

Technical Document

Templates Guide

August 13, 2015

niagara⁴

Templates Guide

Tridium, Inc.

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

Confidentiality

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

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

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

Trademark notice

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

Copyright and patent notice

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 © 2015 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

About this Guide	5
Document change log	5
Related documents (Templates)	5
Chapter 1 Template overview	7
Station templates	8
Basic checklist for designing a template	9
Chapter 2 Common templating tasks.....	11
Creating device templates	11
Making a template	11
Modifying components.....	13
Configuring exposed properties	14
Adding relations.....	14
Managing I/O links	15
Managing graphic views	17
Creating a station template	18
Making a template module	20
Deploying templates	21
Deploying a template via drag-and-drop	22
Deploying a template via the Device Manager	25
Editing a template.....	26
Chapter 3 Template reference	29
Template troubleshooting	29
Template side bar	30
Components	30
Template Service.....	31
TemplateConfig	31
Plugins (views).....	32
Template Manager view	32
Template view.....	34
Glossary	39

About this Guide

This document describes how to use Niagara templates to improve productivity and consistency when working with station and enterprise systems.

Document change log

Updates (changes and additions) to this document are listed below.

- Initial release document, August 13, 2015.

Related documents (Templates)

Following are documents related to this guide.

- Hierarchies Guide
- Relations Guide
- Tagging Guide

Chapter 1 Template overview

Topics covered in this chapter

- ◆ Station templates
- ◆ Basic checklist for designing a template

A template is a deployable package of Niagara objects. Templates contain a component tree with a single root component and associated support objects. The purpose of templates is to allow you to take a set of configured objects and encapsulate those to be deployed as a unit. The benefit of using templates is that it eliminates unnecessary repetition of configuration steps when making multiple installations with similar functionality.

The deployment process extends the normal installation processes used for a single component. At deployment, the template user only needs to make modifications that are unique to that installed instance. One or more instances of a template may be deployed to a station.

The template designer is the person who creates and maintains the template. The template user is the person who deploys the template to configure an installed instance. These, of course, may be the same person.

When creating a new template, the designer may expose certain properties as special configuration or linkable input/output properties. Configuration properties are properties a user must set to customize different instances of the template. I/O properties are properties that the template designer expects to be linked during deployment.

A template is contained in a file that has a `.ntpl` (Niagara template) file extension. Typically, template files are stored in the Workbench User Home `templates` directory.

Optionally, you can create specialized “station templates” which contain everything needed for the initial starting point of a new station. Station template files are stored in the Workbench User Home `stationTemplates` directory and are used exclusively by the Workbench **New Station** tool.

Template files may be grouped into a module for ease of distribution. Each module may contain multiple template files. Template modules are stored in the Sys Home `modules` directory.

License requirements

Templates are licensed in two places, with the same feature: **template**. On the Workbench side, a local check is used to enable Template editing and creation. On the station side, the Template Service is licensed. A station without a template license cannot have templates deployed to it.

This template licensing structure provides four possible combinations of features:

	Station unlicensed	Station licensed
Workbench unlicensed	No template functionality available.	Workbench can deploy templates to station. Workbench cannot create templates from station. Workbench cannot edit existing templates.
Workbench licensed	Workbench cannot deploy templates to station. Workbench can create templates from station. Workbench can edit existing templates.	Full template functionality.

Template Service

The **TemplateService** provides management support for templates that are deployed in the station. One of the primary functions of the service is to identify and help to resolve any unlinked template inputs, outputs and relations during deployment. The **TemplateService** must be installed and licensed for the running station in order to take advantage of this feature.

About tags

Tags are an important aspect of templates. Since tagging is a significant effort, you should incorporate tagging early. A best practice when creating templates is using tagged devices exclusively. In doing that, you get the most benefit by reusing the tagged components and points. Eventually, it may be necessary to add or edit tags in a template. Optionally, you can make a new template using an existing template as your starting point. Then the next step might be to add or edit tags in the new template.

NOTE: The **tags** license is required to use the **TagDictionaryService** and tag dictionaries on a station.

The following is a basic workflow for tagging and templating:

1. Tag a device (after discovery and adding).
2. Make a template of the device.
3. Use the already tagged template for adding devices to a station.

Station templates

A station template is a specialization which contains a complete set of configured objects, everything required for the initial starting point for a new station. The **New Station** wizard in **Workbench Tools** utilizes default station templates (`NewControllerStation.npt1` and `NewSupervisorStation.npt1`) as well as any user-defined station templates.

User-defined station templates are saved in the `stationTemplates` directory of your Workbench User Home. Template files in this location are available for inclusion in the **New Station** wizard.

You can edit a station template just as you would edit a component template except that you cannot configure I/Os for a station template. For that reason, when creating a station template in the **Template** view, there is no **Template I/O** tab.

Once you have created and saved a station template it is immediately available for use by the **New Station** wizard.

Basic checklist for designing a template

The template designer uses Workbench to create and edit templates. The concept is to fully engineer a collection of tagged control logic and self-contained graphics with a single component as the root. For example, a Device component that contains a collection of proxy points, control logic, and graphics. It is important to use tagged components.

Use the following basic checklist when designing a template:

1. Identify the desired function of the template control logic.
2. Identify any external inputs that are required by the template functionality. These will typically be an input slot of a **WritablePoint**.
3. Identify the configuration properties of the template. These will typically be selected properties of components contained in the template. For example, Alarm limits of a point alarm extension.
4. Identify any outputs produced by the template functionality for external consumption. These will typically be the out slot of a **ControlPoint**.
5. Identify any graphic view requirements of the template. All graphic binding must be contained within the template component tree.

TIP: The best practice is to completely engineer the graphic in a station prior to making a template.

6. Identify the tagging requirements of the template. If necessary, add the appropriate tags to the components prior to creating the template.
 - Consider possible uses: enterprise structure navigation, systems maintenance views, end user navigation.
 - Consider necessary dictionaries: Niagara, custom, Haystack.
7. Identify any relations that need to be established upon deployment.

Chapter 2 Common templating tasks

Topics covered in this chapter

- ◆ Creating device templates
- ◆ Creating a station template
- ◆ Making a template module
- ◆ Deploying templates
- ◆ Editing a template

The following sections describe some common ways to use templates.

Creating device templates

Configuring the device properties for a specific group of identical or similar devices is the ideal situation in which to create and use a device template. Do this just after you finish configuring properties for the first device.

For example, assume you are configuring four air handler units (AHUs), one for each floor of a four-story building and the temperature limits vary from floor to floor. Using an AHU component that you have already tagged and configured for the first floor, you can create an AHU template. In the Template view, on the **Configuration** tab, select and expose the temperature limits. On the Relations tab, define any desired relations between the root component of this template and other components in the station in which the template is deployed. On the **Template I/O** tab, setup the required links for the template inputs and outputs. On the **Graphics** tab, create or modify the graphic views associated with the components in the template component tree. When finished, save the template. Using the **Device Manager**, you can apply the AHU template to the discovered AHU devices for the remaining floors. During template deployment you are prompted to set the temperature limits and resolve the inputs and outputs, relations for each device.

NOTE: To avoid future problems, make sure that you tag and configure your first device correctly before using the resulting template to configure other devices. Niagara does not automatically apply subsequent template changes to already-configured devices.

Prerequisites

- The device driver is located under the **Config→Drivers** node in the Nav tree.
- The device is located under the driver network.
- You have assembled a collection of tagged control logic including self-contained graphic view(s) with a single component as the root.

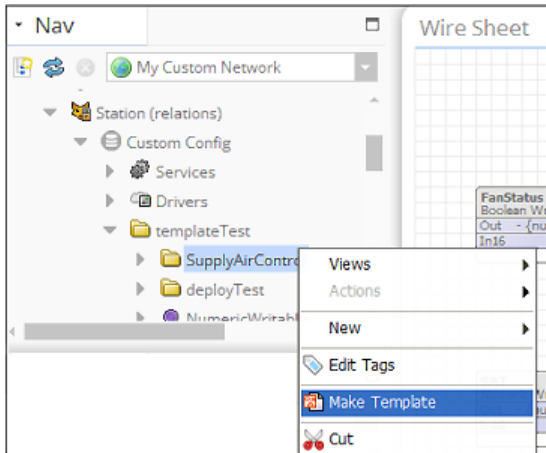
In order to simplify instructions for the template creation process, a separate procedure describes the steps to be done on each of the six configuration tabs in the **Template** view. Those procedures are described in the following sections.

Making a template

Making a template is the first of several procedures in the template creation process.

Step 1 In the Nav tree, expand the **Config→Drivers** folder.

Step 2 Right-click on the desired root component (shown here, the root component is a folder), and click **Make Template**.



The newly created template of the selected root component appears in the **Template** view, with the **Template Info** configuration tab selected. By default, the template name is the same as that of the root component.

Step 3 Enter the following information on the tab:

- **Name** — edit the default name to enter a unique name. This becomes the template filename.
- **Version** — enter a version number.
- **Description** — enter a short description of the template.

The **Info** field is useful for providing additional details or instructions. Also, if you select an **Icon** image, it displays next to the template name.

Template:SupplyAirResetNew2 Vendor:Tridium Version:1.1

Template Info Component Configuration Relations Template I/O Graphics

Name SupplyAirResetNew2

Vendor Tridium

Version 1.1

Description

Info

Icon NO ICON SELECTED

Any of the following notices may display on the **Template Info** tab:

- If the template component tree contains links whose source or target components are not contained within the same component tree, a notice alerts you to review the automatically added I/O links and associated **bindHints** in the **Template I/O** tab.
- If the template root component is a Device component, a notice alerts you to review the **Component** tab, and to set the device address property to its default value.
- If the template component tree contains a **Password** property, a notice alerts you that during deployment the user may need to set the password.

The next procedure in creating a template is modifying the template's components and links which is done using the **Components** tab.

