Technical Document

Relations Guide

April 19, 2016

niagara4

Relations Guide

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

Confidentiality

Trademark notice

Copyright and patent notice

Contents

About this guide

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

Product Documentation

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

Document Content

This document provides an introduction to concepts and procedural information about the relations feature in Niagara 4.

Document change log

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

- April 19, 2016: Document updated to remove several procedures that are not needed.
- July 25, 2015: Initial release document

Related documentation

The following documents may relate to the content in this guide and provide additional information.

- *Tagging Guide*
- *Templates Guide*
- *Hierarchies Guide*

Chapter 1 Using Relations

Topics covered in this chapter

- ♦ Adding a Relation using Relation Mark
- ♦ Adding a Relation using drag and drop
- ♦ Editing a Relation from the Relation Sheet
- ♦ Editing a Relation from the Wire Sheet

Relations provide metadata used primarily in building hierarchies for logical views of your system based on relationships between components. So that you can organize the display of those components in various meaningful ways.

Relations Concepts

Regardless of the actual structure of a system, you can define a hierarchy that includes "relation level definitions" which query for certain tags and relationships. Provided components are already tagged and relations already setup, executing the hierarchy definition results in a specific navigation tree hierarchy. For example, the resulting hierarchy groups all the variable air volume (VAV) controllers by the air handler units (AHUs) supplying them.

In summary, 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.

Common Relations Tasks

Common relations tasks include adding relations between one or more components. Also, you can edit existing relations.

Typical relations tasks are described in the following sections:

Adding a Relation using Relation Mark

You can add a relation between components using the **Relation Mark** menu option and then selecting either **Relate From** or **Relate To** from a popup menu.

Step 1 Select one or more components to mark for either the "From" relation or the "To" relation component(s).

Step 2 Right-click on the selection and click **Relation Mark**.

Step 3 Select one or more components to use for the other side (From/To) of the relationship.

Step 4 Perform one of the following, depending on your choice in step 2:

- If the mark was for the "Relate From" components, right click on the selection and click **Relate To**.
- If the mark was for the "Relate To" components, right-click on the selection and click **Relate From**.

A Relation dialog appears.

Step 5 Select the desired **Relation Id** form the pull-down list.

NOTE: This list is populated from **RelationInfoLists** defined in the **Tag Dictionaries** installed in the **TagDictionaryService**.

Opening the **Relation Sheet** view of the components included in the relationship shows this added relation.

Adding a Relation using drag and drop

You can add a relation between two components within a Wire Sheet simply by dragging a connector from lower end of one component to the lower end of another.

Step 1 In a Wire Sheet view, click the Workbench **Wire Sheet** menu and click **Show Relations**.

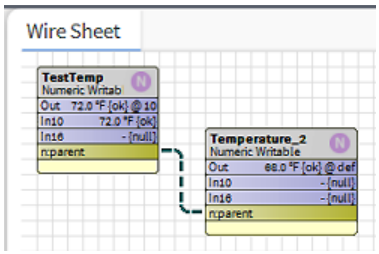
Step 2 On the desired "Relate To" component, click the component's footer and drag to the component footer for the desired "Relate From" component.

A Relation dialog appears.

Step 3 From the pull-down list, click on the desired `Relation Id` and click OK.

NOTE: This pull-down list is populated from the Relation Definitions defined in the selected Tag Dictionary.

In the Wire Sheet view, the relation is shown as a dashed line between the two components. Both of the components have the Relation Id for the slot displayed name.



Editing a Relation from the Relation Sheet

You can modify an existing relation by invoking the **Edit** dialog from the component **Relation Sheet** view.

Prerequisites:

- Component with existing relations

Step 1 Select the desired component and open the **Relation Sheet** view.

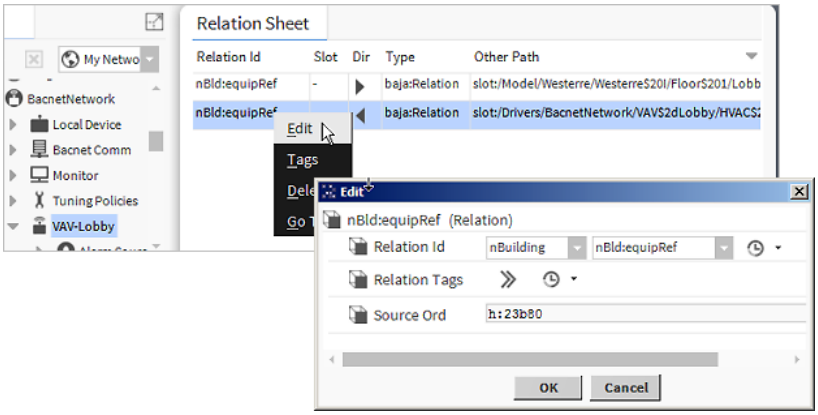
Step 2 Right-click on the desired relation row and click **Edit**.

