

Saia PCD[®] Supervisor V4.11

Easy Binding



Disclaimer

The material in this document is for information purposes only. The content and the product described are subject to change without notice. Honeywell makes no representations or warranties with respect to this document. In no event shall Honeywell be liable for technical or editorial omissions or mistakes in this document, nor shall it be liable for any damages, direct or incidental, arising out of or related to the use of this document. No part of this document may be reproduced in any form or by any means without prior written permission from Honeywell.

Copyright © 2022 HONEYWELL International, Inc. All rights reserved.

Niagara Framework ® is a registered trademark of Tridium Inc.

Trademarks

Saia PCD® is a registered trademark of Saia-Burgess Controls AG.

Technical changes are subject to the state of technology.

Saia-Burgess Controls AG, 2021. © All rights reserved.

Document History

Date	Description
August, 2021	First draft
November, 2021	<ul style="list-style-type: none">• Updated the supervisor version 4.11 on the front cover page.• In update a palette section:<ul style="list-style-type: none">• Removed the steps for “check box availability for restarting the station,” see page 20.• Added steps to encrypt the easy palette image library, see page 20.

TABLE OF CONTENTS

EASY PALETTE.....	1
About the Easy Palette.....	1
Installation Requirements And Licensing	2
System Requirements.....	2
License Easy Binding on a Supervisor Station	2
Creating an Easy Palette.....	4
Prepare the Image Files.....	4
Setup the Folder Structure.....	4
Folder Structure Overview.....	4
Create the Folder Structure	10
Create a ZIP file for a Palette	12
Create an Easy Palette	13
Create a new palette.....	13
Update a palette.....	17
Sign a Third Party Module.....	21
Create a Self-signed Certificate	21
Sign the Module.....	26
Import the Self-signed Certificate into the Platform	29
Common operations of an Easy palette	31
Open a Palette.....	31
Add Widgets to a Px Page.....	31
Bind Widgets to the Datapoint	33
To bind a widget with a datapoint	33
Reset Binding.....	36
Turn Value Display ON/OFF	37
Change Value Alignment.....	38

Delete Binding	39
Using the Widgets	40

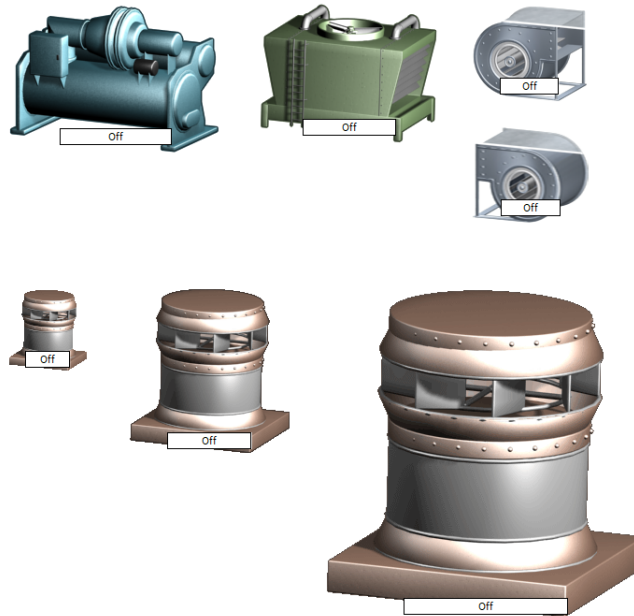
EASY PALETTE

About the Easy Palette

Easy Binding enables you to create a palette of images (widget) that can be dropped onto the Easy Binding tool, which is designed to automatically bind to the images with alarm, value, and override slots of a datapoint. This helps you to create complex schematics where the displayed images changes depending on the value or when the datapoint is in an alarm condition or overridden state.

You can create a widget library and include a range of widgets ready for use with the Easy Binding Tool. These widgets can be added from the Easy Palette or from the honeywellAXPlatinum or honeywellAXPlatinumHR graphic jar files and configured in conjunction with the Easy Binding Tool, which allows you to search and bind the relevant datapoints to the value, alarm, and override slots of the Easy Widget.

Note: *Easy Binding only supported by supervisors.*



Installation Requirements And Licensing

This section provides information about the system requirements and installation process of the Easy Binding on a supervisor station. It is recommended that you have some training or previous experience working with Niagara 4 platform before using Easy Binding.

Important: *The current Easy Binding feature is fully supported only on the supervisor environment running on recommended PC and Operating Systems declared in the product datasheet. This feature is not supported in an embedded controller environment.*

System Requirements

- Niagara 4.9 or higher
- Web browser (Chrome, Firefox, Microsoft Edge)
- Windows 10 (64 bit) or higher version

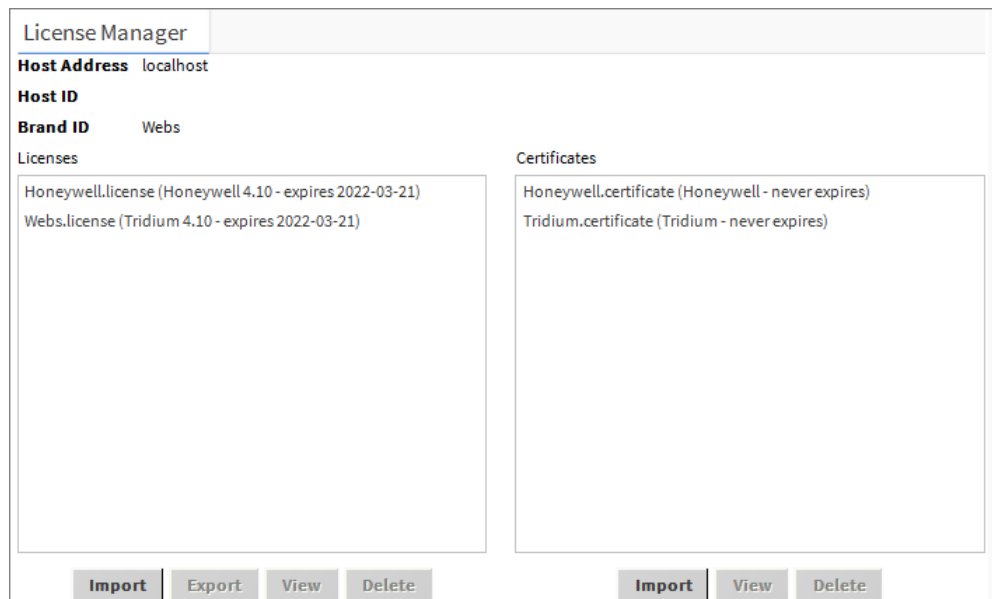
License Easy Binding on a Supervisor Station

Before you install the Easy Binding feature on a supervisor station you must add an Easy Binding license to the Niagara N4 license.

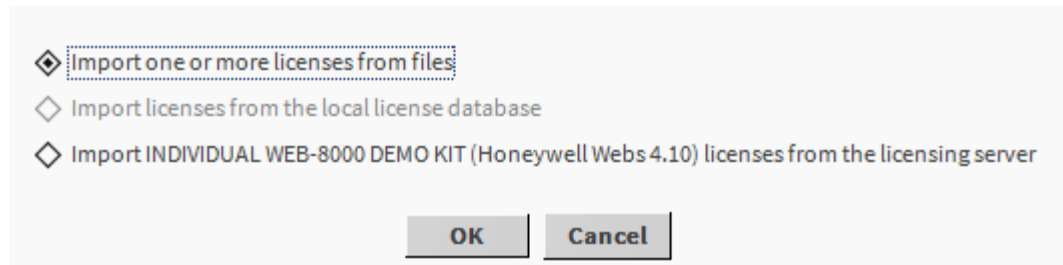
Contact your local distributor or customer support representative to obtain a license for Easy Binding.

To install a license file on a supervisor station

- Step 1. Run the supervisor **Platform**, in the **Nav** tree open the **Platform folder**, and select **License Manager**. The **License Manager** window is displayed.



Step 2. Select **Import** from the bottom of the **License** section. The **Import License** window is displayed.

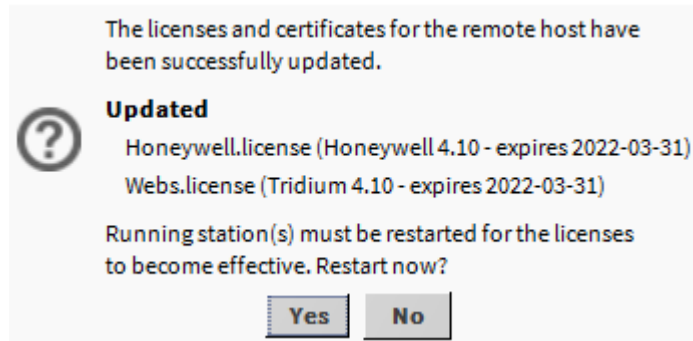


You can import the license files in three ways.

- **Import one or more licenses from files:** Import the license from the local file.
- **Import license from the local license database:** Import license from the local database. This option is only enabled when there is a local license database.
- **Import (Niagara 4.10) licenses from the licensing server:** Import the license from the licensing server if there are no licenses present in the local files and databases. When you select this option, the application displays the following window to restart the station.

Note: *Importing the supervisor license from the licensing server depends on the supervisor station type. For more details about Licensing workbench, refer to the Niagara documentation.*

Step 3. Select the required option and click OK to complete licensing.



Step 4. Click Yes to restart the station. Easy Binding feature available on supervisor workbench.

Creating an Easy Palette

An Easy Palette consists of a collection of widgets. The Easy Palette Wizard guides you through the creation of a new palette or updating an existing one.

To create a palette:

- [Prepare the Image Files](#)
- [Setup the Folder Structure](#)
- [Create a ZIP file for a Palette](#) (optional)
- [Create an Easy Palette](#)
- [Sign a Third Party Module](#)

Prepare the Image Files

Each widget is created from two or more image files. The minimum requirement is for two images, one used when the value is in the 'Off' state and one used when it is in the 'On' state. All images for a widget must be the same size and of the same file format.

BMP, JPG, PNG, or SVG files are supported. SVG files allow the image to appear to move when in a particular state, e.g. a spinning fan image, that appears to spin when on and is stationary when off.

You can use any image file editor to create the required image files, the image file names are not important.

Setup the Folder Structure

It is necessary to setup a folder structure to contain the images that are to be used by the palette.

[Folder Structure Overview](#)

[Create the Folder Structure](#)

Folder Structure Overview

You need to create the folder structure to contain the images that are to be used by the palette.

Top Level Folder

A folder must be created to contain all the sub-folders and image files for the palette. The name of this top level folder does not matter, but it is good practice to use a name that clearly identifies the palette e.g 'Easy Binding Image Library'. This folder can be located anywhere on the computer.

Sub Folders

The top level folder can contain sub folders that group widgets together, for example a sub folder could be created to group all the widgets for cooling coils. The name of these sub folders must be in the form below:

Folder_<folder name>

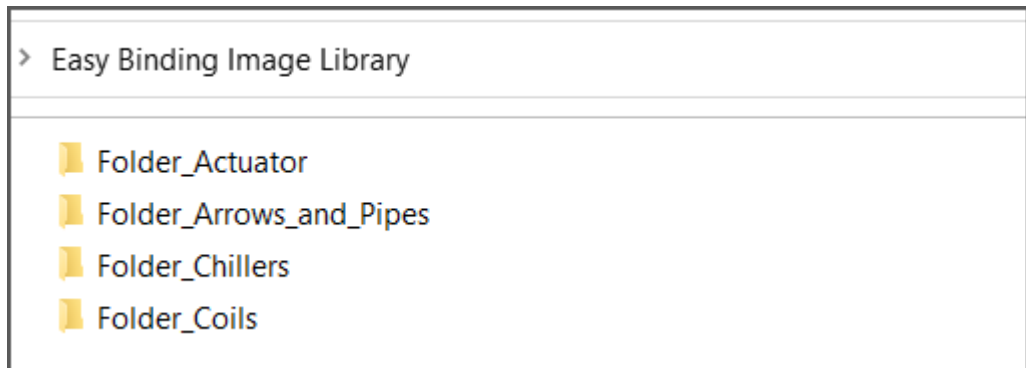
<folder name> is the name of the folder as it will appear in the palette.

Note: No special character except '_' are allowed in the folder name.

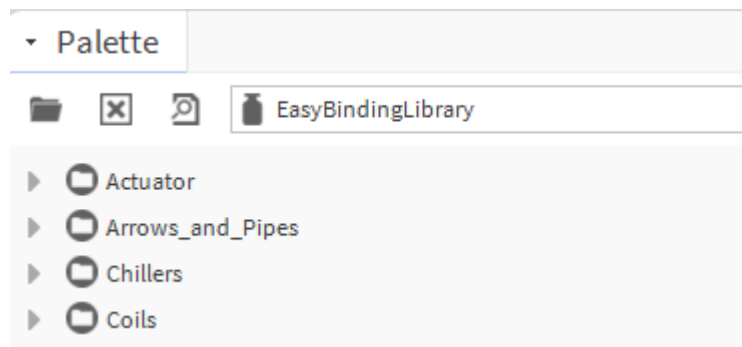
For example to create a 'coils' folder in the palette you must create a sub folder called 'Folder_Coils'.

