

1. Perspective Components	3
1.1 Perspective - Chart Palette	4
1.1.1 Perspective - Chart Range Selector	5
1.1.2 Perspective - Gauge	16
1.1.3 Perspective - Pie Chart	23
1.1.4 Perspective - Power Chart	29
1.1.5 Perspective - Simple Gauge	52
1.1.6 Perspective - Time Series Chart	55
1.1.7 Perspective - XY Chart	73
1.1.7.1 XY Chart Example - Candlestick Chart	91
1.1.7.2 XY Chart Example - Gantt Chart	100
1.1.7.3 XY Chart Example - Heat Map	110
1.1.7.4 XY Chart Example - Line Chart Target Area	120
1.2 Perspective - Container Palette	123
1.2.1 Perspective - Breakpoint Container	124
1.2.1.1 Perspective - Breakpoint Container Scripting	127
1.2.2 Perspective - Column Container	128
1.2.2.1 Perspective - Column Container Scripting	133
1.2.3 Perspective - Coordinate Container	134
1.2.3.1 Perspective - Coordinate Container Scripting	137
1.2.4 Perspective - Flex Container	139
1.2.4.1 Perspective - Flex Container Scripting	148
1.2.5 Perspective - Tab Container	149
1.2.5.1 Perspective - Tab Container Scripting	155
1.2.6 Perspective - Split Container	156
1.2.6.1 Perspective - Split Container Scripting	158
1.3 Perspective - Display Palette	159
1.3.1 Perspective - Alarm Journal Table	160
1.3.1.1 Perspective - Alarm Journal Table Scripting	170
1.3.2 Perspective - Alarm Status Table	172
1.3.2.1 How To Restrict Acknowledgement on the Perspective Alarm Status Table	189
1.3.2.2 Perspective - Alarm Status Table Scripting	190
1.3.3 Perspective - Audio	193
1.3.3.1 Perspective - Audio Scripting	195
1.3.4 Perspective - Barcode	198
1.3.5 Perspective - Cylindrical Tank	202
1.3.6 Perspective - Dashboard	205
1.3.6.1 Configuring a Dashboard	211
1.3.7 Perspective - Equipment Schedule	228
1.3.7.1 Perspective - Equipment Schedule Scripting	232
1.3.8 Perspective - Google Map	238
1.3.8.1 Perspective - Google Map Scripting	252
1.3.9 Perspective - Icon	270
1.3.10 Perspective - Image	272
1.3.11 Perspective - Inline Frame	275
1.3.12 Perspective - Label	277
1.3.13 Perspective - LED Display	279
1.3.14 Perspective - Linear Scale	281
1.3.15 Perspective - Map	285
1.3.15.1 Perspective Map - Adding GeoJSON Shapes	303
1.3.15.2 Perspective Map - Custom Controls Example	323
1.3.15.3 Perspective - Map Scripting	327
1.3.16 Perspective - Markdown	333
1.3.17 Perspective - Moving Analog Indicator	338
1.3.18 Perspective - PDF Viewer	341
1.3.18.1 Perspective - PDF Viewer Scripting	343
1.3.19 Perspective - Progress	344
1.3.20 Perspective - Sparkline	347
1.3.21 Perspective - Table	353
1.3.21.1 Displaying a Subview in a Table Row	370
1.3.21.2 Perspective - Table Scripting	380
1.3.21.3 Table Column Configurations	389
1.3.22 Perspective - Tag Browse Tree	398
1.3.22.1 Perspective - Tag Browse Tree Scripting	401
1.3.23 Perspective - Thermometer	403
1.3.24 Perspective - Tree	406
1.3.24.1 Perspective - Tree Scripting	410
1.3.25 Perspective - Video Player	411
1.4 Perspective - Embedding Palette	415
1.4.1 Perspective - Accordion	416
1.4.1.1 Perspective - Accordion Scripting	421
1.4.2 Perspective - Carousel	422
1.4.2.1 Carousel Component Examples	426
1.4.3 Perspective - Embedded View	439
1.4.4 Perspective - Flex Repeater	442
1.4.5 Perspective - View Canvas	447
1.4.5.1 Perspective - View Canvas Scripting	449
1.5 Perspective - Input Palette	452
1.5.1 Perspective - Barcode Scanner Input	453

1.5.2 Perspective - Button	455
1.5.3 Perspective - Checkbox	458
1.5.4 Perspective - DateTime Input	461
1.5.5 Perspective - DateTime Picker	463
1.5.6 Perspective - Dropdown	467
1.5.7 Perspective - File Upload	473
1.5.7.1 Download and Upload Files	474
1.5.7.2 Perspective - File Upload Scripting	478
1.5.8 Perspective - Multi-State Button	481
1.5.9 Perspective - Numeric Entry Field	484
1.5.10 Perspective - One-Shot Button	488
1.5.11 Perspective - Password Field	491
1.5.12 Perspective - Radio Group	493
1.5.13 Perspective - Signature Pad	495
1.5.13.1 Perspective - Signature Pad Scripting	500
1.5.14 Perspective - Slider	503
1.5.15 Perspective - Text Area	505
1.5.16 Perspective - Text Field	507
1.5.17 Perspective - Toggle Switch	509
1.6 Perspective - Navigation Palette	511
1.6.1 Perspective - Horizontal Menu	512
1.6.1.1 Navigating with the Horizontal Menu Component	518
1.6.1.2 Perspective - Horizontal Menu Scripting	527
1.6.2 Perspective - Link	529
1.6.3 Perspective - Menu Tree	530
1.6.3.1 Perspective - Menu Tree Scripting	539
1.7 Perspective - Report Viewer	541
1.8 Perspective - Symbols Palette	546
1.8.1 Perspective - Motor	547
1.8.2 Perspective - Pump	550
1.8.3 Perspective - Sensor	553
1.8.4 Perspective - Valve	555
1.8.5 Perspective - Vessel	557
1.9 Perspective - View Object	560

Perspective Components

This section covers all the built-in Perspective components. While a component is selected, you can use the [Property Editor](#) to alter the component's properties, which changes the component's appearance and behavior. Components are the building blocks of the Designer Interface that when combined create the visual part of a view to do something useful, like display dynamic information or control a device.

Here is a complete list of Perspective components and a link pointing to a page containing the component's description, properties, and usage examples.



[Perspective - Chart Palette](#)

[Perspective - Container Palette](#)

[Perspective - Embedding Palette](#)

[Perspective - Display Palette](#)

[Perspective - Navigation Palette](#)

[Perspective - View Object](#)

[Perspective - View Object](#)

[Perspective - Report Viewer](#)

[Perspective - Report Viewer](#)

[Perspective - Input Palette](#)

[Perspective - Symbols Palette](#)

Perspective - Chart Palette

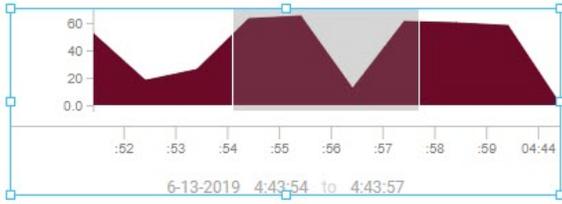
Chart Components

Charts allow you to display and show off your data in a graphical way.

The following is a complete list of Chart components that give you various options for displaying data, and a link pointing to a page containing the component's description, properties, and an example of how to configure it.

[In This Section ...](#)

Perspective - Chart Range Selector



Component Palette Icon:



On this page ...

- [User Interaction](#)
- [Properties](#)
- [Component Events](#)
- [Examples](#)
 - [Example 1 - Using a Format Property in a prop.timeAxis.tick.label](#)
 - [Example 2 - Using the Chart Range Selector](#)

The Chart Range Selector component is a small recreation of a chart that operators can use to select a time range based on seeing the existing data. This component complements the [Time Series Chart](#) component, and should always be used with a chart. Its features include:

- Zoom and pan in/out via mouse wheel interaction.
- Click-and-drag brush range selection and panning.
- Start and End property values that are updated as the brush range changes. These properties can govern the start/end points of data queries to return a dataset.
- Time range showing the overall range of the data being displayed by the brush. (The range updates as the brush is updated.)
- Simple display customization for the axes, baselines, markers, and the overall chart data appearance.
- Label properties have their own dedicated styling properties, such as color and size.

User Interaction

Interaction	Description
Zoom	The user can zoom in and out on the Chart Range Selector, but can not zoom out past its standard level of zoom.
Pan	The user can pan across the Chart Range Selector.
Refresh	The Chart Range Selector will not refresh its time range if it is zoomed in.
Pinch Zoom	On a mobile device, the user can now pinch-zoom the Chart Range Selector. Zooms must originate from within the boundaries of the displaying chart data.
Brushes	On a mobile device, the user can draw brushes in the Chart Range Selector via a single touch point (multiple touch points will allow zooming to occur). Brushes can be moved in the Chart Range Selector via a single touch point.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description
enablePanZoom	Allow the chart to be panned and zoomed. The chart cannot be zoomed out past its base range.

data	<p>Objects that are the data source for the chart (required). Data can be an object containing a time entry and value entries (all must be numbers) (required).</p> <p>Each value entry must be labeled with the column name to which it corresponds. Data can also be an array containing value entries (all must timestamp (which must be the first value) and one or more values that were captured at that time. Finally, data can also be in the form of a data bound to a Tag History binding to display either realtime data, or historical data (via start and end dates).</p>												
selectdRange	<p>The start and end points of the selected range (required). This property is updated as you interact with the brush.</p> <table border="1" data-bbox="256 344 841 483"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>start</td> <td>A Unix timestamp in milliseconds.</td> <td>value: numeric</td> </tr> <tr> <td>end</td> <td>A Unix timestamp in milliseconds.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	start	A Unix timestamp in milliseconds.	value: numeric	end	A Unix timestamp in milliseconds.	value: numeric			
Name	Description	Property Type											
start	A Unix timestamp in milliseconds.	value: numeric											
end	A Unix timestamp in milliseconds.	value: numeric											
brushRange	<p>An object used to control the display of the date/time range values at the bottom of the component.</p> <table border="1" data-bbox="256 554 1159 735"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>visible</td> <td>Whether the brush range is visible.</td> <td>value: boolean</td> </tr> <tr> <td>dateFormat</td> <td>The date format of the range using a MomentJS date string.</td> <td>value: string dropdown</td> </tr> <tr> <td>timeFormat</td> <td>The time format of the range using a MomentJS time string.</td> <td>value: string dropdown</td> </tr> </tbody> </table>	Name	Description	Property Type	visible	Whether the brush range is visible.	value: boolean	dateFormat	The date format of the range using a MomentJS date string.	value: string dropdown	timeFormat	The time format of the range using a MomentJS time string.	value: string dropdown
Name	Description	Property Type											
visible	Whether the brush range is visible.	value: boolean											
dateFormat	The date format of the range using a MomentJS date string.	value: string dropdown											
timeFormat	The time format of the range using a MomentJS time string.	value: string dropdown											
timeAxis	<p>This property provides settings for the X Axis. This property uses the same configuration as the timeAxis property of the Time Series Component.</p> <table border="1" data-bbox="256 848 1490 1197"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>visible</td> <td>The visible state of the the axis.</td> </tr> <tr> <td>tickCount</td> <td>The number of ticks on the axis (as a multiple of 2, 5, or 10).</td> </tr> <tr> <td>height</td> <td>The height of the axis.</td> </tr> <tr> <td>color</td> <td>The color of the axis. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selection</td> </tr> </tbody> </table>	Name	Description	visible	The visible state of the the axis.	tickCount	The number of ticks on the axis (as a multiple of 2, 5, or 10).	height	The height of the axis.	color	The color of the axis. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selection		
Name	Description												
visible	The visible state of the the axis.												
tickCount	The number of ticks on the axis (as a multiple of 2, 5, or 10).												
height	The height of the axis.												
color	The color of the axis. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selection												

tick	The configuration of the ticks on the axis.																							
	Click here to see the tick properties																							
	Name	Description																						
	color	The color of the ticks. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Selector .																						
	label	The configuration of the label drawn on the tick.																						
		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>angled</td> <td>If set to true, the tick labels will be angled rather than horizontal.</td> </tr> <tr> <td>format</td> <td>The date/time format displayed by each tick using a MomentJS date string (https://momentjs.com/). Options are: Auto, Millisecond [638], Second [:35], Hour Minute [8:15], Hour with Meridiem [8 AM], Day of Week, Month, and Hour [Monday 2nd, 08 AM], Abbreviated Day of Week and Month [Mon 2nd], Abbreviated Month and Day of Month [Jan 2nd], Full Month [January], Abbreviated Month and Year [Jan 20], Full Year [2020], [3-2-2020 8 :15:35], [2020-3-2 8:15:35], Unix Millisecond Timestamp [1563464737269] , Unix Timestamp [1563464737]. Default is "Auto", where the <code>property</code> attempts to figure out the best format. For a listing of suggested formats, refer to https://momentjs.com/docs/#/parsing/string-format/ Go to the Example at the bottom of this page to see the "Unit [Example] : Notation"</td> </tr> <tr> <td>font</td> <td>The settings for the label's font.</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color of the label text.</td> <td>value: string</td> </tr> <tr> <td>size</td> <td>The font size, in pixels, of the label text.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> </tr> <tr> <td>style</td> <td>Custom CSS styles to apply to the tick labels. Any style that applies to an SVG text element can be used. See also style options.</td> </tr> <tr> <td>style</td> <td>Custom CSS styles to apply to the ticks. Any style that applies to an SVG line element can be used. See also style options</td> </tr> </tbody></table>	Name	Description	angled	If set to true, the tick labels will be angled rather than horizontal.	format	The date/time format displayed by each tick using a MomentJS date string (https://momentjs.com/). Options are: Auto, Millisecond [638], Second [:35], Hour Minute [8:15], Hour with Meridiem [8 AM], Day of Week, Month, and Hour [Monday 2nd, 08 AM], Abbreviated Day of Week and Month [Mon 2nd], Abbreviated Month and Day of Month [Jan 2nd], Full Month [January], Abbreviated Month and Year [Jan 20], Full Year [2020], [3-2-2020 8 :15:35], [2020-3-2 8:15:35], Unix Millisecond Timestamp [1563464737269] , Unix Timestamp [1563464737]. Default is "Auto", where the <code>property</code> attempts to figure out the best format. For a listing of suggested formats, refer to https://momentjs.com/docs/#/parsing/string-format/ Go to the Example at the bottom of this page to see the "Unit [Example] : Notation"	font	The settings for the label's font.		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color of the label text.</td> <td>value: string</td> </tr> <tr> <td>size</td> <td>The font size, in pixels, of the label text.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color of the label text.	value: string	size	The font size, in pixels, of the label text.	value: numeric	style	Custom CSS styles to apply to the tick labels. Any style that applies to an SVG text element can be used. See also style options .	style
Name	Description																							
angled	If set to true, the tick labels will be angled rather than horizontal.																							
format	The date/time format displayed by each tick using a MomentJS date string (https://momentjs.com/). Options are: Auto, Millisecond [638], Second [:35], Hour Minute [8:15], Hour with Meridiem [8 AM], Day of Week, Month, and Hour [Monday 2nd, 08 AM], Abbreviated Day of Week and Month [Mon 2nd], Abbreviated Month and Day of Month [Jan 2nd], Full Month [January], Abbreviated Month and Year [Jan 20], Full Year [2020], [3-2-2020 8 :15:35], [2020-3-2 8:15:35], Unix Millisecond Timestamp [1563464737269] , Unix Timestamp [1563464737]. Default is "Auto", where the <code>property</code> attempts to figure out the best format. For a listing of suggested formats, refer to https://momentjs.com/docs/#/parsing/string-format/ Go to the Example at the bottom of this page to see the "Unit [Example] : Notation"																							
font	The settings for the label's font.																							
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color of the label text.</td> <td>value: string</td> </tr> <tr> <td>size</td> <td>The font size, in pixels, of the label text.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color of the label text.	value: string	size	The font size, in pixels, of the label text.	value: numeric														
Name	Description	Property Type																						
color	Color of the label text.	value: string																						
size	The font size, in pixels, of the label text.	value: numeric																						
style	Custom CSS styles to apply to the tick labels. Any style that applies to an SVG text element can be used. See also style options .																							
style	Custom CSS styles to apply to the ticks. Any style that applies to an SVG line element can be used. See also style options																							

grid	<div style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.16 Click here to check out the other new features</p> </div>	
	Configuration for gridlines to display on this axis.	
	Name	Description
	visible	Visible state of the gridlines. Gridlines are shown only for axes that connect directly to the chart. Any satellite axes will their tick configurations instead of gridlines.
	color	Color of the gridlines.
	opacity	Opacity of the gridlines.
dashArray	Dashed appearance of the gridlines.	
style	Style for the gridlines. Full menu of style options is available. You can also specify a style class .	
style	A style object containing properties which are applied to the horizontal line of the axis. Any property that would apply to an SVG line. See also style options .	

yAxis

An object used to control the display of the Y Axis. This component was built with the assumption that this axis may be secondary information

Name	Description																					
visible	The visible state of the the axis.																					
width	The width of the axis, in pixels (required).																					
label	The configuration of the Y axis label. <table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>visible</td><td>Whether or not the label is visible.</td></tr><tr><td>text</td><td>The text for the label.</td></tr><tr><td>offset</td><td>Offset the Y axis label from its default position. This allows you to fine tune the label location, which may be necessary on the scale and how much room the tick labels take up. This may be positive or negative.</td></tr><tr><td>font</td><td>The settings for the label's font. <table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>color</td><td>The color of the label text.</td><td>value: string</td></tr><tr><td>size</td><td>The font size, in pixels, of the label text.</td><td>value: numeric</td></tr></tbody></table></td></tr><tr><td>style</td><td>Custom CSS styles to apply to the Y axis label. Any style that applies to an SVG text element can be used. See also</td></tr></tbody></table>	Name	Description	visible	Whether or not the label is visible.	text	The text for the label.	offset	Offset the Y axis label from its default position. This allows you to fine tune the label location, which may be necessary on the scale and how much room the tick labels take up. This may be positive or negative.	font	The settings for the label's font. <table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>color</td><td>The color of the label text.</td><td>value: string</td></tr><tr><td>size</td><td>The font size, in pixels, of the label text.</td><td>value: numeric</td></tr></tbody></table>	Name	Description	Property Type	color	The color of the label text.	value: string	size	The font size, in pixels, of the label text.	value: numeric	style	Custom CSS styles to apply to the Y axis label. Any style that applies to an SVG text element can be used. See also
Name	Description																					
visible	Whether or not the label is visible.																					
text	The text for the label.																					
offset	Offset the Y axis label from its default position. This allows you to fine tune the label location, which may be necessary on the scale and how much room the tick labels take up. This may be positive or negative.																					
font	The settings for the label's font. <table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>color</td><td>The color of the label text.</td><td>value: string</td></tr><tr><td>size</td><td>The font size, in pixels, of the label text.</td><td>value: numeric</td></tr></tbody></table>	Name	Description	Property Type	color	The color of the label text.	value: string	size	The font size, in pixels, of the label text.	value: numeric												
Name	Description	Property Type																				
color	The color of the label text.	value: string																				
size	The font size, in pixels, of the label text.	value: numeric																				
style	Custom CSS styles to apply to the Y axis label. Any style that applies to an SVG text element can be used. See also																					

tick

The configuration for the ticks drawn on the axis.

Name	Description
color	The color of the ticks. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Selector .
count	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;">The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</div> Number of ticks to display on the Y axis. Default value is Auto. When the tick count is Auto, Y axis ticks will be added based on the height of the chart, and the tick spacing is adjusted around whole numbers.

label

Name	Description									
format	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;">The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</div> Sets the numeric format for the tick label. The value must be a valid D3 Format value. Default value is Auto. When the tick label format is Auto, the D3 format <code>~f</code> is used. This creates a fixed point that will trim any insignificant trailing zeroes. <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"><u>This feature was changed in Ignition version 8.1.13.</u></div> The Y axis now supports numeric locale formatting. Numeric values are automatically formatted based on the session locale and the <code>tick.label.format</code> property.									
font	The font style for the label. <table border="1" style="width: 100%; border-collapse: collapse;"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>color</td><td>The color of the label text.</td><td>value: string</td></tr><tr><td>size</td><td>The font size, in pixels, of the label text.</td><td>value: numeric</td></tr></tbody></table>	Name	Description	Property Type	color	The color of the label text.	value: string	size	The font size, in pixels, of the label text.	value: numeric
Name	Description	Property Type								
color	The color of the label text.	value: string								
size	The font size, in pixels, of the label text.	value: numeric								
style	Custom CSS styles to apply to the baseline label. Any style that applies to an SVG <code>text</code> element can be used. See also style options .									
style	Custom CSS styles to apply to the ticks. Any style that applies to an SVG <code>text</code> element can be used. See also style									

grid	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.16 Click here to check out the other new features</p> </div> <p>Configuration for gridlines to display on this axis.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Name</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>visible</td> <td>Visible state of the gridlines. Gridlines are shown only for axes that connect directly to the chart. Any satellite axes will their tick configurations instead of gridlines.</td> </tr> <tr> <td>color</td> <td>Color of the gridlines.</td> </tr> <tr> <td>opacity</td> <td>Opacity of the gridlines.</td> </tr> <tr> <td>dashArray</td> <td>Dashed appearance of the gridlines.</td> </tr> <tr> <td>style</td> <td>Style for the gridlines. Full menu of style options is available. You can also specify a style class.</td> </tr> </tbody> </table>	Name	Description	visible	Visible state of the gridlines. Gridlines are shown only for axes that connect directly to the chart. Any satellite axes will their tick configurations instead of gridlines.	color	Color of the gridlines.	opacity	Opacity of the gridlines.	dashArray	Dashed appearance of the gridlines.	style	Style for the gridlines. Full menu of style options is available. You can also specify a style class .
Name	Description												
visible	Visible state of the gridlines. Gridlines are shown only for axes that connect directly to the chart. Any satellite axes will their tick configurations instead of gridlines.												
color	Color of the gridlines.												
opacity	Opacity of the gridlines.												
dashArray	Dashed appearance of the gridlines.												
style	Style for the gridlines. Full menu of style options is available. You can also specify a style class .												
areaStyles	<p>An object providing default style to the chart trends as a whole.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Name</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>colorScheme</td> <td>A Color Brewer color scheme to use on the series. See ColorBrewer2.org for available color schemes.</td> </tr> <tr> <td>colors</td> <td>A list of colors to apply to the columns (in order) for each trend. If these values are provided, they will override the value provided by the colorScheme.</td> </tr> </tbody> </table>	Name	Description	colorScheme	A Color Brewer color scheme to use on the series. See ColorBrewer2.org for available color schemes.	colors	A list of colors to apply to the columns (in order) for each trend. If these values are provided, they will override the value provided by the colorScheme.						
Name	Description												
colorScheme	A Color Brewer color scheme to use on the series. See ColorBrewer2.org for available color schemes.												
colors	A list of colors to apply to the columns (in order) for each trend. If these values are provided, they will override the value provided by the colorScheme.												
style	<p>Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</p>												

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menu bar or by right clicking on the component.

Examples

Example 1 - Using a Format Property in a prop.timeAxis.tick.label

The new `format` property represents the preferred date/time format for the `timeAxis` property. You can enter any preferred date/time format as defined by <https://momentjs.com/docs/#/parsing/string-format/>. Below is a listing of suggested formats and how they can be used.

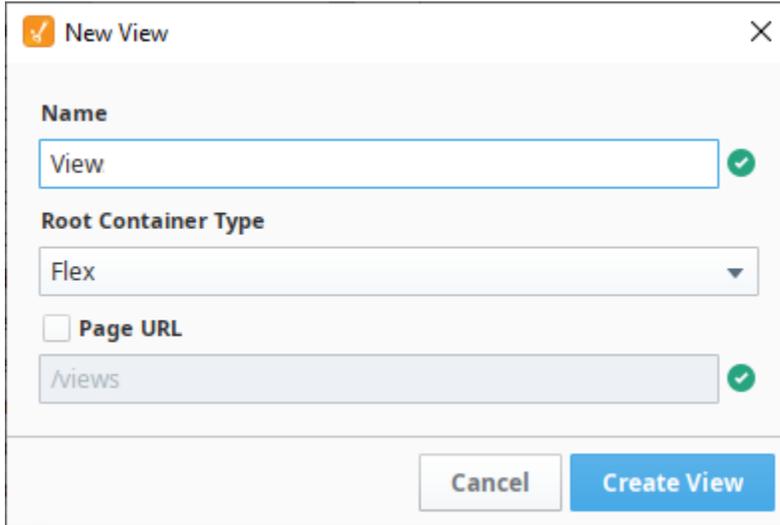
Unix Example : Notation

```
"Millisecond [638]": "SSS",
"Second [:35]": ":ss",
"Hour Minute [8:15]": "h:mm",
"Hour with Meridiem [8 AM]": "h A",
"Day of Week, Month, and Hour [Monday 2nd, 08 AM]": "dddd Do, hh A",
"Abbreviated Day of Week and Month [Mon 2nd]": "ddd Do",
"Abbreviated Month and Day of Month [Jan 2nd]": "MMM Do",
"Full Month [January]": "MMMM",
"Abbreviated Month and Year [Jan 20]": "MMM YY",
"Full Year [2020]": "YYYY",
"[3-2-2020 8:15:35]": "M-D-YYYY h:mm:ss",
"[2020-3-2 8:15:35]": "YYYY-M-D h:mm:ss",
"Unix Millisecond Timestamp [1563464737269]": "x",
"Unix Timestamp [1563464737]": "X"
```

Example 2 - Using the Chart Range Selector

To begin using the Chart Range Selector, a Time Series Chart with trend data will be needed. This example shows how to configure the Chart Range Selector.

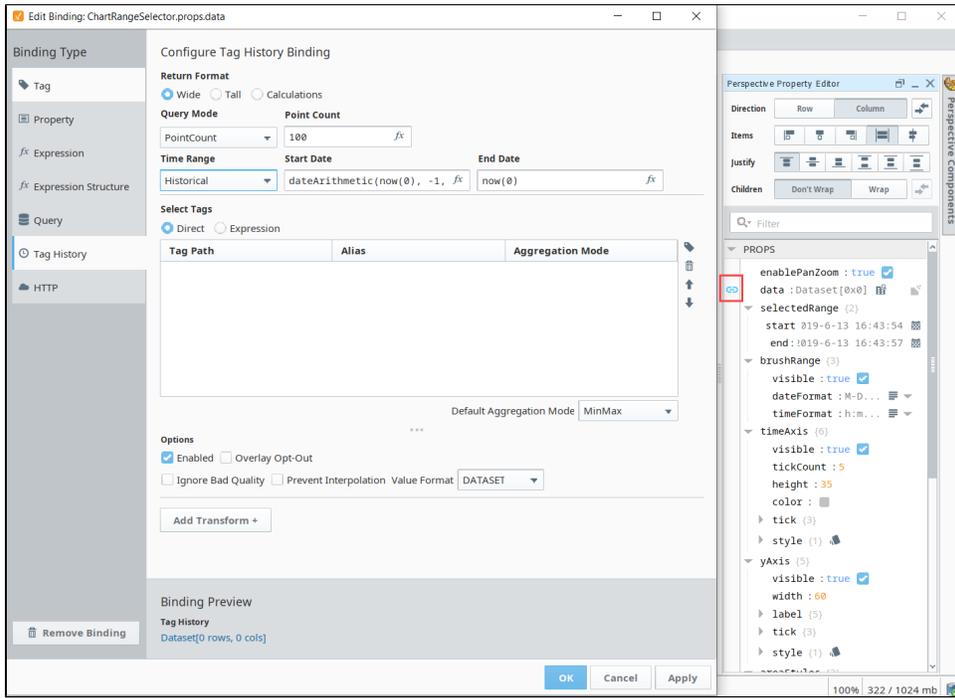
1. Begin by [configuring tag history](#) on a Tag of your choice.
2. From the Perspective section of the **Project Browser** on your Designer, right click on the **Views** folder and select **New View...** to create a new view.
3. This will bring up the New View window. Give your view a name and select the **Flex** Root Container Type. The Page URL setting will remain unchecked for this example.



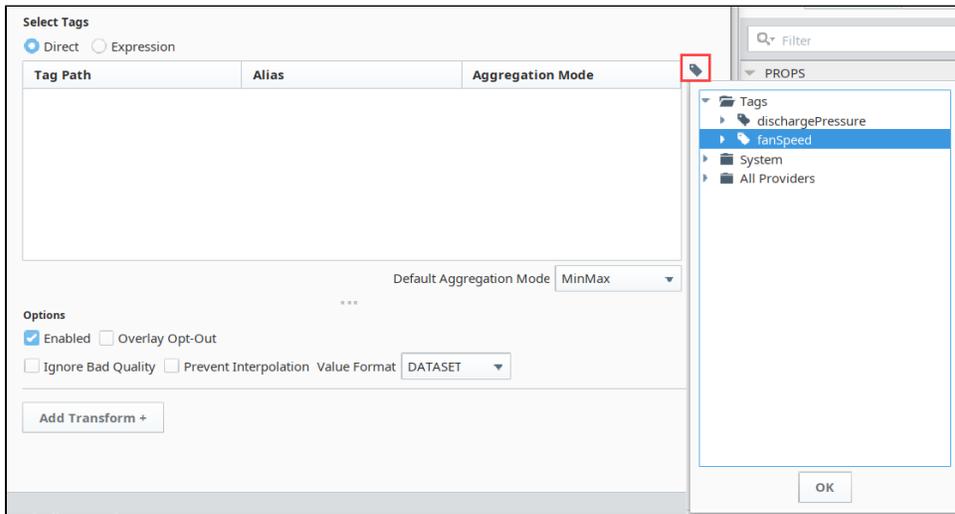
The image shows a 'New View' dialog box with the following fields and options:

- Name:** A text input field containing 'View' with a green checkmark icon to its right.
- Root Container Type:** A dropdown menu with 'Flex' selected.
- Page URL:** An unchecked checkbox followed by a text input field containing '/views' with a green checkmark icon to its right.
- Buttons:** 'Cancel' and 'Create View' buttons at the bottom right.

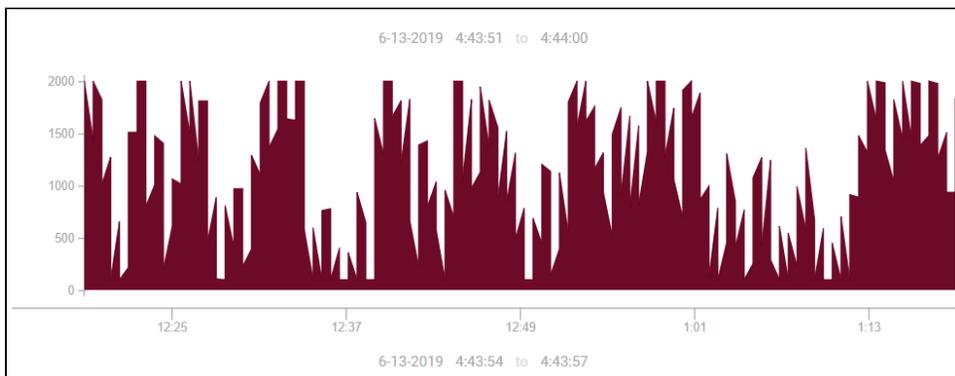
4. From the PerspectiveComponent Palette, drag and drop a **Time Series Chart** onto your newly created view. In the **Position** category of the Property Editor, set the **position.grow** property to '1'.
5. Drag and drop a Chart Range Selector component onto your view. Go to the **Position** category and set the Chart Range Selector's **position.grow** property to '1'.
6. With the Chart Range Selector selected, go to the **data** property, click on the binding icon  to bring up the Binding Editor window and select the **Tag History** binding type as shown in the image below.
7. Set the **Time Range** to **Historical**. We'll configure the binding to span the last one hour of historical data by making the following changes:
Start Date: `dateArithmetic(now(0), -1, 'hour')`
End Date: `now(0)`



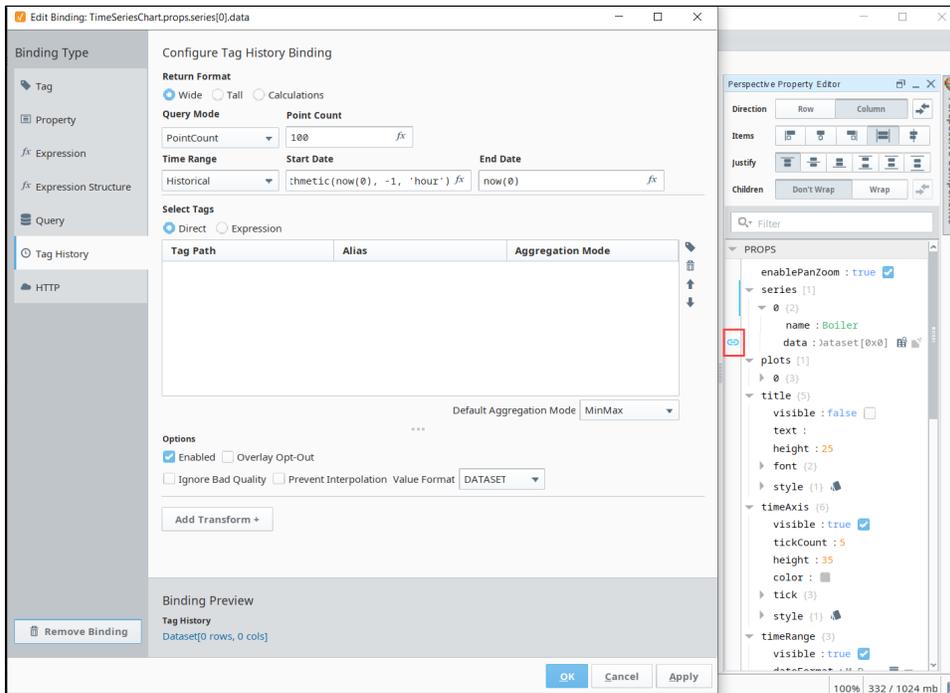
8. Under the Select Tags section, click on the **Tag** icon and use the Tag browser to drill down to the Tag you created in Step 1
9. Click **OK**.



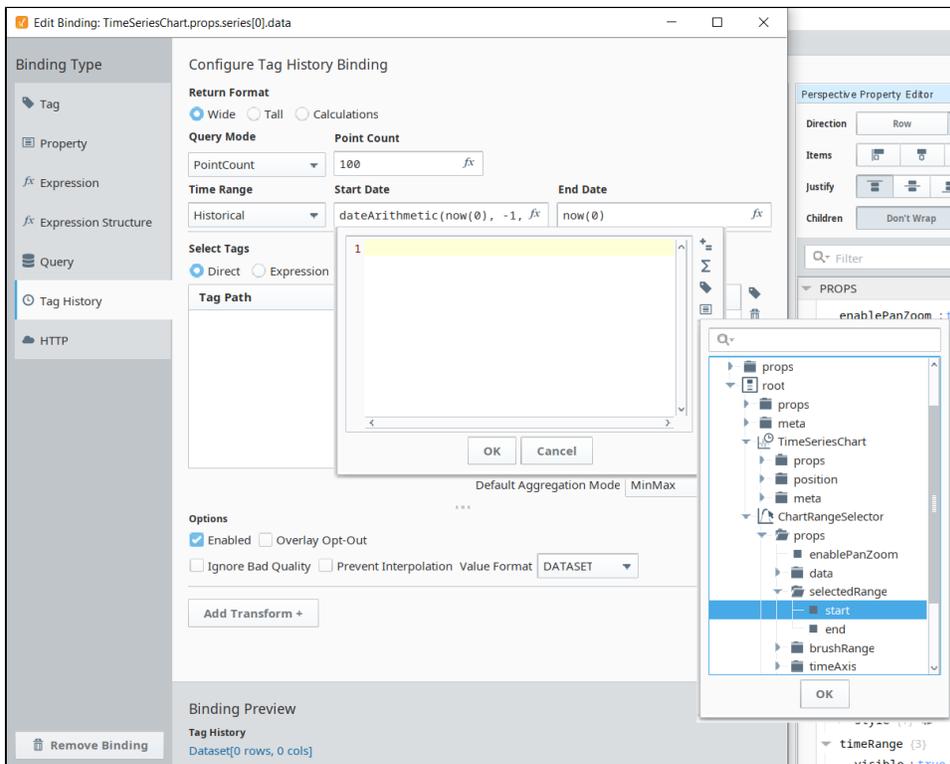
10. Click **OK** on the Binding Editor window to accept the binding changes. You should now have a Chart Range Selector displaying the last 1 hour of historical data for your Tag created in Step 1.



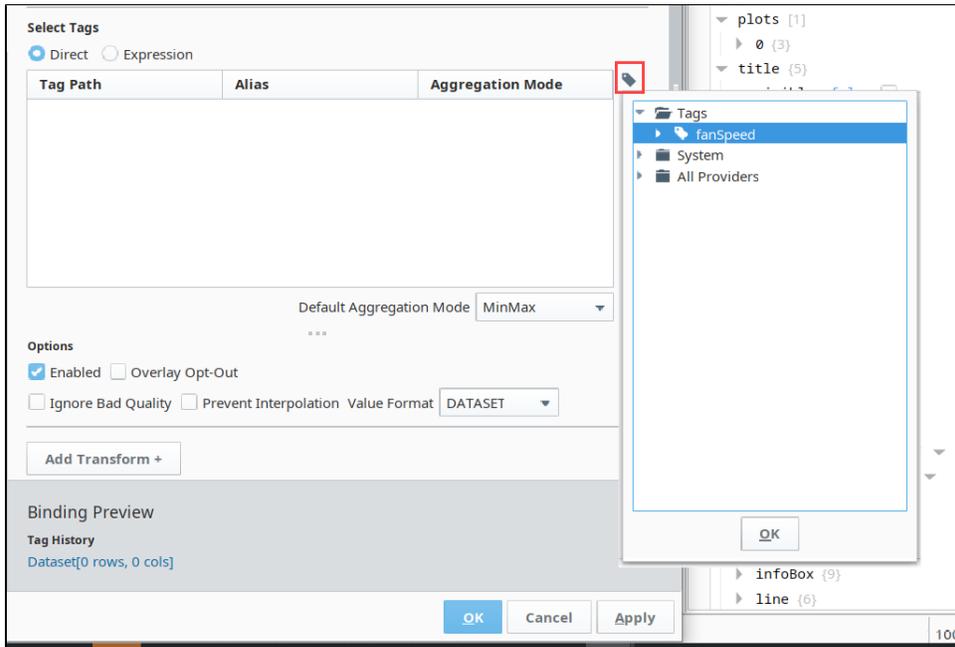
- Now select your Time Series Chart and from the Perspective Property Editor, and click on the **binding**  icon for the **series[0].data** property to open the Binding Editor window.
- Select the **Tag History** binding type as shown in the following image.



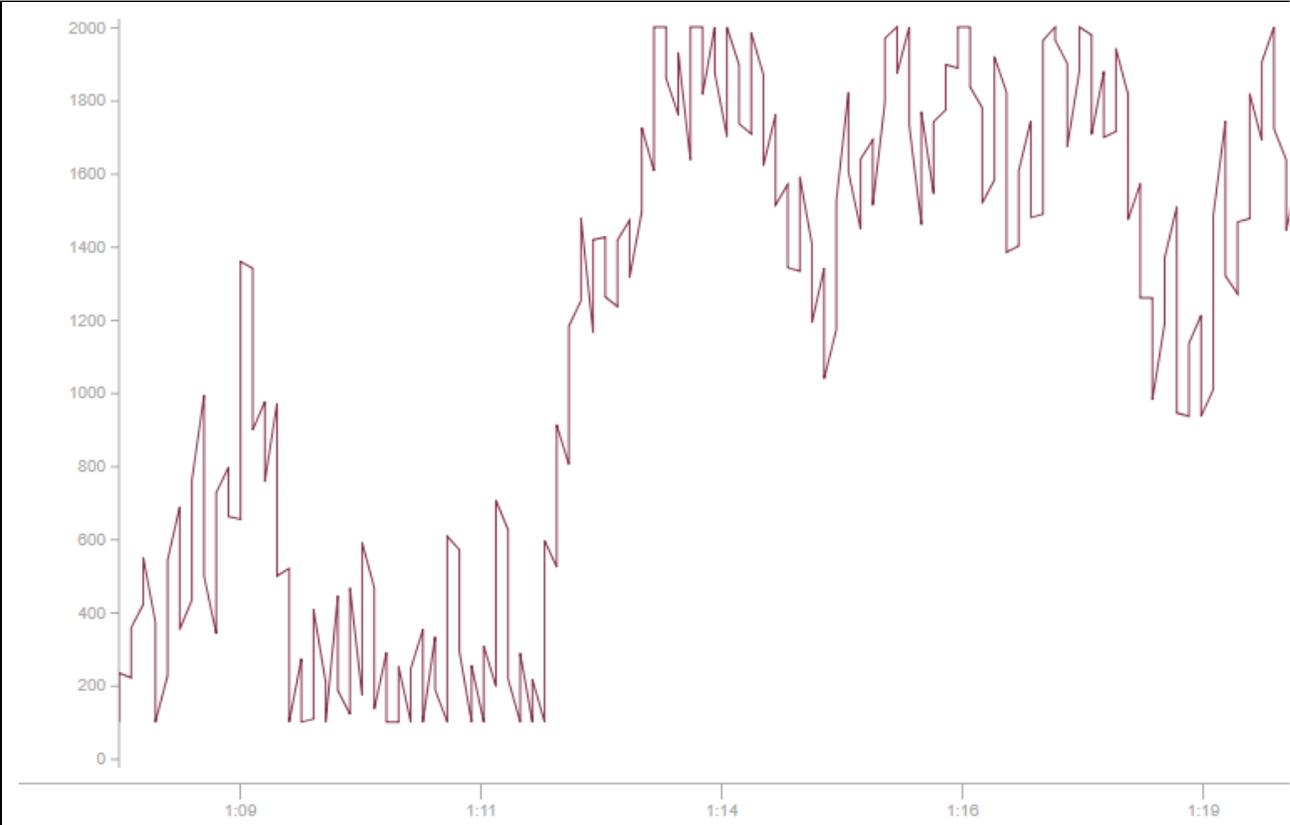
- Set the Time Range to **Historical**.
- The Start Date needs to have a property binding configured pointing to the Chart Range Selector's **props.selectedRange.start** property as in the image below.



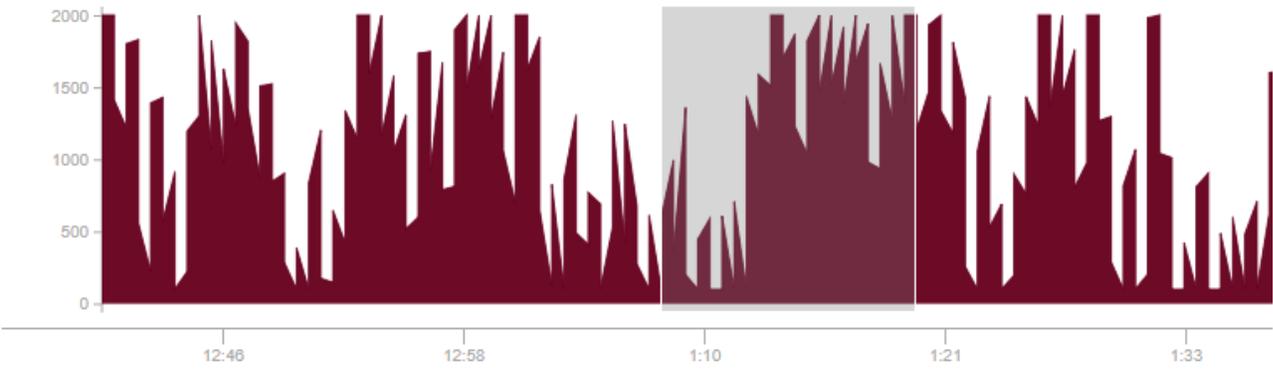
- Similarly, the **End Date** needs to have a property binding configured pointing to the Chart Range Selector's **props.selectedRange.end**.
- For the **Tag History** Binding configuration, click on the **Tag**  icon on the right of the Select Tags table and drill down to the Tag from Step 1, then click **OK**.