Step 3 In the **Edit** dialog, make any of the following changes:

- Change either the tag dictionary or tag name referenced in the Relation Id
- Change the Source Ord

NOTE: Although Relation Tags are persisted as BFacets type and can be edited in this dialog, the recommended method for editing relation tags is via the **Relation Tags** dialog (in the **Relation Sheet** view, right-click on the relation row and click **Tags**).

Step 4 Click **OK** to save your changes.



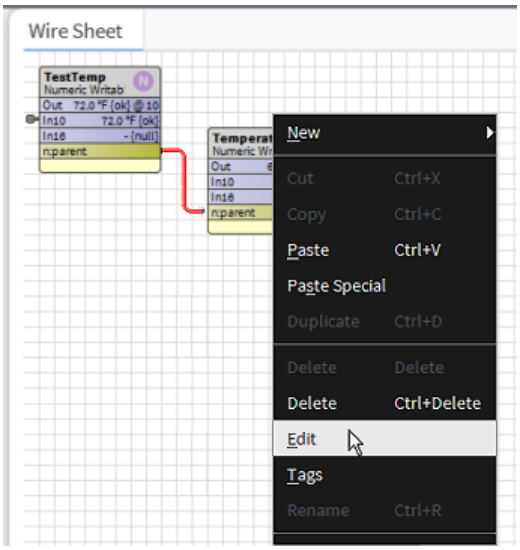
Editing a Relation from the Wire Sheet

You can modify an existing relation by invoking the **Edit** dialog from the component **Wire Sheet** view.

Prerequisites:

- Component with existing relations

Step 1 In the **Wire Sheet** view, right-click on a dashed relation line and click **Edit**.



Step 2 In the **Edit** dialog, make any of the following changes:

- Change either the tag dictionary or tag name referenced in the Relation Id
- Change the Source Ord

NOTE: Although Relation Tags are persisted as BFacets type and can be edited in this dialog, the recommended method for editing relation tags is via the **Relation Tags** dialog (in the **Relation Sheet** view, right-click the relation row and click **Tags**).

Step 3 Click **OK** to save your changes.

Chapter 2 Relations Reference

Topics covered in this chapter

- ♦ Entity-Relationship Modeling
- ♦ Types of Relations
- ♦ Relation Id Structure
- ♦ Relation Definitions
- ♦ Support for Relations

Entity-Relationship Modeling

An entity is any identifiable object (point, device, etc.) that exists independently. A relationship captures how one entity relates to another. For example, in your system you might have an AHU device that supplies air to a specific VAV device.

Using an English grammar analogy, entities can be thought of as *nouns*, while relationships can be thought of as *verbs* connecting two or more nouns (entities). For example: where AHU1 supplies VAV1, both AHU1 and VAV1 are nouns (entities) and "supplies" is the verb (relationship). Taking it further, AHU1 (the subject) supplies (the verb/predicate) VAV1 (the object).

You can also determine the relationship direction by examining the structure of the relationship. From the context of the first entity (the subject), the relation is an "outbound" relation. While from the context of the second entity (the object) the relation is an "inbound" relation.

In the Component space, a `BComponent` is an entity, and `BRelation` is used to declare a relationship from one BComponent to another BComponent. The *from* component is the subject of the relation and the *to* component is the object of the relation. A relation is created by adding a dynamic BRelation slot on the object of the relationship. A "BLink" is a specialized type of relation that defines a "data-flow" relationship between a value slot of one component to a value slot of one or more other components.

Types of Relations

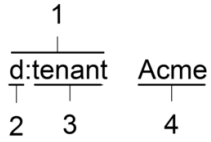
Relations can be either "direct" or "implied" and are defined as follows:

- **Direct relations** are relations that you apply directly to a component. When adding a relation, your choices are limited to relations that are defined in the any of the Tag Dictionaries on your system.
- **Implied relations** are determined automatically and applied to a component by the system. Implied relations are defined in a SmartTagDictionary under its Tag Rules folder (BTagRuleList). When an application queries for the relations on a component in the station, the SmartTagDictionary executes code that interprets the Tag Rules against the given component and returns a list of implied relations.

