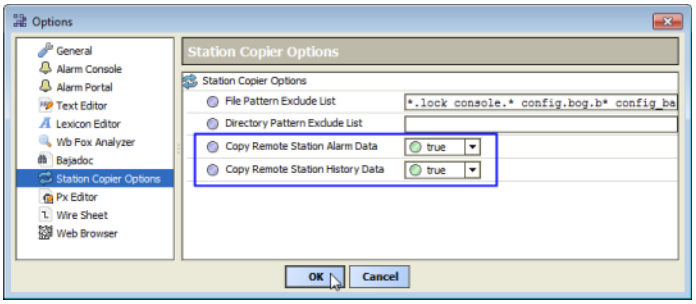
When installing Niagara 4 software on the Supervisor PC, be sure to select the option "this instance of Niagara 4 will be used as an installation tool". Refer to the *Niagara 4 Framework Install Guide* for details.

Alternatively, you may have used a *different* Niagara 4 PC to run the N4 migration tool, and perhaps even to (eventually) use as the new N4.0 Supervisor host for the job.

As different combinations are possible in a transition from AX-3.8 Supervisor to N4.0 Supervisor, this basic sequence applies to all:

1. After the Niagara 4 (N4.0) software is installed on the Supervisor PC and Niagara 4 Workbench is started, you open a platform connection to, and stop the AX-3.8 Supervisor station.
2. With the AX-3.8 Supervisor station stopped, at the N4.0 Supervisor PC (with access to N4.0-migrated station folders), you 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.
3. Start Niagara 4 Workbench, if not already started.
4. From the N4.0 Workbench menubar, click **Tools** > Options →**Station Copier Options**.

   Station Copier Options

   Station Copier Options

   File Pattern Exclude List: *.lock console.* config.bog.b* config_ba

   Directory Pattern Exclude List:

   Copy Remote Station Alarm Data: true

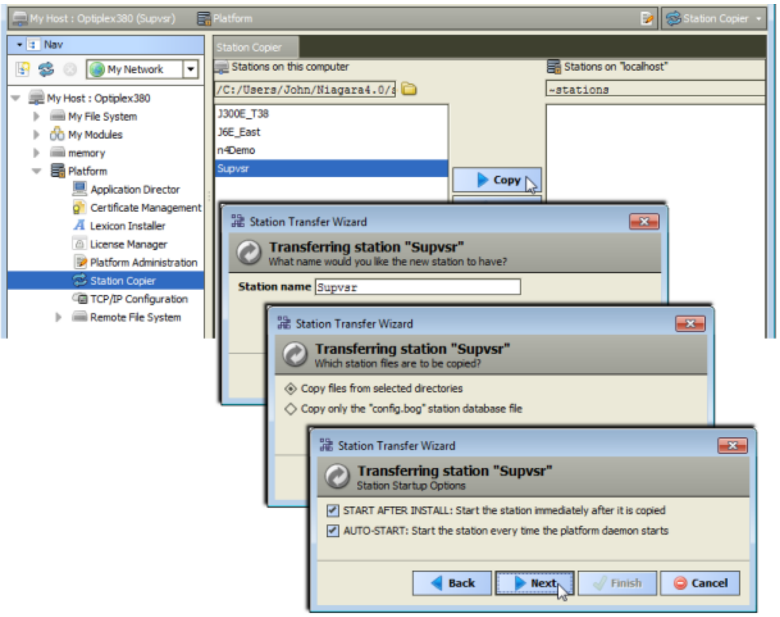
   Copy Remote Station History Data: true

   As shown above in the popup dialog, set to `true` both options to "Copy Remote Station Alarm Data" and "Copy Remote Station History Data", and click **OK**.
5. In Workbench, open a local platform connection
   (in the Nav tree, right-click **My Host**, and choose **Open Platform**).
6. After platform login, go to the **Station Copier** platform view.

   Typically, 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\WindowsLoginName\Niagara4.0\<brand>\stations

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



/C:/Users/John/Niagara4.0/s

~stations

J300E_T38
J6E_East
n4Demo
Supvsr

Transferring station "Supvsr"
What name would you like the new station to have?

Station name: Supvsr

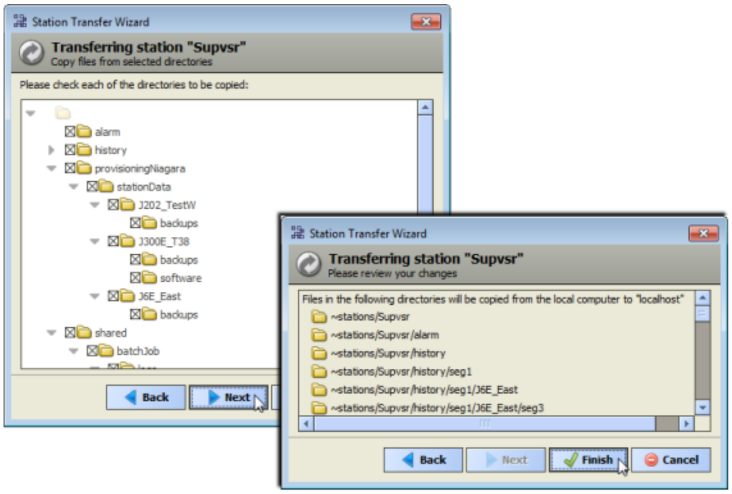
Transferring station "Supvsr"
Which station files are to be copied?

● Copy files from selected directories
○ Copy only the "config.bog" station database file

Transferring station "Supvsr"
Station Startup Options

☑ START AFTER INSTALL: Start the station immediately after it is copied
☑ AUTO-START: Start the station every time the platform daemon starts

As shown above, choose to “copy files from selected directories”, and for both start after install and auto-start. As shown below, select *all station folders* to transfer in this station copy.



