Vision Component Customizers

The Vision module provides a number of customizers to configure components in ways that are more complex or detailed for basic properties.

The two main customizers are the Component Customizer and the Style Customizer. These two customizers are used repeatedly for many different components. For special purpose components like the Easy Chart, Table, Tab Strip, and Multi-State Button, they have their own special customizers for you to create your own custom properties.

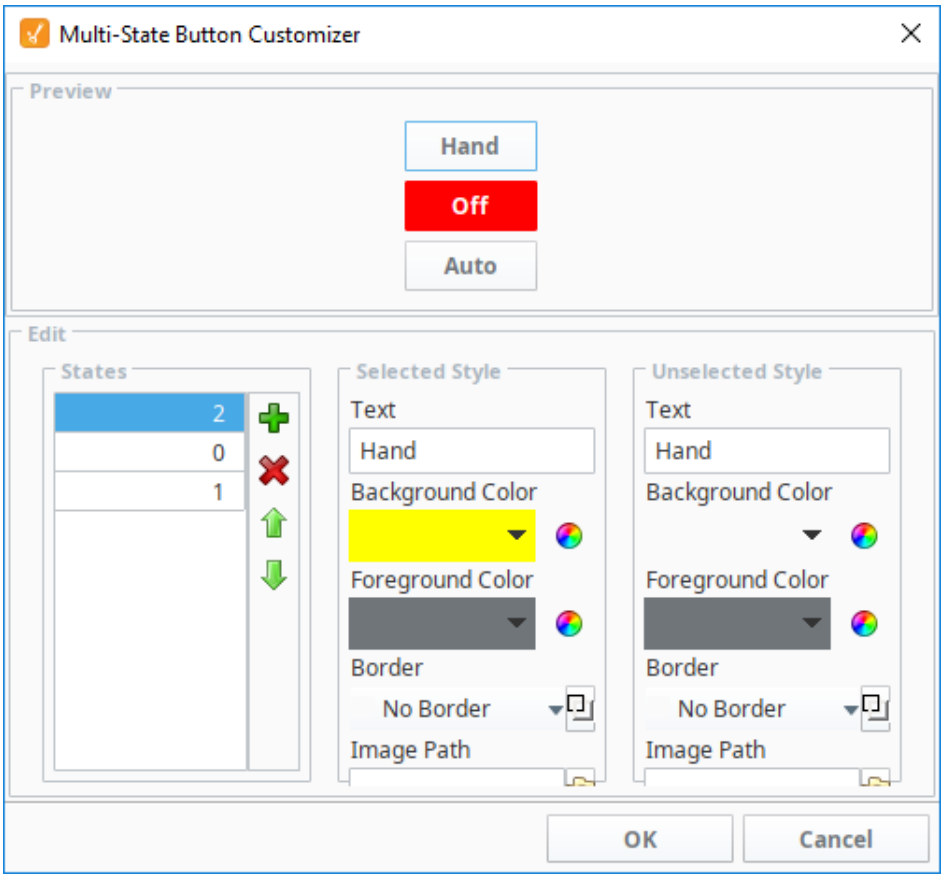
On this page

...

Component Customizers

To use a customizer, right-click on the component, choose **Customizers**, and select the customizer for the the component you are working with. You can also select the component and click the **Customizer**

icon in the Vision Main Toolbar at the top of the window. The following is an example of the customizer for the Multi-State Button Component.

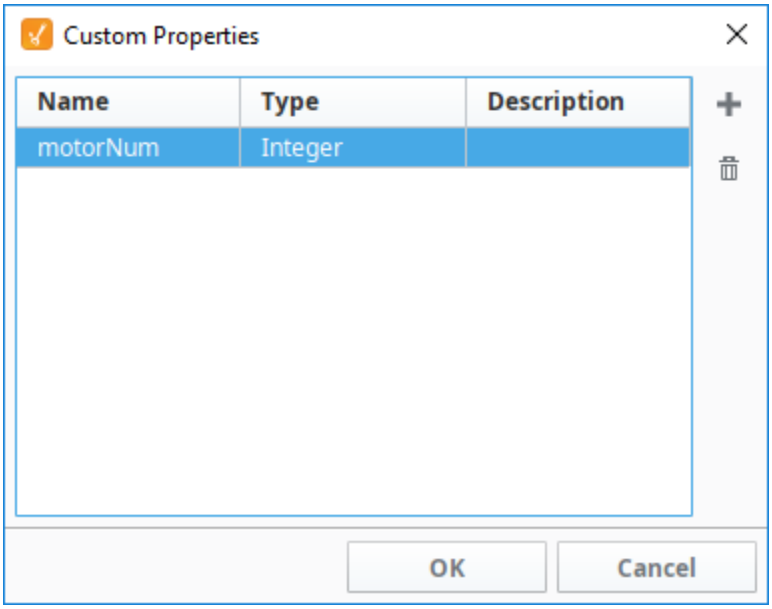


Expert Tip
Often, a Customizer works as a user-friendly user interface to one or more expert properties. For example, the **Easy Chart Customizer** modifies the contents of the pens, tagPens, calcPens, axes, and subplot dataset properties.This means you can also use Property Bindings and scripting to modify the values of these expert properties at runtime, giving you the ability to dynamically perform complex manipulations of components.