The folder structure below would create a palette with four folders 'Actuator', 'Arrows_and_Pipes', 'Chillers', and 'Coils' as shown.

Folder Structure

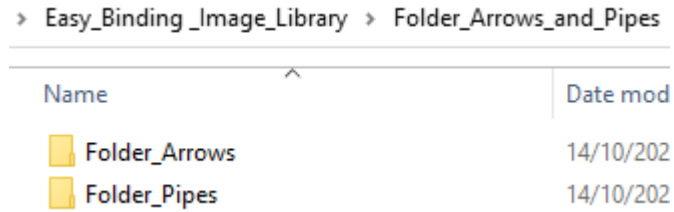


Palette



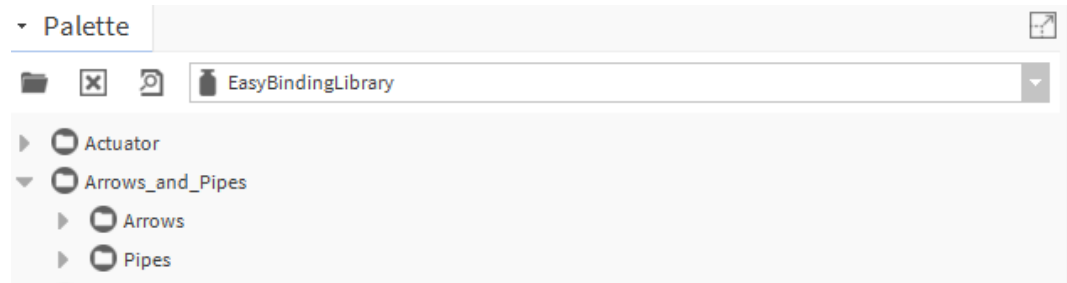
Each sub folder can contain other sub folders e.g. two sub folders 'Arrows' and 'Pipes' could be added to the 'Folder_Arrows_and_Pipes' folder to create an 'Arrows_and_Pipes' containing two sub folders 'Arrows' and 'Pipes'.

Folder Structure



Name	Date mod
Folder_Arrows	14/10/202
Folder_Pipes	14/10/202

Palette



Widget Folders

Each sub folder can contain a number of different 'Widget' folders. Each widget folder defines an instance of a widget and contains the images used by that widget.

The name of the each widget folder must be in the form below:

Widget_<widget name>

<widget name> is the name of the widget as it will appear in the palette.

Note: *No special character except '_' are allowed in the folder name.*

For example to create a widget called 'Coil_Cooling' you must create a widget folder called 'Widget_Coil_Cooling'.

The folder structure below would create four widgets 'Coil_Cooling', 'Coil_Cooling_2Port', 'Coil_Cooling_3Port', and 'Coil_DX' as shown.

Folder Structure

> Easy Binding Image Library > Folder_Coils

- Widget_Coil_Cooling
- Widget_Coil_Cooling_2Port
- Widget_Coil_Cooling_3Port
- Widget_Coil_DX

Palette

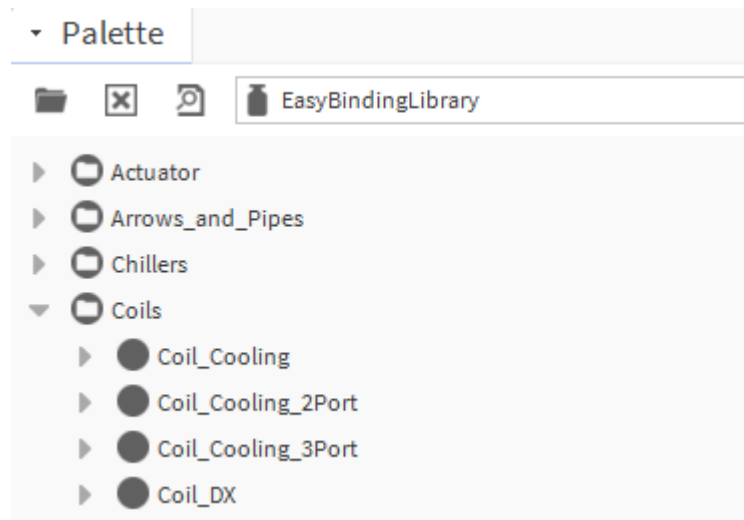


Image Folders

Each widget folder must contain image folders that store the images used in the palette. There must be at least two image folders 'On' and 'Off':

> Folder_Coils > _Widget_Coil_Cooling

- Off
- On

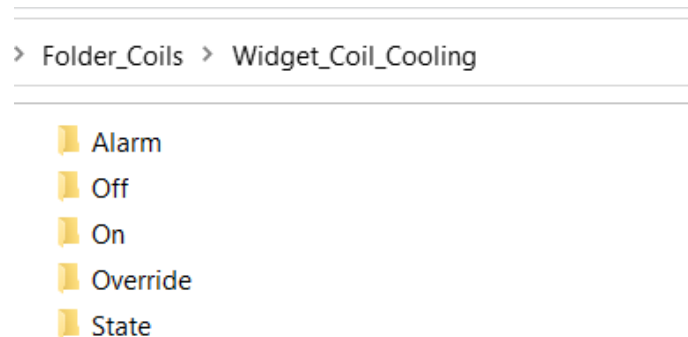
On

Contains the image used for the widget's 'On' state.

Off

Contains the image used for the widget's 'Off' state.

Optional folders can be included in the widget folder to allow different images to be used in different circumstances e.g. when the data point is in alarm:



Alarm

Contains the image for the widget's 'Alarm' overlay.

Override

Contains the image for the widget's 'Override' overlay.

Note: *If the On, Off, Alarm or Override folder contains more than one image file, only the first image will be used, all others will be ignored.*

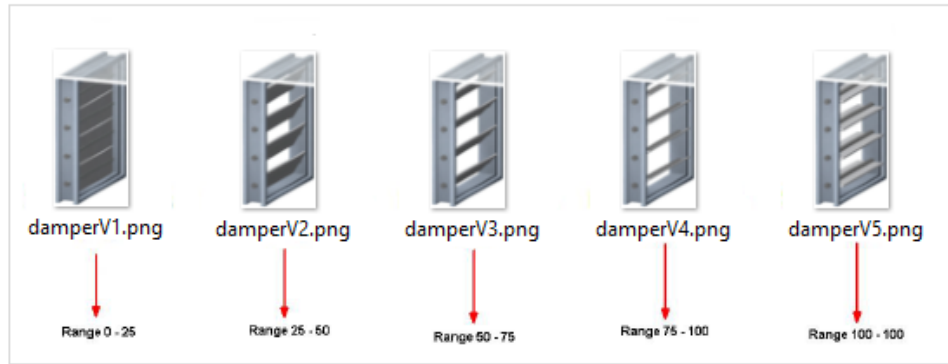
Note: *If an alarm or override folders/images are not included, then the widget will use the default overlays.*

State

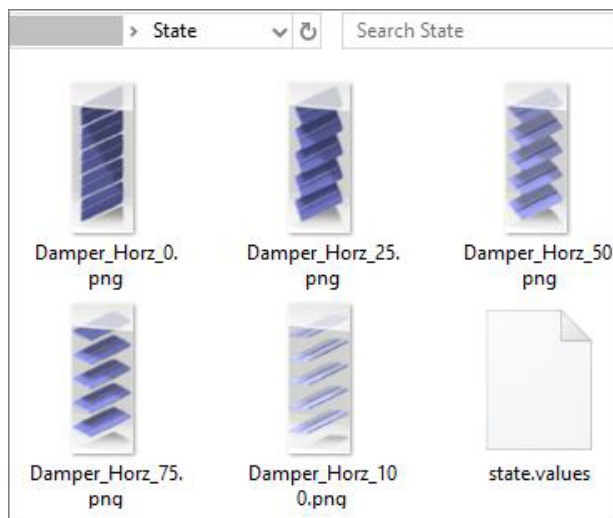
Contains a set of images that can be used to indicate multiple states or levels for **Numeric** or **Enumerated** datapoint values, each graphic defines the value range.

The graphic displayed for state or range is defined in the *state.values* file included in the State folder. This file maps a value or range to the required graphic. The file is a text file that can be created in a standard text editor (e.g. Notepad++). For example, to represent

the different positions of a damper as below:



The following folder would be as below:



The following folder would be as below:

```
0=Damper_Horz_0.png  
25=Damper_Horz_25.png  
50=Damper_Horz_50.png  
75=Damper_Horz_75.png  
100=Damper_Horz_100.png
```

Note: The State folder does not support Boolean datapoint type

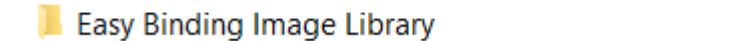
Create the Folder Structure

To create the Folder structure:

- Step 1. In File Explorer create the folder (Top Level Folder) that is to contain all the sub-folders and image files for the palette. This folder can be located anywhere on the computer, see [Top Level Folder](#).




> Current Work

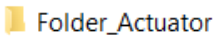


Easy Binding Image Library

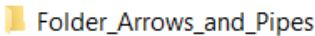
- Step 2. Create the required sub folders inside the Top Level Folder. The folder names must be prefixed with 'Folder_'. For example to create a folder called 'Coils' the folder should be named 'Folder_Coils', see [Sub Folders](#).



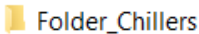
> Easy Binding Image Library



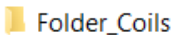
Folder_Actuator



Folder_Arrows_and_Pipes

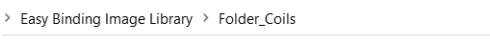


Folder_Chillers



Folder_Coils

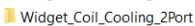
- Step 3. Create the required widget folders inside the sub folders. The folder names must be prefixed with 'Widget_'. For example to create a folder called 'Coil_Cooling' the folder must be named 'Widget_Coil_Cooling', see [Widget Folders](#).



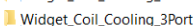
> Easy Binding Image Library > Folder_Coils



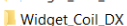
Widget_Coil_Cooling



Widget_Coil_Cooling_2Port

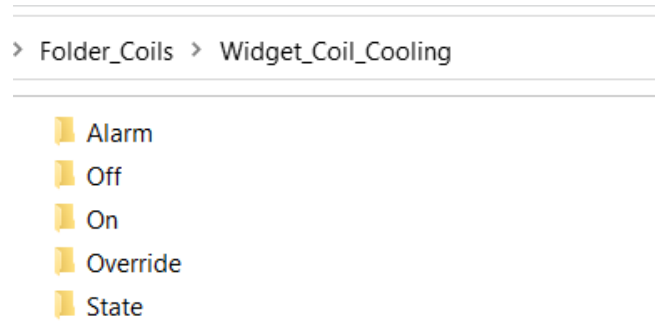


Widget_Coil_Cooling_3Port



Widget_Coil_DX

Step 4. Create the required image folders inside the widget folders. Each image folder must contain at least two sub folders 'On' and 'Off' and can optionally contain 'Alarm', 'Override', and 'State' folders, see [Image Folders](#).



Step 5. Add the image files to the image folders ('On', 'Off', 'Alarm', 'Override', and 'State') - see [Prepare the Image Files](#) for details of creating the image files.

Alarm Folder

> Easy Binding Image Library > Folder_Coils > _Widget_Coil_Cooling > Alarm

Name	Date modified	Type	Size
Fan_Sensor_Fail_Vert_Left.png	15/10/2021 10:05	PNG File	

Off Folder

> Easy Binding Image Library > Folder_Coils > _Widget_Coil_Cooling > Off

Name	Date modified	Type	Size
Coil_Cooling_100.png	15/10/2021 10:05	PNG File	

On Folder

> Easy Binding Image Library > Folder_Coils > _Widget_Coil_Cooling > On

Name	Date modified	Type	Size
Coil_Cooling_0.png	15/10/2021 10:05	PNG File	







Override Folder

> Easy Binding Image Library > Folder_Coils > _Widget_Coil_Cooling > Override

Name	Date modified	Type	Size
Coil_DX_100.png	15/10/2021 10:05	PNG File	

State Folder

Current Work > Easy Binding Image Library > Folder_Coils > _Widget_Coil_Cooling > State

Name	Date modified	Type
 Damper_Horz_0.png	15/10/2021 10:05	PNG File
 Damper_Horz_25.png	15/10/2021 10:05	PNG File
 Damper_Horz_50.png	15/10/2021 10:05	PNG File
 Damper_Horz_75.png	15/10/2021 10:05	PNG File
 Damper_Horz_100.png	15/10/2021 10:05	PNG File
 state.values	15/10/2021 10:15	VALUES File

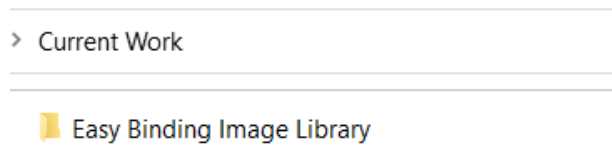
Note: All images for each widget must be the same size and of the same file format.