Transferring station "Supvsr"
Copy files from selected directories

Please check each of the directories to be copied:

☒ alarm
☒ history
☒ provisioningNiagara
☒ stationData
☒ J202_TestW
☒ backups
☒ J300E_T38
☒ backups
☒ software
☒ J6E_East
☒ backups
☒ shared
☒ batchJob

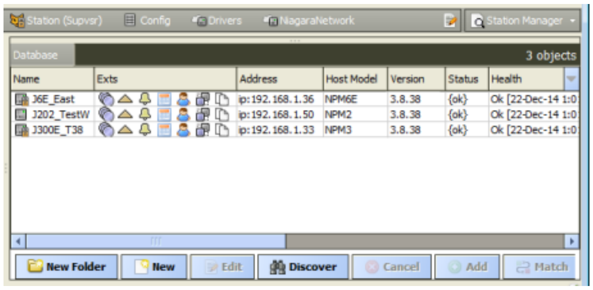
Transferring station "Supvsr"
Please review your changes

Files in the following directories will be copied from the local computer to "localhost"
~stations/Supvsr
~stations/Supvsr/alarm
~stations/Supvsr/history
~stations/Supvsr/history/seg1
~stations/Supvsr/history/seg1/J6E_East
~stations/Supvsr/history/seg1/J6E_East/seg3

8. When the station is copied, open the **Application Director** and observe startup messages.

9. Open a local station (Fox or Foxs) connection to the N4.0 Supervisor when it has started, and begin preliminary checkout.

   Note that stations representing 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 atAX-3.8 (JACE-2s for example), review, and if necessary, adjust client connection properties.



Name	Exts	Address	Host Model	Version	Status	Health
J6E_East		ip:192.168.1.36	NPM6E	3.8.38	{ok}	Ok [22-Dec-14 1:0
J202_TestW		ip:192.168.1.50	NPM2	3.8.38	{ok}	Ok [22-Dec-14 1:0
J300E_T38		ip:192.168.1.33	NPM3	3.8.38	{ok}	Ok [22-Dec-14 1:0

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

10. Save any changes made to the running N4.0 Supervisor before upgrading JACE controllers.

Convert JACE controllers to Niagara 4

Typically, after upgrading an AX-3.8 Supervisor to be a Niagara 4 (N4.0) Supervisor, you convert JACE controllers to N4.0 as well—or at least 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 N4 migration was successful first (all Programs ok, for example), *before installing* the "AXtoN4" dist file. See "Running the N4 migration tool" on page 31, page 34 and "Post-migration database tasks" on page 39, page 43.

Converting a JACE controller to Niagara 4 (N4.0)

Perform the following for each AX-3.8 JACE controller for which you have secured a N4.0 license, and have successfully migrated its station database for N4.0.

Prerequisites:

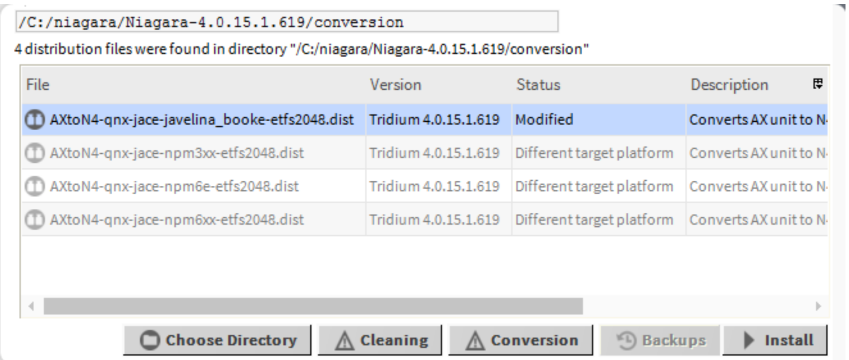
Prerequisite: You performed a preliminary checkout of the N4-migrated station for this JACE controller using Niagara 4 Workbench, making sure it looks OK.

Step 1 At the Supervisor in N4.0 Workbench, open a platform connection to an AX-3.8 JACE controller to be converted, and access the **Distribution File Installer**.

NOTE:

N4 "cleanDist" or other files for other controllers may display, but only the file appropriate for your controller is selectable. 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. Use the buttons at the bottom of the view to navigate and browse for the file, if necessary.

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



/C:/niagara/Niagara-4.0.15.1.619/conversion

4 distribution files were found in directory "/C:/niagara/Niagara-4.0.15.1.619/conversion"

File	Version	Status	Description
AXtoN4-qnx-jace-javelina_booke-etfs2048.dist	Tridium 4.0.15.1.619	Modified	Converts AX unit to N
AXtoN4-qnx-jace-npm3xx-etfs2048.dist	Tridium 4.0.15.1.619	Different target platform	Converts AX unit to N
AXtoN4-qnx-jace-npm6e-etfs2048.dist	Tridium 4.0.15.1.619	Different target platform	Converts AX unit to N
AXtoN4-qnx-jace-npm6xx-etfs2048.dist	Tridium 4.0.15.1.619	Different target platform	Converts AX unit to N

Only *one* of several listed should be selectable. For example, if you are converting a JACE-6E controller:

AXtoN4-qnx-jace-npm6e-etfs2048.dist

Step 2 Select the appropriate "AX to N4" 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 JACE controller (using AX-3.8 Workbench), access the platform **Certificate Manager** view, and do the following:

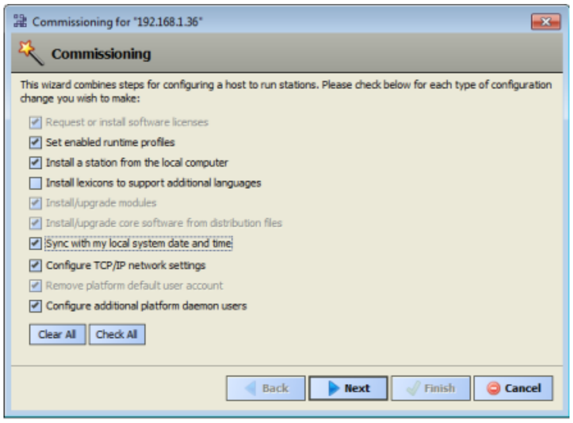
a. On the **User Trust Store** tab, click **Import**, and navigate to select the previously saved public key certificate (typically common among all platforms at the job).

b. On the **User Key Store** tab, click **Import**, and navigate to select the previously saved private key certificate unique to this platform.

c. Access the platform **Platform Administration** view, click the **Change SSL Settings** button, and select the new server certificate you just imported.

d. Run the platform **Commissioning Wizard** (right-click the **Platform** node, and select **Commissioning Wizard**).

