Technical Document

Relations Guide

August 18, 2015

niagara4

Relations 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

Contents

About this guide

This document provides tasks and conceptual information about the relations feature in Niagara 4.

Document change log

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

Initial release document: August 18, 2015

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 RelationMark
♦ Adding a Relation using drag and drop
♦ Editing a relation from the Relation Sheet
♦ Editing a relation from the Wire Sheet
♦ Adding/editing/removing relation tags

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 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 and relations tags.

Typical relations tasks are described in the following sections:

Adding a Relation using RelationMark

You can add a relation between components using the **RelationMark** menu option and then selecting either **RelateFrom** or **RelateTo** 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 **RelationId** 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**.

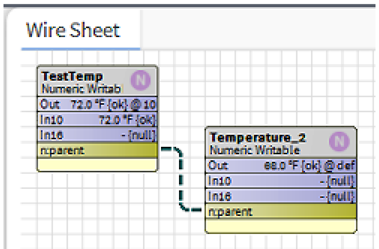
Step 2 On the desired “relate to” component, click the component’s footer and drag to the component footer for the desired “relate from” component.

A Relation dialog appears.

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

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

In the Wire Sheet view, the relation is shown as a dashed line between the two components. Both of the components have the relationId 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.

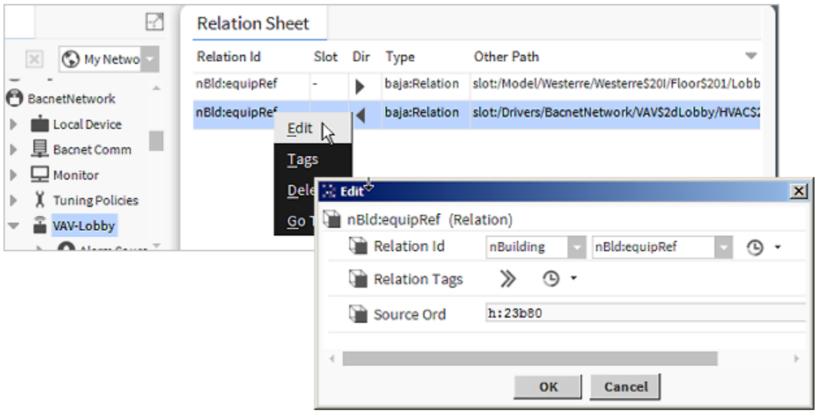
Step 2 Right-click on the desired relation row and click **Edit**.

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 the relation row and click **Tags**).

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



Relation Id	Slot	Dir	Type	Other Path
nBld:equipRef	-	▶	baja:Relation	slot:/Model/Westerre/Westerre$20I/Floor$201/Lobb
nBld:equipRef		◀	baja:Relation	slot:/Drivers/BacnetNetwork/VAV$2dLobby/HVAC$:

nBld:equipRef (Relation)

Relation Id: nBuilding, nBld:equipRef

Relation Tags

Source Ord: h:23b80

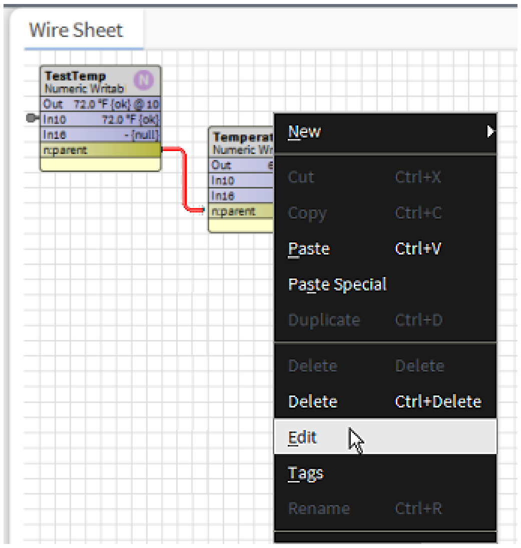
Editing a relation from the Wire Sheet

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.

Adding/editing/removing relation tags

Adding tags to a relation adds metadata which can be targeted by RelationLevelDefs in hierarchy definitions. You can add or remove direct tags on existing relations, as well as edit direct value tags.