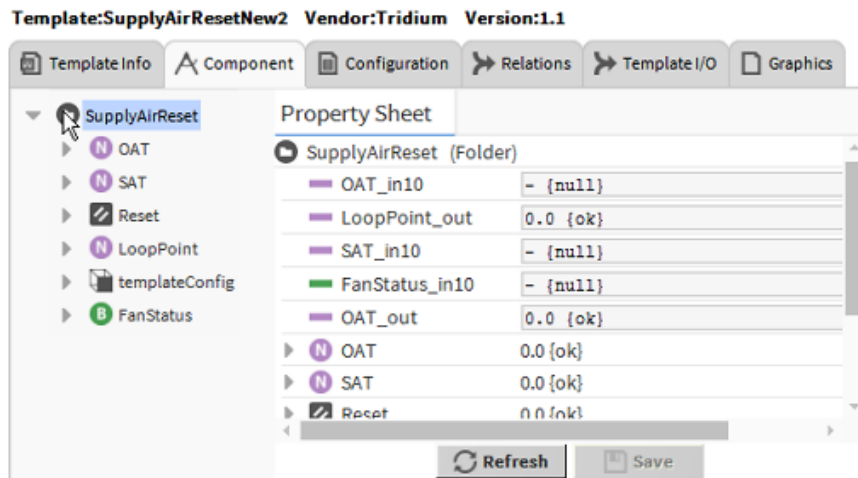
Modifying components

The **Component** tab in the **Template** view allows you to make changes to the template's control logic by adding or removing components, as well as adding or removing links.

Step 1 In the **Template** view, click the **Component** tab.

Step 2 In the left pane, right-click the template root component to select a view (Property Sheet, Wire Sheet or Slot Sheet) of the components.

The selected view displays in the right pane, as shown here:



Step 3 To add or remove components:

Action	Description
Add a component	a. Drag and drop a component from the Nav tree or a palette onto the root component of the template in either the left or right pane.
Delete a component	a. Right-click the component and click Delete .

Step 4 To add or remove a link:

Option	Description
Add a link	a. Right-click on a component you wish to link to another and click Link mark . This is now the "link-marked component". b. Right-click the other component and select either Link from... (link-marked componentName) or Link to... (link-marked componentName).
Remove a link	a. On the Wire Sheet view of the root component of the template, right-click on the link you wish to remove and click Delete Links .

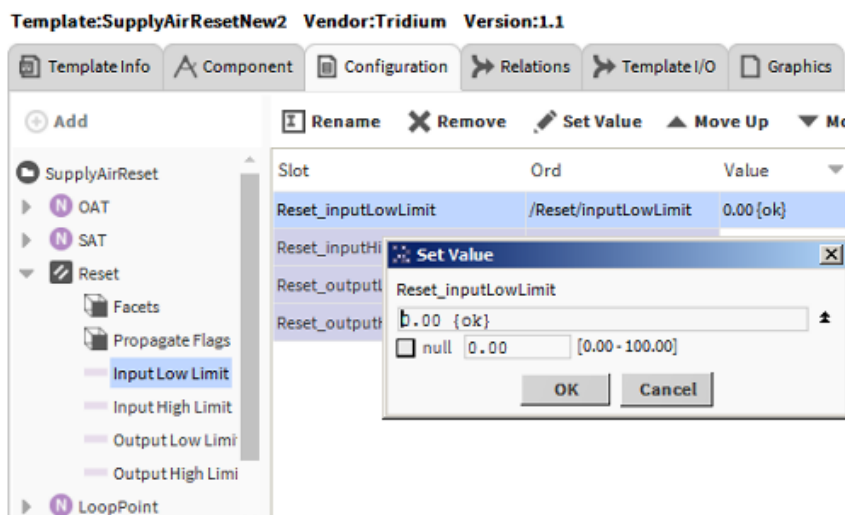
Step 5 Click **Save**.

The next procedure in creating a template is exposing specific properties that the user must set during deployment. This is done using the **Configuration** tab.

Configuring exposed properties

The **Configuration** tab in the **Template** view allows you to select and expose specific component properties contained within the template control logic. During deployment a popup prompts the user to edit the values for these properties.

- Step 1 In the **Template** view, click the **Configuration** tab.
- Step 2 In the left pane, expand components until the desired property is visible.
- Step 3 Select one or more properties to expose.
- Step 4 Expose the property by double-clicking it or select it and click **Add**.
The selected property is added to the right pane.
- Step 5 Edit the default value for the first exposed property shown in the right pane by double-clicking on the a Value cell or select a row and click on **Set Value**.



- Step 6 In the **Set Value** dialog, enter the desired default value.
- Step 7 For each exposed property, repeat Steps 5 and 6.
- Step 8 Click **Save**.


NOTE: The value entered into the template for the exposed property becomes the default value upon deployment.

In the example shown here, the `Reset_InputLowLimit` property is exposed so that during deployment the user is prompted edit the value.

The next procedure in creating a template is adding desired relations between the root component of this template and any other components in the template. This is done using the **Relations** tab.

Adding relations

The **Relations** tab in the **Template** view allows you to define desired relations between the root component of this template and other components in the station in which the template is deployed. You can choose from **RelationIds** found in the installed tag dictionaries or you can define "Ad Hoc" relations. Upon adding, if desired you can reverse the default relation direction. Also, in order for the relation to resolve when deployed you must configure a NEQL search predicate via the `RelateHints` cell.

- Step 1 On the **Relations** tab in the **Template** view, add a relation using either of these methods:
 - Add from the installed tag dictionaries: Select a **RelationId** in the lower left pane and click  **Add** (or double-click the **RelationId**).

- Add an Ad Hoc relation (one not found in a tag dictionary): With nothing selected click **Add** and enter the desired **RelationId** in the popup dialog.

The added relation appears in the right pane.

NOTE: Typically, the default **Relation Direction** is sufficient. However, you may need to change a relation direction in order to get the desired query results, i.e., hierarchy results.

Step 2 If desired, change the relation direction. In the right pane, select the row containing the added relation and click **Reverse**.

NOTE: You can add a given RelationId only two times. The first time you add a relation, the default relation direction is “inbound.” If you add the same RelationId a second time, by default it will have the opposite relation direction.

Step 3 Enter a **RelateHints** by selecting the row in the right pane, double-clicking on the **RelateHints** cell, and in the **Set Value** dialog, enter a NEQL search predicate value and click **OK**.

NOTE: In this step you are tagging the relation. This tag is a string value that is used as a NEQL search predicate during deployment to search for and suggest potential related components.

If you added the `hs:ahuRef` **RelationId**, you might enter this search predicate: `hs:ahu`.

If your entry is a valid NEQL search predicate, the table in the right pane is updated with this value. If not valid, the error is listed in the table.

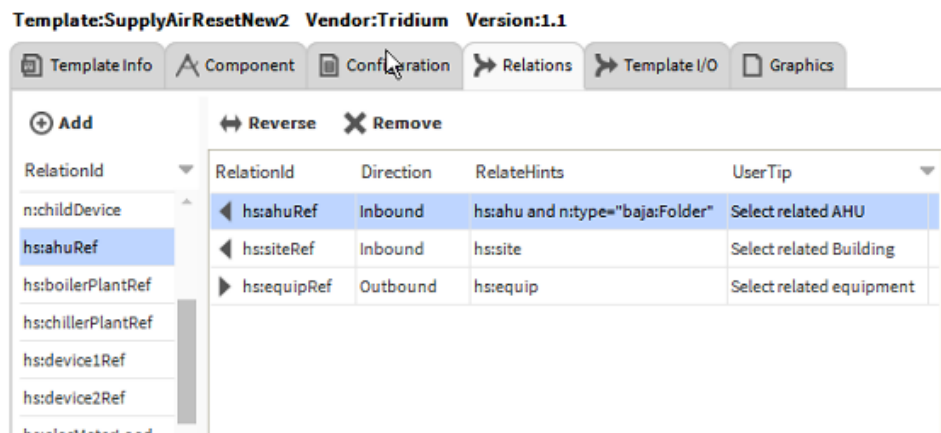
Step 4 Enter a **UserTip** by selecting the row in the right pane, double-clicking on the **UserTip** cell, and in the **Set Value** dialog enter a string value and click **OK**.

You might enter: `Select Related AHU`.

The **UserTip** is additional string value that appears in a popup during deployment providing guidance for selecting a related component.

The table in the right pane is updated with the **UserTip**.

You have completed defining relations for this template.



The next procedure in creating a template is exposing specific control inputs and outputs. This is done using the **Template I/O** tab.

Managing I/O links

The **Template I/O** tab allows you to expose specific control inputs that are required by the template and control outputs provided by the template. Also, (if licensed for `tags` and `tagdictionaries`) you can apply additional tags to the exposed I/Os. During deployment the template attempts to resolve these linkable I/O properties, and may prompt the user to choose among multiple sources.

This procedure describes adding a control input to expose it as a linkable point in the template.

- Step 1 In the **Template** view, click the **Template I/O** tab.
- Step 2 In the left pane, click to expand the desired control point and double-click on the desired input slot to add it (or select it and click **+ Add**).

The input slot is added to the template’s slot sheet shown in the upper right pane.

- Step 3 In the upper right pane, select the added input slot.

NOTE: By default, the `n:input` and `n:bindHints` tags are added automatically and display in the **I/O Tags** pane (lower right) along with any other tags previously applied to this input. The `n:bindHints` tag is the string value that is used as a NEQL search predicate. During deployment, this assists the installation tool in resolving required template inputs.