In this initial Niagara 4 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).



Commissioning for "192.168.1.36"

Commissioning

This wizard combines steps for configuring a host to run stations. Please check below for each type of configuration change you wish to make:

- ☑ Request or install software licenses
- ☑ Set enabled runtime profiles
- ☑ Install a station from the local computer
- ☐ Install lexicons to support additional languages
- ☑ Install/upgrade modules
- ☑ Install/upgrade core software from distribution files
- ☑ Sync with my local system date and time
- ☑ Configure TCP/IP network settings
- ☑ Remove platform default user account
- ☑ Configure additional platform daemon users

In most cases you select to:

- Install the Niagara 4 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 N4 station that you migrated for this *controller*, choosing to:

  "Copy files from selected directories"

  NOTE:

  You typically *select all directories* shown in the migrated JACE station, including the alarm and history subfolder (to migrate prior alarms and histories). Apart from the N4 migration, this is an unusual practice.

  If you are installing the same station in different/multiple JACE 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:

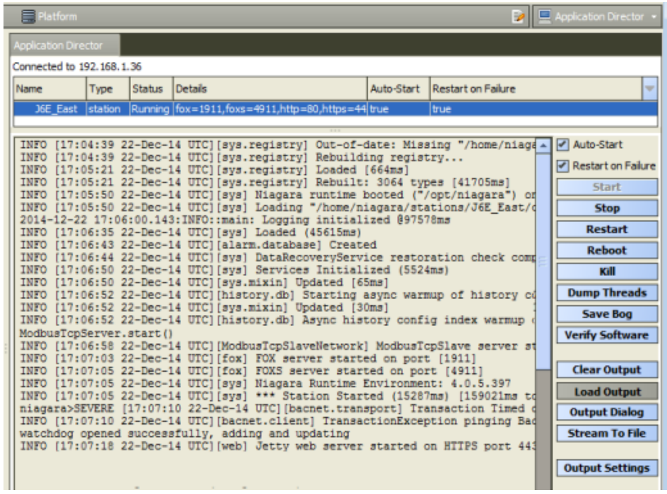
  For this platform session, Workbench remembers the first platform admin user you created to replace the factory default platform user. This simplifies re-connection after the controller reboots.

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 52 topic..

Step 5 Review and finish the commissioning wizard, to install all selected items and reboot the controller. After several minutes, the conversion to an N4.0 platform 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 N4.0 JACE controller, using a *new platform account* you entered in the wizard. Go to the **Application Director** and view the station standard output.



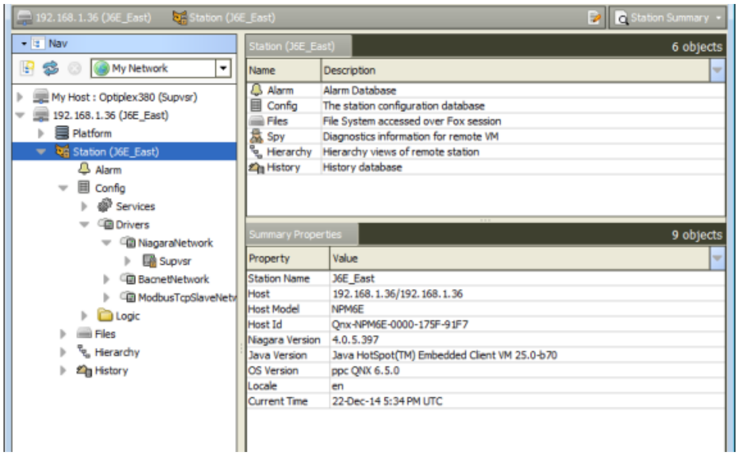
Connected to 192.168.1.36

Name	Type	Status	Details	Auto-Start	Restart on Failure
J6E_East	station	Running	fox=1911,foxs=4911,http=80,https=44	true	true

```
INFO [17:04:39 22-Dec-14 UTC][sys.registry] Out-of-date: Missing "/home/niaga
INFO [17:04:39 22-Dec-14 UTC][sys.registry] Rebuilding registry...
INFO [17:05:21 22-Dec-14 UTC][sys.registry] Loaded [664ms]
INFO [17:05:21 22-Dec-14 UTC][sys.registry] Rebuilt: 3064 types [41705ms]
INFO [17:05:50 22-Dec-14 UTC][sys] Niagara runtime booted ("/opt/niagara") on
INFO [17:05:50 22-Dec-14 UTC][sys] Loading "/home/niagara/stations/J6E_East/c
2014-12-22 17:06:00.143:INFO::main: Logging initialized @97578ms
INFO [17:06:35 22-Dec-14 UTC][sys] Loaded (45615ms)
INFO [17:06:43 22-Dec-14 UTC][alarm.database] Created
INFO [17:06:44 22-Dec-14 UTC][sys] DataRecoveryService restoration check comp
INFO [17:06:50 22-Dec-14 UTC][sys] Services Initialized (5524ms)
INFO [17:06:50 22-Dec-14 UTC][sys.mixin] Updated [65ms]
INFO [17:06:52 22-Dec-14 UTC][history.db] Starting async warmup of history co
INFO [17:06:52 22-Dec-14 UTC][sys.mixin] Updated [30ms]
INFO [17:06:52 22-Dec-14 UTC][history.db] Async history config index warmup c
ModbusTcpServer.start()
INFO [17:06:58 22-Dec-14 UTC][ModbusTcpSlaveNetwork] ModbusTcpSlave server st
INFO [17:07:03 22-Dec-14 UTC][fox] FOX server started on port [1911]
INFO [17:07:05 22-Dec-14 UTC][fox] FOXS server started on port [4911]
INFO [17:07:05 22-Dec-14 UTC][sys] Niagara Runtime Environment: 4.0.5.397
INFO [17:07:05 22-Dec-14 UTC][sys] *** Station Started (15287ms) [159021ms to
niagara>SEVERE [17:07:10 22-Dec-14 UTC][bacnet.transport] Transaction Timed o
INFO [17:07:10 22-Dec-14 UTC][bacnet.client] TransactionException pinging Bac
watchdog opened successfully, adding and updating
INFO [17:07:18 22-Dec-14 UTC][web] Jetty web server started on HTTPS port 443
```

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.