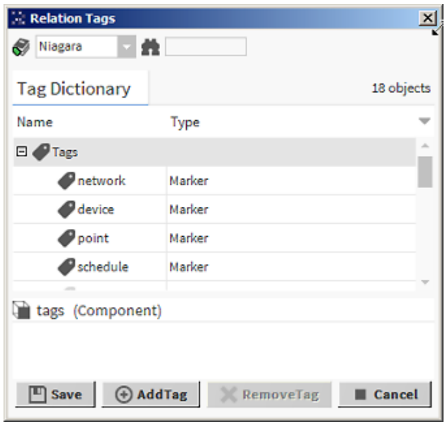
NOTE: Tags on relations are distinctly separate from component tags. Both types of tags contain metadata however the tools that consume the data most likely are not the same.

Relations usually have one or more “implied” tags which are automatically applied by the system. Any tags that you add to a relation are referred to as “direct” tags. Any direct tags added to a relation display in the lower **tags** pane of the **Relation Tags** dialog. These direct tags are the only ones that you can edit (if value tags) or remove.

The **Relation Tags** dialog, shown here, allows you to do the following:

- Add tags from installed TagDictionaries
- Add Ad Hoc tags which you define as needed
- Edit direct value tags
- Remove direct tags

Figure 1 Relation Tags dialog



Name	Type
Tags	
network	Marker
device	Marker
point	Marker
schedule	Marker

tags (Component)

Adding a tag from a tag dictionary

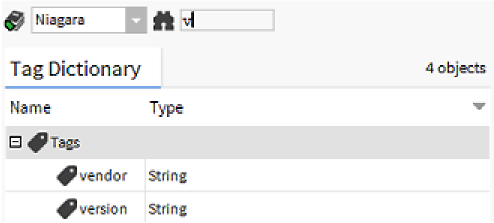
Step 1 Right-click on a relation row in the **Relation Sheet** view, or on a relation line in the **Wire Sheet** view, and click **Tags**

Step 2 In the **Relation Tags** dialog, select the desired dictionary from the tag dictionary drop-down list located in the upper-left corner.

NOTE: Optionally, you can enter the namespace followed by “ : ” (colon symbol) in the Search field, located to the right of the drop-down list, to select a tag dictionary. For example, entering `n:` selects the NiagaraTagDictionary.

Step 3 Select one or more tags or tagGroups from the dropdown list and click **AddTag**.

NOTE: You can enter the first letter of a tag name in the Search field to filter the list of tags displayed to those beginning with that letter, as shown here. When filtering results in a single item, it is automatically selected.



Name	Type
Tags	
vendor	String
version	String

The added tags appear in the lower **tags** pane in the dialog.

Step 4 If any of the added tags are value tags, you can edit the value field in the lower **tags** pane.

Step 5 Click **Save**.

Your changes are saved to the relation.

Adding an Ad Hoc tag

You can add Ad Hoc tags (ones that you define as needed) to component relations.

Step 1 Right-click on a relation row in the **Relation Sheet** view, or on a relation line in the **Wire Sheet** view, and click **Tags**.

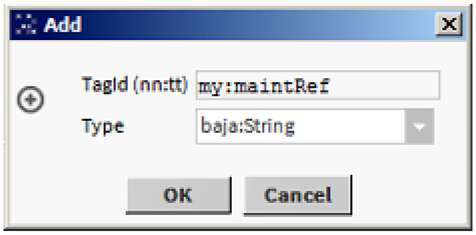
Step 2 In the **Relation Tags** dialog, click the **AddTag** button.

An **Add** dialog appears.

Step 3 Enter a **`TagId`** in the form of `nameSpace:tagName`.

NOTE: Best practice is to use a namespace that is different from any of the installed tag dictionaries. For example, `my:maintRef`.

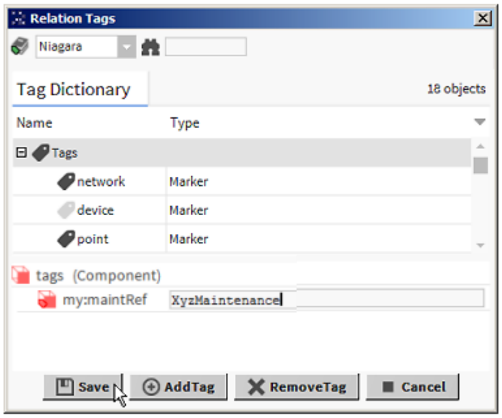
Step 4 Click the **`Type`** dropdown list and select a data type for the tag, as shown here, and click **OK**.



TagId (nn:tt) my:maintRef

Type baja:String

The added tag appears in the lower **tags** pane and the tag icons are red indicating the need to save.