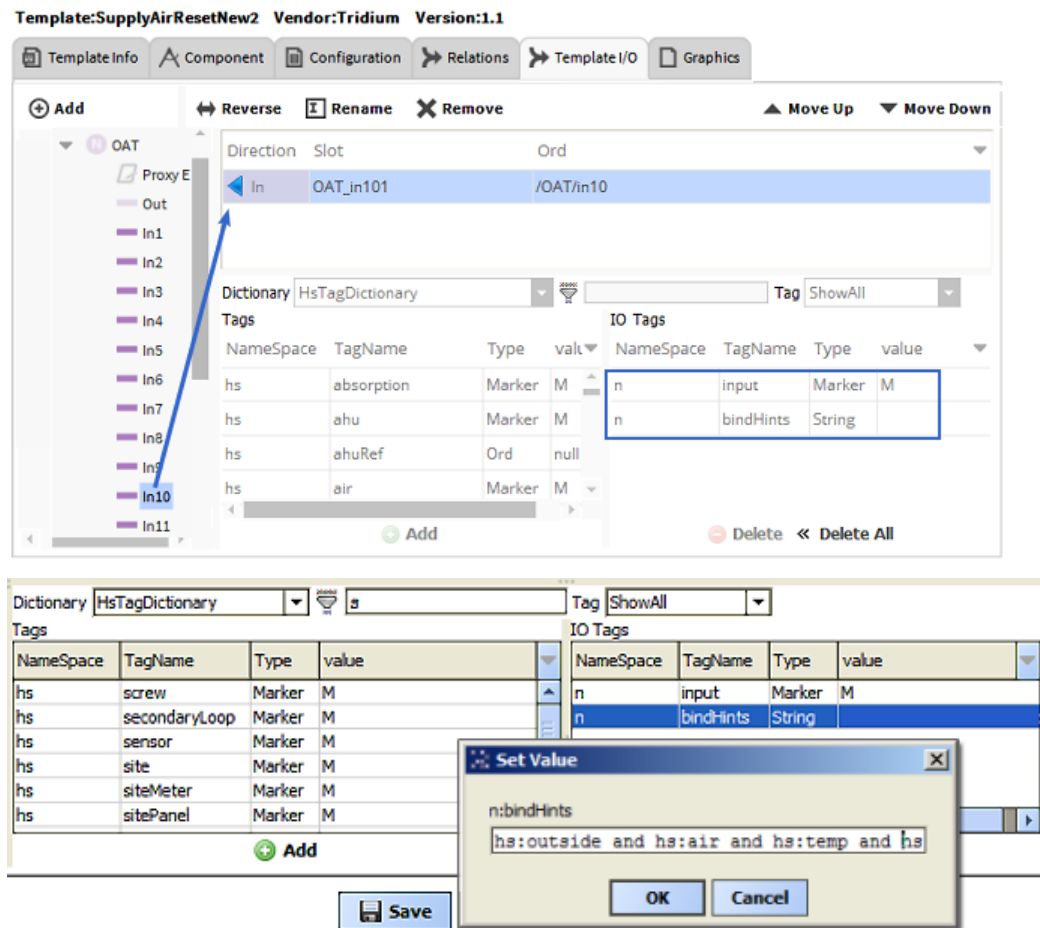
- Step 4 To edit the `n:bindHints` tag, in the **I/O Tags** pane (lower right), select the `n:bindHints` row and double-click on the **Value** cell.

The **Set Value** dialog displays.

- Step 5 Enter a NEQL search predicate for the input slot and click **OK**.

TIP: Use the tag dictionaries in the lower left pane to identify tags to include. Also, the `n:bindHints` value is validated upon entry and any errors are indicated in the **I/O Tags** pane (lower right).

In the example shown here the selected input is for the current outside air temperature and in the lower left pane the `HaystackTagDictionary` is selected. The NEQL search predicate that must be entered as the value for `n:bindHints` is: `hs:outside and hs:air and hs:temp and hs:sensor`.



At this point you can expose additional inputs and outputs as needed.

Expose template *outputs* using the same procedure except that you typically select the “out” property of a control point.

- Step 6 To expose a template output, in the left pane click to expand the desired control point and double-click on the desired output slot to add it (or select it and click **Add**).

The output slot is added to the template’s slot sheet shown in the upper right pane.

NOTE: By default, the `n:output`, `n:bindHints`, and `n:targetSlotHint` tags are added automatically and display in the **I/O Tags** pane (lower right). These tags are used during deployment to assist in linking the template outputs to other writable points in the station.

- Step 7 To edit the `n:bindHints` tag, in the **I/O Tags** pane (lower right), select the `n:bindHints` row and double-click on the **Value** cell.

The **Set Value** dialog displays.

- Step 8 To edit the `n:targetSlotHint` tag, in the **I/O Tags** pane (lower right), select the `n:targetSlotHint` row and double-click on the **Value** cell.

This tag is a string value that is used to suggest the target slot of any writable points that match the `n:bindHints` query. The value is a comma separated list of inputs. The order of the list will be the order in which the deploy tool will suggest the target slot. For example: If the `n:targetSlotHint` tag value is: “in10, in13”, the deploy tool would first suggest linking to the target slot “in10” if it was linkable. If “in10” was not linkable, then it would next suggest “in13”. If neither are linkable, then it would not suggest an input. Instead it would let the user choose. In any case, the user can override the suggested target slot.

The **Set Value** dialog displays.

- Step 9 Enter a **UserTip** by selecting the row in the right pane, double-clicking on the **UserTip** cell, and in the **Set Value** dialog enter a string value and click **OK**.

The **UserTip** is an additional string value that appears in a popup during deployment providing guidance in making a selection choice.

The table in the right pane is updated with the **UserTip**.

On completion, the template’s linkable inputs and outputs are configured.

The next procedure in creating a template is adding or modifying graphic views in the template. This is done using the **Graphics** tab.

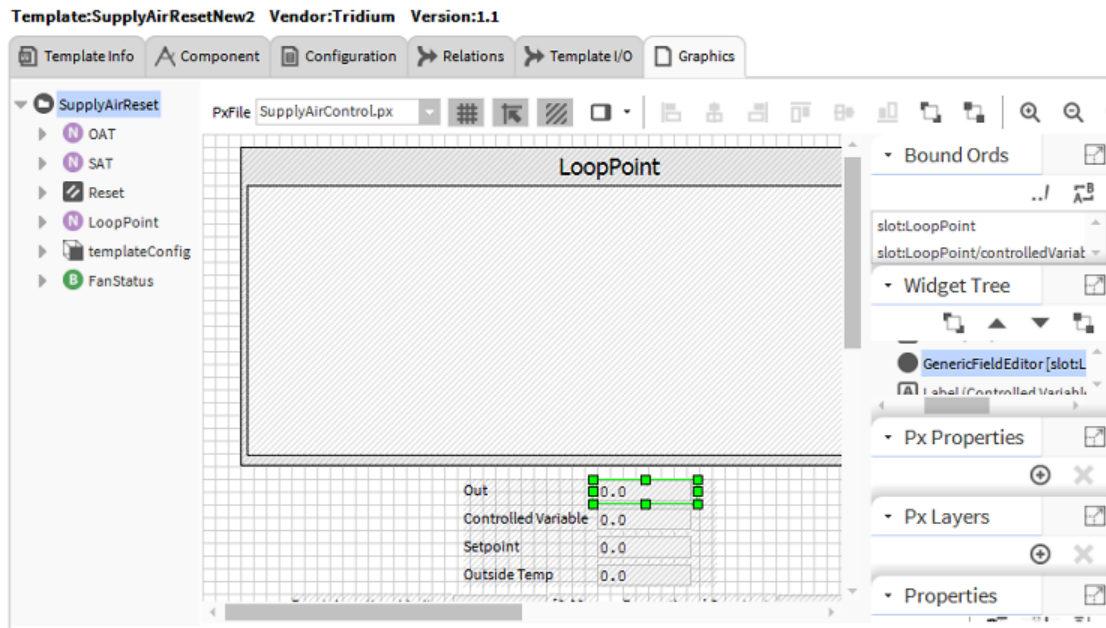
Managing graphic views

The **Graphics** tab in the **Template** view allows you to create or modify the graphic views contained in the template.

NOTE:

The best practice for creating and editing graphics for templates is to fully engineer them in a station before making the template. This is especially true if image files are used as they will likely not be visible in the template graphic editor. Images from .jar files will be visible in the template editor. The template graphic editor can be used to make minor changes to the graphic.

The **Graphics** tab is a wrapper view of the **PxEditor** view. The left pane is the Nav pane and the right pane is the PxEditor pane. Selecting a PxFile loads the file into the editor and selects the component(s) in the Nav pane that have a view using the selected PxFile.



Note the toolbar at the top of the PxEditor pane, shown here:



- Step 1 In the **Template** view, click the **Graphics** tab.
- Step 2 In the right pane, click the **PxFile** dropdown and select one of the PxFiles included in the template.
- Step 3 Edit the graphic as needed. You can drag items from the nav pane (left) and drop onto the edit pane (right).

NOTE: All value bindings must be relative bindings. The bindings are forced to be relative to the root component of the template.
- Step 4 Click **Save** or **Save As**.

This saves your changes to the PxFile and saves the configured template, creating a template .ntpl file.

The new template file is saved with the default name or the name entered on the **Template Info** tab and is stored in your Workbench User Home/`templates` directory. Once saved, the template is available from the **Templates** side bar.

NOTE: Be sure to test your template (on another component of the same type) and modify the template as needed before using it to configure other devices, Niagara does not automatically apply subsequent template changes to already-configured devices.

Creating a station template

Create a user-defined station template from a configured station that contains everything needed for the initial starting point of a new station. Once created, the station template is added to the list of selectable templates presented in the Workbench **New Station Wizard**.

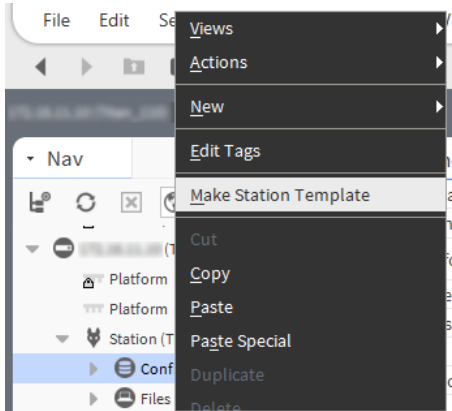
Prerequisites:

- An existing, fully-configured Niagara station suitable for use in a generic new station template

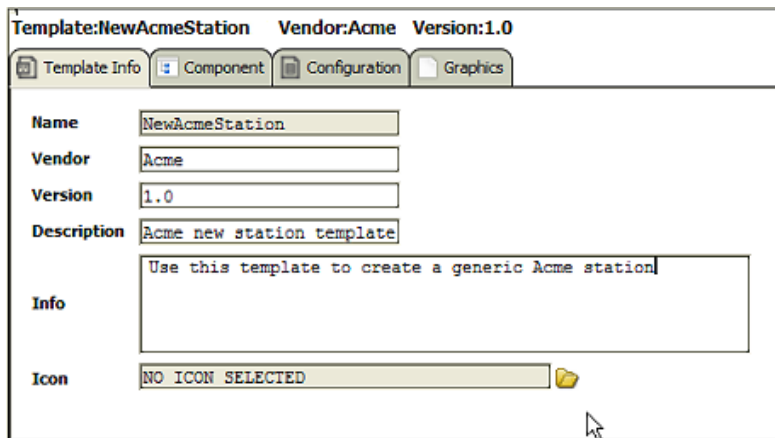
Station templates are automatically saved in the `stationTemplates` sub-directory of your Workbench User Home. User-defined station templates stored in this location are automatically included in the **Station Templates** table shown in the Workbench **New Station Wizard** and are available for selection when using the tool.