Custom Properties

In addition to the component's basic property settings, you can also create your own custom properties to enhance and add functionality to a component. You can use the custom properties like any other properties - with data binding, scripting, styles, and so on. Custom properties are important for passing parameters from one window to another, especially with a popup window. Properties on the window's Root Container are special in that they double as a window's parameters. For example, when you click on a Button component to open a popup window, it can pass a set of values into the window, which then get set to the custom property on the Root Container for use on that window.

1. To configure a custom property, right click on the component, and select **Customizers > Custom Properties**.
2. Click the plus
   
   icon to add a row. Enter the **Name** (i.e., motorNum) of the custom property and data **Type**. Click **OK.**
3. In the **Property Editor**, scroll to the bottom of the panel to see your custom property in blue.



Custom Properties can be any of the basic property types, but can also be a User Defined Type (UDT) from a UDT that was created earlier. The UDT property contains values for each of the Tags within the UDT, each of which can be bound to and used. The custom UDT property itself is typically bound to a UDT instance to pull in the entire instance Tag.

Style Customizer

Many components support the Style Customizer which lets you define a set of visual styles that change based on a single property. Typically, you'll have a property (often a custom property) on your component that you want to use as a driving property, usually a discrete state, and you have multiple visual properties, like the font, border, foreground color, visibility, and so on that you want to change based on that one driving property. Style Customizer enables you define these relationships all at once, and lets you preview them as well. Without styles, you would have to go to every property and bind them all individually.

Configuring the Style Customizer

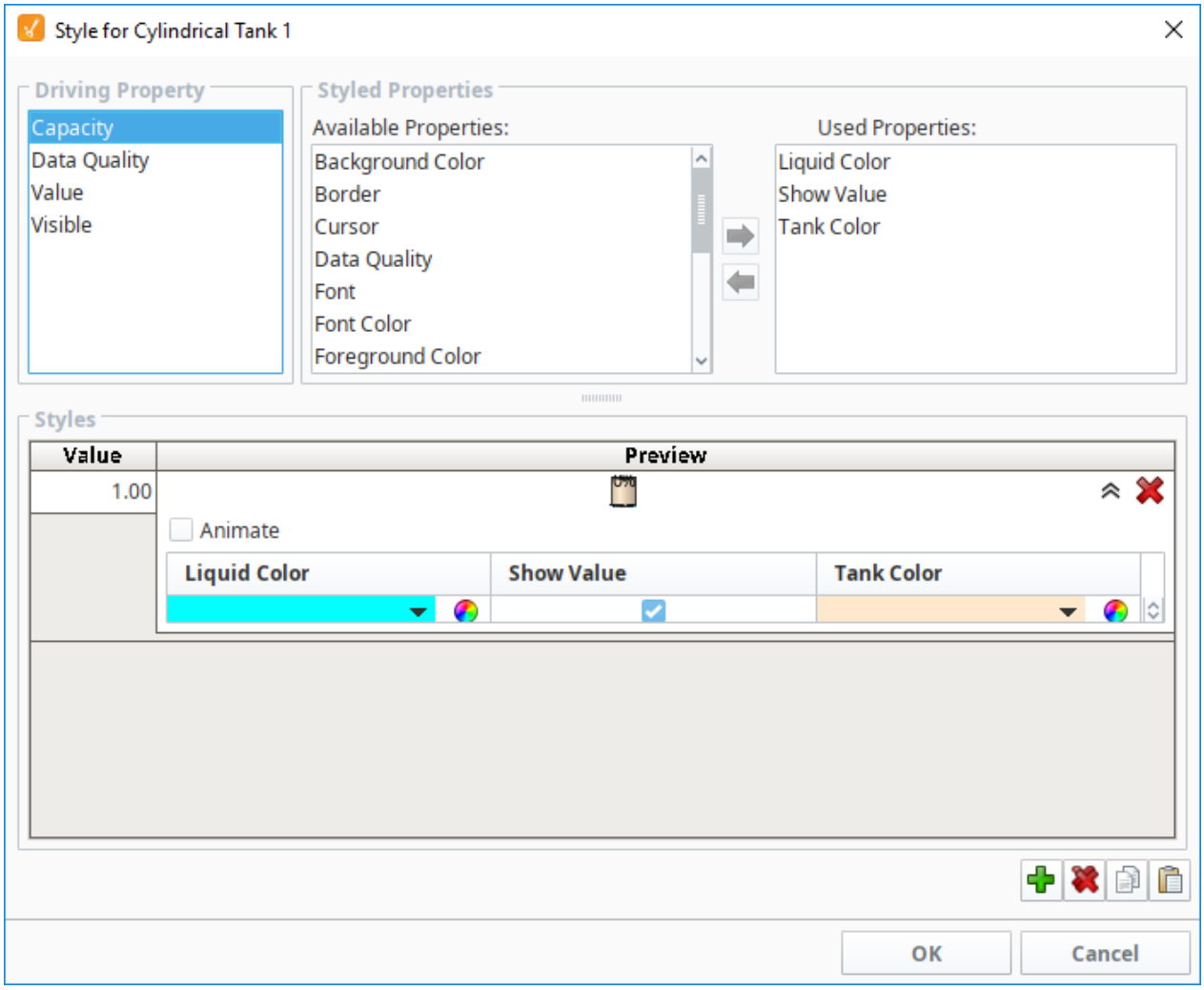
Some components have styles already setup and others do not. The following example involves a component that already has a styles defined.