Create a ZIP file for a Palette

During the process of creating a palette you can choose to create the palette from a zip file of the file structure or from the directory structure itself.

To create a zip file:

Step 1. In File Explorer Navigate to the Top level folder.



Right click the top level folder and select **Send to > Compressed (zipped) folder**.

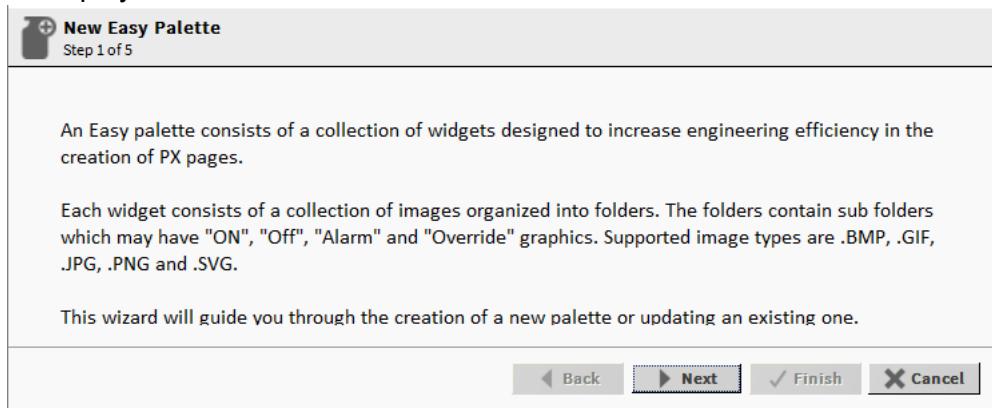
Create an Easy Palette

You can create a new palette or modify an existing one.

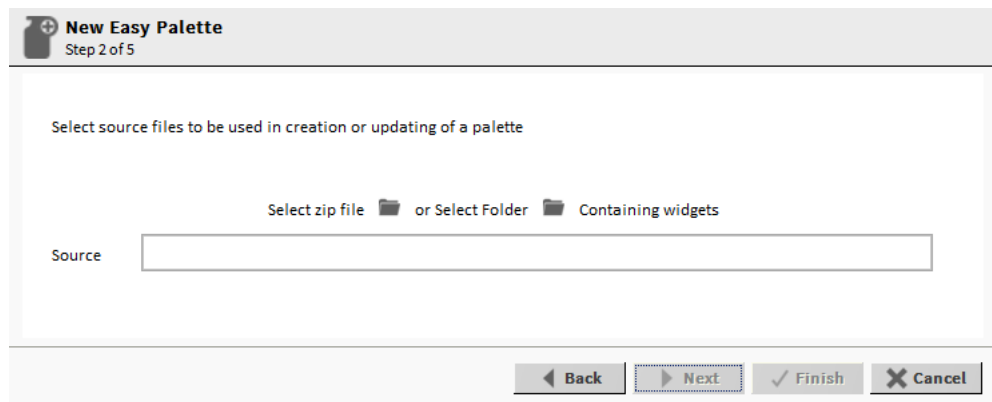
Create a new palette

To create new palette

- Step 1. Create the folder structure- see [Create the Folder Structure](#).
- Step 2. Select **Tools > New Easy Palette**. The **New Easy Palette** wizard is displayed.



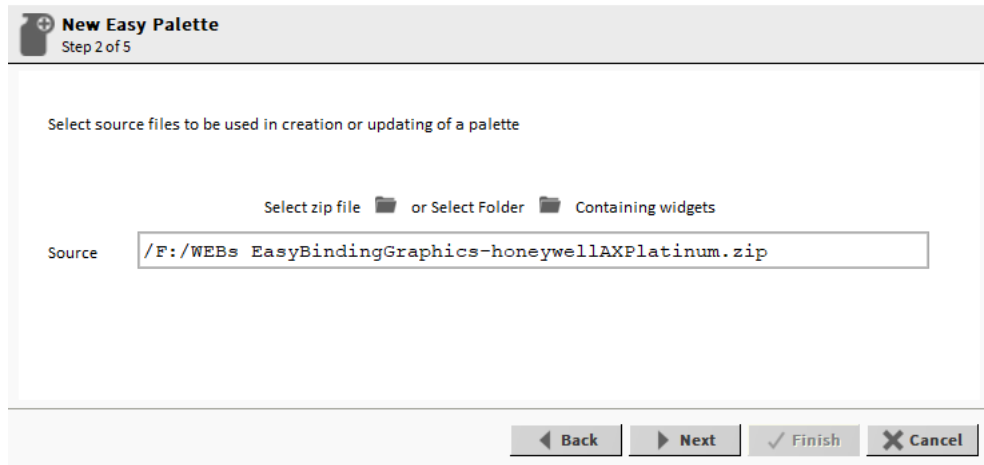
- Step 3. Click **Next**. This displays the next window (Step 2 of 5) that allows you to select the source files to be used to create a palette.



- Step 4. Specify the files to be used to create the palette.

To create the palette from a previously configured file structure that has been saved to a zip file:

- Select  (folder icon) next to **Select zip file**, navigate to the zip file, select it, and select **Open**.

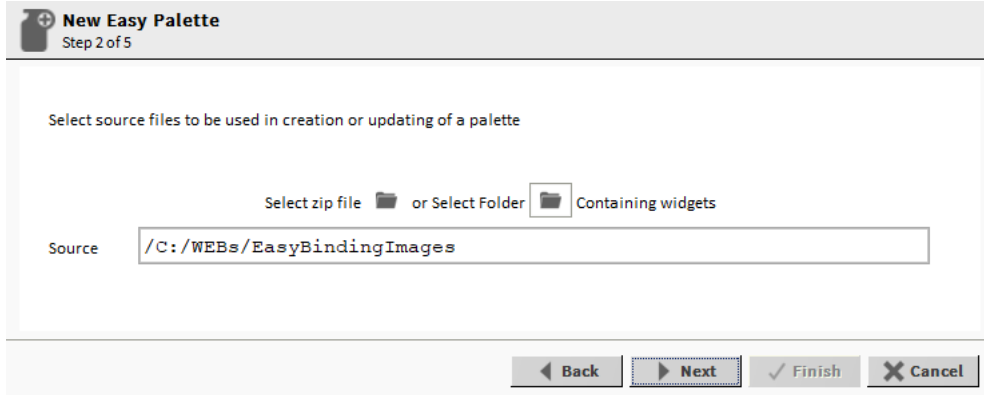


Note: You must have admin rights to access the Zip file.

Or

To create a palette from a directory structure that has been created. (To Create a folder structure refer to [Create the Folder Structure](#)).

- Select  (folder icon) next to **Select Folder**, navigate to the level folder to be used for the palette, select it, and select **Choose**.



Step 5. Click **Next**. This action displays the next window (Step 3 of 5) that allows user to create a new palette or update an existing palette.

Step 6. Select **Create a new palette**.

Step 7. Enter a **Name** and **Description** for the palette in the appropriate field.

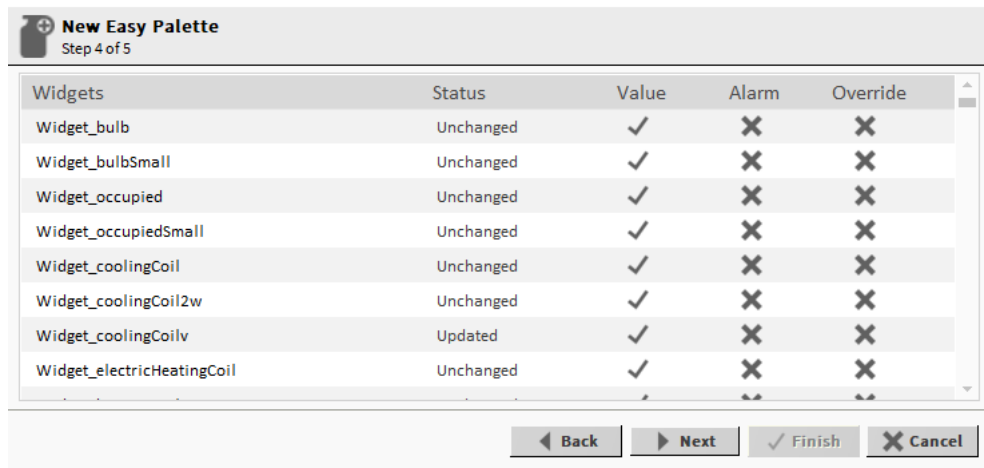
Name - The palette/module name.

Description - A description of the palette or module.

Note: Special characters are not allowed in the Name and Description.

Step 8. Click **Next**. The status for Value, Alarm and Override are displayed for all the widgets.

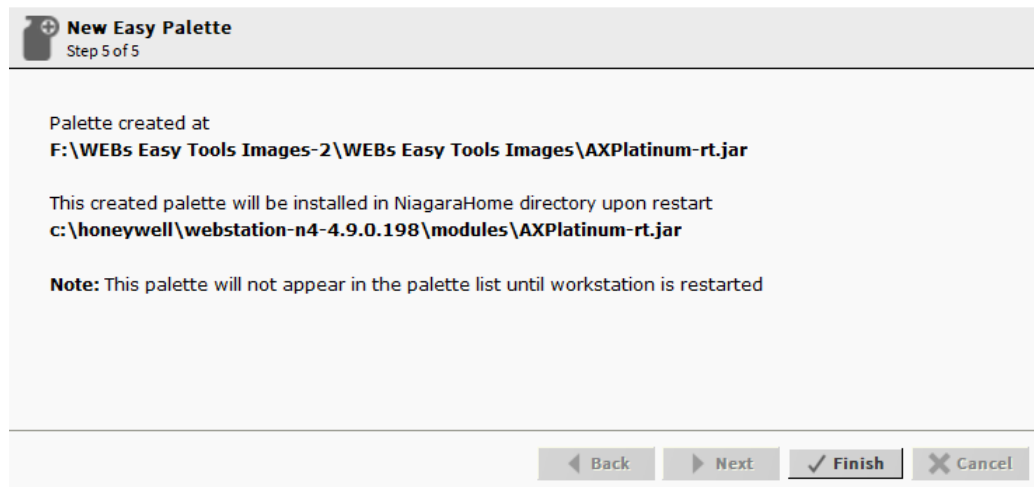
Widgets	Status	Value	Alarm	Override
Widget_Coil_Cooling	New	✓	✓	✓
Widget_Coil_Cooling_2Port	New	✓	✗	✗
Widget_Coil_Cooling_3Port	New	✓	✗	✗
Widget_Coil_DX	New	✓	✗	✗
Widget_Coil_Electric	New	✓	✗	✗
Widget_Coil_Heating	New	✓	✓	✗
Widget_Coil_Heating_2Port	New	✓	✓	✗



Column	Description
Widgets	The widgets in the folder.
Status	The status of the widget folder: New - The widget folder is new. Unchanged - The widget folder existed previously and is unchanged. Updated - The widget folder existed previously and has been modified.
Value (On and off)	✓ , indicates that the images are placed inside the respective sub-folder (Alarm/ Value/Override). ✗ , this indicates that the images or graphics are not placed inside the respective sub-folder (Alarm/Value/Override).
Alarm	
Override	

Note: After clicking Next, you cannot undo the changes.

Step 9. Click **Next**.The final step (Step 5 of 5) is displayed.



Step 10. Click **Finish**.

Note: The station must be restarted for the palette to be available.

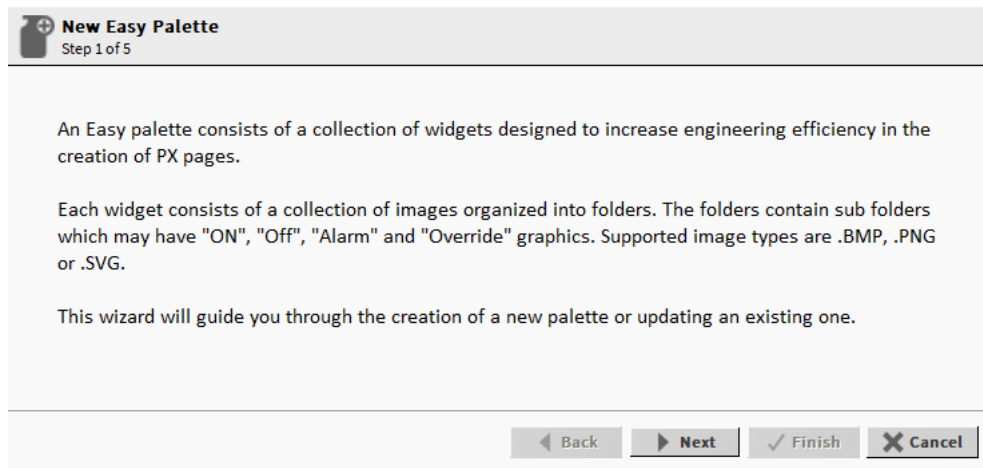
Important: After creating your own Easy Binding Widget, you must sign in with a valid certificate using Niagara's Jar Signer Tool - see Sign a Third Party Module.