This procedure is one the Systems Integrator might perform on a fully configured “basic” station for purposes of reuse and standardization.

Step 1 In the Nav tree, right-click on the station’s Config node and select **Make Station Template** (as shown here) to open the **Template** view.

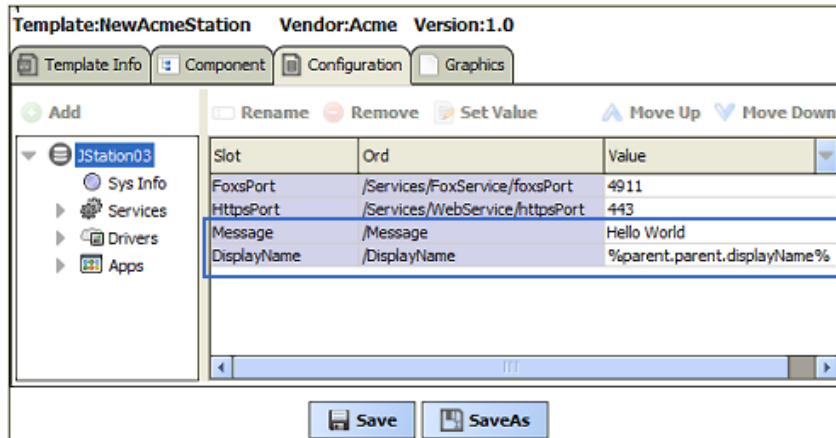


Step 2 In the **Template** view, click each of the tabs to edit the station template as needed.
 a. On the **Template Info** tab, fill-in desired info about the template, as shown here.



- b. On the **Component** tab, make desired changes for the default station template.
- c. On the **Configuration** tab, add configurable parameters as needed. For example, to add the **Foxxs Port**:
 - i. In the left pane, expand **Services**→**FoxService**→**Foxxs Port**
 - ii. Double-click **Public Server Port** to add it to the right pane.
 - iii. Change the default value as needed and click **Set Value**.
 - iv. Click **Rename** and change the name to “Foxxs Port”.

NOTE: The value entered into the stationTemplate for the exposed property becomes the default value upon creating a new station.



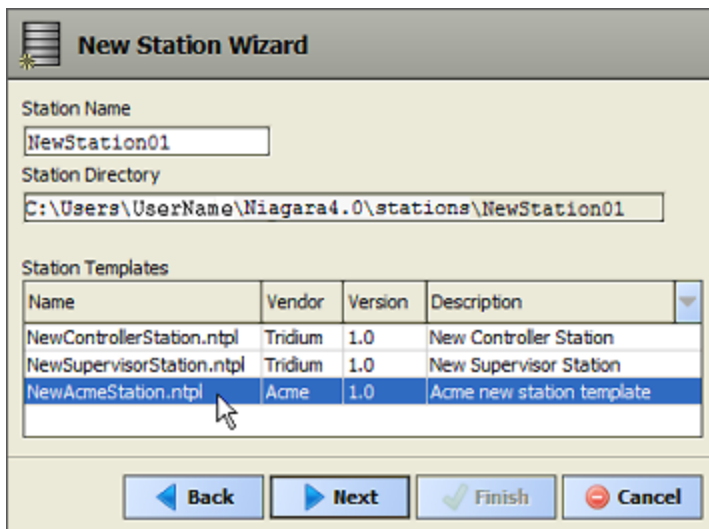
NOTE: Any exposed component properties defined in the template displays as configurable parameters in the **New Station Wizard** when this template is used.

- d. On the **Graphics** tab, make desired changes for the default station template.

Step 3 When finished modifying the template, click **Save**.

The new station template is saved in the `~stationTemplates` sub-directory of your Workbench User Home.

Also, the station template is immediately available for inclusion in the Workbench **New Station Wizard**, where your template appears as a selectable option in the **Station Templates** table as shown here:



NOTE: A Workbench restart is not necessary in order for the user-defined station template to appear in the **New Station** wizard. The file simply needs to be saved to the `~stationTemplates` directory.


Making a template module

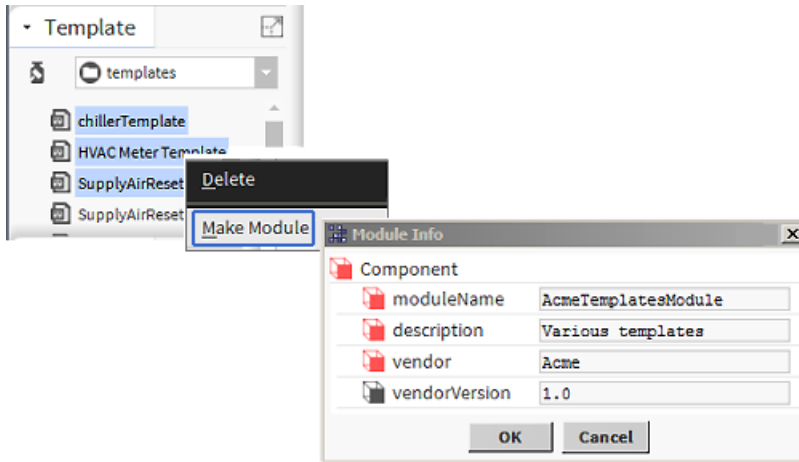
A template module is a group of templates contained in a single `.jar` file that can be distributed as needed, either internally or to customers.

Prerequisites:

- One or more templates (previously created and tested).

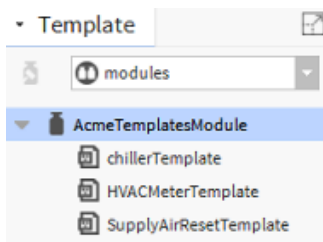
This procedure explains how to group one or more existing templates into a `.jar` file.

- Step 1 In the **Template** side bar, view the **templates** folder.
- Step 2 Select one or more templates, right-click and select **Make Module** or click the  (Make Module) icon in the **Template** sidebar header.
The **Module Info** dialog appears.
- Step 3 Fill in this dialog and click **OK**.



The **moduleName** that you enter in this dialog becomes the `.jar` name as well as the node name in the **Template** side bar.

- Step 4 To view templates in the module you just created, do one of the following:
 - a. In the **Template** side bar, select the **modules** folder from the dropdown list.
Available template modules display in the **Template** side bar.
 - b. In the Nav tree, expand **My Modules** and navigate to your new `.jar` file.
- Step 5 Click to expand the module to see the templates contained within, as shown here.



NOTE:

A template stored in a module cannot be edited. If opened, you will see “ReadOnly” in the top left corner of the **Template** view. In order to make any changes you must first click **Save As** and save it as a new template in the **templates** folder.

Deploying templates

Template deployment is a multi-step copy operation. Once installed, the objects from the template become part of the runtime station and can be modified as needed. You can deploy a template by dragging and dropping from the **Template** sidebar onto the Nav tree, a **Wire Sheet**, or **Property Sheet** view, or by device discovery using **Device Manager** and adding a selected template.

NOTE: Be careful to always match a template to a device for which the template is designed. Do not associate just any template with any device.

Deploying a template via drag-and-drop

Deploy a template on a device by dragging from the **Template** side bar and dropping it onto the device's component space. This is similar to dragging and dropping components from a palette.

Prerequisites:

- An existing template

This procedure demonstrates the deployment steps using a template for a Supply Air Reset component.

Step 1 Right-click on the device and open the wire sheet view where you want to deploy the template.

Step 2 In the **Template** sidebar, click on the template and drag it to the wire sheet view.

The template **Name** dialog appears.

Step 3 Change the template name as needed and click **OK**.

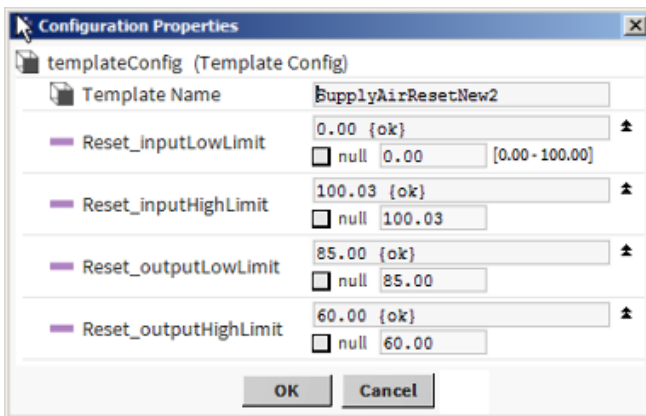
NOTE: Clicking **Cancel** on any of these dialogs aborts the deployment process and removes the installed devices. It does not remove any installed Px or image files.

The **Please Wait** window appears, indicating which files are being transferred to the target station. Additionally, the dialog updates with information on NEQL queries that are done for inputs, outputs, and relations.

NOTE:

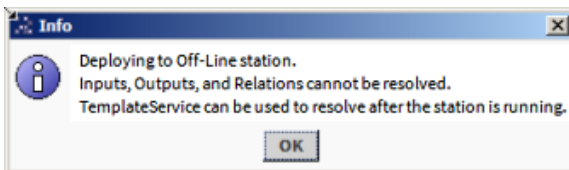
If the template file contains a **Password** with the installed value "BPassword.DEFAULT", a dialog displays alerting the user of this fact. The TemplateService **Template Manager** view can be used to later locate the password and assign a value.

The **Configuration Properties** dialog appears, similar to the one shown here.



Step 4 If you wish, edit the configuration properties and click **OK**.

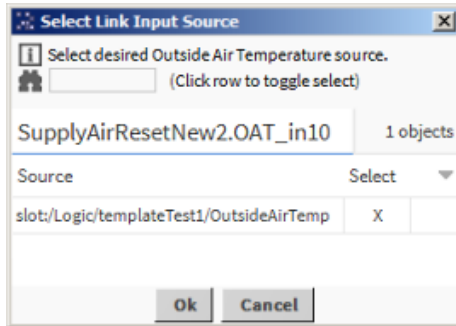
NOTE: If deploying a template to an off-line station the following informational popup appears. At this point deployment is finished. Once the station is running, you can access the Template Service **Template Manager** view to resolve the unbound inputs, outputs, and relations.