1. Drag in a **Cylindrical tank** from the component palette on to your window.
2. Right click on the Cylindrical tank **component** and scroll down to **Customizers > Style Customizer**. There are four driving

properties that can have styles configured: Capacity, Data Quality, Value, and Visible.

3. For this example, click on **Capacity** > **Liquid Color** and then the **Add Property**

icon. Repeat this step for the **Show Value** and **Tank Color** Properties.

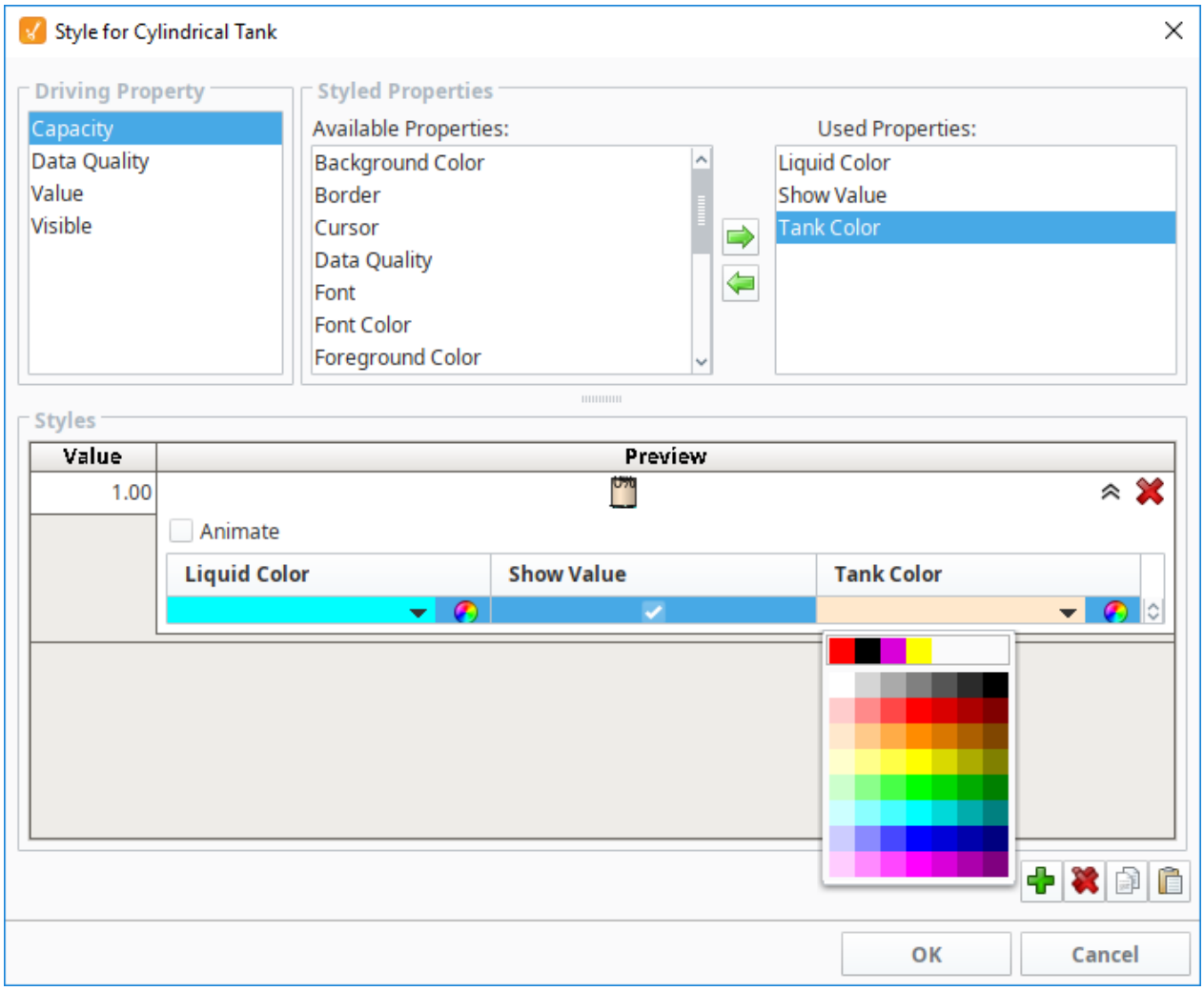


4. Next under Styles, click the **Add**

icon.

