

# Technical Document

## NiagaraAX Export Tags

*For Supervisor Auto-Configuration*

Niagara<sup>AX-3.7u1</sup>

May 31, 2013



# Niagara<sup>AX</sup> Export Tags

For Supervisor Auto-Configuration

## Confidentiality Notice

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

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

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

## Trademark Notice

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

## Copyright and Patent Notice

This document may be copied by parties who are authorized to distribute Tridium products in connection with distribution of those products, subject to the contracts that authorize such distribution. It may not otherwise, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior written consent from Tridium, Inc.

Copyright © 2013 Tridium, Inc.

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

# CONTENTS

<b>Preface .....</b>	<b>vii</b>
<b>Export tag FAQs .....</b>	<b>vii</b>
<b>What's new in the AX-3.7u1 exportTags module .....</b>	<b>viii</b>
<b>Document Change Log .....</b>	<b>viii</b>
<b>Export Tags Quick Start.....</b>	<b>1-1</b>
<b>Export tag requirements .....</b>	<b>1-1</b>
<b>Add the NiagaraNetwork Export Tag extensions .....</b>	<b>1-2</b>
Add the SupervisorExportTagNetworkExt .....	1-2
<i>To add the SupervisorExportTagNetworkExt .....</i>	<i>1-2</i>
Add the SubordinateExportTagNetworkExt .....	1-2
<i>To add the SubordinateExportTagNetworkExt .....</i>	<i>1-2</i>
Enable the JoinProfile for the Supervisor NiagaraStation .....	1-3
<i>To enable the JoinProfile for a Supervisor NiagaraStation .....</i>	<i>1-3</i>
Verify and edit the station profile.bog file .....	1-4
<i>Verify and edit the Station (.bog) File .....</i>	<i>1-4</i>
<b>Add export tags .....</b>	<b>1-6</b>
<i>Open the exportTags palette and JACE station in Workbench .....</i>	<i>1-6</i>
Adding PxViewTags .....	1-6
<i>To add PxViewTags .....</i>	<i>1-6</i>
Adding PointTags .....	1-7
<i>Adding PointTags .....</i>	<i>1-7</i>
Adding HistoryImportTags .....	1-8
<i>To add HistoryImportTags .....</i>	<i>1-8</i>
Adding SystemHistoryImportTags .....	1-8
<i>To add SystemHistoryImportTags .....</i>	<i>1-8</i>
Adding ScheduleImportTags .....	1-9
<i>Adding ScheduleImportTags .....</i>	<i>1-9</i>
Adding ScheduleExportTags .....	1-10
<i>To add ScheduleExportTags .....</i>	<i>1-10</i>
Adding FileImportTags .....	1-11
<i>To add FileImportTags .....</i>	<i>1-11</i>
Adding ComponentTags .....	1-11
<i>To add ComponentTags .....</i>	<i>1-11</i>
<b>Issue Join commands .....</b>	<b>1-12</b>
Join from the Supervisor .....	1-12
<i>To Join from the Supervisor .....</i>	<i>1-12</i>
Join from the subordinate (JACE) station .....	1-13
<i>To Join from the subordinate (JACE) station .....</i>	<i>1-13</i>
<b>NiagaraAX Export Tag Concepts.....</b>	<b>2-1</b>
<b>Export tags engineering overview .....</b>	<b>2-2</b>
About a Join .....	2-2
Join is a merge .....	2-3
Join and rejoin .....	2-3
Join from either side .....	2-3
Join profile is on subordinate .....	2-3

About the JoinProfile .....	2-3
About the Join process .....	2-4
Join process from subordinate .....	2-5
<b>exportTags palette .....</b>	<b>2-6</b>
<b>About the SupervisorExportTagNetworkExt .....</b>	<b>2-7</b>
<b>About CategoryFilters .....</b>	<b>2-8</b>
About the Category Filter Manager .....	2-8
Adding CategoryFilters .....	2-9
About applied CategoryFilters .....	2-10
<b>About the SubordinateExportTagNetworkExt .....</b>	<b>2-11</b>
Actions on the SubordinateExportTagNetworkExt .....	2-12
About the Join Profile Manager .....	2-12
Editing the Join Profile .....	2-13
Adding default subordinate user credentials to a pre-engineered NiagaraStation .....	2-14
About Station Info .....	2-15
Adding Station Info .....	2-16
Using Station Info .....	2-16
About the Export Tag Summary Manager .....	2-16
Editing export tags in the Summary Manager .....	2-17
Specifying the Supervisor station globally in all export tags .....	2-18
<b>About the niagaraStation profile.bog file .....</b>	<b>2-18</b>
Device extension properties importance in profile.bog file .....	2-19
Provisioning slot additions in a profile.bog file .....	2-20
To add provisioning properties in the station profile.bog file .....	2-20
<b>Station Slot Path considerations .....</b>	<b>2-21</b>
BFormat options in Station Slot Paths .....	2-22
%networkFolderPath% .....	2-22
%deviceFolderPath% .....	2-23
%parent.name% .....	2-24
Conclusions .....	2-24
<b>About PxViewTags .....</b>	<b>2-25</b>
PxViewTag properties .....	2-25
PxViewTag action .....	2-27
About the Supervisor SubstitutePxView .....	2-27
About ord substitution overrides .....	2-28
Ords that do not virtualize .....	2-29
<b>About PointTags .....</b>	<b>2-30</b>
PointTag properties .....	2-30
<b>About HistoryImportTags .....</b>	<b>2-31</b>
HistoryImportTag properties .....	2-31
<b>About SystemHistoryImportTags .....</b>	<b>2-32</b>
SystemHistoryImportTag properties .....	2-32
<b>About ScheduleImportTags .....</b>	<b>2-34</b>
ScheduleImportTag properties .....	2-34
<b>About ScheduleExportTags .....</b>	<b>2-34</b>
ScheduleExportTag properties .....	2-35
<b>About FileImportTags .....</b>	<b>2-36</b>
FileImportTag properties .....	2-36
<b>About ComponentTags .....</b>	<b>2-38</b>
ComponentTag properties .....	2-38
ComponentTag example (ExportTagProgram) .....	2-39

## **Export Tag Plugin Guides .....** **3-1**

<b>Export Tag Plugin Guides Summary .....</b>	<b>3-1</b>
<i>exportTags-CategoryFilterManager</i> .....	3-1

*exportTags-JoinProfileManager* ..... 3-1  
*exportTags-ExportTagSummaryManager* ..... 3-2

## **Export Tag Component Guides ..... 4-1**

### **ExportTag Component Reference Summary ..... 4-1**

*exportTags-CategoryFilter* ..... 4-1  
*exportTags-CategoryFilterExt* ..... 4-1  
*exportTags-ComponentTag* ..... 4-1  
*exportTags-FileImportTag* ..... 4-1  
*exportTags-HistoryImportTag* ..... 4-1  
*exportTags-JoinProfile* ..... 4-2  
*exportTags-PointTag* ..... 4-2  
*exportTags-PxViewTag* ..... 4-2  
*exportTags-ScheduleExportTag* ..... 4-2  
*exportTags-ScheduleImportTag* ..... 4-2  
*exportTags-StationInformation* ..... 4-2  
*exportTags-SubordinateExportTagNetworkExt* ..... 4-2  
*exportTags-SupervisorExportTagNetworkExt* ..... 4-3  
*exportTags-SystemHistoryImportTag* ..... 4-3



# PREFACE

## Preface

---

- [Export tag FAQs](#)
- [What's new in the AX-3.7u1 exportTags module](#)
- [Document Change Log](#)

### Export tag FAQs

The following are frequently asked questions (FAQs) about NiagaraAX Export Tags:

**Q: *What type of jobs benefit the most by using export tags?***

A: The biggest benefit from export tag usage is on jobs where JACE stations under a Supervisor are largely “replicated”, often with many such subordinate stations. Such jobs may have scores of JACEs, or even hundreds or more—and possibly even multiple “tiered” subordinate Supervisor stations. Further, export tags allow an operating Supervisor to be dynamically expanded with additional subordinates (JACEs, typically), without doing the work on the Supervisor station itself.

Export tag usage is not limited to such large jobs, however. One particular type of export tag may be useful even on jobs where each subordinate (JACE) station is thoroughly unique. Specifically, the PxViewTag allows PxViews served from a subordinate station to be also dynamically served from the Supervisor, with very little engineering effort.

**Q: *Where do export tags go?***

A: Export tags are added in subordinate (typically JACE) stations. The Supervisor uses only *one* component from the exportTags palette: the SupervisorExportTagNetworkExt component.

**Q: *Ok, so where do I drop export tags in the JACE station?***

A: That depends upon the export tag type—for the most “advanced” export tag, the PxViewTag, you drop it onto the same folder (or container) that has the Px view. You drop other export tag types at different spots, for example, drop a HistoryImportTag onto a point’s history extension, or drop a PointTag onto a proxy point under some driver network.

**Q: *What do export tags do?***

A: Essentially, export tags determine which station objects (components, histories, files) should be represented in a particular Supervisor, under its NiagaraStation component for that subordinate station. Included is any folder structure needed under that NiagaraStation, which you can model first in the subordinate station in its “profile.bog file”. In that same file, you also specify any “non-default” property values needed in that NiagaraStation, including properties in its “device extension” components. Then, when the export tag “Join” command is given, those objects (along with their folder structure), are implemented in the Supervisor station.

While engineering, and particularly while learning about export tags, the Join process is often an iterative process. In other words, you typically issue *many* export tag Joins before a NiagaraStation is considered “done”. Each Join dynamically modifies the NiagaraStation in the Supervisor, as needed.

Note that each Join is “merged” with the existing NiagaraStation representation in the Supervisor station. This includes Niagara components that were created in “the traditional way” as well as those made from use of export tags. Therefore, it is possible to have a “hybrid-sourced” representation of a subordinate station in a Supervisor’s NiagaraNetwork, where some items were added by export tags in the subordinate station, and other items were added by engineering in the Supervisor station.

**Q: *Can a JACE build its NiagaraNetwork from export tags in other JACEs?***

- A: This is not recommended, because of the extra memory consumed by the “Supervisor” JACE. Also, because JACEs are not typically licensed for Niagara “virtual” components, they could not take advantage of “PxViewTags”.

## What's new in the AX-3.7u1 exportTags module

The AX-3.7u1 release contains security enhancements which affect how a subordinate station 'Joins' with the Supervisor station. Specifically, a new slot has been added to the JoinProfile, the “Default Subordinate User Password” slot, which is used by the Supervisor during an initial Join with a Subordinate station. After upgrading, failure to update this slot's password information for the selected user will cause authentication errors if a Join is issued from the Subordinate station and that station does not already exist in the Supervisor's NiagaraNetwork (the Supervisor is not “aware” of this Subordinate).

## Document Change Log

Updates (changes/additions) to this *NiagaraAX Export Tags* document are listed below.

- Updated: May 31, 2013  
Document updates to cover exportTag changes starting in the “update 1” release of AX-3.7, denoted in this document as “AX-3.7u1”. The following changes were made:
  - Added the preceding section on “Whats new in the AX-3.7u1 exportTags module”.
  - Added notes explaining the need for Default Subordinate User/Password information in several places throughout the document, including information on required permissions.
  - Added notes clarifying ComponentTag usage in several places throughout the document.
  - Added the new JoinProfile property to the last paragraph under “[Join process from subordinate](#)” on page 2-5, a new paragraph and procedure for “[Adding default subordinate user credentials to a pre-engineered NiagaraStation](#)” on page 2-14, and added the new property under the “[exportTags-JoinProfileManager](#)” on page 3-1.
- Updated: August 26, 2012  
Document is now a “revision-split” document that applies only to a AX-3.7 and later Supervisor and JACE controllers, with most references to using export tags in AX-3.5 and AX-3.6 *removed*. Screen captures were updated to show the default AX-3.7 “Lucid” theme in Workbench. Other areas of document change include the following:
  - Throughout the document changes were made to describe (or “Note”) the configuration possibility of hosts installed using certificate-based SSL connectivity, available starting in AX-3.7. This includes Fox SSL (foxs) for station-to-station communications, used in export tag “Joins”. Related changed sections include the “[Export Tags Quick Start](#)” subsections “[Enable the JoinProfile for the Supervisor NiagaraStation](#)” on page 1-3 and related new properties described in the section “[Editing the Join Profile](#)” on page 2-13.
  - This document revision now emphasizes the editing of the station “profile.bog” file in a JACE's file space (formerly referred to as a “station bog” file), which determines *more* that just the folder structure under the NiagaraStation (that represents the JACE) in the Supervisor station. *Properties* in the profile.bog's “niagaraStation”, and in its device extensions, are important too. Related document changes are in the “[Export Tags Quick Start](#)” subsections “[Verify and edit the station profile.bog file](#)” on page 1-4, with more details in “[About the niagaraStation profile.bog file](#)” on page 2-18, including new subsections “[Device extension properties importance in profile.bog file](#)” on page 2-19 and “[Provisioning slot additions in a profile.bog file](#)” on page 2-20.
  - The reworked “short slot name” method used in Niagara “virtual components” (starting in AX-3.7) was mentioned in descriptions of results from using PxViewTags, which automatically create “Niagara virtuals” on the Supervisor station, as needed. Changed document sections include “[About PxViewTags](#)” on page 2-25 and subsections “[About the Supervisor SubstitutePxView](#)” on page 2-27 and “[About ord substitution overrides](#)” on page 2-28.
- Updated: April 5, 2011  
Document updates to cover export tag-related changes made in NiagaraAX-3.6 (AX-3.6). Many sections in the document changed, mostly related to the AX-3.6 and later enhancement to the “Station Slot Path” property in several types of export tags. The main areas of change in this document include new sections “[Station Slot Path considerations](#)” on page 2-21 and “[Actions on the SubordinateExportTagNetworkExt](#)” on page 2-12. Various procedures in the “[Add export tags](#)” sections were also modified to reflect differences when working in a AX-3.6 or later system vs. a AX-3.5 system.
- Updated: February 3, 2010  
Minor corrections from a late SME review. One answer in the [Export tag FAQs](#) was modified—export tags are currently not a licensed feature; however, use as a “Supervisor” in a JACE station is not



recommended. Other changes were made in the “About the Join process” section, in diagrams for a “Supervisor Join” (Figure 2-2 on page 4) and a “Subordinate Join” (Figure 2-3 on page 5).

- Published: January 29, 2010  
Initial document.



# CHAPTER 1

## Export Tags Quick Start

---

This section lists export tag requirements and provides procedures to use export tags in a system with one or more subordinate JACE stations and a Supervisor. The following sections are included:

- “Export tag requirements” on page 1-1
- “Add the NiagaraNetwork Export Tag extensions” on page 1-2
  - “Add the SupervisorExportTagNetworkExt” on page 1-2
  - “Add the SubordinateExportTagNetworkExt” on page 1-2
  - “Enable the JoinProfile for the Supervisor NiagaraStation” on page 1-3
  - “Verify and edit the station profile.bog file” on page 1-4
- “Add export tags” on page 1-6
  - “Adding PxViewTags” on page 1-6
  - “Adding PointTags” on page 1-7
  - “Adding HistoryImportTags” on page 1-8
  - “Adding SystemHistoryImportTags” on page 1-8
  - “Adding ScheduleImportTags” on page 1-9
  - “Adding ScheduleExportTags” on page 1-10
  - “Adding FileImportTags” on page 1-11
  - “Adding ComponentTags” on page 1-11
- “Issue Join commands” on page 1-12
  - “Join from the Supervisor” on page 1-12
  - “Join from the subordinate (JACE) station” on page 1-13

### Export tag requirements

To use the export tag feature (as described in this document revision), you must have a Supervisor running NiagaraAX-3.7 or later. In addition, any subordinate station you wish to “tag up” with export tags should also be running AX-3.7 or later, where that host is licensed for enterprise connectivity (its license should have the “niagaraDriver” feature with a non-zero device limit).

**Note:** *On each JACE, make sure the **exportTags** module is installed (if not already present, or upgrade if an older revision). The **exportTags** module is required to support “tagging up” the JACE station.*

For more details about software installation, see “About the Commissioning Wizard” in the *JACE NiagaraAX Install and Startup Guide*.

Following this, the subordinate station is now ready for export tag configuration, as described in the rest of this document.

**Note:** *Before starting, it is recommended you read “NiagaraAX Export Tag Concepts” on page 2-1 for an overview. That section is followed by more detailed conceptual and reference information.*

Basic procedures for immediately using export tags are in the next sections.

- [Add the NiagaraNetwork Export Tag extensions](#)
- [Add export tags](#)
- [Issue Join commands](#)

## Add the NiagaraNetwork Export Tag extensions

To enable stations to utilize export tags, perform the following tasks:

- [Add the SupervisorExportTagNetworkExt](#) (do this once, in the Supervisor station only)  
Then for each JACE station:
- [Add the SubordinateExportTagNetworkExt](#)
  - [Enable the JoinProfile for the Supervisor NiagaraStation](#)
  - [Verify and edit the station profile.bog file](#)

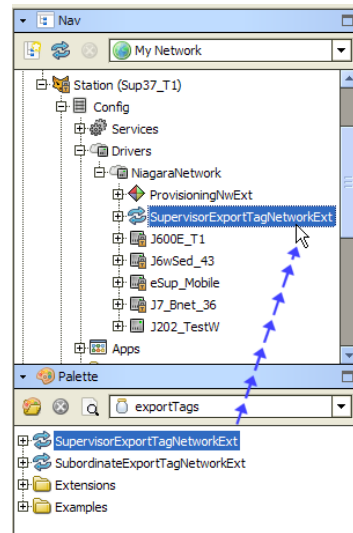
### Add the SupervisorExportTagNetworkExt

Do this once, in the Supervisor station's NiagaraNetwork only.

#### To add the SupervisorExportTagNetworkExt

- Step 1 In Workbench, open the **exportTags** palette in your side bar (see “Using the palette side bar” in the *User Guide* for general details).
- Step 2 Open the Supervisor station. Expand the Config **Drivers/NiagaraNetwork** container. If a child **SupervisorExportTagNetworkExt** is already there, go to the next procedure.
- Step 3 From the **exportTag** palette, *drag* (or copy and paste) a **SupervisorExportTagNetworkExt** into the **NiagaraNetwork**. In the popup **Name** dialog, you can rename it—or, simply use the default name.

**Figure 1-1** Copy SupervisorExportTagNetworkExt into NiagaraNetwork on the Supervisor station



The extension is now under the NiagaraNetwork, and includes child nodes “Category Filters” and “Alarm Source Info”. You can leave properties in them at default values for now. For more details, see [“About the SupervisorExportTagNetworkExt”](#) on page 2-7 and [“About CategoryFilters”](#) on page 2-8.

### Add the SubordinateExportTagNetworkExt

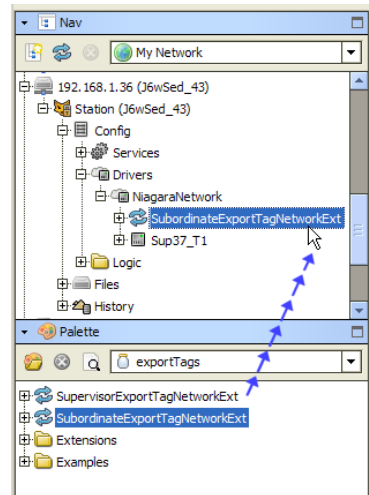
**Note:** Before starting station engineering, ensure all JACEs have the exportTags module installed. See [“Export tag requirements”](#) on page 1-1.

Do this using a station connection to each JACE station that is subordinate to the Supervisor.

#### To add the SubordinateExportTagNetworkExt

- Step 1 In Workbench, open the subordinate (JACE) station.
- Step 2 Open the **exportTags** palette in your Workbench palette side bar, if not already open.
- Step 3 Expand the subordinate station's **Drivers/NiagaraNetwork** container. If a child **SubordinateExportTagNetworkExt** is already there, go to the next procedure.
- Step 4 From the **exportTags** palette, *drag* (or copy and paste) a **SubordinateExportTagNetworkExt** onto the **NiagaraNetwork**. In the popup **Name** dialog, you can rename it—or, use the default name.

**Figure 1-2** Copy SubordinateExportTagNetworkExt into NiagaraNetwork on each JACE station



The extension is under the NiagaraNetwork, and includes two special views. By addition, it automatically creates a “JoinProfile” child under each NiagaraStation component in the subordinate’s NiagaraNetwork.

In the subordinate’s NiagaraNetwork, only the *NiagaraStation that represents the Supervisor* is of interest when working with export tags. You must enable the JoinProfile under this NiagaraStation, and typically configure other properties of that JoinProfile, in order to use export tags. See the next section, “[Enable the JoinProfile for the Supervisor NiagaraStation](#)”.

**Note:** A *NiagaraStation for the Supervisor* should already be in the subordinate’s NiagaraNetwork, providing the Supervisor station had a *NiagaraStation added (and correctly configured)* to connect to the subordinate. In case this *NiagaraStation* shows a status of disabled:




- Expand its Client Connection child and make sure that the (Fox) Port, Username, and Password are correct for station connection back to the Supervisor.
- Enable the parent NiagaraStation (from its property sheet), and issue a right-click **Ping** action. Verify that its status is ok.

*In case the JACE is not currently installed on the LAN with the Supervisor, you can still work with export tags, but you will not be able to “Join”, much less ping, the Supervisor. However, in this case you may wish to configure the “Default” connection properties under the JoinProfile of the Supervisor NiagaraStation in the subordinate. For related details, see “[Editing the Join Profile](#)” on page 2-13.*

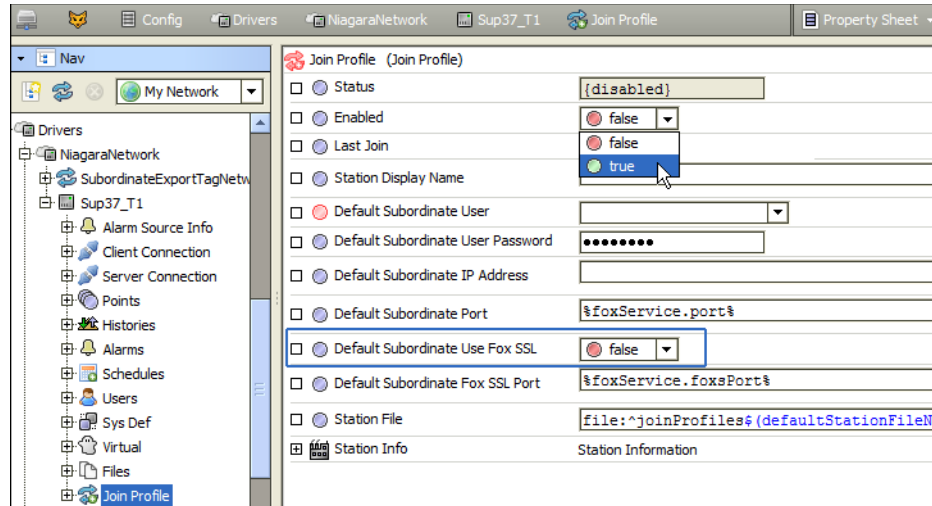
### **Enable the JoinProfile for the Supervisor NiagaraStation**


Do this for each JACE station that is subordinate to the Supervisor.

#### **To enable the JoinProfile for a Supervisor NiagaraStation**

- Step 1 In the subordinate (JACE) station, expand its **NiagaraNetwork**, then the NiagaraStation  or  that represents the Supervisor.
- Step 2 Double-click its  **JoinProfile** component to see its property sheet.
- Step 3 Set its Enabled property to true ([Figure 1-3](#)), and click **Save**.

**Figure 1-3** Enable JoinProfile under NiagaraStation that represents Supervisor



Step 4 Providing the JACE is configured to use SSL for station connections, that is *Foxs* (Fox SSL), an option available starting in AX-3.7, change the “Default Subordinate Use Fox SSL” property from the default `false` to `true`, and click **save**. This property is shown encircled above, but has not been changed yet. Note that *Foxs* is not supported in JACE-2 or -4/5 series hosts (JACEs that use the IBM J9 JVM), so if working in one of these stations, this property must be left `false`. Typically, if the JACE is configured for SSL, you have it opened now in Workbench using a “Station SSL Connection” (*Foxs*, showing icon ).

**Note:** Complete details on SSL in AX-3.7 and later hosts are in the NiagaraAX SSL Connectivity Guide. Edits to other properties in this JoinProfile are often optional. In general, usage often applies to very large systems. For more details on all JoinProfile properties, see “Editing the Join Profile” on page 2-13.

**Note:** Starting in AX-3.7u1, when issuing a Join from a subordinate JACE station, and that station does not already exist in the Supervisor, the “Default Subordinate User Password” property must be configured with a valid password for the selected “Default Subordinate User”. If no valid user and password has been entered the Join will fail due to authentication errors.

After enabling the JoinProfile, the next task to do in this station is “Verify and edit the station profile.bog file”.

### Verify and edit the station profile.bog file

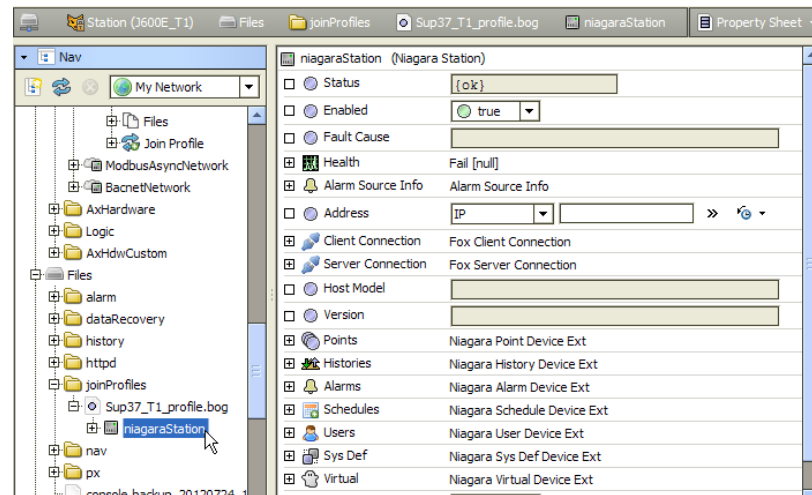
Do this for each JACE station that is subordinate to the Supervisor.