Station (J6E_East) 6 objects

Name	Description
Alarm	Alarm Database
Config	The station configuration database
Files	File System accessed over Fox session
Spy	Diagnostics information for remote VM
Hierarchy	Hierarchy views of remote station
History	History database

Summary Properties 9 objects

Property	Value
Station Name	J6E_East
Host	192.168.1.36/192.168.1.36
Host Model	NPM6E
Host Id	Qnx-NPM6E-0000-175F-91F7
Niagara Version	4.0.5.397
Java Version	Java HotSpot(TM) Embedded Client VM 25.0-b70
OS Version	ppc QNX 6.5.0
Locale	en
Current Time	22-Dec-14 5:34 PM UTC

Step 8 For each remaining AX-3.8 controller to be converted, repeat the above steps

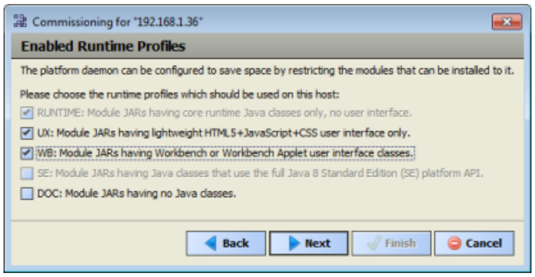
When done with this, all JACE controllers to be converted should be running stations as N4.0 platforms, as well as the Supervisor station.

Controller conversion examples

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

Example conversion steps

Figure 17 Example “Set enable runtime profiles” step



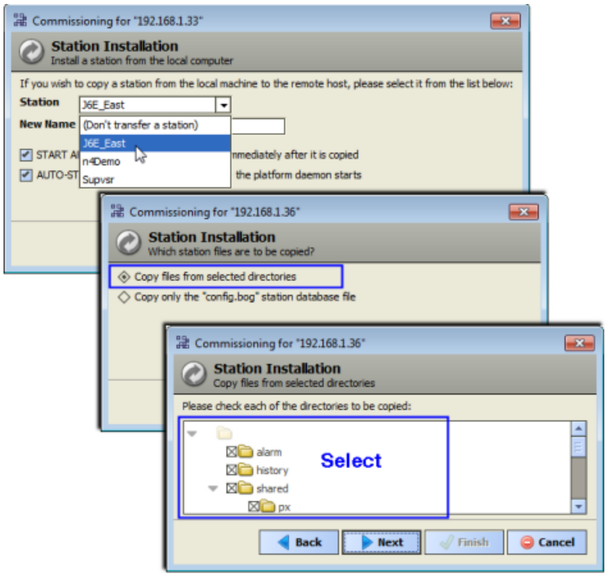
Enabled Runtime Profiles

The platform daemon can be configured to save space by restricting the modules that can be installed to it.

Please choose the runtime profiles which should be used on this host:

☑ RUNTIME: Module JARs having core runtime Java classes only, no user interface.
☑ UX: Module JARs having lightweight HTML5+JavaScript+CSS user interface only.
☑ WB: Module JARs having Workbench or Workbench Applet user interface classes.
□ SE: Module JARs having Java classes that use the full Java 8 Standard Edition (SE) platform API.
□ DOC: Module JARs having no Java classes.

Figure 18 Example “Install station” step (pick migrated station)



Commissioning for "192.168.1.33"

Station Installation
Install a station from the local computer

If you wish to copy a station from the local machine to the remote host, please select it from the list below:

Station: J6E_East

New Name: (Don't transfer a station) / J6E_East / n4Demo / Supvsr

☑ START A… …mmediately after it is copied

☑ AUTO-ST… the platform daemon starts

Commissioning for "192.168.1.36"

Station Installation
Which station files are to be copied?

◈ Copy files from selected directories

◇ Copy only the "config.bog" station database file

Commissioning for "192.168.1.36"

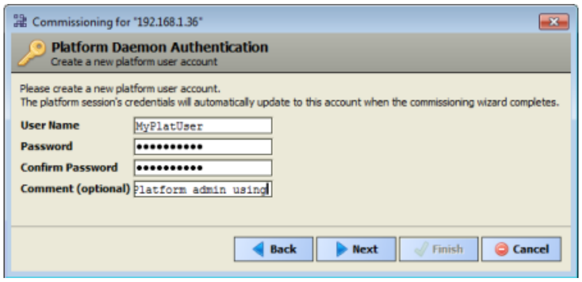
Station Installation
Copy files from selected directories

Please check each of the directories to be copied:

☒ alarm
☒ history
☒ shared
☒ px

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)



Commissioning for "192.168.1.36"

Platform Daemon Authentication
Create a new platform user account

Please create a new platform user account.
The platform session's credentials will automatically update to this account when the commissioning wizard completes.

User Name: MyPlatUser
Password: ••••••••••
Confirm Password: ••••••••••
Comment (optional): Platform admin using

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.

Convert Niagara 4 controllers to NiagaraAX

In some cases it may be desirable or necessary to convert a controller from Niagara 4 to NiagaraAX. During the downgrade commissioning of a JACE, all station data is deleted. SSL certificates and key files will be deleted. Niagara 4 stations cannot be converted to NiagaraAX stations. 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 Ethernet TCP/IP configuration. Platform credentials will revert to factory defaults. N4 license is not be retained and a NiagaraAX license will be required. All station data will be deleted. N4 stations be cannot be converted down to AX.