5. Click the **Expand**

icon to see the color palette for the Liquid Color.

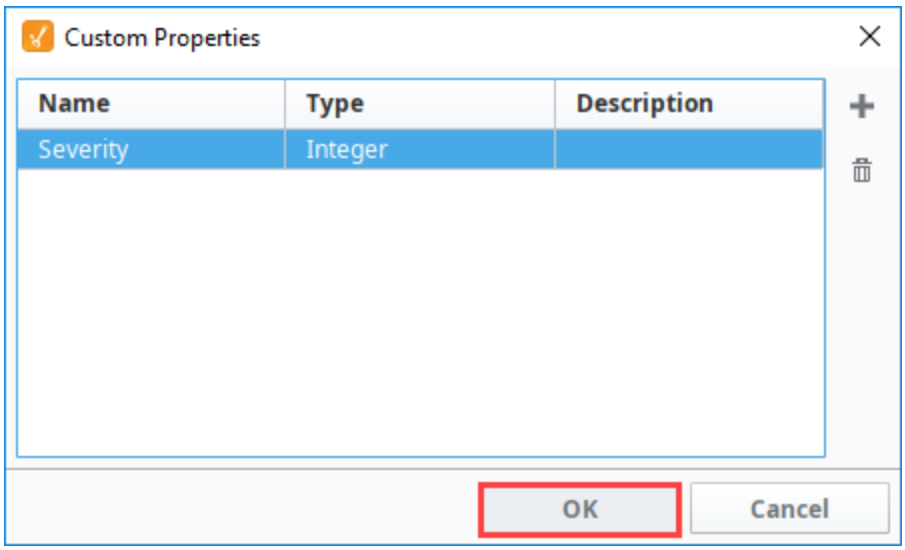


6. Choose a color from the palette. Repeat this step for the **Show Value** and **Tank Color** Properties.
7. Then click the **OK** to save your updates.