17. After clicking OK and accepting the binding configurations on the Time Series Chart, you will be able to use your Chart Range Selector to select what data you want on your Time Series Chart to display. Simply drag and re-size the Chart Range Selector's brush section as shown below.

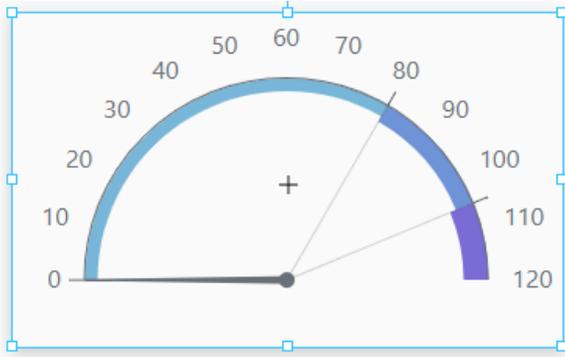


6-10-2020 1:07:55 to 1:20:20



6-10-2020 1:07:55 to 1:20:27

Perspective - Gauge



Component Palette Icon:



On this page ...

- [Properties](#)
- [Component Events](#)
- [Examples](#)
 - [Example 1](#)
 - [Example 2 - 3/4 Circle Variant](#)
 - [Example 3 - Full Axis Variant](#)

The Gauge component in Perspective provides a way to show realtime values in a range as they change. The gauge can have one or two axis. It is fully customizable in its appearance, from colors, line widths, needle length, radius, and more.

The following feature is new in Ignition version **8.1.2**
[Click here](#) to check out the other new features

The Gauge component has three pre-configured [variants](#):

- Half Circle - Default layout with a half-circle gauge.
- 3/4 Circle - Layout with a 3/4 circle gauge.
- Full Axis - Layout with a full axis gauge.

For an example of each variant, see the examples section below.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
value	Numeric value for the gauge to display. Default is 0.	value: numeric
secondaryValue	Optional secondary value for the gauge to display on a second axis. Default is 0.	value: numeric
startAngle	Radial position for the start of the gauge's axis. Default is 180.	value: numeric
endAngle	Radial position for the end of the gauge's axis. Default is 360.	value: numeric
outerAxis	Sets the values for an outer axis on the gauge.	object

Name	Description	Property Type															
data	What value this axis and its needle should display. Can be set to the value or secondary value of the gauge.	value: string dropdown															
show	Whether the outer axis is displayed. Default is true (show).	value: boolean															
minValue	Minimum gauge value for this axis. Default is zero (0).	value: numeric															
maxValue	Maximum gauge value for this axis. Default is 120.	value: numeric															
width	Width of the line (in pixels) that represents the outer axis.	value: numeric															
color	Color of the arc line that represents the outer axis. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color															
percentRadius	Radius of the outer axis, as a percentage of the total chart radius.	value: numeric															
ranges	Zones defined on the gauge arc line with a unique color. Array values always start at 0 and increment. Each array item has the following properties: <table border="1" data-bbox="358 779 1195 1152"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>start</td> <td>Value at which this range starts.</td> <td>value: numeric</td> </tr> <tr> <td>end</td> <td>Value at which this range ends.</td> <td>value: numeric</td> </tr> <tr> <td>width</td> <td>Width of this axis, in pixels.</td> <td>value: numeric</td> </tr> <tr> <td>color</td> <td>Color to apply to this range of the dial. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value.</td> <td>color</td> </tr> </tbody> </table>	Name	Description	Property Type	start	Value at which this range starts.	value: numeric	end	Value at which this range ends.	value: numeric	width	Width of this axis, in pixels.	value: numeric	color	Color to apply to this range of the dial. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value.	color	array
Name	Description	Property Type															
start	Value at which this range starts.	value: numeric															
end	Value at which this range ends.	value: numeric															
width	Width of this axis, in pixels.	value: numeric															
color	Color to apply to this range of the dial. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value.	color															
needle	Settings for the needle on the gauge. Options as follows: <table border="1" data-bbox="358 1220 1195 1593"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>origin</td> <td>Distance from the gauge's center or at which the needle originates, as a percentage of the radius. For example, a value of 0 indicates the needle starts at the center point of the gauge. A value of 50 indicates it starts 50% from the center point.</td> <td>value: numeric</td> </tr> <tr> <td>reach</td> <td>How far the needle reaches from the center of the gauge towards the outer dial, as a percentage of the radius. For example, a value of 100 indicates the needle will reach all the way to the outer axis.</td> <td>value: numeric</td> </tr> <tr> <td>color</td> <td>Color of the gauge's needle. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>color</td> </tr> </tbody> </table>	Name	Description	Property Type	origin	Distance from the gauge's center or at which the needle originates, as a percentage of the radius. For example, a value of 0 indicates the needle starts at the center point of the gauge. A value of 50 indicates it starts 50% from the center point.	value: numeric	reach	How far the needle reaches from the center of the gauge towards the outer dial, as a percentage of the radius. For example, a value of 100 indicates the needle will reach all the way to the outer axis.	value: numeric	color	Color of the gauge's needle. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color	object			
Name	Description	Property Type															
origin	Distance from the gauge's center or at which the needle originates, as a percentage of the radius. For example, a value of 0 indicates the needle starts at the center point of the gauge. A value of 50 indicates it starts 50% from the center point.	value: numeric															
reach	How far the needle reaches from the center of the gauge towards the outer dial, as a percentage of the radius. For example, a value of 100 indicates the needle will reach all the way to the outer axis.	value: numeric															
color	Color of the gauge's needle. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color															
tickMarks	Settings for the display of the tick marks on the outer access. Options as follows: <table border="1" data-bbox="358 1656 1195 1959"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color of the tick marks. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>color</td> </tr> <tr> <td>thickness</td> <td>Thickness of the tick marks, in pixels. Default is 1.</td> <td>value: numeric</td> </tr> <tr> <td>length</td> <td>Length of the tick marks, in pixels. Default is 10.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color of the tick marks. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color	thickness	Thickness of the tick marks, in pixels. Default is 1.	value: numeric	length	Length of the tick marks, in pixels. Default is 10.	value: numeric	object			
Name	Description	Property Type															
color	Color of the tick marks. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color															
thickness	Thickness of the tick marks, in pixels. Default is 1.	value: numeric															
length	Length of the tick marks, in pixels. Default is 10.	value: numeric															

innerAxis Sets the values for an inner axis on the gauge.

object

Name	Description	Property Type															
data	What value this axis and its needle should display. Can be set to the value or secondary value of the gauge.	value: numeric															
show	Whether the inner axis is displayed on the gauge. Default is false (don't show).	value: boolean															
minValue	Minimum gauge value for this axis. Default is zero (0).	value: numeric															
maxValue	Maximum gauge value for this axis. Default is 80.	value: numeric															
width	Width of the line (in pixels) that represents the inner axis.	value: numeric															
color	Color of the arc line that represents the inner axis. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color															
percent Radius	Radius of the inner axis, as a percentage of the total chart radius.	value: numeric															
ranges	Number of zones defined on the gauge arc line with a unique color.	array															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>start</td> <td>Value at which this range starts.</td> <td>value: numeric</td> </tr> <tr> <td>end</td> <td>Value at which this range ends.</td> <td>value: numeric</td> </tr> <tr> <td>width</td> <td>Width of this axis, in pixels.</td> <td>value: numeric</td> </tr> <tr> <td>color</td> <td>Color to apply to this range of the dial. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>color</td> </tr> </tbody> </table>	Name	Description	Property Type	start	Value at which this range starts.	value: numeric	end	Value at which this range ends.	value: numeric	width	Width of this axis, in pixels.	value: numeric	color	Color to apply to this range of the dial. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color	
Name	Description	Property Type															
start	Value at which this range starts.	value: numeric															
end	Value at which this range ends.	value: numeric															
width	Width of this axis, in pixels.	value: numeric															
color	Color to apply to this range of the dial. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color															
needle	Settings for the display of the needle on the inner access. Options as follows:	object															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>origin</td> <td>Distance from the gauge's center at which the needle originates, as a percentage of the radius. For example, a value of 0 indicates the needle starts at the center point of the gauge. For example, a value of 50 indicates it starts 50% from the center point.</td> <td>value: numeric</td> </tr> <tr> <td>reach</td> <td>Length of the needle in percentage. For example, a value of 100 indicates the needle will reach all the way to the inner axis.</td> <td>value: numeric</td> </tr> <tr> <td>color</td> <td>Color of the gauge's needle. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>color</td> </tr> </tbody> </table>	Name	Description	Property Type	origin	Distance from the gauge's center at which the needle originates, as a percentage of the radius. For example, a value of 0 indicates the needle starts at the center point of the gauge. For example, a value of 50 indicates it starts 50% from the center point.	value: numeric	reach	Length of the needle in percentage. For example, a value of 100 indicates the needle will reach all the way to the inner axis.	value: numeric	color	Color of the gauge's needle. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color				
Name	Description	Property Type															
origin	Distance from the gauge's center at which the needle originates, as a percentage of the radius. For example, a value of 0 indicates the needle starts at the center point of the gauge. For example, a value of 50 indicates it starts 50% from the center point.	value: numeric															
reach	Length of the needle in percentage. For example, a value of 100 indicates the needle will reach all the way to the inner axis.	value: numeric															
color	Color of the gauge's needle. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color															
tickMarks	Settings for the display of the tick marks on the inner access. Options as follows:	object															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color of the tick marks. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>color</td> </tr> <tr> <td>thickness</td> <td>Thickness of the tick marks, in pixels. Default is 1.</td> <td>value: numeric</td> </tr> <tr> <td>length</td> <td>Length of the tick marks, in pixels. Default is 10.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color of the tick marks. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color	thickness	Thickness of the tick marks, in pixels. Default is 1.	value: numeric	length	Length of the tick marks, in pixels. Default is 10.	value: numeric				
Name	Description	Property Type															
color	Color of the tick marks. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color															
thickness	Thickness of the tick marks, in pixels. Default is 1.	value: numeric															
length	Length of the tick marks, in pixels. Default is 10.	value: numeric															

backgroundColor	Color applied as a background within the gauge. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color
animate	Whether needle should be animated in a sweeping motion when value changes. Default is false.	value: boolean
reverseScale	If true, the gauge will reverse the direction from min Value to max Value on its dial.	value: boolean
style	Sets a style for this gauge. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object

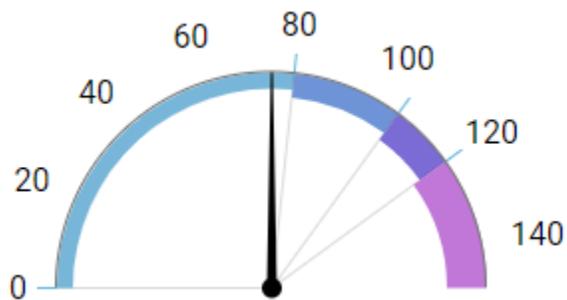
Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Examples

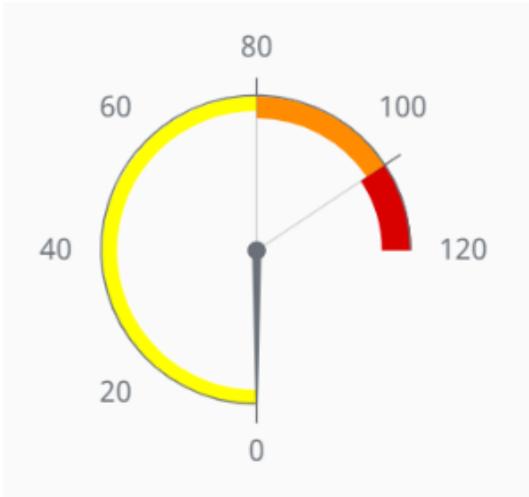
Example 1



In this example we added a fourth axis (`outerAxis.ranges.3`) and configured it.

Property	Value
Value	75
<code>props.outerAxis.maxValue</code>	150
<code>props.outerAxis.ranges.3.start</code>	120
<code>props.outerAxis.ranges.3.end</code>	150
<code>props.outerAxis.ranges.3.width</code>	20
<code>props.outerAxis.ranges.3.color</code>	#C077D8

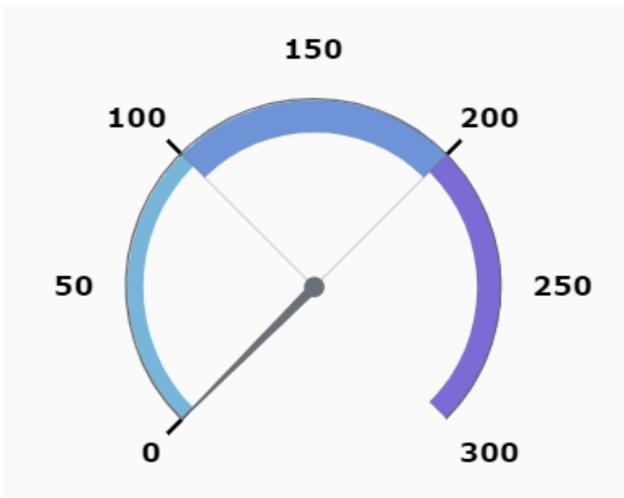
Example 2 - 3/4 Circle Variant



Start with the Gauge 3/4 Circle variant. Set the properties as follows:

Property	Value
props.startAngle	90
props.endAngle	360
props.outerAxis.ranges.0.color	#FFFF00
props.outerAxis.ranges.1.color	#FF8C00
props.outerAxis.ranges.2.color	#D90000

Example 3 - Full Axis Variant

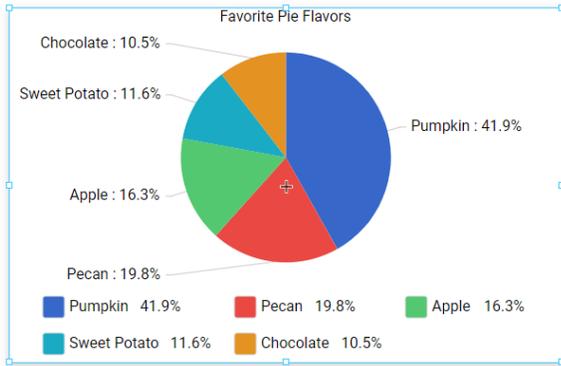


Start with the Gauge Full Axis variant. Set the properties as follows:

Property	Value
props.startAngle	135
props.endAngle	405
props.outerAxis.maxValue	300
props.outerAxis.ranges.0.end	100
props.outerAxis.ranges.1.start	100
props.outerAxis.ranges.1.end	200

props.outerAxis.ranges.2.start	200
props.outerAxis.ranges.2.end	300
props.outerAxis.tickMarks.color	#000000
props.outerAxis.tickMarks.thickness	2
props.style.color	#000000
props.style.fontFamily	Verdana
props.style.fontSize	12
props.style.fontWeight	bold

Perspective - Pie Chart



On this page ...

- [Properties](#)
 - [Formatting Options](#)
- [Component Events](#)
- [Example](#)

Component Palette Icon:



A Pie Chart displays a list of named items, each of which has a value that is part of a total. The total is the sum of the value of each item. The key to the Pie Chart component is the data property, which contains the items that will be displayed as pie wedges. It is fully customizable in its appearance, from colors, line widths, text styles, and more.

The following feature is new in Ignition version **8.1.2**
[Click here](#) to check out the other new features

The Pie Chart component had two pre-configured [variants](#):

- Flat Chart - Default component described above.
- Three-Dimensional Chart - Component pre-set with three-dimensional pie wedges.

Properties

Name	Description	Property Type									
data	Data source for the chart. Each object within an array defines the name and value for a single pie section.	array									
colors	Colors that correspond to each pie section, respective of order in data. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	array									
title	Name to display for this chart.	value: string									
tileColor	Color of the title. Can be chosen from color wheel, chosen from color palette, or entered as <u>RGB</u> or <u>HSL</u> value. See Color Selector .	color									
valueFormat	Label and legend value format configuration. <table border="1" data-bbox="245 1486 1117 1625"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>showPercentSymbol</td> <td>Whether to show the percent symbol next to the percent value.</td> <td>value: boolean</td> </tr> <tr> <td>showValueAsPercent</td> <td>Whether to show the value as percent.</td> <td>value: boolean</td> </tr> </tbody> </table>	Name	Description	Property Type	showPercentSymbol	Whether to show the percent symbol next to the percent value.	value: boolean	showValueAsPercent	Whether to show the value as percent.	value: boolean	array
Name	Description	Property Type									
showPercentSymbol	Whether to show the percent symbol next to the percent value.	value: boolean									
showValueAsPercent	Whether to show the value as percent.	value: boolean									
showLabels	Whether to show labels for each section of this chart. Default is true (show).	value: boolean									

labels	Settings for the labels.		object														
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> </table>	Name	Description	Property Type													
Name	Description	Property Type															
showName	Whether to show the name on the label.	value: boolean															
showValue	Whether to show the value on the label. Hiding values will disable any value formats set.	value: boolean															
bent	Bend labels around chart slices. Default is false.	value: boolean															
align	Whether the labels should be aligned in vertical columns.	value: boolean															
inside	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.2 Click here to check out the other new features</p> </div> <p>Settings for showing labels inside of the chart slices instead of outside.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Value that determines whether to show the labels inside of the chart slices instead of outside, based on if the value percentage is below the percentLimit threshold.</td> <td>value: boolean</td> </tr> <tr> <td>radius</td> <td>Distance in percentage towards center of Pie Chart while inside is enabled. 0 represents outside edge while 100 would be directly in the middle.</td> <td>value: numeric</td> </tr> <tr> <td>color</td> <td>Label color for labels while they are displayed inside the chart.</td> <td>value: color</td> </tr> <tr> <td>percentLimit</td> <td>Value that determines at what value percentage to place label on outside of chart instead of inside.</td> <td>value: numeric</td> </tr> </tbody> </table>		Name	Description	Property Type	enabled	Value that determines whether to show the labels inside of the chart slices instead of outside, based on if the value percentage is below the percentLimit threshold.	value: boolean	radius	Distance in percentage towards center of Pie Chart while inside is enabled. 0 represents outside edge while 100 would be directly in the middle.	value: numeric	color	Label color for labels while they are displayed inside the chart.	value: color	percentLimit	Value that determines at what value percentage to place label on outside of chart instead of inside.	value: numeric
Name	Description	Property Type															
enabled	Value that determines whether to show the labels inside of the chart slices instead of outside, based on if the value percentage is below the percentLimit threshold.	value: boolean															
radius	Distance in percentage towards center of Pie Chart while inside is enabled. 0 represents outside edge while 100 would be directly in the middle.	value: numeric															
color	Label color for labels while they are displayed inside the chart.	value: color															
percentLimit	Value that determines at what value percentage to place label on outside of chart instead of inside.	value: numeric															
wrap	<p>Label text wrapping configuration. Ability to wrap long labels.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enables label text wrapping. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>maxWidth</td> <td>The maximum allowable label width, in pixels.</td> <td>value: numeric</td> </tr> </tbody> </table>		Name	Description	Property Type	enabled	Enables label text wrapping. Default is false.	value: boolean	maxWidth	The maximum allowable label width, in pixels.	value: numeric						
Name	Description	Property Type															
enabled	Enables label text wrapping. Default is false.	value: boolean															
maxWidth	The maximum allowable label width, in pixels.	value: numeric															
color	Color of the labels. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color															

tooltipFormat

The following feature is new in Ignition version 8.1.5
[Click here](#) to check out the other new features

value: string

Value that determines the format of the tooltips.

Placeholder Objects

The value of this property expects a string, and recognizes certain placeholder objects. The objects are fully detailed in [amChart's documentation](#), and we listed some commonly used objects below. Note that each object should be wrapped in a set of braces ("{" and "}") characters).

Object	Description	Example
Category	References the value of the wedge's category (the string value for the wedge), allowing the tooltip to display a string value for the wedge. The value of category is based off the string value for the wedge. Thus, in the Pie Chart Example on this page, category would return the value of the "flavor" property.	{category}
Value	Returns the value of the wedge. In addition to just the value of the wedge, additional modifiers can be added to obtain pre-calculated derivative values, such as the sum or largest value. A full list of modifiers can be found in the amCharts documentation .	{value} {value.sum} {value.high}

Formatting Options

In addition to placeholder objects, the tooltip can also be formatted in the following ways. All formatting is placed between a set of brackets ("[" and "]" characters) as a single **block**. One or more formatting options can be placed into a single block. When specifying multiple options, each should be delimited by a single space. Example: [**bold red**]

Option	Description	Example Format	Example Output
Bold	Applies bold formatting to the characters in the block.	Hello [bold] World	Hello World
Colors	Allows you to specify the color of the text. Colors can be specified as RGB color codes, or named colors.	Hello [red] World OR Hello [ff0000] World	Hello World
In-line CSS	Allows you to directly apply CSS attributes.	Hello [font-style: italic] World	Hello <i>World</i>
Closing Bracket [/]	Terminates the formatting string. Note that this is optional. Style formatting will automatically be terminated by either the next opening style block ("[" character) or the end of the string, whichever comes first.	[bold]Hello [/] World!	Hello World!
Formatting Functions	In addition to the above, several functions are available to further format the placeholder objects. A full list of functions can be found in the amChart documentation .	{value. formatNumber ('###.00')}	100.00

showLegend

Whether to show a legend for this chart. Default is true (show).

value: boolean

legend	<p>The following feature is new in Ignition version 8.1.2 Click here to check out the other new features</p> <p>Settings for legend.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>fontSize</td> <td>Font size for legend labels.</td> <td>value: numeric</td> </tr> <tr> <td rowspan="4">icon</td> <td colspan="2">Settings for the icon on entries in the legend.</td> </tr> <tr> <td>Name</td> <td>Description</td> </tr> <tr> <td>height</td> <td>Height value of legend icon.</td> </tr> <tr> <td>width</td> <td>Width value of legend icon.</td> </tr> <tr> <td>enabled</td> <td>Value that determines whether to show the legend icons or hide them.</td> <td>value: boolean</td> </tr> <tr> <td>position</td> <td>Aligns legend to specified direction.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	fontSize	Font size for legend labels.	value: numeric	icon	Settings for the icon on entries in the legend.		Name	Description	height	Height value of legend icon.	width	Width value of legend icon.	enabled	Value that determines whether to show the legend icons or hide them.	value: boolean	position	Aligns legend to specified direction.	value: string	object
Name	Description	Property Type																					
fontSize	Font size for legend labels.	value: numeric																					
icon	Settings for the icon on entries in the legend.																						
	Name	Description																					
	height	Height value of legend icon.																					
	width	Width value of legend icon.																					
enabled	Value that determines whether to show the legend icons or hide them.	value: boolean																					
position	Aligns legend to specified direction.	value: string																					
legendLabelColor	Color of the legend labels.	color																					
cutoutRadius	Percent of total radius to cut out of center of chart. If greater than zero, the chart becomes ring-style instead of pie.	value: numeric																					
selection	<p>The following feature is new in Ignition version 8.1.10 Click here to check out the other new features</p> <p>An object that contains selection related properties.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enables selection of pie chart slices.</td> <td>value: boolean</td> </tr> <tr> <td>data</td> <td>A read-only list of selected pie chart slices.</td> <td>array</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Enables selection of pie chart slices.	value: boolean	data	A read-only list of selected pie chart slices.	array	object												
Name	Description	Property Type																					
enabled	Enables selection of pie chart slices.	value: boolean																					
data	A read-only list of selected pie chart slices.	array																					
selectOutline	<p>Outline for each section of the pie. Options are as follows:</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>width</td> <td>Width of the border (in pixels) around the pie section.</td> <td>value: numeric</td> </tr> <tr> <td>color</td> <td>Color of border around each pie section. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of border around each pie section. 0 is fully transparent, 1 is fully opaque.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	width	Width of the border (in pixels) around the pie section.	value: numeric	color	Color of border around each pie section. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color	opacity	Opacity of border around each pie section. 0 is fully transparent, 1 is fully opaque.	value: numeric	object									
Name	Description	Property Type																					
width	Width of the border (in pixels) around the pie section.	value: numeric																					
color	Color of border around each pie section. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color																					
opacity	Opacity of border around each pie section. 0 is fully transparent, 1 is fully opaque.	value: numeric																					
enableTransitions	Whether the chart has visual transition effects for changes in chart data.	value: boolean																					
threeDimensional	Whether the chart has depth effect to look three-dimensional.	value: boolean																					
style	Use styles to customize the visual style of the component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																					

Component Events

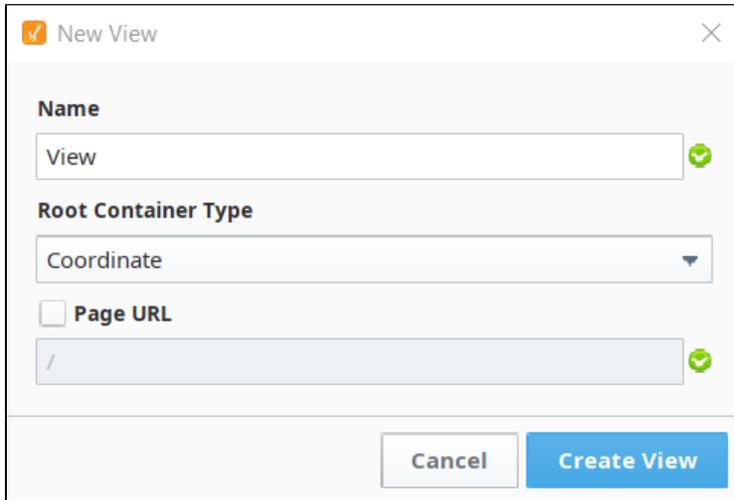
Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Example

The Pie Chart component can be used for things like inventory tracking. Below is an example that uses a Pie Chart to display the inventory of a local ice cream shop.

1. From the Perspective section of the Project Browser in your Designer, right-click on the Views folder and select **New View...** to create a new view.
2. This will bring up the New View window. Give your view a name and select the Coordinate Root Container type. The Page URL setting will remain unchecked for this example.



The screenshot shows a 'New View' dialog box with the following fields and options:

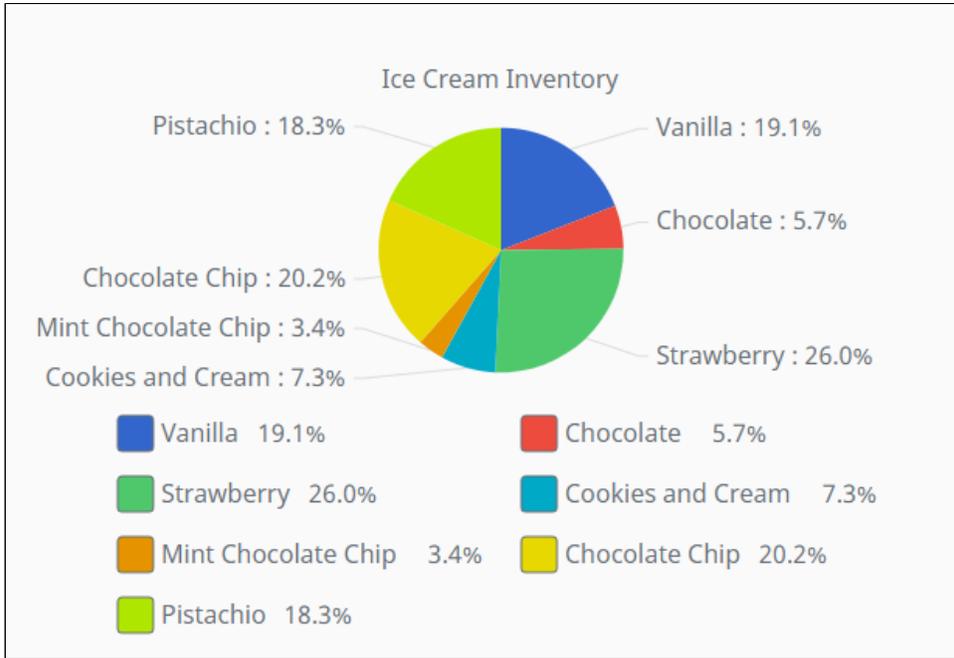
- Name:** A text input field containing 'View' with a green checkmark icon to its right.
- Root Container Type:** A dropdown menu with 'Coordinate' selected.
- Page URL:** An unchecked checkbox followed by a text input field containing a forward slash (/).
- Buttons:** 'Cancel' and 'Create View' buttons at the bottom.

3. Drag and drop a Pie Chart from the Perspective Component Palette onto your newly created view.
4. Set the Pie Chart's title property to **Ice Cream Inventory**.
5. Copy the array below and paste it on the Pie Chart's data property.

```
[
  {
    "flavor": "Vanilla",
    "count": "50"
  },
  {
    "flavor": "Chocolate",
    "count": "15"
  },
  {
    "flavor": "Strawberry",
    "count": "68"
  },
  {
    "flavor": "Cookies and Cream",
    "count": "19"
  },
  {
    "flavor": "Mint Chocolate Chip",
    "count": "9"
  },
  {
    "flavor": "Chocolate Chip",
    "count": "53"
  },
  {
    "flavor": "Pistachio",
    "count": "48"
  }
]
```

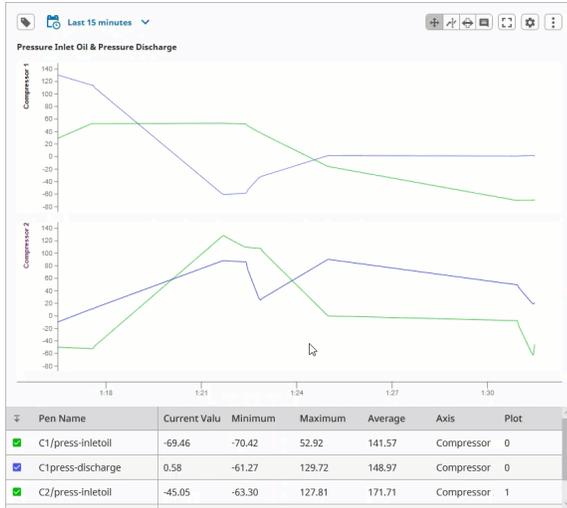
The array above is an array of objects. Each object is a dictionary containing key/value pairs where the keys of each dictionary represent ice cream flavors and ice cream inventory counts with their respective values for each. Value types can vary as you can pass both "50" and 50 as counts and the Pie Chart will still be able to render the data correctly. The Pie Chart can have various forms of data sources. The data source array can be built via scripting following the above format or it can be built using a [query binding](#) on the Pie Chart's data property. The query used must return two columns in any order where each column represents a string and a numeric value to be rendered by the chart.

6. Once you have done this, your Pie Chart should accurately represent of the inventory data for the ice cream shop.



Perspective - Power Chart

The following feature is new in Ignition version **8.1.0**
[Click here](#) to check out the other new features



On this page ...

- [Component Anatomy](#)
- [User Interaction](#)
- [Properties](#)
- [Component Events](#)

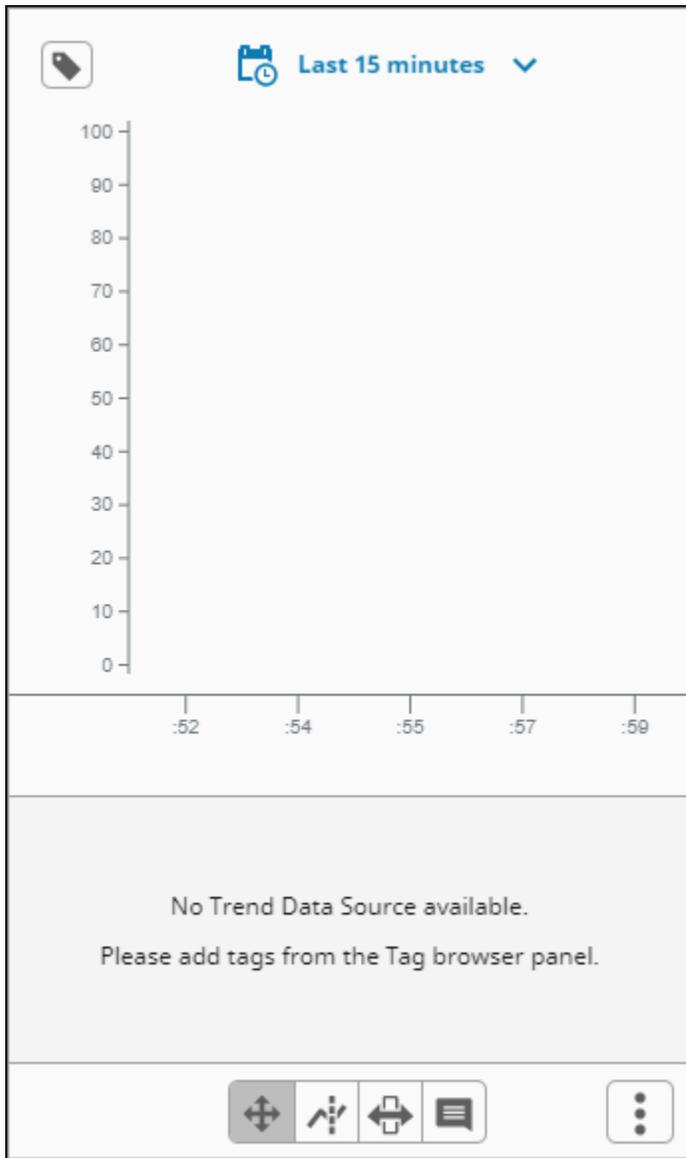
Component Palette Icon:



The Power Chart collects and displays data based on the pens that have been configured on the chart. Users can add or remove pens from the chart, which in turn changes the underlying data. It is fully customizable in its appearance, from labels, colors, line widths, legend, scroll bars, and text styles.

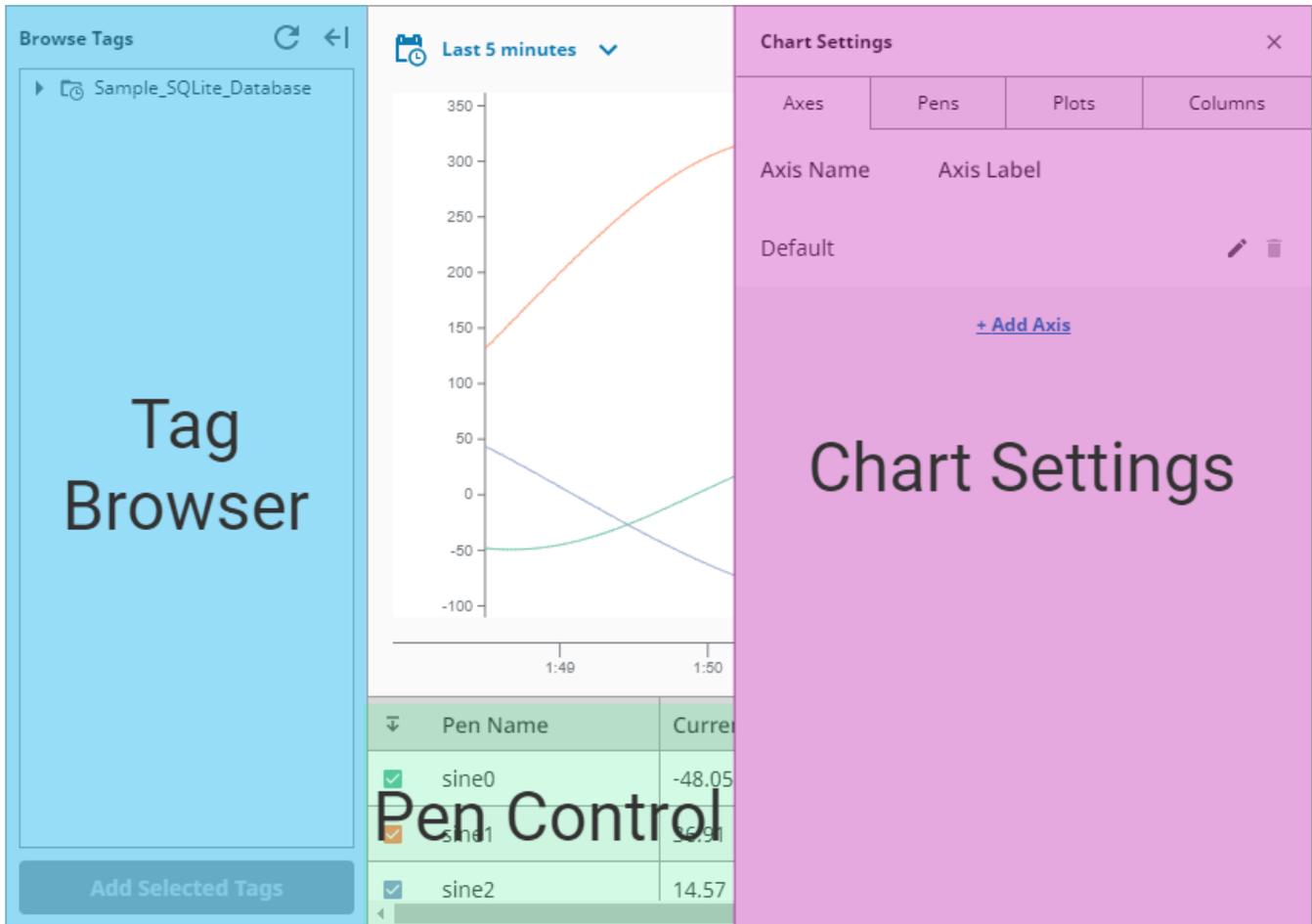
Note: The Power Chart utilizes functionality provided by the [Tag Historian](#) module, and requires a Tag Historian license to function.

The Power Chart has a responsive design and a mobile-optimized display that is different than the standard display. It has a mobile breakpoint so it fits better on mobile devices. The mobile breakpoint is 750px, and is configurable on the 'config' property, `responsiveDesignWidth`, which is described in the Property Table on this page. The chart can change how it's rendered when viewed on smaller devices. The image below shows how the Power Chart renders on a smaller device.



Component Anatomy

Aside from the trending area, the component features several additional areas that provide additional functionality. The diagram below identifies these areas on a standard device.



Browse Tags

The Browse Tags panel allows you to browse for any available historical data, and add it to the chart. It is similar to the Designer's Tag Browser, but this tree reports any records that are accessible from the Tag Historian system, including tables provided by the [DB Table Historian Provider](#). There are two ways to add Tags to the chart's display:

- Select any nodes (entries with the Tag icon), and click **Add Selected Tags**, which will add a pen to the chart that represents the node that was selected.
- Drag selected nodes onto the chart. You can select multiple items by using Ctrl-click. You'll see a prompt indicating how many Tags are selected, i.e., 4 Tags.)

If there's more than one plot in the chart configuration, you will be prompted to choose which plot to add the pen to.

Pen Control

This table shows each pen that's currently on the component, offers some aggregates based on the chart's current range, and provides some quick actions such as hiding the pen and changing its color.

The following feature is new in Ignition version **8.1.13**
[Click here](#) to check out the other new features

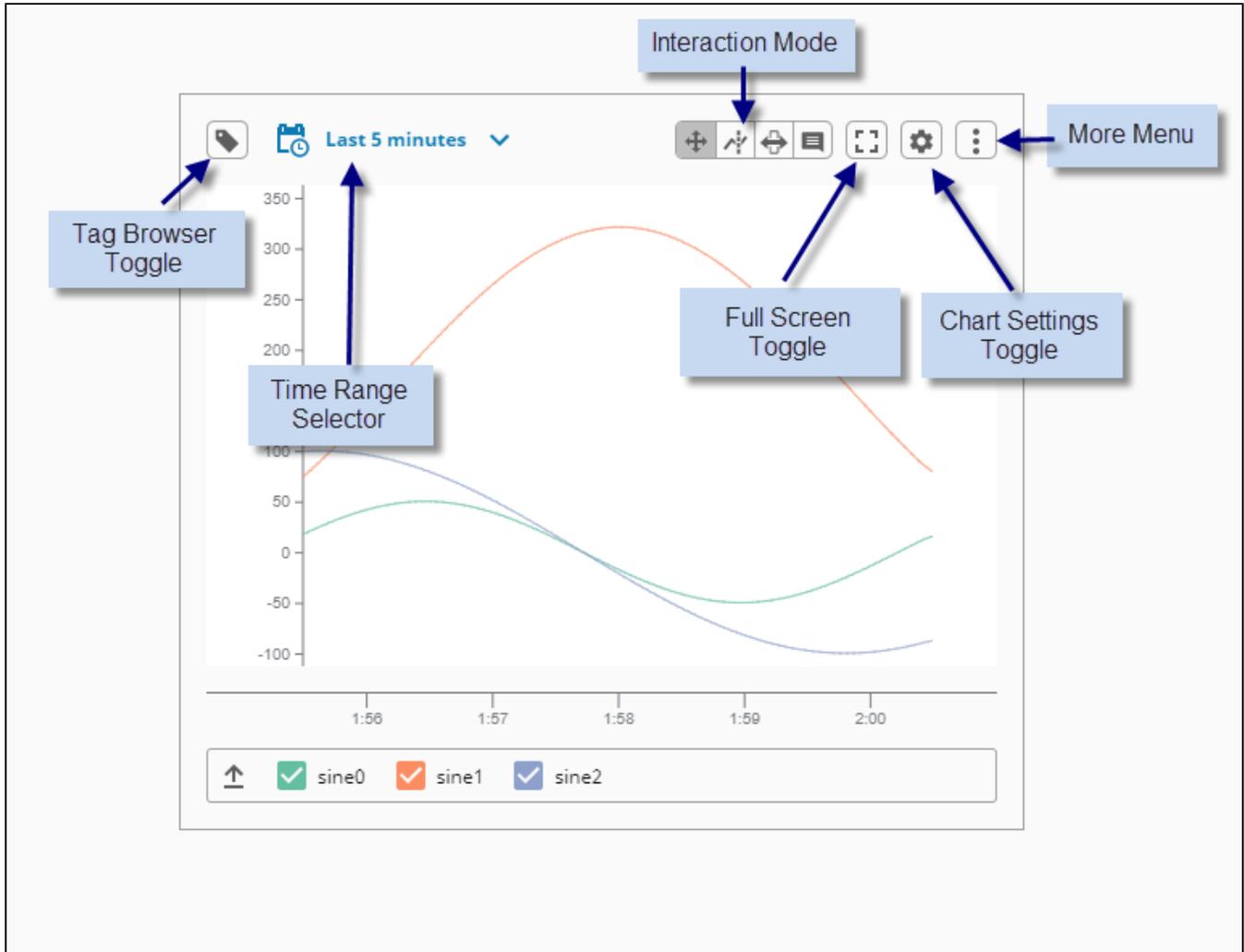
The Pen Data display now supports numeric locale formatting. Numeric values are automatically formatted based on the session locale and the [axes.dataFormat](#) property.