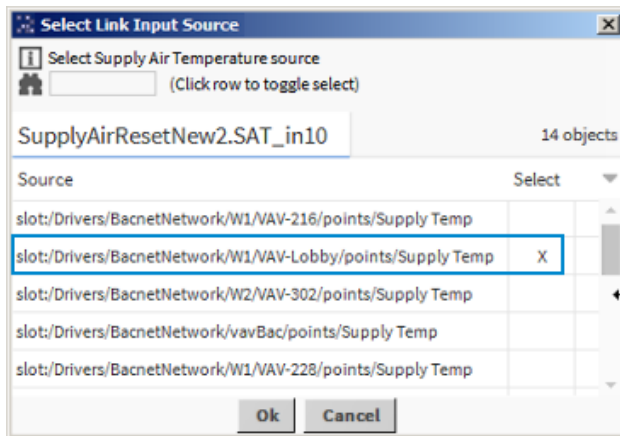
The **Select Link Input Source** dialog appears.

Step 5 In the **Select Link Input Source** dialog, approve or configure input sources (repeat as needed to approve or configure all template input links):

- If an input resolves to a single source it is automatically selected for you, click **OK** to accept the source.



- If the input resolves to multiple possible sources, you must select one and click **OK**.



NOTE: The input source points can be located anywhere in the station. Also, you may sort the list of sources by clicking on the **Source** column heading, as well as filter the list by typing in the search text field.

- If no sources are detected for an input, a popup appears indicating that fact. Click **OK** to proceed.

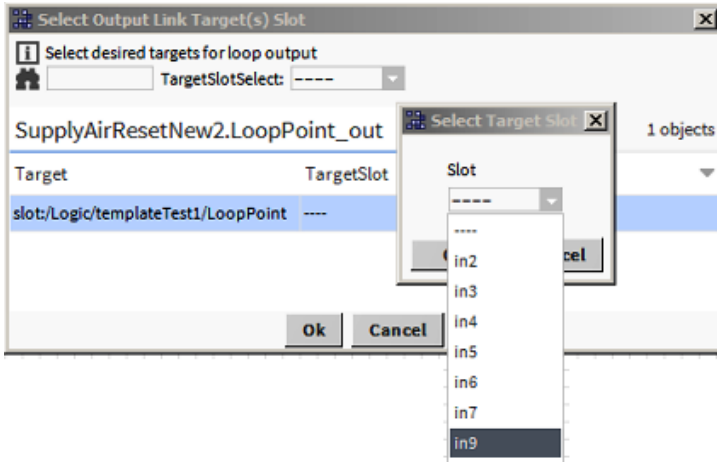
NOTE: In this situation, you may need to create an input source point(s) and tag it appropriately so that the input link resolves correctly. You can either cancel deployment in order to create and tag any undetected input source points and then redeploy the template, or just proceed and after deployment create and tag any missing input sources and then resolve those inputs by accessing the Template Service.

On completion, if the template is setup with one or more output links, the **Select Output Link Target(s) Slot** dialog appears.

- Step 6 In the **Select Output Link Target(s) Slot** dialog, click the **TargetSlotSelect** dropdown or double-click the **TargetSlot** cell (as shown here), click to select a slot and then click **OK** (repeat as needed to configure all output links).

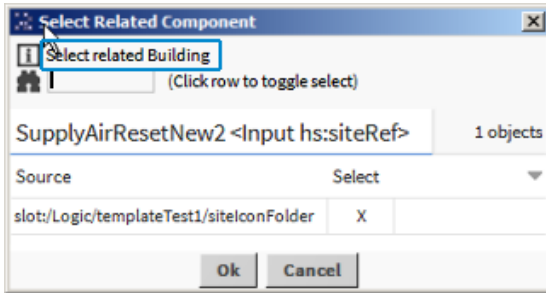
If multiple points are detected, each point will be listed and the target slot selection initialized based on the `targetSlotHint` tag value. If no `targetSlotHint` inputs are linkable, the `----` (do not link) option will be selected. The target slot choices will be limited to the target point's linkable input slots, i.e. if the `in10` input is already linked it will not appear in the choice list. If you do not wish to link to a matching point select the `----` option.

NOTE: A template output can be linked to more than one target.

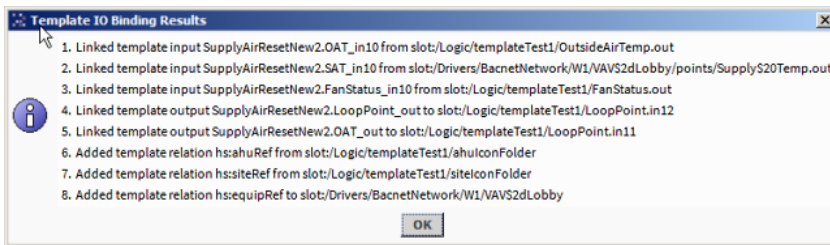


On completion, if the template is setup with one or more component relations, the **Select Related Component** dialog appears.

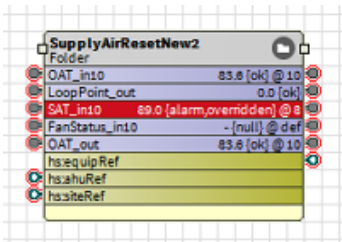
Step 7 In the **Select Related Component** dialog, approve or configure component relations (repeat as needed to approve or configure all template component relations):



Step 8 When finished the **Template IO Binding Results** dialog appears indicating the linked inputs, outputs, and added relations. Click OK.



Template deployment is complete. The new fully configured component is visible on the device's wire sheet.



Deploying a template via the Device Manager

Device Manager provides support for deploying templates whose base component is a Device component. If a driver supports discovery and matching operations then these mechanisms allow template deployment as well.

Prerequisites:


- An existing template
- A network with one or more devices for which the template is designed

Step 1 Navigate to the **Station→Drivers→Network** folder (for example, **NiagaraNetwork**) and double-click on the network to open the **Device Manager** view.

The **Database** table contains all devices that have already been configured.

Step 2 Click **Discover**.

The **Discovered** pane appears listing devices discovered on the driver's network, as candidates for inclusion in the station database.

Step 3 Click  (**Dev Template Mode** button) on the Workbench toolbar, or on the menu bar click **Manager→DevTemplate Mode**.

This opens a **Templates** pane in the upper portion of the **Device Manager** view, displaying a list of templates that are valid for the network.

Step 4 Select one or more devices (of the same type) from the **Discovered** pane.

The **Templates** pane changes to display only those templates that are valid for the selected discovered device(s).

Step 5 In the **Templates** pane, select the template you wish to add for the selected discovered device(s).

NOTE: Be careful to always match a template to a device for which the template is designed. Do not associate just any template with any device, even if it is valid for a discovered device.

Step 6 Click **Add** to start the deployment process. At this point the following occurs:

- The template copies an instance of the template bog file for each selected discovered device into the station. The devices are disabled at this time.
- The template copies a single instance of any Px and image files into the station.
- For each deployed instance, the template queries for the input, output and relation source, target, and related choices .

NOTE: Clicking **Cancel** on any of the following selection dialogs aborts the deployment process and removes the installed devices. It does not remove any installed Px or image files.

- For each input, the template prompts the user to select the source. This dialog will have the option to reuse the selection for this input for all the remaining deployed devices. It will not prompt for this same input for other devices.
- For each output, the template prompts the user to select one or more targets. This dialog does NOT provide the option to reuse the selection.
- For each inbound relation, the template prompts the user to select one or more related sources. This dialog provides the option to reuse the selection for the other deployed devices.
- For each outbound relation, the template prompts the user to select one or more related targets. This dialog provides the option to reuse the selection for the other deployed devices.
- The selected template is deployed for each of the selected discovered devices. A **Template IO Binding Results** dialog appears for each of the selected devices. Click OK to close.

- For each installed device template, the template attempts to match to the corresponding selected discovered device, and enable the device.

Next, the Device Manager **Edit** dialog appears. A list of the added devices displays in a table at the top of the dialog with a tabbed pane below. The **Template** tab allows you to edit the template configuration properties. The **Properties** tab allows you to edit the typical properties for this device.

Step 7 In the **Edit** dialog, select each device in the table and make desired edits (on the tabs) as needed:

- Update any properties that appear on the **Template** tab with the values that are unique for the current device.
- And if not already applied, you can select the **Properties** tab, create a unique name for each device or change its MAC address.

Step 8 After editing each device listed in the upper table, click **OK**.

Template deployment is complete.

NOTE: Unmatched templates can be added to the station by simply double-clicking on a template or dragging and dropping a template on the **Database** table. The user is prompted for template input sources and template configuration changes, as described above. In this case, the device can be manually addressed or matched to a device at a later time.

For drivers that do not support discovery

For drivers that do not support discovery, you can still use the network **Device Manager** view to deploy a template.

1. Click the **DevTemplate Mode** button on the Workbench toolbar, and in the resulting **Template** pane select the desired template.
2. Click **New** and add one or more instances of a device using the selected template.

Editing a template

The **Template** side bar provides access to all template files located in the `~templates`, `~stationTemplates`, and `!modules` directories.

Prerequisites:

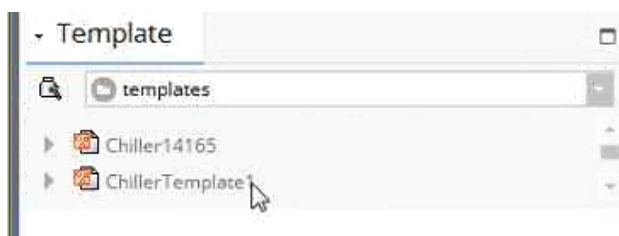
- An existing template

NOTE: Niagara does not apply subsequent template changes to devices which have already been configured using a template.

All templates have a file extension of `.ntpl`, for Niagara template. Template and station template files are stored in the Workbench User Home `templates` and `stationTemplates` directories, respectively.

Template modules have a file extension of `.jar` and are stored in the Sys Home `modules` directory. A template module must be expanded in order to access the template file(s) within.

Step 1 To locate a template, open the **Template** side bar by clicking **Window**→**Side Bars**→**Template**



Step 2 In the **Template** side bar, click the pull down menu and select either: **templates** folder or **modules** folder.

NOTE: To see templates stored in a template module, select the **modules** folder and expand the desired module.

Step 3 Double-click on the desired template.

The **Template** view appears displaying the template configuration tabs with the **Template Info** tab selected.