Relation Id Structure

A Relation Id contains different parts that, together, make the ID useful as additional information on objects in a station. The following diagram shows the basic parts of a Relation Id.

Figure 1 Parts of a Relation Id



1. Relation Id
2. Tag Dictionary
3. Tag Name
4. Tag Value

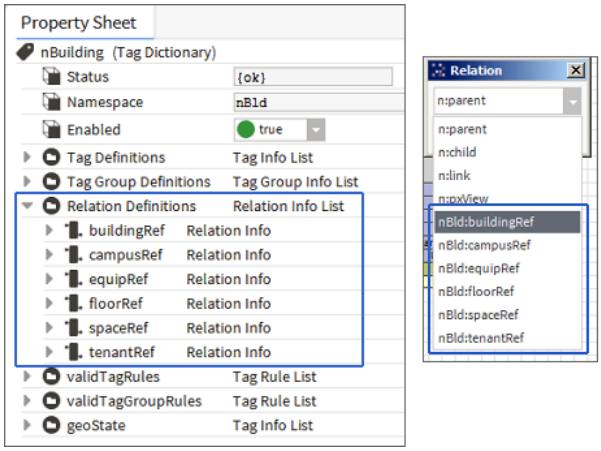
The following table provides definitions of the different parts of a Relation Id:

Relation Element	Description
Relation Id	The Relation Id is comprised of a dictionary and name, generally displayed as two pieces of text separated by a colon (:), as shown in the following example: `<dictionaryNamespace>:<name>`.
Tag Dictionary	The dictionary string is used to link or assign a tag to a particular "namespace" (tag dictionary). This is typically a very short string of one or two characters.
Tag Name	The name string provides the semantic information and is often paired with the Tag Value.
Tag Value	A string value assigned to the tag for more information, for example: building name, device name, location, or other.

Relation Definitions

Tag Dictionaries often contain a collection of Relation Definitions (shown in the following image) which are standardized Relation Id's with semantic meaning for a given domain or namespace. These relation definitions come into play when adding a relation to a component. In the Relation dialog, your choices are limited to the relations that are defined in the any of the Tag Dictionaries on your system.

Figure 2 Relation Definitions in custom Tag Dictionary (left) provide choices seen in the Relation dialog (right)



Support for Relations

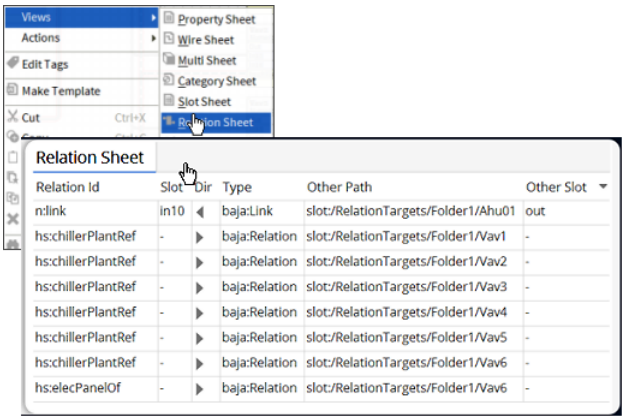
Workbench support for relations is evident in the **Relation Sheet** and **Wire Sheet** views. In either view you are able to create, edit or remove relations. Also, the **Spy** view includes relations information.

The workflow for creating and editing Relations in Workbench is very similar to that of creating and editing links in previous versions of Workbench. A `BRelation` slot is added to the component that is the inbound side of a relation. The outbound Relation component will have a **RelationKnob**, which is similar to a Link-Knob.

Relation Sheet view

The **Relation Sheet** is the main view for managing relations on a component. The view displays the relations of a selected component as well as any links, as shown here.

Figure 3 Relation Sheet view shows relations and links



NOTE: The **Relation Sheet** view replaces the Link Sheet view available in NiagaraAX.