A station can be discovered into another station in a via the network, this allows you to work on Easy Binding Widgets across different stations. To work with the Easy Binding Widgets in this way, the code signing certificate and Easy Binding Certificate must be signed with a valid CA certificate which must be available in the Niagara User key store, for more detail of creating a self signed CA certificate - see Sign a Third Party Module.

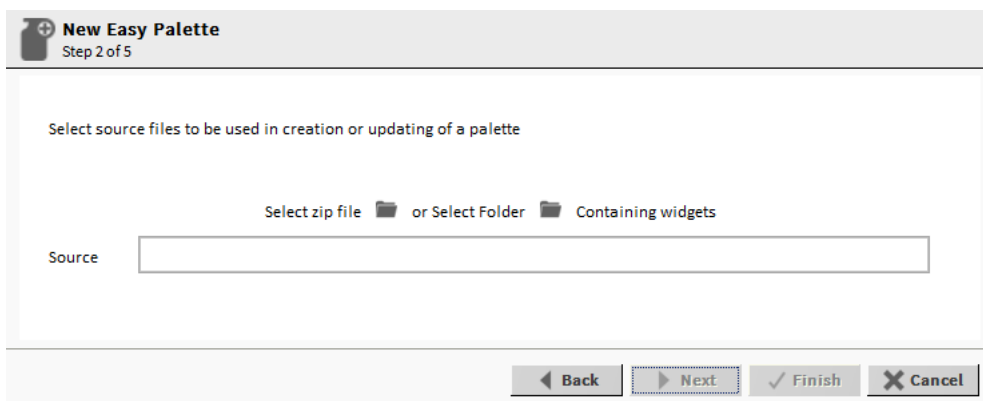
Update a palette

Step 1. Create the folder structure- see [Create the Folder Structure](#).

Step 2. Select **Tools > New Easy Palette**. This opens the **New Easy Palette** wizard.



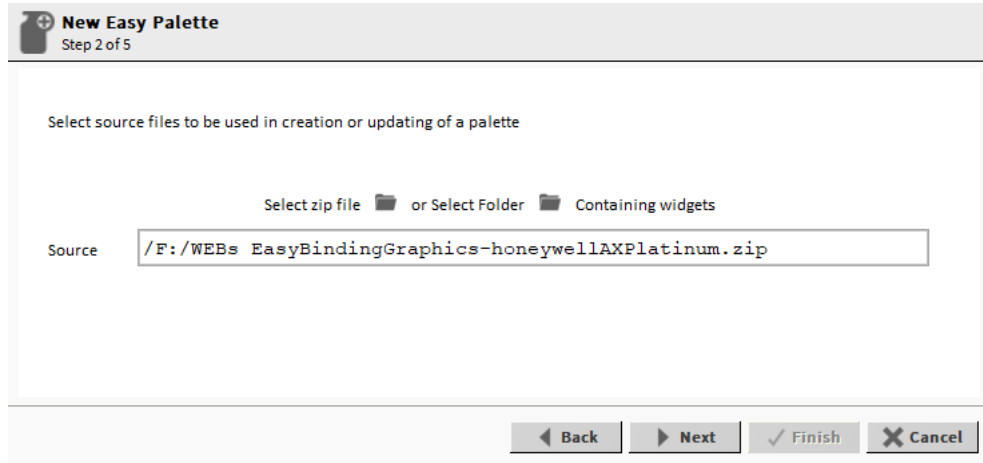
Step 3. Click **Next**. This displays the next window (Step 2 of 5) that allows you to select the source files to be used to create a palette.



Step 4. Specify the files to be used to create the palette.

To create the palette from a previously configured file structure that has been saved to a zip file:

- Select  (folder icon) next to **Select zip file**, navigate to the zip file, select it, and select **Open**.

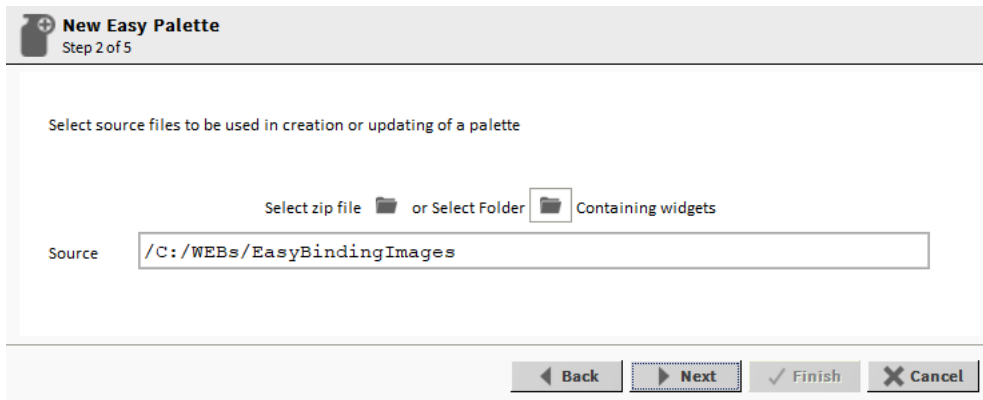


Note: You must have admin rights to access the Zip file.

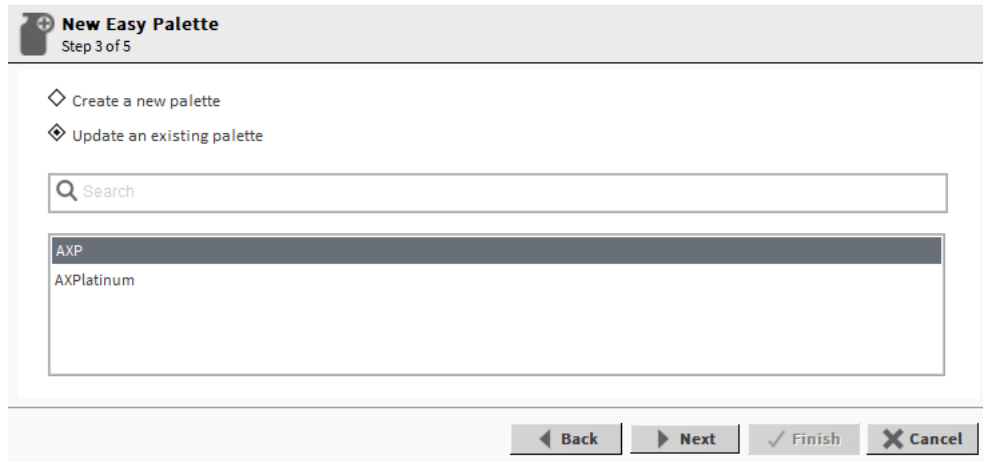
Or

To create a palette from a directory structure that has been created.

- Select  (folder icon) next to **Select Folder**, navigate to the level folder to be used for the palette, select it, and select **Choose**.

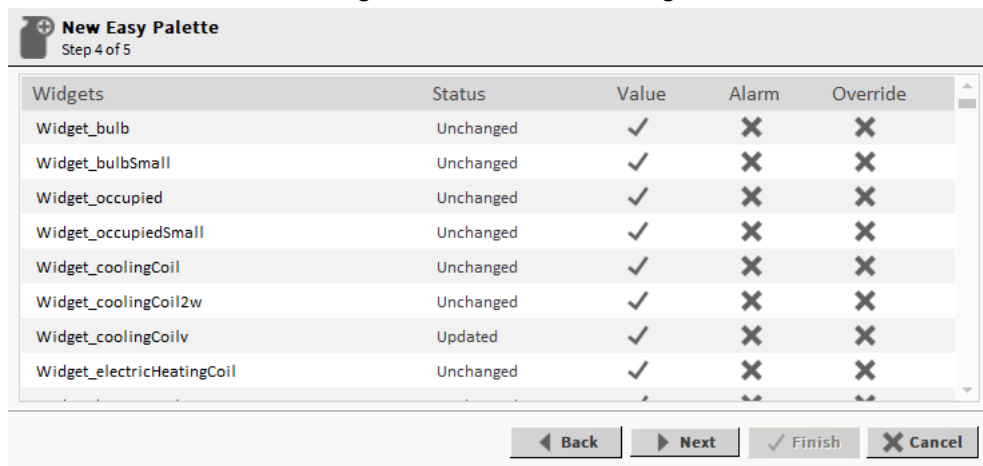


- Step 5. Click **Next**. This displays the next window (Step 3 of 5) that allows you to update an existing palette



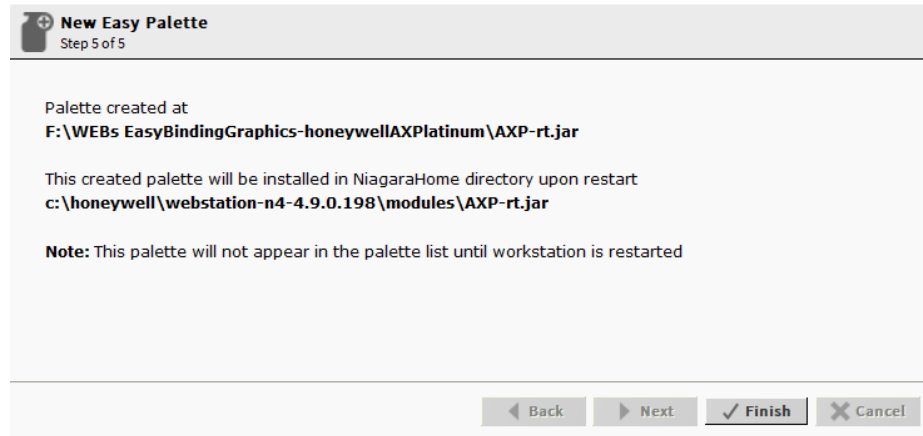
- Step 6. Select **Update an existing palette**.
- Step 7. Enter the name of the palette that is to be changed in the search field. A list of palettes matching the search criteria are displayed.
- Step 8. Select the palette that is to be modified.
- Step 9. Click **Next**. The wizard displays all status for Value, Alarm, and Override displayed for all the widgets.

If you updated any graphics or added new ones "New" is displayed in the Status column for the updated widgets and "Unchanged" is displayed in the Status column for widgets that have not changed.



Note: After clicking Next, you cannot undo the changes.

Step 10. Click **Next**. The wizard changes.

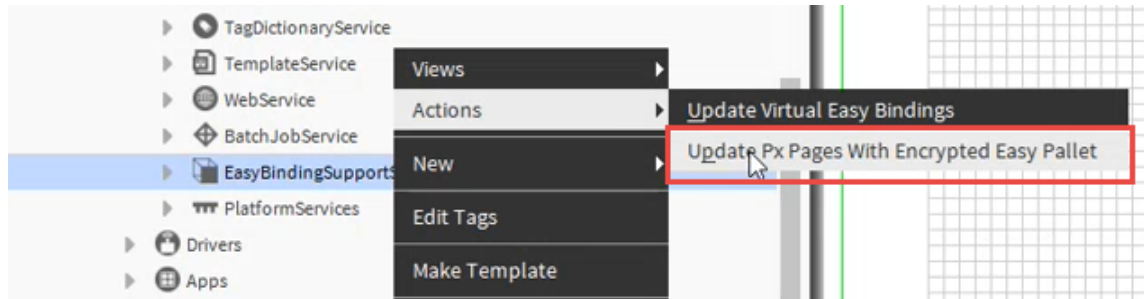


Step 11. Click **Finish**.

Note: After creating the Easy palette using an existing palette, you must encrypt the Easy palette to secure the existing image library which can be viewed only on the Px pages.

Expand the Services folder in the Nav Sidebar, right-click EasyBindingSupportService > Actions > click Update Px Pages With Encrypted Easy Pallet

Step 12. Expand the Services folder in the **Nav** Sidebar, right-click **EasyBindingSupportService > Actions > click Update Px Pages With Encrypted Easy Pallet.**



Note: The station must be restarted for the palette to be available.

Important: After creating your own Easy Binding Widget, you must sign in with a valid certificate using Niagara's Jar Signer Tool - see [Sign a Third Party Module](#)

Important: Niagara station can be discovered into another station in a Niagara network, this allows users to work on Easy Binding modules across the station. To work with the Easy Binding modules across the station, the code signing certificate and Easy Binding Certificate must be signed with a valid CA certificate and should be available in the Niagara User key store.