As a result of enabling the `JoinProfile`, the JACE station should have a *new folder* under its *file space*. Using defaults, this file folder is `joinProfiles`. Inside is a *niagaraStation* “profile.bog file” that defines the station “template” for it, which upon a “Join” gets built *on the Supervisor’s station*, under its *NiagaraNetwork* to model this JACE

### Verify and edit the Station (.bog) File

Step 1 Verify that a “profile.bog file” is now under the JACE station’s file space.  
See [Figure 1-4](#) for an example of opening this profile.bog file in Workbench.

**Figure 1-4** Station bog is !Files/joinProfile/supervisorStationName\_profile.bog



Note the profile.bog file specifies the parent “container/folder structure”, and is not meant for adding individual components. However, *properties* of this niagaraStation and its “device components” are used.

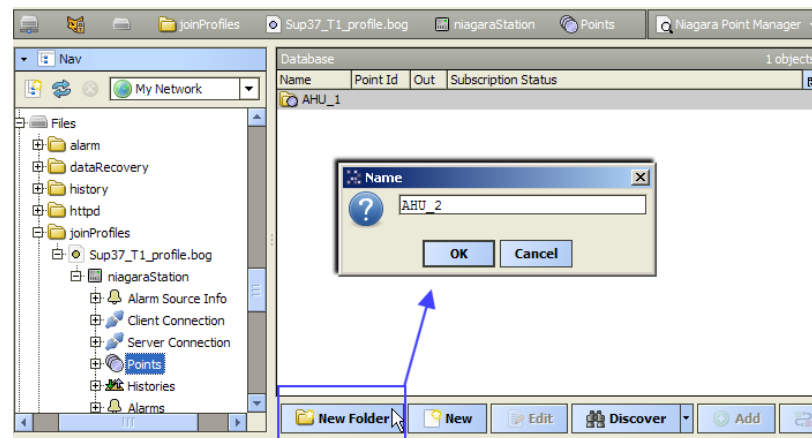
- Step 2 As needed, edit properties of the NiagaraStation (representing this JACE) as it needs to be configured on the Supervisor station, including properties of its “device extensions” such as **Alarms** and **Users**. Note that upon any export tag “Join”, these property values *overwrite* any property values in the corresponding NiagaraStation component in the Supervisor station.

To do this, expand the **niagaraStation** node under the *supervisorStationName\_profile.bog* file and access property sheet views as if working with a “live” station.

For example, if “network users” (copied from the Supervisor) are, or will be, used in this JACE station, under its **Users** device extension, set the “Sync Out Enabled” property to `true`. For related details, see “About the niagaraStation profile.bog file” on page 2-18 and “Device extension properties importance in profile.bog file” on page 2-19.

- Step 3 Optionally, you can also edit this file to add new folders, as shown being done in Figure 1-5.

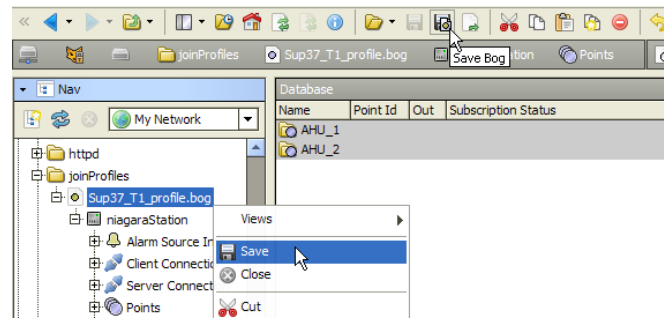
**Figure 1-5** Example of adding Point folders in profile.bog file



Later, when you are “tagging up” components in this station, note that each export tag has a “Station Slot Path” property in which you specify the parent folder. You can either type a slot path in directly (so that the folder will be made “dynamically” on the Supervisor), or else click a “folder” link. This folder link opens a dialog to this niagaraStation in the JACE’s “profile.bog” file—for parent selection only. You can come back and edit this profile.bog file whenever convenient, including between subsequent “Joins”.

**Note:** Remember to save the profile.bog file each time when finished making changes. Right-click on the *supervisorStationName\_profile.bog* file and select **Save**, as shown in Figure 1-6.

**Figure 1-6** Save profile.bog file each time when done with changes



## Add export tags

To “tag up” your subordinate (JACE) stations, add and configure export tags, as needed.

### Open the exportTags palette and JACE station in Workbench

- Step 1 Open the **exportTags** palette in your Workbench palette side bar, if not already open. Expand its **Extensions** folder to see the different types of export tags.
- Step 2 Open a JACE station in Workbench.
- Step 3 In the Nav tree, expand the station’s component space (Config) as needed to drag and drop export tags from the palette’s **Extensions** folder onto various components in the station.  
The following sections include “quick start” procedures to get started.


- [Adding PxViewTags](#)
- [Adding PointTags](#)
- [Adding HistoryImportTags](#)
- [Adding SystemHistoryImportTags](#)
- [Adding ScheduleImportTags](#)
- [Adding ScheduleExportTags](#)
- [Adding FileImportTags](#)
- [Adding ComponentTags](#)

### Adding PxViewTags

If your JACE station serves PxViews directly, and you wish to add any number of these PxViews at the Supervisor station, you can do this easily using PxViewTags.

**Note:** Among the different types of export tags, PxViewTags offer the biggest return on engineering time when considering the functionality provided. In some cases, they may preclude most use of PointTags, except where “link control” in the Supervisor station requires Niagara proxy points.

### To add PxViewTags





- Step 1 Drag the PxViewTag from the exportTags palette onto the folder or container with the original PxView. The **Name** dialog appears, in which you can accept “PxViewTag”, or enter another name, and click **OK**. Note the resulting PxView on the Supervisor uses the name of the referenced PxView, and not this name.
- Step 2 Double-click the PxViewTag  to access its property sheet.
  1. In the **Supervisor Station** property, click the  control and select the Supervisor station, then click **Save**.
  2. For **Station Slot Path**, do either of the following:

- Enter a desired slot path to use on the Supervisor station for the folder to have this PxView. For example, if you enter `slot:Ground_Floor/AHU2`  
Upon a Join, at the Supervisor station, this Px view will be put on a folder at this location:

```
NiagaraNetwork
├── JaceStationName
│   ├── Ground_Floor (folder)
│   │   └── AHU2 (folder with SubstitutePxV
```

Note “BFormat” variables are an alternative, and can save engineering time. See “[Station Slot Path considerations](#)” on page 2-21 and “[BFormat options in Station Slot Paths](#)” on page 2-22. Or, (if desired) use the station “profile.bog” file method”, described next.






- Click  open folder, for the **Select Parent Target** dialog. Expand the **niagaraStation** and select a target folder for the PxView to appear in the Supervisor station (default puts the PxView on the root of the NiagaraStation).  
*Note:* You may edit the “profile.bog file” to specify a new target folder structure—see “Verify and edit the station profile.bog file” on page 1-4 for details.
3. In the **Px View** property, click the  control and select the PxView for export, by name.
- Step 3 Remaining properties of the PxViewTag are often left at defaults, noting briefly:
- **Required Permissions** — is Operator read (r) by default, click the  control to change.
  - **Order Number** — is 0 default. If assigning multiple PxViewTags to the same target station slot path, assign a different order number to each one, to specify menu order of the views. The menu displays the views “highest number first”, that is, order number 2 above number 1 above number 0.
  - **File** related properties (Create File Import, and File Import Name, Overwrite Policy, Execution Time, Execution Time, Use Category Mask) relate to FileImportTags that may be created automatically in support of the PxView export.
  - **Supervisor Px File** Directory (and Media Directory) properties specify the Supervisor folder locations for associated imported px and image files, respectively. Default values use “ord variables” to duplicate the same folder organization on the JACE.
  - **Substitute Override Ords** — provides a popup  **View** dialog in which you can edit “substitute ords” used in the PxView to be created on the Supervisor, using the “Choose from Px” feature. In many scenarios, this may be useful to change hyperlink destinations in Px bindings.
- Step 4 Click  **Save** after finishing all property changes.  
For further details, see “About PxViewTags” on page 2-25, including subsection “About ord substitution overrides” on page 2-28.


## Adding PointTags

Point tags create Niagara proxy points on the Supervisor, under the **Points** container of the NiagaraStation that represents this subordinate (JACE) station. Typically, you may limit usage to points you need for “output links” or control logic in the Supervisor, because of the “Niagara virtual point” feature automatically provided in PxViewTags. However, if using the scheme where all PxView tags use “slot:points/%networkFolderPath” as “Station Slot Path”, it is recommended to add a PointTag to at least one proxy point under each device with a PxView. For related details, see “BFormat options in Station Slot Paths” on page 2-22.

## Adding PointTags

- Step 1 Drag the **PointTag** from the exportTags palette onto the proxy point or other target component. The **Name** dialog appears, in which you can accept “PointTag”, or enter another name, and click **OK**. Note the resulting Niagara proxy point uses the name of the referenced component, and not this name.
- Step 2 Double-click the PointTag  to access its property sheet.
1. In the **Supervisor Station** property, click the  control and select the Supervisor station, then click  **Save**.
  2. For **Station Slot Path**, do either of the following:
    - Enter a desired slot path to use on the Supervisor station for the Niagara points folder to hold this proxy point. Note the entered property value *must begin* with `slot:points`  
For example, if you enter `slot:points/Floor_4/AHU5`  
Upon a Join, at the Supervisor station, the resulting proxy point will be in this location:  




```
NiagaraNetwork
├─JaceStationName
│   └─Points (existing Points device extension)
│       └─Floor_4 (points folder)
│           └─AHU5 (points folder that contains this proxy point)
```
    - Note “BFormat” variables are an alternative, and can save engineering time. See “Station Slot Path considerations” on page 2-21 and “BFormat options in Station Slot Paths” on page 2-22. Or, (if desired) use the station “profile.bog” file method”, described next.
    - Or, click  open folder, for the **Select Parent Target** dialog. Expand the **niagaraStation** and select a target container for the Niagara proxy point to appear in the Supervisor station (default (`slot:points`) puts the proxy point in the root of the **Points** extension).
- Note:* You may edit the “profile.bog file” to specify a new target folder structure—see “Verify and edit the station profile.bog file” on page 1-4 for details.


- Step 3 Remaining properties of the PointTag are often left at defaults, noting briefly:
- **Conversion** — standard Conversion property for a Niagara proxy point, typically left at Default.
  - **Tuning Policy Name** — standard Tuning Policy Name property for a Niagara proxy point. For related details, see “Niagara Tuning Policy notes” in the *Drivers Guide*.
- Step 4 Click  **Save** after finishing all property changes.  
For further details, see “About PointTags” on page 2-30.

### Adding HistoryImportTags

Add a HistoryImportTag to the history extension of a point or component to import/archive that history on the Supervisor. This tag creates a Niagara history import descriptor in the Supervisor, under the **Histories** container of the NiagaraStation that represents this subordinate (JACE) station.

#### To add HistoryImportTags

- Step 1 Drag the **HistoryImportTag** from the **exportTags** palette onto the point’s history extension, or if desired, onto its child **HistoryConfig** extension (either location is acceptable).  
The **Name** dialog appears. Either accept “HistoryImportTag”, or enter another name and click **OK**. Note by default, the resulting Niagara history import descriptor uses the parent name of the history extension, and not this name.
- Step 2 Double-click the HistoryImportTag  to access its property sheet.
1. In the **Supervisor Station** property, click the  control and select the Supervisor station, then click  **Save**.
  2. For **Station Slot Path**, either accept the default “slot:histories”, or else either:
    - Enter a slot path to use on the Supervisor station for the Niagara *archives folder* to hold this history import descriptor. Note the entered property value *must begin* with `slot:histories`. Note “BFormat” variables are an alternative too. See “Station Slot Path considerations” on page 2-21 for more details. Or, (if desired) use the station “profile.bog” file method”, described next.
    - Or, click  open folder, for the **Select Parent Target** dialog. Expand the **niagaras-tation** and select a target container under **Histories** for the Niagara history import descriptor to appear in the Supervisor station (default “slot:histories” puts the import descriptor in the root of the **Histories** extension).

*Note:* You may edit the “profile.bog file” to specify a new target folder structure—see “Verify and edit the station profile.bog file” on page 1-4 for details.
- Step 3 Most remaining properties of the HistoryImportTag are standard to Niagara history import descriptors, noting briefly:
- **History Import Name** — name of the NiagaraHistoryImport descriptor, by default set using a format variable, resulting in name similar to: “Remote\_pointName”. Does not affect history name.
  - **Execution Time** — standard selections for a Niagara history import.
  - **Config Overrides** — standard selections for a Niagara history import, containing settings for Capacity and Full Policy.
  - **On Demand Poll** (Enabled and Frequency) — standard selections for a Niagara history import. For related details on these four properties, see “History Import Edit” in the *Drivers Guide*.
  - **Use Category Mask** — Boolean to specify whether the history imported by this tag will have any export tag *category filters* applied to it (default is true). For related details, see “Adding CategoryFilters” on page 2-9.
- Step 4 Click  **Save** after finishing all property changes.  
For further details, see “About HistoryImportTags” on page 2-31.




### Adding SystemHistoryImportTags


Add one or more SystemHistoryImportTags anywhere in the subordinate (JACE) station to import histories on the Supervisor using the “System Tags” feature. Each tag creates a Niagara system history import descriptor in the Supervisor, under the **Histories** container of the NiagaraStation that represents this subordinate (JACE) station.

#### To add SystemHistoryImportTags

- Step 1 Drag the **SystemHistoryImportTag** from the **exportTags** palette anywhere in the station’s Config space (you may wish to create a separate folder just to hold these tags).

The **Name** dialog appears. Either accept “SystemHistoryImportTag”, or enter another name and click **OK**. By default, this is the name of the NiagaraSystemHistoryImport descriptor created in the Supervisor.

- Step 2 Double-click the SystemHistoryImportTag  to access its property sheet.
1. In the **Supervisor Station** property, click the  control and select the Supervisor station, then click  **Save**.
  2. For **Station Slot Path**, either accept the default “slot:histories”, or else either:
    - Enter a slot path to use on the Supervisor station for the Niagara *archives folder* to hold this history import descriptor. Note the entered property value *must begin* with slot:histories. Note “BFormat” variables are an alternative too. See “[Station Slot Path considerations](#)” on page 2-21 for more details. Or, (if desired) use the station “profile.bog” file method, described next.
    - Or, click  open folder, for the **Select Parent Target** dialog. Expand the **niagaraStation** and select a target container under **Histories** for the Niagara history import descriptor to appear in the Supervisor station (default “slot:histories” puts the import descriptor in the root of the **Histories** extension).



*Note:* You may edit the “profile.bog file” to specify a new target folder structure—see “[Verify and edit the station profile.bog file](#)” on page 1-4 for details.
- Step 3 In the **System Tags Pattern** property (last property listed in property sheet), enter one or more System Tags, as used in history extensions in the subordinate (JACE) station. For related details, see the section “Using System Tags to import Niagara histories” in the *Drivers Guide*.
- Step 4 Most remaining properties of the SystemHistoryImportTag are standard to Niagara system history import descriptors, noting briefly:
- **History Import Name** — name of the NiagaraSystemHistoryImport descriptor, where the default format variable results in the same name as the export tag itself.
  - **Execution Time** — standard selections for a Niagara history import.
  - **Config Overrides** — standard selections for a Niagara history import, containing settings for Capacity and Full Policy.
  - **On Demand Poll** (Enabled and Frequency) — standard selections for a Niagara history import. For related details on these four properties, see “History Import Edit” in the *Drivers Guide*.
  - **Use Category Mask** — Boolean to specify whether histories imported by this tag will have any export tag *category filters* applied to them (default is true).
- Note:* For this export tag type, this feature is future use only—currently, only histories exported using the HistoryImportTag can make use of the category mask feature.
- Step 5 Click  **Save** after finishing all property changes.
- For further details, see “[About SystemHistoryImportTags](#)” on page 2-32.

## Adding ScheduleImportTags

Although usage is expected to be infrequent, you can add a ScheduleImportTag onto an existing schedule component (BooleanSchedule, NumericSchedule, etc.) in the subordinate (JACE) station. Upon a Join, this creates a corresponding “slave” schedule in the Supervisor (and a Niagara ScheduleImportExt descriptor), as well as a Niagara ScheduleExportExt descriptor in this subordinate (JACE) station.

### Adding ScheduleImportTags

---

- Step 1 Drag the **ScheduleImportTag** from the **exportTags** palette onto the schedule component. The **Name** dialog appears. Either accept “ScheduleImportTag”, or enter another name and click **OK**. By default, the resulting Niagara slave schedule (and ScheduleImportExt) in the Supervisor use the parent name of the schedule component. By default, the ScheduleExportExt is named *ScheduleNameExport*.
- Step 2 Double-click the ScheduleImportTag  to access its property sheet.
- Step 3 In the **Supervisor Station** property, click the  control and select the Supervisor station. Note that Station Slot Path is read-only at “slot:schedules” (**Schedules** under the NiagaraStation).
- Step 4 If needed, change the **Execution Time** property settings.
- Step 5 Click  **Save** after finishing all property changes.
- For further details, see “[About ScheduleImportTags](#)” on page 2-34.












## Adding ScheduleExportTags

Among the different export tag types, ScheduleExportTags may be the least “intuitive”. To use one, you must have an existing Niagara schedule with ScheduleImportExt descriptor in the subordinate (JACE) station, under the NiagaraStation that represents the Supervisor. Drag the ScheduleExportTag onto the ScheduleImportExt component, that is, making it a child of the Niagara schedule import descriptor.

If you want to “slave” to an existing schedule in the Supervisor, you need to know its ord in that station (component path)—in which case, use of this tag is somewhat redundant.

Or, you can use a ScheduleExportTag with a *new* schedule to create a new, blank, “master” schedule on the Supervisor that corresponds to the “slave” schedule on the subordinate (JACE) station.

### To add ScheduleExportTags




- Step 1 In the Nav tree, expand the  **Schedules** extension of the NiagaraStation that represents the Supervisor, and double-click it for the **Niagara Schedule Import Manager** view. This view lists schedules that are currently imported from the Supervisor (if any).
- If there are none, click the  **New** button in the Schedule Import Manager to add a new schedule, selecting the type in the first **New** dialog, and enter the desired name in the second **New** dialog. After adding it, it should appear in the Nav tree with a Niagara ScheduleImportExt descriptor ( ext).
  - If imported schedules already exist, expand a  schedule in the Nav tree to see if its Niagara ScheduleImportExt descriptor ( ext) already has a ScheduleExportTag .
- Step 2 Drag a **ScheduleExportTag** from the **exportTags** palette onto a schedule’s Niagara schedule import descriptor ( ext).
- The **Name** dialog appears. Either accept “ScheduleExportTag”, or enter another name and click **OK**. Note by default, the resulting Niagara master schedule (and ScheduleExportExt descriptor) in the Supervisor use the name of the (grandparent) schedule.
- Step 3 Double-click the parent Niagara ScheduleImportExt descriptor ( ext) to access its property sheet, which also contains the ScheduleExportTag  (expand it also).
- The following properties are key:
- **Supervisor Id** — (ScheduleImportExt) location of source Supervisor’s (master) schedule, using syntax as follows:  
`slot:path`  
where *path* includes parent folder(s) and the name of the schedule component, for example:  
`slot:/Schedules/Ltg/LtgAnnex`  
for a schedule named “LtgAnnex” located in the “Schedules”, “Ltg” folder on the Supervisor. If adding a new schedule, this property may be blank—it is populated upon a successful Join.
  - **Override Supervisor Id** — (ScheduleExportTag) Boolean as to whether to override any existing Supervisor Id in the parent import descriptor, and use instead the “NiagaraStation-relative” Override Supervisor Id Station Slot Path value (next property). By default, this is `false`. Typically, you set this to true only when making a *new* master schedule on the Supervisor, at the location specified in the next property (relative to the NiagaraStation that represents this JACE).
  - **Override Supervisor Id Station Slot Path** — (ScheduleExportTag) Parent folder location of the source Supervisor’s (master) schedule if “Override Supervisor Id” is set to `true`. The default slot path is simply “slot:”, which means the root of the NiagaraStation that represents this subordinate (JACE) station on the Supervisor.  
Or, click  open folder, to see the **Select Parent Target** dialog with station ready to expand.  
**Note:** You may edit the “profile.bog file” to specify a new target folder structure—see “Verify and edit the Station (.bog) File” on page 1-4 for details.  
When overriding the Supervisor Id, a new master schedule is created on the Supervisor in this parent folder, named the same as the (grandparent) schedule component (slave) that has this export tag.
- Step 4 The remaining ScheduleExportTag property is standard to Niagara schedule export descriptors, noting briefly:
- **Export Execution Time** — standard selections for a Niagara schedule export to occur, using trigger selections for mode and time, as well as last and next (calculated) times for schedule export.
- Step 5 Click  **Save** after finishing all property changes.  
For further details, see “About ScheduleExportTags” on page 2-34.

## Adding FileImportTags

The FileImportTag lets you import a file (or entire folder of files) from the subordinate (JACE) station to the Supervisor station. This tag creates a Niagara file import descriptor in the Supervisor, under the **Files** device extension of the NiagaraStation that represents this subordinate (JACE) station, and imports the associated file(s).

**Note:** *When using PxViewTags, related file imports are automatically included in an “as-needed” manner, without requiring use of FileImportTags.*

### To add FileImportTags

- Step 1 Drag the **FileImportTag** from the **exportTags** palette and drop anywhere in the JACE station’s component space (under **Config**)—no particular parentage is necessary.  
The **Name** dialog appears. Either accept “FileImportTag”, or enter another name and click **OK**. By default, the resulting Niagara file import descriptor (created on the Supervisor) uses this name.
- Step 2 Double-click the FileImportTag  to access its property sheet.
  1. In the **Supervisor Station** property, click the  control and select the Supervisor station, then click **Save**.
  2. For **Station Slot Path**, accept the default “slot:files” (for root of **Files** under the NiagaraStation). This is the *only valid parent* for the resulting Niagara file import descriptor.
  3. In **File Import Name**, either accept the default variable (%defaultFileImportName%), or enter an explicit name for the Niagara FileImport descriptor. The default (variable) results in a name that uses the name of source file (or folder) appended to “\_import”.
- Step 3 In the **File** property, you specify the local file or directory to import to the Supervisor. The default value is “file:^”, which imports all objects in the local file space.  
Click either:
  - if a single file, the folder  control for the **File Chooser** dialog.
  - if a directory, the adjacent drop-down  control, then choose **Directory Ord Chooser** from the menu for the **Directory Chooser** dialog.Navigate to the source file or directory in the dialog and then select it.  
Example File property values are: file:^reports (for a “reports” directory in the local file space) and file:^reports/moreReports/subReport2.txt (for a single file found in a specific subdirectory).
- Step 4 In the **Supervisor Directory** property, you can accept the default value using ord variables: “file:^stations\$(stationName)\$(currentLocation)”, which replicates a similar file/subfolder location on the Supervisor, but under a “stations” file subfolder organized by stations, including this one. Or, you specify another explicit location under the Supervisor’s file space (file:^location).
- Step 5 In the two other file import properties, you can accept default values or change, if needed.
  - **File Overwrite Policy** — either Checksum (default) or Last Modified.
  - **Execution Time** — standard selections for a Niagara file import.
- Step 6 The **Use Category Mask** boolean specifies whether the imported file(s) will have any export tag *category filters* applied to it (default is true). For related details, see “[About CategoryFilters](#)” on page 2-8.
- Step 7 Click **Save** after finishing all property changes.  
For further details, see “[About FileImportTags](#)” on page 2-36.



## Adding ComponentTags

Component tags create a copy of the component from the source station and save it on the Supervisor, for limited “general purpose” usage. Component tags do not create a proxy, meaning that if you change properties on the component with the tag in the source station, those changes are not reflected in the copy of the component located in the Supervisor station. Often, the component tags apply to source components that have an “out” slot but have a “null” (or no) proxyExt, such as many found in the kitControl module. Or, they may be used with “custom” components. In general, usage on any “container” component is not recommended.

### To add ComponentTags


- Step 1 Drag the **ComponentTag** from the exportTags palette onto the target component.  
The **Name** dialog appears, in which you can accept “ComponentTag”, or enter another name, and click **OK**. Note the resulting component uses the name of the referenced component, and not this name.
- Step 2 Double-click the ComponentTag  to access its property sheet.




1. In the **Supervisor Station** property, click the  control and select the Supervisor station, then click  **Save**.
2. For **Station Slot Path**, do either of the following:
  - Enter a slot path to use on the Supervisor station for the container holding this component. The default is “slot:”, which means the root of the NiagaraStation representing this JACE. For example, if you enter `slot:Stuff/Programs`  
Upon a Join, at the Supervisor station, the resulting component will be in this location:

```
NiagaraNetwork
├─JaceStationName
│   └─Stuff (folder)
│       └─Programs (folder with exported component)
```

“BFormat” variables are also an alternative. See “[Station Slot Path considerations](#)” on page 2-21. Or, (if desired) use the station “profile.bog” file method”, described next.

- Or, click  open folder, for the **Select Parent Target** dialog. Expand the **niagaras-tation** and select a target container for the component to appear in the Supervisor station (default (slot:) puts the component in the root of NiagaraStation).

**Note:** You may edit the “profile.bog file” to specify a new target folder structure—see “[Verify and edit the station profile.bog file](#)” on page 1-4 for details.

Step 3 Click  **Save** after finishing all property changes.

For further details, see “[About ComponentTags](#)” on page 2-38.

## Issue Join commands

After adding NiagaraNetwork Export Tag extensions and one or more export tags, you can issue Join commands to make additions/changes in the Supervisor station. (For those previous procedures, see “[Add the NiagaraNetwork Export Tag extensions](#)” on page 1-2 and “[Add export tags](#)” on page 1-6.)