Converting (downgrading) a JACE controller from Niagara 4 (N4.0) to NiagaraAX

Perform the following for each N4.0 JACE controller for which you have secured a NiagaraAX license. Also, if you want 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 N4.0 Workbench, open a platform connection to the N4.0 JACE 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 on **Conversion**.

NOTE:

Some non-applicable files, such as N4 "cleanDist", "AXtoN4... Dist" or others 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 "N4 to AX " 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 4 After this dist file installs and the controller reboots, re-open a platform connection using AX-3.8 Workbench (port 3011) using 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 an AX-3.8 controller, see the *JACE NiagaraAX 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 Niagara 4 system verification

Topics covered in this chapter

Following the migration and initial upgrade of a system from AX-3.8 to Niagara 4 (N4.0), 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 had 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 JACE controller stations, via the

NiagaraNetwork. 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 JACE 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, note you need to use AX-3.8 Workbench for this.

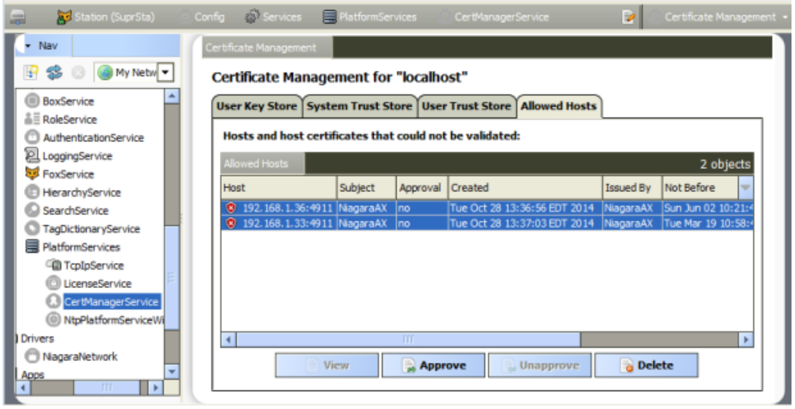
Some variation of the following sequence may be required:

1. In N4.0 Workbench, re-open the Supervisor station and go to the **Station Manager** view of its **NiagaraNetwork**.

   Note if *new* JACE controllers were added (not migrated from AX) there will not be an existing station child for them—you need to select and "Add" them.

2. For each NiagaraStation in the Supervisor station's NiagaraNetwork, verify communications are ok. The status of all stations in the network should be "ok."

   In cases where Foxs (SSL) 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.

   Certificate Management for "localhost"

   Hosts and host certificates that could not be validated:

   Allowed Hosts 2 objects

Host	Subject	Approval	Created	Issued By	Not Before
192.168.1.36:4911	NiagaraAX	no	Tue Oct 28 13:36:56 EDT 2014	NiagaraAX	Sun Jun 02 10:21:
192.168.1.33:4911	NiagaraAX	no	Tue Oct 28 13:37:03 EDT 2014	NiagaraAX	Tue Mar 19 10:58:

   The figure above shows "Allowed Hosts" exceptions for migrated JACEs that need approval. This may not be necessary, if previously-exported SSL certificates were imported before commissioning.

3. 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 (SSL) is being used by a Hotspot JACE station back to the Supervisor (and possibly other JACE stations) you may need to go the JACE station's **CertManagerService** platform service, and perform other actions.

4. 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.

Niagara provisioning notes

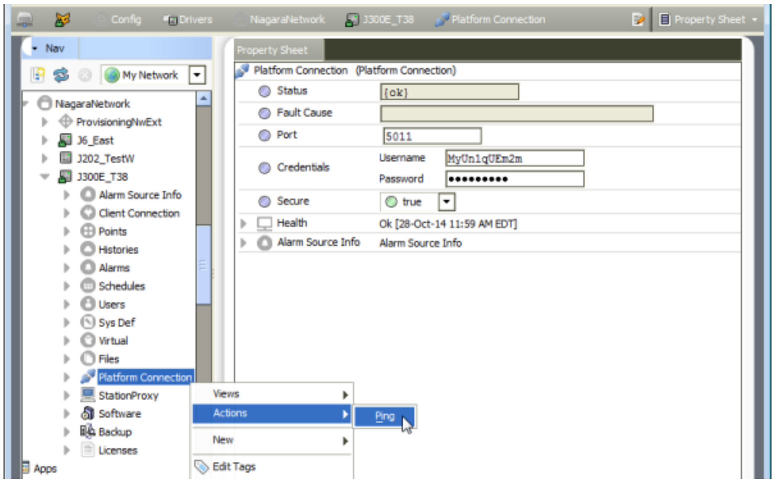
For any Supervisor migrated to N4.0 that was configured for provisioning stations in its **NiagaraNetwork**, that is, using the `provisioningNiagara` components (e.g. **ProvisioningNwExt**, also **BatchJobService**), there may be related post-migration measures needed.

- "Verify Supervisor to controller platform daemon communications", page 57

- “Provisioning considerations when AX-3.8 JACE controllers remain”, page 57

Verify Supervisor to controller platform daemon communications

Verify communications from the Supervisor to the platform daemon of each JACE controller are ok, by expanding the Supervisor’s **NiagaraNetwork**, and issuing a ping from the **PlatformConnection** device extension of each child **NiagaraStation**.



Platform Connection (Platform Connection)

Status {ok}

Fault Cause

Port 5011

Credentials Username MyUn1qUEm2m Password ●●●●●●●●●

Secure true

Health Ok [28-Oct-14 11:59 AM EDT]

Alarm Source Info Alarm Source Info

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 N4.0-migrated controller. And/or, perhaps (again) access the Supervisor’s **CertManagerService** platform service, and perform other actions.