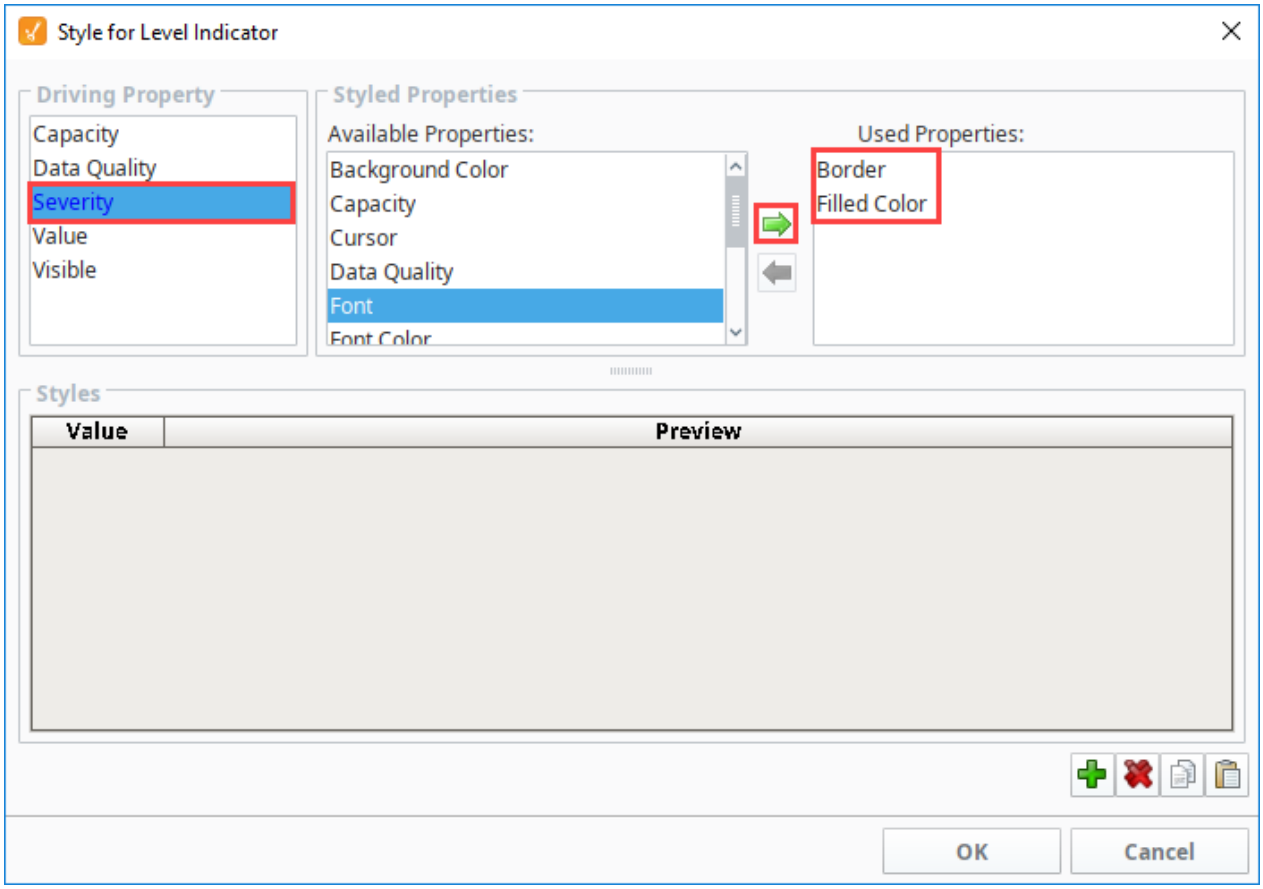
Example 1

In this example, we have a Level Indicator component that is displaying the level in a tank. Let's say that you want to have its appearance change based on the alarm state of the tank's temperature.

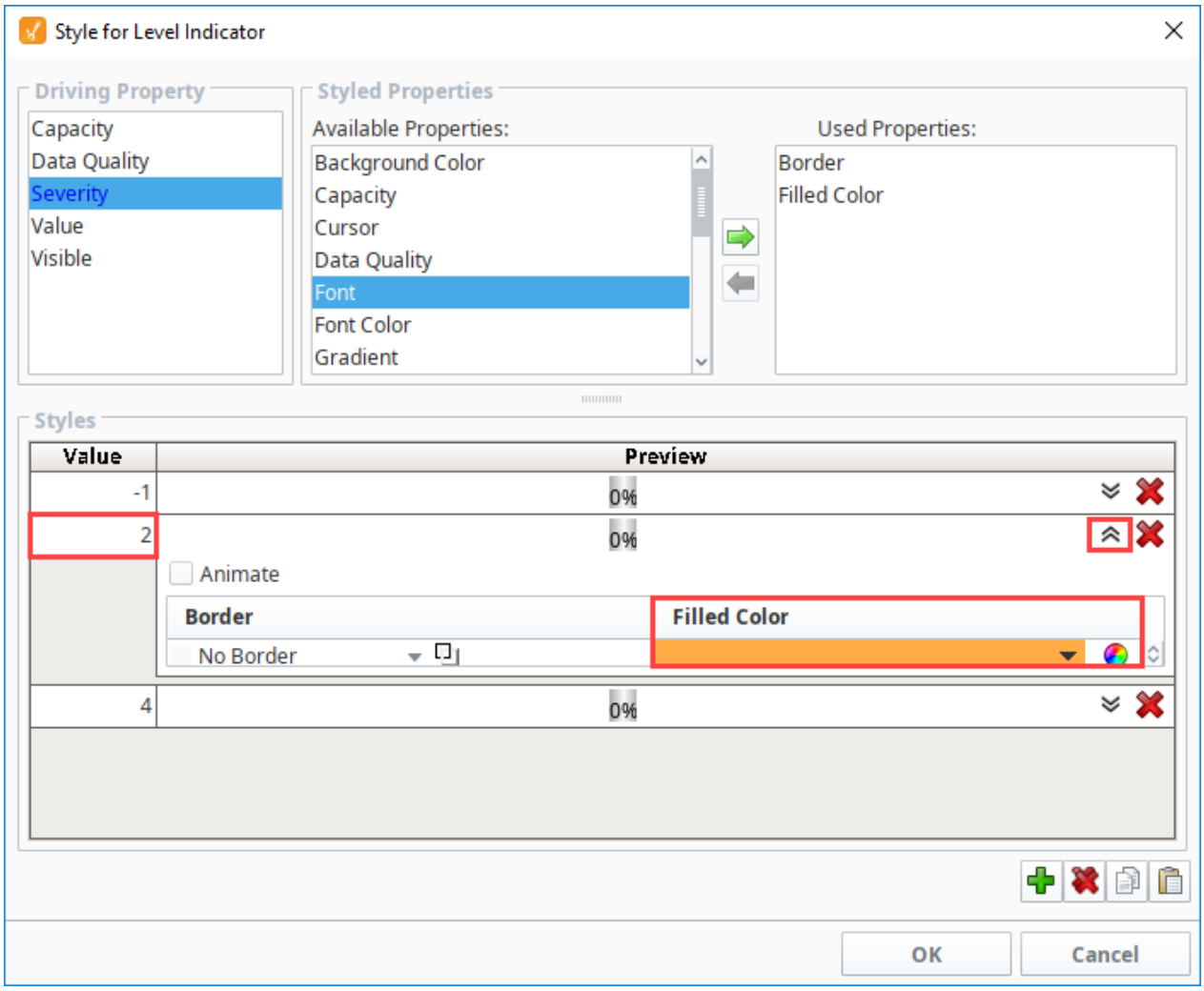
1. In the Vision window, right click on the component and choose **Custom Properties**.
2. Click the Add
   