You can issue a Join from “either side,” that is, from the Supervisor station or from the subordinate (JACE) station.

- [Join from the Supervisor](#)
- [Join from the subordinate \(JACE\) station](#)

### Join from the Supervisor

A Join from the Supervisor is one of *two* ways to perform an export tag Join. The other way is to [Join from the subordinate \(JACE\) station](#).

#### To Join from the Supervisor

- Step 1 In Workbench, open the Supervisor station.
- Step 2 In the Nav tree, expand its Config **Drivers/NiagaraNetwork**  container, to reveal subordinate (JACE)  or  stations.
- Step 3 Right-click a station and select **Actions > Join**.  
You can also issue a **Join** for selected station(s) in the Supervisor’s **Station Manager** view, as well as from the property sheet of any NiagaraStation that represents a subordinate (JACE).  
The Join silently executes, except for a brief Job popup “Supervisor Join” notification near your PC’s system tray area.
- Step 4 After issuing a Join, to see Job Log details in the Supervisor station, do the following:
  1. In the Nav tree, expand the  **Services** folder.
  2. Double-click the  **JobService** for the **Job Service Manager** view.
  3. In the row for any “Supervisor Join” job, click the double-arrow  details control.

A popup **Job Log** dialog lists all operations performed in the Join.

**Note:** Starting in AX-3.7, “Fox SSL” may be used in connections between the JACE stations and the Supervisor, depending on configuration. In this case, note that initial Join attempts may fail due to “non-approved certificate host name validation”. If so, you will see the appropriate errors in a Join’s job log. You must manually approve certificates between the client and server before Joins can succeed. Typically you do this from the “Allowed Hosts” tab of the “Certificate Management” view (on platform service: CertManagerService) in the stations. For complete details on AX-3.7 and later SSL, refer to the NiagaraAX SSL Connectivity Guide.

For related details about a Join (in general) as well as a Join from a Supervisor, refer to “About a Join” on page 2-2 and “About the Join process” on page 2-4.

### Join from the subordinate (JACE) station

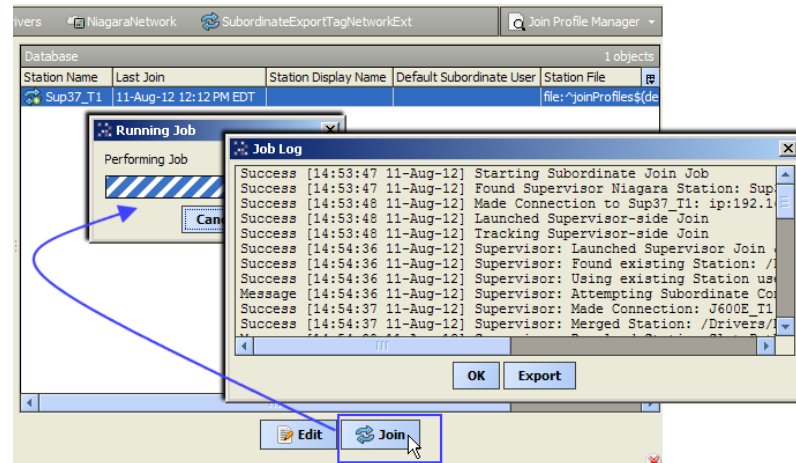
A Join from a subordinate JACE station is one of *two* ways to perform an export tag Join. The other way is to [Join from the Supervisor](#) (providing that the Supervisor already has a working NiagaraStation entry for this JACE in its NiagaraNetwork).

However, a Join from the JACE is possible even in the scenario where the Supervisor is currently “unaware” of this JACE. Additionally, a Join from a JACE provides a built-in “job log”, as described.

### To Join from the subordinate (JACE) station

- Step 1 In Workbench, open the JACE station.
- Step 2 In the Nav tree, expand its Config **Drivers/NiagaraNetwork** container, to reveal the child **SubordinateExportTagNetworkExt** and Supervisor) or station.
- Step 3 Double-click this network extension for its **Join Profile Manager** view
- Step 4 In the **Join Profile Manager**, click the Supervisor station to select it, then click the **Join** button at the bottom of the view.

**Figure 1-7** Join from subordinate’s Join Profile Manager view provides automatic Job Log popup



As shown in Figure 1-7, the Join executes and ends with a popup **Job Log** that lists all operations performed in the Join. This lets you review all changes, and if desired, save (export) to a text file. In case of any *errors* encountered in the Join, the job log can help locate specific export tag problems.

In addition, a brief Job popup “Subordinate Join” notification appears near your PC’s system tray area.

**Note:** Starting in AX-3.7, “Fox SSL” may be used in connections between the JACE stations and the Supervisor, depending on configuration. In this case, note that initial Join attempts may fail due to “non-approved certificate host name validation”. If so, you will see the appropriate errors in a Join’s job log. You must manually approve certificates between the client and server before Joins can succeed. Typically you do this from the “Allowed Hosts” tab of the “Certificate Management” view (on platform service: CertManagerService) in the stations. For complete details on AX-3.7 and later SSL, refer to the NiagaraAX SSL Connectivity Guide.

You can also issue a **Join** in the subordinate station as a “right-click action” on the **Subordinate-ExportTagNetworkExt**, or on the Supervisor station as listed in the Join Profile Manager view. However, if you invoke the Join using either of those methods, you do not see the “automatic” popup Job Log dialog with all the details from the Join, as shown in Figure 1-7.

**Note:** In AX-3.7u1, when issuing a Join from a subordinate JACE station, and that station does not already exist in the Supervisor, the “Default Subordinate User Password” property must be configured with a valid password for the selected “Default Subordinate User”. If no valid user and password has been entered the Join will fail due to authentication errors. For further details, see “Editing the Join Profile” on page 2-13. For related details about a Join (in general) as well as a Join from a JACE, refer to “About a Join” on page 2-2 and “Join process from subordinate” on page 2-5.





# CHAPTER 2

## NiagaraAX Export Tag Concepts

---

This section provides details on the “Export Tags” feature first introduced in AX-3.5. Included is an engineering overview, followed by details about the exportTags palette, various export tag components, and related views.

The following main sections are included:

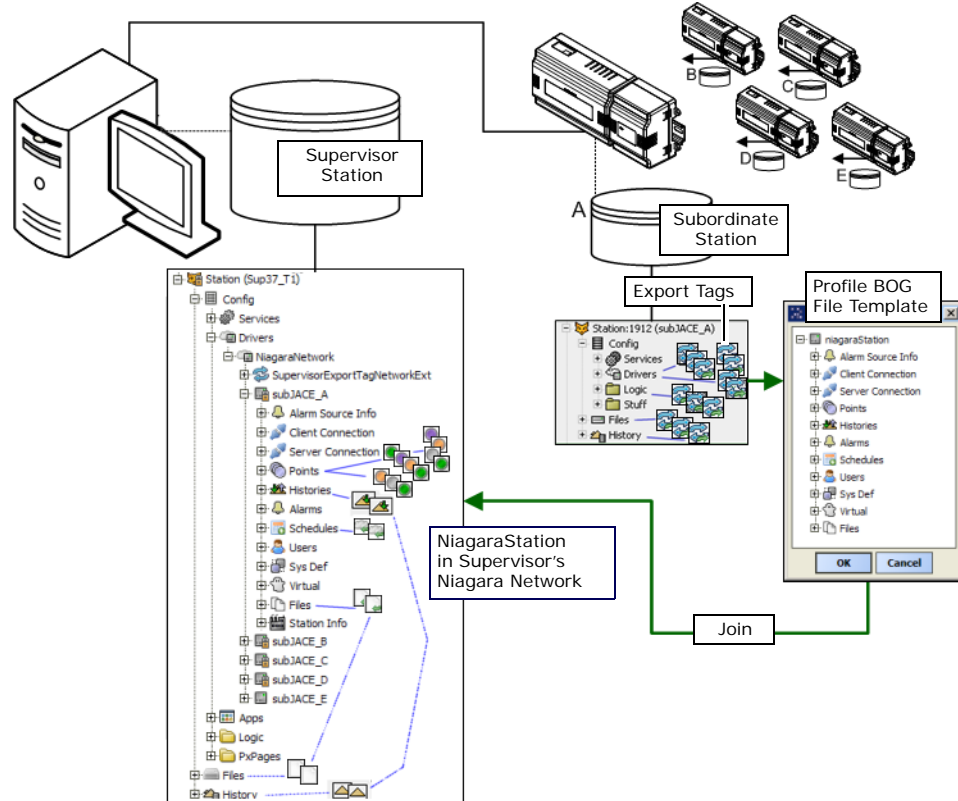
- “Export tags engineering overview” on page 2-2
  - “About a Join” on page 2-2
  - “About the JoinProfile” on page 2-3
  - “About the Join process” on page 2-4
- “exportTags palette” on page 2-6
- “About the SupervisorExportTagNetworkExt” on page 2-7
- “About CategoryFilters” on page 2-8
  - “About the Category Filter Manager” on page 2-8
  - “Adding CategoryFilters” on page 2-9
- “About the SubordinateExportTagNetworkExt” on page 2-11
  - “About the Join Profile Manager” on page 2-12
  - “Editing the Join Profile” on page 2-13
  - “About Station Info” on page 2-15
  - “About the Export Tag Summary Manager” on page 2-16
  - “About the Export Tag Summary Manager” on page 2-16
- “About the niagaraStation profile.bog file” on page 2-18
  - “Device extension properties importance in profile.bog file” on page 2-19
  - “Provisioning slot additions in a profile.bog file” on page 2-20
- “Station Slot Path considerations” on page 2-21
  - “BFormat options in Station Slot Paths” on page 2-22
- “About PxViewTags” on page 2-25
  - “PxViewTag properties” on page 2-25
  - “PxViewTag action” on page 2-27
  - “About the Supervisor SubstitutePxView” on page 2-27
  - “About ord substitution overrides” on page 2-28
- “About PointTags” on page 2-30
  - “PointTag properties” on page 2-30
- “About HistoryImportTags” on page 2-31
  - “HistoryImportTag properties” on page 2-31
- “About ScheduleImportTags” on page 2-34
  - “ScheduleImportTag properties” on page 2-34
- “About ScheduleExportTags” on page 2-34
  - “ScheduleExportTag properties” on page 2-35
- “About FileImportTags” on page 2-36
  - “FileImportTag properties” on page 2-36
- “About ComponentTags” on page 2-38
  - “ComponentTag properties” on page 2-38
  - “ComponentTag example (ExportTagProgram)” on page 2-39

## Export tags engineering overview

Essentially, the export tags engineering process is “place export tags in subordinate station”, that is, you work from the “JACE side”. Using Workbench, do this by copying export tags from the **Extensions** folder in the `exportTags` palette. This is commonly called “tagging up the JACE”.

When you issue the *Join* command, resulting objects are created in the Supervisor (Figure 2-1).

**Figure 2-1** Export tags in subordinate station create components, histories, and files in Supervisor



Exported *components* are created in the Supervisor under the NiagaraStation component representing the subordinate station that was joined. This includes Px views (complete with Niagara “virtual points” and imported image files), history and file import descriptors, schedule import and export descriptors, plus components such as Niagara proxy points.

As part of the “JoinProfile” on a subordinate station, a “profile.bog file” on the JACE determines the property values used in the corresponding NiagaraStation (including its device components) in the Supervisor station. Optionally, this profile.bog file can also determine the container/folder structure used in this NiagaraStation in the Supervisor. Each export tag can reference a specific location in this profile.bog station, to be used as the parent container for its associated exported component.

Exported *files* and *histories* result in objects created in the Supervisor’s file space (Files) and history space (History), in addition to their associated file and history import descriptors (components) created under a NiagaraStation. Under the Files and History object spaces, created files and histories are automatically organized by subordinate station (not shown in Figure 2-1).

The following sections explain more export tag concepts:

- [About a Join](#)
- [About the JoinProfile](#)
- [About the Join process](#)

### About a Join

The Join command launches the export tag process on the Supervisor, where resulting changes are made in the Supervisor’s database. Specifically, all component changes from a join apply to the NiagaraStation (and its child components) that represents that subordinate.

Some additional Join concepts are as follows:

- [Join is a merge](#)
- [Join and rejoin](#)
- [Join from either side](#)
- [Join profile is on subordinate](#)

### Join is a merge

A Join *merges* export tags and join profile information found in the remote subordinate station with any existing NiagaraStation representation, *before* making any changes. Existing Niagara proxy points, imported histories, and so on, are not arbitrarily overwritten—instead, merge “intelligence” is applied.

For example, points previously added using the Niagara Proxy Point Manager, and imported histories are typically retained. Such items may co-exist with items made from export tags. Sometimes, a component may be deleted and re-added in a new location. For example, this happens if one or more export tags (that were used in a previous join) were since edited to use a different “station slot path”.

### Join and rejoin

Export tag Joins may be repeated as often as needed. When first adding a subordinate station, you may need to issue many Joins before the NiagaraStation is “finished” in the Supervisor. After engineering is completed for that subordinate, a Join may be infrequently issued, if at all.

In the case of very large jobs with replicated subordinate (JACE) stations, new subordinates may be added and be fully operational after a single Join.

### Join from either side

You can issue an export tag Join on a station from either the Supervisor station or from the subordinate station, as needed—you get the same results.


Issuing a Join from the subordinate’s **Join Profile Manager** ends with a built-in **Job Log** popup dialog that lists all changes made in the Supervisor as a result of the join. This includes both additions as well as possible deletions, each with related ords. Whereas, a Join issued from the Supervisor silently executes, without an ending popup job log. However, note that the same (join) job log data is available under the Supervisor’s JobService, in the details for that Join job.

A Join issued from the subordinate makes an extra initial connection to the Supervisor. Through a few available JoinProfile properties, this allows even adding a new subordinate in cases where the Supervisor has no “prior knowledge” of a remote host. However, you may issue Joins from a subordinate simply for the convenience of popup job logs.

### Join profile is on subordinate

All Join information about a subordinate station is configured and maintained on that *subordinate station*—and not the Supervisor. In addition to any distributed export tags, the JoinProfile component under the NiagaraStation that represents the Supervisor is the key holder of important properties. See the next section “[About the JoinProfile](#)” for a summary.

### About the JoinProfile

As a result of adding a subordinate network extension (SubordinateExportTagNetworkExt) to a subordinate (JACE) station’s NiagaraNetwork, each of its child NiagaraStation components has a  JoinProfile component added. The *only one of importance* is the one under the NiagaraStation that represents the Supervisor *station*—any others are not used, and can be left at defaults.

The JoinProfile has several properties you can edit with optional values, such as “Station Display Name”, as well as default properties for the Supervisor to make a Fox client connection back to this subordinate station (not required if the subordinate is already configured in the Supervisor’s NiagaraNetwork).

**Note:** *In AX-3.7u1, when issuing a Join from a subordinate JACE station, and that station does not already exist in the Supervisor, the “Default Subordinate User Password” property must be configured with a valid password for the selected “Default Subordinate User”. If no valid user and password has been entered the Join will fail due to authentication errors. These user and password properties may be left blank if the subordinate station already exists in the Supervisor’s NiagaraNetwork. For further details, “[Editing the Join Profile](#)” on page 2-13.*

Also included, a “Station File” property with the file ord to the “profile.bog file” that resides on the subordinate station. This file is automatically created when you *enable* the JoinProfile. You can use it to model the folder/container hierarchy of this subordinate station, as it will be created (after a Join) on the Super-

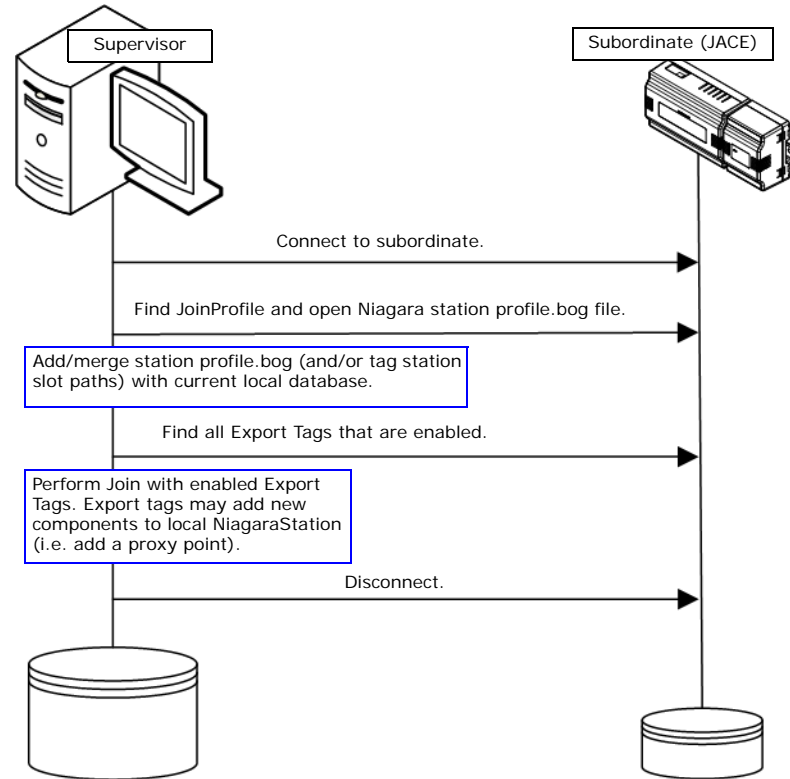
visor under its NiagaraNetwork. In addition, this profile.bog file holds various property values used in this NiagaraStation (and its device extensions) on the Supervisor. For more details, see “About the niagaraStation profile.bog file” on page 2-18.

Also, the Join Profile has a “Station Info” child container, empty by default. You can add string slots under it and assign values that can provide organizational utility, useful in very large systems. For more details, see “About Station Info” on page 2-15.

### About the Join process

Figure 2-2 below shows the basic process steps in a Join performed from the Supervisor station.

Figure 2-2 Join process steps when Join is issued from Supervisor station.



As shown in this diagram, once connected, the Supervisor finds the JoinProfile on the subordinate and opens its station “profile.bog file”. The Supervisor then merges the profile.bog file (and/or individual “Station Slot Paths” in export tags) with the current NiagaraStation structure. Then the Supervisor finds all enabled export tags on the subordinate, and performs the join, merging with existing components under the NiagaraStation.

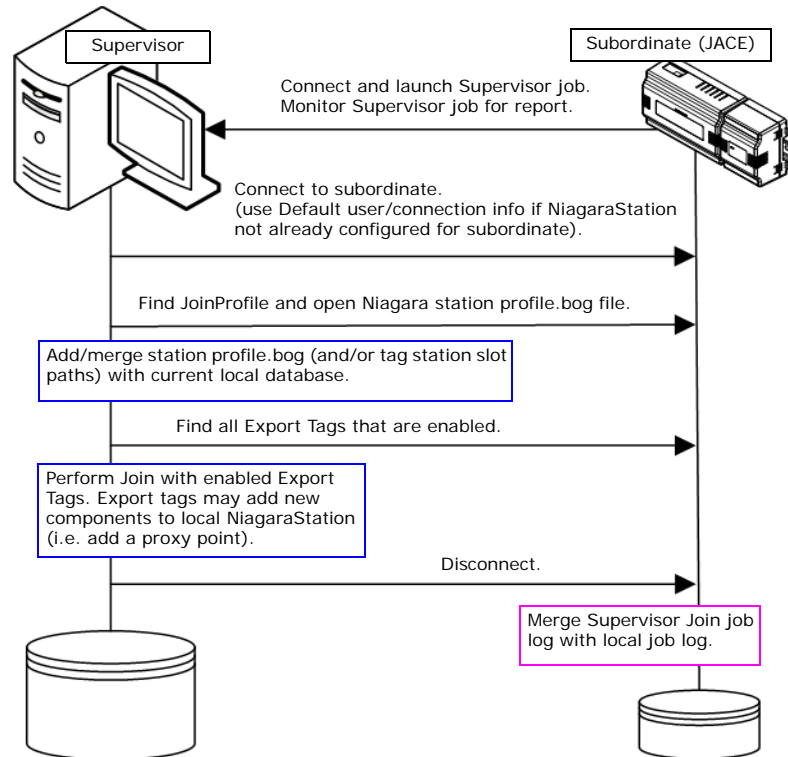
The Supervisor then disconnects. Details about the Supervisor Join are in its local job log.

**Note:** Starting in AX-3.7, “Fox SSL” may be used in connections between the JACE stations and the Supervisor, depending on configuration. In this case, note that initial Join attempts may fail due to “non-approved certificate host name validation”. If so, you will see the appropriate errors in a Join’s job log. You must manually approve certificates between the client and server before Joins can succeed. Typically you do this from the “Allowed Hosts” tab of the “Certificate Management” view (on platform service: CertManagerService) in the stations. For complete details on AX-3.7 and later SSL, refer to the NiagaraAX SSL Connectivity Guide.

### Join process from subordinate

If the Join is issued from the subordinate station, the same process steps occur as shown in Figure 2-2, plus an extra, initial connection step from the subordinate to the Supervisor, as shown in Figure 2-3.

**Figure 2-3** Extra initial connection in processes for Join issued from subordinate station



Note that this method allows a new subordinate to be added to the NiagaraNetwork of the Supervisor even in cases where the Supervisor is “unaware” of the remote host, providing the subordinate station’s NiagaraNetwork has a NiagaraStation node representing the Supervisor, complete with configured Fox client connection.

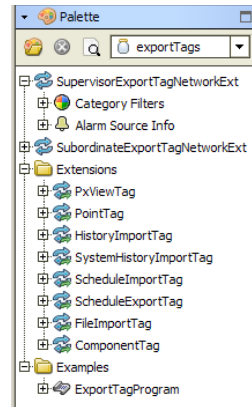
In this scenario the subordinate’s JoinProfile would require valid entries for “Default Subordinate User” (typically a service account for station to station communications with full, admin-level, permissions,) “Default Subordinate User Password”, “Default Subordinate IP Address”, and possibly “Default Subordinate Port” properties. For further details, see “Editing the Join Profile” on page 2-13.

**Note:** Starting in AX-3.7, “Fox SSL” may be used in connections between the JACE stations and the Supervisor, depending on configuration. In this case, note that initial Join attempts may fail due to “non-approved certificate host name validation”. If so, you will see the appropriate errors in a Join’s job log. You must manually approve certificates between the client and server before Joins can succeed. Typically you do this from the “Allowed Hosts” tab of the “Certificate Management” view (on platform service: CertManagerService) in the stations. For complete details on AX-3.7 and later SSL, refer to the NiagaraAX SSL Connectivity Guide.

## exportTags palette

When “tagging up” a subordinate (JACE) station opened in Workbench, you can copy (drag and drop) from the exportTags module’s palette, shown in [Figure 2-4](#).

**Figure 2-4** Palette of the exportTags module



**Note:** Often, only a few types of export tag extensions are used—for example; *PxViewTag*, *HistoryImportTag*, and possibly *PointTag*.

Each of these components is briefly described as follows:

- **SupervisorExportTagNetworkExt**  
This is the *only* component that goes in the Supervisor station (*not* subordinate station). You copy *one* into the NiagaraNetwork of the Supervisor. For configuration details on its properties and child components, see [“About the SupervisorExportTagNetworkExt”](#) on page 2-7.
- **SubordinateExportTagNetworkExt**  
You copy *one* into the NiagaraNetwork of a subordinate station. It provides two special views for working with export tags, and also produces a JoinProfile container under each NiagaraStation component in the station’s NiagaraNetwork. For configuration details on its properties, child components, and views, see [“About the SubordinateExportTagNetworkExt”](#) on page 2-11
- **PxViewTag**  
Use this export tag to export any PxView in the subordinate station up to the Supervisor station. Copy it on the folder/container in the JACE station with the PxView, then specify it as a view on any designated folder/container under the “profile.bog file” (determines the NiagaraStation structure on the Supervisor). Several properties in the tag require configuration.  
PxViewTags are a powerful feature, and automatically encapsulate several other functions, including “on demand” Niagara virtual components for real-time values, along with importing of any graphics files used in the PxView. The resulting “virtual” or “substitute PxView” operates like the original PxView in the subordinate station, without the overhead of Niagara proxy points.  
In some use cases, it is possible that PxViewTags may preclude the need for PointTags and ComponentTags, comprising the vast majority of export tags in a subordinate station. For details, see [“About PxViewTags”](#) on page 2-25.
- **PointTag**  
Use a PointTag to export any control point or proxy point in the subordinate station as a Niagara proxy point in the Supervisor station. Simply copy the PointTag onto the source point in the subordinate station (for example, a Lonworks proxy point or BACnet proxy point). For details, see [“About PointTags”](#) on page 2-30.
- **HistoryImportTag**  
Use a HistoryImportTag to export a particular history in the subordinate station. Simply copy the HistoryImportTag onto the point’s history extension, or alternatively, under the history extension’s “HistoryConfig” child container. For details, see [“About HistoryImportTags”](#) on page 2-31.
- **SystemHistoryImportTag**  
Use a SystemHistoryImportTag to export groups of histories in the subordinate station that have matching “System Tags”, utilizing a feature introduced starting in AX-3.4. You can copy a SystemHistoryImportTag anywhere in the station. For details, see [“About SystemHistoryImportTags”](#) on page 2-32.
- **ScheduleImportTag**  
Use a ScheduleImportTag to designate an existing schedule on the subordinate station as a “master”, thereby creating a “slave” schedule on the Supervisor station. Copy a ScheduleImportTag onto any



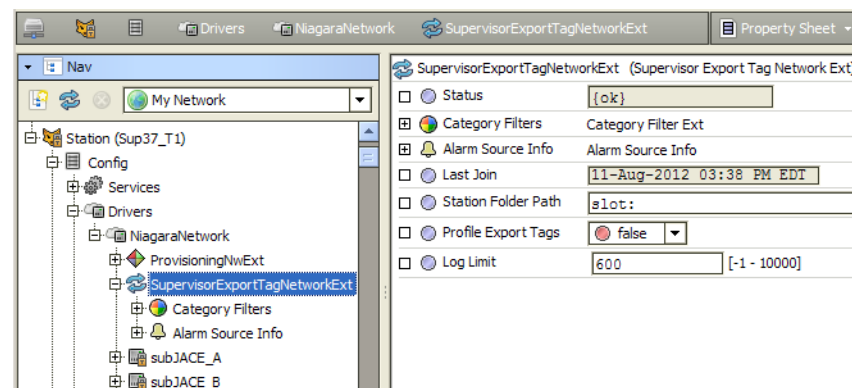
existing schedule (BooleanSchedule, NumericSchedule, and so on) in the station. Usage of this export tag is expected to be infrequent. For details, see “About ScheduleImportTags” on page 2-34.