↕	Pen Name	Current Value	Minimum	Maximum	Average	Axis	Plot	X Trace
☑	sensor 2/reading	56,34	-39,32	158,85	53,89	Default	0	98,09
☑	sensor 3/reading	47,71	-60,08	152,45	47,37	Default	0	48,59
☑	ambientum	102,65	-8,79	200,53	99,81	Default	0	102,69

Chart Settings

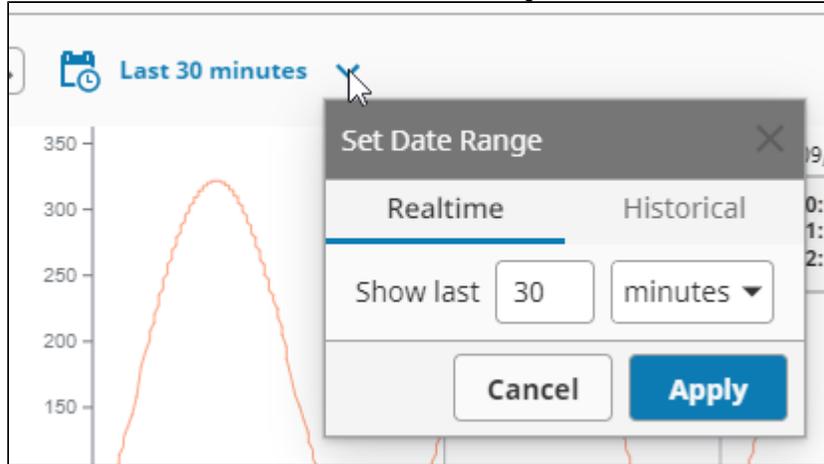
This panel allows users to add new objects to the chart, such as new axes and plots.

User Interaction

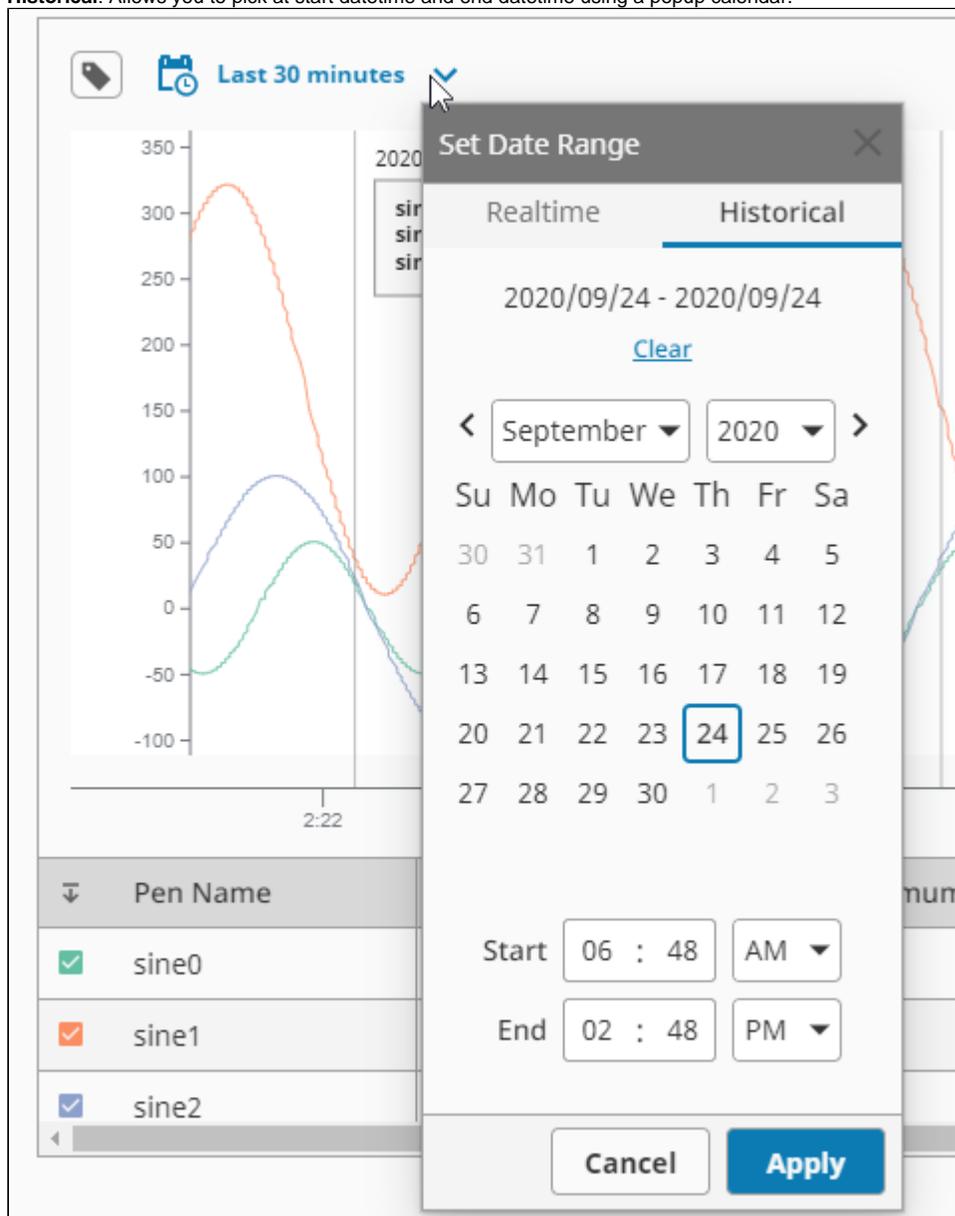


Interaction		Description
Browse Tags		Toggles the Tag Browser panel.
Date Range Selector		Allows you to set the range on the chart. There are two modes:

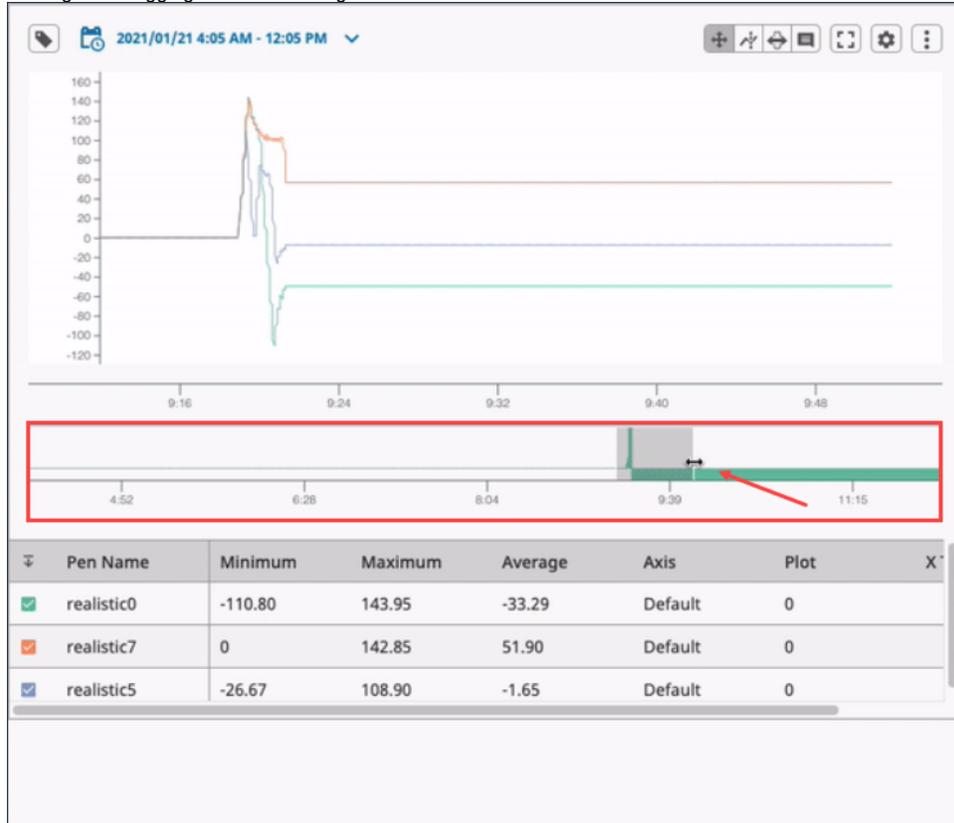
- **Realtime:** Shows the most recent chart data based on the given timeframe. Useful in cases where you want to display recor



- **Historical:** Allows you to pick at start datetime and end datetime using a popup calendar.



When the chart is in Historical mode, an additional time axis appears. On this access you can narrow down what the chart clicking and dragging the mouse along the axis.



Pan and Zoom



In this mode you can drag or swipe to pan forward and backward in time. On desktop device, clicking and dragging will pan across mobile/touchscreen device, tapping and dragging will pan. The "pinch" and "spread" gestures will zoom.

Zoom reset

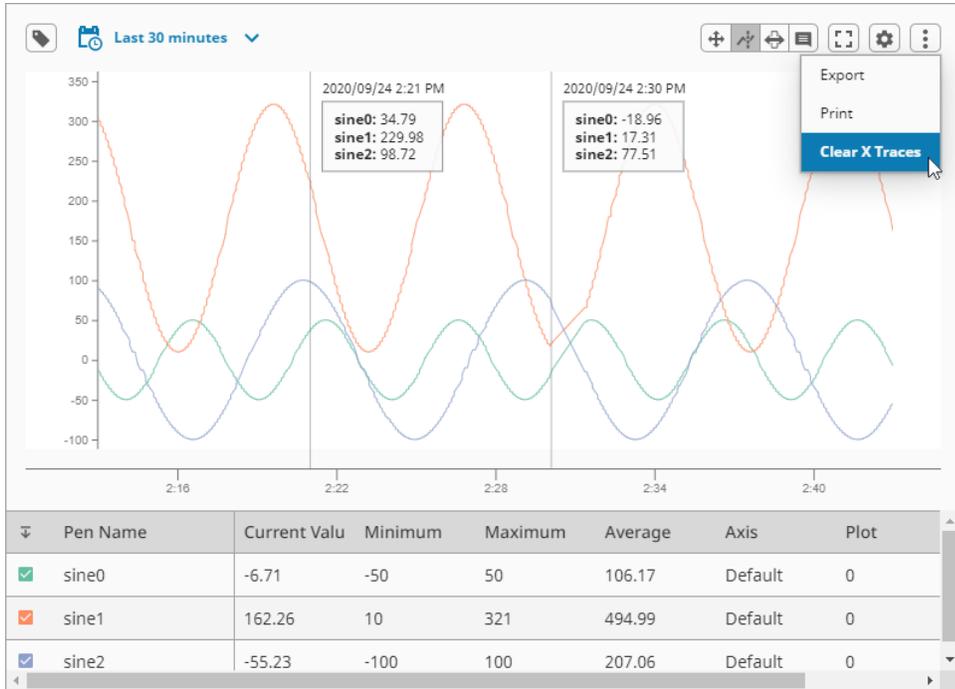


To reset the zoom to default, click the zoom reset icon.

X Trace

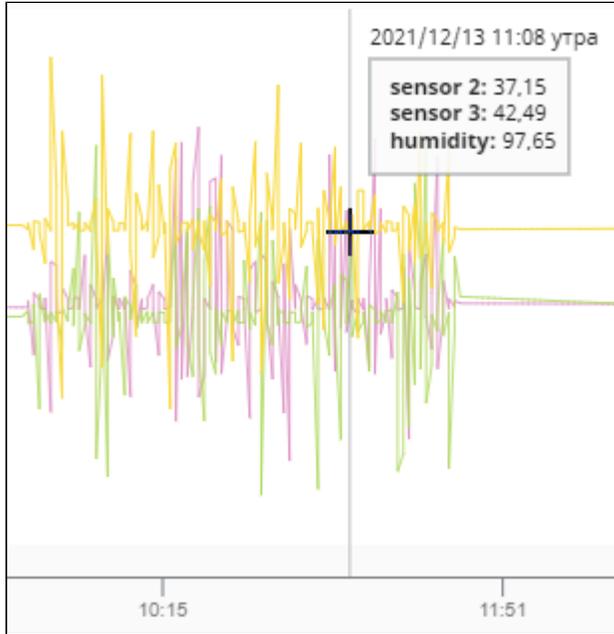


Tap to place a vertical line, which shows an interpolated value for each pen on the plot. To clear the X Trace values, select **Clear**



The following feature is new in Ignition version 8.1.13
[Click here](#) to check out the other new features

The X Trace display now supports numeric locale formatting. Numeric values are automatically formatted based on the session locale.

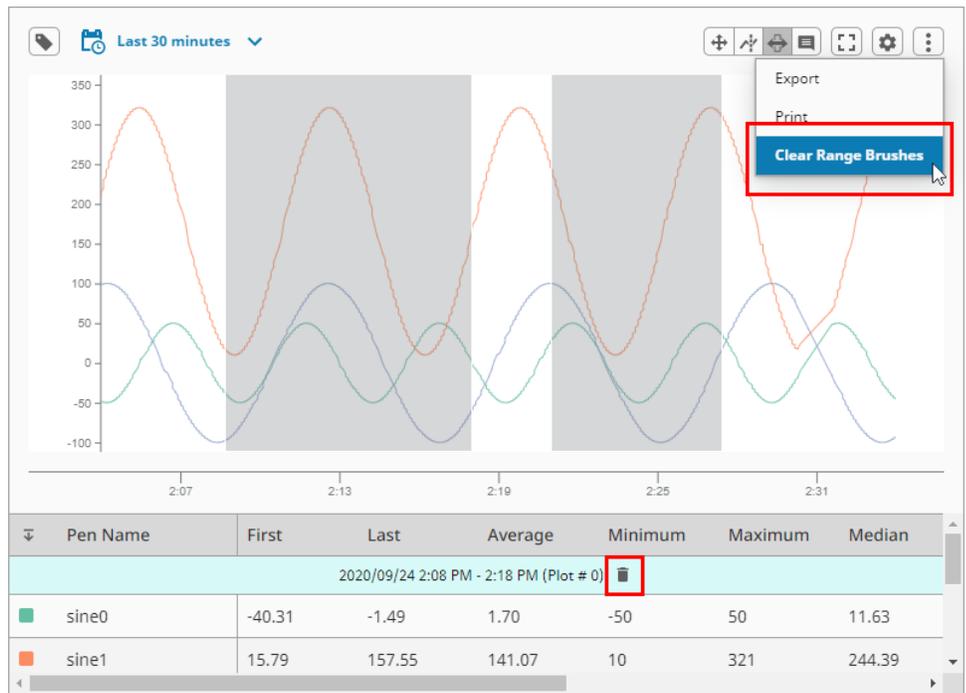


Range Brush



Allows you to select a range of data on the chart. While a brush is active, the Pen Control Table will show aggregate summaries. Multiple brushes will create multiple aggregation summaries.

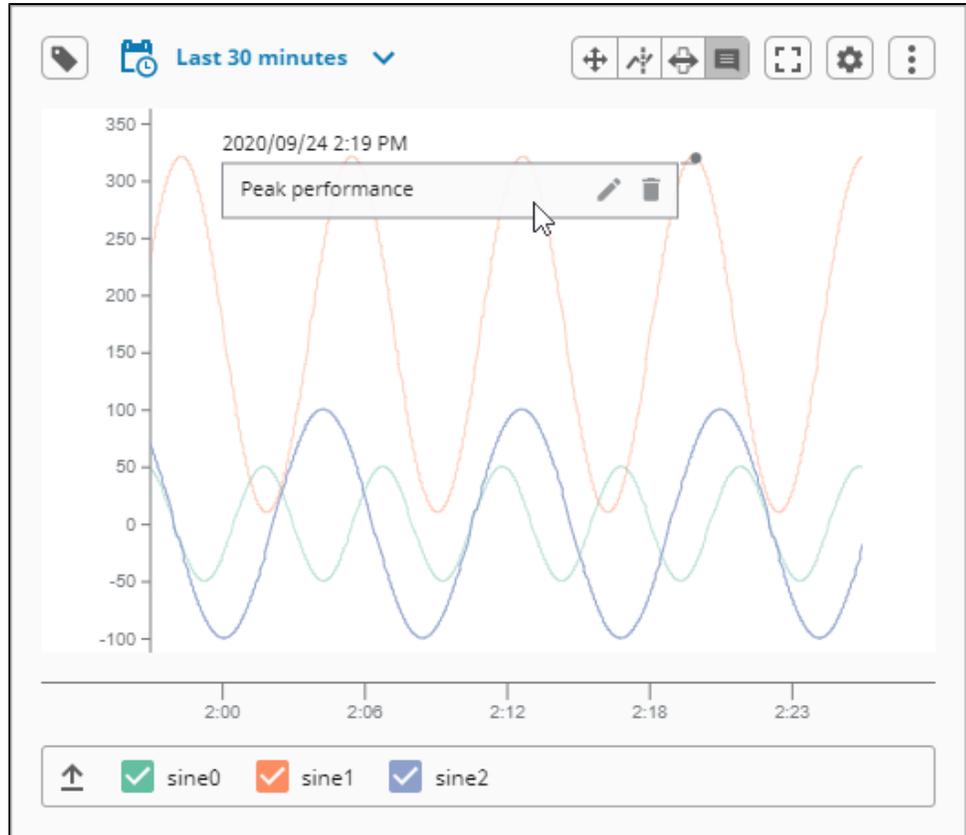
Individual selections can be removed by clicking the trashcan icon in the Pen Control Table, or by selecting **Clear Range Brushes**



Annotate



Click near a trend, line, or data point and you'll have the opportunity to add an annotation. The annotation is stored with the Tag ID. Annotations can be edited by clicking the **Edit** icon and deleted by clicking the **Delete** icon, which appear when hovering over



Full Screen



Puts the chart into full screen mode.

Settings		<p>Opens the Chart settings panel, allowing users to modify various aspects of the chart from the session. There are four tabs: Axes</p> <table border="1" data-bbox="365 184 1500 449"> <thead> <tr> <th>Tab</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Axes</td> <td>You can add or delete Axes here, or click the Edit  icon to edit an existing axis. The options are the same on the</td> </tr> <tr> <td>Pens</td> <td>You can add or delete pens here or click the Edit  icon to edit an existing pen. The options are the same on the A</td> </tr> <tr> <td>Plots</td> <td>You can add or delete plots here or click the Edit  icon to edit an existing plot. The options are the same on the A</td> </tr> <tr> <td>Columns</td> <td>The Columns Chart Settings tab has options for datapoints to display on the Pen Control Panel and datapoints to dis</td> </tr> </tbody> </table>	Tab	Description	Axes	You can add or delete Axes here, or click the Edit  icon to edit an existing axis. The options are the same on the	Pens	You can add or delete pens here or click the Edit  icon to edit an existing pen. The options are the same on the A	Plots	You can add or delete plots here or click the Edit  icon to edit an existing plot. The options are the same on the A	Columns	The Columns Chart Settings tab has options for datapoints to display on the Pen Control Panel and datapoints to dis
Tab	Description											
Axes	You can add or delete Axes here, or click the Edit  icon to edit an existing axis. The options are the same on the											
Pens	You can add or delete pens here or click the Edit  icon to edit an existing pen. The options are the same on the A											
Plots	You can add or delete plots here or click the Edit  icon to edit an existing plot. The options are the same on the A											
Columns	The Columns Chart Settings tab has options for datapoints to display on the Pen Control Panel and datapoints to dis											
More Menu		<p>Provides additional contextual options, depending on the current state of the chart. The button to open the More Menu <i>only</i> appears when the following options are available:</p> <table border="1" data-bbox="365 537 1463 764"> <thead> <tr> <th>Option</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Export</td> <td>Takes the datapoints visible on the various plots, and exports them to a CSV.</td> </tr> <tr> <td>Print</td> <td>Opens print dialog box so user can print the chart.</td> </tr> <tr> <td>Clear X Traces</td> <td>Clears all X Traces on the chart. Only appears when there are X Traces on the chart.</td> </tr> <tr> <td>Clear Range Brushes</td> <td>Clears all range brushes on the chart. Only appears when there are brush selections on the chart.</td> </tr> </tbody> </table>	Option	Description	Export	Takes the datapoints visible on the various plots, and exports them to a CSV.	Print	Opens print dialog box so user can print the chart.	Clear X Traces	Clears all X Traces on the chart. Only appears when there are X Traces on the chart.	Clear Range Brushes	Clears all range brushes on the chart. Only appears when there are brush selections on the chart.
Option	Description											
Export	Takes the datapoints visible on the various plots, and exports them to a CSV.											
Print	Opens print dialog box so user can print the chart.											
Clear X Traces	Clears all X Traces on the chart. Only appears when there are X Traces on the chart.											
Clear Range Brushes	Clears all range brushes on the chart. Only appears when there are brush selections on the chart.											

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description						
config	<p>Configuration for the data feeding the chart.</p> <table border="1" data-bbox="251 1161 1500 1948"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="258 1245 342 1318">penNamePathDepth</td> <td data-bbox="347 1245 1494 1507"> <div data-bbox="358 1272 1269 1356" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.24 Click here to check out the other new features</p> </div> <p>This value will set the depth of the tag path to include in the pen name.</p> <p>Using the Sample Quick Start project as an example, the realistic0 Tag has a parent folder of realistic and a grandparent folder Power Chart will be realistic0 if the penNamePathDepth is set to 1, which is the default value. If the penNamePathDepth is set to would be sample_tags/realistic/realistic1. You can also add new Tags with path depth values that differ from existing Tags if ne</p> </td> </tr> <tr> <td data-bbox="258 1514 342 1587">tagBrowserStartPath</td> <td data-bbox="347 1514 1494 1940"> <p>A path to a nested Tag History provider structure from which browsing will start. The path is expected to contain key-value pairs s</p> <pre data-bbox="358 1570 1487 1598">histprov:Sample_SQLite_Database:/drv:My_Gateway:My_Tag_Provider:/tag:My_Folder/Another_Folder</pre> <ul data-bbox="375 1654 1500 1745" style="list-style-type: none"> • histprov - The name of the Tag Historian Provider • drv - The historian driver, which is typically a combination of a gateway name and tag provider name separated by a colon. E • tag - A path to a node that has children. Typically should lead to either a folder or the root node of a UDT instance. If omitted drv, <p>While providing a path to this property, the tag component can be omitted, which will set the starting path for the Tag Browser par example:</p> <pre data-bbox="358 1850 1149 1877">histprov:Sample_SQLite_Database:/drv:My_Gateway:My_Tag_Provider:/</pre> </td> </tr> </tbody> </table>	Name	Description	penNamePathDepth	<div data-bbox="358 1272 1269 1356" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.24 Click here to check out the other new features</p> </div> <p>This value will set the depth of the tag path to include in the pen name.</p> <p>Using the Sample Quick Start project as an example, the realistic0 Tag has a parent folder of realistic and a grandparent folder Power Chart will be realistic0 if the penNamePathDepth is set to 1, which is the default value. If the penNamePathDepth is set to would be sample_tags/realistic/realistic1. You can also add new Tags with path depth values that differ from existing Tags if ne</p>	tagBrowserStartPath	<p>A path to a nested Tag History provider structure from which browsing will start. The path is expected to contain key-value pairs s</p> <pre data-bbox="358 1570 1487 1598">histprov:Sample_SQLite_Database:/drv:My_Gateway:My_Tag_Provider:/tag:My_Folder/Another_Folder</pre> <ul data-bbox="375 1654 1500 1745" style="list-style-type: none"> • histprov - The name of the Tag Historian Provider • drv - The historian driver, which is typically a combination of a gateway name and tag provider name separated by a colon. E • tag - A path to a node that has children. Typically should lead to either a folder or the root node of a UDT instance. If omitted drv, <p>While providing a path to this property, the tag component can be omitted, which will set the starting path for the Tag Browser par example:</p> <pre data-bbox="358 1850 1149 1877">histprov:Sample_SQLite_Database:/drv:My_Gateway:My_Tag_Provider:/</pre>
Name	Description						
penNamePathDepth	<div data-bbox="358 1272 1269 1356" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.24 Click here to check out the other new features</p> </div> <p>This value will set the depth of the tag path to include in the pen name.</p> <p>Using the Sample Quick Start project as an example, the realistic0 Tag has a parent folder of realistic and a grandparent folder Power Chart will be realistic0 if the penNamePathDepth is set to 1, which is the default value. If the penNamePathDepth is set to would be sample_tags/realistic/realistic1. You can also add new Tags with path depth values that differ from existing Tags if ne</p>						
tagBrowserStartPath	<p>A path to a nested Tag History provider structure from which browsing will start. The path is expected to contain key-value pairs s</p> <pre data-bbox="358 1570 1487 1598">histprov:Sample_SQLite_Database:/drv:My_Gateway:My_Tag_Provider:/tag:My_Folder/Another_Folder</pre> <ul data-bbox="375 1654 1500 1745" style="list-style-type: none"> • histprov - The name of the Tag Historian Provider • drv - The historian driver, which is typically a combination of a gateway name and tag provider name separated by a colon. E • tag - A path to a node that has children. Typically should lead to either a folder or the root node of a UDT instance. If omitted drv, <p>While providing a path to this property, the tag component can be omitted, which will set the starting path for the Tag Browser par example:</p> <pre data-bbox="358 1850 1149 1877">histprov:Sample_SQLite_Database:/drv:My_Gateway:My_Tag_Provider:/</pre>						

mode	The type of query that is being made against the data source. Options are realtime or historical.
refreshRate	Duration (in milliseconds) that data will be queried for updated results. (realtime mode only)
pointCount	Number of data points returned for the selected time range. Note: Changing the pointCount property's value to -1 will retrieve pen data points as it is stored in the database. In other words, binding's Query Mode to "AsStored". While querying data with this mode, multiple value changes at the same timestamp will result in a single unique value.
startDate	Start date for a historical data query. (historical mode only)
endDate	End date for a historical data query. (historical mode only)
dateFormat	The date format displayed when in historical mode using a MomentJS date string (https://momentjs.com). (historical mode only)
timeFormat	The time format displayed when in historical mode using using a MomentJS time string (https://momentjs.com). (historical mode only)
rangeSelectorPen	The pen that will drive the data display of the range selector. (historical mode only)
unitOfTime	Time unit used for a realtime data query. (realtime mode only)
measureOfTime	Time measurement used for a realtime data query. Options are seconds, minutes, hours, days, weeks, months, or years. (realtime mode only)
rangeStartDate	Start date for the modified chart data range that the user has selected, either with the range brush or by panning/zooming. Read-only
rangeEndDate	End date for the modified chart data range that the user has selected either with the range brush or by panning/zooming. Read-only
responsiveDesignWidth	A number (in pixels) that will be used as the switch over width to the responsive design for the chart so it fits better on mobile devices and is configurable.

visibility

The following feature is new in Ignition version **8.1.2**
[Click here](#) to check out the other new features

Settings to show/hide elements within the component interface.

Name	Description																		
showTagBrowser	Flag representing the visible state of the Tag Browser. Toggling this property will show or hide the Tag E																		
showDateRangeSelector	Flag representing the visible state of the Date Range Selector. Toggling this property will show or hide th Selector.																		
showPenControlDisplay	Flag representing the visible state of the Pen Control display. Toggling this property will show or hide the lay																		
buttons	Settings to show/hide the buttons used in the interface																		
	<table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>showTagBrowserButton</td><td>Flag representing the visible state of the "Open Tag Browser" and "Close Tag Browser" buttons.</td></tr><tr><td>showPanZoomButton</td><td>Flag representing the visible state of the "Pan/Zoom" toggle button.</td></tr><tr><td>showXTraceButton</td><td>Flag representing the visible state of the "X Trace" toggle button.</td></tr><tr><td>showRangeBrushButton</td><td>Flag representing the visible state of the "Range Brush" toggle button.</td></tr><tr><td>showAnnotationButton</td><td>Flag representing the visible state of the "Annotation" toggle button.</td></tr><tr><td>showFullscreenButton</td><td>Flag representing the visible state of the "Fullscreen" toggle button.</td></tr><tr><td>showSettingsButton</td><td>Flag representing the visible state of the "Settings" toggle button.</td></tr><tr><td>showMoreButton</td><td>Flag representing the visible state of the "Show More" toggle button.</td></tr></tbody></table>	Name	Description	showTagBrowserButton	Flag representing the visible state of the "Open Tag Browser" and "Close Tag Browser" buttons.	showPanZoomButton	Flag representing the visible state of the "Pan/Zoom" toggle button.	showXTraceButton	Flag representing the visible state of the "X Trace" toggle button.	showRangeBrushButton	Flag representing the visible state of the "Range Brush" toggle button.	showAnnotationButton	Flag representing the visible state of the "Annotation" toggle button.	showFullscreenButton	Flag representing the visible state of the "Fullscreen" toggle button.	showSettingsButton	Flag representing the visible state of the "Settings" toggle button.	showMoreButton	Flag representing the visible state of the "Show More" toggle button.
Name	Description																		
showTagBrowserButton	Flag representing the visible state of the "Open Tag Browser" and "Close Tag Browser" buttons.																		
showPanZoomButton	Flag representing the visible state of the "Pan/Zoom" toggle button.																		
showXTraceButton	Flag representing the visible state of the "X Trace" toggle button.																		
showRangeBrushButton	Flag representing the visible state of the "Range Brush" toggle button.																		
showAnnotationButton	Flag representing the visible state of the "Annotation" toggle button.																		
showFullscreenButton	Flag representing the visible state of the "Fullscreen" toggle button.																		
showSettingsButton	Flag representing the visible state of the "Settings" toggle button.																		
showMoreButton	Flag representing the visible state of the "Show More" toggle button.																		

export

The following feature is new in Ignition version **8.1.2**
[Click here](#) to check out the other new features

Settings to control the format of data exported from the chart via the More button's "Export" option.

Name	Description	Property Type
dateFormat	The date format of the exported data. See https://numeraljs.com for formats.	value: string
timeFormat	The time format of the exported data. See https://momentjs.com for formats.	value: string

interaction

Configuration for the presentation of, and interaction with, chart data.

Name	Description
mode	Current user interaction mode of the chart. Options are panAndZoom, xTrace, rangeBrush, or annotation.

panAndZoom

Configuration settings for the chart in panAndZoom mode.

Name	Description
freeRange	<p>The following feature is new in Ignition version 8.1.18 Click here to check out the other new features</p> <p>When enabled, this setting allows panning and zooming to dictate the time range used for the chart display.</p>

xTrace

Configuration settings for the chart in xTrace mode.

Name	Description
values	An array of read-only timestamp values representing the visible x-trace positions.

infoBox	Configuration to build the box portion of the x-trace display.																	
	Name	Description																
	visible	Whether or not the box is visible.																
	showTime	Whether or not the time value above the box is visible.																
	width	The width of the box.																
	dataFormat	<p>A numeral.js data format for displaying the data for this axis. See https://numeraljs.com for formats.</p> <div style="border: 1px solid orange; padding: 5px; margin: 5px 0;"> <p>The following feature is new in Ignition version 8.1.2 Click here to check out the other new features</p> </div> <p>As of Ignition 8.1.2, setting the dataFormat property to an empty string will result in no formatting be applied to the value. Useful in cases where a binding is returning the data in a preformatted state.</p>																
	dateFormat	The date format displayed when in historical mode using a MomentJS date string (https://momentjs.com).																
	timeFormat	The time format displayed when in historical mode using using a MomentJS time string (https://momentjs.com).																
	stroke	A configuration object describing the properties that will be applied to the stroke of the box.																
		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>The color to apply to the line stroke, if applicable.</td> <td>color</td> </tr> <tr> <td>width</td> <td>The width to apply to the line stroke, if applicable.</td> <td>value: numeric</td> </tr> <tr> <td>opacity</td> <td>The opacity to apply to the line stroke, if applicable.</td> <td>value: numeric</td> </tr> <tr> <td>dashArray</td> <td>The spacing to apply between dashes of the line stroke, if applicable.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	The color to apply to the line stroke, if applicable.	color	width	The width to apply to the line stroke, if applicable.	value: numeric	opacity	The opacity to apply to the line stroke, if applicable.	value: numeric	dashArray	The spacing to apply between dashes of the line stroke, if applicable.	value: numeric	
Name	Description	Property Type																
color	The color to apply to the line stroke, if applicable.	color																
width	The width to apply to the line stroke, if applicable.	value: numeric																
opacity	The opacity to apply to the line stroke, if applicable.	value: numeric																
dashArray	The spacing to apply between dashes of the line stroke, if applicable.	value: numeric																
	fill	The fill configuration.																
		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>The color to apply to the trend fill, if applicable.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>The opacity to apply to the trend fill, if applicable.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	The color to apply to the trend fill, if applicable.	color	opacity	The opacity to apply to the trend fill, if applicable.	value: numeric							
Name	Description	Property Type																
color	The color to apply to the trend fill, if applicable.	color																
opacity	The opacity to apply to the trend fill, if applicable.	value: numeric																
	style	Style for the box. Full menu of style options is available. You can also specify a style class .																
line	Configuration to build the vertical line portion of the x-trace display.																	
	Name	Description	P															
	visible	Whether or not the line is visible.	val															
	color	The color to apply to the line stroke, if applicable.	col															
	width	The width to apply to the line stroke, if applicable.	val															
	opacity	The opacity to apply to the line stroke, if applicable.	val															
	dashArray	The spacing to apply between dashes of the line stroke, if applicable.	val															
	style	Style for the box. Full menu of style options is available. You can also specify a style class .	obj															

r
a
n
g
e
B
r
u
s
h

Configuration settings for the chart in rangeBrush mode.

Name	Description									
values	An array of config objects to build each range brush. <table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>start</td><td>The start timestamp position.</td><td>value: string</td></tr><tr><td>end</td><td>The end timestamp position.</td><td>value: string</td></tr></tbody></table>	Name	Description	Property Type	start	The start timestamp position.	value: string	end	The end timestamp position.	value: string
Name	Description	Property Type								
start	The start timestamp position.	value: string								
end	The end timestamp position.	value: string								
active	Configuration to build the active range brush display. <table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>color</td><td>The color of the active range brush.</td></tr><tr><td>opacity</td><td>The opacity of the active range brush.</td></tr><tr><td>style</td><td>Style settings for the active range brush. Full menu of style options is available. You can also specify a style class.</td></tr></tbody></table>	Name	Description	color	The color of the active range brush.	opacity	The opacity of the active range brush.	style	Style settings for the active range brush. Full menu of style options is available. You can also specify a style class .	
Name	Description									
color	The color of the active range brush.									
opacity	The opacity of the active range brush.									
style	Style settings for the active range brush. Full menu of style options is available. You can also specify a style class .									
inactive	Configuration to build the inactive range brush displays. <table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>color</td><td>The color of the inactive range brush.</td></tr><tr><td>opacity</td><td>The opacity of the inactive range brush.</td></tr><tr><td>style</td><td>Style settings for the inactive range brush. Full menu of style options is available. You can also specify a style class.</td></tr></tbody></table>	Name	Description	color	The color of the inactive range brush.	opacity	The opacity of the inactive range brush.	style	Style settings for the inactive range brush. Full menu of style options is available. You can also specify a style class .	
Name	Description									
color	The color of the inactive range brush.									
opacity	The opacity of the inactive range brush.									
style	Style settings for the inactive range brush. Full menu of style options is available. You can also specify a style class .									

a
n
n
o
t
a
t
i
o
n

Configuration settings for the chart in annotation mode.

Name	Description
------	-------------

info
Box

Configuration to build the box portion of the annotation display.

Name	Description															
visible	Whether or not the box is visible.															
showTime	Whether or not the time value above the box is visible.															
width	The width of the box.															
dateFormat	The date format displayed when in historical mode using a MomentJS date string (https://momentjs.com)															
timeFormat	The time format displayed when in historical mode using using a MomentJS time string (https://moment)															
stroke	A configuration object describing the properties that will be applied to the stroke of the box. <table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>color</td><td>The color to apply to the line stroke, if applicable.</td><td>color</td></tr><tr><td>width</td><td>The width to apply to the line stroke, if applicable.</td><td>value: numeric</td></tr><tr><td>opacity</td><td>The opacity to apply to the line stroke, if applicable.</td><td>value: numeric</td></tr><tr><td>dashArray</td><td>The spacing to apply between dashes of the line stroke, if applicable.</td><td>value: numeric</td></tr></tbody></table>	Name	Description	Property Type	color	The color to apply to the line stroke, if applicable.	color	width	The width to apply to the line stroke, if applicable.	value: numeric	opacity	The opacity to apply to the line stroke, if applicable.	value: numeric	dashArray	The spacing to apply between dashes of the line stroke, if applicable.	value: numeric
Name	Description	Property Type														
color	The color to apply to the line stroke, if applicable.	color														
width	The width to apply to the line stroke, if applicable.	value: numeric														
opacity	The opacity to apply to the line stroke, if applicable.	value: numeric														
dashArray	The spacing to apply between dashes of the line stroke, if applicable.	value: numeric														
fill	The fill configuration. <table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>color</td><td>The color to apply to the trend fill, if applicable.</td><td>color</td></tr><tr><td>opacity</td><td>The opacity to apply to the trend fill, if applicable.</td><td>value: numeric</td></tr></tbody></table>	Name	Description	Property Type	color	The color to apply to the trend fill, if applicable.	color	opacity	The opacity to apply to the trend fill, if applicable.	value: numeric						
Name	Description	Property Type														
color	The color to apply to the trend fill, if applicable.	color														
opacity	The opacity to apply to the trend fill, if applicable.	value: numeric														
font	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;">The following feature is new in Ignition version 8.1.17 Click here to check out the other new features</div> A configuration object describing the properties that will be applied to the font of text in the annotation. <table border="1"><thead><tr><th>Name</th><th>Description</th><th></th></tr></thead><tbody><tr><td>color</td><td>The text color of the annotation label and datetime text.</td><td>c</td></tr><tr><td>size</td><td>The font size of the annotation label and datetime text.</td><td>v r</td></tr><tr><td>style</td><td>Custom CSS styles to apply to the annotation text. Any style that applies to an SVG <code>line</code> element can be used. See also style options.</td><td>c</td></tr></tbody></table>	Name	Description		color	The text color of the annotation label and datetime text.	c	size	The font size of the annotation label and datetime text.	v r	style	Custom CSS styles to apply to the annotation text. Any style that applies to an SVG <code>line</code> element can be used. See also style options .	c			
Name	Description															
color	The text color of the annotation label and datetime text.	c														
size	The font size of the annotation label and datetime text.	v r														
style	Custom CSS styles to apply to the annotation text. Any style that applies to an SVG <code>line</code> element can be used. See also style options .	c														
style	Style for the box. Full menu of style options is available. You can also specify a style class .															

line	Configuration to build the connecting line portion of the annotation display.		
	Name	Description	Prop
	visible	Whether or not the line is visible.	value:
	color	The color to apply to the line stroke, if applicable.	color
	width	The width to apply to the line stroke, if applicable.	value:
	opacity	The opacity to apply to the line stroke, if applicable.	value:
	dashArray	The spacing to apply between dashes of the line stroke, if applicable.	value:
	style	Style for the box. Full menu of style options is available. You can also specify a style class .	object
dot	Configuration to build the dot portion of the annotation display.		
	Name	Description	
	visible	Whether or not the dot portion is visible.	
	color	The color to apply to the dot.	
	radius	The radius of the dot.	
	opacity	The opacity of the dot.	
	style	The style settings for the dot. Full menu of style options is available. You can also specify a style class .	

fullscreen Flag representing the full screen presentation mode of the chart.

chartZoomLevel

The following feature is new in Ignition version **8.1.11**
[Click here](#) to check out the other new features

Read-only value that corresponds to the current zoom level on the chart. External changes to this value will not update the level of

rangeZoomLevel

The following feature is new in Ignition version **8.1.11**
[Click here](#) to check out the other new features

Read-only value that corresponds to the current zoom level on the range brush when the chart is in Historical Mode. External changes to this value will not update the level of zoom displayed on the chart.

axes

Collection of predefined axes against which the data visualizations can be drawn.