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

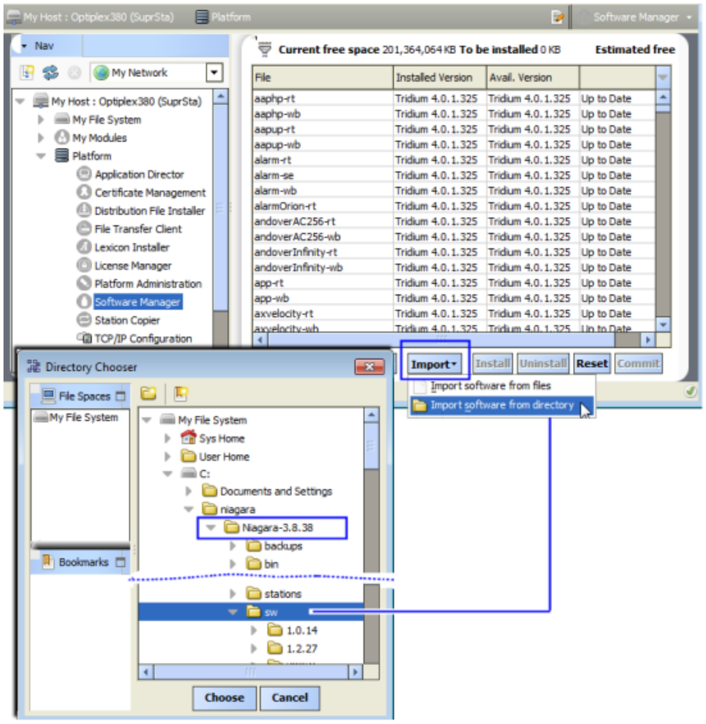
Provisioning considerations when AX-3.8 JACE controllers remain

Sometimes one or more JACE controllers may need to remain running AX-3.8, yet continue to be in the migrated N4.0 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, we recommend you *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, available for provisioning of AX-3.8 platforms.

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

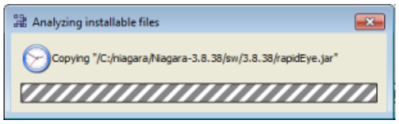
1. Use N4.0 Workbench to open a *platform connection* to the N4.0 Supervisor, then access the **Software Manager** view.
2. Click the **Import** button at the bottom, and then “`Import software from directory`”.



Current free space 201,364,064 KB **To be installed** 0 KB **Estimated free**

File	Installed Version	Avail. Version	
aaphp-rt	Tridium 4.0.1.325	Tridium 4.0.1.325	Up to Date
aaphp-wb	Tridium 4.0.1.325	Tridium 4.0.1.325	Up to Date
aapup-rt	Tridium 4.0.1.325	Tridium 4.0.1.325	Up to Date
aapup-wb	Tridium 4.0.1.325	Tridium 4.0.1.325	Up to Date
alarm-rt	Tridium 4.0.1.325	Tridium 4.0.1.325	Up to Date
alarm-se	Tridium 4.0.1.325	Tridium 4.0.1.325	Up to Date
alarm-wb	Tridium 4.0.1.325	Tridium 4.0.1.325	Up to Date
alarmOrion-rt	Tridium 4.0.1.325	Tridium 4.0.1.325	Up to Date
andoverAC256-rt	Tridium 4.0.1.325	Tridium 4.0.1.325	Up to Date
andoverAC256-wb	Tridium 4.0.1.325	Tridium 4.0.1.325	Up to Date
andoverInfinity-rt	Tridium 4.0.1.325	Tridium 4.0.1.325	Up to Date
andoverInfinity-wb	Tridium 4.0.1.325	Tridium 4.0.1.325	Up to Date
app-rt	Tridium 4.0.1.325	Tridium 4.0.1.325	Up to Date
app-wb	Tridium 4.0.1.325	Tridium 4.0.1.325	Up to Date
axvelocity-rt	Tridium 4.0.1.325	Tridium 4.0.1.325	Up to Date
axvelocity-wb	Tridium 4.0.1.325	Tridium 4.0.1.325	Up to Date

3. 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**.



Copying "/C:/niagara/Niagara-3.8.38/sw/3.8.38/rapidEye.jar"

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 displays.

Note 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**. Among these changes are the following:

- Permissions moved to Roles, page 58
- Authentication is user-specific, page 60

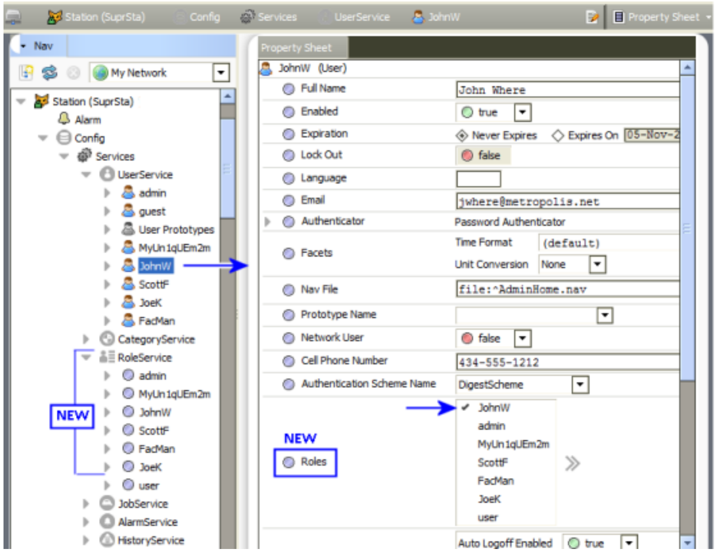
Permissions moved to Roles

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**Rolecreated for eachUser**—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