   icon.
3. Name the new property **Severity** and set it to an Integer type. Click **OK**.



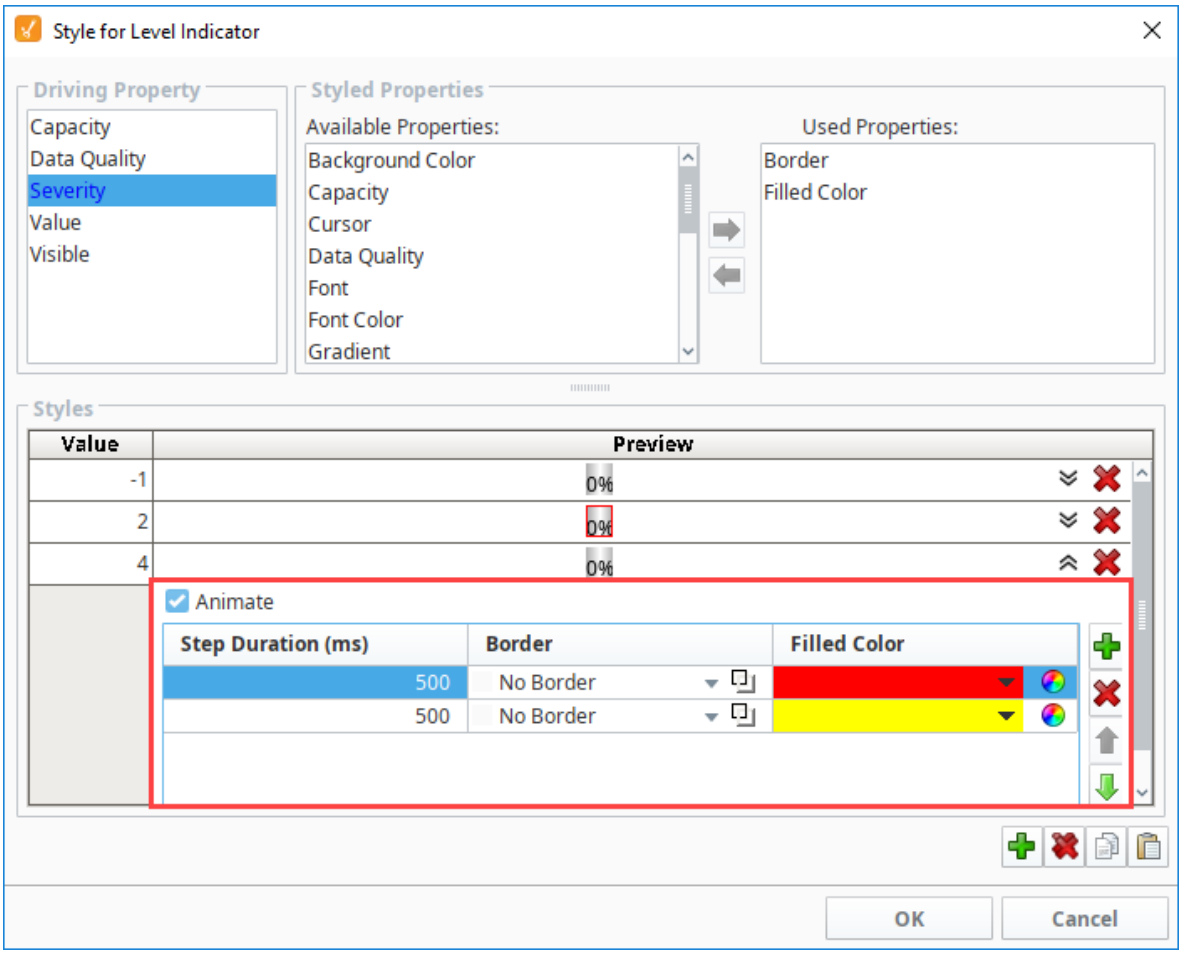
4. On the window, right click on the component and choose **Style Customizer**.
5. Choose your **Severity** property as the driving property, and the **Border** and **Filled Color** properties as the styled properties.