Name	Description
name	The name of the axis.
position	The side of the plot upon which the axis should be drawn. Options are left or right.
width	The width of the axis.
color	The color of the Y axis vertical bar.
dataFormat	A numeral.js data format for displaying the data displayed in the pen control portion of the chart for this axis. See https://numeraljs.com/

range	<p>Configuration for the upper and lower limits of the axis.</p> <table border="1"> <thead> <tr> <th data-bbox="358 180 456 233">Name</th> <th data-bbox="456 180 1508 233">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="358 233 456 432">auto</td> <td data-bbox="456 233 1508 432"> <p>If true, the minimum and maximum displaying values for the axis will be auto calculated.</p> <div data-bbox="467 289 1377 401" style="border: 1px solid black; padding: 5px;"> <p><u>This feature was changed in Ignition version 8.1.16:</u> Auto range Y axes are now constrained to the upper/lower limits of the pen data bound to said axis.</p> </div> </td> </tr> <tr> <td data-bbox="358 432 456 478">min</td> <td data-bbox="456 432 1508 478">Minimum range value. If no value is provided, a minimum value will be calculated from the data bound to this axis.</td> </tr> <tr> <td data-bbox="358 478 456 527">max</td> <td data-bbox="456 478 1508 527">Maximum range value. If no value is provided, a maximum value will be calculated from the data bound to this axis.</td> </tr> </tbody> </table>	Name	Description	auto	<p>If true, the minimum and maximum displaying values for the axis will be auto calculated.</p> <div data-bbox="467 289 1377 401" style="border: 1px solid black; padding: 5px;"> <p><u>This feature was changed in Ignition version 8.1.16:</u> Auto range Y axes are now constrained to the upper/lower limits of the pen data bound to said axis.</p> </div>	min	Minimum range value. If no value is provided, a minimum value will be calculated from the data bound to this axis.	max	Maximum range value. If no value is provided, a maximum value will be calculated from the data bound to this axis.										
Name	Description																		
auto	<p>If true, the minimum and maximum displaying values for the axis will be auto calculated.</p> <div data-bbox="467 289 1377 401" style="border: 1px solid black; padding: 5px;"> <p><u>This feature was changed in Ignition version 8.1.16:</u> Auto range Y axes are now constrained to the upper/lower limits of the pen data bound to said axis.</p> </div>																		
min	Minimum range value. If no value is provided, a minimum value will be calculated from the data bound to this axis.																		
max	Maximum range value. If no value is provided, a maximum value will be calculated from the data bound to this axis.																		
label	<p>The label configuration for the Y axis.</p> <table border="1"> <thead> <tr> <th data-bbox="358 583 456 663">Name</th> <th data-bbox="456 583 1508 663">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="358 663 456 730">text</td> <td data-bbox="456 663 1508 730">The text of the Y axis label.</td> </tr> <tr> <td data-bbox="358 730 456 800">offset</td> <td data-bbox="456 730 1508 800">Offset the Y axis label from its default position. This enables you to fine tune the label location, which may be necessary to scale and how much room the tick labels take up. Value may be positive or negative. Default is 0.</td> </tr> <tr> <td data-bbox="358 800 456 846">font</td> <td data-bbox="456 800 1508 846">Font configuration for the Y access label. Font size and color options for the font.</td> </tr> <tr> <td data-bbox="358 846 456 898">style</td> <td data-bbox="456 846 1508 898">Style for the Y access label. Full menu of style options is available. You can also specify a style class.</td> </tr> </tbody> </table>	Name	Description	text	The text of the Y axis label.	offset	Offset the Y axis label from its default position. This enables you to fine tune the label location, which may be necessary to scale and how much room the tick labels take up. Value may be positive or negative. Default is 0.	font	Font configuration for the Y access label. Font size and color options for the font.	style	Style for the Y access label. Full menu of style options is available. You can also specify a style class .								
Name	Description																		
text	The text of the Y axis label.																		
offset	Offset the Y axis label from its default position. This enables you to fine tune the label location, which may be necessary to scale and how much room the tick labels take up. Value may be positive or negative. Default is 0.																		
font	Font configuration for the Y access label. Font size and color options for the font.																		
style	Style for the Y access label. Full menu of style options is available. You can also specify a style class .																		
tick	<p>Tick configuration.</p> <table border="1"> <thead> <tr> <th data-bbox="358 955 456 1035">Name</th> <th data-bbox="456 955 1508 1035">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="358 1035 456 1125">color</td> <td data-bbox="456 1035 1508 1125">The color of the ticks.</td> </tr> <tr> <td data-bbox="358 1125 456 1339">count</td> <td data-bbox="456 1125 1508 1339"> <div data-bbox="467 1157 1377 1241" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</p> </div> <p>Number of ticks to display on the Y axis. Default value is Auto. When the tick count is Auto, Y axis ticks will be added to the height of the chart, and the tick spacing is adjusted around whole numbers.</p> </td> </tr> <tr> <td data-bbox="358 1339 456 1808">label</td> <td data-bbox="456 1339 1508 1808"> <p>Tick label configuration</p> <table border="1"> <thead> <tr> <th data-bbox="467 1396 565 1476">Name</th> <th data-bbox="565 1396 1508 1476">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="467 1476 565 1522">font</td> <td data-bbox="565 1476 1508 1522">Label font configuration. Font size and color options for the font.</td> </tr> <tr> <td data-bbox="467 1522 565 1759">format</td> <td data-bbox="565 1522 1508 1759"> <div data-bbox="576 1549 1485 1633" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</p> </div> <p>Sets the numeric format for the tick label. The value must be a valid D3 Format value. Default value is Auto. When the tick label format is Auto, the D3 format <code>~f</code> is used. This creates a fixed point that will trim any insignificant trailing zeroes.</p> </td> </tr> <tr> <td data-bbox="467 1759 565 1808">style</td> <td data-bbox="565 1759 1508 1808">Style for the tick label. Full menu of style options is available. You can also specify a style class.</td> </tr> </tbody> </table> </td> </tr> <tr> <td data-bbox="358 1808 456 1858">style</td> <td data-bbox="456 1808 1508 1858">Style for the tick. Full menu of style options is available. You can also specify a style class.</td> </tr> </tbody> </table>	Name	Description	color	The color of the ticks.	count	<div data-bbox="467 1157 1377 1241" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</p> </div> <p>Number of ticks to display on the Y axis. Default value is Auto. When the tick count is Auto, Y axis ticks will be added to the height of the chart, and the tick spacing is adjusted around whole numbers.</p>	label	<p>Tick label configuration</p> <table border="1"> <thead> <tr> <th data-bbox="467 1396 565 1476">Name</th> <th data-bbox="565 1396 1508 1476">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="467 1476 565 1522">font</td> <td data-bbox="565 1476 1508 1522">Label font configuration. Font size and color options for the font.</td> </tr> <tr> <td data-bbox="467 1522 565 1759">format</td> <td data-bbox="565 1522 1508 1759"> <div data-bbox="576 1549 1485 1633" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</p> </div> <p>Sets the numeric format for the tick label. The value must be a valid D3 Format value. Default value is Auto. When the tick label format is Auto, the D3 format <code>~f</code> is used. This creates a fixed point that will trim any insignificant trailing zeroes.</p> </td> </tr> <tr> <td data-bbox="467 1759 565 1808">style</td> <td data-bbox="565 1759 1508 1808">Style for the tick label. Full menu of style options is available. You can also specify a style class.</td> </tr> </tbody> </table>	Name	Description	font	Label font configuration. Font size and color options for the font.	format	<div data-bbox="576 1549 1485 1633" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</p> </div> <p>Sets the numeric format for the tick label. The value must be a valid D3 Format value. Default value is Auto. When the tick label format is Auto, the D3 format <code>~f</code> is used. This creates a fixed point that will trim any insignificant trailing zeroes.</p>	style	Style for the tick label. Full menu of style options is available. You can also specify a style class .	style	Style for the tick. Full menu of style options is available. You can also specify a style class .
Name	Description																		
color	The color of the ticks.																		
count	<div data-bbox="467 1157 1377 1241" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</p> </div> <p>Number of ticks to display on the Y axis. Default value is Auto. When the tick count is Auto, Y axis ticks will be added to the height of the chart, and the tick spacing is adjusted around whole numbers.</p>																		
label	<p>Tick label configuration</p> <table border="1"> <thead> <tr> <th data-bbox="467 1396 565 1476">Name</th> <th data-bbox="565 1396 1508 1476">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="467 1476 565 1522">font</td> <td data-bbox="565 1476 1508 1522">Label font configuration. Font size and color options for the font.</td> </tr> <tr> <td data-bbox="467 1522 565 1759">format</td> <td data-bbox="565 1522 1508 1759"> <div data-bbox="576 1549 1485 1633" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</p> </div> <p>Sets the numeric format for the tick label. The value must be a valid D3 Format value. Default value is Auto. When the tick label format is Auto, the D3 format <code>~f</code> is used. This creates a fixed point that will trim any insignificant trailing zeroes.</p> </td> </tr> <tr> <td data-bbox="467 1759 565 1808">style</td> <td data-bbox="565 1759 1508 1808">Style for the tick label. Full menu of style options is available. You can also specify a style class.</td> </tr> </tbody> </table>	Name	Description	font	Label font configuration. Font size and color options for the font.	format	<div data-bbox="576 1549 1485 1633" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</p> </div> <p>Sets the numeric format for the tick label. The value must be a valid D3 Format value. Default value is Auto. When the tick label format is Auto, the D3 format <code>~f</code> is used. This creates a fixed point that will trim any insignificant trailing zeroes.</p>	style	Style for the tick label. Full menu of style options is available. You can also specify a style class .										
Name	Description																		
font	Label font configuration. Font size and color options for the font.																		
format	<div data-bbox="576 1549 1485 1633" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</p> </div> <p>Sets the numeric format for the tick label. The value must be a valid D3 Format value. Default value is Auto. When the tick label format is Auto, the D3 format <code>~f</code> is used. This creates a fixed point that will trim any insignificant trailing zeroes.</p>																		
style	Style for the tick label. Full menu of style options is available. You can also specify a style class .																		
style	Style for the tick. Full menu of style options is available. You can also specify a style class .																		

grid

The following feature is new in Ignition version **8.1.16**
[Click here](#) to check out the other new features

Configuration for gridlines to display on this axis.

Name	Description
visible	Visible state of the gridlines. Gridlines are shown only for axes that connect directly to the chart. Any satellite axes will configurations instead of gridlines.
color	Color of the gridlines.
opacity	Opacity of the gridlines.
dashArray	Dashed appearance of the gridlines.
style	Style for the gridlines. Full menu of style options is available. You can also specify a style class .

style Style for the display. Full menu of [style options](#) is available. You can also specify a [style class](#).

pens

Visual representation of each active item

Name	Description
name	Name of the pen.
visible	Whether or not the pen is visible on the chart.
enabled	Availability of the pen on the chart and pen configuration panel.
selectable	Flag to allow the pen to be responsive to user selection.
axis	Name of an axis in the "axes" array to plot against. If left blank, a default axis will be created based on data values.
plot	The plot to which this pen is bound.

display

Configuration that drives the display of the pen.

Name	Description																																								
type	The type of chart to be built. Options are line, area, bar, or scatter.																																								
interpolation	Type of curve that should be used to draw the line portion of the chart. Options are: curveBasis, curveBasisOpen, curveCardinalOpen, curveCatmullRom, curveCatmullRomOpen, curveLinear, curveMonotoneX, curveMonotoneY, curveStepAfter, or curveStepBefore. More information on the interpolation methods above can be found in D3's documentation .																																								
breakLine	If true, the line will be broken on either side of bad/missing data values. If false, bad/missing data values are removed and connected.																																								
styles	Settings for the display when it is normal, highlighted, selected, or muted. <table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>normal</td><td>An object providing style configuration for the "normal" state (no user interaction) of a column, or data entry trend. Any color values specified here will override values set in the colorScheme or colors properties.</td></tr><tr><td>highlighted</td><td>An object providing style configuration for the "highlighted" state (mouse hover) of a column, or data entry trend. Any color values specified here will override values set in the colorScheme or colors properties. The highlighted property uses the same configuration properties as the 'normal' property above.</td></tr><tr><td>selected</td><td>An object providing style configuration for the "selected" state (mouse click) of a column, or data entry, in a trend. Any color values specified here will override values set in the colorScheme or colors properties.</td></tr><tr><td>muted</td><td>An object providing style configuration for the "muted" state (non selected) of a column, or data entry, in a trend. Any color values specified here will override values set in the colorScheme or colors properties.</td></tr></tbody></table> <p>Each of these four display settings has the same set of properties.</p> <table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>stroke</td><td>A configuration object describing the properties that will be applied to the stroke of the trend type being displayed (if applicable). The line, area, and step trend types will have these styles applied to them. <table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>color</td><td>The color to apply to the line stroke, if applicable.</td><td>color</td></tr><tr><td>width</td><td>The width to apply to the line stroke, if applicable.</td><td>value: numeric</td></tr><tr><td>opacity</td><td>The opacity to apply to the line stroke, if applicable.</td><td>value: numeric</td></tr><tr><td>dashArray</td><td>The spacing to apply between dashes of the line stroke, if applicable.</td><td>value: numeric</td></tr></tbody></table></td></tr><tr><td>fill</td><td>The fill configuration. <table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>color</td><td>The color to apply to the trend fill, if applicable.</td><td>color</td></tr><tr><td>opacity</td><td>The opacity to apply to the trend fill, if applicable.</td><td>value: numeric</td></tr></tbody></table></td></tr></tbody></table>	Name	Description	normal	An object providing style configuration for the "normal" state (no user interaction) of a column, or data entry trend. Any color values specified here will override values set in the colorScheme or colors properties.	highlighted	An object providing style configuration for the "highlighted" state (mouse hover) of a column, or data entry trend. Any color values specified here will override values set in the colorScheme or colors properties. The highlighted property uses the same configuration properties as the 'normal' property above.	selected	An object providing style configuration for the "selected" state (mouse click) of a column, or data entry, in a trend. Any color values specified here will override values set in the colorScheme or colors properties.	muted	An object providing style configuration for the "muted" state (non selected) of a column, or data entry, in a trend. Any color values specified here will override values set in the colorScheme or colors properties.	Name	Description	stroke	A configuration object describing the properties that will be applied to the stroke of the trend type being displayed (if applicable). The line, area, and step trend types will have these styles applied to them. <table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>color</td><td>The color to apply to the line stroke, if applicable.</td><td>color</td></tr><tr><td>width</td><td>The width to apply to the line stroke, if applicable.</td><td>value: numeric</td></tr><tr><td>opacity</td><td>The opacity to apply to the line stroke, if applicable.</td><td>value: numeric</td></tr><tr><td>dashArray</td><td>The spacing to apply between dashes of the line stroke, if applicable.</td><td>value: numeric</td></tr></tbody></table>	Name	Description	Property Type	color	The color to apply to the line stroke, if applicable.	color	width	The width to apply to the line stroke, if applicable.	value: numeric	opacity	The opacity to apply to the line stroke, if applicable.	value: numeric	dashArray	The spacing to apply between dashes of the line stroke, if applicable.	value: numeric	fill	The fill configuration. <table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>color</td><td>The color to apply to the trend fill, if applicable.</td><td>color</td></tr><tr><td>opacity</td><td>The opacity to apply to the trend fill, if applicable.</td><td>value: numeric</td></tr></tbody></table>	Name	Description	Property Type	color	The color to apply to the trend fill, if applicable.	color	opacity	The opacity to apply to the trend fill, if applicable.	value: numeric
Name	Description																																								
normal	An object providing style configuration for the "normal" state (no user interaction) of a column, or data entry trend. Any color values specified here will override values set in the colorScheme or colors properties.																																								
highlighted	An object providing style configuration for the "highlighted" state (mouse hover) of a column, or data entry trend. Any color values specified here will override values set in the colorScheme or colors properties. The highlighted property uses the same configuration properties as the 'normal' property above.																																								
selected	An object providing style configuration for the "selected" state (mouse click) of a column, or data entry, in a trend. Any color values specified here will override values set in the colorScheme or colors properties.																																								
muted	An object providing style configuration for the "muted" state (non selected) of a column, or data entry, in a trend. Any color values specified here will override values set in the colorScheme or colors properties.																																								
Name	Description																																								
stroke	A configuration object describing the properties that will be applied to the stroke of the trend type being displayed (if applicable). The line, area, and step trend types will have these styles applied to them. <table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>color</td><td>The color to apply to the line stroke, if applicable.</td><td>color</td></tr><tr><td>width</td><td>The width to apply to the line stroke, if applicable.</td><td>value: numeric</td></tr><tr><td>opacity</td><td>The opacity to apply to the line stroke, if applicable.</td><td>value: numeric</td></tr><tr><td>dashArray</td><td>The spacing to apply between dashes of the line stroke, if applicable.</td><td>value: numeric</td></tr></tbody></table>	Name	Description	Property Type	color	The color to apply to the line stroke, if applicable.	color	width	The width to apply to the line stroke, if applicable.	value: numeric	opacity	The opacity to apply to the line stroke, if applicable.	value: numeric	dashArray	The spacing to apply between dashes of the line stroke, if applicable.	value: numeric																									
Name	Description	Property Type																																							
color	The color to apply to the line stroke, if applicable.	color																																							
width	The width to apply to the line stroke, if applicable.	value: numeric																																							
opacity	The opacity to apply to the line stroke, if applicable.	value: numeric																																							
dashArray	The spacing to apply between dashes of the line stroke, if applicable.	value: numeric																																							
fill	The fill configuration. <table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>color</td><td>The color to apply to the trend fill, if applicable.</td><td>color</td></tr><tr><td>opacity</td><td>The opacity to apply to the trend fill, if applicable.</td><td>value: numeric</td></tr></tbody></table>	Name	Description	Property Type	color	The color to apply to the trend fill, if applicable.	color	opacity	The opacity to apply to the trend fill, if applicable.	value: numeric																															
Name	Description	Property Type																																							
color	The color to apply to the trend fill, if applicable.	color																																							
opacity	The opacity to apply to the trend fill, if applicable.	value: numeric																																							

data	Name	Description
	source	Source or tag path of the data behind the pen. <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p>This feature was changed in Ignition version 8.1.14:</p> </div> <p>Non-historical tag paths are converted to a historical format (e.g. <code>histprov:default:/tag:_Simulator_/Ramp</code> to the pen via the UI will write the converted path back to this property. This keeps all of the <code>data.source</code> properties in still allowing an initial non-historical tag path to add pen data to the chart display.</p>
	aggregateMode	Mode to use to group the data. Options are: default (MinMax), Average, MinMax, LastValue, SimpleAverage, Sum, M DurationOn, DurationOff, CountOn, CountOff, Count, Range, Variance, StdDev, PctGood, or PctBad.

plots A plot represents a row containing one or more pens.

Name	Description																																							
relativeWeight	Ratio between all plots.																																							
color	Background color of the plot.																																							
markers	An array of markers that can be added to the plot to better visualize the data being displayed. <table border="1" style="margin-top: 10px;"> <tr> <td>Name</td> <td>Description</td> </tr> <tr> <td>type</td> <td>Type of marker to add to the plot. Options are line or band.</td> </tr> <tr> <td>axis</td> <td>Name of the axis against which the marker should be drawn. This must be specified for the marker to be drawn.</td> </tr> <tr> <td>value</td> <td>Value where the line marker should be drawn.</td> </tr> <tr> <td>display</td> <td>Configuration for the display of the marker. <table border="1" style="margin-top: 10px;"> <tr> <td>Name</td> <td>Description</td> </tr> <tr> <td>color</td> <td>Color of the marker</td> </tr> <tr> <td>width</td> <td>Width of the line.</td> </tr> <tr> <td>opacity</td> <td>Opacity of the marker</td> </tr> <tr> <td>dashArray</td> <td>Dashed appearance of the marker.</td> </tr> <tr> <td>label</td> <td>Configuration for the label of the marker. <table border="1" style="margin-top: 10px;"> <tr> <td>Name</td> <td>Description</td> <td>Property Type</td> </tr> <tr> <td>text</td> <td>Text for the label.</td> <td>value: string</td> </tr> <tr> <td>position</td> <td>The position of the label relative to the line. Options are right or left.</td> <td>value: string</td> </tr> <tr> <td>font</td> <td>Label font configuration. Font size and color options for the font.</td> <td>object</td> </tr> <tr> <td>style</td> <td>Style for the label. Full menu of style options is available. You can also specify a style class</td> <td>object</td> </tr> </table> </td> </tr> <tr> <td>style</td> <td>Style for the display. Full menu of style options is available. You can also specify a style class.</td> </tr> </table> </td> </tr> </table>	Name	Description	type	Type of marker to add to the plot. Options are line or band.	axis	Name of the axis against which the marker should be drawn. This must be specified for the marker to be drawn.	value	Value where the line marker should be drawn.	display	Configuration for the display of the marker. <table border="1" style="margin-top: 10px;"> <tr> <td>Name</td> <td>Description</td> </tr> <tr> <td>color</td> <td>Color of the marker</td> </tr> <tr> <td>width</td> <td>Width of the line.</td> </tr> <tr> <td>opacity</td> <td>Opacity of the marker</td> </tr> <tr> <td>dashArray</td> <td>Dashed appearance of the marker.</td> </tr> <tr> <td>label</td> <td>Configuration for the label of the marker. <table border="1" style="margin-top: 10px;"> <tr> <td>Name</td> <td>Description</td> <td>Property Type</td> </tr> <tr> <td>text</td> <td>Text for the label.</td> <td>value: string</td> </tr> <tr> <td>position</td> <td>The position of the label relative to the line. Options are right or left.</td> <td>value: string</td> </tr> <tr> <td>font</td> <td>Label font configuration. Font size and color options for the font.</td> <td>object</td> </tr> <tr> <td>style</td> <td>Style for the label. Full menu of style options is available. You can also specify a style class</td> <td>object</td> </tr> </table> </td> </tr> <tr> <td>style</td> <td>Style for the display. Full menu of style options is available. You can also specify a style class.</td> </tr> </table>	Name	Description	color	Color of the marker	width	Width of the line.	opacity	Opacity of the marker	dashArray	Dashed appearance of the marker.	label	Configuration for the label of the marker. <table border="1" style="margin-top: 10px;"> <tr> <td>Name</td> <td>Description</td> <td>Property Type</td> </tr> <tr> <td>text</td> <td>Text for the label.</td> <td>value: string</td> </tr> <tr> <td>position</td> <td>The position of the label relative to the line. Options are right or left.</td> <td>value: string</td> </tr> <tr> <td>font</td> <td>Label font configuration. Font size and color options for the font.</td> <td>object</td> </tr> <tr> <td>style</td> <td>Style for the label. Full menu of style options is available. You can also specify a style class</td> <td>object</td> </tr> </table>	Name	Description	Property Type	text	Text for the label.	value: string	position	The position of the label relative to the line. Options are right or left.	value: string	font	Label font configuration. Font size and color options for the font.	object	style	Style for the label. Full menu of style options is available. You can also specify a style class	object	style	Style for the display. Full menu of style options is available. You can also specify a style class .
Name	Description																																							
type	Type of marker to add to the plot. Options are line or band.																																							
axis	Name of the axis against which the marker should be drawn. This must be specified for the marker to be drawn.																																							
value	Value where the line marker should be drawn.																																							
display	Configuration for the display of the marker. <table border="1" style="margin-top: 10px;"> <tr> <td>Name</td> <td>Description</td> </tr> <tr> <td>color</td> <td>Color of the marker</td> </tr> <tr> <td>width</td> <td>Width of the line.</td> </tr> <tr> <td>opacity</td> <td>Opacity of the marker</td> </tr> <tr> <td>dashArray</td> <td>Dashed appearance of the marker.</td> </tr> <tr> <td>label</td> <td>Configuration for the label of the marker. <table border="1" style="margin-top: 10px;"> <tr> <td>Name</td> <td>Description</td> <td>Property Type</td> </tr> <tr> <td>text</td> <td>Text for the label.</td> <td>value: string</td> </tr> <tr> <td>position</td> <td>The position of the label relative to the line. Options are right or left.</td> <td>value: string</td> </tr> <tr> <td>font</td> <td>Label font configuration. Font size and color options for the font.</td> <td>object</td> </tr> <tr> <td>style</td> <td>Style for the label. Full menu of style options is available. You can also specify a style class</td> <td>object</td> </tr> </table> </td> </tr> <tr> <td>style</td> <td>Style for the display. Full menu of style options is available. You can also specify a style class.</td> </tr> </table>	Name	Description	color	Color of the marker	width	Width of the line.	opacity	Opacity of the marker	dashArray	Dashed appearance of the marker.	label	Configuration for the label of the marker. <table border="1" style="margin-top: 10px;"> <tr> <td>Name</td> <td>Description</td> <td>Property Type</td> </tr> <tr> <td>text</td> <td>Text for the label.</td> <td>value: string</td> </tr> <tr> <td>position</td> <td>The position of the label relative to the line. Options are right or left.</td> <td>value: string</td> </tr> <tr> <td>font</td> <td>Label font configuration. Font size and color options for the font.</td> <td>object</td> </tr> <tr> <td>style</td> <td>Style for the label. Full menu of style options is available. You can also specify a style class</td> <td>object</td> </tr> </table>	Name	Description	Property Type	text	Text for the label.	value: string	position	The position of the label relative to the line. Options are right or left.	value: string	font	Label font configuration. Font size and color options for the font.	object	style	Style for the label. Full menu of style options is available. You can also specify a style class	object	style	Style for the display. Full menu of style options is available. You can also specify a style class .										
Name	Description																																							
color	Color of the marker																																							
width	Width of the line.																																							
opacity	Opacity of the marker																																							
dashArray	Dashed appearance of the marker.																																							
label	Configuration for the label of the marker. <table border="1" style="margin-top: 10px;"> <tr> <td>Name</td> <td>Description</td> <td>Property Type</td> </tr> <tr> <td>text</td> <td>Text for the label.</td> <td>value: string</td> </tr> <tr> <td>position</td> <td>The position of the label relative to the line. Options are right or left.</td> <td>value: string</td> </tr> <tr> <td>font</td> <td>Label font configuration. Font size and color options for the font.</td> <td>object</td> </tr> <tr> <td>style</td> <td>Style for the label. Full menu of style options is available. You can also specify a style class</td> <td>object</td> </tr> </table>	Name	Description	Property Type	text	Text for the label.	value: string	position	The position of the label relative to the line. Options are right or left.	value: string	font	Label font configuration. Font size and color options for the font.	object	style	Style for the label. Full menu of style options is available. You can also specify a style class	object																								
Name	Description	Property Type																																						
text	Text for the label.	value: string																																						
position	The position of the label relative to the line. Options are right or left.	value: string																																						
font	Label font configuration. Font size and color options for the font.	object																																						
style	Style for the label. Full menu of style options is available. You can also specify a style class	object																																						
style	Style for the display. Full menu of style options is available. You can also specify a style class .																																							
style	Style for the individual plot. Full menu of style options is available for text, background, margin and padding, border, shape and specify a style class																																							

dataColumns

Configuration for the data columns that can be shown in tabular displays throughout the chart.

Name	Description	Property Type																																				
penControl	Configuration for the data columns that can display for pens.	boolean																																				
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>currentValue</td> <td>Show the "current value" column for the pen based on the time range.</td> <td>value: boolean</td> </tr> <tr> <td>minimum</td> <td>Show the "minimum" column for the pen based on the time range.</td> <td>value: boolean</td> </tr> <tr> <td>maximum</td> <td>Show the "minimum" column for the pen based on the time range.</td> <td>value: boolean</td> </tr> <tr> <td>average</td> <td>Show the "average" column for the pen based on the time range.</td> <td>value: boolean</td> </tr> <tr> <td>axis</td> <td>Show the "axis" column for the pen.</td> <td>value: boolean</td> </tr> <tr> <td>plot</td> <td>Show the "plot" column for the pen.</td> <td>value: boolean</td> </tr> <tr> <td>xTrace</td> <td>Show the "xTrace" column for the pen based on the time range.</td> <td>value: boolean</td> </tr> </tbody> </table>	Name	Description	Property Type	currentValue	Show the "current value" column for the pen based on the time range.	value: boolean	minimum	Show the "minimum" column for the pen based on the time range.	value: boolean	maximum	Show the "minimum" column for the pen based on the time range.	value: boolean	average	Show the "average" column for the pen based on the time range.	value: boolean	axis	Show the "axis" column for the pen.	value: boolean	plot	Show the "plot" column for the pen.	value: boolean	xTrace	Show the "xTrace" column for the pen based on the time range.	value: boolean													
Name	Description	Property Type																																				
currentValue	Show the "current value" column for the pen based on the time range.	value: boolean																																				
minimum	Show the "minimum" column for the pen based on the time range.	value: boolean																																				
maximum	Show the "minimum" column for the pen based on the time range.	value: boolean																																				
average	Show the "average" column for the pen based on the time range.	value: boolean																																				
axis	Show the "axis" column for the pen.	value: boolean																																				
plot	Show the "plot" column for the pen.	value: boolean																																				
xTrace	Show the "xTrace" column for the pen based on the time range.	value: boolean																																				
rangeSelection	Configuration for the data columns that can display for the range brush.	boolean																																				
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>first</td> <td>Show the "first" column for the pen based on the time range.</td> <td>value: boolean</td> </tr> <tr> <td>last</td> <td>Show the "last" column for the pen based on the time range.</td> <td>value: boolean</td> </tr> <tr> <td>average</td> <td>Show the "average" column for the pen based on the time range.</td> <td>value: boolean</td> </tr> <tr> <td>minimum</td> <td>Show the "minimum" column for the pen based on the time range.</td> <td>value: boolean</td> </tr> <tr> <td>maximum</td> <td>Show the "minimum" column for the pen based on the time range.</td> <td>value: boolean</td> </tr> <tr> <td>median</td> <td>Show the "median" column for the pen based on the time range.</td> <td>value: boolean</td> </tr> <tr> <td>delta</td> <td>Show the "delta" column for the pen based on the time range.</td> <td>value: boolean</td> </tr> <tr> <td>sum</td> <td>Show the "median" column for the pen based on the time range.</td> <td>value: boolean</td> </tr> <tr> <td>ucl</td> <td>Show the "UCL" column for the pen based on the time range.</td> <td>value: boolean</td> </tr> <tr> <td>lcl</td> <td>Show the "LCL" column for the pen based on the time range.</td> <td>value: boolean</td> </tr> <tr> <td>standardDeviation</td> <td>Show the "standardDeviation" column for the pen based on the time range.</td> <td>value: boolean</td> </tr> </tbody> </table>	Name	Description	Property Type	first	Show the "first" column for the pen based on the time range.	value: boolean	last	Show the "last" column for the pen based on the time range.	value: boolean	average	Show the "average" column for the pen based on the time range.	value: boolean	minimum	Show the "minimum" column for the pen based on the time range.	value: boolean	maximum	Show the "minimum" column for the pen based on the time range.	value: boolean	median	Show the "median" column for the pen based on the time range.	value: boolean	delta	Show the "delta" column for the pen based on the time range.	value: boolean	sum	Show the "median" column for the pen based on the time range.	value: boolean	ucl	Show the "UCL" column for the pen based on the time range.	value: boolean	lcl	Show the "LCL" column for the pen based on the time range.	value: boolean	standardDeviation	Show the "standardDeviation" column for the pen based on the time range.	value: boolean	
Name	Description	Property Type																																				
first	Show the "first" column for the pen based on the time range.	value: boolean																																				
last	Show the "last" column for the pen based on the time range.	value: boolean																																				
average	Show the "average" column for the pen based on the time range.	value: boolean																																				
minimum	Show the "minimum" column for the pen based on the time range.	value: boolean																																				
maximum	Show the "minimum" column for the pen based on the time range.	value: boolean																																				
median	Show the "median" column for the pen based on the time range.	value: boolean																																				
delta	Show the "delta" column for the pen based on the time range.	value: boolean																																				
sum	Show the "median" column for the pen based on the time range.	value: boolean																																				
ucl	Show the "UCL" column for the pen based on the time range.	value: boolean																																				
lcl	Show the "LCL" column for the pen based on the time range.	value: boolean																																				
standardDeviation	Show the "standardDeviation" column for the pen based on the time range.	value: boolean																																				

title

Configuration for the title of the chart.

Name	Description	Property Type
visible	Whether or not the title is visible.	value: boolean
text	The text for the title.	value: string
font	Title font configuration. Font size and color options for the font.	object
style	Style for the display. Full menu of style options is available. You can also specify a style class .	object

timeAxis Configuration for the time axis (X axis) of the chart.

Name	Description																														
visible	Whether or not the time axis is visible.																														
tickCount	<p>The number of ticks on the axis.</p> <div style="border: 1px solid orange; padding: 5px; margin: 5px 0;"> <p>The following feature is new in Ignition version 8.1.22 Click here to check out the other new features</p> </div> <p>Setting the tickCount to 0 will automatically scale the chart's X Axis based on the zoom level, panning, and scrolling.</p>																														
height	The height of the time axis.																														
color	The color of the axis.																														
tick	<p>Tick configuration.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>The color of the ticks.</td> <td></td> </tr> <tr> <td>label</td> <td>Tick label configuration</td> <td></td> </tr> <tr> <td></td> <td> <table border="1" style="width: 100%;"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>angled</td> <td>Whether or not the tick labels are angled.</td> <td>value:</td> </tr> <tr> <td>format</td> <td>Date/time format displayed by each tick using a MomentJS data string (https://momentjs.com).</td> <td>value:</td> </tr> <tr> <td>font</td> <td>Label font configuration. Font size and color options for the font.</td> <td>object</td> </tr> <tr> <td>style</td> <td>Style for the tick label. Full menu of style options is available. You can also specify a style class .</td> <td>object</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>style</td> <td>Style for the tick. Full menu of style options is available. You can also specify a style class .</td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	color	The color of the ticks.		label	Tick label configuration			<table border="1" style="width: 100%;"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>angled</td> <td>Whether or not the tick labels are angled.</td> <td>value:</td> </tr> <tr> <td>format</td> <td>Date/time format displayed by each tick using a MomentJS data string (https://momentjs.com).</td> <td>value:</td> </tr> <tr> <td>font</td> <td>Label font configuration. Font size and color options for the font.</td> <td>object</td> </tr> <tr> <td>style</td> <td>Style for the tick label. Full menu of style options is available. You can also specify a style class .</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	angled	Whether or not the tick labels are angled.	value:	format	Date/time format displayed by each tick using a MomentJS data string (https://momentjs.com).	value:	font	Label font configuration. Font size and color options for the font.	object	style	Style for the tick label. Full menu of style options is available. You can also specify a style class .	object		style	Style for the tick. Full menu of style options is available. You can also specify a style class .	
Name	Description	Property Type																													
color	The color of the ticks.																														
label	Tick label configuration																														
	<table border="1" style="width: 100%;"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>angled</td> <td>Whether or not the tick labels are angled.</td> <td>value:</td> </tr> <tr> <td>format</td> <td>Date/time format displayed by each tick using a MomentJS data string (https://momentjs.com).</td> <td>value:</td> </tr> <tr> <td>font</td> <td>Label font configuration. Font size and color options for the font.</td> <td>object</td> </tr> <tr> <td>style</td> <td>Style for the tick label. Full menu of style options is available. You can also specify a style class .</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	angled	Whether or not the tick labels are angled.	value:	format	Date/time format displayed by each tick using a MomentJS data string (https://momentjs.com).	value:	font	Label font configuration. Font size and color options for the font.	object	style	Style for the tick label. Full menu of style options is available. You can also specify a style class .	object															
Name	Description	Property Type																													
angled	Whether or not the tick labels are angled.	value:																													
format	Date/time format displayed by each tick using a MomentJS data string (https://momentjs.com).	value:																													
font	Label font configuration. Font size and color options for the font.	object																													
style	Style for the tick label. Full menu of style options is available. You can also specify a style class .	object																													
style	Style for the tick. Full menu of style options is available. You can also specify a style class .																														
grid	<div style="border: 1px solid orange; padding: 5px; margin: 5px 0;"> <p>The following feature is new in Ignition version 8.1.16 Click here to check out the other new features</p> </div> <p>Configuration for gridlines to display on this axis.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>visible</td> <td>Visible state of the gridlines. Gridlines are shown only for axes that connect directly to the chart. Any satellite axes will use satellite axis configurations instead of gridlines.</td> </tr> <tr> <td>color</td> <td>Color of the gridlines.</td> </tr> <tr> <td>opacity</td> <td>Opacity of the gridlines.</td> </tr> <tr> <td>dashArray</td> <td>Dashed appearance of the gridlines.</td> </tr> <tr> <td>style</td> <td>Style for the gridlines. Full menu of style options is available. You can also specify a style class .</td> </tr> </tbody> </table>	Name	Description	visible	Visible state of the gridlines. Gridlines are shown only for axes that connect directly to the chart. Any satellite axes will use satellite axis configurations instead of gridlines.	color	Color of the gridlines.	opacity	Opacity of the gridlines.	dashArray	Dashed appearance of the gridlines.	style	Style for the gridlines. Full menu of style options is available. You can also specify a style class .																		
Name	Description																														
visible	Visible state of the gridlines. Gridlines are shown only for axes that connect directly to the chart. Any satellite axes will use satellite axis configurations instead of gridlines.																														
color	Color of the gridlines.																														
opacity	Opacity of the gridlines.																														
dashArray	Dashed appearance of the gridlines.																														
style	Style for the gridlines. Full menu of style options is available. You can also specify a style class .																														
style	Sets a style for this timeAxis. Full menu of style options is available for text, background, margin and padding, border, shape and style class .																														

legend Configuration for the display of the legend for the chart.

Name	Description	Property Type
visible	Whether or not the legend is visible.	value: boolean

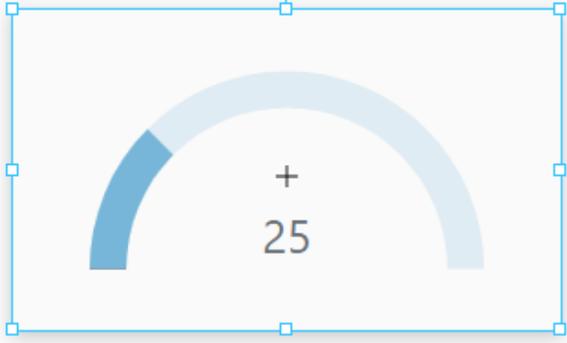
style	Sets a style for this chart. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous
-------	--

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Perspective - Simple Gauge



On this page ...

- [Properties](#)
- [Component Events](#)
- [Example](#)

Component Palette Icon:



The Simple Gauge component in Perspective provides a way to show realtime values in a range as they change. This gauge is a less complicated version of the [Gauge](#) component. It has just one axis, is easy to configure and is customizable in its appearance.

The following feature is new in Ignition version **8.1.2**
[Click here](#) to check out the other new features

The Simple Gauge component has three pre-configured [variants](#):

- Half Circle - Default layout with a half-circle gauge.
- 3/4 Circle - Layout with a 3/4 circle gauge.
- Full Axis - Layout with a full axis gauge.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
value	Numeric value for the gauge to display. Default is 0.	value: numeric
minValue	Minimum gauge value for this gauge. Default is 0.	value: numeric
maxValue	Maximum gauge value for this gauge. Default is 100.	value: numeric
startAngle	Radial position for the start of the gauge's arc. Default is 180.	value: numeric
endAngle	Radial position for the end of the gauge's arc. Default is 360.	value: numeric

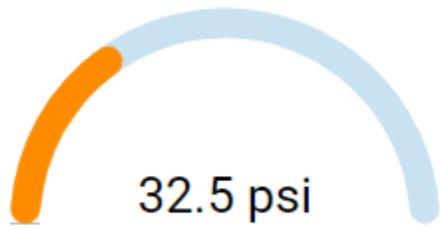
arc	The arc is a radial band that displays the gauge's value.	object																								
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>width</td> <td>Width of the line (in pixels) that represents the arc. Default is 20.</td> <td>value: numeric</td> </tr> <tr> <td>color</td> <td>Color of the arc line showing the gauge's value. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>color</td> </tr> <tr> <td>cornerRadius</td> <td>Amount to round the edges of the arc. Default is 0.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	width	Width of the line (in pixels) that represents the arc. Default is 20.	value: numeric	color	Color of the arc line showing the gauge's value. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color	cornerRadius	Amount to round the edges of the arc. Default is 0.	value: numeric													
Name	Description	Property Type																								
width	Width of the line (in pixels) that represents the arc. Default is 20.	value: numeric																								
color	Color of the arc line showing the gauge's value. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color																								
cornerRadius	Amount to round the edges of the arc. Default is 0.	value: numeric																								
arcBackground	Background or 'track' for the gauge arc. Shows shape and total potential value behind the arc.	value: numeric																								
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color of the arc background. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the arc. 0 is fully transparent, 1 is fully opaque.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color of the arc background. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value.	color	opacity	Opacity of the arc. 0 is fully transparent, 1 is fully opaque.	value: numeric																
Name	Description	Property Type																								
color	Color of the arc background. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value.	color																								
opacity	Opacity of the arc. 0 is fully transparent, 1 is fully opaque.	value: numeric																								
label	The label for the gauge is displayed as text, with optional units.	object																								
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>visible</td> <td>Indicates whether or not the label is visible. Default is true (visible).</td> <td>value: boolean</td> </tr> <tr> <td>size</td> <td>Font size to display label text. Default is 25.</td> <td>value: numeric</td> </tr> <tr> <td>offsetX</td> <td>Offset position on the x axis, relative to the middle of the gauge.</td> <td>value: numeric</td> </tr> <tr> <td>offsetY</td> <td>Offset position on the y axis, relative to the middle of the gauge.</td> <td>value: numeric</td> </tr> <tr> <td>color</td> <td>Color of the label.</td> <td>color</td> </tr> <tr> <td>units</td> <td>Any unit information to append to the value on the label.</td> <td>value: string</td> </tr> <tr> <td>maxDecimal</td> <td>Maximum number of digits after decimal to display in the label. If null, full value will display.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	visible	Indicates whether or not the label is visible. Default is true (visible).	value: boolean	size	Font size to display label text. Default is 25.	value: numeric	offsetX	Offset position on the x axis, relative to the middle of the gauge.	value: numeric	offsetY	Offset position on the y axis, relative to the middle of the gauge.	value: numeric	color	Color of the label.	color	units	Any unit information to append to the value on the label.	value: string	maxDecimal	Maximum number of digits after decimal to display in the label. If null, full value will display.	value: numeric	
Name	Description	Property Type																								
visible	Indicates whether or not the label is visible. Default is true (visible).	value: boolean																								
size	Font size to display label text. Default is 25.	value: numeric																								
offsetX	Offset position on the x axis, relative to the middle of the gauge.	value: numeric																								
offsetY	Offset position on the y axis, relative to the middle of the gauge.	value: numeric																								
color	Color of the label.	color																								
units	Any unit information to append to the value on the label.	value: string																								
maxDecimal	Maximum number of digits after decimal to display in the label. If null, full value will display.	value: numeric																								
animate	Whether needle should be animated in a sweeping motion when value changes. Default is false.	value: boolean																								
style	Sets a style for this gauge. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																								

Component Events

Perspective Component Events

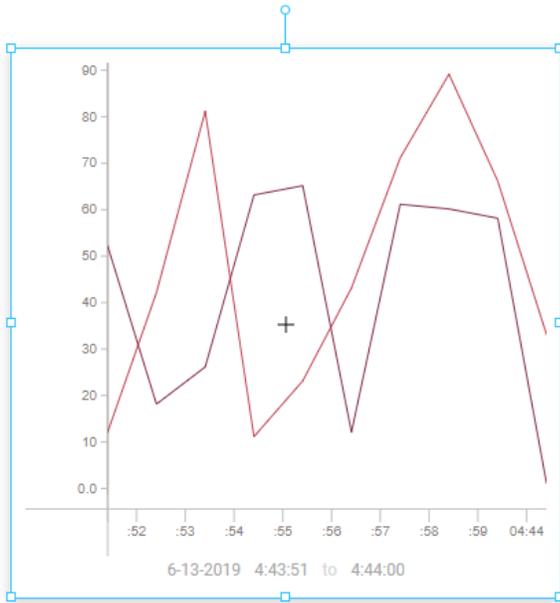
The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menu bar or by right clicking on the component.

Example



Property	Value
Value	32.4567
props.arc.width	15
props.arc.cornerRadius	25
props.arc.color	#FF8C00
props.arcBackground.opacity	0.4
props.label.units	psi
props.label.maxDecimal	1

Perspective - Time Series Chart



On this page ...

- [User Interaction](#)
- [Properties](#)
- [Component Events](#)
- [Examples](#)

Component Palette Icon:



The Time Series component provides an efficient way to visualize data from a variety of different data sources as chart data. Time series chart requires that the X axis of the chart represents time and the Y axis represents values. The Time Series Chart includes the following features:

- Zoom or pan in and out via mouse wheel interaction.
- X-Trace display showing data at the hovered time position.
- Multiple chart display types (Area, Bar, Line, and Scatter).
- Multiple Y axes with the ability to align to the left or right side of the chart.
- Multiple plots as well as multiple trends per plot.
- Baselines and markers.
- Custom axes.
- Time range showing the overall range of the data being displayed in the chart
- Simple display customization for the axes, different trend display types, baselines, and markers.
- Label and Title properties have their own dedicate styling properties, such as color and size.

By default, the charts contain example data, but typically a [tag history binding](#) or [named query binding](#) will be used to feed data to the charts.

The [Chart Range Selector](#) provides a complement to this chart. The Time Series Chart and Chart Range Selector components are most powerful when paired together.

The following feature is new in Ignition version **8.1.2**
[Click here](#) to check out the other new features

The Time Series Chart component has four pre-configured [variants](#):

- Line chart - Default layout with appearance of a line chart.
- Area chart - Layout set up as an area chart.
- Bar chart - Layout set up as a bar chart.
- Scatter chart - Layout set up as a scatter chart.