The **Relation Sheet** view displays the following information:

Name	Description
Relation Id	The relation ID for this relation.
Slot	The connecting slot of this component for a link. Applies to links only.
Dir	Indicates the direction of this relation. Options are either an inbound target or an outbound source.
Type	Indicates the class type of the relation.
Other Path	The slot path to the other related component.
Other Slot	The connecting slot of the other component for a link.
Enabled (hidden)	Indicates whether this link is currently enabled. A “relation” is always enabled.
Tags (hidden)	Tags can be applied to a relation. Any applied tags are shown here.

Although hidden by default, the **Enabled** and **Tags** columns can be exposed in the view by clicking on the pull-down list located at the far right of the column heading row, and then clicking to select each of these column heads.

Popup menu commands

Right-clicking a row in the **Relation Sheet** view invokes a popup menu with the following options:

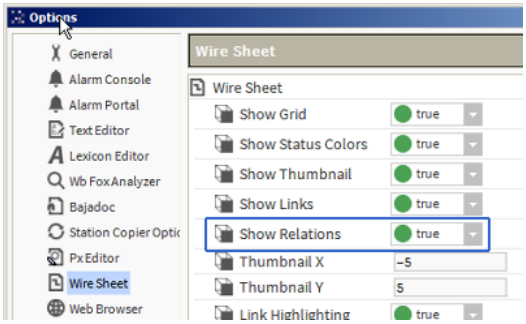
Option	Description
Edit	Edit the selected relation or link.
Tags	Edit the tags applied to the selected relation or link.
Delete	Delete the selected relation or link.
Go To	Go to the main view of the selected relation.

Wire Sheet view

Workbench has a **Wire Sheet Relation** mode when selected component relations are displayed on the wire sheet. This option can be selected by any of the following methods:

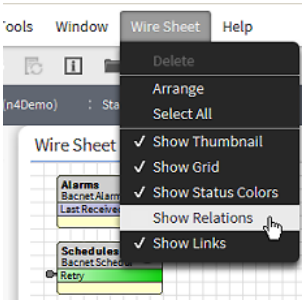
- Workbench **Tools→Options→Wire Sheet→Show Relations**

Figure 4 WorkbenchTools option



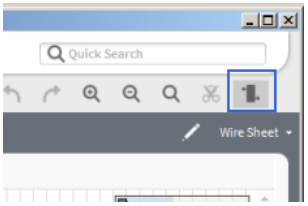
- In a Wire Sheet view, select **Wire Sheet→Show Relations**

Figure 5 Wire Sheet menu option



- On the Workbench Tool Bar, click the **Show Relations** icon

Figure 6 Workbench Tool Bar icon

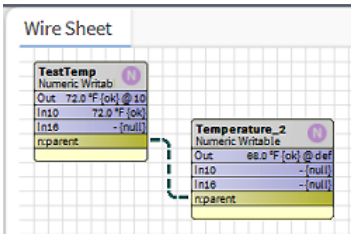


When invoked, the **Wire Sheet Relation** mode displays relations of the selected component as either of the following:

- **On-sheet relations**

  This situation exists when both components of a relationship are present on the current wire sheet, the relation is shown as a dashed line connecting the two components. The `Relation Id` is in the name of the colored bar of the component glyph. The relation line exits the right side of the subject (outbound relation component) and enters the left side of the object (inbound relation component).

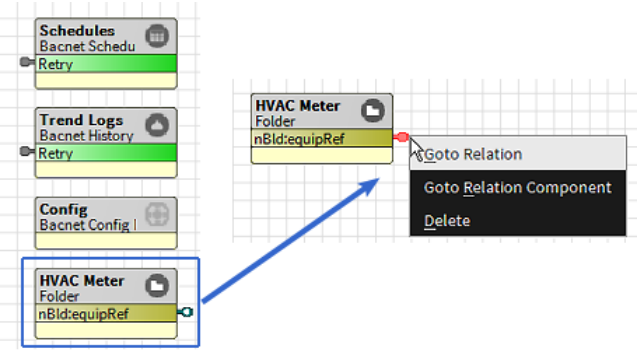
Figure 7 On-sheet relation depicted with dashed line



- **Off-sheet relations**

  This situation exists when the other component of a relationship is NOT present on the current wire sheet. The relation is depicted with a relation "stub" recognizable by the hollow stub appearance (as shown in the following image), distinguishable from a "link" stub which has a solid appearance. Access the main view of the other component in a relationship by right-clicking the relation stub and selecting **Goto Relation**. Also, double-clicking the relation stub switches to the **Wire Sheet** view of the other component in the relationship.

Figure 8 Off-sheet relation depicted with hollow stub



Index