NOTE: A template stored in a module cannot be edited. When you open it, you will see “ReadOnly” in the top left corner of the **Template** view. In order to make changes you must first click **Save As** and save it as a new template in the **templates** folder.

Step 4 Click the **Configuration** tab in the **Template** view to modify components and properties as needed.

For more details on using the **Template** view **Configuration** tab, see topics on “Creating a template” and “Template reference”.

Step 5 When finished, click **Save** to save your changes to the template.

- Or, click **Save As** to create a new variation of the template with a different filename (leaving the original template unchanged).

Chapter 3 Template reference

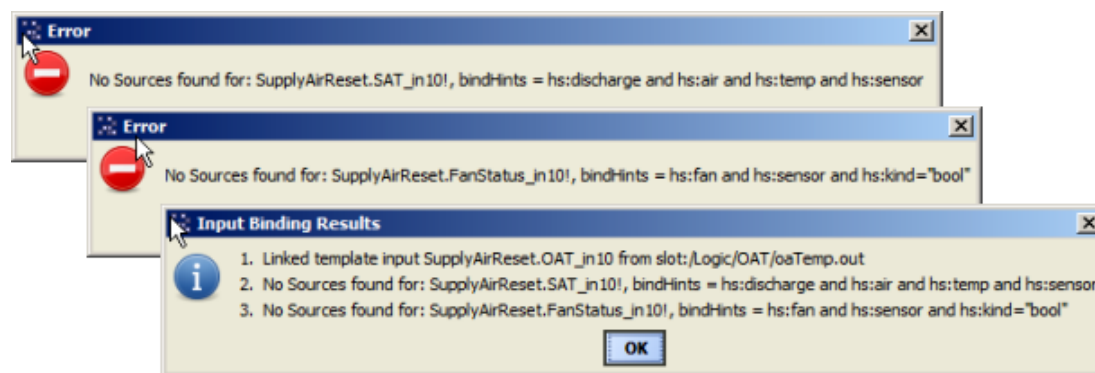
Topics covered in this chapter

- ◆ Template troubleshooting
- ◆ Template side bar
- ◆ Components
- ◆ Plugins (views)

Template troubleshooting

The section provides examples of issues you may encounter using templates and recommended steps to resolve the issues.

While deploying a template you got the following errors but the template deployed anyway:



The template is working correctly. The errors messages alert you to the fact that the deployed template is seeking inputs that have the referenced tags applied however, those inputs are not found. If the necessary input source points are not contained in the deployed template, you may need to add them, apply the specified tags, and resolve the links. The input source points can be located anywhere in the station.

NOTE: Note the same functionality exists for undetected output link targets and component relations.

The Input Binding Results dialog in the above image shows that the template is unable to locate input sources for Reset_SupplyAirTemp and Reset_FanStatus. To resolve those links do the following:

1. In the wiresheet view of the parent of the deployed object, add the necessary input source points, for the first one right-click and add a **Numeric writable point** named SAT.
2. Right-click the added numeric writable point and click **Edit tags**.
3. Select the Haystack tag dictionary and in the search field enter the letter "d" for (discharge air temp).
4. In the resulting tag list, scroll to **Tag Groups** and select the **dischargeAirTempSensor** tag group.

TIP: Using Tag Groups lets you add multiple tags at once. For example, the **dischargeAirTempSensor** tag group applies the following tags: discharge, air, temp, and sensor.

5. Repeat the steps 1–5, this time adding a **Boolean writable point** named FanStatus. In the **Edit Tags** dialog, search the Haystack tag dictionary for tags starting with the letter "f" and from the **Tag Groups** list select the **fanSensor** tag
6. Click **Save**

- Next, navigate to the station's **Template Service** and double-click to open the **Template Manager** view. The unlinked inputs for this template have the Fault background color. Right-click there to resolve the unlinked inputs and click **OK**. Repeat as needed to resolve all unlinked inputs.

The template is successfully deployed once all input links are resolved.

You deployed a template to a device and it did not match the device to the template. So it does not work.

There is a mismatch between the device template and the device you are using. You may have deployed a template that was designed for a different type of device.

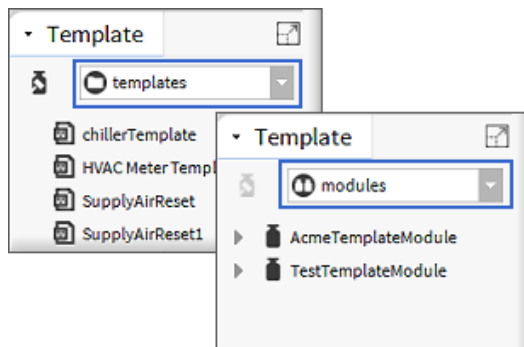
If it is a programmable device, set up with a different collection of points, that doesn't match the template type, it will not work.

Deploying a template using the Device Manager method is useful since it filters and matches the device to appropriate device template(s). To do this, go to the **Device Manager** view and perform **Learn** (identifies devices connected to the network), and **Match** (matches any unbound templated devices).

Template side bar

The **Template** side bar provides access to template files located in the Workbench User Home /templates folder as well as to templates stored in modules located in the SysHome /modules folder.

Figure 1 Template side bar



The pull-down list in the side bar switches the view between the \templates and \modules folders. When the \modules folder is selected, expand any module to see the template files contained within.

Double-click on a template file to open it in the **Template** view. When you open a file in the ~templates folder you can proceed to make changes and save the file. Optionally, you can create a new variation of an existing template by clicking **Save As** in the view to save it with a new name.

NOTE:

Any template stored in a module is a read-only file which you cannot edit. When you open a template in a module, you will see "ReadOnly" in the top left corner of the **Template** view. In order to make changes you must first click **Save As** and save the template with a different filename in the Workbench User Home /templates folder.

Components

Component include services, folders and other model building blocks associated with a module. They may be dragged and dropped onto a Property or Wire sheet from a palette.

The descriptions included in the following topics appear as headings in documentation. They also appear as context-sensitive help topics when accessed by:

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

- Clicking **Help**→**Guide On Target**

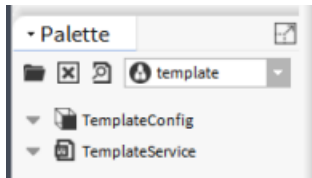
Following is a list of the components in the **template** module:

Template Service

The Template Service provides management support for templates that are deployed in the station. The **Template Manager** is the main view of the **TemplateService**. One of the primary functions of the service is to identify unlinked template inputs, outputs, and component relations and help to resolve them. The Template Service is available in the **template** palette.

NOTE: The TemplateService must be installed in the running station in order to take advantage of the templating feature.

Figure 2 TemplateService available in the template palette



Property	Value	Description
Status [component]	text	Read-only field. Indicates the condition of the component at last polling. <ul style="list-style-type: none"> • {ok} indicates that the component is polling successfully. • {down} indicates that polling is unsuccessful, perhaps because of an incorrect property. • {disabled} indicates that the Enable property is set to false. • fault indicates another problem.
Fault Cause	text	Read-only field. Indicates why the network, component, or extension is in fault.
Enabled [general]	true or false	Activates and deactivates use of the function.
Template Name	text string	A name you choose to identify the entity.

TemplateConfig

The **TemplateConfig** component contains a template’s configuration properties. Composite links are created from slots in the TemplateConfig component to selected component target slots. TemplateConfig can be accessed in the property sheet view of the station’s `config.bog`. TemplateConfig is available in the **template** palette.

Property	Value	Description
Template Name	text string	Name to identify the entity
Exposed properties		Slot for each exposed configured property in template
Linkable I/Os		Slot for each composite link in template
PxView(s)		Slot for each PxView contained in template

Plugins (views)

Plugins provide views of components and can be accessed in many ways. For example, double-click a component in the Nav tree to see its default view. In addition, you can right-click on a component and select from its **Views** menu.

For summary documentation on any view, select **Help→On View (F1)** from the menu or press **F1** while the view is open.

Template Manager view

The **Template Manager** is the primary view of the TemplateService.

The view displays a table of all deployed templates in the station. The table contains the following information for each template:

- location
- name
- vendor
- version
- number of inputs, outputs, and relations

Figure 3 Template Manager view

Path	TemplateName	Vendor	Version	Inputs	Outputs	Relations
/Drivers/LonNetwork/W2/NAVS2d412	vavLon	Tridium	1.0	2	2	1
/Drivers/LonNetwork/W2/NAVS2d413	vavLon	Tridium	1.0	2	2	1
/Logic/templateTest1/SupplyAirReset	SupplyAirResetNew2	Tridium	1.0	3	2	3
/TemplateTest1	SupplyAirResetNew	Tridium	1.0	3	1	3

NOTE: If a deployed template has any unbound inputs, outputs, or relations, the view displays those table cells with the Fault background color, as shown in the above image. Right-clicking one of the Fault rows displays an additional command, **LinkUnboundIO**. Selecting this steps through all the unlinked inputs, outputs, and relations.

The **TemplateManager** view provides the following functionality:

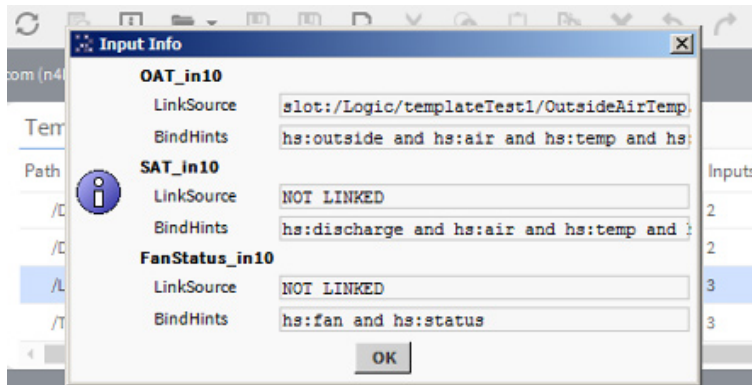
GoTo navigation

Path	TemplateName	Vendor	Version	Inputs	Outputs	Relations
/Drivers/LonNetwork/W2/NAVS2d412	vavLon	Tridium	1.0	2	2	1
/Drivers/LonNetwork/W2/NAVS2d413	vavLon	Tridium	1.0	2	2	1
/Logic/templateTest1/SupplyAirReset	SupplyAirResetNew2	Tridium	1.0	3	2	3
/TemplateTest1	SupplyAirResetNew	Tridium	1.0	3	1	3

Selecting **GoTo** navigates to the default view of the root component of the deployed template. Invoke the popup by double-clicking on the desired path cell or right-click on the desired path cell and then click on the **GoTo** popup.