User Interaction

Interaction	Description
-------------	-------------

Zoom	<p>The user can zoom in and out on the Time Series chart. When zoomed to any level past its base time range, the Time Series chart will display a zoom reset  icon in the upper right corner. Click on the icon to return the chart to its base range. Note that the range will not refresh while zoomed in.</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p><u>This feature was changed in Ignition version 8.1.17:</u></p> </div> <p>When pen data sources change, the zoom level will be reset to its base time range.</p>
Pan	<p>The user can pan across the Time Series chart. When panned past its base time range, the Time Series chart will display a pan reset  icon in the upper right corner. Click on the icon to return the chart to its base range. Note that the range will not refresh while panning.</p>
Pinch Zoom	<p>On a mobile device, the user can pinch-zoom the Time Series chart. Zooms must originate from within the boundaries of the displaying chart data.</p>
Tracker Position	<p>On a mobile device, the user can move the tracker position on the Time Series chart via touch.</p>

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description						
enablePanZoom	Allows the chart to be panned and zoomed. The chart cannot be zoomed out past its base range.						
autoGenerateSeriesNames	<div style="border: 1px solid #f96; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.31 Click here to check out the other new features</p> </div> <p>If true, auto generates series names if no name is provided. This helps prevent the series name from resetting to the default value when the s</p>						
series	<p>A list of series entries used as the base data source for the chart display. Each series will be a new line drawn on the chart. At a minimum, the display data. With no other configuration provided, an initial display will be created from the series data (required). Each series entry requires</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>The name of the series. A name must be provided or one will be added. This will also be used as the label of the auto-generated</td> </tr> <tr> <td>data</td> <td> <p>Data can be an object containing a time entry and value entries (all must be numbers) (required).</p> <p>Each value entry must be labeled with the column name to which it corresponds. Data can also be an array containing value entry timestamp (which must be the first value) and one or more values that were captured at that time. Finally, data can also be in the bound to a Tag History binding to display either realtime data, or historical data (via start and end dates).</p> <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> <p>Note: When using a dataset, the Time column should be the first column.</p> </div> </td> </tr> </tbody> </table>	Name	Description	name	The name of the series. A name must be provided or one will be added. This will also be used as the label of the auto-generated	data	<p>Data can be an object containing a time entry and value entries (all must be numbers) (required).</p> <p>Each value entry must be labeled with the column name to which it corresponds. Data can also be an array containing value entry timestamp (which must be the first value) and one or more values that were captured at that time. Finally, data can also be in the bound to a Tag History binding to display either realtime data, or historical data (via start and end dates).</p> <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> <p>Note: When using a dataset, the Time column should be the first column.</p> </div>
Name	Description						
name	The name of the series. A name must be provided or one will be added. This will also be used as the label of the auto-generated						
data	<p>Data can be an object containing a time entry and value entries (all must be numbers) (required).</p> <p>Each value entry must be labeled with the column name to which it corresponds. Data can also be an array containing value entry timestamp (which must be the first value) and one or more values that were captured at that time. Finally, data can also be in the bound to a Tag History binding to display either realtime data, or historical data (via start and end dates).</p> <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> <p>Note: When using a dataset, the Time column should be the first column.</p> </div>						
plots	<p>A list of plots (subplots) for the chart. At least one entry is required. Plot entries contain properties that allow much finer control over the way t own row in the component (plots always stretch to fill the width of the chart). A plot contains the following properties:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> </tbody> </table>	Name	Description				
Name	Description						

relative Weight

The following feature is new in Ignition version **8.1.23**
[Click here](#) to check out the other new features

Relative weight sets the amount of height space plots take up in a Time Series Chart when multiple plots are included. By default, Customizing the value adjusts the plot height to the corresponding ratio of available space in the chart.

trends

If specified, a trend will become the display mechanism for the plot in which it resides. It describes the way that the data should be displayed. It is a `Line` display type that is provided when building from series data. Multiple trends will be built on top of each other in the same plot; they share a common axis (either custom created, or generated from `series` data).

Note: When making changes to a setting under trends, an axis must also be defined under `plots.axes`

A trend contains the following properties:

Name	Description																		
visible	If specified, this is the visible state of the trend. If not visible, the trend data will be hidden, but the time range of the trend will still be used to determine the time range of the plot.																		
type	The type of chart to create (required). Options are area, bar, line, and scatter.																		
series	The series used to feed data to this trend (required).																		
interpolation	The type of curve that should be used to draw the line portion of the chart. Options are: <code>curveBasis</code> , <code>curveBasisOpen</code> , <code>curveCatmullRom</code> , <code>curveCatmullRomOpen</code> , <code>curveLinear</code> , <code>curveMonotoneX</code> , <code>curveMonotoneY</code> , <code>curveNatural</code> , <code>curveStep</code> . More information on the interpolation methods above can be found in D3's documentation .																		
breakLine	This property will be available when a trend of type line is being used. If true, the line will be broken on either side of null values and the adjoining points are connected.																		
stack	This property will be available when a trend of type area or bar is being used. If true, the multiple columns of the chart will be stacked on top of each other.																		
axis	If specified, the name of an axis that is described in the <code>axes</code> property of the plot to which this trend belongs. This axis must be defined in the <code>plots.axes</code> property.																		
radius	This property will be available when a trend of type scatter is being used. A number specifying the radius (in pixels) of the markers.																		
columns	If provided, only the columns in this list will be shown. Any style properties provided here will also override any existing style properties for the columns. The state value contains the following properties: <table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>key</td><td>This needs to match a column name from the series to which this trend is bound (required). Once in place, it will be used to override the styles provided in the <code>defaultStyles</code> property.</td></tr><tr><td>color</td><td>If provided, this value will override any previous color values for the column (both stroke and fill). Can be chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td></tr><tr><td>styles</td><td>If provided, the styles for the state values listed here will override any previous state values. The state value contains the following properties:<table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>normal</td><td>Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.</td></tr><tr><td>highlighted</td><td>Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.</td></tr><tr><td>selected</td><td>Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.</td></tr><tr><td>muted</td><td>Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.</td></tr></tbody></table></td></tr></tbody></table>	Name	Description	key	This needs to match a column name from the series to which this trend is bound (required). Once in place, it will be used to override the styles provided in the <code>defaultStyles</code> property.	color	If provided, this value will override any previous color values for the column (both stroke and fill). Can be chosen from color palette, or entered as RGB or HSL value. See Color Selector .	styles	If provided, the styles for the state values listed here will override any previous state values. The state value contains the following properties: <table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>normal</td><td>Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.</td></tr><tr><td>highlighted</td><td>Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.</td></tr><tr><td>selected</td><td>Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.</td></tr><tr><td>muted</td><td>Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.</td></tr></tbody></table>	Name	Description	normal	Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.	highlighted	Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.	selected	Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.	muted	Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.
Name	Description																		
key	This needs to match a column name from the series to which this trend is bound (required). Once in place, it will be used to override the styles provided in the <code>defaultStyles</code> property.																		
color	If provided, this value will override any previous color values for the column (both stroke and fill). Can be chosen from color palette, or entered as RGB or HSL value. See Color Selector .																		
styles	If provided, the styles for the state values listed here will override any previous state values. The state value contains the following properties: <table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>normal</td><td>Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.</td></tr><tr><td>highlighted</td><td>Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.</td></tr><tr><td>selected</td><td>Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.</td></tr><tr><td>muted</td><td>Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.</td></tr></tbody></table>	Name	Description	normal	Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.	highlighted	Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.	selected	Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.	muted	Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.								
Name	Description																		
normal	Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.																		
highlighted	Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.																		
selected	Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.																		
muted	Takes the same configuration options as the same named state value under the <code>defaultStyles</code> property.																		

baselines	If specified, a line will be drawn on the trend based upon a given type of functionality. Options as follows:		
	Name	Description	
	visible	Whether the baseline is visible or not.	
	function	The type of baseline that will be drawn (required). Options are min, max, avg, ucl (upper control limit), and	
	column	The column against which the baseline should be calculated. If not specified, the first column in the series bound will be used.	
	axis	The axis against which the baseline should be calculated. The trend must be bound to an axis for this to	
	color	The color of the line. Can be chosen from color wheel, chosen from color palette, or entered as RGB or H-Selector.	
	width	The width of the line, in pixels.	
	opacity	The opacity of the line, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	
	dashArray	The dashed appearance (SVG dashed array) of the line. The pattern of dashes and gaps used to paint the space separated lengths (in pixels) and percentages (percentage of the total stroke length) that specify the dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number. Thus, "5,3,2" is equivalent to "5,3,2,5,3,2".	
	label	The configuration used for the label drawn on the baseline.	
		Name	Description
		text	The label text.
position		The position of the label relative to the baseline.	
font		The font style for the label.	
		Name	Description
	color	The color of the label text.	value: string
size	The font size, in pixels, of the label text.	value: numeric	
style	Custom CSS styles to apply to the baseline label. Any style that applies to an SVG text element can be used. See also style options .		
style	Custom CSS styles to apply to the line portion of the baseline. Any style that applies to an SVG line element can be used. See also style options .		

axes A list of axis entries for that can be used for the plot. These can be used to override the Y Axis that is generated based on data in

Note: When making changes to a setting under axes, a trend must also be defined under `plots.trends`

An axis entry has the following properties:

Name	Description
name	The name of the axis (required). Also used as the display label of the axis.
min	The minimum value of the axis (required). If no value is specified, auto range will be used. A minimum value will be calculated based on the data.
max	The maximum value of the axis (required). The maximum range value of the axis. If no value is specified, auto range will be used. A maximum value will be calculated based on the data.

alignme nt	The side of the trend upon which the axis should be presented (required).										
width	The width of the axis, in pixels (required).										
label	The configuration of the Y axis label.										
	Name	Description									
	visible	Whether or not the label is visible.									
	text	The text for the label.									
	offset	Offset the Y axis label from its default position. This allows you to fine tune the label location, which may vary on the scale and how much room the tick labels take up. This may be positive or negative.									
	font	The settings for the label's font.									
		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>The color of the label text.</td> <td>value: string</td> </tr> <tr> <td>size</td> <td>The font size, in pixels, of the label text.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	The color of the label text.	value: string	size	The font size, in pixels, of the label text.	value: numeric
Name	Description	Property Type									
color	The color of the label text.	value: string									
size	The font size, in pixels, of the label text.	value: numeric									
	style	Custom CSS styles to apply to the Y axis label. Any style that applies to an SVG <code>text</code> element can be used.									

tick

The configuration for the ticks drawn on the axis.

Name	Description
color	The color of the ticks. Can be chosen from color wheel, chosen from color palette, or entered as RGB or Selector .
count	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;">The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</div> <p>Number of ticks to display on the Y axis. Default value is Auto. When the tick count is Auto, Y axis ticks w based on the height of the chart, and the tick spacing is adjusted around whole numbers.</p>

label

Name	Description									
format	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;">The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</div> <p>Sets the numeric format for the tick label. The value must be a valid D3 Format value. Default Auto. When the tick label format is Auto, the D3 format <code>~f</code> is used. This creates a fixed point trim any insignificant trailing zeroes.</p> <div style="border: 1px solid gray; padding: 5px; margin-bottom: 10px;"><u>This feature was changed in Ignition version 8.1.13:</u></div> <p>The Y axis now supports numeric locale formatting. Numeric values are automatically format on the session locale and the <code>tick.label.format</code> property.</p>									
font	The font style for the label. <table border="1" style="margin-top: 10px;"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>color</td><td>The color of the label text.</td><td>value: string</td></tr><tr><td>size</td><td>The font size, in pixels, of the label text.</td><td>value: numeric</td></tr></tbody></table>	Name	Description	Property Type	color	The color of the label text.	value: string	size	The font size, in pixels, of the label text.	value: numeric
Name	Description	Property Type								
color	The color of the label text.	value: string								
size	The font size, in pixels, of the label text.	value: numeric								
style	Custom CSS styles to apply to the baseline label. Any style that applies to an SVG <code>text</code> element can be used. See also style options .									
style	Custom CSS styles to apply to the ticks. Any style that applies to an SVG <code>text</code> element can be used. See									

grid

The following feature is new in Ignition version **8.1.16**
[Click here](#) to check out the other new features

Configuration for gridlines to display on this axis.

Name	Description
visible	Visible state of the gridlines. Gridlines are shown only for axes that connect directly to the chart. Any sate their tick configurations instead of gridlines.
color	Color of the gridlines.
opacity	Opacity of the gridlines.
dashArray	Dashed appearance of the gridlines.
style	Style for the gridlines. Full menu of style options is available. You can also specify a style class .

markers Settings for the markers, a list of visual indicators that can be added to the plot (optional). These are meant to draw emphasis to t only available option.

Name	Description
visible	Indicates whether or not the marker is visible .
value	The numeric value represented by the marker.
type	The type of marker. Currently the only option is line.
axis	The axis against which the marker should be drawn (required).

line The configuration for the line portion of the marker.

Name	Description
width	The width of the marker line, in pixels.
color	The color of the marker line. Can be chosen from color wheel, chosen from color palette, or entered as R or Selector.
opacity	The opacity of the marker line, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.
dashArray	The pattern of dashes and gaps (SVG dashed array) used to paint the marker line. It's a list of space sep and percentages (percentage of the total stroke length) that specify the lengths of alternating dashes and of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5 3 2" i 2".

label The configuration for marker's label.

Name	Description	P	T
text	The marker label text.		
position	The position of the label relative to the line.		
font	The font settings for the label.		
color	The color of the label text. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	va	str
size	The font size, in pixels, of the label text.	va	nu
style	Custom CSS styles to apply to the marker label. Any style that applies to an SVG <code>text</code> element can be used. See also style options .	ob	
style	Custom CSS styles to apply to the marker line. Any style that applies to an SVG <code>line</code> elem be used. See also style options .		

style Custom CSS styles to apply to the marker line. Any style that applies to an SVG line element can be use

title

Settings for the title of the chart.

Name	Description						
visible	Indicates whether or not the title is visible. Default is false.						
text	Text for the title of the chart.						
height	The vertical space taken up by the title.						
font	Title font configuration. Options as follows: <table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>color</td><td>Color of the title text. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. S</td></tr><tr><td>size</td><td>The font size, in pixels, of the title text.</td></tr></tbody></table>	Name	Description	color	Color of the title text. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. S	size	The font size, in pixels, of the title text.
Name	Description						
color	Color of the title text. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. S						
size	The font size, in pixels, of the title text.						
style	Sets a style for the title. Full menu of style options is available for text, background, margin and padding, border, shape and misc						

timeAxis

This property provides settings for the X Axis. Note that multiple plots share the same axis. (required)

Name	Description																									
visible	The visible state of the the axis.																									
tickCount	The number of ticks on the axis (as a multiple of 2, 5, or 10). <div style="border: 1px solid orange; padding: 5px; margin: 5px 0;">The following feature is new in Ignition version 8.1.22 Click here to check out the other new features</div> Setting the tickCount to 0 will automatically scale the chart's X Axis based on the zoom level, panning, and scrolling.																									
height	The height of the axis.																									
color	The color of the axis. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color																									
tick	The configuration of the ticks on the axis. <table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>color</td><td>The color of the ticks. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. S</td></tr><tr><td>label</td><td>The configuration of the label drawn on the tick.<table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>angled</td><td>If set to true, the tick labels will be angled rather than horizontal.</td></tr><tr><td>format</td><td>The date/time format displayed by each tick using a MomentJS date string (https://momentjs.com/). Default is "Auto", where the property attempts to figure out the best format. For a listing of suggested formats, refer to https://momentjs.com/docs/#/parsing/string-format/</td></tr><tr><td>font</td><td>The settings for the label's font.<table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>color</td><td>Color of the label text.</td><td>value: string</td></tr><tr><td>size</td><td>The font size, in pixels, of the label text.</td><td>value: numeric</td></tr></tbody></table></td></tr><tr><td>style</td><td>Custom CSS styles to apply to the tick labels. Any style that applies to an SVG text element can be used options.</td></tr></tbody></table></td></tr></tbody></table>	Name	Description	color	The color of the ticks. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. S	label	The configuration of the label drawn on the tick. <table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>angled</td><td>If set to true, the tick labels will be angled rather than horizontal.</td></tr><tr><td>format</td><td>The date/time format displayed by each tick using a MomentJS date string (https://momentjs.com/). Default is "Auto", where the property attempts to figure out the best format. For a listing of suggested formats, refer to https://momentjs.com/docs/#/parsing/string-format/</td></tr><tr><td>font</td><td>The settings for the label's font.<table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>color</td><td>Color of the label text.</td><td>value: string</td></tr><tr><td>size</td><td>The font size, in pixels, of the label text.</td><td>value: numeric</td></tr></tbody></table></td></tr><tr><td>style</td><td>Custom CSS styles to apply to the tick labels. Any style that applies to an SVG text element can be used options.</td></tr></tbody></table>	Name	Description	angled	If set to true, the tick labels will be angled rather than horizontal.	format	The date/time format displayed by each tick using a MomentJS date string (https://momentjs.com/). Default is "Auto", where the property attempts to figure out the best format. For a listing of suggested formats, refer to https://momentjs.com/docs/#/parsing/string-format/	font	The settings for the label's font. <table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>color</td><td>Color of the label text.</td><td>value: string</td></tr><tr><td>size</td><td>The font size, in pixels, of the label text.</td><td>value: numeric</td></tr></tbody></table>	Name	Description	Property Type	color	Color of the label text.	value: string	size	The font size, in pixels, of the label text.	value: numeric	style	Custom CSS styles to apply to the tick labels. Any style that applies to an SVG text element can be used options .
Name	Description																									
color	The color of the ticks. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. S																									
label	The configuration of the label drawn on the tick. <table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>angled</td><td>If set to true, the tick labels will be angled rather than horizontal.</td></tr><tr><td>format</td><td>The date/time format displayed by each tick using a MomentJS date string (https://momentjs.com/). Default is "Auto", where the property attempts to figure out the best format. For a listing of suggested formats, refer to https://momentjs.com/docs/#/parsing/string-format/</td></tr><tr><td>font</td><td>The settings for the label's font.<table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>color</td><td>Color of the label text.</td><td>value: string</td></tr><tr><td>size</td><td>The font size, in pixels, of the label text.</td><td>value: numeric</td></tr></tbody></table></td></tr><tr><td>style</td><td>Custom CSS styles to apply to the tick labels. Any style that applies to an SVG text element can be used options.</td></tr></tbody></table>	Name	Description	angled	If set to true, the tick labels will be angled rather than horizontal.	format	The date/time format displayed by each tick using a MomentJS date string (https://momentjs.com/). Default is "Auto", where the property attempts to figure out the best format. For a listing of suggested formats, refer to https://momentjs.com/docs/#/parsing/string-format/	font	The settings for the label's font. <table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>color</td><td>Color of the label text.</td><td>value: string</td></tr><tr><td>size</td><td>The font size, in pixels, of the label text.</td><td>value: numeric</td></tr></tbody></table>	Name	Description	Property Type	color	Color of the label text.	value: string	size	The font size, in pixels, of the label text.	value: numeric	style	Custom CSS styles to apply to the tick labels. Any style that applies to an SVG text element can be used options .						
Name	Description																									
angled	If set to true, the tick labels will be angled rather than horizontal.																									
format	The date/time format displayed by each tick using a MomentJS date string (https://momentjs.com/). Default is "Auto", where the property attempts to figure out the best format. For a listing of suggested formats, refer to https://momentjs.com/docs/#/parsing/string-format/																									
font	The settings for the label's font. <table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>color</td><td>Color of the label text.</td><td>value: string</td></tr><tr><td>size</td><td>The font size, in pixels, of the label text.</td><td>value: numeric</td></tr></tbody></table>	Name	Description	Property Type	color	Color of the label text.	value: string	size	The font size, in pixels, of the label text.	value: numeric																
Name	Description	Property Type																								
color	Color of the label text.	value: string																								
size	The font size, in pixels, of the label text.	value: numeric																								
style	Custom CSS styles to apply to the tick labels. Any style that applies to an SVG text element can be used options .																									

grid

The following feature is new in Ignition version **8.1.16**
[Click here](#) to check out the other new features

Configuration for gridlines to display on this axis.

Name	Description
visible	Visible state of the gridlines. Gridlines are shown only for axes that connect directly to the chart. Any satellite axes will not have gridlines.
color	Color of the gridlines.
opacity	Opacity of the gridlines.
dashArray	Dashed appearance of the gridlines.
style	Style for the gridlines. Full menu of style options is available. You can also specify a style class .

style A style object containing properties which are applied to the horizontal line of the axis. Any property that would apply to an SVG li

timeRange

An object describing the presentation of the time range display below the chart. The following properties are available:

Name	Description	Property Type
visible	The visible state of the time range display as a whole.	value: boolean
dateFormat	The date format of the range using a MomentJS date string.	value: string
timeFormat	The time format of the range using a MomentJS date string.	value: string

xTrace

Configuration to build the x-trace display when hovering over the chart.

The following feature is new in Ignition version **8.1.13**
[Click here](#) to check out the other new features

The X Trace display now supports numeric locale formatting. Numeric values are automatically formatted based on the session locale and the

Name	Description
value	A read only timestamp representing the current x-trace position. If there is no active x-trace position, this value will be an empty string.
visible	The visible state of the x-trace display.

infoBox

Configuration to build the box portion of the x-trace display.

Name	Description										
visible	The visible state of the box.										
showTime	Whether to display the timestamp of the current X Trace value above the info box.										
width	Width of the info box, in pixels.										
dateFormat	The date format of the xtrace date/time display using a Momentjs date string (https://momentjs.com/). Options are: [7 19-07-18], [Jul 18th 19], [Jul 18th 2019], or none.										
timeFormat	The time format of the xtrace date/time display using a MomentJS time string (https://momentjs.com/). Options are: 06 AM], 24 hour [08:41:06], 24 hour w/milliseconds [08:41:06:269], Unix Millisecond Timestamp [1563464737269], U										
dataFormat	A NumeralJS value used to format the data found at the current timestamp of the X Trace display. See numeral.js for are: number [1,000.12], integer [1,200], four decimal precision [1.1200], percent [10.12%], scientific [1.01E+03], account currency [\$1,000.12], currency (rounded) [\$1,012], duration [24:01:00], abbreviation [1.2k], or ordinal [100th].										
stroke	A configuration object describing the properties that will be applied to the stroke of the box display. <table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>color</td><td>The color of the box stroke. Can be chosen from color wheel, chosen from color palette, or entered as Color Selector.</td></tr><tr><td>width</td><td>The width of the box stroke, in pixels.</td></tr><tr><td>opacity</td><td>The opacity of the box stroke, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td></tr><tr><td>dashArray</td><td>The dashed appearance (SVG dashed array) of the box stroke. It's a list of space separated lengths (in percentage of the total stroke length) that specify the lengths of alternating dashes and gaps. If an odd number is provided, then the list of values is repeated to yield an even number of values. Thus, "5,3,2" is equivalent</td></tr></tbody></table>	Name	Description	color	The color of the box stroke. Can be chosen from color wheel, chosen from color palette, or entered as Color Selector .	width	The width of the box stroke, in pixels.	opacity	The opacity of the box stroke, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	dashArray	The dashed appearance (SVG dashed array) of the box stroke. It's a list of space separated lengths (in percentage of the total stroke length) that specify the lengths of alternating dashes and gaps. If an odd number is provided, then the list of values is repeated to yield an even number of values. Thus, "5,3,2" is equivalent
Name	Description										
color	The color of the box stroke. Can be chosen from color wheel, chosen from color palette, or entered as Color Selector .										
width	The width of the box stroke, in pixels.										
opacity	The opacity of the box stroke, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.										
dashArray	The dashed appearance (SVG dashed array) of the box stroke. It's a list of space separated lengths (in percentage of the total stroke length) that specify the lengths of alternating dashes and gaps. If an odd number is provided, then the list of values is repeated to yield an even number of values. Thus, "5,3,2" is equivalent										
fill	A configuration object describing the properties that will be applied to the fill of the box display. <table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>color</td><td>The color of the box fill. Can be chosen from color wheel, chosen from color palette, or entered as RGB color Selector.</td></tr><tr><td>opacity</td><td>The opacity of the box fill, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td></tr></tbody></table>	Name	Description	color	The color of the box fill. Can be chosen from color wheel, chosen from color palette, or entered as RGB color Selector .	opacity	The opacity of the box fill, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.				
Name	Description										
color	The color of the box fill. Can be chosen from color wheel, chosen from color palette, or entered as RGB color Selector .										
opacity	The opacity of the box fill, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.										
font	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"><p>The following feature is new in Ignition version 8.1.17 Click here to check out the other new features</p></div> <p>A configuration object describing the properties that will be applied to the font of text in the infoBox.</p> <table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>color</td><td>The text color of the info box label and datetime text.</td></tr><tr><td>size</td><td>The font size of the info box label and datetime text.</td></tr><tr><td>style</td><td>Custom CSS styles to apply to the info box text. Any style that applies to an SVG <code>line</code> element can be used. See also style options.</td></tr></tbody></table>	Name	Description	color	The text color of the info box label and datetime text.	size	The font size of the info box label and datetime text.	style	Custom CSS styles to apply to the info box text. Any style that applies to an SVG <code>line</code> element can be used. See also style options .		
Name	Description										
color	The text color of the info box label and datetime text.										
size	The font size of the info box label and datetime text.										
style	Custom CSS styles to apply to the info box text. Any style that applies to an SVG <code>line</code> element can be used. See also style options .										
style	Custom CSS styles to apply to the info box. Any style that applies to an SVG <code>line</code> element can be used. See also style options .										

line	Configuration to build the vertical line portion of the x-trace display.	
	Name	Description
	visible	The visible state of the line.
	color	The color of the line. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. S
	width	The width of the line, in pixels.
	opacity	The opacity of the line, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.
	dashArray	The dashed appearance (SVG dashed array) of the line. It's a list of space separated lengths (in pixels) and percenta that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values values. Thus, "5,3,2" is equivalent to "5,3,2,5,3,2".
	style	Custom CSS styles to apply to the line. Any style that applies to an SVG line element can be used. See also style opt

legend	Configuration for the display and position of the legend for the Time Series Chart.	
	Name	Description
	visible	The visible state of the legend. Default is false.
	position	The position of the legend. Options are top, right, bottom, or left.
	style	A style object containing properties which are applied to the legend. See also style options . You can also specify a style class .

defaultStyles	An object providing style settings to the chart trends as a whole.	
	Name	Description
	colorScheme	Specifies a Color Brewer color scheme to use on the series. See ColorBrewer2.org for available color schemes.
	colors	A list of colors to apply to the columns (in order) for each trend. If these values are provided, they will override the value provided

normal An object providing style configuration for the “normal” state (no user interaction) of a column, or data entry, in a trend. Any color values are defined by the colorScheme or colors properties.

Name	Description								
stroke	A configuration object describing the properties that will be applied to the stroke of the trend type being displayed (if applicable). The line, and area trend types will have these styles applied to them. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>width</td> <td>The width of the trend stroke, in pixels.</td> </tr> <tr> <td>opacity</td> <td>The opacity of the trend stroke, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> </tr> <tr> <td>dashArray</td> <td>The dashed appearance (SVG dashed array) of the trend stroke. It's a list of space separated lengths (in percentage of the total stroke length) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5,3,2" is equivalent to "5,3,2,5,3,2".</td> </tr> </tbody> </table>	Name	Description	width	The width of the trend stroke, in pixels.	opacity	The opacity of the trend stroke, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	dashArray	The dashed appearance (SVG dashed array) of the trend stroke. It's a list of space separated lengths (in percentage of the total stroke length) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5,3,2" is equivalent to "5,3,2,5,3,2".
Name	Description								
width	The width of the trend stroke, in pixels.								
opacity	The opacity of the trend stroke, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.								
dashArray	The dashed appearance (SVG dashed array) of the trend stroke. It's a list of space separated lengths (in percentage of the total stroke length) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5,3,2" is equivalent to "5,3,2,5,3,2".								
fill	A configuration object describing the properties that will be applied to the fill of the trend type being displayed (if applicable). The area, and box, and scatter trend types will have these styles applied to them. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>opacity</td> <td>The opacity of the trend fill, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	opacity	The opacity of the trend fill, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric		
Name	Description	Property Type							
opacity	The opacity of the trend fill, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric							

highlighted An object providing style configuration for the “highlighted” state (mouse hover) of a column, or data entry, in a trend. Any color values are defined by the colorScheme or colors properties. The highlighted property uses the same configuration properties as the 'normal' property.

Name	Description						
stroke	A configuration object describing the properties that will be applied to the stroke for the “highlighted” state (mouse hover). The line, and area trend types will have these styles applied to them. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>opacity</td> <td>The opacity of the trend stroke, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	opacity	The opacity of the trend stroke, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric
Name	Description	Property Type					
opacity	The opacity of the trend stroke, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric					
fill	A configuration object describing the properties that will be applied to the stroke for the “highlighted” state (mouse hover). The area, and box, and scatter trend types will have these styles applied to them. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>opacity</td> <td>The opacity of the trend fill, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	opacity	The opacity of the trend fill, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric
Name	Description	Property Type					
opacity	The opacity of the trend fill, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric					

selected	<p>An object providing style configuration for the “selected” state (mouse click) of a column, or data entry, in a trend. Any color value: the colorScheme or colors properties.</p> <table border="1" data-bbox="358 205 1500 751"> <thead> <tr> <th data-bbox="358 205 456 281">Name</th> <th data-bbox="456 205 1382 281">Description</th> <th data-bbox="1382 205 1500 281"></th> </tr> </thead> <tbody> <tr> <td data-bbox="358 281 456 520">stroke</td> <td data-bbox="456 281 1382 520">A configuration object describing the properties that will be applied to the stroke for the “selected” state (mouse hover). The line, and area trend types will have these styles applied to them.</td> <td data-bbox="1382 281 1500 520"></td> </tr> <tr> <td data-bbox="358 520 456 751"></td> <td data-bbox="456 520 1382 751"> <table border="1" data-bbox="467 359 1500 457"> <thead> <tr> <th data-bbox="467 359 565 405">Name</th> <th data-bbox="565 359 1382 405">Description</th> <th data-bbox="1382 359 1500 405">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="467 405 565 457">opacity</td> <td data-bbox="565 405 1382 457">The opacity of the trend stroke, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td data-bbox="1382 405 1500 457">value: numeric</td> </tr> </tbody> </table> </td> <td data-bbox="1382 520 1500 751"></td> </tr> <tr> <td data-bbox="358 520 456 751">fill</td> <td data-bbox="456 520 1382 751">A configuration object describing the properties that will be applied to the stroke for the “selected” state (mouse hover). The area, and box, and scatter trend types will have these styles applied to them.</td> <td data-bbox="1382 520 1500 751"></td> </tr> <tr> <td data-bbox="358 520 456 751"></td> <td data-bbox="456 520 1382 751"> <table border="1" data-bbox="467 590 1500 688"> <thead> <tr> <th data-bbox="467 590 565 636">Name</th> <th data-bbox="565 590 1382 636">Description</th> <th data-bbox="1382 590 1500 636">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="467 636 565 688">opacity</td> <td data-bbox="565 636 1382 688">The opacity of the trend fill, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td data-bbox="1382 636 1500 688">value: numeric</td> </tr> </tbody> </table> </td> <td data-bbox="1382 520 1500 751"></td> </tr> </tbody> </table>	Name	Description		stroke	A configuration object describing the properties that will be applied to the stroke for the “selected” state (mouse hover). The line, and area trend types will have these styles applied to them.			<table border="1" data-bbox="467 359 1500 457"> <thead> <tr> <th data-bbox="467 359 565 405">Name</th> <th data-bbox="565 359 1382 405">Description</th> <th data-bbox="1382 359 1500 405">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="467 405 565 457">opacity</td> <td data-bbox="565 405 1382 457">The opacity of the trend stroke, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td data-bbox="1382 405 1500 457">value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	opacity	The opacity of the trend stroke, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric		fill	A configuration object describing the properties that will be applied to the stroke for the “selected” state (mouse hover). The area, and box, and scatter trend types will have these styles applied to them.			<table border="1" data-bbox="467 590 1500 688"> <thead> <tr> <th data-bbox="467 590 565 636">Name</th> <th data-bbox="565 590 1382 636">Description</th> <th data-bbox="1382 590 1500 636">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="467 636 565 688">opacity</td> <td data-bbox="565 636 1382 688">The opacity of the trend fill, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td data-bbox="1382 636 1500 688">value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	opacity	The opacity of the trend fill, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric	
Name	Description																											
stroke	A configuration object describing the properties that will be applied to the stroke for the “selected” state (mouse hover). The line, and area trend types will have these styles applied to them.																											
	<table border="1" data-bbox="467 359 1500 457"> <thead> <tr> <th data-bbox="467 359 565 405">Name</th> <th data-bbox="565 359 1382 405">Description</th> <th data-bbox="1382 359 1500 405">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="467 405 565 457">opacity</td> <td data-bbox="565 405 1382 457">The opacity of the trend stroke, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td data-bbox="1382 405 1500 457">value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	opacity	The opacity of the trend stroke, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric																					
Name	Description	Property Type																										
opacity	The opacity of the trend stroke, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric																										
fill	A configuration object describing the properties that will be applied to the stroke for the “selected” state (mouse hover). The area, and box, and scatter trend types will have these styles applied to them.																											
	<table border="1" data-bbox="467 590 1500 688"> <thead> <tr> <th data-bbox="467 590 565 636">Name</th> <th data-bbox="565 590 1382 636">Description</th> <th data-bbox="1382 590 1500 636">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="467 636 565 688">opacity</td> <td data-bbox="565 636 1382 688">The opacity of the trend fill, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td data-bbox="1382 636 1500 688">value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	opacity	The opacity of the trend fill, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric																					
Name	Description	Property Type																										
opacity	The opacity of the trend fill, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric																										
muted	<p>An object providing style configuration for the “muted” state (non selected) of a column, or data entry, in a trend. Any color values: the colorScheme or colors properties.</p> <table border="1" data-bbox="358 829 1500 1381"> <thead> <tr> <th data-bbox="358 829 456 905">Name</th> <th data-bbox="456 829 1382 905">Description</th> <th data-bbox="1382 829 1500 905"></th> </tr> </thead> <tbody> <tr> <td data-bbox="358 905 456 1144">stroke</td> <td data-bbox="456 905 1382 1144">A configuration object describing the properties that will be applied to the stroke for the “muted” state (mouse hover). The line, and area trend types will have these styles applied to them.</td> <td data-bbox="1382 905 1500 1144"></td> </tr> <tr> <td data-bbox="358 1144 456 1381"></td> <td data-bbox="456 1144 1382 1381"> <table border="1" data-bbox="467 982 1500 1081"> <thead> <tr> <th data-bbox="467 982 565 1029">Name</th> <th data-bbox="565 982 1382 1029">Description</th> <th data-bbox="1382 982 1500 1029">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="467 1029 565 1081">opacity</td> <td data-bbox="565 1029 1382 1081">The opacity of the trend stroke, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td data-bbox="1382 1029 1500 1081">value: numeric</td> </tr> </tbody> </table> </td> <td data-bbox="1382 1144 1500 1381"></td> </tr> <tr> <td data-bbox="358 1144 456 1381">fill</td> <td data-bbox="456 1144 1382 1381">A configuration object describing the properties that will be applied to the stroke for the “muted” state (mouse hover). The area, and box, and scatter trend types will have these styles applied to them.</td> <td data-bbox="1382 1144 1500 1381"></td> </tr> <tr> <td data-bbox="358 1144 456 1381"></td> <td data-bbox="456 1144 1382 1381"> <table border="1" data-bbox="467 1213 1500 1312"> <thead> <tr> <th data-bbox="467 1213 565 1260">Name</th> <th data-bbox="565 1213 1382 1260">Description</th> <th data-bbox="1382 1213 1500 1260">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="467 1260 565 1312">opacity</td> <td data-bbox="565 1260 1382 1312">The opacity of the trend fill, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td data-bbox="1382 1260 1500 1312">value: numeric</td> </tr> </tbody> </table> </td> <td data-bbox="1382 1144 1500 1381"></td> </tr> </tbody> </table>	Name	Description		stroke	A configuration object describing the properties that will be applied to the stroke for the “muted” state (mouse hover). The line, and area trend types will have these styles applied to them.			<table border="1" data-bbox="467 982 1500 1081"> <thead> <tr> <th data-bbox="467 982 565 1029">Name</th> <th data-bbox="565 982 1382 1029">Description</th> <th data-bbox="1382 982 1500 1029">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="467 1029 565 1081">opacity</td> <td data-bbox="565 1029 1382 1081">The opacity of the trend stroke, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td data-bbox="1382 1029 1500 1081">value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	opacity	The opacity of the trend stroke, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric		fill	A configuration object describing the properties that will be applied to the stroke for the “muted” state (mouse hover). The area, and box, and scatter trend types will have these styles applied to them.			<table border="1" data-bbox="467 1213 1500 1312"> <thead> <tr> <th data-bbox="467 1213 565 1260">Name</th> <th data-bbox="565 1213 1382 1260">Description</th> <th data-bbox="1382 1213 1500 1260">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="467 1260 565 1312">opacity</td> <td data-bbox="565 1260 1382 1312">The opacity of the trend fill, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td data-bbox="1382 1260 1500 1312">value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	opacity	The opacity of the trend fill, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric	
Name	Description																											
stroke	A configuration object describing the properties that will be applied to the stroke for the “muted” state (mouse hover). The line, and area trend types will have these styles applied to them.																											
	<table border="1" data-bbox="467 982 1500 1081"> <thead> <tr> <th data-bbox="467 982 565 1029">Name</th> <th data-bbox="565 982 1382 1029">Description</th> <th data-bbox="1382 982 1500 1029">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="467 1029 565 1081">opacity</td> <td data-bbox="565 1029 1382 1081">The opacity of the trend stroke, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td data-bbox="1382 1029 1500 1081">value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	opacity	The opacity of the trend stroke, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric																					
Name	Description	Property Type																										
opacity	The opacity of the trend stroke, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric																										
fill	A configuration object describing the properties that will be applied to the stroke for the “muted” state (mouse hover). The area, and box, and scatter trend types will have these styles applied to them.																											
	<table border="1" data-bbox="467 1213 1500 1312"> <thead> <tr> <th data-bbox="467 1213 565 1260">Name</th> <th data-bbox="565 1213 1382 1260">Description</th> <th data-bbox="1382 1213 1500 1260">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="467 1260 565 1312">opacity</td> <td data-bbox="565 1260 1382 1312">The opacity of the trend fill, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td data-bbox="1382 1260 1500 1312">value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	opacity	The opacity of the trend fill, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric																					
Name	Description	Property Type																										
opacity	The opacity of the trend fill, ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric																										
style	Sets a style for this chart. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous.																											

Component Events

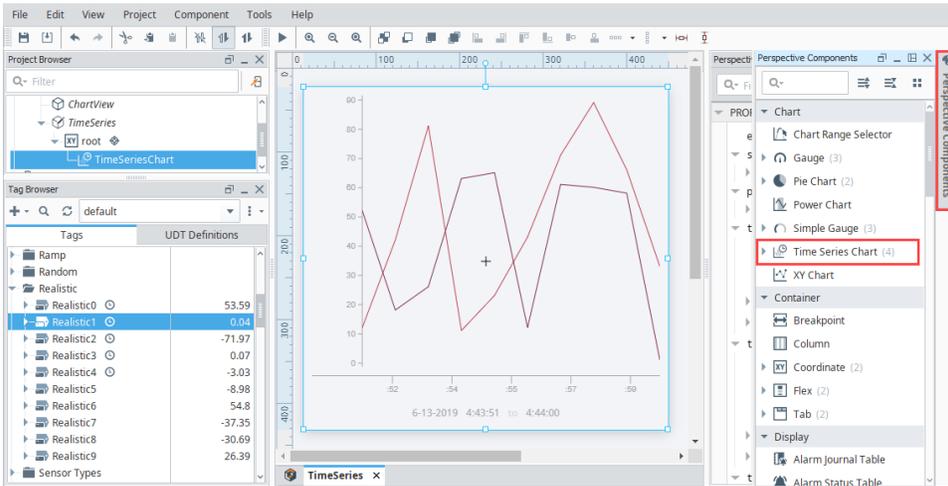
Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

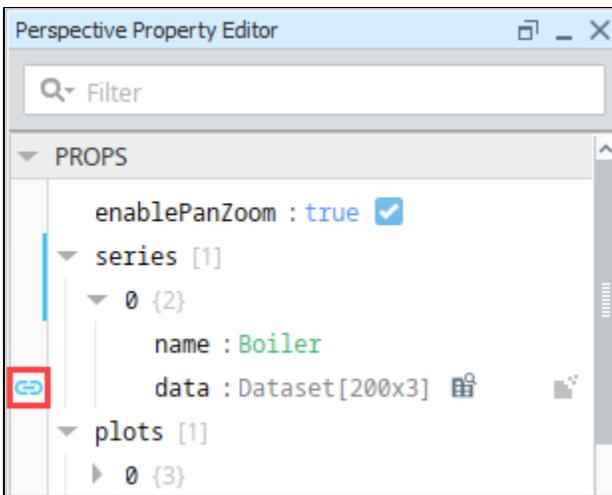
Examples

This example shows a Time Series Chart displaying Temperature and Pressure values for Tank 100.

1. Create a new view, and drag an **Time Series Chart** component on to your view.

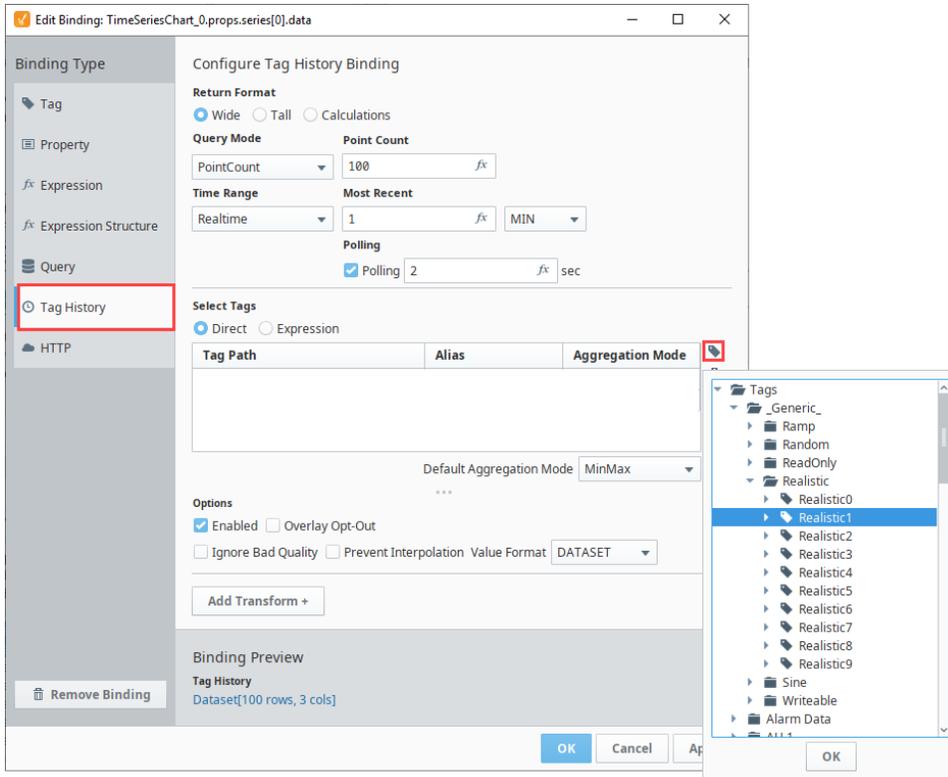


2. With the Time Series Chart selected, click on the chain link **Binding icon** under the **series > 0 > data** property.



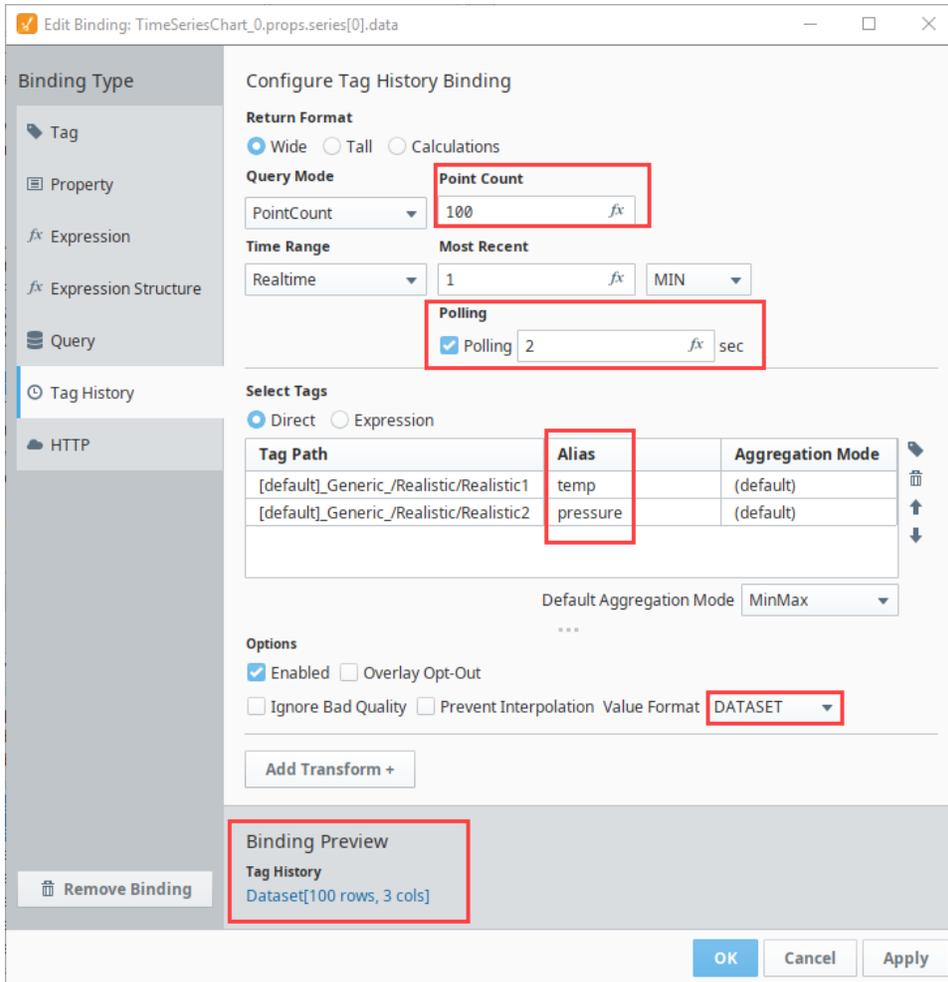
3. This will open the Edit Binding window. Select the **Tag History** binding type.