6. Under Styles, click the **Add** ✚ icon three times
7. Now create three styles for the three alarm states you want to show. For the first style, enter a value of -1 (not an alarm) and don't change anything else.
8. For the second, enter a value of 2 (medium alarm). Set the filled color to orange.

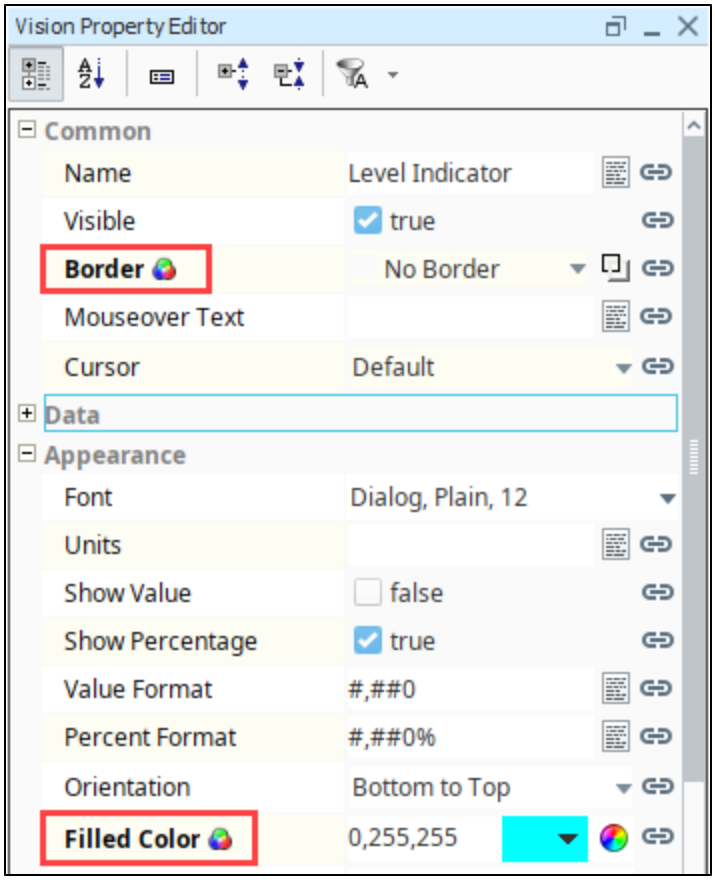


9. For the third style, enter a **Value** of 4 (high alarm).
    a. Click the **Expand**
    icon.
    b. Select the **Animate** checkbox.
    c. Click the **Add**
    icon.
    d. Set the **StepDuration** to 500 for both frames.
    e. For the first frame set the **FilledColor** to red.
    f. For the second frame, set the **FilledColor** to yellow.



10. Click **OK.** Notice that the styled properties you chose are now bold and have the **Styles**

icon next to them. This is to help remind you that those properties are being driven, so if you change their values directly, your changes will be overwritten.

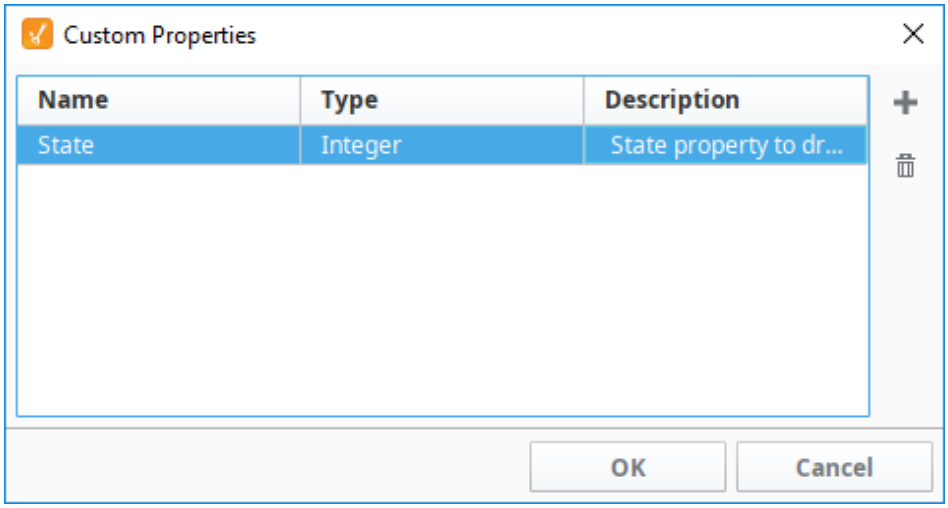


11. In the Property Editor, click on the **Binding** icon for the **Severity** custom property. Bind it to the tank temperature tag's **AlertCurrentSeverity** property.