- **ScheduleExportTag**  
Use a ScheduleExportTag to associate (or create) a “master” schedule on the Supervisor station with a new “slave” schedule on the subordinate station. Copy a ScheduleExportTag onto the Niagara ScheduleImportExt descriptor of a new schedule (BooleanSchedule, NumericSchedule, and so on) in the station’s Schedules extension under the NiagaraStation for the Supervisor. For details, see “About ScheduleExportTags” on page 2-34.
- **FileImportTag**  
Use a FileImportTag to export a specific file or folder(s) of files to the Supervisor. Copy the FileImportTag anywhere under the component space (Config) of the subordinate station, and configure the source file ord in the “File” property. Note that file imports can also automatically occur as a result of using PxViewTags. For details, see “About FileImportTags” on page 2-36.
- **ComponentTag**  
Use a ComponentTag to export a copy of a component other than a control point or proxy point, for example a kitControl component or Program component.  
*Note:* Component tags do not create a proxy. If you change properties on the component with the tag in the source station, those changes are not reflected in the copy of the component located in the Supervisor station.  
Simply copy the ComponentTag onto the source component. You can specify the exported component to be in any designated folder/container under the NiagaraStation (that represents the subordinate). For details, see “About ComponentTags” on page 2-38.
- **ExportTagProgram**  
This is an example Program component with a ComponentTag already applied, to demonstrate one application of the ComponentTag. Usage allows automating any “post join” operations from any additional supplied program logic, within the “template” design of the program code. Use is targeted for users who are experienced with Program components. Possible post-join operations might include creating links. For details, see “ComponentTag example (ExportTagProgram)” on page 2-39.

## About the SupervisorExportTagNetworkExt

The SupervisorExportTagNetworkExt is required for a Supervisor station to utilize export tags used in remote subordinate (JACE) stations. It is the *only* component you copy directly from the **exportTags** palette into a Supervisor station. Place it in the root of the NiagaraNetwork, as shown in Figure 2-5.

Figure 2-5 SupervisorExportTagNetworkExt in Supervisor station, showing property sheet



The property sheet is the default view (no specialized views), with properties described as follows:

- **Status**  
Read-only status of the network extension, typically “ok”.
- **Category Filters**  
Container for any CategoryFilter export tag components, where usage is optional. The default view is the **Category Filter Manager**, for adding, editing, and deleting CategoryFilter components. For more details, see the next section “About CategoryFilters”.
- **Alarm Source Info**  
A container slot you can use to differentiate export tag alarms (from a failed Join) from other alarms in the station. See “About network Alarm Source Info” in the *Drivers Guide* for general details.

**Note:** *The Supervisor's AuditLog also records both successful and failed "exportTagJoin" jobs, including information on changed slot values (old and new), and also the originating user.*

- **Station Folder Path**  
Allows usage of "Station Info" metadata in remote JACE stations to automatically, upon a Join, create NiagaraStationFolders (device folders) on the Supervisor, to organize NiagaraStations under them. To do this, this property value must use "ord variables" that correspond to known string slots that have been added in the remote stations, under each JoinProfile, "Station Info" slot. Typically, this feature is useful on extremely large systems, where a Supervisor may have many subordinate JACE stations.
  - The default "slot:" is used for normal operation, meaning *without* creation of station folders.
  - Optionally (although rarely used), you can append explicit text string(s), used by any subordinate station Join, regardless of its JoinProfile. For example: "slot:test/Example" will put all joined NiagaraStation components in a NiagaraStationFolder named "Example", itself a child of another NiagaraStationFolder named "test".
  - More typically, you append one or more *ord variables*, using a "\$()" syntax for each hierarchical level of NiagaraStationFolder you want created. To work, each ord variable must match the name of a string slot added to the "Station Info" container under a remote station's JoinProfile. For example, a value of: "slot:\$(State)\$ (City)" that corresponds to remote Station Info string slots "State" and "City", will result in two levels of NiagaraStationFolders. Each folder will be named the value of that slot, for example a "Virginia" folder may have subfolders "Richmond" and "Norfolk", each of which contains one or more NiagaraStation components. For related details, see "About Station Info" on page 2-15.
- **Profile Export Tags**  
Specifies whether to include "Join time" statistics in the job log for each Join, reported in milliseconds (ms) for each item evaluated in a "Pre-Join", "Join Time", and "Post Join" fashion. The default is false, meaning no Join time statistics. Enabling may be useful in some cases where slower connections between the Supervisor and subordinate hosts are used, for example via GPRS modem.
- **Log Limit**  
Specifies the maximum (line) entries allowed in any job log in the Supervisor for any export tag Join, where the default is 600. If a job log exceeds this limit, earlier entries are not captured. Range is from 1 to 10,000. Note that if Profile Export Tags (above) is true, many additional line entries result.

## About CategoryFilters

**CategoryFilters** is the frozen container slot of the [SupervisorExportTagNetworkExt](#) in a Supervisor station, used to hold **CategoryFilter** components that specify security categories on objects created via export tags. This container has no configuration properties; however, its importance is in its default view, as well as any child CategoryFilter components.

Usage of export tag category filters is optional. However, in some larger Supervisor configurations, this feature may be needed for "station level separation" of user access via security categories.


For example, consider a Supervisor that is accessed by two diverse organizations: "bigOil" and "clean-Water". Users from each organization need login access to monitor their own sites, including PxViews, histories, and so on. However, it is important that they cannot see each other's data. In this case, this separation of access in the Supervisor can be quickly accomplished at the "NiagaraStation level" using Category Filters.

A prerequisite of CategoryFilters is the configuration of "Station Info" under the JoinProfile in each subordinate (JACE) station. This involves adding one or more string slots and assigning string property values. Upon a Join, this metadata is exported to the Supervisor as properties of the "Station Info" extension of the NiagaraStation component that represents each JACE station. For related details, see "About Station Info" on page 2-15.

The following subsections provide more details on a Supervisor's export tag Category Filters:

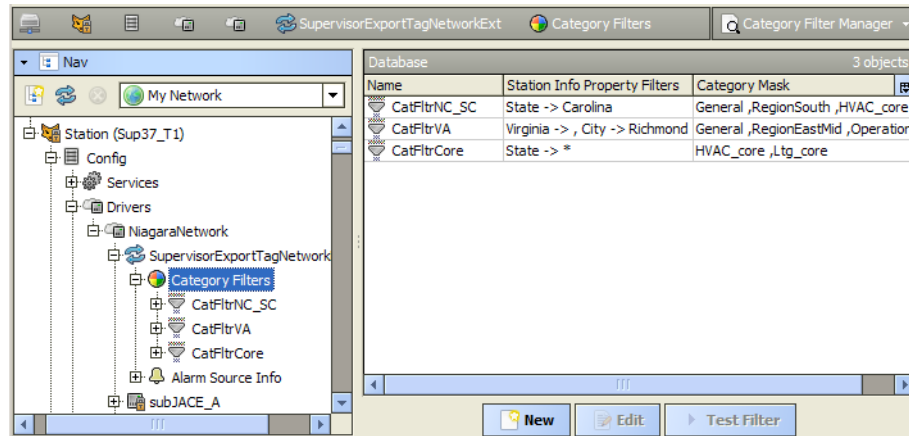
- [About the Category Filter Manager](#)
- [Adding CategoryFilters](#)

### About the Category Filter Manager

The **Category Filter Manager** is the default view of  **CategoryFilters** under the **SupervisorExportTagNetworkExt** in the Supervisor station. Double-click this container slot to access this view, as shown in [Figure 2-6](#).



**Figure 2-6** Category Filter Manager is default view of CategoryFilters container



**Note:** The order of CategoryFilters can be important, as they are evaluated in order from “top down”, as only one CategoryFilter (and associated category mask) is applied to any NiagaraStation. In other words, category masks are not “multiplexed” or “additive”.

As needed, use this view for adding, editing, reordering, and reviewing child **CategoryFilter** components. Default columns displayed in the **Category Filter Manager** include the following:

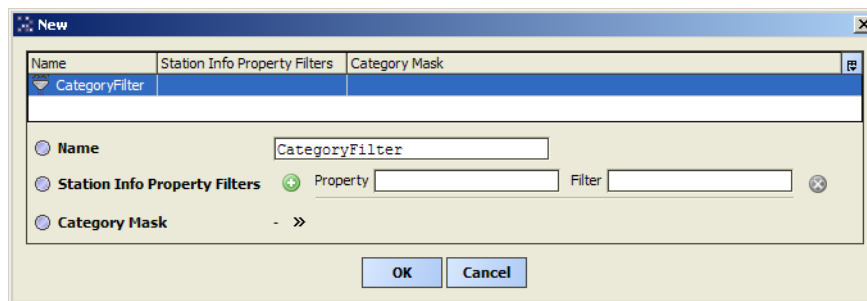
- **Name**  
Name of each CategoryFilter component, which must be unique.
- **Station Info Property Filters**  
Currently configured “Station Info Property Filters”, using “slotName -> propertyValue” notation. If multiple filters, each is separated by a comma ( , ).
- **Category Mask**  
The category(ies), by name, currently assigned to each CategoryFilter, where a comma ( , ) separates each one if multiples are assigned.

Buttons at the bottom of this view allow you to make **New** or **Edit** existing CategoryFilter components. See the next section “Adding CategoryFilters”.

### Adding CategoryFilters

When you click **New** in the Supervisor’s export tag **Category Filter Manager**, a popup dialog appears asking how many (the only type selectable is Category Filter). Following this, another popup dialog for configuration appears, as shown in Figure 2-7.

**Figure 2-7** New dialog for adding CategoryFilter



The following fields are in this dialog:

- **Name**  
Name of each CategoryFilter component, where each must be unique.
- **Station Info Property Filters**  
This is where you enter the name/value pair that matches a slot in the “Station Info” under the Join-Profile of one or more subordinate stations. Note that in each field, case must be exact.
  - Property — this is the slot name, for example, “State” or “City”.
  - Filter — the string *value* of that slot to filter on, for example, “Virginia” or “Richmond”.

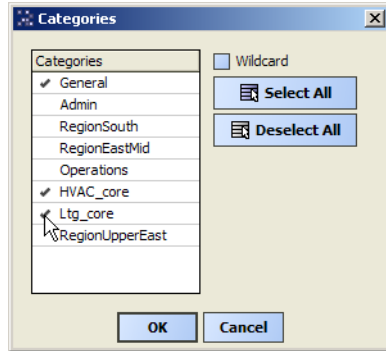
If needed, use “wildcard” (asterisk) characters, for example “\*Carolina\*” for slot “State” could potentially filter on “North Carolina” and “South Carolina”.

Use the plus control to add fields for additional name/value pairs, if needed in the filter match. If using multiple pairs in the same filter, all must be true (match) to apply the category mask. If needed, click the delete control to remove the fields for a name/value pair.

- **Category Mask**

Click the double-arrow control for a popup dialog listing available Categories in the Supervisor, as shown in [Figure 2-8](#).

**Figure 2-8** Category popup dialog to define Category Mask in Supervisor



**Note:** Category configurations (and names) vary greatly from one job to another. Your Supervisor will have different category names and likely even more categories.

Click to select and deselect categories. If “Wildcard” is checked, this is similar to “Select All”, plus any new (future) categories will also be selected. Other buttons are self-explanatory.

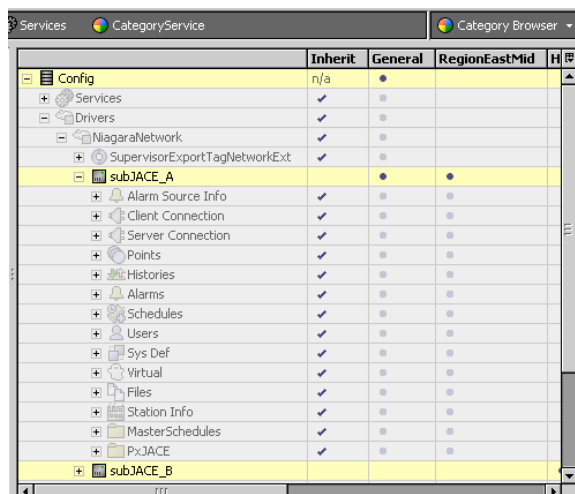
After adding the category mask, it appears in the **New** or **Edit** dialog using hexadecimal notation, reflecting a bitmapped weighting by category indices, e.g. “14” or “ff”, or “\*” if Wildcard is selected.

The **Test Filter** button allows you to check the configured “Station Info Property Filters” against the Supervisor’s collection of “Station Info” in all of its NiagaraStation components (since the last Join from each), and provides a “total match count” for each filter selected.

### About applied CategoryFilters

Category filters are applied to components, files, and histories of a subordinate as represented in the Supervisor, when you issue a Join from (or to) that station. [Figure 2-9](#) shows the Category Browser view in the Supervisor station following a Join from remote subordinate “subJACE\_A”.

**Figure 2-9** Example categories applied to components under a NiagaraStation in the Supervisor

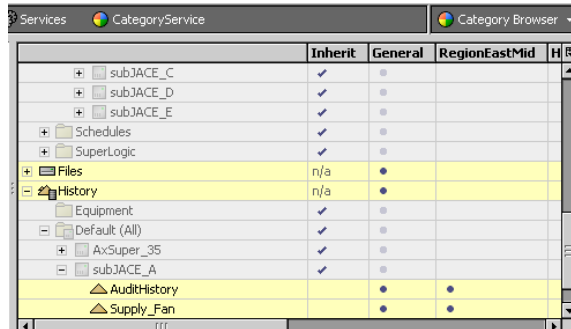


In the [Figure 2-9](#) example, “Station Info” in the subordinate’s JoinProfile includes two slots/values: State=Virginia, City=Richmond. A matching CategoryFilter in the Supervisor applies a category mask including categories “General”, “RegionEastMid”, “HVAC\_core”, “Ltg\_Core”, and “Operations” (some are not visible in the figure).

**Note:** The order of CategoryFilters can be important, as they are evaluated in order from “top down”, as only one CategoryFilter (and associated category mask) is applied to any NiagaraStation. In other words, category masks are not “multiplexed” or “additive”. See Figure 2-6.

In addition to components under each NiagaraStation, categories are also applied to files and histories of stations too. Figure 2-10 shows the Supervisor’s Category Browser in the Files and Histories sections.

**Figure 2-10** Example categories applied to files and histories of a NiagaraStation in the Supervisor

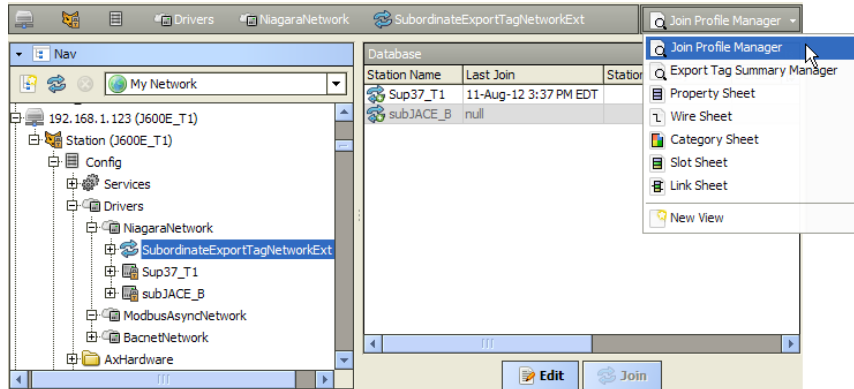


**Note:** In the initial release of export tags, CategoryFilters do not apply to histories created as a result of a SystemHistoryImportTag, only from HistoryImportTags. You must manually apply categories to histories created with SystemHistoryImportTags, if necessary.

## About the SubordinateExportTagNetworkExt

The SubordinateExportTagNetworkExt is required for any JACE station to use export tags. You copy it directly from the exportTags palette into the root of the NiagaraNetwork.

**Figure 2-11** SubordinateExportTagNetworkExt in JACE station, showing view selector



As shown by the drop-down “view selector” in Figure 2-11, there are two specialized views besides the property sheet. The SubordinateExportTagNetworkExt property sheet only has these two items:

- **Status**  
Read-only status of the network extension, typically “ok”.
- **Log Limit**  
Specifies the maximum (line) entries allowed in any job log in this JACE station for any export tag Join, where the default is 600. If a job log exceeds this limit, earlier entries are not captured. Range is from 1 to 10,000.

The following sections describe other aspects of the SubordinateExportTagNetworkExt:

- [Actions on the SubordinateExportTagNetworkExt](#)
- [About the Join Profile Manager](#)
- [Editing the Join Profile](#)
- [About Station Info](#)
- [About the niagaraStation.profile.bog file](#)
- [About the Export Tag Summary Manager](#)

## Actions on the SubordinateExportTagNetworkExt

Two actions are available on the SubordinateExportTagNetworkExt component.

- **Join**  
A way to issue the Join from this JACE station.
- **Set All Supervisor Station Names**  
Allows you to globally specify the target NiagaraStation that represents the Supervisor, applied to the “Supervisor Station” property in *all the export tags* in this station. This can be useful if you have copied in saved devices (from previous jobs) that contain export tags naming other Supervisor stations, as one example. Or, if you did not set the “Supervisor Name” property in export tags already added, either by mistake or by design. This command is also available in the **Export Tag Summary Manager**—see “Specifying the Supervisor station globally in all export tags” on page 2-18.

## About the Join Profile Manager


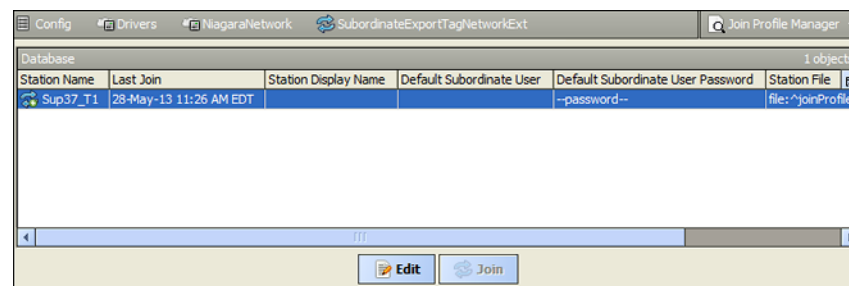

The **Join Profile Manager** is the default view of the  SubordinateExportTagNetworkExt in a subordinate (JACE) station. Double-click this extension to access this view, as shown in Figure 2-12.

Figure 2-12 Join Profile Manager is default view of SubordinateExportTagNetworkExt in JACE station



You can use this view for the initial setup and enabling of the  **JoinProfile** for the Supervisor (click to select this station, then click **Edit**). Later, this is the preferred view to launch Joins via the button at the bottom of the view, in order to get the automatic popup **Job Log** dialog at the end of the Join.

Available columns to display in the **Join Profile Manager** include the following:

- **Station Name**  
Reflects the name of the parent NiagaraStation in the JACE’s NiagaraNetwork.
- **Last Join**  
Date/timestamp of the last Join of the local station to the Supervisor station. This is “null” in the case of a newly-added SubordinateExportTagNetworkExt, before any Join has been issued. This value should remain “null” for any peer JACE stations (those JoinProfiles should also remain *disabled*).
- **Station Display Name**  
Shows any “display name” for the local station (if configured) as it should appear in the Supervisor station. This feature is useful on larger jobs where station names are often short and cryptic, and otherwise not easily identifiable.
- **Default Subordinate User**  
Shows the local station user account the Supervisor should use to connect back to this station, upon receiving an unsolicited Join request. Needed only if the Supervisor’s NiagaraNetwork is not already configured with a NiagaraStation for this JACE, including the proper Fox Service client connection user and password.
- **Default Subordinate User Password**  
In AX-3.7u1 and later, this shows a “placeholder” for the local station user password the Supervisor should use to connect back to this station, upon receiving an unsolicited Join request. This is needed only if the Supervisor’s NiagaraNetwork is not already configured with a NiagaraStation for this JACE, including the proper Fox Service client connection user and password.
- **Default Subordinate IP Address**  
Shows the local host’s IP address the Supervisor should use to connect back to this station, upon receiving an unsolicited Join request. Needed only if the Supervisor’s NiagaraNetwork is not already configured with a NiagaraStation for this JACE.
- **Default Subordinate Port**  
Shows the software port the Supervisor should use for a Fox connection back to this station, upon receiving an unsolicited Join request. Needed only if the Supervisor’s NiagaraNetwork is not already configured with a NiagaraStation for this JACE.

- **Default Subordinate Use Fox SSL**  
 (New starting in AX-3.7) Whether or not a Fox SSL connection should be used by the Supervisor to connect back to this station, upon an unsolicited Join request. The default is false, where a normal Fox connection is used.
- **Default Subordinate Fox SSL Port**  
 (New starting in AX-3.7) Shows the software port the Supervisor should use for a Fox SSL connection back to this station, upon receiving an unsolicited Join request. Applies only if Default Subordinate Use Fox SSL is true.
- **Station File**  
 Shows the local file ord to the “profile.bog file” that acts as the folder/container “template” used on the NiagaraStation that represents this station, as it will be exported to the Supervisor station.

Buttons at the bottom of this view allow you to **Edit** many of these properties in the JoinProfile for the selected Supervisor station, as well as issue a **Join** command (complete with ending Job Log popup).

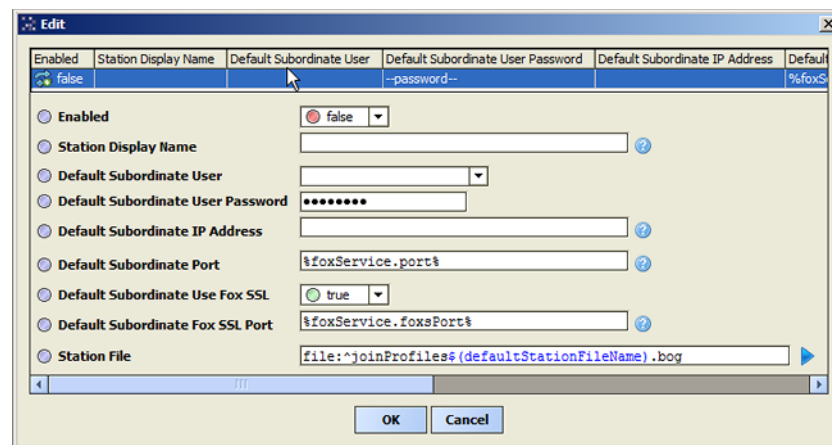
See the next section “Editing the Join Profile” for more information.

### Editing the Join Profile

In the **Join Profile Manager**, select the JoinProfile for the Supervisor, and then click the **Edit** button for a popup dialog, as shown in Figure 2-13.

*Note:* Alternatively, you can simply double-click the JoinProfile for its property sheet, which contains all the properties shown below, plus a “Status” property and a “Station Info” container. See “About Station Info” on page 2-15 for additional details.

**Figure 2-13** Edit dialog for JoinProfile of Supervisor



The following fields are in this dialog:

- **Enabled**  
 By default, this is false. Set this to true on the JoinProfile for the Supervisor station (only).
- **Station Display Name**  
 (Optional) Here you can enter whatever “friendly name” is desired to represent this local JACE station in the Supervisor station. Upon a Join, this appears as the display name of the NiagaraStation that represents this station, instead of the actual component name.  
*Note:* This field also accepts “BFormat” scripting using the JoinProfile’s “Station Info” properties. For example, if Station Info has properties “State” and “City”, and you wanted this incorporated in the station display name, you could enter something similar to this:  

```
Relay Tower (%stationInfo.City% %stationInfo.State%)
```

  
 If the Station Info properties were State=Virginia and City=Blacksburg, the NiagaraStation created in the Supervisor would have the display name: Relay Tower (Blacksburg Virginia)
- **Default Subordinate User**  
 This is the local (JACE) station user account to be used by the Supervisor for Fox client connection—typically a service type account (with admin-level permissions) used for station-to-station connections. If the subordinate station includes component tags added to program objects which are also enabled, both the service account (used for subordinate to Supervisor communication) and the user invoking the Join job require “super user” permissions. If the Supervisor already has a configured NiagaraStation for this station, it is not used (may be left blank).

- **Default Subordinate User Password**  
This is an added property in AX-3.7u1 and later. The local (JACE) station user password to be used by the Supervisor to connect back to this station. Like the other “Default” properties in this dialog, it may be left blank if the Supervisor already has a configured NiagaraStation for this station.  
*Note:* You must configure the Default Subordinate User Password when pre-engineering a subordinate station to join to a Supervisor. Failure to configure the password will result in authentication errors when the Supervisor attempts to make a connection to this subordinate JACE (that the Supervisor is not already “aware” of).
- **Default Subordinate IP Address**  
This is the IP address of the local (JACE) hosting this station, used by the Supervisor for Fox client connection. If the Supervisor already has a configured NiagaraStation for this station, it is not used (may be left blank).
- **Default Subordinate Port**  
This is the local (JACE) software port to be used by the Supervisor for Fox client connection—where the default variable %FoxService.port% tracks the currently used Fox port. If the Supervisor already has a configured NiagaraStation for this station, it is not used.
- **Default Subordinate Use Fox SSL**  
(New starting in AX-3.7, by default *false*) Whether or not the Supervisor should use Fox SSL (Foxs) client connection to the local JACE for Join operations. Set this to true *only* if the local JACE is properly configured for certificate-based SSL communications, with Foxs enabled in its FoxService. The Supervisor is also typically configured for SSL communications, and both platforms require a valid certificate which is signed by a trusted CA whose root certificate is located in the reciprocal station's trust store.
- **Default Subordinate Fox SSL Port**  
(New starting in AX-3.7) Applies *only* if “Default Subordinate Use Fox SSL” is true. This is the local (JACE) software port to be used by the Supervisor for Foxs client connection—where the default variable %FoxService.foxsPort% tracks the currently used Foxs port. If the Supervisor already has a configured NiagaraStation for this station (using Foxs), it is not used.
- **Station File**  
Typically left at defaults, this is the local file ord to the “profile.bog file” that acts as the folder/container “template” used on the NiagaraStation that represents this station, as it will be exported to the Supervisor station. The default value is:  
`file:^joinProfiles$(defaultStationFileName).bog`  
For related details, see “[About the niagaraStation profile.bog file](#)” on page 2-18.  
*Note:* If you edit this ord after the **JoinProfile** has already been enabled, you will need to disable and then re-enable the **JoinProfile** for a new profile.bog file to be created—otherwise, Join errors may result. Be aware that this does not delete (or rename) the original profile.bog file.