Selecting **GoTo Parent** navigates to the default view of the template root component's parent.

Informational popups



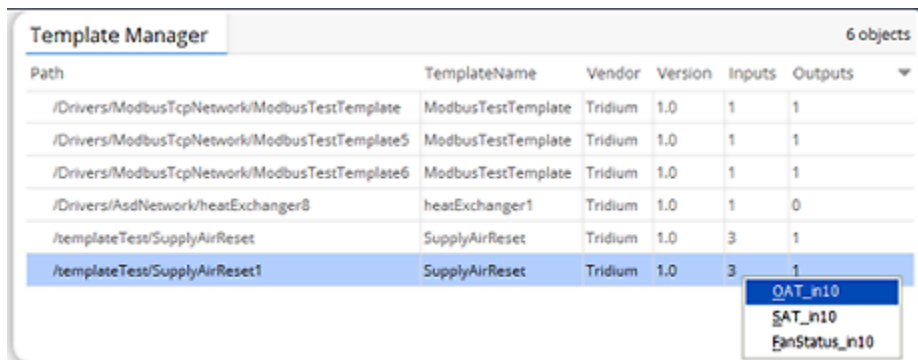
Read-only informational popups can be invoked by clicking table cells in the Inputs, Outputs, or Relations columns in the view. For example, the **Input Info** popup provides you with additional information about the inputs of the deployed template. Invoke the popup by double-clicking an **Inputs** table cell.

In the popup, an unlinked input status is indicated as “NOT LINKED” in the **LinkSource** field, as shown.

Double-clicking either an Outputs or Relations table cell invokes the **Outputs Info** and **Relation Info** popups respectively. Referring to the Info popups can be useful when attempting to resolve links and relations. For example, to resolve an unlinked output you can check the **bindHints** shown in the popup to determine which tags you may need to add to a link target point.

Resolving unbound template I/Os or relations

The **Template Manager** view assists you in linking any unlinked template inputs, outputs, or adding component relations. Right-clicking any cell indicating Fault color selects the row and invokes a popup that lists the unlinked items, shown here.



1. Click an input in the popup to resolve the link.
2. In the **Select Link Input Source** dialog, approve or configure input sources as described here:
 - If the selected input resolves to a single source it is automatically selected for you, simply click **OK** to accept it.
 - If the input resolves to multiple possible sources, you must select one and click **OK**.
 - If the selected input cannot be resolved, a popup appears indicating that fact. Click **OK** to proceed. In this situation, you may need to create an input source point(s) and tag it appropriately so that the input link resolves correctly.

NOTE: The same functionality exists for linking unlinked outputs and relating components.

Batch resolution of unlinked inputs

Provides a means of selecting multiple deployed templates and attempting to link all of the selected templates with unlinked inputs or outputs.. To invoke this functionality:

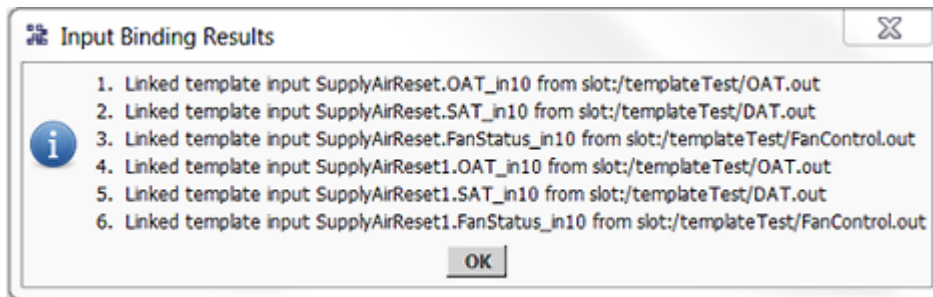
1. Select multiple rows in the **Template Manager** view and then right-click one of the selected rows.

The **LinkUnboundIO** popup displays.

2. Click on this to begin the linking process.

NOTE: During this process the system remembers the choices made by the user for each template type and input. If there are any additional template instances of the same type, the system uses the same choices rather than repeatedly prompting for the same information.

On completion, a popup dialog displays indicating each linked input and its source.



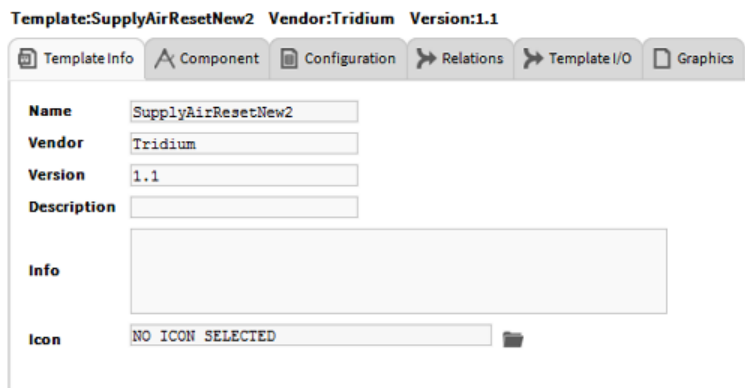
3. Click **OK** to close the dialog.

Template view

The **Template** view is the primary view for creating and editing templates. Invoked when you right-click a component and select **Make template** or double-click an existing template. The tabs present in the view allow you to configure template *.ntpl files.

These tabs are used during the creation and editing process or to make template modifications.

Template Info tab



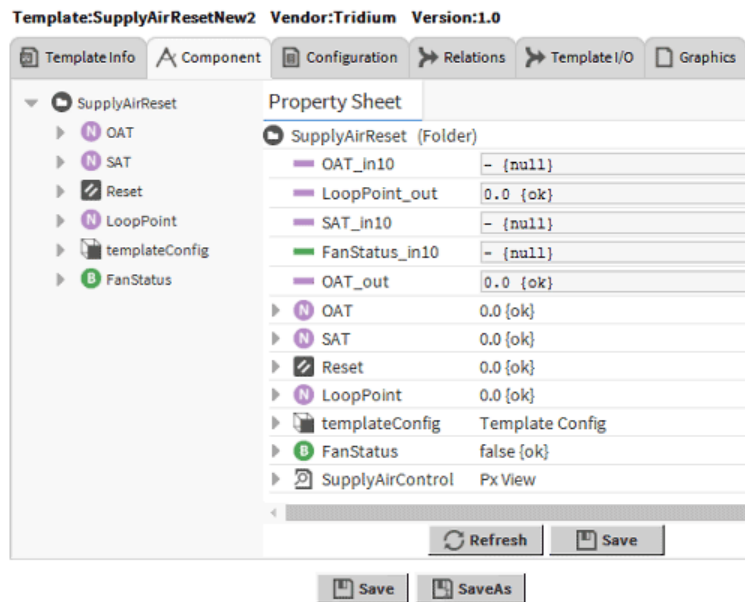
In addition to the properties listed below, any of these notices may display on the **Template Info** tab:

- If the template component tree contains links whose source or target components are not contained within the same component tree, the editor automatically generates template inputs and outputs to match each. Further, the template editor automatically sets the **bindHints** for each input or output from the marker tags that are present on the external components. A notice alerts you to review the added links in the **Template I/O** tab.

- If the template root component is a Device component, a notice alerts you that (1) this is a new device template, (2) to review the **Component** tab, and (3) to set the device address property to its default value. This is important because the device address property and its default value are device specific. This check helps to ensure that when this device template is deployed using the Device Manager view that the "Match" operation is successful.
- If the template component tree contains a **Password** property, a notice alerts you that the template has internal passwords that may need to be set during deployment. For security purposes, when passwords are copied from a template to a station, the password value may be set to `Password.DEFAULT`. In this case, during deployment the user must assign a valid value to the **Password** property.

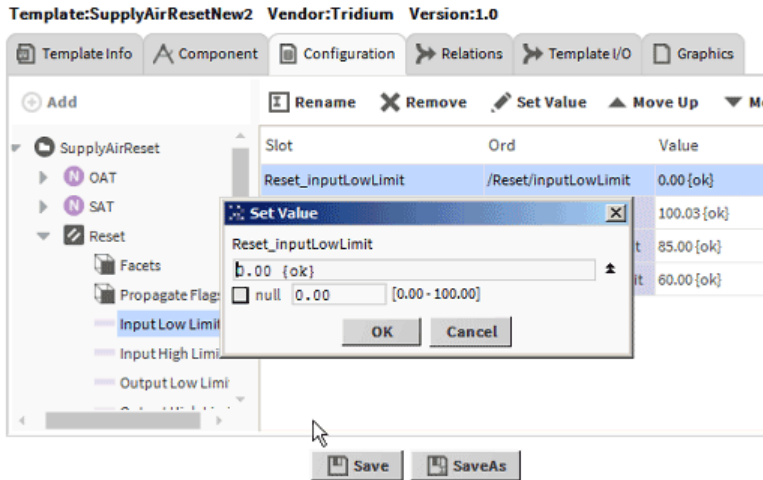
Type	Description
Name	The template file name, which can be modified during the creation process. Once created, this name is read-only.
Vendor	Optional name of the device vendor who is responsible for the design and creation of the template.
Version	Optional template version number. Numeric characters only. This number defaults to 1.0.
Description	A short description (up to 25 characters) used to distinguish similar templates. This name is displayed in the Device Manager template side pane.
Info	Any additional explanation to describe the template. Multiple lines are allowed. This text is visible in the template manifest.
Icon	Any .png file to associate with the template. This icon displays in the Device Manager template side pane.

Component tab



This tab shows the `config.bog` settings for the base component and its descendents. The tab contains two panes. The left pane shows a tree view of the components that is similar to the Nav sidebar. The right pane shows a Property sheet view of the component.