Niagara allows to create a Self signed CA certificate, for more details refer to Niagara internal help link (Ord)

<http://docplatform/doc/GeneratingACACertificateAndSignedSe-89FC4303.htm>|module:/docPlatform/doc/GeneratingACACertificateAndSignedSe-89FC4303.html

Sign a Third Party Module

Code signing is the process of applying a digital signature to a piece of code so that it can be verified to ensure that it has not been modified after it was signed. Any module you create e.g. an Easy Palette must be signed. To do this you must:

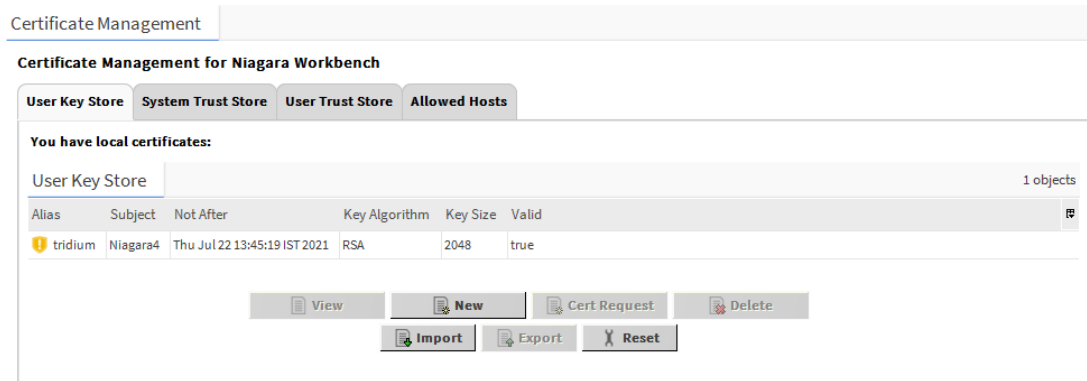
- [Create a Self-signed Certificate](#)
- [Sign the Module](#)
- [Import the Self-signed Certificate into the Platform](#)

Note: If the module is to be used on a different workbench, the certificate must be imported into the target platform - see [Import the Self-signed Certificate into the Platform](#).

Create a Self-signed Certificate

To create a self-signed certificate:

Step 1. Run the workbench and select **Tools > Certificate Management**. The **Certificate Management** screen is displayed.



Step 2. Click **New**. The **Generate Self Signed Certificate** dialogue box is displayed.

The dialog box 'Generate Self Signed Certificate' contains the following fields and options:

- Alias: codesigning1 (required)
- Common Name (CN): codesigning1 (required)
* this may contain the host name or address of the server
- Organizational Unit (OU): ProductDev
- Organization (O): Tridium (required)
- Locality (L): Richmond
- State/Province (ST): VA
- Country Code (C): US (required)
- Not Before: 17-Jul-2020 09:40 AM EDT
- Not After: 17-Jul-2025 09:40 AM EDT
- Key Size: 1024 bits, 2048 bits, 3072 bits, 4096 bits
- Certificate Usage: Server, Client, CA, Code Signing (highlighted with a red box)
- Alternate Server Name: (empty)

In the **Alias** box enter the alias for the certificate.

In the **Common Name (CN)** box enter the common name for the certificate.

In the **Organization (O)** box enter the organisation name.

In the **Country Code (C)** box then US.

Select the **Code Signing** option under **Certificate Usage**.

Specify any other properties as required.

Generate Self Signed Certificate

Generate Self Signed Certificate
Generates a self signed certificate and inserts it into the keystore

Alias (required)

Common Name (CN) (required)
* this may contain the host name or address of the server

Organizational Unit (OU)

Organization (O) (required)

Locality (L)

State/Province (ST)

Country Code (C) (required)

Not Before

Not After

Key Size 1024 bits 2048 bits 3072 bits 4096 bits

Certificate Usage Server Client CA Code Signing

Alternate Server Name

Email Address

Key Usage Digital signature Non-repudiation Key encipherment
 Data encipherment Key agreement Certificate signing
 CRL signing Encipher only Decipher only

Step 3. Click **OK**. the **Private Key** dialogue box is displayed.

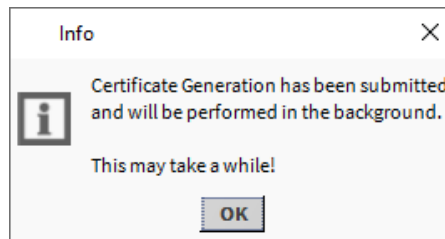


In the **Password** box specify the password.

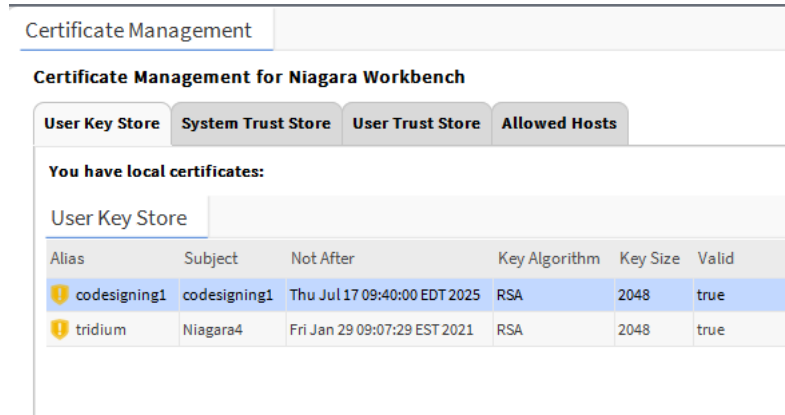
Note: *The password should be at least 10 characters. At least one character should be a digit, one must be lowercase, and one must be an uppercase letter.*

In the **Confirm** box confirm the password.

Step 4. Click **OK**. The Info dialogue box is displayed.

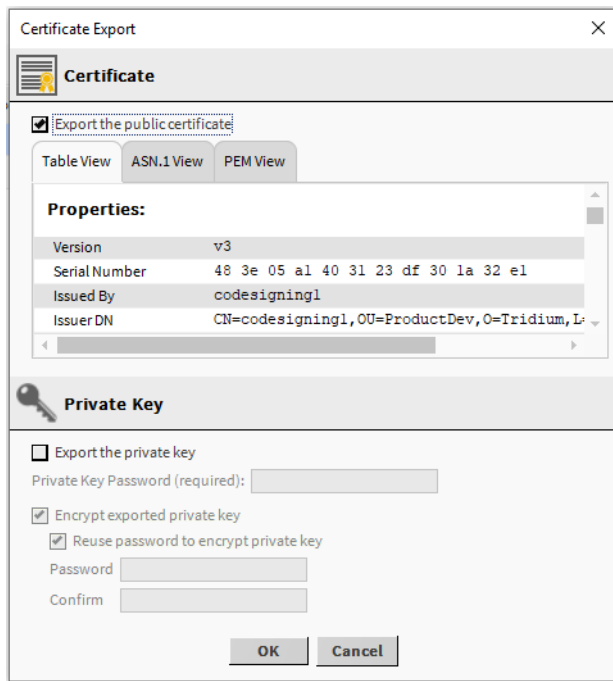


Step 5. Click **OK**. The certificate will now be visible under the **User Key Store** tab.

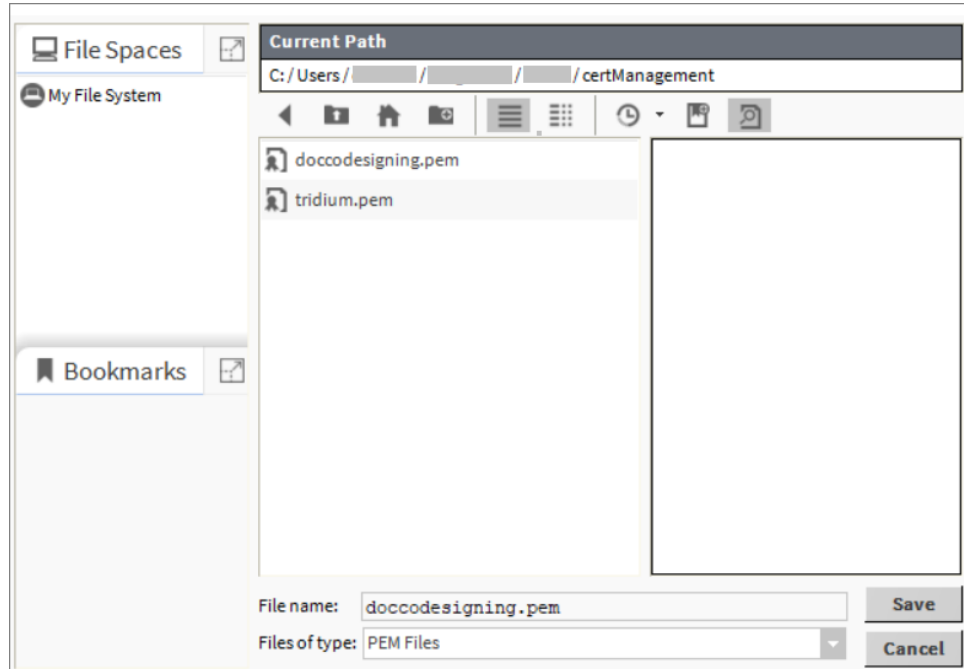


Step 6. Click the certificate that has just been created.

Step 7. Click **Export**. The **Certificate Export** dialogue box is displayed.



Step 8. Click **OK**. The dialogue box changes.

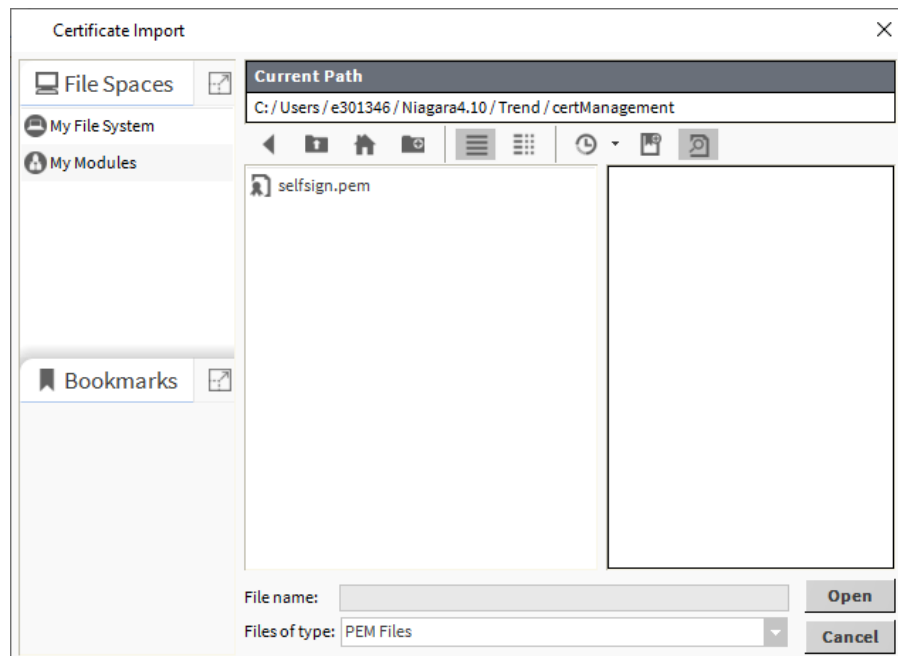


Step 9. Navigate to the location the certificate is to be exported to.

Step 10. Click **Save**. The certificate will be exported.

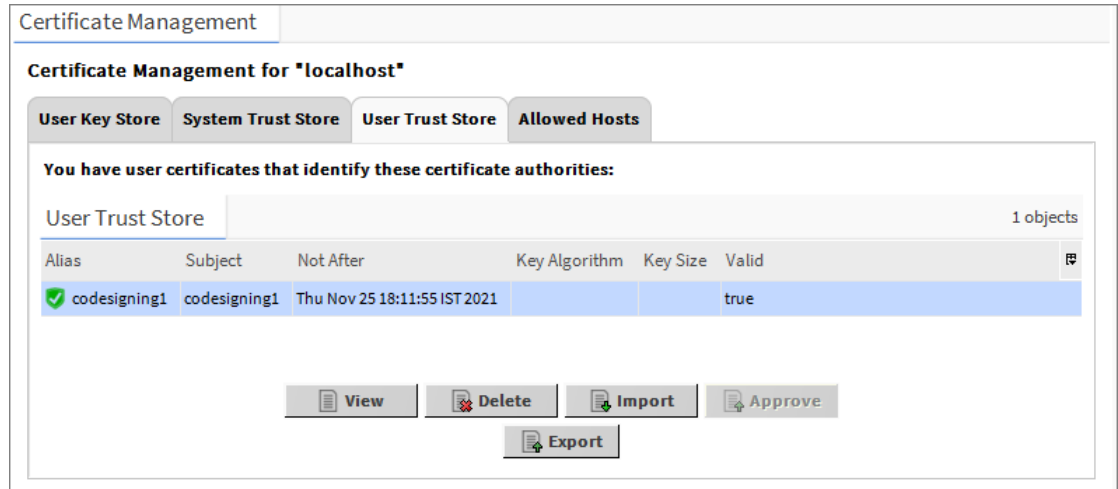
Step 11. Select the **User Trust Store** tab.

Step 12. Click **Import**. The **Certificate Import** dialogue box is displayed.



Step 13. Navigate to the location of the certificate and select it.

Step 14. Click **Open**. The certificate will be imported and displayed with a green tick in the **User Trust Store** tab.