In AX-3.7u1 and later, when you are pre-engineering a JACE for eventual Join to a Supervisor, (where you want to do as much engineering as possible in the JACE station or if you do not have access to the Supervisor station), there are properties that you need to configure. Specifically, the default subordinate user credentials in the JoinProfile must be configured in order to avoid authentication errors during the Join process.

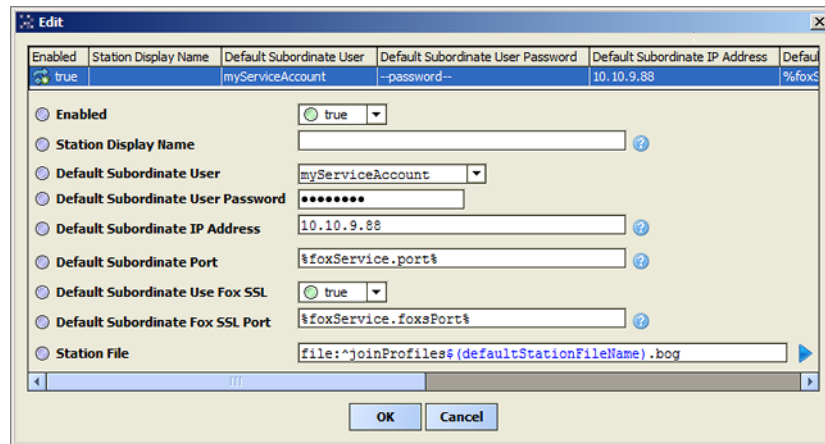
### Adding default subordinate user credentials to a pre-engineered NiagaraStation

- Step 1 Under the subordinate JACE station's NiagaraNetwork, double-click the SubordinateExportTagNetworkExt to open the **Join Profile Manager** view.
- Step 2 Select the JoinProfile for the Supervisor, and then click the **Edit** button for a popup dialog.
- Step 3 Enter a **Default Subordinate User** name (something that the Supervisor will recognize).  
*Note:* Typically, the Default Subordinate User account should be a user that was created especially for station-to-station access, typically with admin write privileges, and not otherwise used for normal (login) access of the station.
- Step 4 Enter the **Default Subordinate User Password** for that user.



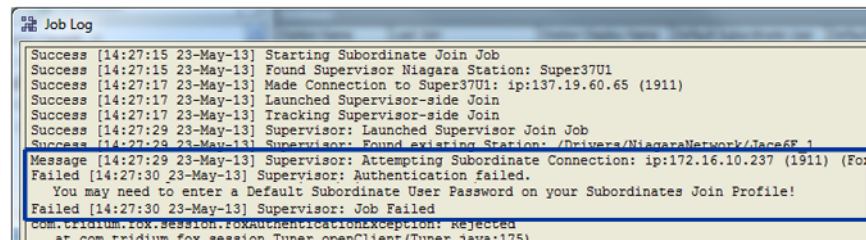
Step 5 Click **OK** after making changes, as shown here:

**Figure 2-14** Adding local station user credentials for Supervisor to connect back to this station



After adding these properties you can issue a Join from the JACE station. The new NiagaraStation (representing that JACE) is added, where it has all the pre-engineered PxViews, points, etc from its export tags. If an incorrect user name or password (or no password) is used, the Join will fail with authentication errors. Refer to the Job Log to confirm, as shown here:

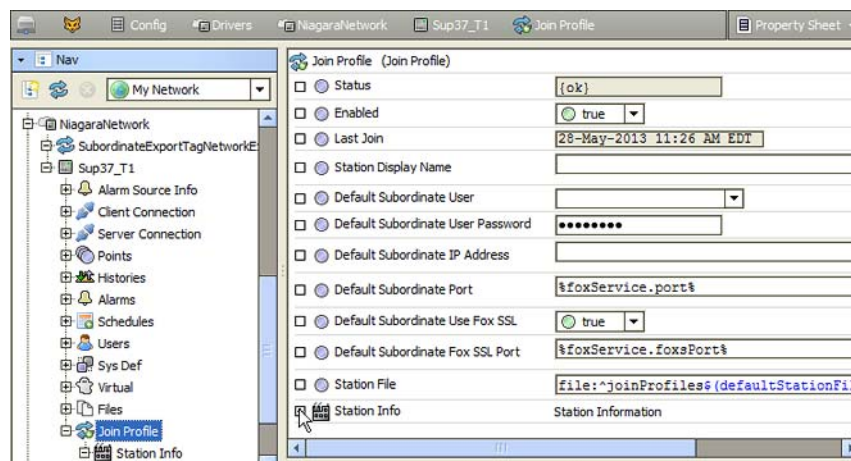
**Figure 2-15** Job Log shows Supervisor Authentication Failed error message



### About Station Info

The **JoinProfile** has a frozen container slot **Station Info** (Station Information), which is empty by default, as shown in [Figure 2-16](#).

**Figure 2-16** Empty (default) StationInfo container under JoinProfile

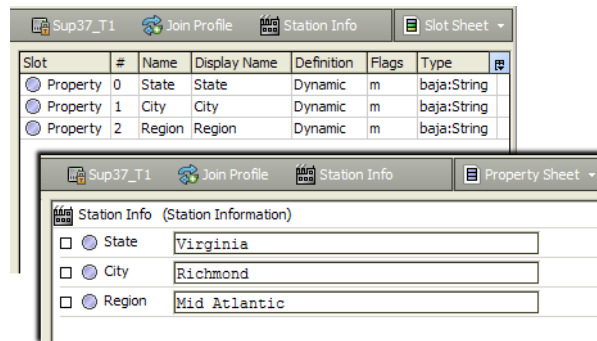


Station Info allows you to add string-value metadata about the subordinate (JACE) station, where it can be used in different ways on export to the Supervisor station. Examples of possible Station Info properties are "State", "City", "StoreNumber", "ZipCode", and so on.

### Adding Station Info

- Step 1 Add properties using the slot sheet (right-click **Station Info** and select **Views > Slot Sheet**).
- Step 2 Right-click in the slot sheet view and select **Add Slot**. The **Add Slot** popup dialog appears.
- Step 3 In the **Add Slot** dialog, type in the property *name* in the **Name** field, for example, *State* or *City*. This will be the default descriptor for this property on the property sheet.
- Step 4 Make sure that **Type** is: *baja, String*.  
Click (check) the **Metadata** flag.
- Step 5 Click **OK** to add and return to the slot sheet.
- Step 6 As needed, repeat steps 2 through 5 to add more Station Info properties.
- Step 7 When finished adding properties, you can assign them string values.  
Double-click **Station Info** for its property sheet.
- Step 8 In the **Station Info** property sheet, type in the property values, and click **Save**.  
Station Info values can now be used in export tag functions.

**Figure 2-17** Example Station Info properties (slot sheet, property sheet)



### Using Station Info

Station Info can be used in various ways, including:

- Automatic, logical “NiagaraStationFolder” creation and organization on the Supervisor station, for exported stations (NiagaraStations). This is particularly useful for extremely large systems, with many JACE stations.  
Do this by incorporating Station Info in the **SupervisorExportTagNetworkExt** property “**Station Folder Path**”. For related details, see property descriptions in the section “[About the SupervisorExportTagNetworkExt](#)” on page 2-7.
- Availability on the Supervisor station for BQL queries of NiagaraStations based upon this metadata.
- Incorporation in the station’s “display name” on the Supervisor, if desired. See the Note in the “Station Display Name” property description in the section “[Editing the Join Profile](#)” on page 2-13.

**Note:** Following an export tag Join, the corresponding **NiagaraStation** on the Supervisor station will also have a “**Station Info**” device extension, containing the same properties and values added in the **Station Info** container under the **JoinProfile** on the JACE station.

Consider the Supervisor copy a “slave” version—although you can modify or add to it, changes are not filtered back down to the subordinate (JACE) station. For this reason, it is recommended that you make all changes in the **JoinProfile**’s **Station Info** on the JACE station.

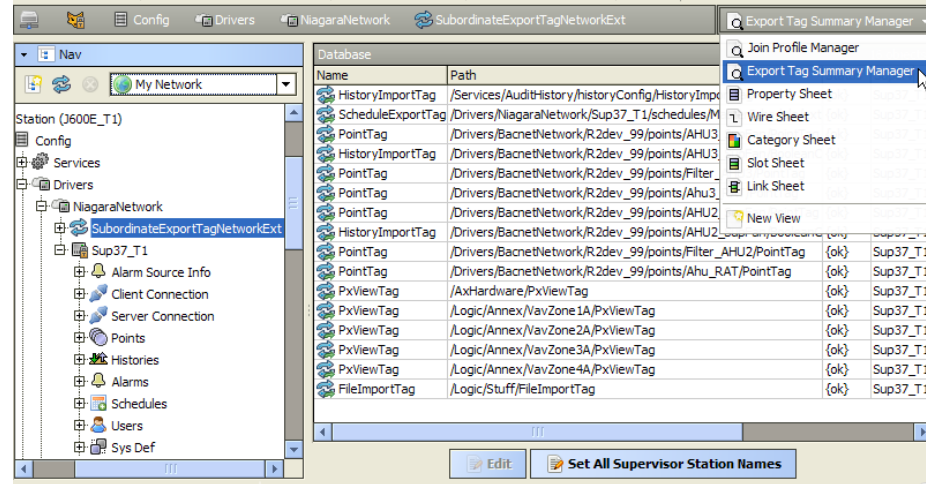
Also, note that the Supervisor’s copy of **Station Info** is used in the Join process. Therefore, if changes are needed, it is recommended you make changes to the **Station Info** of the **JoinProfile** before the next Join.

### About the Export Tag Summary Manager

The **Export Tag Summary Manager** is a secondary view of the **SubordinateExportTagNetworkExt** in a subordinate (JACE) station. To access this view, right-click the extension and select it from the **Views** menu, or use the view selector to access it, as shown in [Figure 2-18](#).



**Figure 2-18** Export Tag Summary Manager of SubordinateExportTagNetworkExt in JACE station



This view lists *all* export tags currently in the station, with columns providing information on station path as well as Supervisor station slot path, among other things. You can simply double-click any export tag to go directly to its property sheet. There is also an **Edit** button for making changes to one or more selected export tags, providing access to a few common properties that include “Enable”.

If needed, use this view to selectively “disable” one or more export tags—upon the next Join, this effectively *removes* corresponding items from the Supervisor. To add these items back, simply re-enable the export tags and issue another Join.

Default columns displayed in the **Export Tag Summary Manager** include the following:

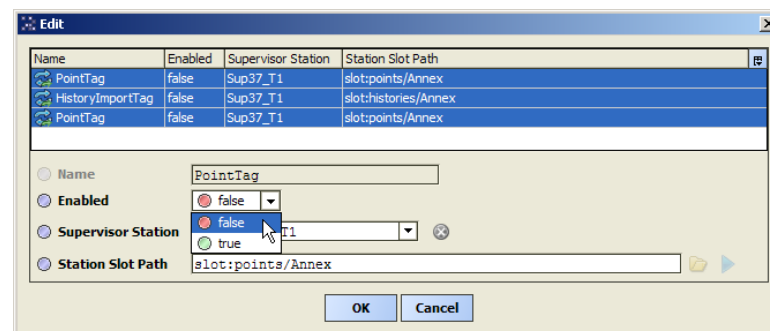
- **Name**  
The name of the export tag.
- **Path**  
Component path of the export tag in the local station, relative to the station root.
- **Status**  
Status of export tag, either “ok” (enabled) or “disabled”.
- **Supervisor Station**  
Name of the export tag’s currently selected Supervisor station (NiagaraStation).
- **Station Slot Path**  
Parent slot name for the exported object on the Supervisor, as configured in the export tag. Component slots are relative to the NiagaraStation component representing this station on the Supervisor.

See the next section “[Editing export tags in the Summary Manager](#)” for more Edit information. Note another button lets you globally set the “Supervisor Station” property value for *all export tags*. See “[Specifying the Supervisor station globally in all export tags](#)” on page 2-18.


### Editing export tags in the Summary Manager

In the **Export Tag Summary Manager**, select one or more export tags, and then click the **Edit** button for a popup dialog, as shown in [Figure 2-19](#).

**Figure 2-19** Edit dialog for export tags in Export Tag Summary Manager




The following fields are in this dialog:

- **Name**  
Of selected export tag, editable only if a single tag selected. If multiple tags are selected, it reflects the first export tag’s name.
- **Enabled**  
Enabled property of the selected export tag(s), which can be toggled between true (default) and false. (Possibly the most commonly edited export tag property in this view.)
- **Supervisor Station**  
Name of the currently selected Supervisor station (NiagaraStation) in the selected export tag(s).
- **Station Slot Path**  
Parent slot name for the exported object on the Supervisor, as configured in the export tag.
  - If multiple tags are selected, you can batch edit the station slot path.  
*Note:* Batch editing a number of tags for slot:points/AHU1 would only be useful for proxy points being tagged under the AHU1 device in the JACE. However, you could batch edit the tags for all proxy points in the JACE using something like slot:points/%networkFolderPath%.
  - If a single export tag is selected, you can click on  open folder, for the **Select Parent Target** dialog, to select a parent container as saved in the profile.bog file.

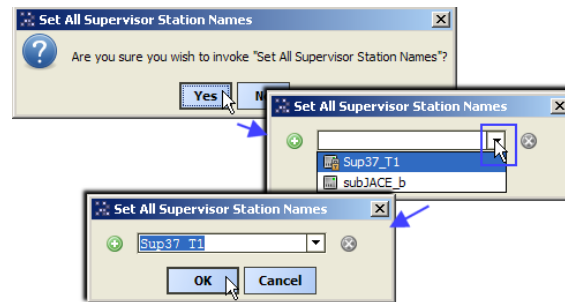
Click **OK** after making any changes in the **Edit** dialog of the selected export tag(s). Note a subsequent Join is necessary for any changes to occur in the Supervisor station.


### Specifying the Supervisor station globally in all export tags

The [Export Tag Summary Manager](#) provides a button at the bottom of the view that you can use to globally assign the “Supervisor Station” property value in *all export tags* in the station.

When you click the  **Set All Supervisor Station Names** button at the bottom of the view, a confirmation dialog appears ([Figure 2-20](#)).

**Figure 2-20** Confirmation dialog and subsequent select Supervisor station dialog



As shown above, after confirmation (click **Yes**) another dialog appears. Click the  drop-down control for NiagaraStations in the station’s NiagaraNetwork—*select the one for the Supervisor*. When you click **OK**, this globally sets the “Supervisor Station” property value in all export tags in the station.

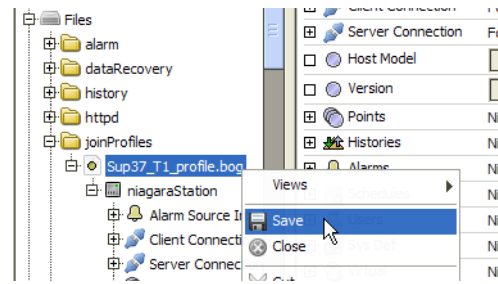
*Note:* This command is also a right-click action on the **SubordinateExportTagNetworkExt (Set All Supervisor Station Names)**, which produces the same dialogs as shown in [Figure 2-20](#) above.

## About the niagaraStation profile.bog file

When you enable the **JoinProfile** (of the NiagaraStation for the Supervisor) on the JACE station, a new folder with a station profile.bog file is automatically created in the JACE station’s file space. You routinely edit *and save* this bog file, as it can determine property values of the NiagaraStation (that represents this JACE) on the Supervisor. These values are written to the Supervisor upon any export tag Join.

*Note:* Remember to save the bog file after any changes. Right-click the bog file and select **Save**. See [Figure 2-21](#).

**Figure 2-21** Saving the profile.bog file



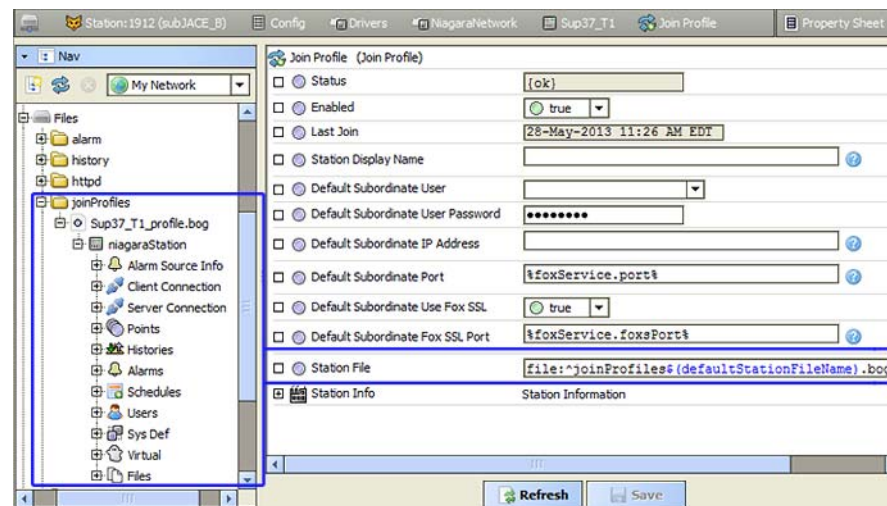
Note the profile.bog file is important for non-default *properties* of the “niagaraStation”, including *also* its “device extensions”, for example the **Alarms** extension and **Users** extension. For related details, see “Device extension properties importance in profile.bog file” on page 2-19.

**Note:** You can also add new folders, point folders, and archive folders (containers) in the niagaraStation in this profile.bog file, saving it as you work. This allows you to select these items as the parent “Station Slot Path” property when adding various export tags.

However, doing this is optional—alternatively, you can type a folder path directly in the “Station Slot Path” property of an export tag, or use special “BFormat” variables instead. For related information, see “Station Slot Path considerations” on page 2-21 and “BFormat options in Station Slot Paths” on page 2-22.

Figure 2-22 shows the relationship of the JoinProfile’s “Station File” property and the resulting file folder and profile.bog file created, using property defaults.

**Figure 2-22** Profile.bog file created in file space of JACE station



The default ord of the Station File property in the JoinProfile is:

```
file:^joinProfiles$(defaultStationFileName).bog
```

where the \$(defaultStationFileName) portion is an ord variable that results in a .bog file named: SupervisorStationName\_profile.bog

In the Figure 2-22 example, where the Supervisor station (and NiagaraStation that represents it) is named “Sup37\_T1”, this results in a file named: Sup37\_T1\_profile.bog

**Note:** If you edit the Station File property after the JoinProfile has already been enabled, you need to disable and then re-enable the JoinProfile for a new profile.bog file to be created—otherwise, Join errors may result. Be aware that this does not delete (or rename) the original profile.bog file.

### Device extension properties importance in profile.bog file

The niagaraStation in the “profile.bog file” on a subordinate JACE has significance in *property values* of the station, for example the “Virtuals Enabled” property, or “Alarm Source Info” properties, as well as properties in various “device extensions” (for example, the **Alarms** and **Users** extension).

Example of device extension properties are the Alarms extension's property "Alarm Class", and the Users extension's property "Sync Out Enabled" (used in "network users"). Default values for these two properties are "Default Alarm Class" and "false", respectively. However, it may be that non-default values are wanted for these two properties in the Supervisor's NiagaraStation (that models this JACE).

If working directly in the Supervisor station, you *can* set these property to non-default values. However, note that those values will be *overwritten* upon the next Join of this JACE (unless of course they have the same values in the profile.bog file in that JACE).

**Note:** This can explain why after configuring "NiagaraStation properties" in the Supervisor station, that property values in that NiagaraStation and/or in its device extensions are later seen to "revert to default values". Likely this happened following a Join, where that JACE's profile.bog file still has default property values.

Therefore, you should review property values in the *profile.bog file's* niagaraStation and its various device extensions, and *change* any ones that need to vary from default values. Again, remember to **save** the profile.bog file after making any changes.

Additionally, in cases where you are pre-engineering a JACE for eventual Join to a Supervisor, and want to do as much engineering as possible in the JACE station (possibly even lacking access to the Supervisor station), there are possible "property additions" that you may need to perform in the profile.bog file. See "Provisioning slot additions in a profile.bog file" on page 2-20.

### Provisioning slot additions in a profile.bog file

**Note:** This applies if the Supervisor is configured for provisioning its NiagaraNetwork (typical).

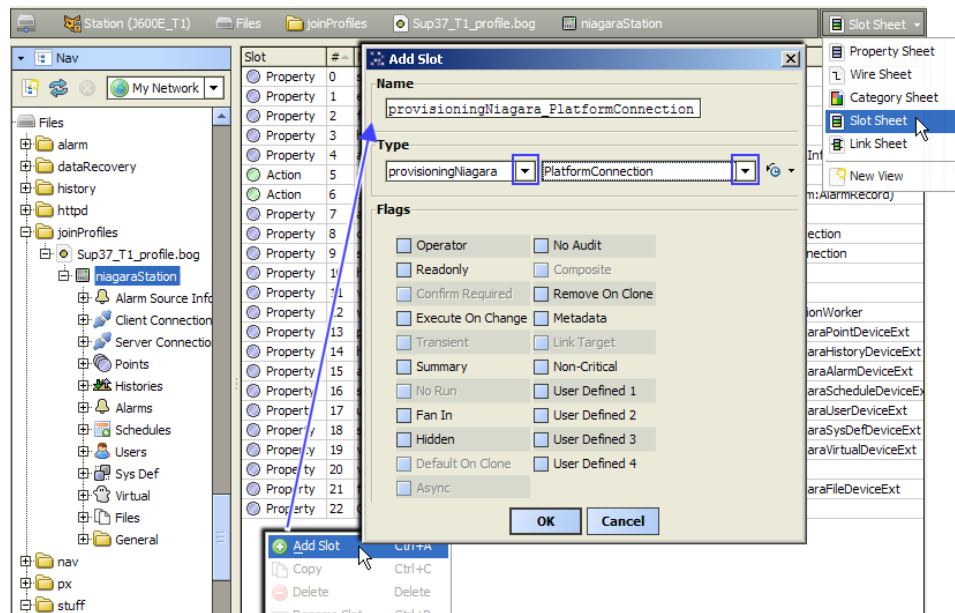
The niagaraStation in the "profile.bog file" on a subordinate JACE can represent both the *folder structure* and "properties/device extension" configuration that the Supervisor uses to model this JACE station. You can (and typically should) perform a lot of pre-engineering on the JACE station, working in its profile.bog file, even before the JACE is attached to the same network (thus before any Join is even possible).

If you want to minimize subsequent Supervisor station engineering required to fully assimilate the JACE, and have it instead Join "as fully configured as possible", there is a missing child container slot that you *should possibly add* under the niagaraStation component in the station profile.bog file on the JACE.

If you add (and configure) this slot and properties prior to the first Join for the JACE, it will minimize the need for subsequent configuration in the Supervisor.

### To add provisioning properties in the station profile.bog file

- Step 1 Open the JACE's platform bog file in its File space.
- Step 2 Go to the **slot sheet** for the **niagaraStation** node.  
(right-click the **niagaraStation**, and select **Views > Slot Sheet** from the popup menu)
- Step 3 In the bottom of the slot sheet view, right-click and select **Add Slot**.

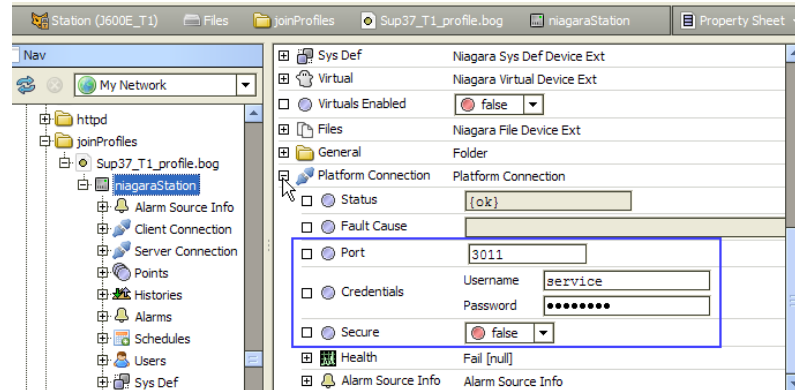


- Step 4 In the popup **Add Slot** dialog (as shown above) enter the following:
  - **Name:** provisioningNiagara\_PlatformConnection (type in that exact string)

- **Type** (select first): provisioningNiagara (select second): PlatformConnection

And click **OK**. A **Platform Connection** device extension is added to the profile.bog file.

- Step 5 Go to the **slot sheet** of this new **Platform Connection** extension.
- Step 6 In the slot sheet, right-click the “secure” slot and select **Config Flags** from the popup menu. Clear (*uncheck*) the **Hidden** flag.
- Step 7 Next go to the **property sheet** and expand the **Platform Connection**.



In the property sheet, enter the necessary values for the Supervisor to make a platform connection to this JACE, including:

- **Port** — 3011 (default) is typical for a “regular” (non-SSL) platform connection, whereas 5011 is typical for a secure (platform SSL) connection.
- **Credentials** — Enter the platform username and password. These are the same credentials used when making a Workbench platform connection to this JACE.
- **Secure** — *false* (default) specifies a “regular” (non-SSL) platform connection—this is appropriate for any JACE-2 or -4/5 series host, or any JACE-6 or JACE-7 that is *not* configured for platform SSL. Otherwise, if the JACE is configured to support platform SSL only, this must be set to *true*.