4. From the **Select Tags** section, click on the Tag Browse icon on the far right. Browse to select your Tag path. We selected **Realistic1** and **Realistic2** from the Generic > Realistic folder.



5. Now let's configure the following Tag History binding settings:
 - a. Double click in the Alias column for each Tag to add an Alias
 - i. Add the **Alias** for **Realistic1** as **temp**.
 - ii. Add the **Alias** for **Realistic2** as **pressure**.
 - b. Set the **Point** count to **100**.
 - c. Under **Most Recent**, select **MIN** from the dropdown.
 - d. Set **Polling** to **2**.
 - e. The Tag History binding returns a **Dataset**. You will see the format type in the Binding Preview in the lower left.

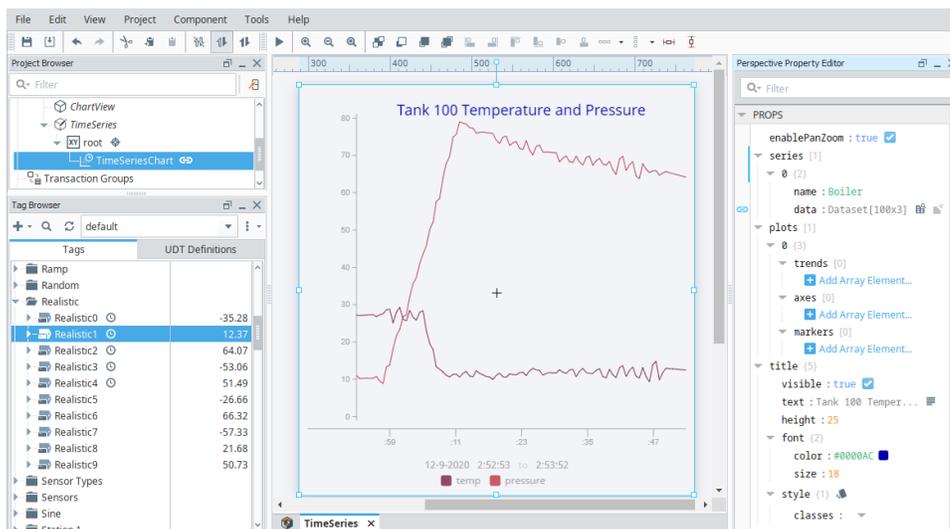
6. Click **OK**.



7. Now you have Tag History data in your Time Series Chart.

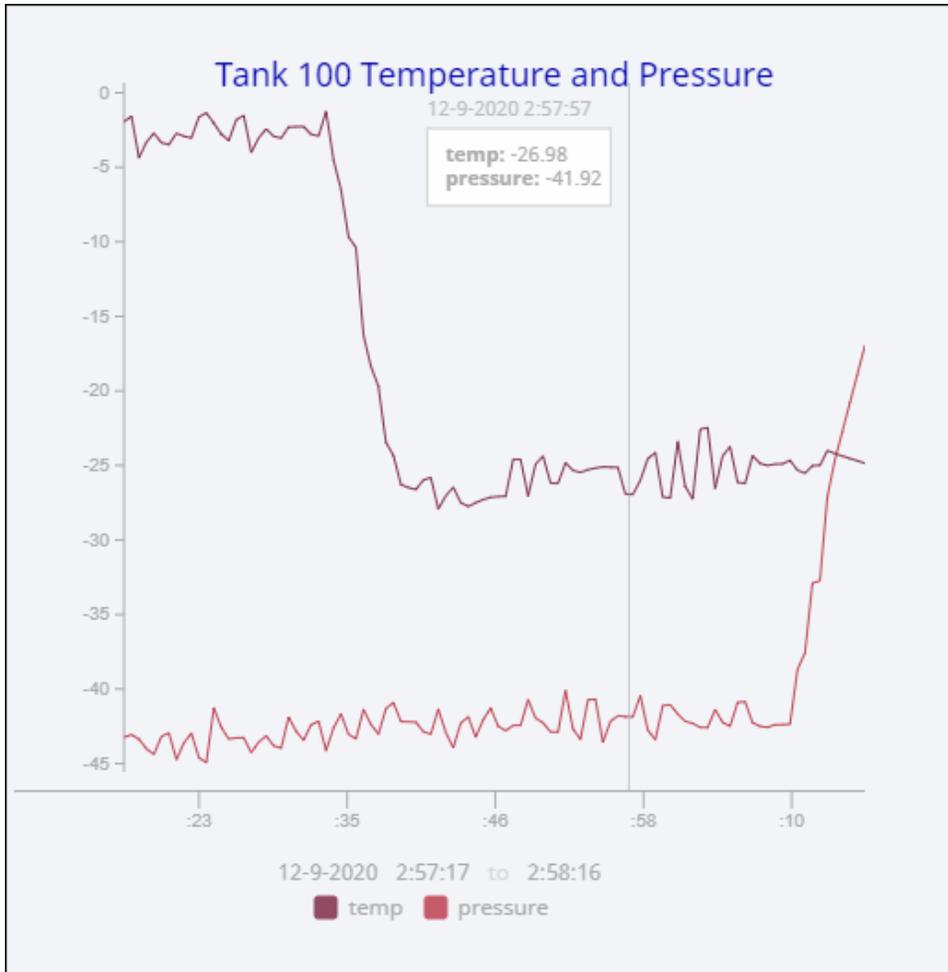
8. To add a title, scroll to the **Title** object in the PROPS section and set the following values:

- title.visible** - Set to true.
- title.text** - Enter a title: **Tank 100 Pressure and Temp Tracking**.
- legend.visible** - Set to true.

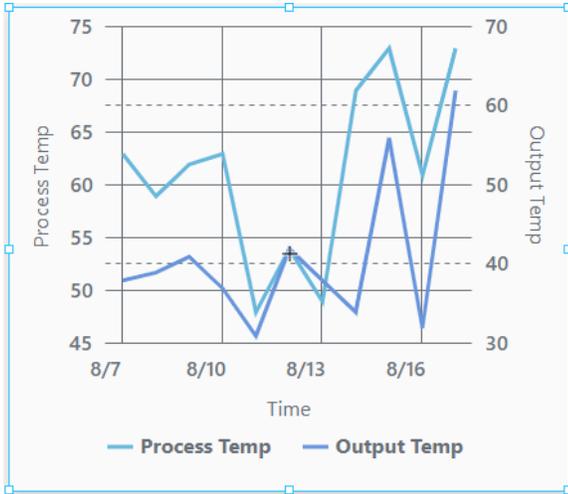


9. Save your project.

10. In **Preview Mode**, when you hover over the chart you will see a timestamp and Temp and Pressure values representing the current x-trace position.



Perspective - XY Chart



On this page ...

- [Properties](#)
- [Component Events](#)
- [Examples](#)

Component Palette Icon:



The XY Chart displays data trends. It provides a flexible way to display either timeseries or X-Y data by entering data in the **dataSources** property. It is fully customizable in its appearance, from labels, colors, line widths, legend, scroll bars, and text styles.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description																							
dataSources	Objects that are the data source for the chart. When configured to show a date or time on the x-axis, the chart expects that each entry in a data source is in timestamp order. It is highly advised that you sort the contents of any given data source.																							
title	Chart title configuration. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>The chart title text.</td> </tr> <tr> <td>appearance</td> <td>Appearance related title options. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color for the title.</td> <td></td> </tr> <tr> <td>font</td> <td>Font settings. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>size</td> <td>Specifies the size of the font in pixels (px) or points (pt). If you enter just a number, Perspective assumes the value is in pixels.</td> <td>value number</td> </tr> <tr> <td>weight</td> <td>Sets how thick or thin characters in the text are displayed.</td> <td>value number</td> </tr> </tbody> </table> </td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	Name	Description	text	The chart title text.	appearance	Appearance related title options. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color for the title.</td> <td></td> </tr> <tr> <td>font</td> <td>Font settings. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>size</td> <td>Specifies the size of the font in pixels (px) or points (pt). If you enter just a number, Perspective assumes the value is in pixels.</td> <td>value number</td> </tr> <tr> <td>weight</td> <td>Sets how thick or thin characters in the text are displayed.</td> <td>value number</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color for the title.		font	Font settings. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>size</td> <td>Specifies the size of the font in pixels (px) or points (pt). If you enter just a number, Perspective assumes the value is in pixels.</td> <td>value number</td> </tr> <tr> <td>weight</td> <td>Sets how thick or thin characters in the text are displayed.</td> <td>value number</td> </tr> </tbody> </table>	Name	Description	Property Type	size	Specifies the size of the font in pixels (px) or points (pt). If you enter just a number, Perspective assumes the value is in pixels.	value number	weight	Sets how thick or thin characters in the text are displayed.	value number
Name	Description																							
text	The chart title text.																							
appearance	Appearance related title options. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color for the title.</td> <td></td> </tr> <tr> <td>font</td> <td>Font settings. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>size</td> <td>Specifies the size of the font in pixels (px) or points (pt). If you enter just a number, Perspective assumes the value is in pixels.</td> <td>value number</td> </tr> <tr> <td>weight</td> <td>Sets how thick or thin characters in the text are displayed.</td> <td>value number</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color for the title.		font	Font settings. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>size</td> <td>Specifies the size of the font in pixels (px) or points (pt). If you enter just a number, Perspective assumes the value is in pixels.</td> <td>value number</td> </tr> <tr> <td>weight</td> <td>Sets how thick or thin characters in the text are displayed.</td> <td>value number</td> </tr> </tbody> </table>	Name	Description	Property Type	size	Specifies the size of the font in pixels (px) or points (pt). If you enter just a number, Perspective assumes the value is in pixels.	value number	weight	Sets how thick or thin characters in the text are displayed.	value number						
Name	Description	Property Type																						
color	Color for the title.																							
font	Font settings. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>size</td> <td>Specifies the size of the font in pixels (px) or points (pt). If you enter just a number, Perspective assumes the value is in pixels.</td> <td>value number</td> </tr> <tr> <td>weight</td> <td>Sets how thick or thin characters in the text are displayed.</td> <td>value number</td> </tr> </tbody> </table>	Name	Description	Property Type	size	Specifies the size of the font in pixels (px) or points (pt). If you enter just a number, Perspective assumes the value is in pixels.	value number	weight	Sets how thick or thin characters in the text are displayed.	value number														
Name	Description	Property Type																						
size	Specifies the size of the font in pixels (px) or points (pt). If you enter just a number, Perspective assumes the value is in pixels.	value number																						
weight	Sets how thick or thin characters in the text are displayed.	value number																						

padding	Padding space around the title. Values can be set for the top, bottom, left, and right.
---------	---

subtitle Chart subtitle configuration.

Name	Description																								
text	The chart subtitle text.																								
appearance	Appearance related subtitle settings.																								
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color for the title.</td> <td>value: string</td> </tr> <tr> <td>font</td> <td>Font settings.</td> <td></td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>size</td> <td>Specifies the size of the font in pixels (px) or points (pt). If you enter just a number, Perspective assumes the value is in pixels.</td> <td>value: numeric</td> </tr> <tr> <td>weight</td> <td>Sets how thick or thin characters in the text are displayed.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>padding</td> <td>Padding space around the subtitle. Values can be set for the top, bottom, left, and right.</td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color for the title.	value: string	font	Font settings.			<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>size</td> <td>Specifies the size of the font in pixels (px) or points (pt). If you enter just a number, Perspective assumes the value is in pixels.</td> <td>value: numeric</td> </tr> <tr> <td>weight</td> <td>Sets how thick or thin characters in the text are displayed.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	size	Specifies the size of the font in pixels (px) or points (pt). If you enter just a number, Perspective assumes the value is in pixels.	value: numeric	weight	Sets how thick or thin characters in the text are displayed.	value: numeric		padding	Padding space around the subtitle. Values can be set for the top, bottom, left, and right.	
Name	Description	Property Type																							
color	Color for the title.	value: string																							
font	Font settings.																								
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>size</td> <td>Specifies the size of the font in pixels (px) or points (pt). If you enter just a number, Perspective assumes the value is in pixels.</td> <td>value: numeric</td> </tr> <tr> <td>weight</td> <td>Sets how thick or thin characters in the text are displayed.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	size	Specifies the size of the font in pixels (px) or points (pt). If you enter just a number, Perspective assumes the value is in pixels.	value: numeric	weight	Sets how thick or thin characters in the text are displayed.	value: numeric															
Name	Description	Property Type																							
size	Specifies the size of the font in pixels (px) or points (pt). If you enter just a number, Perspective assumes the value is in pixels.	value: numeric																							
weight	Sets how thick or thin characters in the text are displayed.	value: numeric																							
padding	Padding space around the subtitle. Values can be set for the top, bottom, left, and right.																								

legend Settings for the chart legend.

Name	Description																																	
enabled	Whether the legend is enabled for this chart. Default is true (enabled).																																	
position	Where the legend is located in relation to the chart (left, right, top, bottom, or absolute). Default is bottom.																																	
absolute	Settings used to position the legend when the position property is set to absolute. Values can be entered for x and y.																																	
markers	Settings for the markers.																																	
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enables all series markers used in the legend.</td> <td>value: boolean</td> </tr> <tr> <td>width</td> <td>The width of each marker, in pixels.</td> <td>value: numeric</td> </tr> <tr> <td>height</td> <td>The height of each marker, in pixels.</td> <td>value: numeric</td> </tr> <tr> <td>mirrorLookOfSeries</td> <td>Markers will be drawn to mirror the look of the series elements. Important: Customizing markers beyond width and height that this value is set to false.</td> <td>value: boolean</td> </tr> <tr> <td>stroke</td> <td>Settings for the marker line stroke.</td> <td></td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Marker line stroke color.</td> <td>value: string</td> </tr> <tr> <td>width</td> <td>Marker line stroke width.</td> <td>value: numeric</td> </tr> <tr> <td>opacity</td> <td>Marker line stroke opacity. 0 is fully transparent, 1 is fully opaque.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Enables all series markers used in the legend.	value: boolean	width	The width of each marker, in pixels.	value: numeric	height	The height of each marker, in pixels.	value: numeric	mirrorLookOfSeries	Markers will be drawn to mirror the look of the series elements. Important: Customizing markers beyond width and height that this value is set to false.	value: boolean	stroke	Settings for the marker line stroke.			<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Marker line stroke color.</td> <td>value: string</td> </tr> <tr> <td>width</td> <td>Marker line stroke width.</td> <td>value: numeric</td> </tr> <tr> <td>opacity</td> <td>Marker line stroke opacity. 0 is fully transparent, 1 is fully opaque.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Marker line stroke color.	value: string	width	Marker line stroke width.	value: numeric	opacity	Marker line stroke opacity. 0 is fully transparent, 1 is fully opaque.	value: numeric	
Name	Description	Property Type																																
enabled	Enables all series markers used in the legend.	value: boolean																																
width	The width of each marker, in pixels.	value: numeric																																
height	The height of each marker, in pixels.	value: numeric																																
mirrorLookOfSeries	Markers will be drawn to mirror the look of the series elements. Important: Customizing markers beyond width and height that this value is set to false.	value: boolean																																
stroke	Settings for the marker line stroke.																																	
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Marker line stroke color.</td> <td>value: string</td> </tr> <tr> <td>width</td> <td>Marker line stroke width.</td> <td>value: numeric</td> </tr> <tr> <td>opacity</td> <td>Marker line stroke opacity. 0 is fully transparent, 1 is fully opaque.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Marker line stroke color.	value: string	width	Marker line stroke width.	value: numeric	opacity	Marker line stroke opacity. 0 is fully transparent, 1 is fully opaque.	value: numeric																					
Name	Description	Property Type																																
color	Marker line stroke color.	value: string																																
width	Marker line stroke width.	value: numeric																																
opacity	Marker line stroke opacity. 0 is fully transparent, 1 is fully opaque.	value: numeric																																

	opaque.														
cornerRadius	Corner radius applied to the rectangle marker. Values can be set for the topLeft, topRight, bottomLeft, and bottomRight.														
icon	Settings for the icon on the legend. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Shorthand path to the icon source, in this format: library/iconName. The materials icon library is a the primary source for icons in Ignition, see https://fonts.google.com/icons?selected=Material+Icons.</td> </tr> <tr> <td>color</td> <td>Color of the icon.</td> </tr> <tr> <td>verticalCenter</td> <td>The vertical anchor point for the icon. Options are none, top, middle, bottom</td> </tr> <tr> <td>horizontalCenter</td> <td>The horizontal anchor point for the icon. Options are none, left, middle, right</td> </tr> <tr> <td>width</td> <td>Width of the icon.</td> </tr> <tr> <td>height</td> <td>Height of the icon.</td> </tr> </tbody> </table>	Name	Description	path	Shorthand path to the icon source, in this format: library/iconName. The materials icon library is a the primary source for icons in Ignition, see https://fonts.google.com/icons?selected=Material+Icons .	color	Color of the icon.	verticalCenter	The vertical anchor point for the icon. Options are none, top, middle, bottom	horizontalCenter	The horizontal anchor point for the icon. Options are none, left, middle, right	width	Width of the icon.	height	Height of the icon.
Name	Description														
path	Shorthand path to the icon source, in this format: library/iconName. The materials icon library is a the primary source for icons in Ignition, see https://fonts.google.com/icons?selected=Material+Icons .														
color	Color of the icon.														
verticalCenter	The vertical anchor point for the icon. Options are none, top, middle, bottom														
horizontalCenter	The horizontal anchor point for the icon. Options are none, left, middle, right														
width	Width of the icon.														
height	Height of the icon.														

labels	Settings for the labels on the legend. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>A template string which is applied to all the labels. Default is [bold]{name}[/].</td> <td>value: string</td> </tr> <tr> <td>font</td> <td>Font settings for the labels. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>weight</td> <td>The weight of the font.</td> <td>value: numeric</td> </tr> <tr> <td>color</td> <td>The color of the font.</td> <td>value: string</td> </tr> <tr> <td>size</td> <td>The font size.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	text	A template string which is applied to all the labels. Default is [bold]{name}[/].	value: string	font	Font settings for the labels. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>weight</td> <td>The weight of the font.</td> <td>value: numeric</td> </tr> <tr> <td>color</td> <td>The color of the font.</td> <td>value: string</td> </tr> <tr> <td>size</td> <td>The font size.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	weight	The weight of the font.	value: numeric	color	The color of the font.	value: string	size	The font size.	value: numeric	object
Name	Description	Property Type																				
text	A template string which is applied to all the labels. Default is [bold]{name}[/].	value: string																				
font	Font settings for the labels. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>weight</td> <td>The weight of the font.</td> <td>value: numeric</td> </tr> <tr> <td>color</td> <td>The color of the font.</td> <td>value: string</td> </tr> <tr> <td>size</td> <td>The font size.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	weight	The weight of the font.	value: numeric	color	The color of the font.	value: string	size	The font size.	value: numeric	object								
Name	Description	Property Type																				
weight	The weight of the font.	value: numeric																				
color	The color of the font.	value: string																				
size	The font size.	value: numeric																				

cursor	Settings for the chart cursor. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enables all chart cursors that are set to show.</td> </tr> <tr> <td>series</td> <td>Binds the chart cursor to a specified series' data source.</td> </tr> <tr> <td>lineX</td> <td>The chart cursor configuration for the line that intersects the X axis. Options as follows: <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>The chart cursor configuration for the line that intersects the X axis.</td> </tr> <tr> <td>stroke</td> <td>Settings for the stroke. Options as follows: <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table> </td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	Name	Description	enabled	Enables all chart cursors that are set to show.	series	Binds the chart cursor to a specified series' data source.	lineX	The chart cursor configuration for the line that intersects the X axis. Options as follows: <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>The chart cursor configuration for the line that intersects the X axis.</td> </tr> <tr> <td>stroke</td> <td>Settings for the stroke. Options as follows: <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	Name	Description	enabled	The chart cursor configuration for the line that intersects the X axis.	stroke	Settings for the stroke. Options as follows: <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table>	Name	Description		
Name	Description																		
enabled	Enables all chart cursors that are set to show.																		
series	Binds the chart cursor to a specified series' data source.																		
lineX	The chart cursor configuration for the line that intersects the X axis. Options as follows: <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>The chart cursor configuration for the line that intersects the X axis.</td> </tr> <tr> <td>stroke</td> <td>Settings for the stroke. Options as follows: <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	Name	Description	enabled	The chart cursor configuration for the line that intersects the X axis.	stroke	Settings for the stroke. Options as follows: <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table>	Name	Description										
Name	Description																		
enabled	The chart cursor configuration for the line that intersects the X axis.																		
stroke	Settings for the stroke. Options as follows: <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table>	Name	Description																
Name	Description																		

color	Cursor line stroke color. See Color Selector .
width	Cursor line stroke width
opacity	Cursor line opacity. 0 is fully transparent, 1 is fully opaque.
dashArray	SVG dashed array. The pattern of dashes and gaps used to paint the stroke. It's a list of comma separated lengths (in pixels) and percentages (percentage of the total stroke length) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5,3,2" is equivalent to "5,3,2,5,3,2".

lineY	The chart cursor configuration for the line that intersects the Y axis. Options as follows:																		
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>The chart cursor configuration for the line that intersects the Y axis.</td> </tr> <tr> <td>stroke</td> <td>Settings for the stroke. Options as follows:</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Cursor line stroke color. See Color Selector.</td> </tr> <tr> <td>width</td> <td>Cursor line stroke width</td> </tr> <tr> <td>opacity</td> <td>Cursor line opacity. 0 is fully transparent, 1 is fully opaque.</td> </tr> <tr> <td>dashArray</td> <td>SVG dashed array. The pattern of dashes and gaps used to paint the stroke. It's a list of comma separated lengths (in pixels) and percentages (percentage of the total stroke length) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5,3,2" is equivalent to "5,3,2,5,3,2".</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	Name	Description	enabled	The chart cursor configuration for the line that intersects the Y axis.	stroke	Settings for the stroke. Options as follows:		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Cursor line stroke color. See Color Selector.</td> </tr> <tr> <td>width</td> <td>Cursor line stroke width</td> </tr> <tr> <td>opacity</td> <td>Cursor line opacity. 0 is fully transparent, 1 is fully opaque.</td> </tr> <tr> <td>dashArray</td> <td>SVG dashed array. The pattern of dashes and gaps used to paint the stroke. It's a list of comma separated lengths (in pixels) and percentages (percentage of the total stroke length) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5,3,2" is equivalent to "5,3,2,5,3,2".</td> </tr> </tbody> </table>	Name	Description	color	Cursor line stroke color. See Color Selector .	width	Cursor line stroke width	opacity	Cursor line opacity. 0 is fully transparent, 1 is fully opaque.	dashArray	SVG dashed array. The pattern of dashes and gaps used to paint the stroke. It's a list of comma separated lengths (in pixels) and percentages (percentage of the total stroke length) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5,3,2" is equivalent to "5,3,2,5,3,2".
Name	Description																		
enabled	The chart cursor configuration for the line that intersects the Y axis.																		
stroke	Settings for the stroke. Options as follows:																		
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Cursor line stroke color. See Color Selector.</td> </tr> <tr> <td>width</td> <td>Cursor line stroke width</td> </tr> <tr> <td>opacity</td> <td>Cursor line opacity. 0 is fully transparent, 1 is fully opaque.</td> </tr> <tr> <td>dashArray</td> <td>SVG dashed array. The pattern of dashes and gaps used to paint the stroke. It's a list of comma separated lengths (in pixels) and percentages (percentage of the total stroke length) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5,3,2" is equivalent to "5,3,2,5,3,2".</td> </tr> </tbody> </table>	Name	Description	color	Cursor line stroke color. See Color Selector .	width	Cursor line stroke width	opacity	Cursor line opacity. 0 is fully transparent, 1 is fully opaque.	dashArray	SVG dashed array. The pattern of dashes and gaps used to paint the stroke. It's a list of comma separated lengths (in pixels) and percentages (percentage of the total stroke length) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5,3,2" is equivalent to "5,3,2,5,3,2".								
Name	Description																		
color	Cursor line stroke color. See Color Selector .																		
width	Cursor line stroke width																		
opacity	Cursor line opacity. 0 is fully transparent, 1 is fully opaque.																		
dashArray	SVG dashed array. The pattern of dashes and gaps used to paint the stroke. It's a list of comma separated lengths (in pixels) and percentages (percentage of the total stroke length) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5,3,2" is equivalent to "5,3,2,5,3,2".																		

enableTransitions	Whether the transition animations are enabled for this chart. Default is false (disabled).
-------------------	--

scrollBars	Configuration for the scroll bars on the chart. Options as follows:																																	
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property</th> </tr> </thead> <tbody> <tr> <td>horizontal</td> <td>Settings for horizontal scrollbars. Options as follows:</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether the horizontal scrollbar is enabled for this chart. Default is true (enabled).</td> <td>value: boolean</td> </tr> <tr> <td>series</td> <td>Binds the horizontal scroll bar to a series.</td> <td>array</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>vertical</td> <td>Settings for vertical scrollbars. Options as follows:</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether the vertical scrollbar is enabled for this chart. Default is true (enabled).</td> <td>value: boolean</td> </tr> <tr> <td>series</td> <td>Binds the vertical scroll bar to a series.</td> <td>array</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property	horizontal	Settings for horizontal scrollbars. Options as follows:	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether the horizontal scrollbar is enabled for this chart. Default is true (enabled).</td> <td>value: boolean</td> </tr> <tr> <td>series</td> <td>Binds the horizontal scroll bar to a series.</td> <td>array</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Whether the horizontal scrollbar is enabled for this chart. Default is true (enabled).	value: boolean	series	Binds the horizontal scroll bar to a series.	array		vertical	Settings for vertical scrollbars. Options as follows:	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether the vertical scrollbar is enabled for this chart. Default is true (enabled).</td> <td>value: boolean</td> </tr> <tr> <td>series</td> <td>Binds the vertical scroll bar to a series.</td> <td>array</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Whether the vertical scrollbar is enabled for this chart. Default is true (enabled).	value: boolean	series	Binds the vertical scroll bar to a series.	array	
Name	Description	Property																																
horizontal	Settings for horizontal scrollbars. Options as follows:	object																																
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether the horizontal scrollbar is enabled for this chart. Default is true (enabled).</td> <td>value: boolean</td> </tr> <tr> <td>series</td> <td>Binds the horizontal scroll bar to a series.</td> <td>array</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Whether the horizontal scrollbar is enabled for this chart. Default is true (enabled).	value: boolean	series	Binds the horizontal scroll bar to a series.	array																								
Name	Description	Property Type																																
enabled	Whether the horizontal scrollbar is enabled for this chart. Default is true (enabled).	value: boolean																																
series	Binds the horizontal scroll bar to a series.	array																																
vertical	Settings for vertical scrollbars. Options as follows:	object																																
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether the vertical scrollbar is enabled for this chart. Default is true (enabled).</td> <td>value: boolean</td> </tr> <tr> <td>series</td> <td>Binds the vertical scroll bar to a series.</td> <td>array</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Whether the vertical scrollbar is enabled for this chart. Default is true (enabled).	value: boolean	series	Binds the vertical scroll bar to a series.	array																								
Name	Description	Property Type																																
enabled	Whether the vertical scrollbar is enabled for this chart. Default is true (enabled).	value: boolean																																
series	Binds the vertical scroll bar to a series.	array																																

selection	
-----------	--

The following feature is new in Ignition version **8.1.10**
[Click here](#) to check out the other new features

An object that contains selection related properties. Data points in a series can only be selected if `series.#.render` is set to either column

Name	Description	Property Type
enabled	Enables selection of bullets, columns, and candlesticks.	value: boolean
data	A read-only list of selected data points.	array
selectedHighlightColor	Selected data points will use this color.	value: color

background

Configuration for the background of the chart.

Name	Description												
render	Sets the render mode for the chart background. Options are none, gradient, or color. Default is none.												
gradient	Sets the gradient configuration for the chart background. Options are: <table border="1" data-bbox="358 684 1269 898"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>direction</td> <td>Sets the direction of the gradient. Options are linear or radial.</td> <td>value: string dropdown</td> </tr> <tr> <td>rotation</td> <td>Gradient rotation. Applies only to the linear gradient. Default is 0.</td> <td>value: numeric</td> </tr> <tr> <td>colors</td> <td>Colors to be used in the gradient.</td> <td>array</td> </tr> </tbody> </table>	Name	Description	Property Type	direction	Sets the direction of the gradient. Options are linear or radial.	value: string dropdown	rotation	Gradient rotation. Applies only to the linear gradient. Default is 0.	value: numeric	colors	Colors to be used in the gradient.	array
Name	Description	Property Type											
direction	Sets the direction of the gradient. Options are linear or radial.	value: string dropdown											
rotation	Gradient rotation. Applies only to the linear gradient. Default is 0.	value: numeric											
colors	Colors to be used in the gradient.	array											
color	Color to be used in background if color property is set under render. Can be chosen from color wheel, chosen from color palette, or HSL value.												
opacity	Opacity of background of the chart. 0 is fully transparent, 1 is fully opaque.												

xAxes

Configuration properties for the X Axes of the chart. Options as follows:

Name	Description															
name	A unique name to identify this axis configuration object. This field is required in order to configure the series.															
label	Enables or disables a label for the x axis. <table border="1" data-bbox="358 1285 1498 1528"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enables a label drawn alongside this axis.</td> </tr> <tr> <td>text</td> <td>Label text.</td> </tr> <tr> <td>color</td> <td>Label color. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> </tr> </tbody> </table>	Name	Description	enabled	Enables a label drawn alongside this axis.	text	Label text.	color	Label color. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .							
Name	Description															
enabled	Enables a label drawn alongside this axis.															
text	Label text.															
color	Label color. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .															
inversed	Indicates if the scale of the axis should be flipped.															
visible	Make the label visible, if label is enabled.															
tooltip	Tool tip configuration for the axis. Options as follows: <table border="1" data-bbox="358 1724 1498 1967"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enables the tooltip. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>text</td> <td>Tooltip text, in the form of a format string. See the AM charts documentation for more details.</td> <td>value: string</td> </tr> <tr> <td>cornerRadius</td> <td>Radius for the corner of tooltip.</td> <td>value: numeric</td> </tr> <tr> <td>pointerLength</td> <td>Length (in pixels) for the pointer on the tooltip.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Enables the tooltip. Default is true.	value: boolean	text	Tooltip text, in the form of a format string. See the AM charts documentation for more details.	value: string	cornerRadius	Radius for the corner of tooltip.	value: numeric	pointerLength	Length (in pixels) for the pointer on the tooltip.	value: numeric
Name	Description	Property Type														
enabled	Enables the tooltip. Default is true.	value: boolean														
text	Tooltip text, in the form of a format string. See the AM charts documentation for more details.	value: string														
cornerRadius	Radius for the corner of tooltip.	value: numeric														
pointerLength	Length (in pixels) for the pointer on the tooltip.	value: numeric														

background	Configuration for the color and opacity of the background of the tooltip. See Color Selector .	color
------------	--	-------

render Sets the axis type to render. Options are category, date, or value. Default is date.

category Category axis configuration. Applied when render is set to category. Groups data items into categories and allots equal space for category axis to remove a certain range from its scale. Options as follows:

Name	Description	Property Type
enabled	Enables a break range. Default is false.	value: boolean
startCategory	Start point of the break.	value: string
endCategory	End point of the break.	value: string
size	The break size as a decimal percentage of the removed values.	value: numeric

date Date axis configuration. Applied when render is set to date. Uses data and time scale. Options as follows:

Name	Description	Property Type															
baseInterval	Adjusts the granularity of the time scale. Otherwise will adjust intelligently by default. Options as follows:																
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enables or disables baseInterval.</td> <td>value</td> </tr> <tr> <td>timeUnit</td> <td>Specifies the base time to apply to this time scale. Options are minute, second, hour, day, week, or year.</td> <td>value dropdown</td> </tr> <tr> <td>count</td> <td>Specifies how many time units each data item was collected.</td> <td>value</td> </tr> <tr> <td>skipEmptyPeriods</td> <td>Removes empty time units from display. Using this feature affects performance. Will reset the use of axis breaks if true.</td> <td>value</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Enables or disables baseInterval.	value	timeUnit	Specifies the base time to apply to this time scale. Options are minute, second, hour, day, week, or year.	value dropdown	count	Specifies how many time units each data item was collected.	value	skipEmptyPeriods	Removes empty time units from display. Using this feature affects performance. Will reset the use of axis breaks if true.	value	
Name	Description	Property Type															
enabled	Enables or disables baseInterval.	value															
timeUnit	Specifies the base time to apply to this time scale. Options are minute, second, hour, day, week, or year.	value dropdown															
count	Specifies how many time units each data item was collected.	value															
skipEmptyPeriods	Removes empty time units from display. Using this feature affects performance. Will reset the use of axis breaks if true.	value															

range You can optionally adjust the date range. Otherwise will auto adjust by default. Options as follows:

Name	Description	Property Type
max	Maximum date in this range.	value: string
min	Minimum date in this range.	value: string
useStrict	Strictly enforces start and end values.	value: boolean

break Tells the date axis to remove a certain range from its scale. Options as follows:

Name	Description	Property Type
inputFormat	Sets the date format of the axis value from the data source.	value: string dropdown
format	Sets the date format for the axis labels.	value: string dropdown

value Value axis configuration. Applied when render is set to date. Uses data and time scale. Options as follows:

Name	Description
range	You can optionally adjust the date range. Otherwise will auto adjust by default. Options as follows:

Name	Description	Property Type
max	Maximum date in this range.	value: string
min	Minimum date in this range.	value: string
useStrict	Strictly enforces start and end values.	value: boolean

logarithmic	Use logarithmic scale. Useful if data varies greatly within the relevant series. Default is false.															
break	Tells the value axis to remove a certain range from its scale. Options as follows: <table border="1" data-bbox="457 430 1334 667"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enables a break range. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>startValue</td> <td>Start point of the break.</td> <td>value: numeric</td> </tr> <tr> <td>endValue</td> <td>End point of the break.</td> <td>value: numeric</td> </tr> <tr> <td>size</td> <td>The break size as a decimal percentage of the removed values.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Enables a break range. Default is false.	value: boolean	startValue	Start point of the break.	value: numeric	endValue	End point of the break.	value: numeric	size	The break size as a decimal percentage of the removed values.	value: numeric
Name	Description	Property Type														
enabled	Enables a break range. Default is false.	value: boolean														
startValue	Start point of the break.	value: numeric														
endValue	End point of the break.	value: numeric														
size	The break size as a decimal percentage of the removed values.	value: numeric														
format	A number format string to be applied against numbers if in number rendering mode. Options are number, integer, four precision, percent, scientific, currency, currency (rounded), or abbreviation.															

appearance	Appearance options for the x axis. Options as follows:															
	Name	Description														
	opposite	Renders the axis on the opposite side. Default is false														
	inside	Renders the axis labels on the inside of the axis. Default is false.														
	labels	Axis label configuration: <table border="1" data-bbox="457 1119 1497 1837"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Axis label color.</td> </tr> <tr> <td>opacity</td> <td>Axis label opacity.</td> </tr> <tr> <td>rotation</td> <td>Rotation of the label. Default is 0.</td> </tr> <tr> <td>wrap</td> <td>Whether or not to wrap the label text. This property is hidden by default, and must be manually added.</td> </tr> <tr> <td>verticalCenter</td> <td> <div data-bbox="574 1480 1485 1564" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.14 Click here to check out the other new features</p> </div> <p>Determines the vertical centering of the label. Options are top, middle, bottom.</p> </td> </tr> <tr> <td>horizontalCenter</td> <td> <div data-bbox="574 1669 1485 1753" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.14 Click here to check out the other new features</p> </div> <p>Determines the horizontal centering of the label. Options are left, middle, right.</p> </td> </tr> </tbody> </table>	Name	Description	color	Axis label color.	opacity	Axis label opacity.	rotation	Rotation of the label. Default is 0.	wrap	Whether or not to wrap the label text. This property is hidden by default, and must be manually added.	verticalCenter	<div data-bbox="574 1480 1485 1564" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.14 Click here to check out the other new features</p> </div> <p>Determines the vertical centering of the label. Options are top, middle, bottom.</p>	horizontalCenter	<div data-bbox="574 1669 1485 1753" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.14 Click here to check out the other new features</p> </div> <p>Determines the horizontal centering of the label. Options are left, middle, right.</p>
	Name	Description														
	color	Axis label color.														
	opacity	Axis label opacity.														
	rotation	Rotation of the label. Default is 0.														
	wrap	Whether or not to wrap the label text. This property is hidden by default, and must be manually added.														
verticalCenter	<div data-bbox="574 1480 1485 1564" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.14 Click here to check out the other new features</p> </div> <p>Determines the vertical centering of the label. Options are top, middle, bottom.</p>															
horizontalCenter	<div data-bbox="574 1669 1485 1753" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.14 Click here to check out the other new features</p> </div> <p>Determines the horizontal centering of the label. Options are left, middle, right.</p>															
grid	Configures the color, opacity and SVG dashed array of the grid lines. <table border="1" data-bbox="457 1900 1497 1967"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> </tbody> </table>	Name	Description													
Name	Description															

color	Axis grid color.
opacity	Axis grid opacity.
dashArray	SVG dashed array. The pattern of dashes and gaps used to paint the stroke. It's a list of comma separated lengths (in pixels) and percentages (percentage of the total stroke length) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5,3,2" is equivalent to "5,3,2,5,3,2".
minDistance	The minimum distance between grid lines.
position	Defines the grid's relative position within the chart. A range from 0 to 1, with 0 meaning start and 1 meaning end.

font Configures the axis font size and weight.

Name	Description	Property Type
size	Axis font size.	value: numeric
weight	Axis font weight.	value: numeric

yAxes

Configuration properties for the Y Axes of the chart.

Name	Description																		
name	A unique name to identify this axis configuration object. This field is required in order to configure the series.																		
label	Enables or disables a label for the y axis. Options as follows: <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enables a label drawn alongside this axis.</td> </tr> <tr> <td>text</td> <td>Label text.</td> </tr> <tr> <td>color</td> <td>Label color. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color</td> </tr> </tbody> </table>	Name	Description	enabled	Enables a label drawn alongside this axis.	text	Label text.	color	Label color. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color										
Name	Description																		
enabled	Enables a label drawn alongside this axis.																		
text	Label text.																		
color	Label color. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color																		
inversed	Indicates if the scale of the axis should be flipped.																		
visible	Make the label visible, if label is enabled.																		
tooltip	Tool tip configuration for the axis. Options as follows: <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enables the tooltip. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>text</td> <td>Tooltip text.</td> <td>value: string</td> </tr> <tr> <td>cornerRadius</td> <td>Radius for the corner of tooltip.</td> <td>value: numeric</td> </tr> <tr> <td>pointerLength</td> <td>Length (in pixels) for the pointer on the tooltip.</td> <td>value: numeric</td> </tr> <tr> <td>background</td> <td>Configuration for the color and opacity of the background of the tooltip.</td> <td>color</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Enables the tooltip. Default is true.	value: boolean	text	Tooltip text.	value: string	cornerRadius	Radius for the corner of tooltip.	value: numeric	pointerLength	Length (in pixels) for the pointer on the tooltip.	value: numeric	background	Configuration for the color and opacity of the background of the tooltip.	color
Name	Description	Property Type																	
enabled	Enables the tooltip. Default is true.	value: boolean																	
text	Tooltip text.	value: string																	
cornerRadius	Radius for the corner of tooltip.	value: numeric																	
pointerLength	Length (in pixels) for the pointer on the tooltip.	value: numeric																	
background	Configuration for the color and opacity of the background of the tooltip.	color																	
render	Sets the axis type to render. Options are category, date, or value. Default is date.																		
category	Applied when render is set to category . Category axis configuration. Groups data items into categories and allots equal space fo <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> </tbody> </table>	Name	Description	Property Type															
Name	Description	Property Type																	

enabled	Enables a break range. Default is false.	value: boolean
startCategory	Start point of the break.	value: string
endCategory	End point of the break.	value: string
size	The break size as a decimal percentage of the removed values.	value: numeric

date	Applied when render is set to date . Date axis configuration. Uses data and time scale. Options as follows:		
	Name	Description	
baseInterval	Adjust the granularity of the time scale. Otherwise will adjust intelligently by default.		
	Name	Description	Property Type
	enabled	Enables or disables baseInterval.	value: boolean
	timeUnit	Specifies the base time to apply to this time scale. Options are minute, second, hour, day, week, or year.	value: string dropdown
	count	Specifies how many time units each data item was collected.	value: numeric
	skipEmptyPeriods	Removes empty time units from display. Using this feature affects performance. Will reset the use of axis breaks if true.	value: boolean
range	You can optionally adjust the date range. Otherwise will auto adjust by default. Options as follows:		
	Name	Description	Property Type
	max	Maximum date in this range.	value: string
	min	Minimum date in this range.	value: string
	useStrict	Strictly enforces start and end values.	value: boolean
break	Tells the date axis to remove a certain range from its scale. Options as follows:		
	Name	Description	Property Type
	enabled	Enables a break range. Default is false.	value: boolean
	startDate	Start date of the break.	value: string
	endDate	End date of the break.	value: string
	size	The break size as a decimal percentage of the removed values.	value: numeric
inputFormat	Sets the date format of the axis value from the data source.		
format	Sets the date format for the axis labels. Options are date, time, or date and time.		

value	Applied when render is set to value . Value axis configuration. Uses data and time scale. Options as follows:		
	Name	Description	
range	You can optionally adjust the date range. Otherwise will auto adjust by default. Options as follows:		
	Name	Description	Property Type
	max	Maximum date in this range.	value: string
	min	Minimum date in this range.	value: string
	useStrict	Strictly enforces start and end values.	value: boolean
logarithmic	Use logarithmic scale. Useful if data varies greatly within the relevant series.		

break	Tells the value axis to remove a certain range from its scale. Options as follows:		
	Name	Description	Property Type
	enabled	Enables a break range. Default is false.	value: boolean
	startValue	Start point of the break.	value: numeric
	endValue	End point of the break.	value: numeric
	size	The break size as a decimal percentage of the removed values.	value: numeric
format	A number format string to be applied against numbers if in number rendering mode. Options are number, integer, four precision, percent, scientific, currency, currency (rounded), or abbreviation.		

appearance	Appearance options for the y axis. Options as follows:			
	Name	Description		
	opposite	Renders the axis on the opposite side. Default is false		
	inside	Renders the axis labels on the inside of the axis. Default is false.		
	labels	Axis label configuration:		
		Name	Description	Property Type
		color	Axis label color.	color
		opacity	Axis label opacity.	value: numeric
		rotation	Rotation of the label. Default is 0.	value: numeric
		wrap	Whether or not to wrap the label text.	value: boolean
grid	Configures the color, opacity and SVG dashed array of the grid lines.			
	Name	Description		
	color	Axis grid color.		
	opacity	Axis grid opacity.		
	dashArray	SVG dashed array. The pattern of dashes and gaps used to paint the grid. It's a list of comma separated lengths (in pixels) and percentages (percentage of the total grid length) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5,3,2" is equivalent to "5,3,2,5,3,2".		
	minDistance	The minimum distance between grid lines.		
	position	Defines the grid's relative position within the chart. A range from 0 to 1, with 0 meaning start and 1 meaning end.		
font	Configures the font size and weight.			
	Name	Description	Property Type	
	size	Axis font size.	value: numeric	
	weight	Axis font weight.	value: numeric	

series An array of series configurations to apply to this chart.