Sign the Module

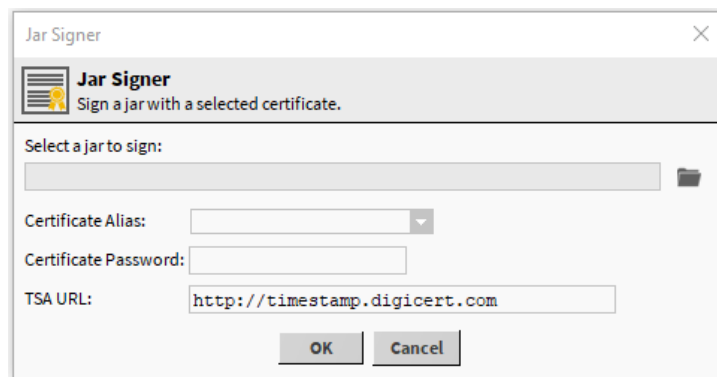
This section explains how to sign the third-party modules using the Jar Signing Tool.



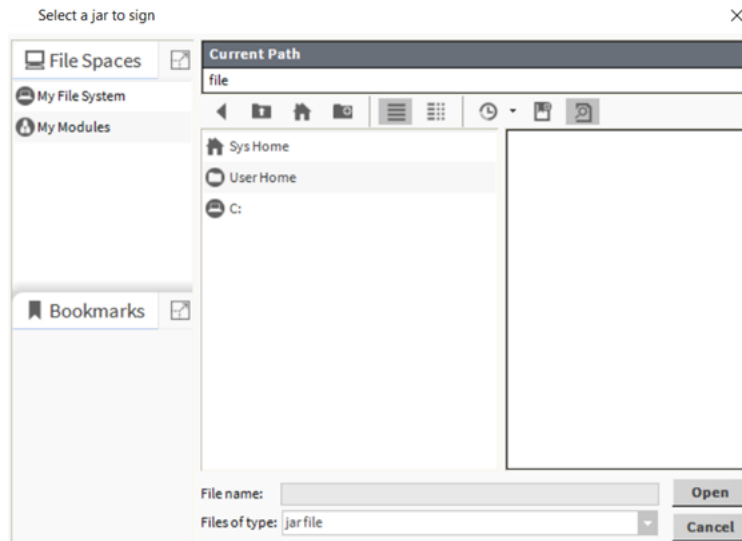
Caution: Security best practice is to encourage the original authors of a module to sign their own code. Signing a module on behalf of a 3rd party should be done only as a last option, and only if you trust the authors.

To sign the module:

Step 1. Run workbench and on the **Tools** menu select **Jar Signer Tool**. The **Jar Signer** dialogue box is displayed.



Step 2. Click  folder icon. The **Select a jar to sign** dialogue box is displayed.

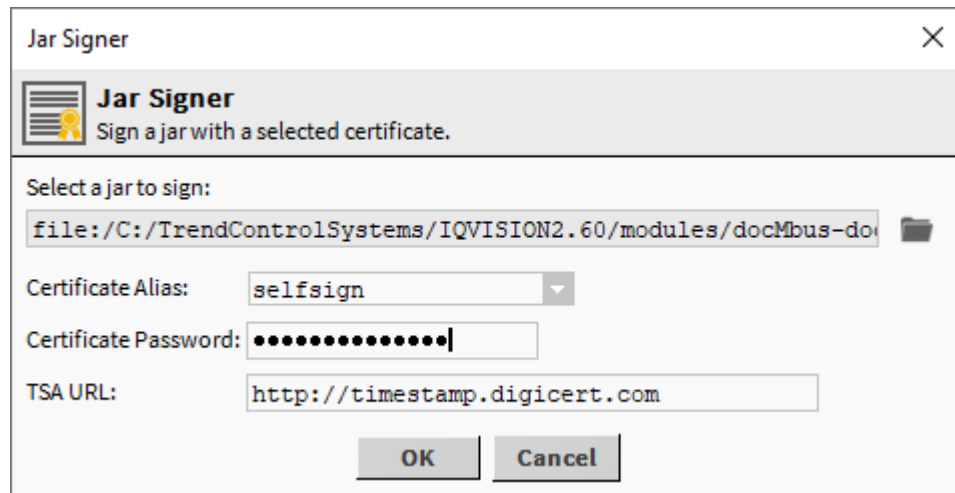


Step 3. Navigate to the location of the module that is to be signed and select it.

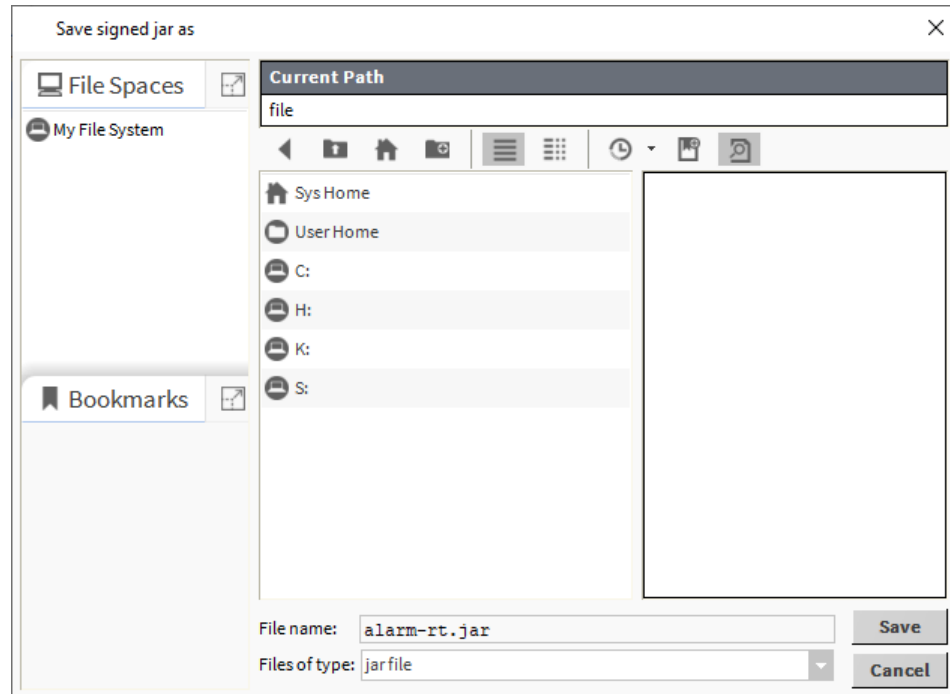
Step 4. Click **Open** to return to the **Jar Signer** dialogue box.

Step 5. In the **Certificate Alias** box select the certificate that is to be used to sign the module.

Step 6. In the **Certificate Password** box enter the password for the certificate.



Step 7. Click **OK**. The **Save signed jar** as dialogue box is displayed.



Note: The signed module should be saved inside the modules folder or save anywhere and then copy into the modules folder.

Step 8. Navigate to the location to which the signed module is to be saved and click **Save**. A confirmation window is displayed.

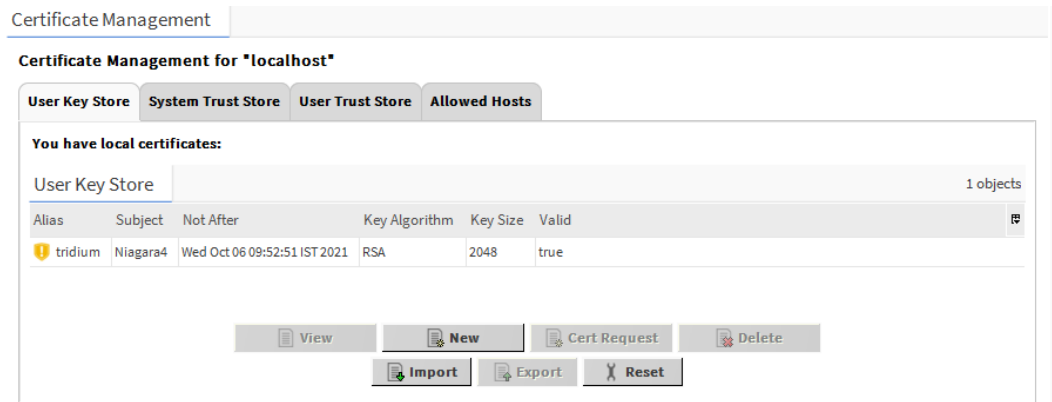
Step 9. Click **OK**.

Import the Self-signed Certificate into the Platform

To import the self-signed certificate into the platform:

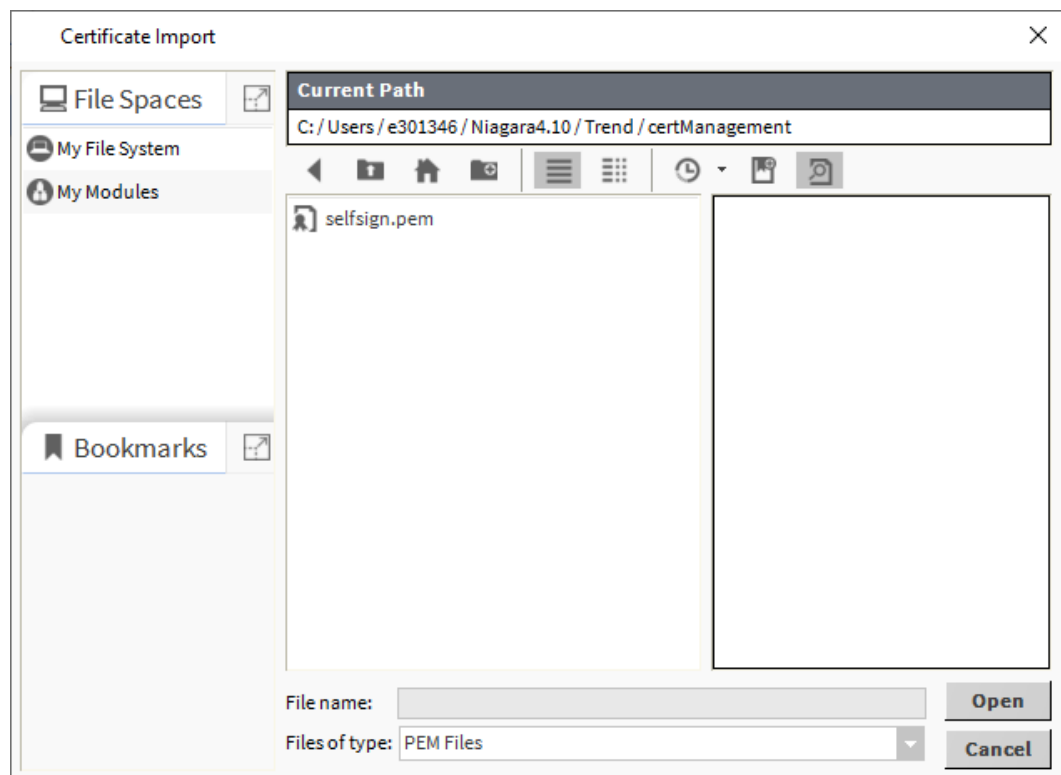
Step 1. Run workbench and connect to the platform.

Step 2. In the **Nav** tree select the **Platform > Certificate Management**. The **Certificate Management** screen is displayed.

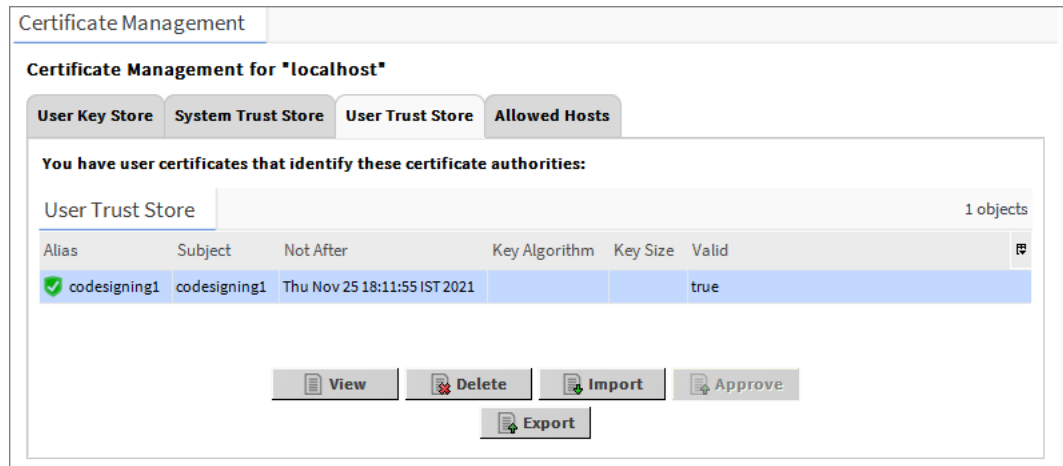


Step 3. Select the **User Trust Store** tab.

Step 4. Click **Import**. The **Certificate Import** dialogue box is displayed.



- Step 5. Navigate to the location of the certificate and select it.
- Step 6. Click **Open**. The certificate will be imported and displayed with a green tick in the **User Trust Store** tab, and you should be able to use the signed Easy Binding widget and 3rd party modules on the platform.