- Step 8 Save the profile.bog file (right-click the *SupervisorName\_profile.bog* file and select **Save** ). By making these configuration changes to the profile.bog file in the JACE’s file space, the client proxy in the Supervisor will automatically be configured to use the appropriate platform connection parameters for provisioning—including (if so configured) for a platform SSL connection.

## Station Slot Path considerations

The “Station Slot Path” property in some export tag types can accept a manually-typed path. In other words, if you type in a slot path value in one of these export tags, upon a Join if the corresponding containers (folders, archive folders, point folders) do not exist, they are *dynamically created* in the Supervisor station. This occurs regardless of whether they exist (or not) in the JACE station’s “profile.bog” file.

**Note:** *This differs from the initial AX-3.5 implementation of export tags, where a “Station Slot Path” property had to be reflected in the JACE’s “profile.bog” file—otherwise, the Join failed to export that item.*

This “dynamic folder” creation results vary, depending on export tag type.

- **PxViewTag:** creates ordinary folder (BFolder) containers, by default.
- **PointTag:** creates Niagara Point Folder (PointFolder) containers.
- **HistoryImportTag** and **SystemHistoryImportTag:** creates History Archive (HistoryArchive) containers.
- **ComponentTag:** creates ordinary folder (BFolder) containers.

In the case of the PxViewTag and ComponentTag (which create ordinary folders), you are free to specify any slot path in the Station Slot Path property, starting with `slot:`

For example `slot:Bldg_A/Floor_1`

Create two levels of folders directly under the NiagaraStation that represents this JACE station (assuming these folders do not already exist).

In the case of the PointTag and HistoryImportTag or SystemHistoryImportTag, you must retain the leading NiagaraStation “device extension name” (`points` or `histories`), for things to work correctly.

- For example, for a PointTag: `slot:points/Bldg_A/Floor_1`



Upon a Join creates a Niagara proxy point (under the **Points** extension of the NiagaraStation representing this JACE) in the second of two levels of *Niagara point folders*.

- Or for a HistoryImportTag or SystemHistoryImportTag: slot:histories/Bldg\_A/Floor\_1  
 Upon a Join creates a Niagara history import descriptor (under the **Histories** extension of the NiagaraStation representing this JACE) in the second of two levels of history *archive folders*.

You can also use special “BFormat” values within manually-typed Station Slot Paths. See “[BFormat options in Station Slot Paths](#)”.

### BFormat options in Station Slot Paths

In addition to “absolute” slot path values accepted in “Station Slot Path” properties of export tags, you can also use a few special variables based on Baja Format (BFormat) scripting, as follows:

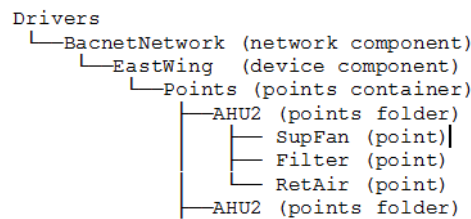
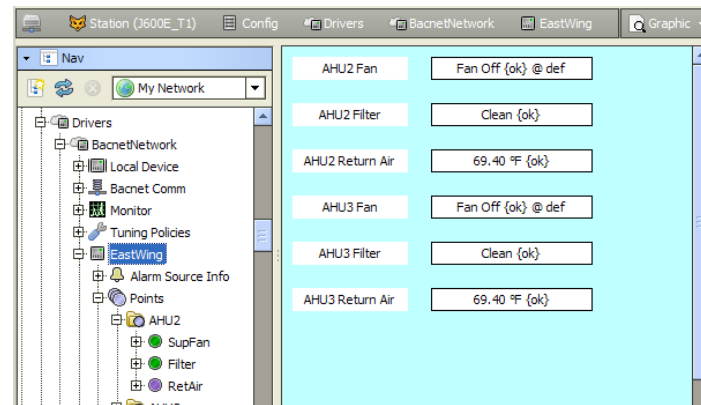
- %networkFolderPath%
- %deviceFolderPath%
- %parent.name%

#### %networkFolderPath%

This replicates container structure in the JACE station from the “network downwards”, that is, *below* the network (but *including* the device). Using the following station slot path value (for both PxViewTags and PointTags) can provide consistent results.

```
slot:points/%networkFolderPath%
```

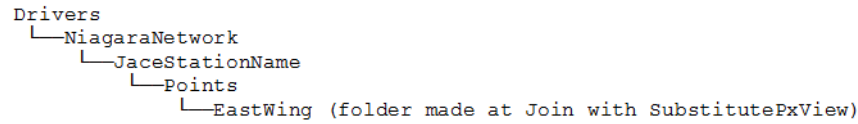
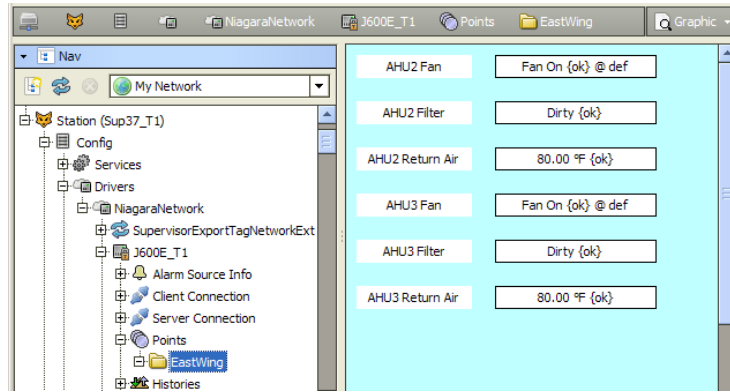
For example, consider a JACE station with a BacnetNetwork that has an architecture like this:



The JACE station has a PxView on the device component “EastWing”—a Px view that you are exporting. If you add a PxViewTag on EastWing, and set its slot path to:

```
slot:points/%networkFolderPath%
```

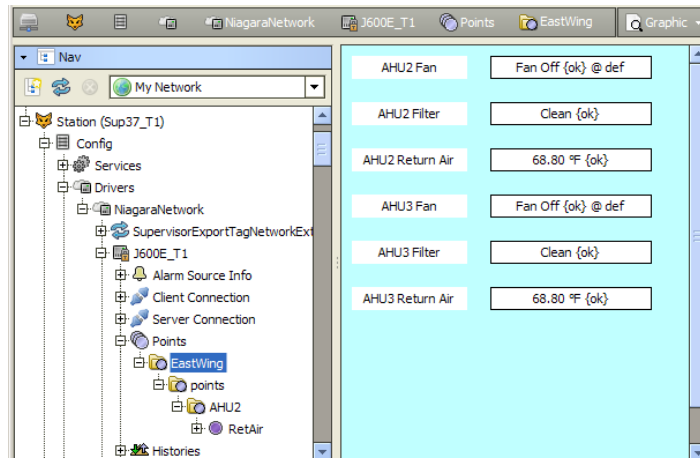
Upon a Join, this results in the following folder structure under the Supervisor’s NiagaraNetwork:



Assuming this slot path did not already exist because of other *PointTags*, or by previous manual definition via the “profile.bog” file in the JACE, the EastWing component created in the Supervisor is a standard “folder” component. As a result, it is not visible in the Supervisor’s **Niagara Point Manager** view.

To *improve this*, it is recommended to *use a PointTag on at least one proxy point* of the device with the PxViewTag. In this case, the EastWing folder (above) is *deleted* in the Supervisor station.

Then in the JACE station, a PointTag is added to the “RetAir” point under the “AHU2” points folder, with its “Station Slot Path” property configured as: `slot:points/%networkFolderPath%`



Following another Join, because of the PointTag, all folders used to replicate the container structure are *Niagara Point Folders*, including the “EastWing” container with the SubstitutePxView, as shown above. Everything is now visible in the **Niagara Point Manager** view. Note this also allows you to *manually* add more proxy points at a later time, if ever desired.

If *all* PxViewTags and PointTags use a Station Slot Path “`slot:points/%networkFolderPath%`” value, this can keep everything exported to the Supervisor in the same (or at least similar) hierarchy as it is in the JACE station. It also can save export tag editing time when duplicating a “template” device to create all of the devices in the station.

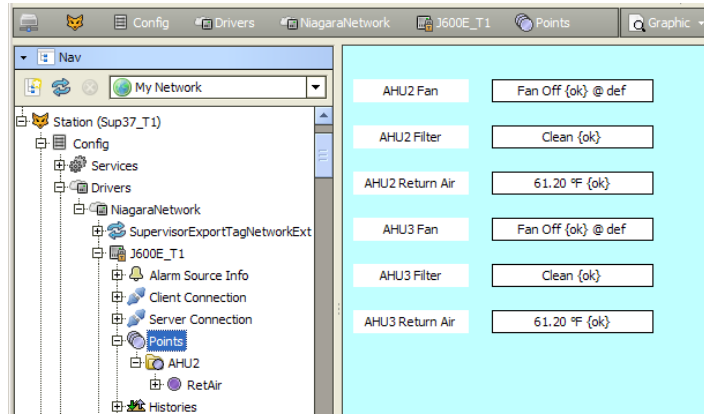
### %deviceFolderPath%

This replicates container structure in the JACE station from the “device downwards”, that is, *below* the device component (but *not including the device*). For example, with the same station example described in the previous “`%networkFolderPath%`” section, if you added the following station slot path to both the PxViewTag on the BACnet “EastWing” device and also on a PointTag for one of its proxy points:

```
slot:points/%deviceFolderPath%
```



This results in the following folder structure under the Supervisor's NiagaraNetwork:



```

Drivers
├── NiagaraNetwork
│   └── JaceStationName
│       └── Points (points folder has SubstitutePxView, instead of device)
│           └── AHU2 (points folder with proxy point)
    
```

This is typically *less desirable* than using the “%networkFolderPath%” variable appended to “slot:points/”, because all device-level Px views get “stacked” onto the NiagaraStation’s **Points** device extension. However, if devices are engineered in the JACE using many point folders, perhaps with PxViews on those point folders (instead of on device components), this may be a useful alternative.

See “%networkFolderPath%” on page 2-22 for a more universal example.

**%parent.name%**

This BFormat entry applies more to PxViewTag usage outside of any “points context”, as it creates a folder named the same as the parent of the export tag. For example, if the JACE station has a folder “Setpoints” with a PxViewTag added and configured with the following Station Slot Path:

```

slot:%parent.name%
or
slot:%parent.name%/this/that
    
```

where *this* and *that* are manually-entered slot paths, this is equivalent to entering

```

slot:Setpoints
or
slot:Setpoints/this/that
    
```

depending on the structure of the system. In this case, the “Setpoints” folder (with SubstitutePxView) will be created in the root of the NiagaraStation component that represents this JACE station.

**Conclusions**

Experimentation is recommended on one or two export tags to verify expected results, especially if using the BFormat options. However, dynamic folder creation is expected to help simplify the export tag process, as many found the station “profile.bog” process to be too time intensive.

## About PxViewTags

The **PxViewTag** is potentially the most important of all export tags, at least in cases where existing PxViews on subordinate (JACE) stations are desired on the Supervisor station. The PxViewTag simplifies this, for the most part, as a “drag and drop” function. Related additional Supervisor background operations such as creating Niagara virtual points, importing referenced image files, and so on, are all handled automatically.

**Note:** See “*To add PxViewTags*” on page 1-6 for a “quick start” procedure on working with these tags. The remainder of this section explains more detailed aspects of working with this export tag type.

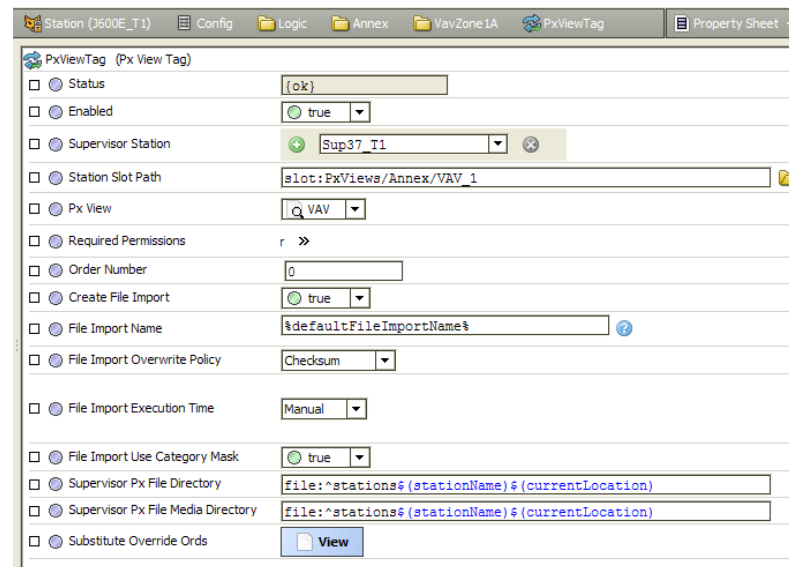
The following sections provide more details on PxViewTags:

- [PxViewTag properties](#)
- [PxViewTag action](#)
- [About the Supervisor SubstitutePxView](#)
- [About ord substitution overrides](#)

### PxViewTag properties

Double-click a PxViewTag for its property sheet (Figure 2-23).

**Figure 2-23** PxViewTag property sheet example



PxViewTag properties are described as follows:

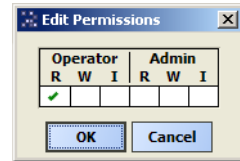
- **Status**  
Read-only status of the export tag, typically “ok”, unless “disabled” (Enabled=false).
- **Enabled**  
Whether the PxViewTag will attempt an export on the next Join (true, the default) or not (false).
- **Supervisor Station**  
Click the drop-down  control to select the NiagaraStation that represents the Supervisor (with enabled **JoinProfile**).  
**Note:** After selection, you must **Save** before you can use the folder (Select Parent Target) feature of the next property. Also, add (and delete ) controls exist to specify more than one Supervisor station; however, this configuration is rarely expected.
- **Station Slot Path**  
Parent slot path for the exported PxView on the Supervisor, relative to the NiagaraStation that represents this station. In the case of multiple Supervisor stations, this applies only to the first (top) one.
  - You can manually type a path, to create corresponding folders under the NiagaraStation (that represents this JACE) on the Supervisor. This includes use of “BFormat” `%value%` options. For example: `slot:MyViews/RTU_1` or `slot:points/%networkFolderPath%`  
**Note:** The default “slot:” puts the SubstitutePxView on the NiagaraStation component. See “Station Slot Path considerations” and “BFormat options in Station Slot Paths” on page 2-22.
  - Or, click the folder for a **Select Parent Target** dialog, and navigate to the desired folder or container. In this case, you must have already made (and saved) the needed folders in the

JACE's station "profile.bog" file.

**Note:** You can edit the "profile.bog file" to add folders—see "Verify and edit the Station (.bog File" on page 1-4 for details.

- **PxView**  
The name of the local PxView that will be exported to the Supervisor station, at the slot parent slot path above. Click the drop-down  control for a list of available PxViews.
- **Required Permissions**  
Required user permissions to access the exported PxView. Click the double-arrow  control for the **Edit Permissions** dialog (Figure 2-24). Default permissions are operator read (r).

**Figure 2-24** Edit Permissions dialog for PxViewTag



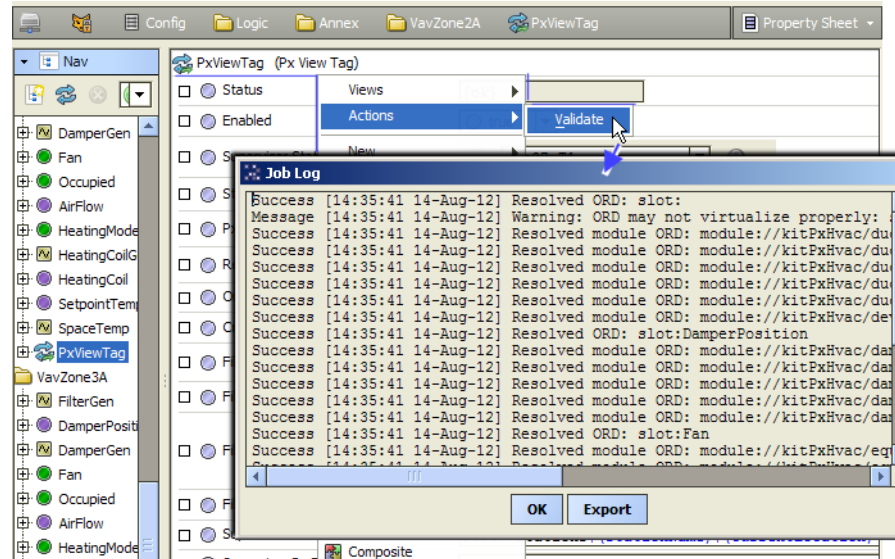
- **Order Number**  
Specifies the PxView priority order on the target parent slot in the Supervisor station, where 0 (default) is the lowest priority—any other positive integer value has a higher priority. This is useful if you use multiple PxViewTags to export to the same Supervisor parent. In this case, assign each PxViewTag a different order number. The "highest" ordered one will be the *default* view, and so on.
- **Create File Import**  
Boolean to specify if to automatically create a Niagara FileImport descriptor in the Supervisor to support referenced graphics; the default is `true`. Other file import-related properties follow.
- **File Import Name**  
Name of the Niagara FileImport descriptor to create on the Supervisor, if needed. The default variable `%defaultFileImportName%` creates an import descriptor named:  
`sourceParentName_sourcePxViewName_Import`  
For example, if a container named "AHU1" in the JACE has an exported PxView named "Graphic", the Niagara FileImport descriptor created on the Supervisor is named `AHU1_Graphic_Import`.
- **File Import Overwrite Policy**  
Standard property for Niagara FileImport descriptors, it determines the policy used to overwrite imported files on the Supervisor with changed source files on the JACE, when the file import executes. The default is `Checksum` (differing), with `Last Modified` as the other choice.
- **File Import Execution Time**  
Standard import execution time selections for Niagara FileImport descriptors, including `Manual` (default) or either `Daily` or `Interval`, where the latter two have various selection fields.
- **File Import Use Category Mask**  
Boolean to specify if `CategoryFilters` defined in the Supervisor should be applied to any file(s) imported by this PxViewTag. The default value is `true`.
- **Supervisor Px File Directory**  
Specifies the file ord to use in the Supervisor station's file space for copied px files. The default uses ord variables as: `file:^stations$(stationName)$(currentLocation)`, the latter of which duplicates the source (JACE) file organization under the `stations/stationName/` folder.
- **Supervisor Px File Media Directory**  
Specifies the file ord to use in the Supervisor station's file space for copied images files. The default uses ord variables as: `file:^stations$(stationName)$(currentLocation)`, the latter of which duplicates the source (JACE) file organization under the `stations/stationName/` folder.  
**Note:** Although a Supervisor typically has a vastly greater capacity to store image files (and px files, above) than does a JACE, in some cases you may wish to edit these two "Directory" properties to known file ords on the Supervisor, to prevent unnecessary duplication of the same files.
- **Substitute Override Ords**  
Provides a  **View** control and dialog to choose bound ords in the source Px file and override them with substitute or new ords. The typical application is to replace hyperlink ords that are no longer "in context" when viewing the dynamic (exported) PxView in the Supervisor station. For related details, see "About ord substitution overrides" on page 2-28.

### PxViewTag action

Unlike other export tags, the PxViewTag has a right-click action, **Validate**. Issuing this command runs a “Px View Tag Validation” job, with a resulting popup **Job Log** (Figure 2-25).

**Note:** Job log details are also available in the station’s **JobLogService**, for each Px View Tag Validation job.

Figure 2-25 Validate action for PxViewTag



Validation checks bound ords in the PxView for their ability to be “virtualized” (resolved) on the Supervisor station. In some cases, validation may fail upon one or more errors, and this job log will provide troubleshooting data that names the particular ord(s).

**Note:** A warning does not always mean that the PxView will fail to resolve on the Supervisor station. For example, using a relative ord of 'slot:' will generate a warning. However, in most cases that relative ord will render as expected.

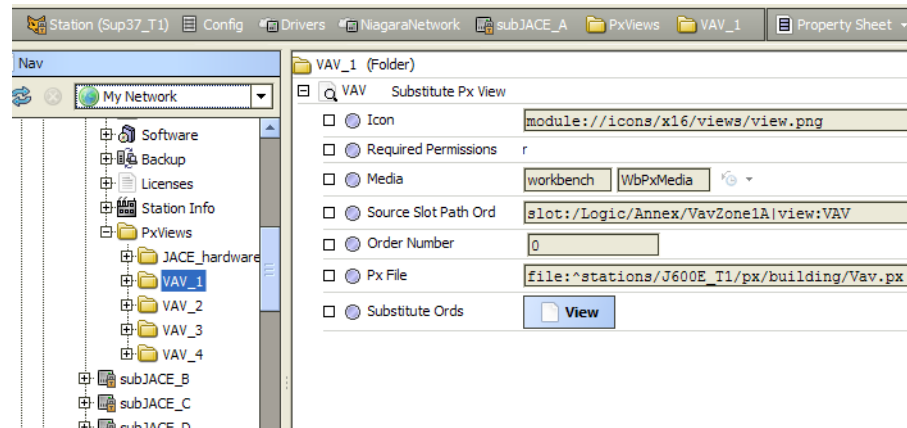
This feature helps you make any needed changes and revalidate *before* issuing a Join.

### About the Supervisor SubstitutePxView

Following a Join, the Supervisor station has a “SubstitutePxView” slot created for each PxViewTag, at the designated “Station Slot Path” under the NiagaraStation that represents that (subordinate) JACE.

Figure 2-26 shows the property sheet for an example SubstitutePxView.

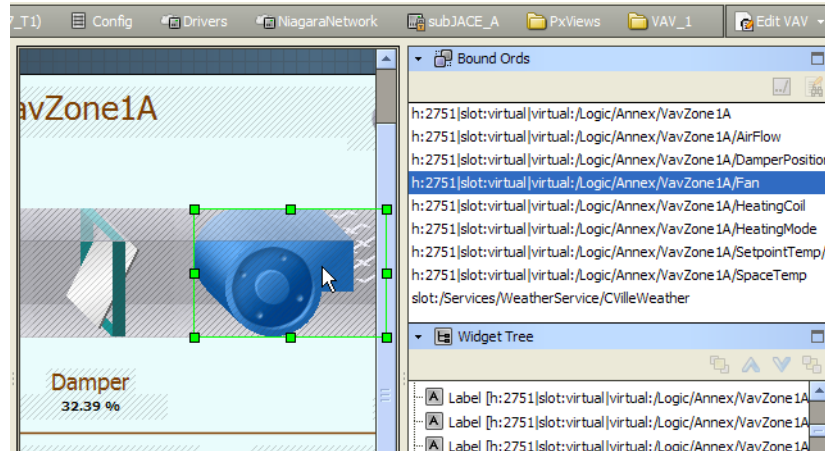
Figure 2-26 SubstitutePxView property sheet (in Supervisorstation)



Essentially, each one is a “virtualized” PxView, which automatically resolves, at runtime, the necessary items. Any source image files needed have already been copied via a Niagara FileImport descriptor.

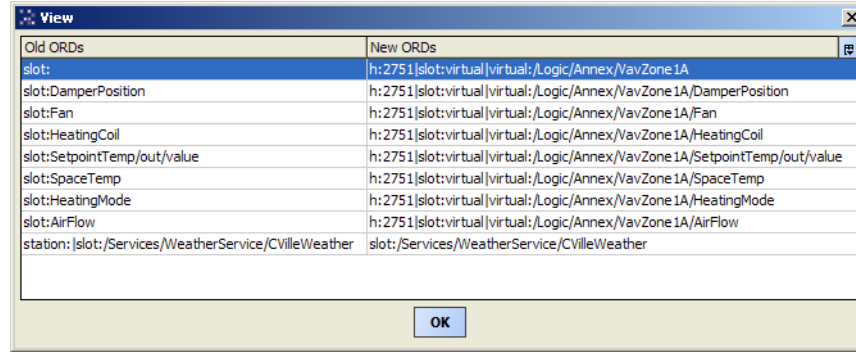
Other items include dynamically-created Niagara *virtual components* to fetch real-time values. Looking at a SubstitutePxView in the Px Editor, notice ords to these virtual components, as shown in Figure 2-27.

**Figure 2-27** Real-time values come from dynamically created Niagara virtual components



The Px XML (in the copied Px file) is actually unchanged from the original, as ord substitution happens at runtime. From the SubstitutePxView property sheet, use the  **View** control to see a read-only dialog comparing the original (old) ords to the substitute (new) ords (Figure 2-28).

**Figure 2-28** View dialog from View control in property sheet of SubstitutePxView (on Supervisor)

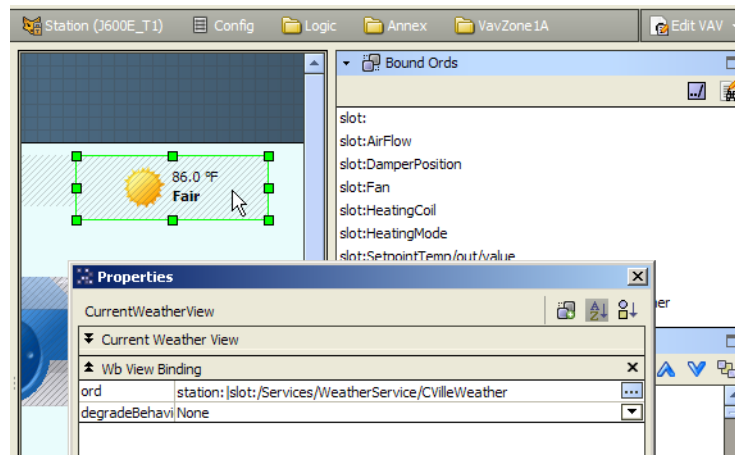


Starting in AX-3.7 (and as shown in Figure 2-28 above). Niagara “virtual points”, which are utilized in SubstitutePxView on a Supervisor, now have shortened “normalized” ord slot paths.