Name	Description
-------------	--------------------

name	A unique name to identify this application of this series.																											
label	Series label to use with legend. <table border="1" data-bbox="358 285 784 386"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Label text.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	text	Label text.	value: string																					
Name	Description	Property Type																										
text	Label text.	value: string																										
visible	Enables series visibility. Default is true.																											
hiddenInLegend	Hides the series in the legend. Default is false.																											
defaultState	Series default state configuration. Default is true (visible).																											
data	Data settings for the series. Options as follows: <table border="1" data-bbox="358 699 1495 1003"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>source</td> <td>Name of the data source to bind to this series.</td> </tr> <tr> <td>x</td> <td>The x value key from the specified data source to be used in this series. Value is a string value if the data source is a array of objects. Value is an index if the data source is an array of arrays.</td> </tr> <tr> <td>y</td> <td>The y value key from the specified data source to be used in this series. Value is a string value if the data source is a array of objects. Value is an index if the data source is an array of arrays.</td> </tr> </tbody> </table>	Name	Description	source	Name of the data source to bind to this series.	x	The x value key from the specified data source to be used in this series. Value is a string value if the data source is a array of objects. Value is an index if the data source is an array of arrays.	y	The y value key from the specified data source to be used in this series. Value is a string value if the data source is a array of objects. Value is an index if the data source is an array of arrays.																			
Name	Description																											
source	Name of the data source to bind to this series.																											
x	The x value key from the specified data source to be used in this series. Value is a string value if the data source is a array of objects. Value is an index if the data source is an array of arrays.																											
y	The y value key from the specified data source to be used in this series. Value is a string value if the data source is a array of objects. Value is an index if the data source is an array of arrays.																											
xAxis	Name of the x axis configuration object to be used with this series.																											
yAxis	Name of the y axis configuration object to be used with this series.																											
zIndex	Sets the series stack order relative to other series.																											
tooltip	Tool tip configuration for the series. Options as follows: <table border="1" data-bbox="358 1272 1333 1713"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enables the tooltip. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>text</td> <td>A format string to apply to the tooltip text.</td> <td>value: string</td> </tr> <tr> <td>cornerRadius</td> <td>Radius for the corner of tooltip.</td> <td>value: numeric</td> </tr> <tr> <td>pointerLength</td> <td>Length (in pixels) for the pointer on the tooltip.</td> <td>value: numeric</td> </tr> <tr> <td>background</td> <td>Configuration for the color and opacity of the background of the tooltip. <table border="1" data-bbox="513 1560 1149 1703"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Background color. See Color Selector.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Background opacity.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Enables the tooltip. Default is true.	value: boolean	text	A format string to apply to the tooltip text.	value: string	cornerRadius	Radius for the corner of tooltip.	value: numeric	pointerLength	Length (in pixels) for the pointer on the tooltip.	value: numeric	background	Configuration for the color and opacity of the background of the tooltip. <table border="1" data-bbox="513 1560 1149 1703"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Background color. See Color Selector.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Background opacity.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Background color. See Color Selector .	color	opacity	Background opacity.	value: numeric	object
Name	Description	Property Type																										
enabled	Enables the tooltip. Default is true.	value: boolean																										
text	A format string to apply to the tooltip text.	value: string																										
cornerRadius	Radius for the corner of tooltip.	value: numeric																										
pointerLength	Length (in pixels) for the pointer on the tooltip.	value: numeric																										
background	Configuration for the color and opacity of the background of the tooltip. <table border="1" data-bbox="513 1560 1149 1703"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Background color. See Color Selector.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Background opacity.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Background color. See Color Selector .	color	opacity	Background opacity.	value: numeric	object																	
Name	Description	Property Type																										
color	Background color. See Color Selector .	color																										
opacity	Background opacity.	value: numeric																										
render	The series render mode. Options are candlestick , column , line , stepLine . Default is line.																											
candlestick	When render is set to candlestick , the following candlestick settings are available. See the XY Chart Example - Candlestick Chart on how to configure a candlestick chart.																											
column	When render is set to column , the following column settings are available.																											

Name	Description	Property Type																																																																																															
open	Open settings.																																																																																																
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>x</td> <td>Name of the field that holds the open data for the horizontal axis.</td> <td>value: string</td> </tr> <tr> <td>y</td> <td>Name of the field that holds the open data for the vertical axis.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	x	Name of the field that holds the open data for the horizontal axis.	value: string	y	Name of the field that holds the open data for the vertical axis.	value: string																																																																																							
Name	Description	Property Type																																																																																															
x	Name of the field that holds the open data for the horizontal axis.	value: string																																																																																															
y	Name of the field that holds the open data for the vertical axis.	value: string																																																																																															
appearance	Appearance settings.																																																																																																
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>fill</td> <td>Fill settings.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Cursor line stroke color.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the stroke.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>stroke</td> <td>Stroke settings.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Cursor line stroke color.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the stroke.</td> <td>value: numeric</td> </tr> <tr> <td>width</td> <td>Width of the stroke, in pixels.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>width</td> <td>Percent</td> <td>value</td> </tr> <tr> <td>height</td> <td>Percent</td> <td>value</td> </tr> <tr> <td>stacked</td> <td>Stacks this column series.</td> <td>value</td> </tr> <tr> <td>deriveFieldsFromData</td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>fill</td> <td>Fill settings.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Cursor line stroke color.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the stroke.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>stroke</td> <td>Stroke settings.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Cursor line stroke color.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the stroke.</td> <td>value: numeric</td> </tr> <tr> <td>width</td> <td>Width of the stroke, in pixels.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table> </td> <td>object</td> </tr> <tr> <td>heatRules</td> <td>These heat rules apply to the fill of the columns.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	fill	Fill settings.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Cursor line stroke color.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the stroke.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Cursor line stroke color.	color	opacity	Opacity of the stroke.	value: numeric		stroke	Stroke settings.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Cursor line stroke color.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the stroke.</td> <td>value: numeric</td> </tr> <tr> <td>width</td> <td>Width of the stroke, in pixels.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Cursor line stroke color.	color	opacity	Opacity of the stroke.	value: numeric	width	Width of the stroke, in pixels.	value: numeric		width	Percent	value	height	Percent	value	stacked	Stacks this column series.	value	deriveFieldsFromData	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>fill</td> <td>Fill settings.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Cursor line stroke color.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the stroke.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>stroke</td> <td>Stroke settings.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Cursor line stroke color.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the stroke.</td> <td>value: numeric</td> </tr> <tr> <td>width</td> <td>Width of the stroke, in pixels.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	fill	Fill settings.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Cursor line stroke color.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the stroke.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Cursor line stroke color.	color	opacity	Opacity of the stroke.	value: numeric		stroke	Stroke settings.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Cursor line stroke color.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the stroke.</td> <td>value: numeric</td> </tr> <tr> <td>width</td> <td>Width of the stroke, in pixels.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Cursor line stroke color.	color	opacity	Opacity of the stroke.	value: numeric	width	Width of the stroke, in pixels.	value: numeric		object	heatRules	These heat rules apply to the fill of the columns.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type				
Name	Description	Property Type																																																																																															
fill	Fill settings.	object																																																																																															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Cursor line stroke color.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the stroke.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Cursor line stroke color.	color	opacity	Opacity of the stroke.	value: numeric																																																																																							
Name	Description	Property Type																																																																																															
color	Cursor line stroke color.	color																																																																																															
opacity	Opacity of the stroke.	value: numeric																																																																																															
stroke	Stroke settings.	object																																																																																															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Cursor line stroke color.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the stroke.</td> <td>value: numeric</td> </tr> <tr> <td>width</td> <td>Width of the stroke, in pixels.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Cursor line stroke color.	color	opacity	Opacity of the stroke.	value: numeric	width	Width of the stroke, in pixels.	value: numeric																																																																																				
Name	Description	Property Type																																																																																															
color	Cursor line stroke color.	color																																																																																															
opacity	Opacity of the stroke.	value: numeric																																																																																															
width	Width of the stroke, in pixels.	value: numeric																																																																																															
width	Percent	value																																																																																															
height	Percent	value																																																																																															
stacked	Stacks this column series.	value																																																																																															
deriveFieldsFromData	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>fill</td> <td>Fill settings.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Cursor line stroke color.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the stroke.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>stroke</td> <td>Stroke settings.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Cursor line stroke color.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the stroke.</td> <td>value: numeric</td> </tr> <tr> <td>width</td> <td>Width of the stroke, in pixels.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	fill	Fill settings.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Cursor line stroke color.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the stroke.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Cursor line stroke color.	color	opacity	Opacity of the stroke.	value: numeric		stroke	Stroke settings.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Cursor line stroke color.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the stroke.</td> <td>value: numeric</td> </tr> <tr> <td>width</td> <td>Width of the stroke, in pixels.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Cursor line stroke color.	color	opacity	Opacity of the stroke.	value: numeric	width	Width of the stroke, in pixels.	value: numeric		object																																																											
Name	Description	Property Type																																																																																															
fill	Fill settings.	object																																																																																															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Cursor line stroke color.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the stroke.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Cursor line stroke color.	color	opacity	Opacity of the stroke.	value: numeric																																																																																							
Name	Description	Property Type																																																																																															
color	Cursor line stroke color.	color																																																																																															
opacity	Opacity of the stroke.	value: numeric																																																																																															
stroke	Stroke settings.	object																																																																																															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Cursor line stroke color.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the stroke.</td> <td>value: numeric</td> </tr> <tr> <td>width</td> <td>Width of the stroke, in pixels.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Cursor line stroke color.	color	opacity	Opacity of the stroke.	value: numeric	width	Width of the stroke, in pixels.	value: numeric																																																																																				
Name	Description	Property Type																																																																																															
color	Cursor line stroke color.	color																																																																																															
opacity	Opacity of the stroke.	value: numeric																																																																																															
width	Width of the stroke, in pixels.	value: numeric																																																																																															
heatRules	These heat rules apply to the fill of the columns.	object																																																																																															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type																																																																																													
Name	Description	Property Type																																																																																															

enabled	Whether or not heat rules are enabled. Default is false.	value: boolean
max	Color for max.	value: string
min	Color for min.	value: string
dataField		value: string

line When **render** is set to **line**, the following line settings are available.

Name	Description
------	-------------

open Configures the open data.

Name	Description	Property Type
x	Name of the field that holds the open data for the horizontal axis	value:string
y	Name of the field that holds the open data for the vertical axis	value:string

appearance Configures the appearance of the line series. Options as follows:

Name	Description
------	-------------

connect Connects the lines over empty data points. Default is true.

tension X Horizontal tension setting of the line. Range is 0 to 1. A 1 value indicates high tension, so the line is maximally attracted to the points it connects (i.e. straight line). A 0 value means the opposite. Default is 1.

tension Y Vertical tension setting of the line. Range is 0 to 1. A 1 value indicates high tension, so the line is maximally attracted to the points it connects (i.e. straight line). A 0 value means the opposite. Default is 1.

minDistance The minimum distance (in pixels) between two points. Default is 0.5.

stroke Series stroke configuration. Options as follows:

Name	Description	Property Type
width	Width of the stroke, in pixels. Default is 3.	value: numeric
opacity	Opacity of the stroke.	value: numeric
color	Cursor line stroke color.	color
dashArray	SVG dash array. The pattern of dashes and gaps used to paint the stroke. It's a list of comma separated lengths (in pixels) and percentages (percentage of the total stroke length) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5,3,2" is equivalent to "5,3,2,5,3,2".	array

fill Series color configuration.

Name	Description	Property Type
color	The color to draw this series.	value: color
opacity	Opacity as a percentage from 0 to 1. 0 is transparent, 1 is opaque.	value: float

bullets Series bullet configuration.

Name	Description	Property Type																		
enabled	Enables bullets.	value: boolean																		
render	Type of bullet to render. Options are circle or label.	value: string dropdown																		
width	Bullet width.	value: numeric																		
height	Bullet height.	value: numeric																		
label	Label properties. <table border="1" data-bbox="701 558 1161 745"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Label text.</td> <td>value: string</td> </tr> <tr> <td>position.dx</td> <td>Label x position.</td> <td>value: numeric</td> </tr> <tr> <td>position.dy</td> <td>Label y position.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	text	Label text.	value: string	position.dx	Label x position.	value: numeric	position.dy	Label y position.	value: numeric	object						
Name	Description	Property Type																		
text	Label text.	value: string																		
position.dx	Label x position.	value: numeric																		
position.dy	Label y position.	value: numeric																		
tooltip	Tooltip configuration. <table border="1" data-bbox="701 810 1346 1182"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enables the tooltip.</td> <td>value: boolean</td> </tr> <tr> <td>text</td> <td>Tooltip text. Can be a format string.</td> <td>value: string</td> </tr> <tr> <td>cornerRadius</td> <td>The corner radius.</td> <td>value: numeric</td> </tr> <tr> <td>pointerLength</td> <td>The pointer length.</td> <td>value: numeric</td> </tr> <tr> <td>background</td> <td>Background color and opacity for the tooltip.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Enables the tooltip.	value: boolean	text	Tooltip text. Can be a format string.	value: string	cornerRadius	The corner radius.	value: numeric	pointerLength	The pointer length.	value: numeric	background	Background color and opacity for the tooltip.	object	object
Name	Description	Property Type																		
enabled	Enables the tooltip.	value: boolean																		
text	Tooltip text. Can be a format string.	value: string																		
cornerRadius	The corner radius.	value: numeric																		
pointerLength	The pointer length.	value: numeric																		
background	Background color and opacity for the tooltip.	object																		
fill	Fill settings. <table border="1" data-bbox="701 1247 1346 1459"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>The fill color for the bullets in this series. See Color Selector.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>The bullet opacity.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	The fill color for the bullets in this series. See Color Selector .	color	opacity	The bullet opacity.	value: numeric	object									
Name	Description	Property Type																		
color	The fill color for the bullets in this series. See Color Selector .	color																		
opacity	The bullet opacity.	value: numeric																		
stroke	Stroke settings. <table border="1" data-bbox="701 1524 1346 1759"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>width</td> <td>Width of the stroke, in pixels. Default is 3.</td> <td>value: numeric</td> </tr> <tr> <td>opacity</td> <td>Opacity of the stroke.</td> <td>value: numeric</td> </tr> <tr> <td>color</td> <td>Cursor line stroke color.</td> <td>color</td> </tr> </tbody> </table>	Name	Description	Property Type	width	Width of the stroke, in pixels. Default is 3.	value: numeric	opacity	Opacity of the stroke.	value: numeric	color	Cursor line stroke color.	color	object						
Name	Description	Property Type																		
width	Width of the stroke, in pixels. Default is 3.	value: numeric																		
opacity	Opacity of the stroke.	value: numeric																		
color	Cursor line stroke color.	color																		
rotation	Rotation of the bullet.	value: numeric																		
deriveFieldsFromData	Settings for derived fields. <table border="1" data-bbox="701 1892 1346 1965"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> </tbody> </table>	Name	Description	Property Type	object															
Name	Description	Property Type																		

fill	Fill settings.		object	
	Name	Description		Property Type
	color	The fill color for the derived fields in this series. See Color Selector .		color
opacity	The derived field opacity.	value: numeric		
stroke	Stroke settings.		object	
	Name	Description		Property Type
	color	Cursor line stroke color.		color
	opacity	Opacity of the stroke.		value: numeric
width	Width of the stroke, in pixels.	value: numeric		
rotation	Derived field rotation (0-360).		value: numeric	
heatRules	These heat rules apply to the radius of a circular bullet.		object	
	Name	Description		Property Type
	enabled	Whether or not heat rules are enabled. Default is false.		value: boolean
	max	Color for max.		value: string
	min	Color for min.		value: string
	dataField	The data field.		value: string

stepLine When **render** is set to **stepLine** . These are the stepLine settings. Options as follows:

Name	Description									
open	Configures the open data.									
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>x</td> <td>Name of the field that holds the open data for the horizontal axis</td> <td>value:string</td> </tr> <tr> <td>y</td> <td>Name of the field that holds the open data for the vertical axis</td> <td>value:string</td> </tr> </tbody> </table>	Name	Description	Property Type	x	Name of the field that holds the open data for the horizontal axis	value:string	y	Name of the field that holds the open data for the vertical axis	value:string
Name	Description	Property Type								
x	Name of the field that holds the open data for the horizontal axis	value:string								
y	Name of the field that holds the open data for the vertical axis	value:string								
appearance	Configures the following options:									
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>connect</td> <td>Connects the lines over empty data points. Default is true.</td> </tr> <tr> <td>tensionX</td> <td>Horizontal tension setting of the line. Used to create smooth or sharp lines Range is 0 to 1. A 1 value indicates high tension, so the line is maximally attracted to the points it connects (i.e. straight line). A 0 value means the opposite. Default is 1.</td> </tr> </tbody> </table>	Name	Description	connect	Connects the lines over empty data points. Default is true.	tensionX	Horizontal tension setting of the line. Used to create smooth or sharp lines Range is 0 to 1. A 1 value indicates high tension, so the line is maximally attracted to the points it connects (i.e. straight line). A 0 value means the opposite. Default is 1.			
Name	Description									
connect	Connects the lines over empty data points. Default is true.									
tensionX	Horizontal tension setting of the line. Used to create smooth or sharp lines Range is 0 to 1. A 1 value indicates high tension, so the line is maximally attracted to the points it connects (i.e. straight line). A 0 value means the opposite. Default is 1.									

tensionY Vertical tension setting of the line. Used to create smooth or sharp lines Range is 0 to 1. A 1 value indicates high tension, so the line is maximally attracted to the points it connects (i.e. straight line). A 0 value means the opposite. Default is 1.

minDistance The minimum distance (in pixels) between two points. Default is 0.5.

stroke Series stroke configuration. Options are as follows:

Name	Description	Property Type
width	Width of the stroke, in pixels. Default is 3.	value: numeric
opacity	Opacity of the stroke. 0 is fully transparent, 1 is fully opaque. Default is 1.	value: numeric
color	Color of border around each pie section. See Color Selector .	color
dashArray	SVG dashed array. The pattern of dashes and gaps used to paint the stroke. It's a list of comma separated lengths (in pixels) and percentages (percentage of the total stroke length) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5,3,2" is equivalent to "5,3,2,5,3,2".	array

fill Fill settings.

Name	Description	Property Type
color	The fill color for the columns in this series. See Color Selector .	color
opacity	The column opacity.	value: numeric

bullets Series bullet configuration.

Name	Description	Property Type															
enabled	Enables bullets.	value: boolean															
render	Type of bullet to render. Options are circle or label.	value: string dropdown															
width	Bullet width.	value: numeric															
height	Bullet height.	value: numeric															
label	Label settings. <table border="1" data-bbox="706 1486 1128 1633"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Label text.</td> <td>value: string</td> </tr> <tr> <td>position</td> <td>Label position.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	text	Label text.	value: string	position	Label position.	object	object						
Name	Description	Property Type															
text	Label text.	value: string															
position	Label position.	object															
tooltip	Tooltip settings. <table border="1" data-bbox="706 1696 1344 1969"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enables the tooltip.</td> <td>value: boolean</td> </tr> <tr> <td>text</td> <td>Tooltip text. Can be a format string.</td> <td>value: string</td> </tr> <tr> <td>cornerRadius</td> <td>The corner radius.</td> <td>value: numeric</td> </tr> <tr> <td>pointerLen</td> <td>The pointer length.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Enables the tooltip.	value: boolean	text	Tooltip text. Can be a format string.	value: string	cornerRadius	The corner radius.	value: numeric	pointerLen	The pointer length.	value: numeric	object
Name	Description	Property Type															
enabled	Enables the tooltip.	value: boolean															
text	Tooltip text. Can be a format string.	value: string															
cornerRadius	The corner radius.	value: numeric															
pointerLen	The pointer length.	value: numeric															

gth		
background	Background color and opacity for the tool tip.	object

fill	Fill settings.		object	
	Name	Description		Property Type
	color	The fill color for the columns in this series. See Color Selector .		color
	opacity	The column opacity.		value: numeric

stroke	Stroke settings.		object	
	Name	Description		Property Type
	color	Cursor line stroke color.		color
	opacity	Opacity of the stroke.		value: numeric
width	Width of the stroke, in pixels.	value: numeric		

rotation	Bullet rotation. Value can be 0 to 360.	value: numeric
----------	---	----------------

deriveFieldsFromData	Settings for derived fields.		object		
	Name	Description		Property Type	
	fill	Fill settings.			
		Name		Description	Property Type
		color		The fill color for the columns in this series. See Color Selector .	color
	opacity	The column opacity.		value: numeric	
	stroke	Stroke settings.			
		Name		Description	Property Type
		color		Cursor line stroke color.	color
		opacity		Opacity of the stroke.	value: numeric
width	Width of the stroke, in pixels.	value: numeric			
rotation	Bullet rotation (0-360).	value: numeric			

heatRules	These heat rules apply to the radius of a circular bullet.		object	
	Name	Description		Property Type
	enabled	Whether or not heat rules are enabled. Default is false.		value: boolean
	max	Color for max.		value: string

				min	Color for min.	value: string
				dataField		value: string
style	Sets a style for this chart. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous					

Component Events

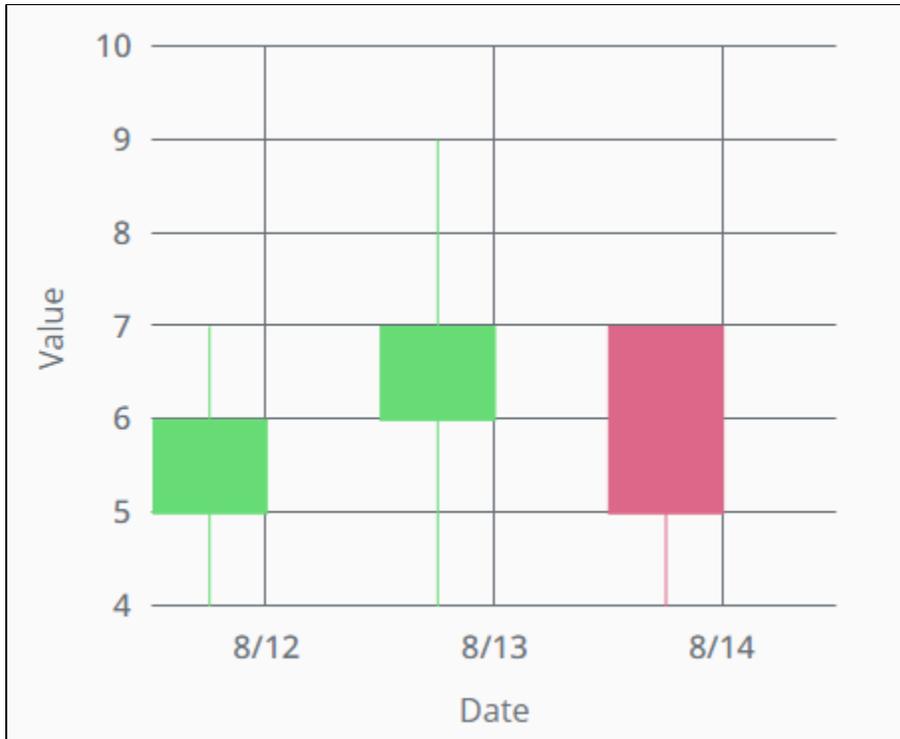
Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Examples

XY Chart Example - Candlestick Chart

This example demonstrates how to configure a candlestick chart. The candlestick chart needs five keys per entry on the chart: a key to represent the date of the entry and four keys that make up the candlestick. These four keys are called open, close, high, and low. These dedicated key properties are visible under the **series** when the render selection is set to candlestick, and therefore this chart does not require that the keys under **dataSources** use the exact naming conventions as shown on this page.



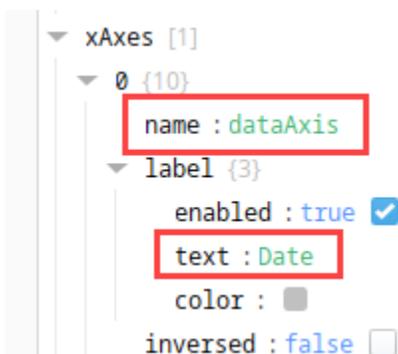
Configure a Candlestick Chart

1. Create an XY chart component and select it.
2. Copy the JSON content below, and paste it onto the chart component's **dataSources.example** property.

```
[
  {
    "close": 6,
    "date": "2021-8-12 15:37:19",
    "high": 7,
    "low": 2,
    "open": 5
  },
  {
    "close": 7,
    "date": "2021-8-13 15:37:19",
    "high": 9,
    "low": 5,
    "open": 6
  },
  {
    "close": 5,
    "date": "2021-8-14 15:37:19",
    "high": 7,
    "low": 4,
    "open": 7
  }
]
```

3. Remove the legend by setting **legend.enabled** to false.

4. Rename `xAxes.0.name` to `dateAxis`.
5. Set `xAxes.0.label.text` to `Date`, to change the label on the x-axis.



6. Our chart only needs a single y-axis, so delete `yAxes.1`.
7. Set `yAxes.0.name` to `valuation`.
8. Set `yAxes.0.label.text` to `Value`.
9. Delete `series.1` since we only need a single series for this example.
10. Navigate to the `series.0.data` property and make sure the `source` is still `example` from the JSON string pasted in step 2.
11. Set `series.0.data.x` to the `date` key
12. Set `series.0.data.y` to the `close` key.
13. Set `series.0.xAxis` to the `xAxes: dateAxis`.
14. Set `series.0.yAxis` to the `yAxes: valuation`.
15. Set `series.0.render` to `candlestick`.
16. Expand the now visible candlestick properties and set the open, high, and low keys:
 - `series.0.candlestick.open`: set `x` to `date` and `y` to `open`.
 - `series.0.candlestick.high`: set `x` to `date` and `y` to `high`.
 - `series.0.candlestick.low`: set `x`, to `date` and `y` to `low`.



Your candlestick chart is now ready. Note that the third entry is in red, while the first two are green. This is because that entry's close value is less than its open value.

Example Configuration

The JSON string below can be used to replicate the example above. Simply copy the contents of the code block and paste it into a container in your Designer.

```
[
  {
    "type": "ia.chart.xy",
    "version": 0,
    "props": {
      "legend": {
        "enabled": false
      },
      "xAxes": [
        {
          "name": "dateAxis",
```

```
"label": {
  "enabled": true,
  "text": "Date",
  "color": ""
},
"visible": true,
"tooltip": {
  "enabled": true,
  "text": "",
  "cornerRadius": 3,
  "pointerLength": 4,
  "background": {
    "color": "",
    "opacity": 1
  }
},
"inversed": false,
"render": "date",
"category": {
  "break": {
    "enabled": false,
    "startCategory": "",
    "endCategory": "",
    "size": 0.05
  }
},
"date": {
  "baseInterval": {
    "enabled": false,
    "timeUnit": "hour",
    "count": 1,
    "skipEmptyPeriods": false
  },
  "range": {
    "max": "",
    "min": "",
    "useStrict": false
  },
  "break": {
    "enabled": false,
    "startDate": "",
    "endDate": "",
    "size": 0.05
  },
  "inputFormat": "yyyy-MM-dd kk:mm:ss",
  "format": "M/d"
},
"value": {
  "range": {
    "max": "",
    "min": "",
    "useStrict": false
  },
  "logarithmic": false,
  "break": {
    "enabled": false,
    "startValue": 0,
    "endValue": 100,
    "size": 0.05
  },
  "format": "#,###.##"
},
"appearance": {
  "opposite": false,
  "inside": false,
  "labels": {
    "color": "",
    "opacity": 1
  }
},
"grid": {
  "color": "",
```

```
        "opacity": 1,
        "dashArray": "",
        "minDistance": 60,
        "position": 0.5
    },
    "font": {
        "size": "",
        "weight": 500
    }
}
],
"yAxes": [
    {
        "name": "valuation",
        "label": {
            "enabled": true,
            "text": "Value",
            "color": ""
        },
        "visible": true,
        "tooltip": {
            "enabled": true,
            "text": "",
            "cornerRadius": 3,
            "pointerLength": 4,
            "background": {
                "color": "",
                "opacity": 1
            }
        },
        "inversed": false,
        "render": "value",
        "category": {
            "break": {
                "enabled": false,
                "startCategory": "",
                "endCategory": "",
                "size": 0.05
            }
        },
        "date": {
            "baseInterval": {
                "enabled": false,
                "timeUnit": "hour",
                "count": 1,
                "skipEmptyPeriods": false
            },
            "range": {
                "max": "",
                "min": "",
                "useStrict": false
            },
            "break": {
                "enabled": false,
                "startDate": "",
                "endDate": "",
                "size": 0.05
            },
            "inputFormat": "yyyy-MM-dd kk:mm:ss",
            "format": "M/d/yyyy HH:mm:ss"
        },
        "value": {
            "range": {
                "max": "",
                "min": "",
                "useStrict": false
            },
            "logarithmic": false,
            "break": {
                "enabled": false,
```

```

        "startValue": 0,
        "endValue": 100,
        "size": 0.05
    },
    "format": "#,###.##"
},
"appearance": {
    "opposite": false,
    "inside": false,
    "labels": {
        "color": "",
        "opacity": 1
    },
    "grid": {
        "color": "",
        "opacity": 1,
        "dashArray": "",
        "minDistance": null,
        "position": 0.5
    },
    "font": {
        "size": "",
        "weight": 500
    }
}
},
],
"series": [
    {
        "name": "process temp",
        "label": {
            "text": "Process Temp"
        },
        "visible": true,
        "hiddenInLegend": false,
        "defaultState": {
            "visible": true
        },
        "data": {
            "source": "example",
            "x": "date",
            "y": "close"
        },
        "xAxis": "dateAxis",
        "yAxis": "valuation",
        "zIndex": 0,
        "tooltip": {
            "enabled": true,
            "text": "{name}: [bold]{valueY}[/]",
            "cornerRadius": 3,
            "pointerLength": 4,
            "background": {
                "color": "",
                "opacity": 1
            }
        },
        "render": "candlestick",
        "candlestick": {
            "open": {
                "x": "date",
                "y": "open"
            },
            "high": {
                "x": "date",
                "y": "high"
            },
            "low": {
                "x": "date",
                "y": "low"
            },
            "appearance": {

```

```
    "fill": {
      "color": "",
      "opacity": 1
    },
    "stroke": {
      "color": "",
      "opacity": 1,
      "width": 1
    },
    "stacked": false,
    "deriveFieldsFromData": {
      "fill": {
        "color": "",
        "opacity": ""
      },
      "stroke": {
        "color": "",
        "opacity": "",
        "width": ""
      }
    },
    "heatRules": {
      "enabled": false,
      "max": "",
      "min": "",
      "dataField": ""
    }
  },
  "column": {
    "open": {
      "x": "",
      "y": ""
    },
    "appearance": {
      "fill": {
        "color": "",
        "opacity": 1
      },
      "stroke": {
        "color": "",
        "opacity": 1,
        "width": 1
      },
      "stacked": false,
      "width": null,
      "height": null,
      "deriveFieldsFromData": {
        "fill": {
          "color": "",
          "opacity": ""
        },
        "stroke": {
          "color": "",
          "opacity": "",
          "width": ""
        }
      },
      "heatRules": {
        "enabled": false,
        "max": "",
        "min": "",
        "dataField": ""
      }
    }
  },
  "line": {
    "open": {
      "x": "",
      "y": ""
    },
  },
```

```

"appearance": {
  "connect": true,
  "tensionX": 1,
  "tensionY": 1,
  "minDistance": 0.5,
  "stroke": {
    "width": 3,
    "opacity": 1,
    "color": "",
    "dashArray": ""
  },
  "fill": {
    "opacity": 0,
    "color": ""
  },
  "bullets": [
    {
      "enabled": false,
      "render": "circle",
      "width": 10,
      "height": 10,
      "label": {
        "text": "{value}",
        "position": {
          "dx": 0,
          "dy": 0
        }
      },
      "fill": {
        "color": "",
        "opacity": 1
      },
      "stroke": {
        "color": "",
        "opacity": 1,
        "width": 1
      },
      "rotation": 0,
      "tooltip": {
        "enabled": true,
        "text": "{name}: [bold]{valueY}[/]",
        "cornerRadius": 3,
        "pointerLength": 4,
        "background": {
          "color": "",
          "opacity": 1
        }
      },
      "deriveFieldsFromData": {
        "fill": {
          "color": "",
          "opacity": ""
        },
        "stroke": {
          "color": "",
          "opacity": "",
          "width": ""
        },
        "rotation": ""
      },
      "heatRules": {
        "enabled": false,
        "max": 100,
        "min": 2,
        "dataField": ""
      }
    }
  ]
},
"stepLine": {

```

```

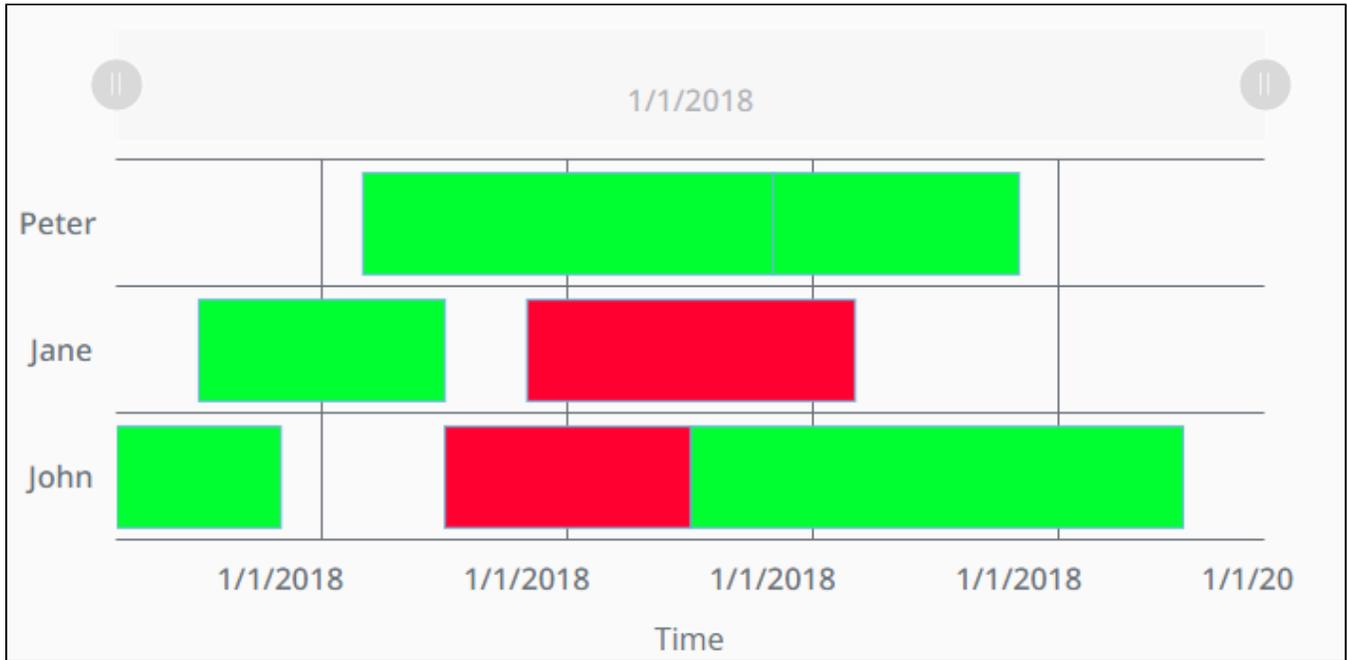
"open": {
  "x": "",
  "y": ""
},
"appearance": {
  "connect": true,
  "tensionX": 1,
  "tensionY": 1,
  "minDistance": 0.5,
  "stroke": {
    "width": 3,
    "opacity": 1,
    "color": "",
    "dashArray": ""
  },
  "fill": {
    "opacity": 0,
    "color": ""
  },
  "bullets": [
    {
      "enabled": true,
      "render": "circle",
      "width": 10,
      "height": 10,
      "label": {
        "text": "{value}",
        "position": {
          "dx": 0,
          "dy": 0
        }
      }
    },
    {
      "fill": {
        "color": "",
        "opacity": 1
      },
      "stroke": {
        "color": "",
        "opacity": 1,
        "width": 1
      },
      "rotation": 0,
      "tooltip": {
        "enabled": true,
        "text": "{name}: [bold]{valueY}[/]",
        "cornerRadius": 3,
        "pointerLength": 4,
        "background": {
          "color": "",
          "opacity": 1
        }
      }
    },
    {
      "deriveFieldsFromData": {
        "fill": {
          "color": "",
          "opacity": ""
        },
        "stroke": {
          "color": "",
          "opacity": "",
          "width": ""
        },
        "rotation": ""
      }
    },
    {
      "heatRules": {
        "enabled": false,
        "max": 100,
        "min": 2,
        "dataField": ""
      }
    }
  ]
}

```

```
    ]
  }
}
],
"dataSources": {
  "example": [
    {
      "close": 6,
      "date": "2021-8-12 15:37:19",
      "high": 7,
      "low": 2,
      "open": 5
    },
    {
      "close": 7,
      "date": "2021-8-13 15:37:19",
      "high": 9,
      "low": 5,
      "open": 6
    },
    {
      "close": 5,
      "date": "2021-8-14 15:37:19",
      "high": 7,
      "low": 4,
      "open": 7
    }
  ]
}
],
"meta": {
  "name": "XYChart_0"
},
"position": {
  "x": 141,
  "y": 103,
  "height": 375,
  "width": 435
},
"custom": {}
}
```

XY Chart Example - Gantt Chart

In addition to basic data plotting, an XY Chart can be used to plot horizontal bar across different lanes, in this fashion:



This can be useful when it comes to machine downtime scheduling, shift scheduling, and maintenance scheduling. To achieve this functionality with the XY Chart, follow the example below:

1. From the Perspective section of the Project Browser on your Designer, right click on the Views folder and select **New View...** to create a new view.
2. This will bring up the New View window. Give your view a name and select the Coordinate Root Container Type. The Page URL setting will remain unchecked for this example.

New View

Name: View ✓

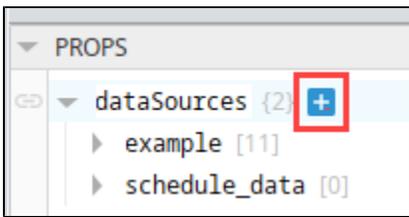
Root Container Type: Coordinate ▼

Page URL

/ ✓

Cancel Create View

3. From the Perspective Component Palette, drag and drop a XY Chart onto your newly created view.
4. Click to select your newly added XY Chart and from the property editor, click on the + icon next to the **dataSources** property and select Array to create a new data source array and name it `schedule_data`.



- Right click on your newly created `schedule_data` `dataSource` and paste the following:

```
[
  {
    "name": "John",
    "fromDate": "2018-01-01 08:00",
    "toDate": "2018-01-01 10:00",
    "color": "#00FF30"
  },
  {
    "name": "John",
    "fromDate": "2018-01-01 12:00",
    "toDate": "2018-01-01 15:00",
    "color": "#FF0030"
  },
  {
    "name": "John",
    "fromDate": "2018-01-01 15:30",
    "toDate": "2018-01-01 21:30",
    "color": "#00FF30"
  },
  {
    "name": "Jane",
    "fromDate": "2018-01-01 09:00",
    "toDate": "2018-01-01 12:00",
    "color": "#00FF30"
  },
  {
    "name": "Jane",
    "fromDate": "2018-01-01 13:00",
    "toDate": "2018-01-01 17:00",
    "color": "#FF0030"
  },
  {
    "name": "Peter",
    "fromDate": "2018-01-01 11:00",
    "toDate": "2018-01-01 16:00",
    "color": "#00FF30"
  },
  {
    "name": "Peter",
    "fromDate": "2018-01-01 16:00",
    "toDate": "2018-01-01 19:00",
    "color": "#00FF30"
  }
]
```

Note: Any data that you wish to plot on the XY Chart to build a Gantt Chart needs to come in the format specified by the `schedule_data` property above.

- Set XY Chart's variables `cursor.series` and `scrollbars.horizontal.series` to "schedule data".
- Set `legend.enabled` to false, to disable the legend.
- Set the X axes properties as follows:
 - Configure the X axes date format by setting the property `xAxes[0].date.format` to be a date format that works for you. For this example, we chose `date` from the dropdown, which sets it to "M/d/yyyy".
 - Set the properties `xAxes[0].appearance.grid.minDistance` to NULL.

9. Set the Y axes properties as follows:
 - a. Delete the **yAxes[1]** property since it will not be used.
 - b. Set the property **yAxes[0].name** to "Operator".
 - c. Disable the Y axes label by setting the **yAxes[0].label.enabled** property to false.
 - d. Set the property **yAxes[0].render** to "category".
 - e. Set your **yAxes[0].appearance.grid.position** to "0".
10. Set the series properties as follows:
 - a. Delete the **series[1]** property since it will not be used.
 - b. Set your **series[0].name** to "schedule data". This links your Horizontal Scroll bar from step 6 to your chart's series.
 - c. Set your **series[0].data.source** to "schedule_data". This links your chart's series to your data source from step 4.
 - d. Configure your **series[0].data.x** to be "toDate" and your **series[0].data.y** to be "name".
 - e. Configure your **series[0].xAxis** to be "time" and your **series[0].yAxis** to be "Operator".
 - f. Set your **series[0].render** property to "column".
 - g. Your **series[0].column.open.x** property must be set to "fromDate".
 - h. To help with data visualization, set your **series[0].tooltip.text** to "{name}: [bold]{fromDate} - {toDate}[/]".
11. Our data includes a key for color, but our chart isn't currently using it. Set **series[0].column.appearance.deriveFieldsFromData.fill.color** to "color". This will map the "color" key in our data source to each block in the chart.
12. Save your project.
13. Put the Designer into Preview mode to see the chart in action.