Common operations of an Easy palette


Open a Palette

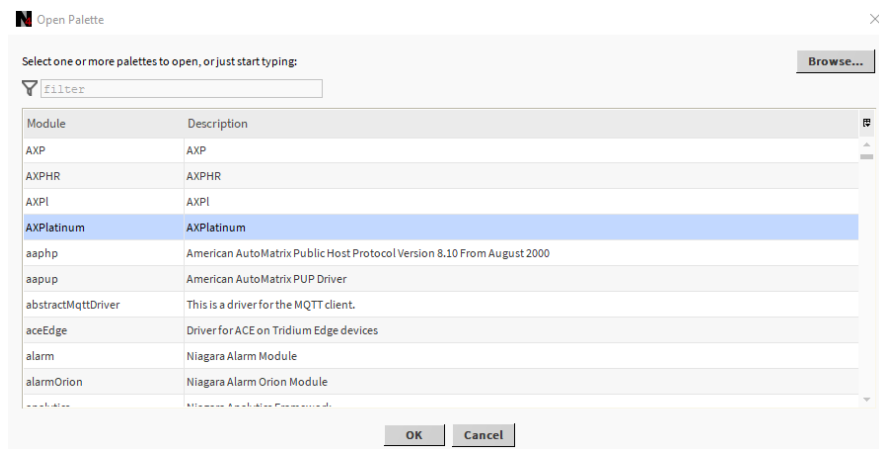
Before you can use the palette to add widgets to a px page the palette must be opened.

Note: User must have an Easy Binding features in license in order to load the palette.

Open a palette

Step 1. On the menu bar select **Window > Side Bars > Palette**.

Step 2. Click  (folder icon). The **Open Palette** dialogue is displayed.



Step 3. Enter the name of the palette in the filter box and select the required palette from the list.

Step 4. Click **OK**. The selected palette is added to the **Palette** section.


Add Widgets to a Px Page

Widgets from an Easy Palette can be used in conjunction with the Easy Binding Tool to add images to a px page that are dependent on the datapoint's value.

To add widgets to a Px page

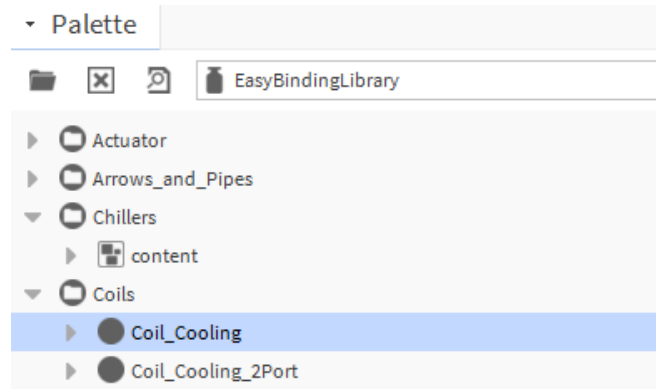
Step 1. Open the Px page the widget is to be added to in edit mode.

Step 2. Open the required Easy Palette — see [Open a Palette](#).

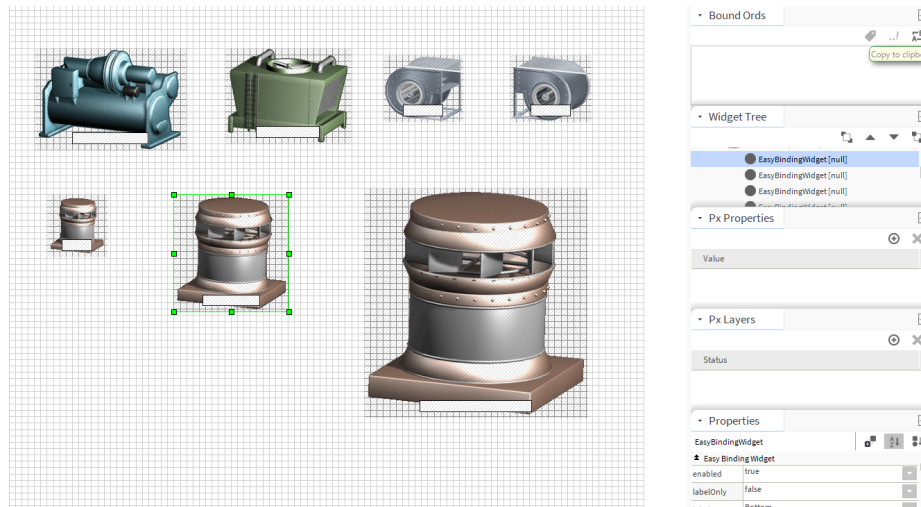
Note: Click  (search icon), in the palette to open a preview pane - this will show a preview of the graphic.

Step 3. Expand the folder structure in the palette to display the widget that is to be

added to the page.



Step 4. Drag the required widget from the palette to the Px page. The widget is added to the page.



Step 5. Position and resize it by clicking any of the pointers and dragging to resize the widget.

Note: Do not change *statusEffect* drop down property in the easy bindings property sheet, if its value is other than “None” then e-signature effect will override easy binding image effects.

Note: When you drag and drop a widget from the palette to the Px page by default, the image from the Off folder will be displayed. If there is no graphic in the Off folder the image from the On folder will be used as a default.

Step 6. Bind the Widgets to the datapoint- see [Bind Widgets to the Datapoint](#).

Bind Widgets to the Datapoint

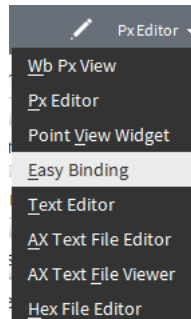
After adding widgets to the Px page, you need to bind the widget with the appropriate datapoint.

There are three types of binding:

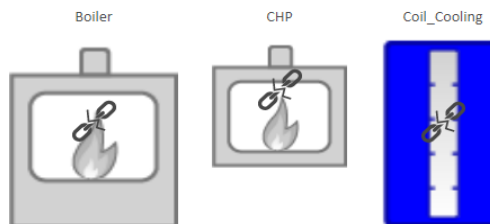
- **Value** - Binds the data point's value to the widget.
- **Alarm** - Binds the data point's alarm status to the widget (used by the alarm overlay)
- **Note:** If the Alarm binding is to be used the point that is to be monitored must have an alarm extension attached to it.
- **Override** - Binds the data point's override status to the widget (used by the override overlay)

To bind a widget with a datapoint

- Step 1. Open the Px page containing the widgets that are to be bound, any changes to the Px page must be saved.
- Step 2. In the view selector on the right select **Easy Binding**.

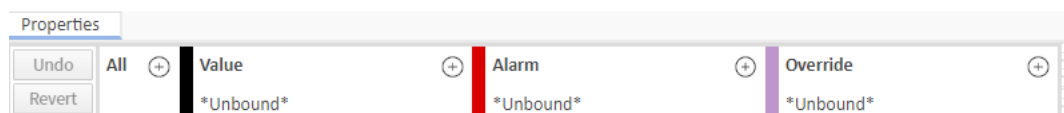


- Step 3. The **Easy Binding** window view is displayed.



Note: The chain icon displayed on the graphic means the graphic is not bound.

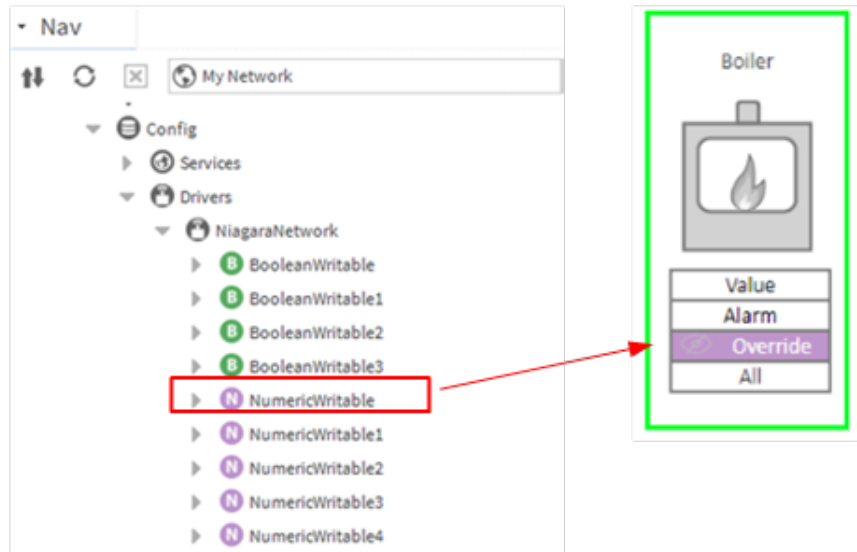
The toolbar is located at the bottom of the Easy Binding view.



Bind the data points with the graphics. This can be done in one of three ways:

Using drag and drop on the widget

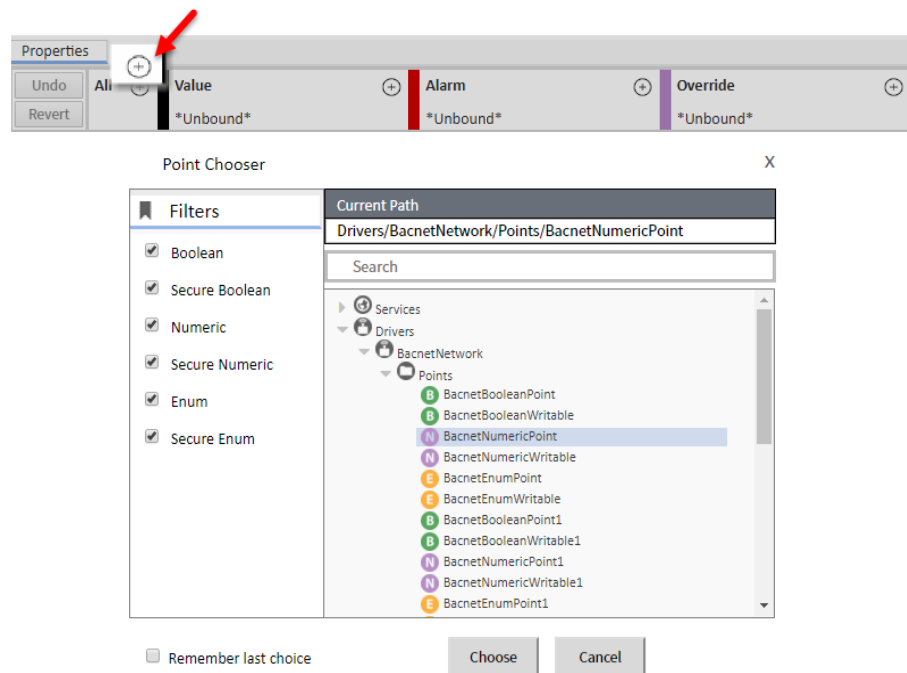
1. In the **Nav** tree select the required datapoint.
2. Drag the datapoint onto the binding type for the widget



This will bind the datapoint to the selected binding type for the widget.

Using Point Chooser



1. Select the widget.
2. Select (plus icon), from the required binding type in the toolbar or select the (plus icon) from the **All** area. The **Point Chooser** window displayed..



3. Enter the required datapoint name in the search field. Filtered datapoints are displayed.

Note: *If you select the **Remember last choice** option, the application will remember the last search history.*

*The displayed datapoints can be filtered by point type by selecting the **Boolean**, **Secure Boolean**, **Numeric**, **Secure Numeric**, **Enum**, or **Secure Enum** checkboxes.*

4. Select the required datapoint and click **Choose**. This will bind the selected datapoint to the selected binding type from which you selected the  (plus icon). If you selected the  (plus icon) from the All area the datapoint will be bound to all the binding types.

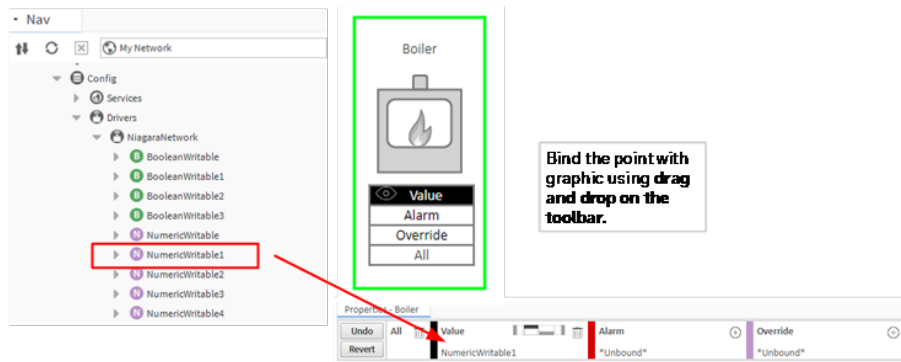
Using drag and drop on the toolbar

1. Select the widget. The toolbar displays the properties and binding points for the widget.



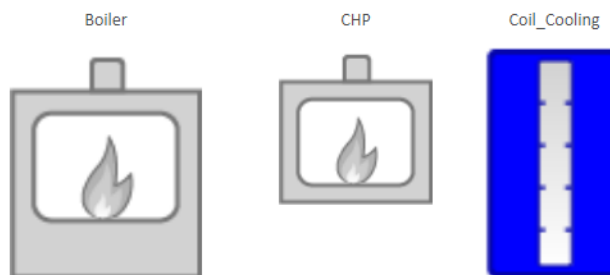
2. In the **Nav** tree select the required datapoint.