Configuration tab

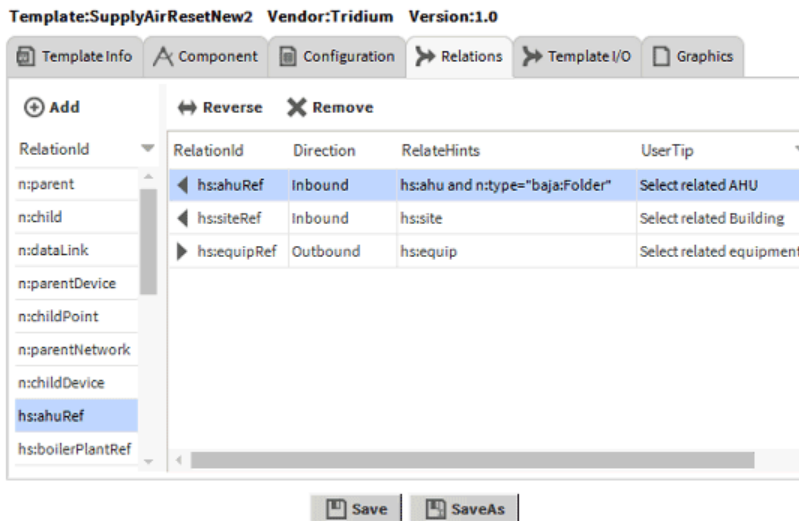


This tab allows the designer to configure the properties that are presented to the user. During deployment, an edit popup prompts the user to set values for specific properties.

This tab contains two panes. The left pane shows the template’s component tree. The right pane shows the current set of configuration entries.

The configuration entries result in properties of the same type in the `templateConfig` object on the base component.

Relations tab

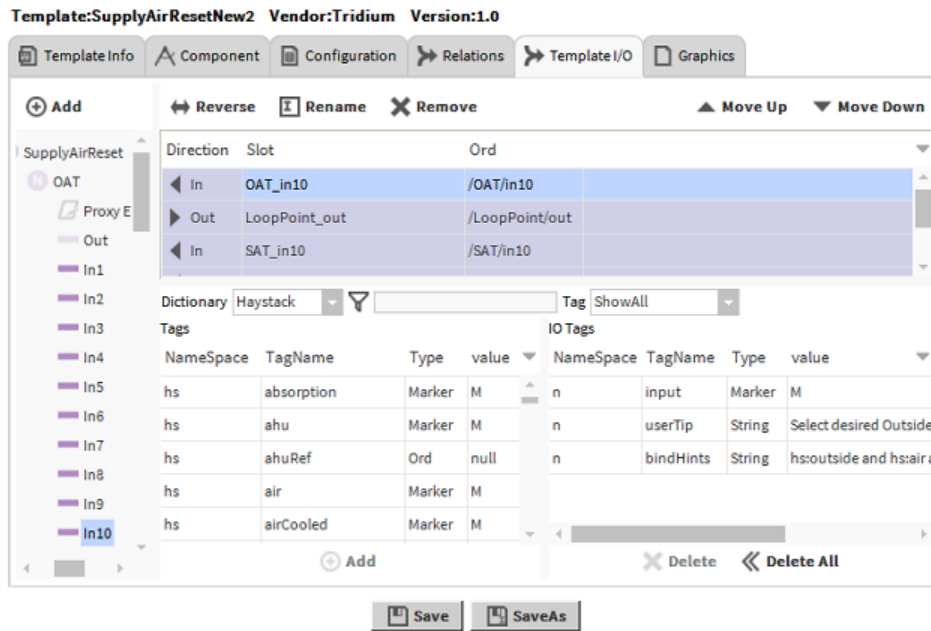


This tab allows you to define relations between the root component of this template and other components in the station in which the template is deployed.

The tab contains two panes. The left pane provides a list of all Relations Id's found in the installed tag dictionaries. The right pane displays a table listing any relations that have been added to the template. Use the buttons above the panes to **+** Add , **↔** Reverse , and **✕** Remove relations.

You can add any relations which are defined in the tag dictionaries installed on your system by selecting from the left pane and clicking **Add**. Optionally, you can add Ad Hoc relations by clicking **Add** without first making a selection, and defining the RelationId in the **Add** dialog. By default, the Relation Direction of an added relation is “inbound”. Clicking **Reverse** changes the relation direction of the selected Relation. Select a relation row in the right pane and double-click the **RelateHints** value cell to enter a NEQL search predicate. This NEQL search is used during template deployment to search for and suggest potential related components.

Template I/O tab



This tab allows the designer to manage linkable control points in the component tree of a device template. You can select and expose control inputs that are required by the template and control outputs provided by the template. Also, you can add and remove tags on the control points. The primary purpose for tagging the inputs and outputs is to assist the installation tool during deployment in resolving links.

NOTE: This tab is intentionally omitted from station templates since you cannot configure I/Os for a station template.

The Template I/O tab contains three panes. The left pane shows the root component and the child control points. The upper right pane shows the Slot sheet listing the template’s current control inputs and outputs. The lower right pane allows application of tags to the I/O points.

In the nav pane you can click to expand control inputs, select one and click **Add** to add it to the upper right pane, exposing that input for linking or you can simply double-click on it in the left pane to have the same effect.

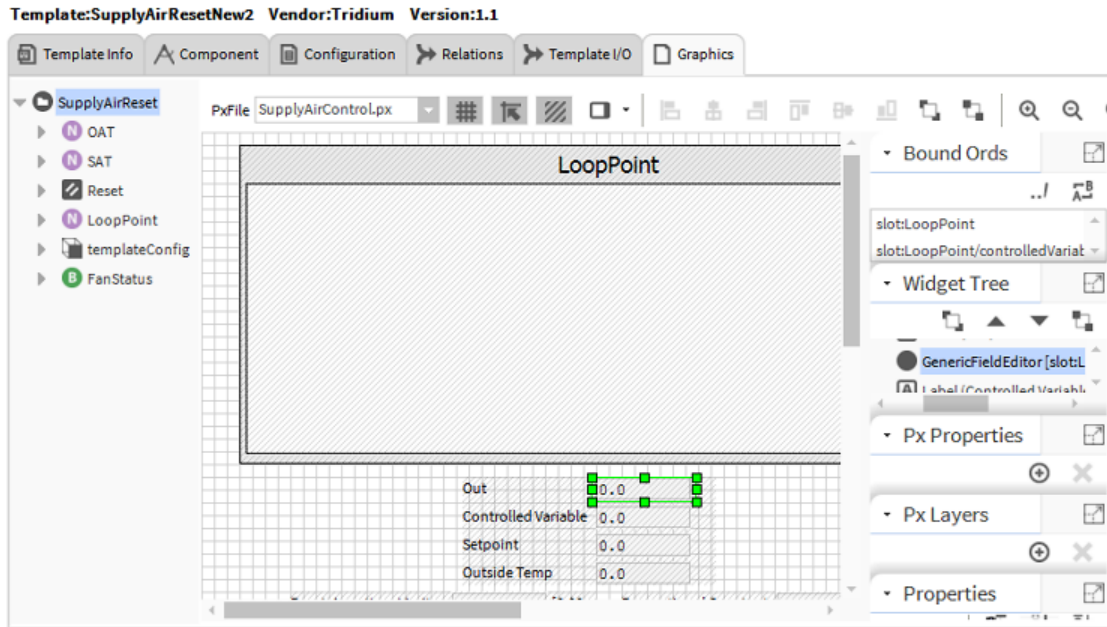
Selecting a control input in the upper right pane displays (in the lower right pane) any existing tags on that input. If desired, use delete options in the lower right pane to remove tags. Use the lower left pane to select additional tags from available dictionaries to add to the input that is currently selected in the upper right pane.

Exposing control outputs is performed in the same manner as inputs.

Graphics tab

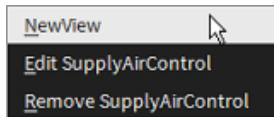
The **Graphics** tab is a wrapper view of the **PxEditor** view. The left pane is the nav pane and the right pane is the pxEditor pane. If there are multiple Px files contained in the template the **PxFile** dropdown list allows you to choose the PxFile to edit.

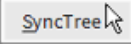
Selecting a PxFile loads the file into the editor and selects the component(s) in the nav pane that have a PxView component referencing the selected PxFile.



Nav tree controls

- Right-clicking on a nav component that already has a view defined displays a popup menu. From the menu, you create a **NewView**, **Edit** a view or **Remove** a view. If the nav component has multiple views defined there will be multiple Edit and Remove commands listed, one for each view.



- Right-clicking in the nav pane white space provides a  command, which selects components in the tree that have a view using the currently selected pxFile.

Glossary

namespace	A container for a set of names in a naming system. A tag dictionary is a namespace.
NEQL	Niagara Entity Query Language provides a simple mechanism for querying objects with tags. Whereas BQL supports the tree semantics and pathing of Workbench component space (for example parent.parent) and BFormat operations, NEQL queries only for tags using the Niagara 4 tagable and entity APIs.
niagara_user_home	<p>The configuration location(s), under which all configurable data resides. Included are stations, templates, tag dictionaries, registry, logs, and other data. Unique to Niagara 4, the Windows host usually has at least two niagara_user_homes:</p> <ul style="list-style-type: none"> • Workbench User Home, which you see in Workbench as “User Home”. • “Daemon” user home, which you access using a local platform connection. <p>For more details, see “Niagara 4 directory (homes) architecture” in the <i>Niagara 4 Platform Guide</i>.</p>
relation	<p>A piece of semantic information (metadata) that indicates how components are related to each other.</p> <p>Relations provide metadata used primarily in building hierarchies for logical views of your system based on relationships between components. You add relations between components for purposes of building hierarchies. Optionally, adding one or more tags to a relation provides additional metadata which allows for more specific filtering when building hierarchies.</p>
station template	A station template is a specialization which contains a complete set of configured objects, everything required for the initial starting point for a new station. The New Station wizard in Tools utilizes default station templates (NewControllerStation.nptl and NewSupervisorStation.nptl) as well as any user-defined station templates.
Tag	<p>A piece of semantic information (metadata) associated with a device or point (entity). Tags identify the purpose of the component and its relationship to other components. For example, JACEs are associated with Supervisors based on tags; searching is done based on tags.</p> <p>Tags are contained in tag dictionaries. Each tag dictionary is referenced by a unique namespace.</p>
tag dictionary	Tag dictionaries contain a set of tag definitions, and may contain tag group definitions, relation definitions, as well as tag rules for smart tags.
template	A deployable package of Niagara objects used to streamline repetitive configuration steps when making multiple installations with similar functionality. For example, when setting up a new device by deploying a device template, only unique device properties require configuration. Templates are indexed and searchable.
template designer	The template designer is the person who creates and maintains a template.
template user	The template user is the person who deploys a template to configure an installed instance.