### About ord substitution overrides

Apart from ord substitutions to image files and real-time values, other ord changes may be necessary, typically “hyperlink” or bound ord types that may not virtualize correctly (or in context) at the Supervisor station. For example, consider a “Current Weather View” Px widget bound to a WeatherReport component in the JACE station (Figure 2-29).

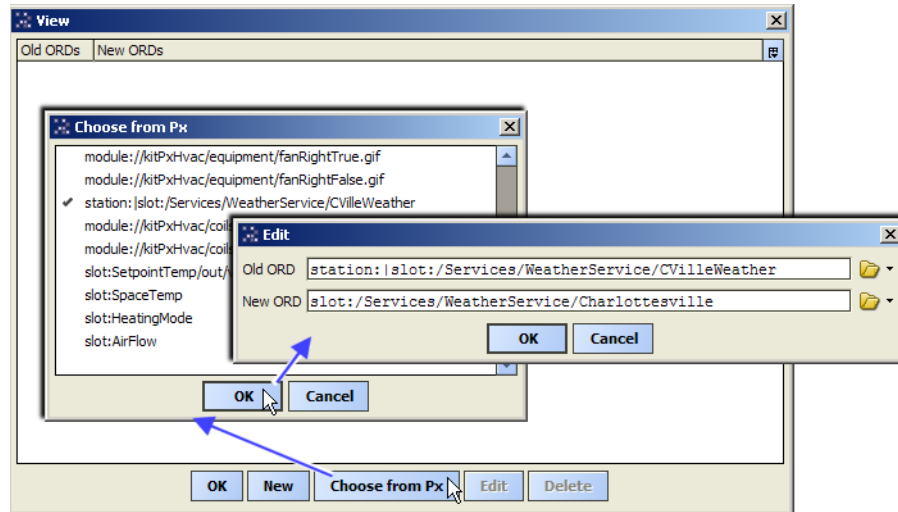
**Figure 2-29** Example bound ord in the original PxView in subordinate (JACE) station



In the [Figure 2-29](#) example, when viewing this on the Supervisor, instead of the original binding (which will not visualize correctly), the desired target is an equivalent WeatherReport component in the Supervisor station. The PxViewTag property sheet provides a  **View** control and dialog that you can use to *override* such ords to different values.

From the PxViewTag property sheet, click the  **View** control for a popup **View** dialog, then click the **Choose from Px** button for a **Choose from Px** dialog, listing ords you can  check one or more to override.

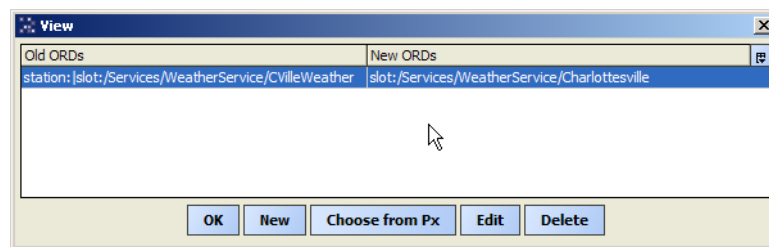
**Figure 2-30** View dialog, Choose from Px edit (from PxViewTag property sheet)



As shown in [Figure 2-30](#), a checked ord provides an **Edit** dialog showing the “Old ORD”, along with a “New ORD” field, in which you can manually type a new ord. This new ord should be relative to the Supervisor station (and not the currently open JACE).

**Note:** In this case, note that the  drop-down control to select the **Component Chooser**, **File Ord Chooser**, and so on, is relative to the opened JACE station—and likely not helpful in this case. After clicking **OK** to finish the override edit, the overridden ord appears in the View dialog ([Figure 2-27](#)).

**Figure 2-31** Example overridden ord in View dialog (from PxViewTag property sheet)



Such overrides are now part of this **PxViewTag** configuration, which you can re-edit to add, delete, or modify again before any Join operation.

### Ords that do not virtualize

In addition to ords with a hyperlink, these other types of ords can cause issues in PxViewTags:

- Ords that contain a BQL query — these cannot resolve correctly.
- Ords to Workbench views on components in a remote station, for example the Current Weather View example shown in [Figure 2-29](#) on page 28. Virtual components do not provide access to the Workbench views associated with the source components.



## About PointTags

The **PointTag** is useful when you need a permanent Niagara proxy point for an item in Supervisor station, say for “output links” into station control logic. Or, perhaps if the JACE is not serving PxViews and you wish to maximize “Supervisor configuration” via export tags before building PxViews. Otherwise, because of the automatically provided “Niagara virtual point” feature of PxViewTags, you may wish to limit the use of PointTags. Note that functionality was added to Niagara virtual points that makes them, in some ways, more powerful than proxy points. For related details, see “About Niagara virtual components” in the *Drivers Guide*.

**Note:** See “[Adding PointTags](#)” on page 1-7 for a “quick start” procedure on working with these tags. The remainder of this section explains more detailed aspects of working with this export tag type.

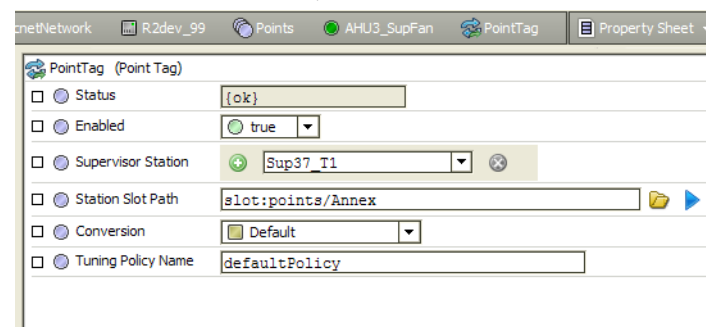
The following sections provide more details on PointTags:

- [PointTag properties](#)

### PointTag properties

Double-click a PointTag for its property sheet ([Figure 2-32](#)).

**Figure 2-32** PointTag property sheet example



PointTag properties are described as follows:

- **Status**  
Read-only status of the export tag, typically “ok”, unless “disabled” (Enabled=false).
- **Enabled**  
Whether the tag will attempt an export on the next Join (true, the default) or not (false).
- **Supervisor Station**  
Click the drop-down  control to select the NiagaraStation that represents the Supervisor (with enabled **JoinProfile**).  
**Note:** After selection, you must **Save** before you can use the folder (Select Parent Target) feature of the next property. Also, add (and delete ) controls exist to specify more than one Supervisor station; however, this configuration is rarely expected.
- **Station Slot Path**  
Parent slot path for the exported PxView on the Supervisor, relative to the NiagaraStation that represents this station. In the case of multiple Supervisor stations, this applies only to the first (top) one.  
**Note:** Station Slot Path must begin with `slot:points`
  - You can manually type a path, to create corresponding folders under the NiagaraStation (that represents this JACE) on the Supervisor. This includes use of “BFormat” `%value%` options. For example: `slot:points/RTU_1` or `slot:points/%networkFolderPath%`. See “[Station Slot Path considerations](#)” and “[BFormat options in Station Slot Paths](#)” on page 2-22.
  - Or, click the folder for a **Select Parent Target** dialog, and navigate to the desired folder or container. In this case, you must have already made (and saved) the needed folders in the JACE’s station “profile.bog” file.  
**Note:** You can edit the “profile.bog file” to add NiagaraPointFolders under Points—see “[Verify and edit the Station \(.bog\) File](#)” on page 1-4 for details. You should not select any parent folder other than under Points (property value must begin with `slot:points`).
- **Conversion**  
The standard Conversion property of a Niagara proxy point, with the default “Default” conversion selected (appropriate in most cases). Click the drop-down  control for a list of available conversion types. For related details, see “ProxyExt properties” in the *Drivers Guide*.



- **Tuning Policy Name**  
The standard Tuning Policy Name property for a Niagara proxy point, with the default “defaultPolicy” specified. To specify another tuning policy, you must know its *exact name* in the Supervisor.

## About HistoryImportTags

Add a **HistoryImportTag** to the history extension of a point or component to import/archive that history on the Supervisor. Upon a Join, it creates a Niagara history import descriptor in the Supervisor, under the Histories container of the NiagaraStation that represents this subordinate (JACE) station. A corresponding history is immediately available in the Supervisor.

**Note:** See “*To add HistoryImportTags*” on page 1-8 for a “quick start” procedure on working with these tags. The remainder of this section explains more detailed aspects of working with this export tag type.

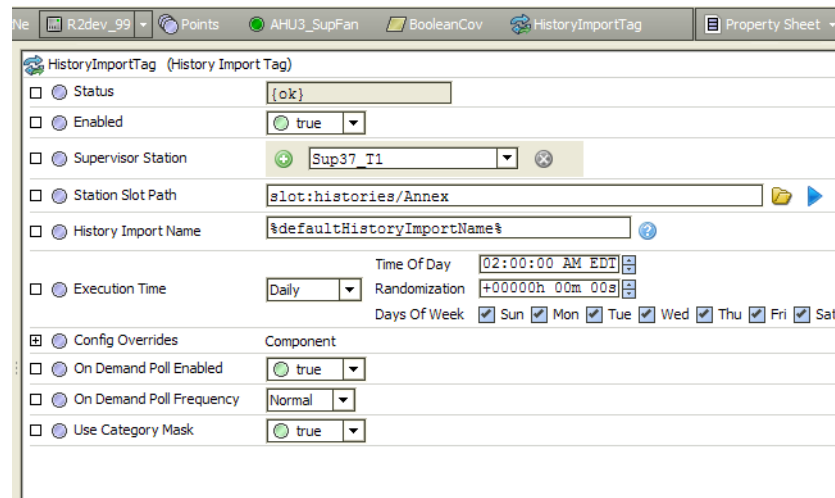
The following sections provide more details on HistoryImportTags:

- [HistoryImportTag properties](#)

### HistoryImportTag properties


Double-click a HistoryImportTag for its property sheet (Figure 2-33).

**Figure 2-33** HistoryImportTag property sheet example



HistoryImportTag properties are described as follows:

- **Status**  
Read-only status of the export tag, typically “ok”, unless “disabled” (Enabled=false).
- **Enabled**  
Whether the tag will attempt an export on the next Join (true, the default) or not (false).
- **Supervisor Station**  
Click the drop-down  control to select the NiagaraStation that represents the Supervisor (with enabled **JoinProfile**).  
**Note:** After selection, you must **Save** before you can use the folder (Select Parent Target) feature of the next property. Also, add (and delete ) controls exist to specify more than one Supervisor station; however, this configuration is rarely expected.
- **Station Slot Path**  
Parent slot path for the history import descriptor on the Supervisor, relative to the NiagaraStation that represents this station. If multiple Supervisor stations, this applies only to the first (top) one.  
**Note:** Station Slot Path must begin with `slot:histories`
  - You can accept the default “slot:histories” value, or else manually type a path, to create corresponding archive folders under the NiagaraStation (that represents this JACE) on the Supervisor. This includes use of “BFormat” %value% options. See “[Station Slot Path considerations](#)” and “[BFormat options in Station Slot Paths](#)” on page 2-22.
  - Or, click the folder for a **Select Parent Target** dialog, and navigate to the desired folder or container. In this case, you must have already made (and saved) the needed folders in the JACE’s station “profile.bog” file.

**Note:** You can edit the “profile.bog file” to add  NiagaraArchiveFolders under Histories—see “Verify and edit the Station (.bog) File” on page 1-4 for details. Do not select any parent folder other than under Histories (property value must begin with slot:histories). Such folders are used to organize Niagara history descriptors of any type (import, systemImport, export).

- **History Import Name**  
The name of the Niagara HistoryImport descriptor that this tag creates on the Supervisor, by default set using a format variable (%defaultHistoryImportName%). The history extension’s parent’s name is used in this default variable, with results similar to: “Remote\_pointName”.
- **Execution Time**  
The standard execution time selections for a Niagara HistoryImport descriptor, where the default is Daily at 2:00 AM, all days of the week. Other selections include Manual or Interval. For more details on this property and most remaining properties, see “History Import Edit” in the *Drivers Guide*.  
**Note:** This property, as well as “Config Overrides” (capacity and fullPolicy) properties and both “On Demand” properties, are all written to the Niagara HistoryImport descriptor upon a Join.
- **Config Overrides**  
Standard selections for a Niagara HistoryImport descriptor, containing settings for:
  - capacity — either Unlimited (default), or some finite number of records.
  - fullPolicy — either Roll (default), or Stop.
- **On Demand Poll Enabled**  
Standard selection for a Niagara HistoryImport descriptor. either true (default) or false. For related details on this (and next) property, see “On demand properties in history import descriptors” in the *Drivers Guide*.
- **On Demand Poll Frequency**  
Standard selection for a Niagara HistoryImport descriptor, either Normal (default), Slow, or Fast.
- **Use Category Mask**  
Boolean to specify whether the history imported by this tag will have any export tag category filters applied to it (default is true). For related details, see “About CategoryFilters” on page 2-8.

## About SystemHistoryImportTags

Add a **SystemHistoryImportTag** anywhere in the station to import histories on the Supervisor using the “System Tags” feature. Each tag creates a Niagara system history import descriptor in the Supervisor, under the Histories container of the NiagaraStation that represents this subordinate (JACE) station. Corresponding histories are immediately available in the Supervisor.

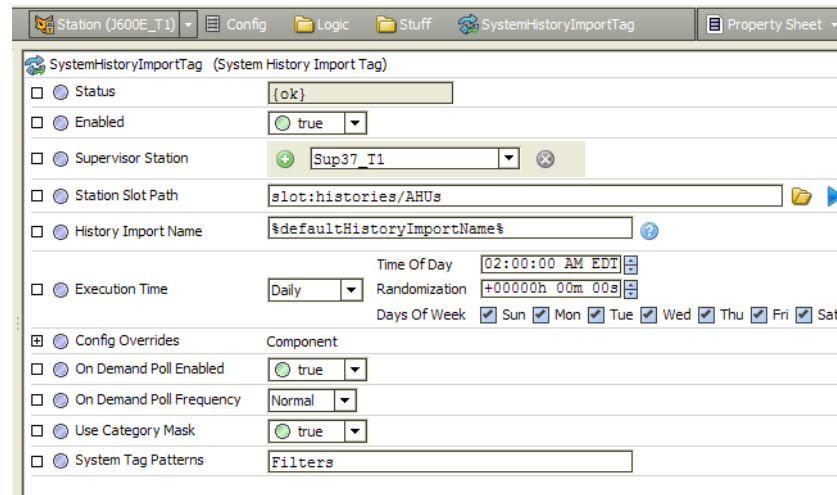
**Note:** See “To add SystemHistoryImportTags” on page 1-8 for a “quick start” procedure on working with these tags. The remainder of this section explains more detailed aspects of working with this export tag type. The following sections provide more details on SystemHistoryImportTags:

- [SystemHistoryImportTag properties](#)






### SystemHistoryImportTag properties

Double-click a SystemHistoryImportTag for its property sheet (Figure 2-34).

**Figure 2-34** SystemHistoryImportTag property sheet example



SystemHistoryImportTag properties are described as follows:

- **Status**  
Read-only status of the export tag, typically “ok”, unless “disabled” (Enabled=false).
- **Enabled**  
Whether the tag will attempt an export on the next Join (true, the default) or not (false).
- **Supervisor Station**  
Click the drop-down  control to select the NiagaraStation that represents the Supervisor (with enabled `JoinProfile`).  
*Note:* After selection, you must **Save** before you can use the  folder (Select Parent Target) feature of the next property. Also, add  (and delete ) controls exist to specify more than one Supervisor station; however, this configuration is rarely expected.
- **Station Slot Path**  
Parent slot path for the system history import descriptor on the Supervisor, relative to the NiagaraStation representing this station. If multiple Supervisor stations, this applies only to the first (top) one.  
*Note:* Station Slot Path must begin with `slot:histories`
  - You can accept the default “`slot:histories`” value, or else manually type a path, to create corresponding archive folders under the NiagaraStation (that represents this JACE) on the Supervisor. This includes use of “BFormat” `%value%` options. See “Station Slot Path considerations” and “BFormat options in Station Slot Paths” on page 2-22.
  - Or, click the  folder for a **Select Parent Target** dialog, and navigate to the desired folder or container. In this case, you must have already made (and saved) the needed folders in the JACE’s station “`profile.bog`” file.  
*Note:* You can edit the “`profile.bog` file” to add  `NiagaraArchiveFolders` under `Histories`—see “Verify and edit the Station (.bog) File” on page 1-4 for details. Do not select any parent folder other than under `Histories` (property value must begin with `slot:histories`). Such folders are used to organize Niagara history descriptors of any type (import, systemImport, export).
- **History Import Name**  
The name of the Niagara SystemHistoryImport descriptor that this tag creates on the Supervisor, by default set using a format variable (`%defaultHistoryImportName%`). For this export tag type, the default matches the name of the export tag itself. If desired, enter a different name here.
- **Execution Time**  
The standard execution time selections for a Niagara HistoryImport descriptor, where the default is `Daily` at 2:00 AM, all days of the week. Other selections include `Manual` or `Interval`. For more details on this property and most remaining properties, see “History Import Edit” in the *Drivers Guide*.  
*Note:* This property, as well as “Config Overrides” (capacity and fullPolicy) properties and both “On Demand” properties, are all written to the Niagara HistoryImport descriptor upon a Join.
- **Config Overrides**  
Standard selections for a Niagara HistoryImport descriptor, containing settings for:
  - `capacity` — either `Unlimited` (default), or some finite number of records.
  - `fullPolicy` — either `Roll` (default), or `Stop`.
- **On Demand Poll Enabled**  
Standard selection for a Niagara HistoryImport descriptor. either `true` (default) or `false`. For related details on this (and next) property, see “On demand properties in history import descriptors” in the *Drivers Guide*.
- **On Demand Poll Frequency**  
Standard selection for a Niagara HistoryImport descriptor, either `Normal` (default), `Slow`, or `Fast`.
- **Use Category Mask**  
Boolean to specify whether the history imported by this tag will have any export tag *category filters* applied to it (default is `true`).  
*Note:* This feature is future use only for **SystemHistoryExportTags**—currently, only histories exported using the `HistoryImportTag` can make use of the category mask feature.
- **System Tag Patterns**  
Text string(s) to search against the “System Tags” property value in the “History Config” slot of all history extensions in the subordinate (JACE) station. You can enter multiple strings, separating each with a semicolon (;), and also a “wildcard” (\*) character, for example: `RTU* ; *Temp ; Ltg`  
Upon a Join, matching histories will be imported at the Supervisor by an added Niagara SystemHistoryImport descriptor. For related details, see the section “Using System Tags to import Niagara histories” in the *Drivers Guide*.

## About ScheduleImportTags

Add a **ScheduleImportTag** to any existing schedule (BooleanSchedule, NumericSchedule, etc.) in the JACE station to have it, upon a Join, imported as a “slave” schedule on the Supervisor (along with a Niagara ScheduleImportExt descriptor). Effectively, the tagged schedule is now the “master schedule”.

Because this is “the reverse” of the typical Supervisor/subordinate schedule configuration (master schedule in Supervisor), usage of this export tag may be infrequent. Note this export tag also results in a Niagara ScheduleExportExt descriptor in this subordinate (JACE) station.

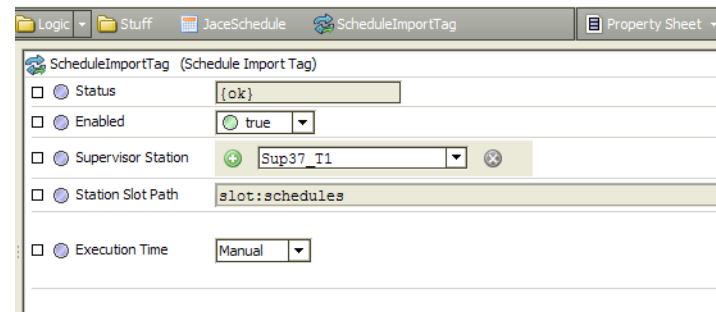
**Note:** See “Adding ScheduleImportTags” on page 1-9 for a “quick start” procedure on working with these tags. The remainder of this section provides more details on this export tag type.

- [ScheduleImportTag properties](#)

### ScheduleImportTag properties

Double-click a ScheduleImportTag for its property sheet (Figure 2-35).

**Figure 2-35** ScheduleImportTag property sheet example



ScheduleImportTag properties are described as follows:

- **Status**  
Read-only status of the export tag, typically “ok”, unless “disabled” (Enabled=false).
- **Enabled**  
Whether the tag will attempt an export on the next Join (true, the default) or not (false).
- **Supervisor Station**  
Click the drop-down  control to select the NiagaraStation that represents the Supervisor (with enabled **JoinProfile**).

**Note:** After selection, you must **Save** before you can use the folder (Select Parent Target) feature of the next property. Also, add (and delete ) controls exist to specify more than one Supervisor station; however, this configuration is rarely expected.

- **Station Slot Path**  
Parent slot path for the slave schedule on the Supervisor, read-only and fixed as the schedules slot (**Schedules** extension) of the NiagaraStation that represent this subordinate (JACE) station.
- **Execution Time**  
Execution time selections for the NiagaraScheduleImport descriptor that results on the Supervisor, where the default is **Manual**. Other selections include **Daily** or **Interval**, where each has a number of selectable parameters.  
Note that by default, the corresponding NiagaraScheduleExport descriptor created on the local subordinate (JACE) station has an Execution Time of **Interval**, at every 5 minutes.

## About ScheduleExportTags

Add a **ScheduleExportTag** as the child of an existing Niagara ScheduleImportExt under an existing schedule component, that is, a BooleanSchedule, NumericSchedule, etc. that resides in the **Schedules** extension of the NiagaraStation component that represents the Supervisor. Note this is the only valid parent for this export tag type. Ordinarily, such a schedule (with a ScheduleImportExt) is created using the online **Discover** feature in the **Niagara Schedule Import Manager** view of this NiagaraStation. In that way, the imported “slave” schedule is pre-configured, matching the (typically) already configured “master” schedule, as it resides in the Supervisor.

To use a ScheduleExportTag to “slave” to an existing schedule in the Supervisor, you must know that Schedule’s ord (component path) in the Supervisor station—in which case, use of this tag is actually *not needed*. Instead, you could just enter that ord in the “Supervisor Id” property of the schedule’s ScheduleImportExt.

Or, you can use a ScheduleExportTag with a *new* schedule to create (as a result of a Join) a new, blank, “master” schedule on the Supervisor that corresponds to the “slave” schedule on the subordinate (JACE) station. This may be how this export tag is used the most. Using this method, you first *manually* make a new schedule in the Schedules extension of the NiagaraStation (for the Supervisor) in the JACE station, using the **Niagara Schedule Import Manager view**. Then, add this ScheduleExportTag to its ScheduleImportExt, and configure it to “override” the Supervisor Id station slot path. Upon a Join, the new schedule on the Supervisor resides under the NiagaraStation component that represents this subordinate (JACE) station.

**Note:** See “To add ScheduleExportTags” on page 1-10 for a “quick start” procedure on working with these tags. The remainder of this section provides more details on this export tag type.

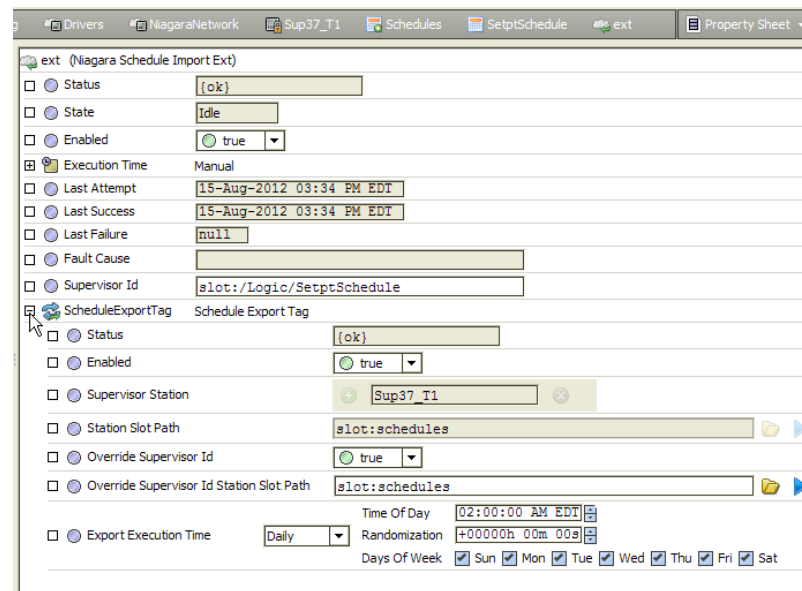
The following sections provide more details on HistoryImportTags:

- [ScheduleExportTag properties](#)

### ScheduleExportTag properties

To see the tag in context, double-click the parent ScheduleImportExt descriptor (🔗 ext) to see its property sheet, then expand the 🗄 ScheduleExportTag, as shown in [Figure 2-36](#).

**Figure 2-36** ScheduleImportExt property sheet example, showing child ScheduleExportTag properties




The important property in the *parent* Niagara **ScheduleImportExt** is the following:

- **Supervisor Id**  
Location of source Supervisor’s (master) schedule, using syntax as follows:  
slot:path  
where *path* includes parent folder(s) *and* the name of the schedule component, for example:  
slot:/Schedules/Ltg/LtgAnnex  
for a schedule named “LtgAnnex” located in the “Schedules”, “Ltg” folder on the Supervisor.  
If adding a new schedule, this property is initially blank—it is populated upon a successful Join.

**ScheduleExportTag** properties are described as follows:

- **Status**  
Read-only status of the export tag, typically “ok”, unless “disabled” (Enabled=false).
- **Enabled**  
Whether the tag will attempt an export on the next Join (true, the default) or not (false).
- **Supervisor Station**  
Reflects the NiagaraStation name for the Supervisor, and is read-only.