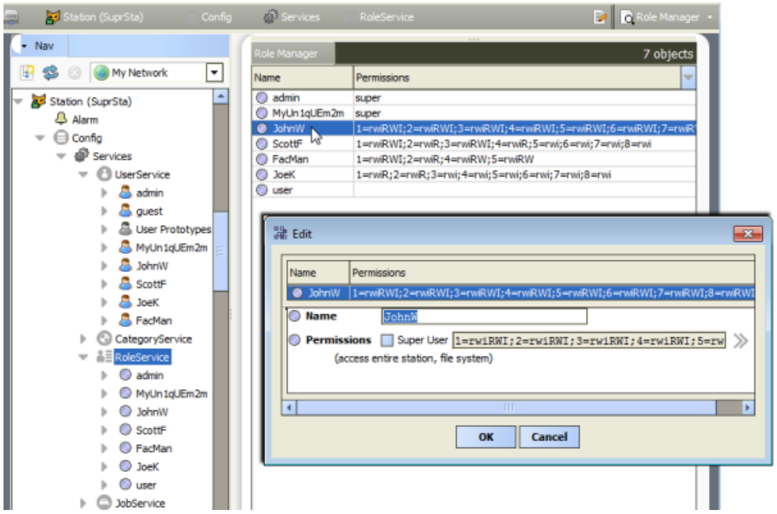
Property Sheet

JohnW (User)

Full Name	John Where
Enabled	true
Expiration	◈ Never Expires ◇ Expires On 05-Nov-2
Lock Out	false
Language	
Email	jwhere@metropolis.net
Authenticator	Password Authenticator
Facets	Time Format (default) Unit Conversion None
Nav File	file:^AdminHome.nav
Prototype Name	
Network User	false
Cell Phone Number	434-555-1212
Authentication Scheme Name	DigestScheme
Roles	✓ JohnW admin MyUn1qUEm2m ScottF FacMan JoeK user
	Auto Logoff Enabled true

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



Role Manager 7 objects

Name	Permissions
admin	super
MyUn1qUEm2m	super
JohnW	1=rwiRWI;2=rwiRWI;3=rwiRWI;4=rwiRWI;5=rwiRWI;6=rwiRWI;7=rwiR
ScottF	1=rwiRWI;2=rwiR;3=rwiRWI;4=rwiR;5=rwi;6=rwi;7=rwi;8=rwi
FacMan	1=rwiRWI;2=rwiR;4=rwiRW;5=rwiRW
JoeK	1=rwiR;2=rwiR;3=rwi;4=rwi;5=rwi;6=rwi;7=rwi;8=rwi
user	

Edit

Name	Permissions
JohnW	1=rwiRWI;2=rwiRWI;3=rwiRWI;4=rwiRWI;5=rwiRWI;6=rwiRWI;7=rwiRWI;8=rwiRWI

Name: JohnW

Permissions: □ Super User (access entire station, file system) 1=rwiRWI;2=rwiRWI;3=rwiRWI;4=rwiRWI;5=rw

As shown above, you can see this in the popup **Edit** dialog 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 done.

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.

For more details on **Roles**, refer to the *Niagara 4 Station Security Guide*.

Authentication is user-specific

When looking at a **User** in a Niagara 4 station, notice that the Password property is now inside a new "**Authenticator**" container, along with "**Password Config**" properties first introduced in AX-3.7. 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 *Niagara 4 Station Security Guide*.

A Migration details

Topics covered in this appendix

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.

Major areas of change for Program objects in Niagara 4 include the following:

- Imports, page 61
- Niagara 4 API changes, page 61
- Primitives, page 61
- Security Manager, page 62
  - File reading/writing, page 62
  - Runtime.exec(), page 62

For a simple example showing a couple of these, see “Example Program object fix” on page 58, page 63.

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 (note that 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.

Niagara 4 API changes

If your Program objects use certain Niagara APIs that have had breaking changes, they will need to have their Program code updated and recompiled using the new Niagara 4 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 4 Developer Documentation* 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.

The places where you will most likely run into this are in the two areas below.

- File reading/writing, page 62
- Runtime.exec(), page 62

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)`. We have created a wrapper called `ProgramRuntime`, which 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 a AX-3.8 station backup .dist file for the demo station is used as source in the N4 migration tool.

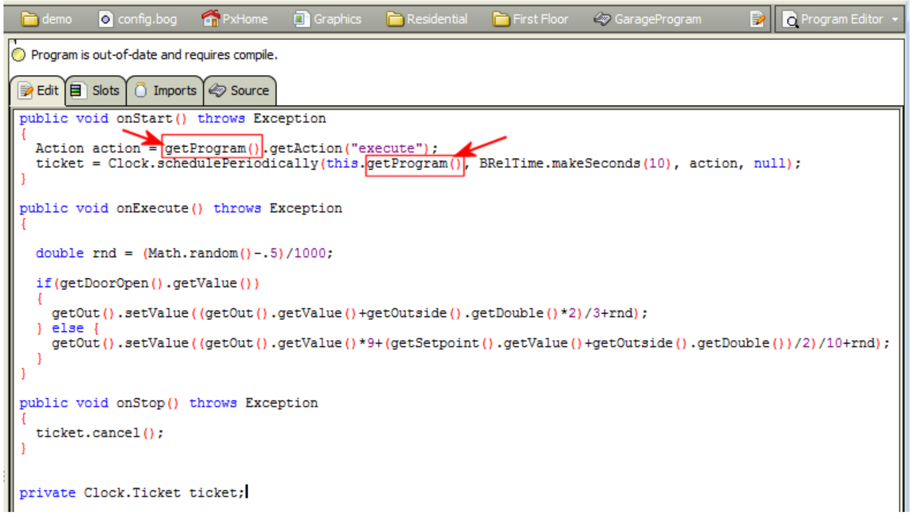
A related snippet from the migration report (log file) looks like below.

```
 WARNING unable to compile Program object PxHome.Graphics.Residential.First Floor.GaragePro
C:\Users\e333988\Niagara4.0\temp\Prog_ea44c5d3b904631ea44c5d3b904631.java:40: error: canno
    Action action = getProgram().getAction("execute");
                    ^
  symbol:   method getProgram()
  location: class Prog_ea44c5d3b904631ea44c5d3b904631
```

This error relates to use of `getProgram()`, as described in "Niagara 4 API changes" on page 57, page 61.