Name	Type
Tags	
network	Marker
device	Marker
point	Marker

tags (Component)

my:maintRef XyzMaintenance

Step 5 Click **Save**.

The tag is added to the relation and the dialog box closes.

Editing a relation tag

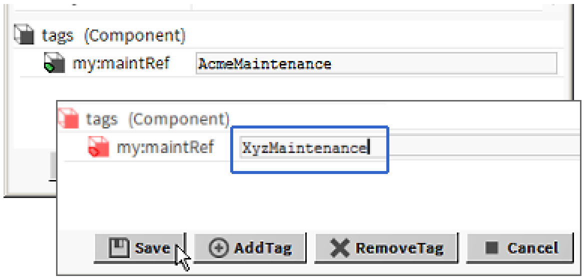
You can edit a direct tag on a relation if it is a value tag.

Prerequisites:

- Existing direct tag on a relation

Step 1 In the **Relation Sheet** view, right click the row to edit.

Step 2 In the lower **tags** pane of the **Relation Tags** dialog, click in the text field of a value tag and edit the value, as shown here.


tags (Component)

my:maintRef AcmeMaintenance

tags (Component)

my:maintRef XyzMaintenance

The relation tag icons turn red indicating the need to save your changes.

Step 3 Click **Save**.

Removing a tag

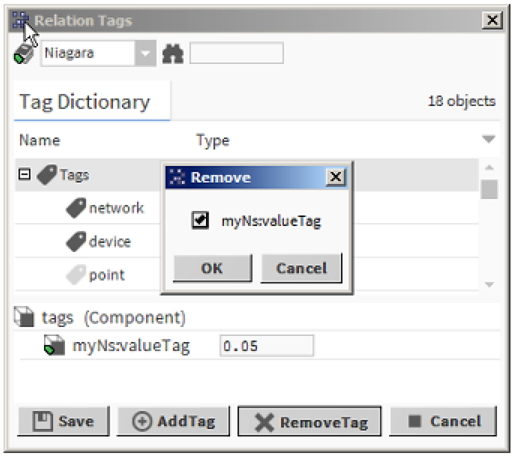
You can remove any direct tags on a component relation. You cannot remove an implied tag.

Step 1 In the **Relation Tags** dialog. click **RemoveTag**.

The Remove Tag dialog appears.

NOTE: The **RemoveTag** button is active only when there is one or more direct tags applied to a relation.

Step 2 Click the checkbox beside any tags you wish to remove and click **OK**.


Relation Tags

Niagara

Tag Dictionary 18 objects

Name	Type
Tags	
network	
device	
point	

Remove

☑ myNs:valueTag

tags (Component)

myNs:valueTag 0.05

NOTE: When there is only one tag, it is automatically selected. If there are multiple tags, you can click **RemoveAll** to remove all of the listed tags.

Step 3 In the **Relation Tags** dialog, click **Save**.

The selected tags are removed the relation changes are saved.

Chapter 2 Relations reference

Topics covered in this chapter

♦ Entity-Relationship Modeling
♦ Types of Relations
♦ RelationID structure
♦ Relation Definitions
♦ Workbench 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 1st entity (the subject), the relation is an "outbound" relation. While from the context of the 2nd 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

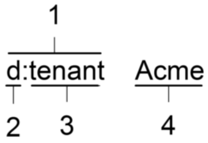
Relations can be either "direct" or "implied" and are defined as follows:

- **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 TagDictionaries 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 TagRules folder (BTagRuleList). When an application queries for the relations on a component in the station, the SmartTagDictionary executes code that interprets the TagRules against the given component and returns a list of implied relations.

RelationID structure

A relationID 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 relationID.

Figure 2 Parts of a RelationID



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

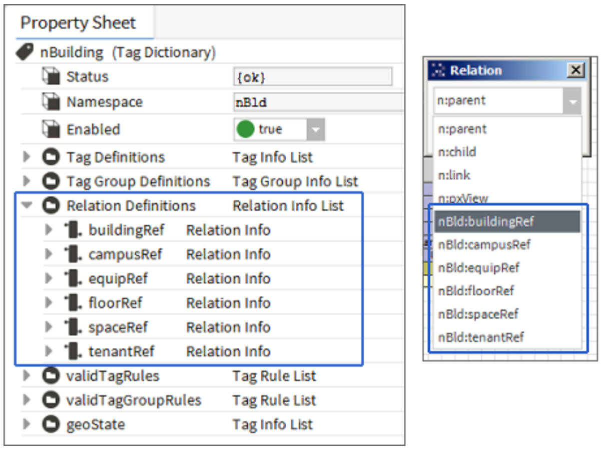
The following table provides definitions of the different parts of a relationID:

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

TagDictionaries 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 TagDictionaries on your system.