- **Station Slot Path**  
Parent slot path for the ScheduleExportExt on the Supervisor, read-only and fixed as the schedules slot (**Schedules** extension) of the NiagaraStation that represent this subordinate (JACE) station.
- **Override Supervisor Id**  
Boolean as to whether to override any existing **Supervisor Id** in the parent ScheduleImportExt, and use instead the NiagaraStation-relative “Override Supervisor Id Station Slot Path” value (next property). By default, this is `false`.  
Typically, you set this to `true` when making a *new* master schedule on the Supervisor, at the location specified in the next property (relative to the NiagaraStation that represents this JACE).
- **Override Supervisor Id Station Slot Path**  
Parent folder location for the source Supervisor’s (master) schedule if “Override Supervisor Id” is set to `true`. The default slot path is simply “slot:”, which means the root of the NiagaraStation that represents this subordinate (JACE) station on the Supervisor.  
Or, click  open folder, to see the **Select Parent Target** dialog with station ready to expand.  
**Note:** You may edit the “profile.bog file” to specify a new target folder structure—see “Verify and edit the Station (.bog File)” on page 1-4 for details.  
When overriding the Supervisor Id, a new master schedule is created on the Supervisor in this parent folder, named the same as the (grandparent) schedule component (slave) that has this export tag. You will need to configure the master schedule in the Supervisor station.
- **Export Execution Time**  
The standard execution time selections for a Niagara ScheduleExportExt descriptor, where the default is `Daily` at 2:00 AM, all days of the week. Other selections include `Manual` or `Interval`.

## About FileImportTags

You can import files from one NiagaraAX station to another. The FileImportTag imports either a single file or an entire folder of files from the subordinate (JACE) station to the Supervisor station. A corresponding Niagara FileImport descriptor is created in the Supervisor, under the **Files** device extension of the NiagaraStation that represents this subordinate (JACE) station. For related details, see “About Niagara file imports” in the *Drivers Guide*.

**Note:** See “To add FileImportTags” on page 1-11 for a “quick start” procedure on working with these tags. The remainder of this section provides more details about this export tag type.

Note that file imports are also included in an “as-needed” manner when using PxViewTags, without a need for separate FileImportTags. For related details, see “About PxViewTags” on page 2-25.

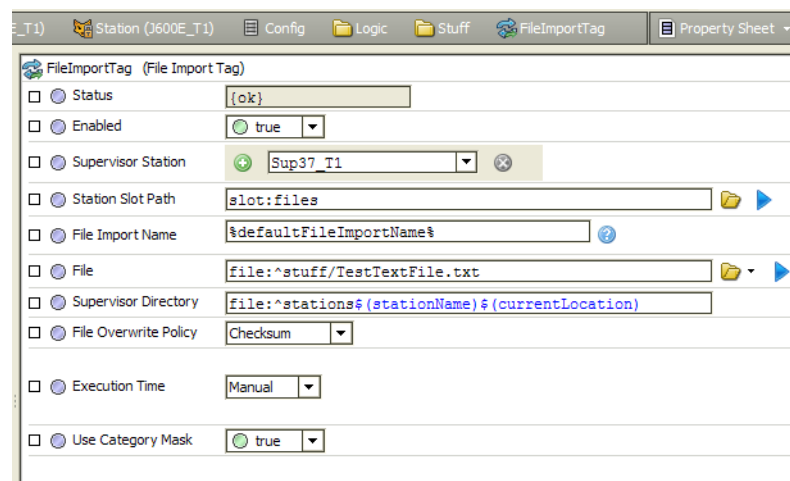
The following sections provide more details on FileImportTags:

- [FileImportTag properties](#)

### FileImportTag properties






Double-click a FileImportTag for its property sheet ([Figure 2-37](#)).

**Figure 2-37** FileImportTag property sheet example



FileImportTag properties are described as follows:



- **Status**  
Read-only status of the export tag, typically “ok”, unless “disabled” (Enabled=false).
- **Enabled**  
Whether the tag will attempt an export on the next Join (true, the default) or not (false).
- **Supervisor Station**  
Click the drop-down  control to select the NiagaraStation that represents the Supervisor (with enabled **JoinProfile**).  
*Note:* After selection, you must **Save** before you can use the  folder (Select Parent Target) feature of the next property. Also, add  (and delete ) controls exist to specify more than one Supervisor station; however, this configuration is rarely expected.
- **Station Slot Path**  
Parent slot path for the FileImport descriptor on the Supervisor, relative to the “Files” extension of the NiagaraStation that represents this station.  
*Note:* Leave this at the default value `slot:files` (the only valid value).
- **File Import Name**  
Name for the Niagara FileImport descriptor created in the Supervisor station. The default value is a variable (`%defaultFileImportName%`), which results in a name that uses the name of source file (or folder) appended to “\_Import”. For example, if the File is “file:^console.txt”, using defaults the name of the FileImport descriptor is `console_Import`. If desired, you can edit to an explicit name.
- **File**  
Specifies the local file or directory to import to the Supervisor. The default value is “file:^”, which imports all objects in the station’s local file space.  
To select, click one of the right-side controls, either:
  - if a single file, the folder  control for the **File Chooser** dialog.
  - if a directory, the adjacent drop-down  control, then choose **Directory Ord Chooser** from the menu for the **Directory Chooser** dialog.Example File property values are: `file:^reports` (for a “reports” directory in the local file space) and `file:^reports/moreReports/subReport2.txt` (for a single file found in a specific subdirectory).
- **Supervisor Directory**  
Specifies the target file directory on the Supervisor, in which file(s) are imported. The default value uses ord variables: “file:^stations\$(stationName)\$(currentLocation)”, to replicate a similar file/subfolder organization on the Supervisor, under a “stations” file subfolder for each station, including this one. Or, specify an explicit location in the Supervisor’s file space (`file:^location`). If the specified directory does not already exist, it will be created upon a Join.  
*Note:* Using the default (variable) property value, if choosing a directory, note its contents (including all subdirectories) are imported “one folder up” from the source (JACE) file location. To include the structure of the chosen directory itself, append it to the end of the variable, that is: `/directoryName`. For example, if importing a directory “reports” (**File**=^reports), enter a Supervisor Directory property value of `file:^stations$(stationName)$(currentLocation)/reports`
- **File Overwrite Policy**  
The standard File Overwrite Policy property for a Niagara FileImport descriptor, either `Checksum` (default) or `Last Modified`.
- **Execution Time**  
The standard Execution Time property for a Niagara FileImport descriptor, where the default is `Manual`. Other selections include `Daily` or `Interval`, where each provides a number of related parameters.
- **Use Category Mask**  
Boolean to specify whether the file(s) imported by this tag will have any export tag *category filters* applied (default is `true`). For related details, see “[About CategoryFilters](#)” on page 2-8.



## About ComponentTags

The **ComponentTag** is an available “general purpose” export tag, to replicate the tagged (parent) component under the NiagaraStation component on the Supervisor. Component tags create a copy of the component from the source station. They do not create a proxy. For example, if you change properties on the component with the tag in the source station, those changes are not reflected in the copy of the component located in the Supervisor station. In general, it may be best for a component that has an “out” slot, but has either a “null” (or no) proxyExt, such as many found in the `kitControl` module. Usage on any “container” type component is generally not recommended.

**Note:** A *ComponentTag* can also replicate a “custom” component, including a “Program” component. An example is included in the **exportTags** palette, in the *Examples* folder (“ExportTagProgram”). Note that starting in AX-3.7, the station user being used in the Join process (Fox connection to Supervisor) must have “super user” privileges for a *ComponentTag* on a Program object to function correctly.

Unlike PointTags, components made from ComponentTags are not Niagara proxy points. Therefore, you can specify them to be located anywhere under the NiagaraStation component (that models this station) in the Supervisor station.

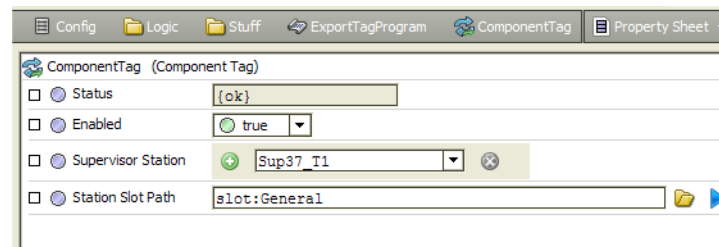
**Note:** See “[To add ComponentTags](#)” on page 1-11 for a “quick start” procedure on working with these tags. The remainder of this section provides more details on this export tag type.

- [ComponentTag properties](#)
- [ComponentTag example \(ExportTagProgram\)](#)

### ComponentTag properties

Double-click a ComponentTag for its property sheet ([Figure 2-38](#)).

**Figure 2-38** ComponentTag property sheet example



ComponentTag properties are described as follows:

- **Status**  
Read-only status of the export tag, typically “ok”, unless “disabled” (Enabled=false).
- **Enabled**  
Whether the tag will attempt an export on the next Join (true, the default) or not (false).
- **Supervisor Station**  
Click the drop-down  control to select the NiagaraStation that represents the Supervisor (with enabled `JoinProfile`).

**Note:** After selection, you must **Save** before you can use the (Select Parent Target) feature of the next property. Also, add (and delete ) controls exist to specify more than one Supervisor station; however, this configuration is rarely expected.

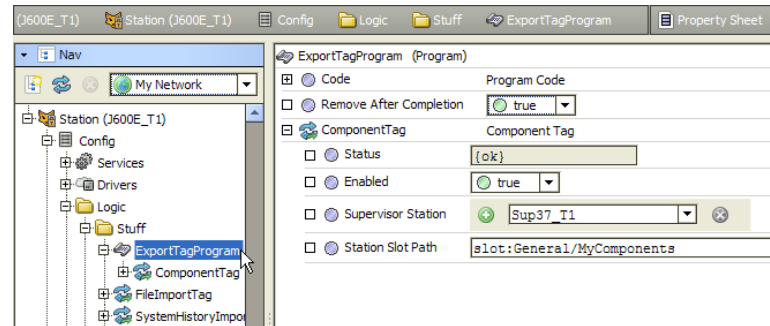
- **Station Slot Path**  
Parent slot path for the replicated component on the Supervisor, relative to the NiagaraStation that represents this station. If multiple Supervisor stations, this applies only to the first (top) one.
  - You can manually type a path, to create corresponding folders under the NiagaraStation (that represents this JACE) on the Supervisor. This includes use of “BFormat” `%value%` options.  
**Note:** The default “slot:” puts the component in the root of the NiagaraStation.  
See “[Station Slot Path considerations](#)” and “[BFormat options in Station Slot Paths](#)” on page 2-22.
  - Or, click the folder for a **Select Parent Target** dialog, and navigate to the desired folder or container. In this case, you must have already made (and saved) the needed folders in the JACE’s station “profile.bog” file.  
**Note:** You can edit the “profile.bog file” to add folders—see “[Verify and edit the Station \(.bog\) File](#)” on page 1-4 for details.

### ComponentTag example (ExportTagProgram)

The **Examples** folder in the `exportTags` palette contains a Program component named “ExportTag-Program”, that is already tagged with a ComponentTag. **Figure 2-39** shows this component copied into a subordinate (JACE) station, and its property sheet, including its configured ComponentTag.

**Note:** A user is required to have “super user” permissions in order to add/edit Program objects or Provisioning Robots in a station. If a subordinate station includes a Component Tag on a Program object, both the service account (for station to station communication) and the user invoking the Join require “super user” permissions. However, such permissions are not required to invoke a Join on a station that includes such pre-configured export tags.

**Figure 2-39** Example ExportTagProgram copied into a station, with configured ComponentTag

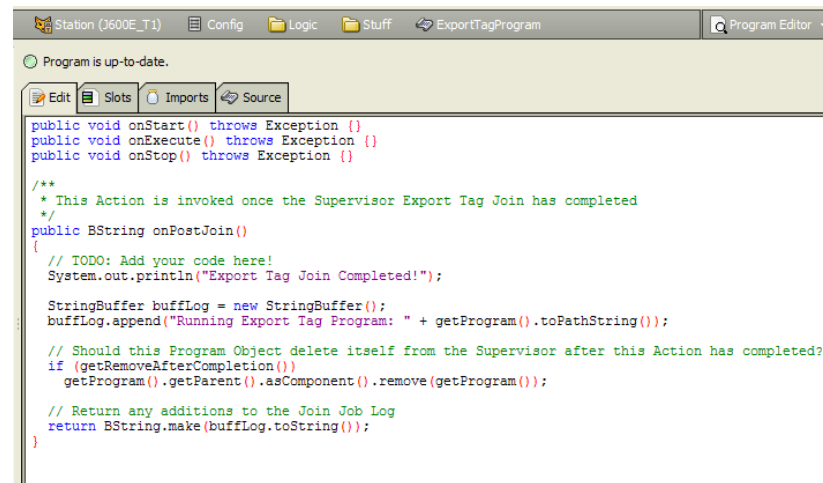


The purpose of this Program is to provide a “template” that automatically executes upon completion of an export tag Join, such that “custom operations” can be automated. This allows a user experienced in writing Niagara (Java) programs to insert the necessary code to make links, or do whatever else may be needed. In this way, it can be considered an extension of the `exportTags` API.

Note that the Program has a “Remove After Completion” property, as shown in **Figure 2-39**. By default, this property is `true`, meaning that the Program is exported to the Supervisor, executes following the Join, and then is removed from the station database. To retain the Program component on the Supervisor, set this property to `false`.

**Figure 2-40** shows the `ExportTagProgram` opened in the **Program Editor**, as copied from the palette.

**Figure 2-40** ExportTagProgram “template” code (unmodified)



In the code shown above, the line

```
public BString onPostJoin()
```

Is the dynamic action (method) that gets called after the Join process is all finished. It also returns a string that within the ending code, gets included in the Job Log.

Immediately below that line is the “template” area, in which you can add additional code. Currently, the placeholder line:

```
System.out.println("Export Tag Join Completed!");
```

Simply prints that line to the Supervisor’s standard *station output*, where it can be seen in the platform’s Application Director view.



# CHAPTER 3

## Export Tag Plugin Guides


Plugins provide *views* of components, and can be accessed many ways—for example, double-click a component in the tree for its *default* view. In addition, you can right-click a component, and select from its **Views** menu. For summary documentation on any view, select **Help > On View (F1)** from the Workbench menu, or press F1 while the view is open.

### Export Tag Plugin Guides Summary

Summary information is provided on views specific to components in the `exportTags` module, as follows:

- [Category Filter Manager](#)
- [Join Profile Manager](#)
- [Export Tag Summary Manager](#)

#### exportTags-CategoryFilterManager


 The Category Filter Manager is a view on the `CategoryFilterExt` (**Category Filters**) container under the `SupervisorExportTagNetworkExt`. To view, right-click **Category Filters** and select **Views > Category Filter Manager**, or simply double-click **Category Filters**.


Use the **Category Filter Manager** to add, edit, and access `CategoryFilter` components that can be configured to automatically apply Categories to NiagaraStations in the Supervisor's NiagaraNetwork.

View columns are summarized below. For details, see [“About the Category Filter Manager”](#) on page 2-8.

- **Name**  
By default, `CategoryFilter`, `CategoryFilter1`, `CategoryFilter2`, and so on. Often you rename each filter to be more descriptive, such as the name of a company or customer.
- **Type**  
(by default, not set to display) Currently this is always `CategoryFilter`.
- **Station Info Property Filters**  
Shows configured “Station Info” slots and value pairs, with a comma (,) separating each one—for example: `State -> Virginia, City -> Richmond`
- **Category Mask**  
Lists `Category(ies)` automatically applied to subordinate `NiagaraStations` that have matching “Station Info” slot/values.

#### exportTags-JoinProfileManager


 The Join Profile Manager is a view on the `SubordinateExportTagNetworkExt` in the NiagaraNetwork of a JACE station. To view, right-click this extension and select **Views > Join Profile Manager**, or simply double-click the `SubordinateExportTagNetworkExt`.

By using the  **Join** button at the bottom of this view (with the Supervisor station) selected, the join automatically completes with an ending popup **Job Log** dialog. From this view you can also select and edit the **JoinProfile** for the Supervisor. A few columns in this view correspond to `JobProfile` properties, with the ones shown by default summarized below. For details, see [“About the Join Profile Manager”](#) on page 2-12.

- **Station Name**  
An entry for the Supervisor is required, and it should be the only one enabled.
- **Last Join**  
Timestamp of the last join for this subordinate station.

- **Station Display Name**  
(usage optional) Shows any entered station “display name” in the JobProfile. This is used for this subordinate station as it appears in the Supervisor’s NiagaraNetwork, instead of the actual
- **Default Subordinate User**  
(usage optional) Shows the JobProfile’s station user property value, utilized for a Fox client connection from the Supervisor. This is necessary only if the subordinate (JACE) was not previously set up in the Supervisor’s NiagaraNetwork.
- **Default Subordinate User Password**  
This is an added property in AX-3.7u1 and later. The local (JACE) station user password to be used by the Supervisor to connect back to this station. If the subordinate station includes component tags added to program objects which are also enabled, both the service account (used for subordinate to supervisor communication) and the user invoking the Join job require “super user” permissions. Like the other “Default” properties in this dialog, it may be left blank if the Supervisor already has a configured NiagaraStation for this station.  
*Note: You must configure the Default Subordinate User Password when pre-engineering a subordinate station to join to a Supervisor. Failure to configure the password will result in authentication errors when the Supervisor attempts to make a connection to this subordinate JACE (that the Supervisor is not already “aware” of).*
- **Station File**  
Shows the file ord to the local “profile.bog file”, used to model how the NiagaraStation for this subordinate JACE is represented in the Supervisor’s NiagaraNetwork.

### exportTags-ExportTagSummaryManager

 The Export Tag Summary Manager is a view on the **SubordinateExportTagNetworkExt** in the NiagaraNetwork of a JACE station. To view, right-click this extension and select **Views > Export Tag Summary Manager**.

This view provides a tabular listing of all export tags used in this subordinate station. Double-click any export tag (row) to see that specific export tag’s property sheet. Or, click to select one or more rows, then click the **Edit** button at the bottom for a popup Edit dialog. In the **Edit** dialog you can change a few selected properties, including whether the tag(s) are enabled or disabled.

Columns in this view correspond to export tag properties, with the ones shown by default summarized below. For details, see “[About the Export Tag Summary Manager](#)” on page 2-16.

- **Name**  
The component name for each export tag, often left at default (PointTag, HistoryImportTag, etc.).
- **Path**  
The ord path of the export tag, relative to the local station root.
- **Status**  
Typically {ok} unless {disabled}.
- **Supervisor Station**  
Station name of the Supervisor station.
- **Station Slot Path**  
The relative slot path of the parent container for the exported component, history, file, or PxView, selectable from the local “profile.bog file.” This bog file determines how the Supervisor station, upon a join, organizes exported objects under its NiagaraStation for this JACE.

# CHAPTER 4

## Export Tag Component Guides


These component guides provides summary help on exportTags components.

### ExportTag Component Reference Summary

Summary information is provided on components in the `exportTags` module, listed alphabetically as follows:

- [CategoryFilter](#)
- [CategoryFilterExt](#)
- [ComponentTag](#)
- [FileImportTag](#)
- [HistoryImportTag](#)
- [JoinProfile](#)
- [PointTag](#)
- [PxViewTag](#)
- [ScheduleExportTag](#)
- [ScheduleImportTag](#)
- [StationInformation](#)
- [SubordinateExportTagNetworkExt](#)
- [SupervisorExportTagNetworkExt](#)
- [SystemHistoryImportTag](#)


#### exportTags-CategoryFilter

 CategoryFilters are child components of the **CategoryFilterExt** (Category Filters) of the **SupervisorExportTagNetworkExt**. CategoryFilters can be configured to automatically assign security Categories to NiagaraStations under an Supervisor's NiagaraNetwork, based upon "Station Info" string values for each subordinate station. For more details, see ["About CategoryFilters"](#) on page 2-8.


#### exportTags-CategoryFilterExt

 CategoryFilterExt (Category Filters) is a frozen container slot on the **SupervisorExportTagNetworkExt**, used to hold child [CategoryFilter](#) components. For details, see ["About CategoryFilters"](#) on page 2-8.


#### exportTags-ComponentTag

 Component Tags are general purpose tags for exporting a copy of any type of component (even a Program components) from the source station to a Supervisor. For example the **ExportTag-Program** component in the **Examples** folder of the `exportTags` palette. Component tags do not "proxy" the source component. For any source proxy point, you should use a PointTag type instead. For more details, see ["About ComponentTags"](#) on page 2-38.

#### exportTags-FileImportTag


 A File Import Tag creates a Niagara FileImport descriptor on the Supervisor, a feature introduced starting in AX-3.5. This allows automatic transfer of the selected file(s) from the JACE station to the Supervisor station, either the selected file or entire directory. Included is a "Use Category Mask" property. For more details, see ["About FileImportTags"](#) on page 2-36.

#### exportTags-HistoryImportTag

 A History Import Tag creates a Niagara HistoryImport descriptor on the Supervisor, to import the tagged history. The HistoryImportTag includes configuration properties of a Niagara HistoryImport descriptor, including "On Demand Poll" properties and "Config Override" properties. Also included is a "Use Category Mask" property. For more details, see ["About HistoryImportTags"](#) on page 2-31




### exportTags-JoinProfile

 As a result of adding a **SubordinateExportTagNetworkExt** in the NiagaraNetwork of a JACE station, a Join Profile component is created under each NiagaraStation component, including the one that represents the Supervisor. That Join Profile contains all information needed when a “Join” command is issued, including (when enabled) the “station (bog) file” created on the JACE.


This profile.bog file allows you to model how the Supervisor’s NiagaraNetwork will model the subordinate JACE, following a join—and all subsequent rejoins. The JoinProfile also contains a StationInformation (Station Info) container, to optionally classify the station using string values.

For more details, see [“About the JoinProfile”](#) on page 2-3 and [“About the SubordinateExportTagNetworkExt”](#) on page 2-11.


### exportTags-PointTag

 Point Tags are used to export selected components to the Supervisor station as Niagara proxy points. They are similar to ComponentTags, but include additional properties for Conversion type and (NiagaraNetwork) Tuning Policy Name. In general, use a PointTag for any source proxy point in the JACE station, that is any component that has a working (non-null) ProxyExt of any type. For more details, see [“About PointTags”](#) on page 2-30.


### exportTags-PxViewTag

 A PxView Tag is used to export entire an entire PxView from the JACE station to the Supervisor station. Automatically included is all underlying exports of files (graphic images) used, as well as ords to Niagara *virtual points* needed for real-time values. The exported PxView can be specified as a view on any container under the NiagaraStation representing the subordinate JACE. An “order number” allows multiple PxViews to be exported to the same container. For details, see [“About PxViewTags”](#) on page 2-25.


### exportTags-ScheduleExportTag

 A Schedule Export Tag creates a Niagara ScheduleExportExt descriptor and “master” schedule on the Supervisor station that corresponds to a “slave” schedule on the JACE station. It must be a child of an existing Niagara ScheduleImportExt. Usage is expected to be more typical than the ScheduleImportTag, which creates a “slave” schedule on the Supervisor station. For details, see [“About ScheduleExportTags”](#) on page 2-34.

### exportTags-ScheduleImportTag


 A Schedule Import Tag creates a Niagara ScheduleImportExt descriptor and “slave” schedule on the Supervisor station, and a corresponding Niagara ScheduleExportExt descriptor on the JACE station. It can be added under any schedule component in the source JACE station. Usage is expected to be infrequent, as typically “master” schedules are in the Supervisor. For details, see [“About ScheduleImportTags”](#) on page 2-34.

### exportTags-StationInformation

 StationInformation (Station Info) appears in two places: first, as source: in each JACE station, under the JoinProfile of the NiagaraStation that represents the Supervisor. By default, this container slot is empty. However, from its slot sheet you can add one or more slots of type `baja:String`, entering a string Name to describe each slot; for example: `State` or `City`. From the property sheet you can then enter string values in each added slot.

Upon a JACE station join, this Station Info is exported to the Supervisor station, appearing under the NiagaraStation component for that JACE. This data can be utilized in various ways in the Supervisor, including BQL queries, “automatic station folder creation” (from the “Station Folder Path” property of the SupervisorExportTagNetworkExt), and in CategoryFilters that use Station Info property values to apply security categories to objects (components, files, and histories) under that NiagaraStation. For more details, see [“About Station Info”](#) on page 2-15.


### exportTags-SubordinateExportTagNetworkExt

 The Subordinate Export Tag Network Ext is required in any JACE station that needs to use export tags. Copy it from the `exportTags` palette to the NiagaraNetwork in the JACE station.

The default view is the **Join Profile Manager**, in which the JoinProfile for the Supervisor appears, and must be enabled. Use the **Join** button in this view to issue a join command that includes a popup **Job Log** at the end, useful for review.


Another view on the SubordinateExportTagNetworkExt is the **Export Tag Summary Manager**. This provides a tabular listing of all export tags used in the local station. Each row provides a hyperlink to the specific export tag. For more details, see [“About the SubordinateExportTagNetworkExt”](#) on page 2-11.

### **exportTags-SupervisorExportTagNetworkExt**

 The Supervisor Export Tag Network Ext is required in the Supervisor station to benefit from export tags. Copy it from the `exportTags` palette to the Supervisor station's NiagaraNetwork. A few configuration properties are found on its property sheet, with no other views.

Among child components, the SupervisorExportTagNetworkExt contains a **CategoryFilters** container with a **Category Filter Manager** view. Use it to create, edit, and delete **Category-Filter** components. For more details, see [“About the SupervisorExportTagNetworkExt”](#) on page 2-7.

### **exportTags-SystemHistoryImportTag**

 A System History Import Tag creates a Niagara SystemHistoryImport descriptor on the Supervisor, to import histories based upon “System Tag” property values. System tag importing of histories was first introduced in AX-3.4.

The System History Import Tag includes configuration properties of the Niagara SystemHistoryImport descriptor, including the “System Tag Patterns”, “On Demand Poll” properties, and “Config Override” properties. Also included is a “Use Category Mask” property. For more details, see [“About SystemHistoryImportTags”](#) on page 2-32.