3. Drag the datapoint on to the area of the toolbar representing the required binding type.

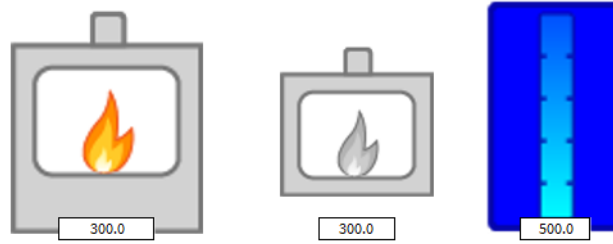


This will bind the datapoint to the selected binding type for the widget

After the widget has been bound to the required datapoint, its appearance will change.



You can switch to **Wb Px View**, to check the display is as required.



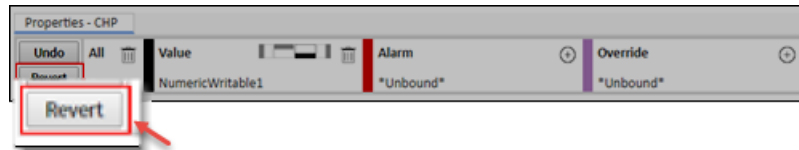
Reset Binding

To Revert the binding

The **Revert** option discards the changes for all the graphics in the Px page.

Step 1. Select the widget.

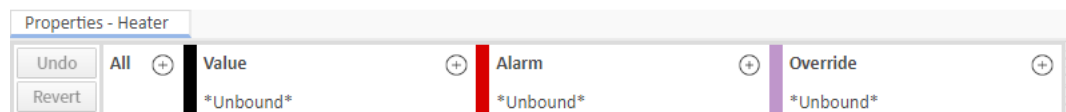
Step 2. Click **Revert** on the toolbar.



A dialogue box is displayed, asking if you want to continue.



Step 3. Click **Continue**. The changes implemented for all the graphics in the Px page will be discarded and graphics values will reset to their default value.

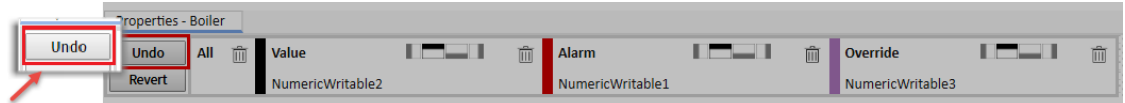


To Undo the last change

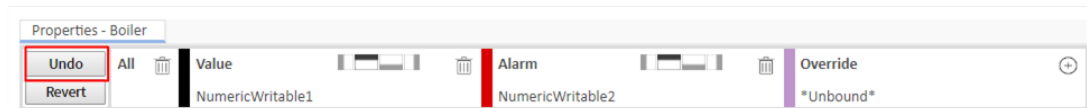
The **Undo** option allows to discard the last changes for the selected widget.

Step 1. Select the widget.

Step 2. Click **Undo** on the toolbar.

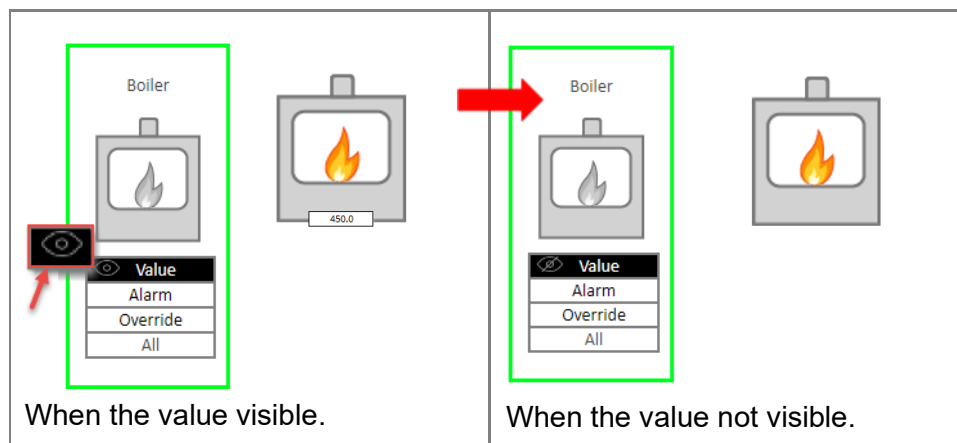


The last change to the binding will be discarded.



Turn Value Display ON/OFF

You can turn the display of the datapoints Value ON/OFF.



When the value visible.

When the value not visible.

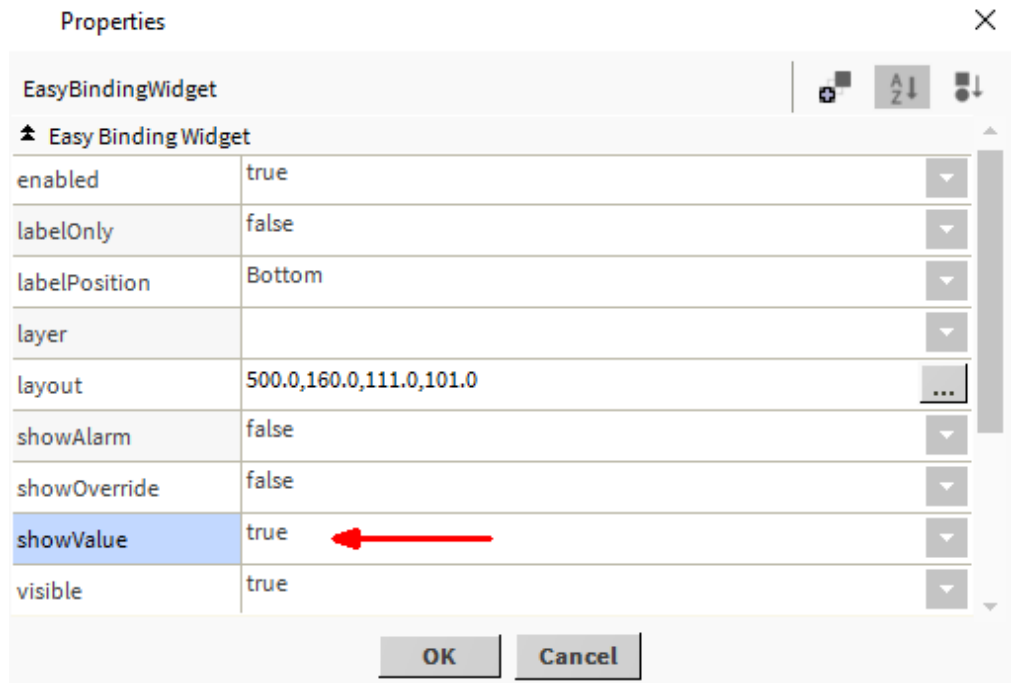
To turn label display ON/OFF

Step 1. Select the widget.

Step 2. Click  (eye icon) to toggle the value display.

Or

- Open the Px page in editor view and double-click the widget to display its property sheet.

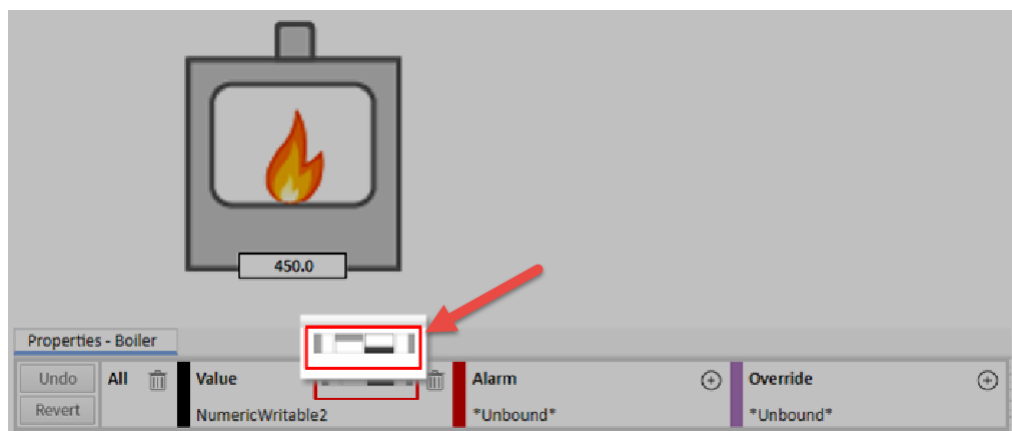






- Set the **showValue** parameter to 'true' or ' false' as required.
- Click **OK**.

Change Value Alignment

To change the value alignment

- Step 1. Select the widget.
- Step 2. Select the required value alignment option from the appropriate binding type section in the toolbar.



-  - Align to left
-  - Align to top
-  - Align to bottom
-  - Align to right


The value alignment is changed

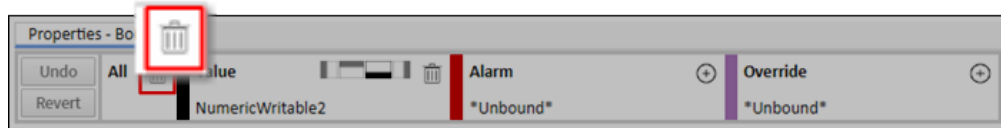
Delete Binding

You can delete some or all bindings for a widget.

To delete all the bindings


Step 1. Select the widget.

Step 2. Click  (delete icon) from the **All** section. All the bindings for the selected widget will be deleted.



To delete a specific binding

Step 1. Select the widget.

Step 2. Click  (delete icon) from the required binding type section of the toolbar. The binding for that binding type will be deleted.



Using the Widgets

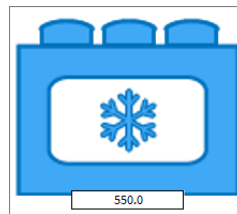
In normal operation widgets display the datapoint value that is bound with the widget value slot.



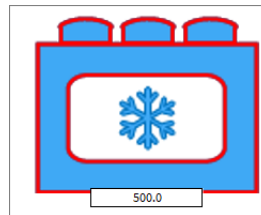
The widget allows the value of the bound datapoint to be changed.

To change the datapoint value:


- Step 1. Right-click the widget and select > **Action** > **Set**.
- Step 2. Enter the value and click **OK**.

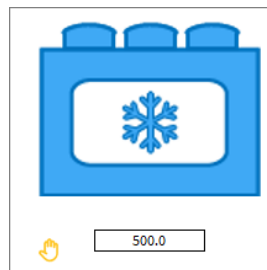


If a the datapoint is bound to the widget's alarm slot, the widget will generate an alarm and indicate an alarm condition when the low or high limit is reached.



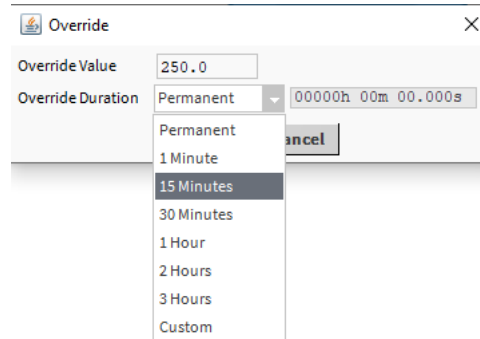
The low or high limit value can be changed. For details of configuring the alarm extension refer to the Niagara documentation or <module://docAlarms/doc/alarm-AlarmSource-Ext.html> Ord link.

If a datapoint is bound to the widget's override slot, the widget will indicate the override status with a  (hand icon). The override value and duration can be changed.



To change the override value and duration:

- Step 1. Right-click the widget and select **Actions > Active** or **Inactive**. The **Override** box is displayed.
- Step 2. Specify the required value in the **Override Value** box.
- Step 3. Select the required override duration from the **Override Duration** box.



The default override duration is permanent meaning the override value will remain effective until the next time this action is auto-detected. Other timed durations are available (1 Minute, 15 Minutes, 30 minutes, 1 hour, 2 Hours, 3 Hours, Custom). Custom enables any duration to be specified (hours, minutes, and seconds).

- Step 4. Click **OK**.

To remove the override:

- Step 1. Right-click the datapoint and select **Actions > Auto**.

Note: Buttons could be added to the px page to perform these actions.

Contact

Saia-Burgess Controls AG

Bahnhofstrasse 18

3280 Murten

Switzerland.

Phone..... +41 26 580 30 00

Phone support..... +41 26 580 31 00

Fax..... +41 26 580 34 99

Email support: support@saia-pcd.com

Support site: www.sbc-support.com

SBC site: www.saia-pcd.com

International Representatives &

SBC Sales Companies: www.saia-pcd.com/contact

IMPORTANT



Saia Burgess Controls Technical Support are only able to provide support for Saia PCD® Supervisor and the Saia Burgess Controls driver features described in this manual. They are unable to provide support for 3rd party drivers and undocumented aspects of Saia PCD® Supervisor's operation.