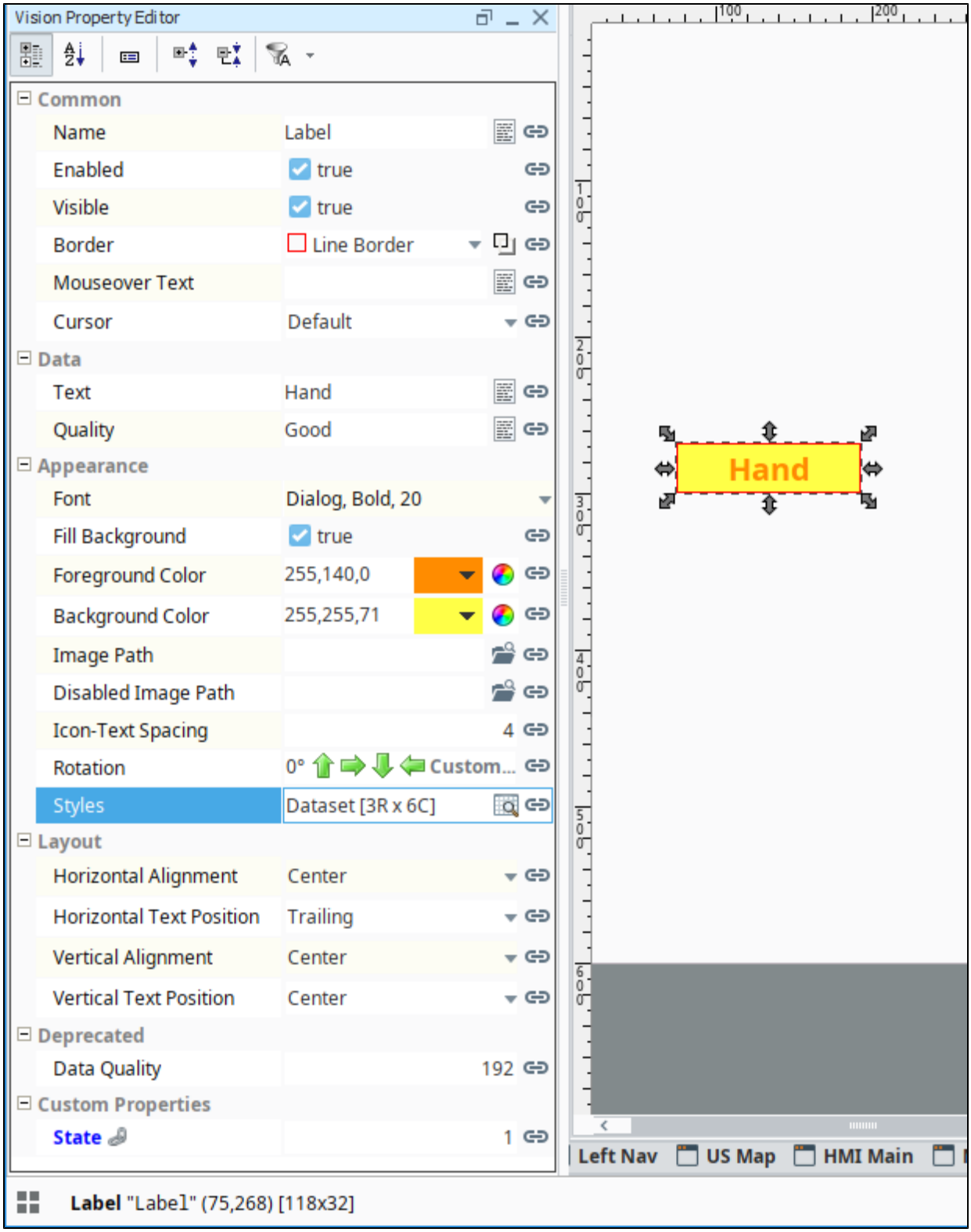
Example 2

Let's look at another example that uses the Custom Properties and the Styles feature together. For example, the Label component seems pretty plain at first: it just displays a string. You can use its foreground color, background color, and border to make it look interesting.

1. Drag a Label component onto a window.
2. Rght click on the Label component and choose **Custom Properties**.
3. Click the Add icon.
4. Name the new property **State** and set it to an **Integer** type. Click **OK**.



5. Bind that property to a discrete state Tag coming out of a PLC.
6. Next use the **State** property to drive its Styles configuration to make the component look different and display different things based on the state being 0, 1, or 2 (maybe for a Hand/Off/Auto indicator).



We could have used the Multi-State Indicator component from the very beginning, but understanding this example will let you create your own types of components by combining the existing components in creative ways.

Some components like the Easy Chart, Table, Power Table, Tab Strip, Multi-State Button, and Multi-State Indicator have default styles already setup, but you can modify them however you like. If you don't like the default styles, change them. They are there to simply help you get started.

Value Conflict

You can bind a property that is already being used by a style, but a warning icon will appear on the property, and the property name turns red in the In the Property Editor. This means there is a conflict between the binding on the property, and the style on the component. As a general practice, only the style or binding should write to the property, not both.