Figure 3 Relation Definitions in custom TagDictionary (left) provide choices seen in the Relation dialog (right)



Property Sheet

nBuilding (Tag Dictionary)

Status	{ok}
Namespace	nBld
Enabled	● true
Tag Definitions	Tag Info List
Tag Group Definitions	Tag Group Info List
Relation Definitions	Relation Info List
buildingRef	Relation Info
campusRef	Relation Info
equipRef	Relation Info
floorRef	Relation Info
spaceRef	Relation Info
tenantRef	Relation Info
validTagRules	Tag Rule List
validTagGroupRules	Tag Rule List
geoState	Tag Info List

Relation

n:parent

n:parent
n:child
n:link
n:pxView
nBld:buildingRef
nBld:campusRef
nBld:equipRef
nBld:floorRef
nBld:spaceRef
nBld:tenantRef

Workbench 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.

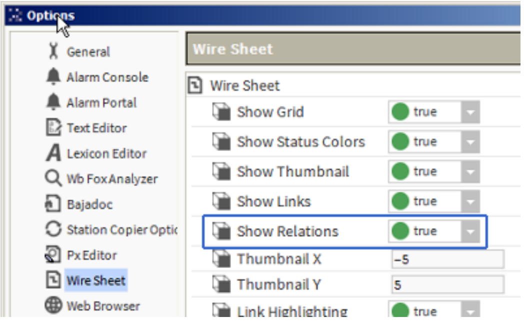
The workflow for creating and editing Relations in Workbench is very similar to that of creating and editing links in previous versions of Workbench. A `BRelation` slot is added to the component that is the inbound side of a relation. The outbound Relation component will have a **RelationKnob**, which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

Wire Sheet view

Workbenchhas 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 Workbench Tools option

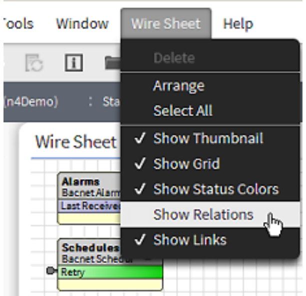


Wire Sheet

Show Grid	true
Show Status Colors	true
Show Thumbnail	true
Show Links	true
Show Relations	true
Thumbnail X	-5
Thumbnail Y	5
Link Highlighting	true

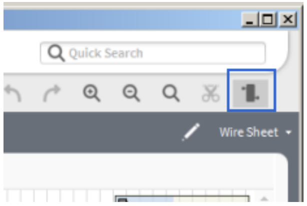
- In a Wire Sheet view, select **Wire Sheet→Show Relations**

Figure 5 Wire Sheet menu option



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

Figure 6 Workbench Tool Bar icon

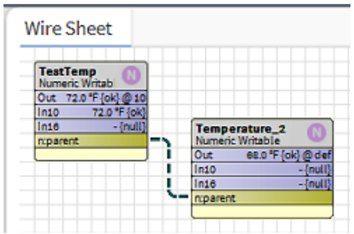


When invoked, the **Wire Sheet Relation** mode displays relations of the selected component as either of the following:

- **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 `RelationId` 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).

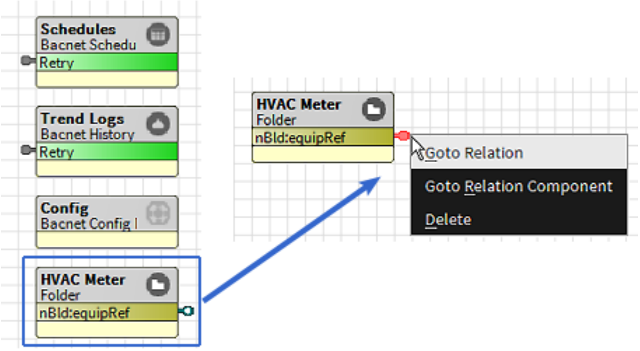
Figure 7 On-sheet relation depicted with dashed line



- **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


Schedules
Bacnet Schedu
Retry
Trend Logs
Bacnet History
Retry
Config
Bacnet Config I
HVAC Meter
Folder
nBld:equipRef
HVAC Meter
Folder
nBld:equipRef
Goto Relation
Goto Relation Component
Delete