Opening the migrated "demo" station in your N4 Workbench **User Home**, 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



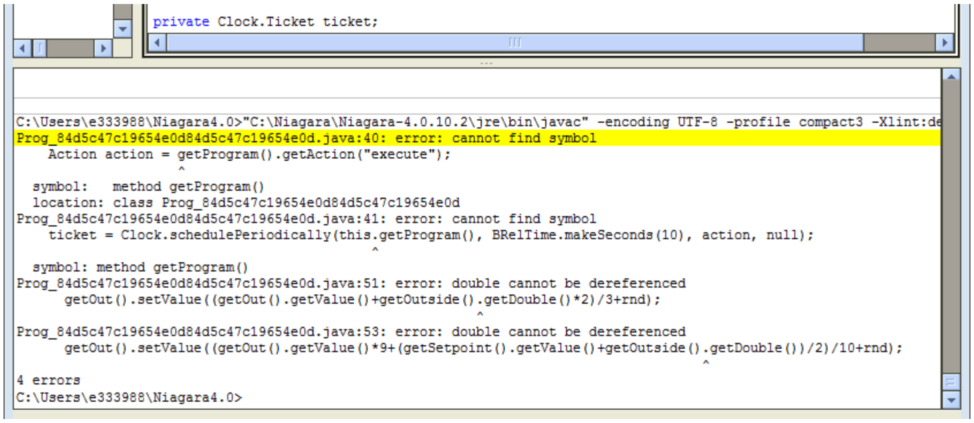
Program is out-of-date and requires compile.

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

public void onExecute() throws Exception
{

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

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

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


private Clock.Ticket ticket;
```

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

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

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

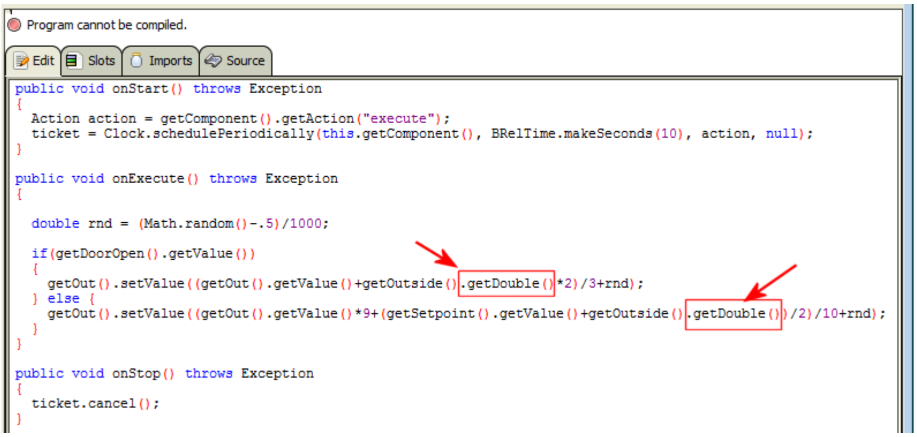


```
private Clock.Ticket ticket;
```

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

After replacing the two `getProgram()` instances to `getComponent()`, saving and then recompiling again, errors are reduced to two—related to use of `getDouble()` for primitives (Figure 24 Figure 28, page 64).

Figure 24 Example Program with remaining issues from primitive syntax errors



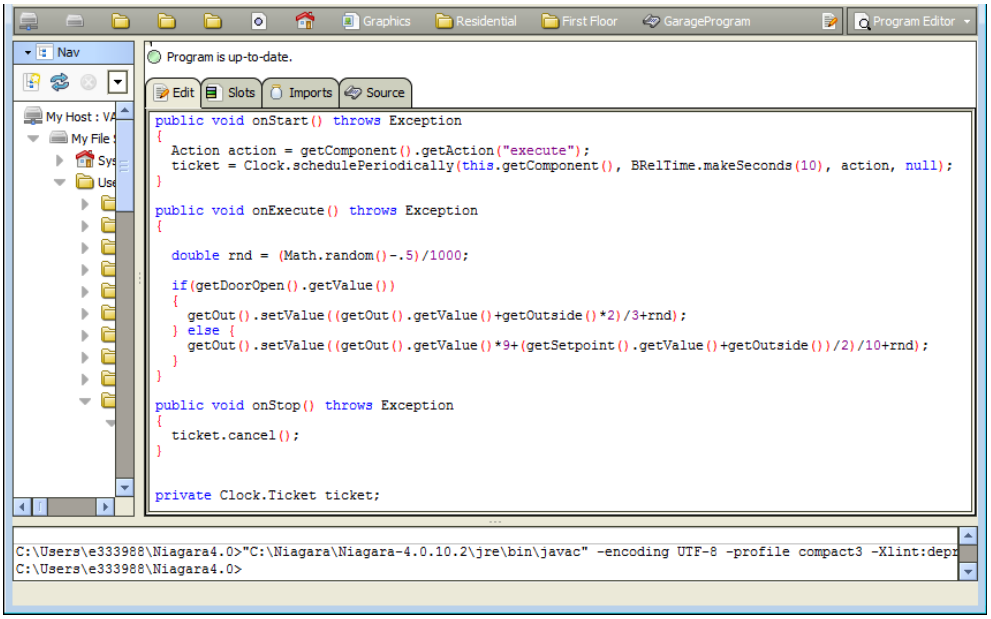
Program cannot be compiled.

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

public void onExecute() throws Exception
{

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

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

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

This is no longer necessary, as explained in the section "Primitives" on page 57, page 61. 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.



Program is up-to-date.

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

public void onExecute() throws Exception
{

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

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

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


private Clock.Ticket ticket;
```

```
C:\Users\e333988\Niagara4.0>"C:\Niagara\Niagara-4.0.10.2\jre\bin\javac" -encoding UTF-8 -profile compact3 -Xlint:depr
C:\Users\e333988\Niagara4.0>
```

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