Example Configuration

The JSON string below can be used to replicate the heatmap example. Simply copy the contents of the code block below (**double click** on any part of the JSON to select all of it), and paste it into a container in your designer.

```
[
  {
    "type": "ia.chart.xy",
    "version": 0,
    "props": {
      "legend": {
        "enabled": false
      },
      "cursor": {
        "series": "mySeries"
      },
      "scrollBars": {
        "horizontal": {
          "series": "mySeries"
        }
      },
      "xAxes": [
        {
          "name": "time",
          "label": {
            "enabled": true,
            "text": "Time",
            "color": ""
          },
          "visible": true,
          "tooltip": {
            "enabled": true,
            "text": "",
            "cornerRadius": 3,
            "pointerLength": 4,
            "background": {
              "color": "",
              "opacity": 1
            }
          },
          "inversed": false,
          "render": "date",
          "category": {
            "break": {
              "enabled": false,
              "startCategory": "",
              "endCategory": "",
              "size": 0.05
            }
          }
        },
        {
          "date": {
            "baseInterval": {
```

```

        "enabled": false,
        "timeUnit": "hour",
        "count": 1,
        "skipEmptyPeriods": false
    },
    "range": {
        "max": "",
        "min": "",
        "useStrict": false
    },
    "break": {
        "enabled": false,
        "startDate": "",
        "endDate": "",
        "size": 0.05
    },
    "inputFormat": "yyyy-MM-dd kk:mm:ss",
    "format": "M/d/yyyy"
},
"value": {
    "range": {
        "max": "",
        "min": "",
        "useStrict": false
    },
    "logarithmic": false,
    "break": {
        "enabled": false,
        "startValue": 0,
        "endValue": 100,
        "size": 0.05
    },
    "format": "#,###.##"
},
"appearance": {
    "opposite": false,
    "inside": false,
    "labels": {
        "color": "",
        "opacity": 1
    },
    "grid": {
        "color": "",
        "opacity": 1,
        "dashArray": "",
        "minDistance": 60,
        "position": 0.5
    },
    "font": {
        "size": "",
        "weight": 500
    }
}
},
],
"yAxes": [
    {
        "name": "Operator",
        "label": {
            "enabled": false,
            "text": "Process Temp",
            "color": ""
        },
        "visible": true,
        "tooltip": {
            "enabled": true,
            "text": "",
            "cornerRadius": 3,
            "pointerLength": 4,
            "background": {
                "color": ""
            }
        }
    }
]

```

```

        "opacity": 1
    }
},
"inversed": false,
"render": "category",
"category": {
    "break": {
        "enabled": false,
        "startCategory": "",
        "endCategory": "",
        "size": 0.05
    }
},
"date": {
    "baseInterval": {
        "enabled": false,
        "timeUnit": "hour",
        "count": 1,
        "skipEmptyPeriods": false
    },
    "range": {
        "max": "",
        "min": "",
        "useStrict": false
    },
    "break": {
        "enabled": false,
        "startDate": "",
        "endDate": "",
        "size": 0.05
    },
    "inputFormat": "yyyy-MM-dd kk:mm:ss",
    "format": "M/d/yyyy HH:mm:ss"
},
"value": {
    "range": {
        "max": "",
        "min": "",
        "useStrict": false
    },
    "logarithmic": false,
    "break": {
        "enabled": false,
        "startValue": 0,
        "endValue": 100,
        "size": 0.05
    },
    "format": "#,###.##"
},
"appearance": {
    "opposite": false,
    "inside": false,
    "labels": {
        "color": "",
        "opacity": 1
    },
    "grid": {
        "color": "",
        "opacity": 1,
        "dashArray": "",
        "minDistance": null,
        "position": 0
    },
    "font": {
        "size": "",
        "weight": 500
    }
}
},
"series": [

```

```

{
  "name": "mySeries",
  "label": {
    "text": "Process Temp"
  },
  "visible": true,
  "hiddenInLegend": false,
  "defaultState": {
    "visible": true
  },
  "data": {
    "source": "example",
    "x": "toDate",
    "y": "name"
  },
  "xAxis": "time",
  "yAxis": "Operator",
  "zIndex": 0,
  "tooltip": {
    "enabled": true,
    "text": "{name} [bold]{fromDate} - {toDate}[/]",
    "cornerRadius": 3,
    "pointerLength": 4,
    "background": {
      "color": "",
      "opacity": 1
    }
  },
  "render": "column",
  "candlestick": {
    "open": {
      "x": "",
      "y": ""
    },
    "high": {
      "x": "",
      "y": ""
    },
    "low": {
      "x": "",
      "y": ""
    },
    "appearance": {
      "fill": {
        "color": "",
        "opacity": 1
      },
      "stroke": {
        "color": "",
        "opacity": 1,
        "width": 1
      },
      "stacked": false,
      "deriveFieldsFromData": {
        "fill": {
          "color": "",
          "opacity": ""
        },
        "stroke": {
          "color": "",
          "opacity": "",
          "width": ""
        }
      },
      "heatRules": {
        "enabled": false,
        "max": "",
        "min": "",
        "dataField": ""
      }
    }
  }
}

```

```

},
"column": {
  "open": {
    "x": "fromDate",
    "y": ""
  },
  "appearance": {
    "fill": {
      "color": "",
      "opacity": 1
    },
    "stroke": {
      "color": "",
      "opacity": 1,
      "width": 1
    },
    "stacked": false,
    "width": null,
    "height": null,
    "deriveFieldsFromData": {
      "fill": {
        "color": "color",
        "opacity": ""
      },
      "stroke": {
        "color": "",
        "opacity": "",
        "width": ""
      }
    },
    "heatRules": {
      "enabled": false,
      "max": "",
      "min": "",
      "dataField": ""
    }
  }
},
"line": {
  "open": {
    "x": "",
    "y": ""
  },
  "appearance": {
    "connect": true,
    "tensionX": 1,
    "tensionY": 1,
    "minDistance": 0.5,
    "stroke": {
      "width": 3,
      "opacity": 1,
      "color": "",
      "dashArray": ""
    },
    "fill": {
      "opacity": 0,
      "color": ""
    },
    "bullets": [
      {
        "enabled": false,
        "render": "circle",
        "width": 10,
        "height": 10,
        "label": {
          "text": "{value}",
          "position": {
            "dx": 0,
            "dy": 0
          }
        }
      }
    ]
  }
},

```

```

    "fill": {
      "color": "",
      "opacity": 1
    },
    "stroke": {
      "color": "",
      "opacity": 1,
      "width": 1
    },
    "rotation": 0,
    "tooltip": {
      "enabled": true,
      "text": "{name}: [bold]{valueY}[/]",
      "cornerRadius": 3,
      "pointerLength": 4,
      "background": {
        "color": "",
        "opacity": 1
      }
    },
    "deriveFieldsFromData": {
      "fill": {
        "color": "",
        "opacity": ""
      },
      "stroke": {
        "color": "",
        "opacity": "",
        "width": ""
      },
      "rotation": ""
    },
    "heatRules": {
      "enabled": false,
      "max": 100,
      "min": 2,
      "dataField": ""
    }
  }
]
},
"stepLine": {
  "open": {
    "x": "",
    "y": ""
  },
  "appearance": {
    "connect": true,
    "tensionX": 1,
    "tensionY": 1,
    "minDistance": 0.5,
    "stroke": {
      "width": 3,
      "opacity": 1,
      "color": "",
      "dashArray": ""
    },
    "fill": {
      "opacity": 0,
      "color": ""
    },
    "bullets": [
      {
        "enabled": true,
        "render": "circle",
        "width": 10,
        "height": 10,
        "label": {
          "text": "{value}",
          "position": {

```

```

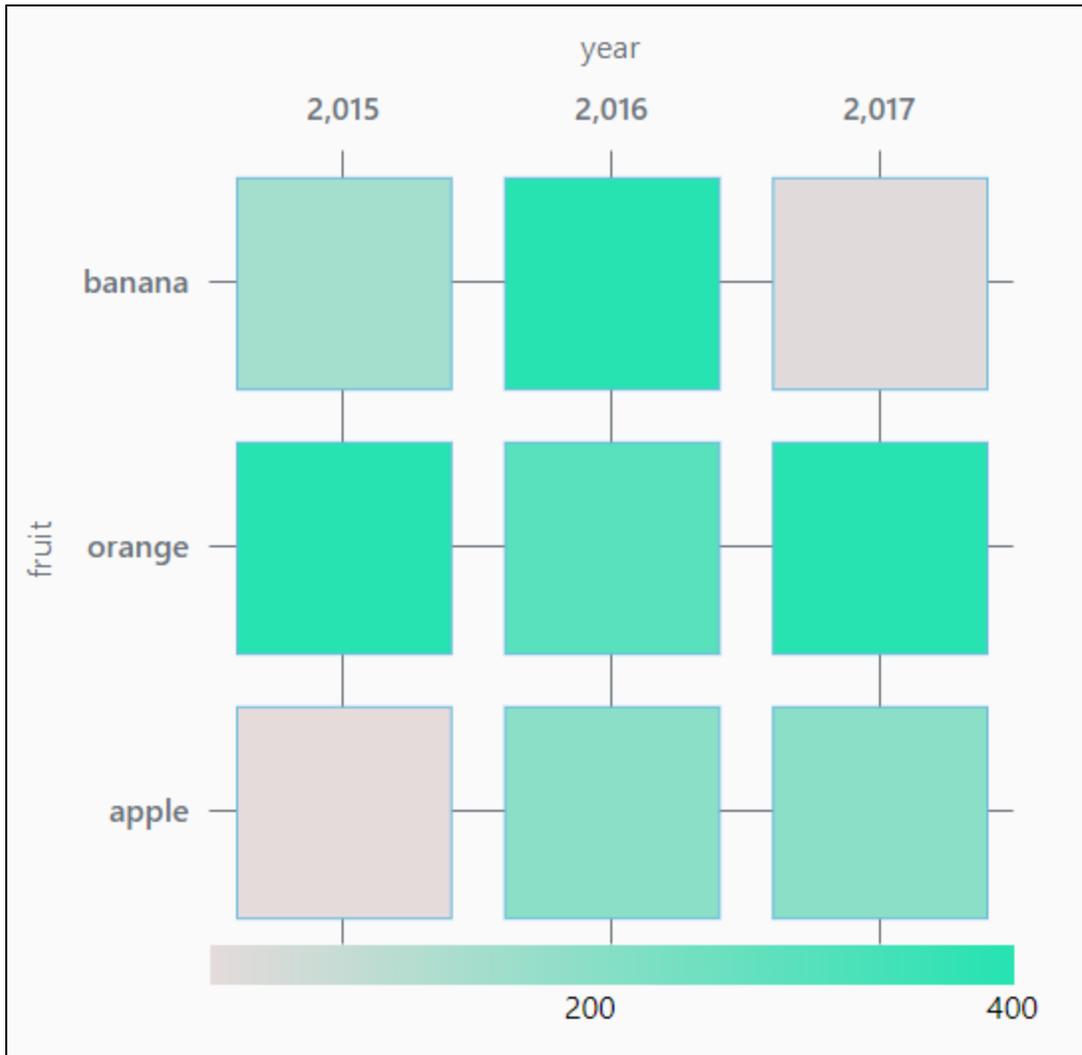
        "dx": 0,
        "dy": 0
    },
    "fill": {
        "color": "",
        "opacity": 1
    },
    "stroke": {
        "color": "",
        "opacity": 1,
        "width": 1
    },
    "rotation": 0,
    "tooltip": {
        "enabled": true,
        "text": "{name}: [bold]{valueY}[/]",
        "cornerRadius": 3,
        "pointerLength": 4,
        "background": {
            "color": "",
            "opacity": 1
        }
    },
    "deriveFieldsFromData": {
        "fill": {
            "color": "",
            "opacity": ""
        },
        "stroke": {
            "color": "",
            "opacity": "",
            "width": ""
        },
        "rotation": ""
    },
    "heatRules": {
        "enabled": false,
        "max": 100,
        "min": 2,
        "dataField": ""
    }
}
]
}
}
},
"dataSources": {
    "example": [
        {
            "name": "John",
            "fromDate": "2018-01-01 08:00",
            "toDate": "2018-01-01 10:00",
            "color": "#00FF30"
        },
        {
            "name": "John",
            "fromDate": "2018-01-01 12:00",
            "toDate": "2018-01-01 15:00",
            "color": "#FF0030"
        },
        {
            "name": "John",
            "fromDate": "2018-01-01 15:30",
            "toDate": "2018-01-01 21:30",
            "color": "#00FF30"
        },
        {
            "name": "Jane",
            "fromDate": "2018-01-01 09:00",

```

```
    "toDate": "2018-01-01 12:00",
    "color": "#00FF30"
  },
  {
    "name": "Jane",
    "fromDate": "2018-01-01 13:00",
    "toDate": "2018-01-01 17:00",
    "color": "#FF0030"
  },
  {
    "name": "Peter",
    "fromDate": "2018-01-01 11:00",
    "toDate": "2018-01-01 16:00",
    "color": "#00FF30"
  },
  {
    "name": "Peter",
    "fromDate": "2018-01-01 16:00",
    "toDate": "2018-01-01 19:00",
    "color": "#00FF30"
  }
]
}
},
"meta": {
  "name": "XYChart_1"
},
"position": {
  "x": 36.5,
  "y": 271,
  "height": 375,
  "width": 700
},
"custom": {}
}
]
```

XY Chart Example - Heat Map

This example demonstrates how to configure a Heatmap, where the value of each cross axis is represented as a color along a gradient. This style of chart is largely accomplished by setting the series "render" property to "column", and enabling "heatRules".



The data used in this example is shown in the code block below: The x-axis is set to the "year" key, while the y-axis is set to the "fruit" key. The "dataField" property under "heatRules" is set to the "count" key.

year - the x-axis

fruit - the y-axis

count - determines the color used in the block.

```
[
  {
    "year": 2015,
    "fruit": "apple",
    "count": 20
  },
  {
    "year": 2015,
    "fruit": "orange",
    "count": 400
  },
  {
    "year": 2015,
```

```

    "fruit": "banana",
    "count": 150
  },
  {
    "year": 2016,
    "fruit": "apple",
    "count": 200
  },
  {
    "year": 2016,
    "fruit": "orange",
    "count": 300
  },
  {
    "year": 2016,
    "fruit": "banana",
    "count": 400
  },
  {
    "year": 2017,
    "fruit": "apple",
    "count": 200
  },
  {
    "year": 2017,
    "fruit": "orange",
    "count": 400
  },
  {
    "year": 2017,
    "fruit": "banana",
    "count": 30
  }
]

```

Notable Property Configurations

The example requires both an X axis and a Y axis. In addition to a series

PROPS Path	Property Description	Value
xAxes.0.render	Makes the chart group values in the X Axis, and provides equal padding between each category. Partially responsible for rendering boxes on the chart.	category
yAxes.0.render	Makes the chart group values in the Y Axis, and provides equal padding between each category. Partially responsible for rendering boxes on the chart.	category
series.0.render	When combined with "category" renders for the X and Y axes, allows the categories boxes to be rendered on the chart.	column
series.0.column.appearance.heatRules.enabled	Makes the chart change the color on each category based on dataField key ("count", in our example), applying the min and max colors.	true
series.0.column.appearance.heatRules.max	The property represents which color to use for higher values.	#26E3B1
series.0.column.appearance.heatRules.min	The property represents which color to use for lower values.	#E5DBDB (pick a color you want to represent low colors)
series.0.column.appearance.heatRules.dataField	Determines which key in the underlying data should be used to determine the vibrancy of each block.	count

Example Configuration

The JSON string below can be used to replicate the heatmap example. Simply copy the contents of the code block below (**double click** on any part of the JSON to select all of it) , and paste it into a container in your designer.

```

[
  {
    "type": "ia.chart.xy",

```

```
"version": 0,
"props": {
  "legend": {
    "enabled": false
  },
  "xAxes": [
    {
      "name": "year",
      "label": {
        "enabled": true,
        "text": "year",
        "color": ""
      },
      "visible": true,
      "tooltip": {
        "enabled": true,
        "text": "",
        "cornerRadius": 3,
        "pointerLength": 4,
        "background": {
          "color": "",
          "opacity": 1
        }
      },
      "inversed": false,
      "render": "category",
      "category": {
        "break": {
          "enabled": false,
          "startCategory": "",
          "endCategory": "",
          "size": 0.05
        }
      },
      "date": {
        "baseInterval": {
          "enabled": false,
          "timeUnit": "hour",
          "count": 1,
          "skipEmptyPeriods": false
        },
        "range": {
          "max": "",
          "min": "",
          "useStrict": false
        },
        "break": {
          "enabled": false,
          "startDate": "",
          "endDate": "",
          "size": 0.05
        },
        "inputFormat": "yyyy-MM-dd kk:mm:ss",
        "format": "M/d"
      },
      "value": {
        "range": {
          "max": "",
          "min": "",
          "useStrict": false
        },
        "logarithmic": false,
        "break": {
          "enabled": false,
          "startValue": 0,
          "endValue": 100,
          "size": 0.05
        },
        "format": "#,###.##"
      },
      "appearance": {
```

```

      "opposite": true,
      "inside": false,
      "labels": {
        "color": "",
        "opacity": 1
      },
      "grid": {
        "color": "",
        "opacity": 1,
        "dashArray": "",
        "minDistance": 60,
        "position": 0.5
      },
      "font": {
        "size": "",
        "weight": 500
      }
    }
  ],
  "yAxes": [
    {
      "name": "fruit",
      "label": {
        "enabled": true,
        "text": "fruit",
        "color": ""
      },
      "visible": true,
      "tooltip": {
        "enabled": true,
        "text": "",
        "cornerRadius": 3,
        "pointerLength": 4,
        "background": {
          "color": "",
          "opacity": 1
        }
      },
      "inversed": false,
      "render": "category",
      "category": {
        "break": {
          "enabled": false,
          "startCategory": "",
          "endCategory": "",
          "size": 0.05
        }
      },
      "date": {
        "baseInterval": {
          "enabled": false,
          "timeUnit": "hour",
          "count": 1,
          "skipEmptyPeriods": false
        },
        "range": {
          "max": "",
          "min": "",
          "useStrict": false
        },
        "break": {
          "enabled": false,
          "startDate": "",
          "endDate": "",
          "size": 0.05
        },
        "inputFormat": "yyyy-MM-dd kk:mm:ss",
        "format": "M/d/yyyy HH:mm:ss"
      },
      "value": {

```

```

    "range": {
      "max": "",
      "min": "",
      "useStrict": false
    },
    "logarithmic": false,
    "break": {
      "enabled": false,
      "startValue": 0,
      "endValue": 100,
      "size": 0.05
    },
    "format": "#,###.##"
  },
  "appearance": {
    "opposite": false,
    "inside": false,
    "labels": {
      "color": "",
      "opacity": 1
    },
    "grid": {
      "color": "",
      "opacity": 1,
      "dashArray": "",
      "minDistance": null,
      "position": 0.5
    },
    "font": {
      "size": "",
      "weight": 500
    }
  }
},
],
"series": [
  {
    "name": "count",
    "label": {
      "text": "Process Temp"
    },
    "visible": true,
    "hiddenInLegend": false,
    "defaultState": {
      "visible": true
    },
    "data": {
      "source": "data",
      "x": "year",
      "y": "fruit"
    },
    "xAxis": "year",
    "yAxis": "fruit",
    "zIndex": 0,
    "tooltip": {
      "enabled": true,
      "text": "{name}: [bold]{valueY}[/]",
      "cornerRadius": 3,
      "pointerLength": 4,
      "background": {
        "color": "",
        "opacity": 1
      }
    }
  },
  "render": "column",
  "candlestick": {
    "open": {
      "x": "",
      "y": ""
    },
    "high": {

```

```

    "x": "",
    "y": ""
  },
  "low": {
    "x": "",
    "y": ""
  },
  "appearance": {
    "fill": {
      "color": "",
      "opacity": 1
    },
    "stroke": {
      "color": "",
      "opacity": 1,
      "width": 1
    },
    "stacked": false,
    "deriveFieldsFromData": {
      "fill": {
        "color": "",
        "opacity": ""
      },
      "stroke": {
        "color": "",
        "opacity": "",
        "width": ""
      }
    },
    "heatRules": {
      "enabled": false,
      "max": "",
      "min": "",
      "dataField": ""
    }
  },
  "column": {
    "open": {
      "x": "",
      "y": ""
    },
    "appearance": {
      "fill": {
        "color": "",
        "opacity": 1
      },
      "stroke": {
        "color": "",
        "opacity": 1,
        "width": 1
      },
      "stacked": false,
      "width": null,
      "height": null,
      "deriveFieldsFromData": {
        "fill": {
          "color": "",
          "opacity": ""
        },
        "stroke": {
          "color": "",
          "opacity": "",
          "width": ""
        }
      },
      "heatRules": {
        "enabled": true,
        "max": "#26E3B1",
        "min": "#E5DBDB",
        "dataField": "count"
      }
    }
  }
}

```

```

    }
  },
  "line": {
    "open": {
      "x": "",
      "y": ""
    },
    "appearance": {
      "connect": true,
      "tensionX": 1,
      "tensionY": 1,
      "minDistance": 0.5,
      "stroke": {
        "width": 3,
        "opacity": 1,
        "color": "",
        "dashArray": ""
      },
      "fill": {
        "opacity": 0,
        "color": ""
      },
    },
    "bullets": [
      {
        "enabled": false,
        "render": "circle",
        "width": 10,
        "height": 10,
        "label": {
          "text": "{value}",
          "position": {
            "dx": 0,
            "dy": 0
          }
        },
        "fill": {
          "color": "",
          "opacity": 1
        },
        "stroke": {
          "color": "",
          "opacity": 1,
          "width": 1
        },
        "rotation": 0,
        "tooltip": {
          "enabled": true,
          "text": "{name}: [bold]{valueY}[/]",
          "cornerRadius": 3,
          "pointerLength": 4,
          "background": {
            "color": "",
            "opacity": 1
          }
        },
        "deriveFieldsFromData": {
          "fill": {
            "color": "",
            "opacity": ""
          },
          "stroke": {
            "color": "",
            "opacity": "",
            "width": ""
          },
          "rotation": ""
        },
        "heatRules": {
          "enabled": false,
          "max": 100,

```

```

        "min": 2,
        "dataField": ""
    }
}
]
},
"stepLine": {
    "open": {
        "x": "",
        "y": ""
    },
    "appearance": {
        "connect": true,
        "tensionX": 1,
        "tensionY": 1,
        "minDistance": 0.5,
        "stroke": {
            "width": 3,
            "opacity": 1,
            "color": "",
            "dashArray": ""
        },
        "fill": {
            "opacity": 0,
            "color": ""
        },
    },
    "bullets": [
        {
            "enabled": true,
            "render": "circle",
            "width": 10,
            "height": 10,
            "label": {
                "text": "{value}",
                "position": {
                    "dx": 0,
                    "dy": 0
                }
            },
            "fill": {
                "color": "",
                "opacity": 1
            },
            "stroke": {
                "color": "",
                "opacity": 1,
                "width": 1
            },
            "rotation": 0,
            "tooltip": {
                "enabled": true,
                "text": "{name}: [bold]{valueY}[/]",
                "cornerRadius": 3,
                "pointerLength": 4,
                "background": {
                    "color": "",
                    "opacity": 1
                }
            }
        },
    ],
    "deriveFieldsFromData": {
        "fill": {
            "color": "",
            "opacity": ""
        },
        "stroke": {
            "color": "",
            "opacity": "",
            "width": ""
        },
        "rotation": ""
    }
}

```

```

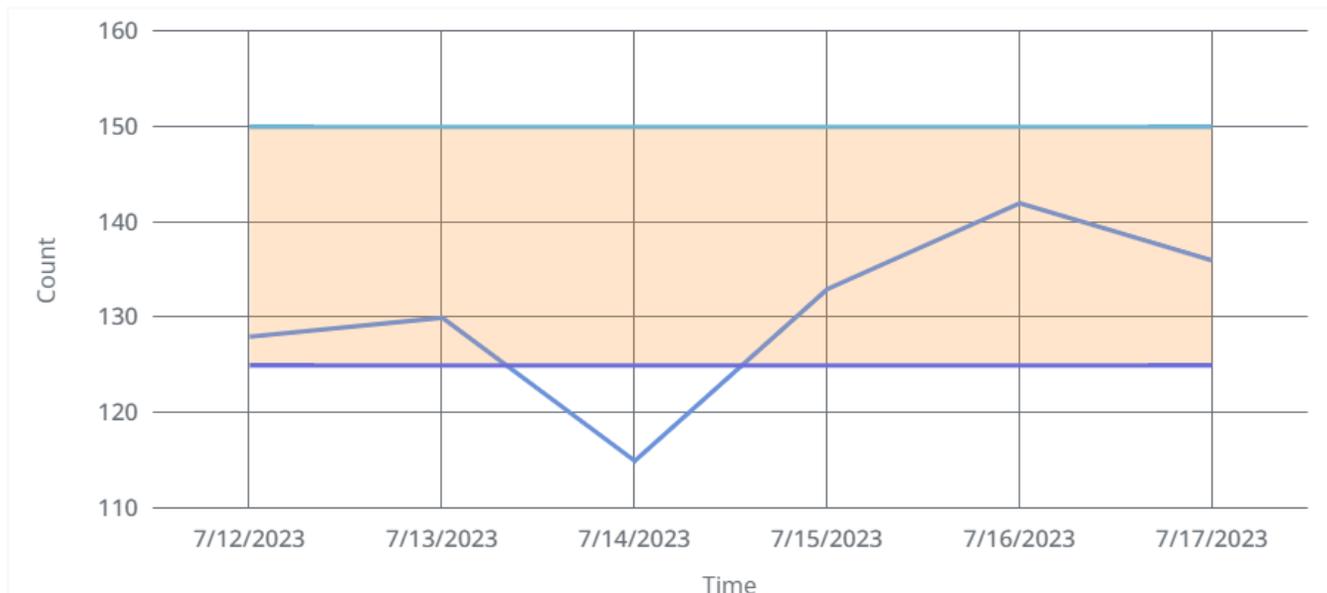
        },
        "heatRules": {
            "enabled": false,
            "max": 100,
            "min": 2,
            "dataField": ""
        }
    ]
}
},
],
"style": {
    "marginRight": "100px",
    "paddingRight": "100px"
},
"dataSources": {
    "data": [
        {
            "year": 2015,
            "fruit": "apple",
            "count": 20
        },
        {
            "year": 2015,
            "fruit": "orange",
            "count": 400
        },
        {
            "year": 2015,
            "fruit": "banana",
            "count": 150
        },
        {
            "year": 2016,
            "fruit": "apple",
            "count": 200
        },
        {
            "year": 2016,
            "fruit": "orange",
            "count": 300
        },
        {
            "year": 2016,
            "fruit": "banana",
            "count": 400
        },
        {
            "year": 2017,
            "fruit": "apple",
            "count": 200
        },
        {
            "year": 2017,
            "fruit": "orange",
            "count": 400
        },
        {
            "year": 2017,
            "fruit": "banana",
            "count": 30
        }
    ]
}
},
"meta": {
    "name": "XYChart"
},
"position": {

```

```
    "basis": "536px"  
  },  
  "custom": {}  
}  
]
```

XY Chart Example - Line Chart Target Area

This example demonstrates how to utilize the `line.open` property to create a target zone. Target zones can be used to represent acceptable values for the data that is displayed to make it obvious if values are good or bad.

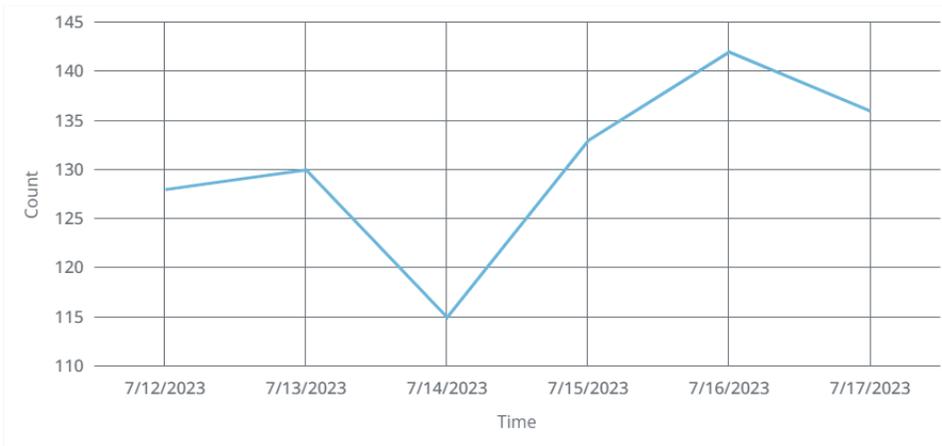


Although, there are many ways to enter data values and high/low limit values for a line chart, the data used in this example is shown in the code block below. Copy the JSON content below and paste it onto the Chart component's `dataSources.example` property.

```
[
  {
    "date": "2023-7-12 07:00:00",
    "actual": 128,
    "highLimit": 150,
    "lowLimit": 125
  },
  {
    "date": "2023-7-13 07:00:00",
    "actual": 130,
    "highLimit": 150,
    "lowLimit": 125
  },
  {
    "date": "2023-7-14 07:00:00",
    "actual": 115,
    "highLimit": 150,
    "lowLimit": 125
  },
  {
    "date": "2023-7-15 07:00:00",
    "actual": 133,
    "highLimit": 150,
    "lowLimit": 125
  },
  {
    "date": "2023-7-16 07:00:00",
    "actual": 142,
    "highLimit": 150,
    "lowLimit": 125
  },
  {
    "date": "2023-7-17 07:00:00",
    "actual": 136,
    "highLimit": 150,
    "lowLimit": 125
  }
]
```

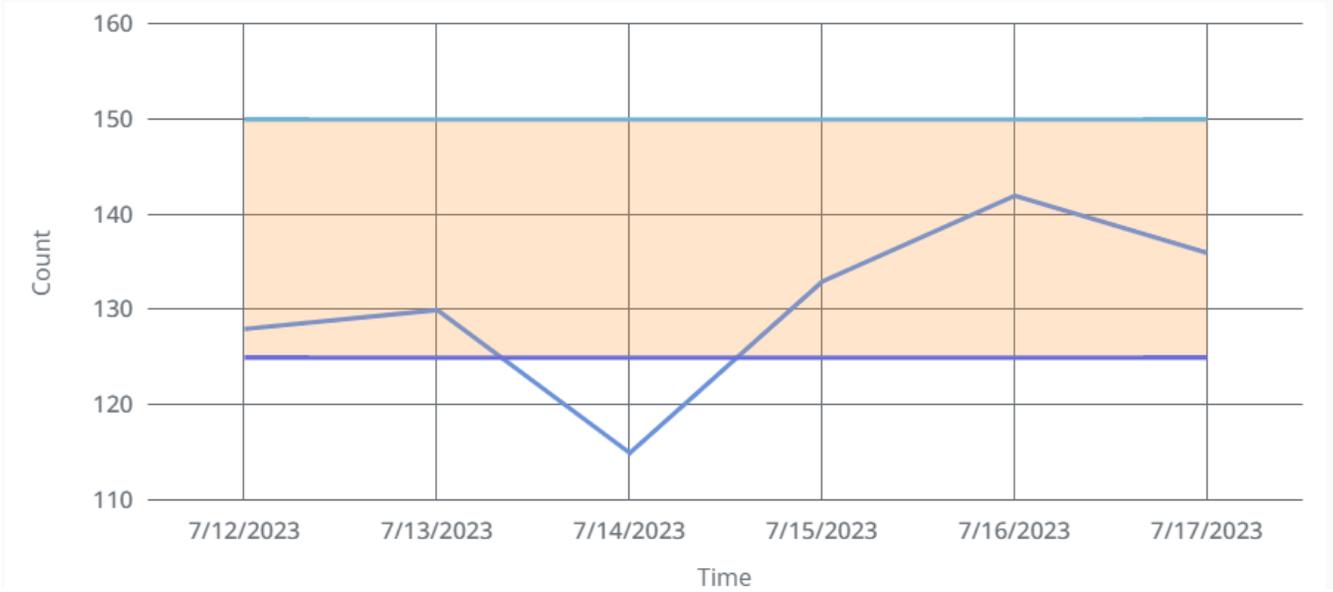
Before creating a target zone, we must first adjust some of the default XY Chart settings to display the **actual** values from our example data source on the line chart:

1. Remove the legend by setting **legend.enabled** to false.
2. Navigate to the xAxis property to set the **xAxis.0.name** and **xAxis.0.label.text** properties. In this example, the name will be **time** and the label.text property will be set to **Time**.
3. Make sure the **xAxes.0.render** property is set to **date** and set your **xAxes.0.date** format if needed. This example uses **date** for the **xAxes.0.date.inputFormat** and **xAxes.0.date.format** properties.
4. Navigate to the yAxes property and delete the **yAxes.1** array that is loaded by default.
5. Then, set the **yAxes.0.name** and **label.text** properties. In this example, the name will be **count** and the label.text property will be set to **Count**.
6. Now, set the following **series.0** properties to display the **actual** values:
 - **name**: actual
 - **data.source**: example
 - **data.x**: time
 - **data.y**: actual
 - **xAxis**: time
 - **yAxis**: count



Now, you'll see a line chart that displays the actual values. However, there is still no clear way to tell if these values are good or bad. This will be achieved by creating two more series using the high and low limit data already built into our data source.

1. Create a new series and set the following **series.1** properties to display the acceptable high limit area.
 - **name**: highLimit
 - **data.source**: example
 - **data.x**: time
 - **data.y**: highLimit
 - **xAxis**: time
 - **yAxis**: count
 - **render**: line
 - **line.open.y**: lowLimit
 - **line.appearance.fill.color**: #FF8C00
 - **line.appearance.fill.opacity**: 0.125
2. Duplicate series.1 and change the following **series.2** properties to display the acceptable low limit area.
 - **name**: lowLimit
 - **data.y**: lowLimit
 - **line.open.y**: highLimit



With the high and low limit areas displayed, it is now easy for any user to see that all counts were within the acceptable range, except for on 7/14/2023.

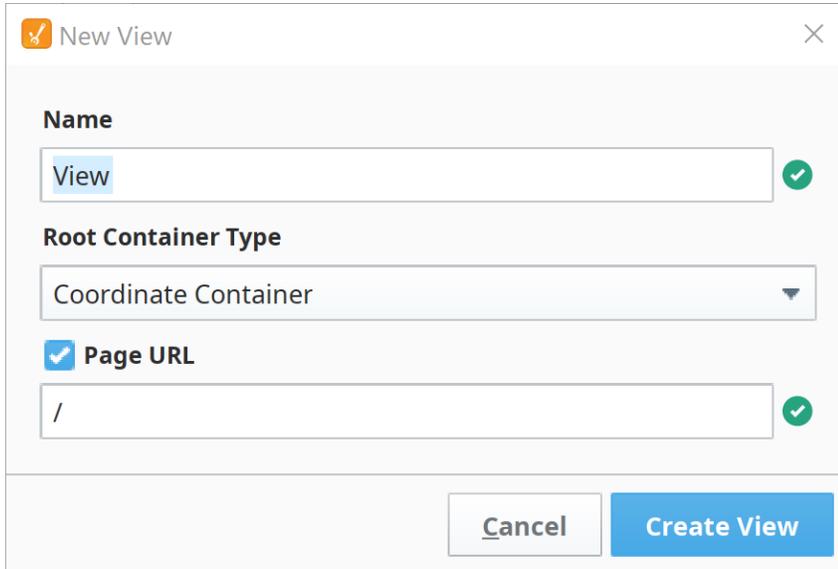
Perspective - Container Palette

Container Components

Container components provide a way of laying out and organizing components within a View. The different container types support different layout strategies. Containers are essential to creating responsive applications, and they allow your applications to gracefully display information across a wide variety of screen sizes and orientations.

The following is a complete list of Container components, and a link pointing to a page containing the component's description, properties, and an example of how to configure it.

The Coordinate Container is the default container type when creating a new View in a project. A different container type can be selected using the dropdown arrow to display the full list of container component options.



Name
View ✓

Root Container Type
Coordinate Container ▼

Page URL
/ ✓

The following feature is new in Ignition version **8.1.22**
[Click here](#) to check out the other new features

After the initial View is created, all new Views will default to the last used container type selected in the Root Container Type field. For example, if the last View you created used the Flex Container, the next time you create a new View the Root Container Type will default to Flex Container instead of Coordinate Container.

The following is a complete list of Container components, and a link pointing to a page containing the component's description, properties, and an example of how to configure it.

[In This Section ...](#)

Perspective - Breakpoint Container



On this page ...

- [Properties](#)
- [Child Component Position Properties](#)
- [Scripting](#)
- [Example \(Nested Breakpoint Containers\)](#)

Component Palette Icon:



The Breakpoint container consists of a single *breakpoint*, with two child views. In other words, using a Breakpoint container offers you a layout with the opportunity to create two different views that are shown at two distinct ranges of layout widths.

This allows for a very simple responsive design that removes one container and replaces it with another when the viewport width is changed. With the Breakpoint container, you can define completely different content to render for each child. This is in contrast to a container such as the column container, where the content for each breakpoint is identical but the layout of it changes according to screen size.

Breakpoint containers are ideal to use in cases where you want completely different components available between a mobile user or a desktop user, since the components in each breakpoint are completely separate instances. Thus, if you're attempting to make a responsive view, and a Flex container isn't quite giving the desired result, a Breakpoint container can be used to switch to a completely different container or view.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
breakpoint	Width (in pixels) breakpoint declarations for child layouts. When the container is sized below minWidth, child position rules will fall back to the next set breakpoint rules.	value: numeric
determinant	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 5px;"> <p>The following feature is new in Ignition version 8.1.32 Click here to check out the other new features</p> </div> <p>Choose between the height or width dimension that the Breakpoint container will use to break. Default is width.</p>	value: string
style	Use Style to customize the visual style of the component. The Style menu contains all the tools for modifying text, background, margins, and borders. You can also specify a style class .	object

Child Component Position Properties

When a component is placed inside of a Breakpoint container, it will inherit the position property listed below.

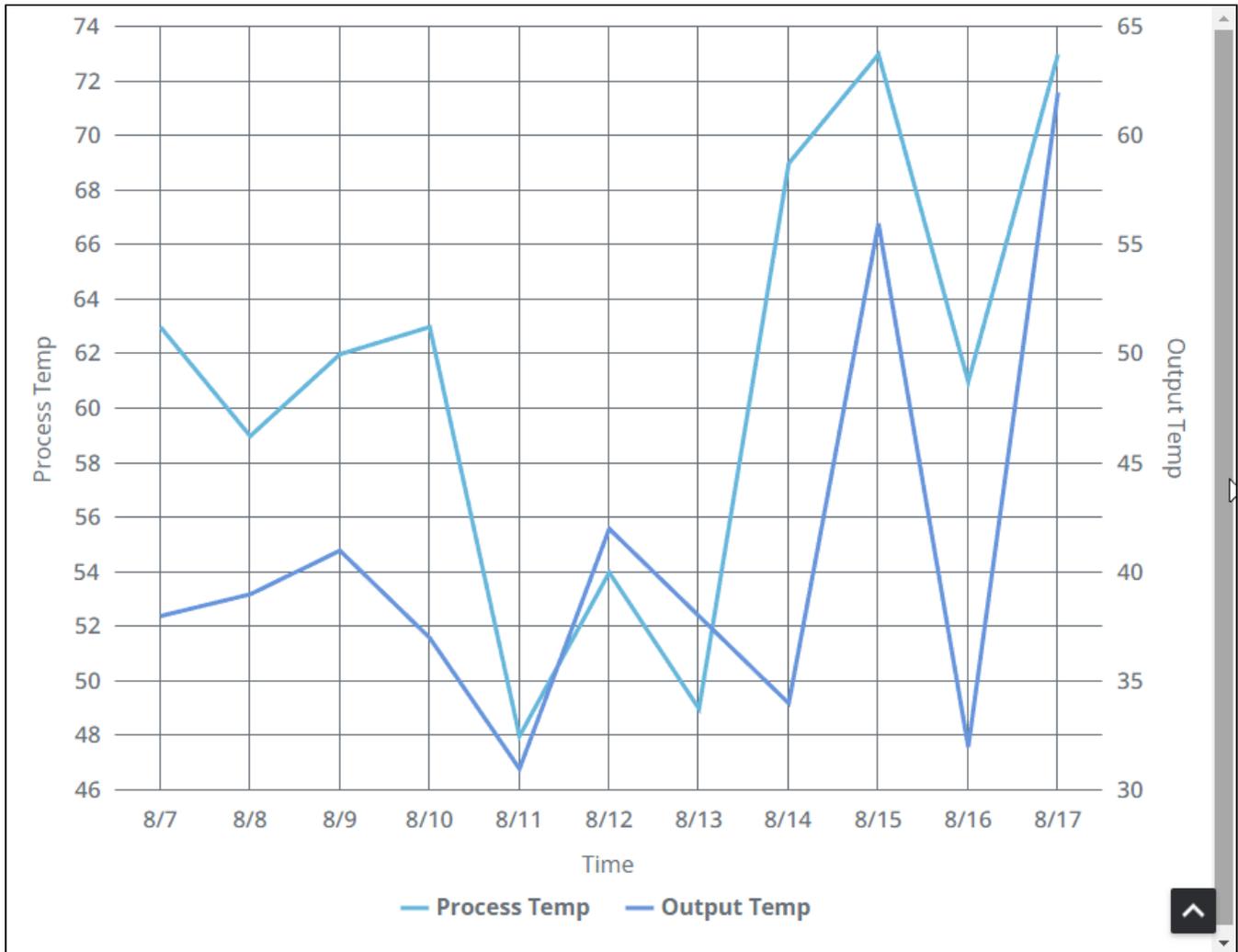
--	--	--

Property	Description	Data Type
size	Indicates which child the component located in. Expected values are "small" and "large".	value: string

Scripting

See the [Perspective - Breakpoint Container Scripting](#) page for the full list of scripting functions available for this component.

Example (Nested Breakpoint Containers)



In this example, we have a Breakpoint container nested inside a root Breakpoint container. The setup is as follows:

Containers

- The nested Breakpoint container is used when the root Breakpoint container's size is less than 800 pixels.
- The nested Breakpoint container will change when its size is less than 700 pixels.

Components

- When the page is greater than 800 pixels, an XY Chart is displayed.
- When the page is between 700 and 800 pixels, an empty Power Chart is shown.
- When the page is less than 700 pixels, an Icon is displayed.

The relevant properties for this example are displayed below:

Root Breakpoint Container Properties:

Property	Value

Breakpoint	800px
------------	-------

Nested Breakpoint Container Properties:

Property	Value
Breakpoint	700px

Perspective - Breakpoint Container Scripting

This page details the various scripting, component, and extension functions available for [Perspective's Breakpoint Container](#) component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

On this page ...

- [Component Events](#)
- [Component Functions](#)
 - [.getChildren\(\)](#)
- [Extension Functions](#)

Component Functions

.getChildren()

- Description

Returns an ArrayList, which contains references to all components inside of the container.

- Parameters

None

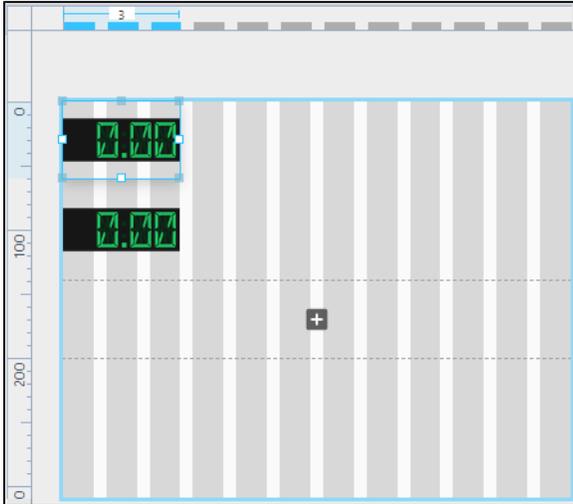
- Return

[Array List](#) - An ArrayList of components in the container. The resulting ArrayList can be iterated over via a for-loop.

Extension Functions

This component does not have extension functions associated with it.

Perspective - Column Container



On this page ...

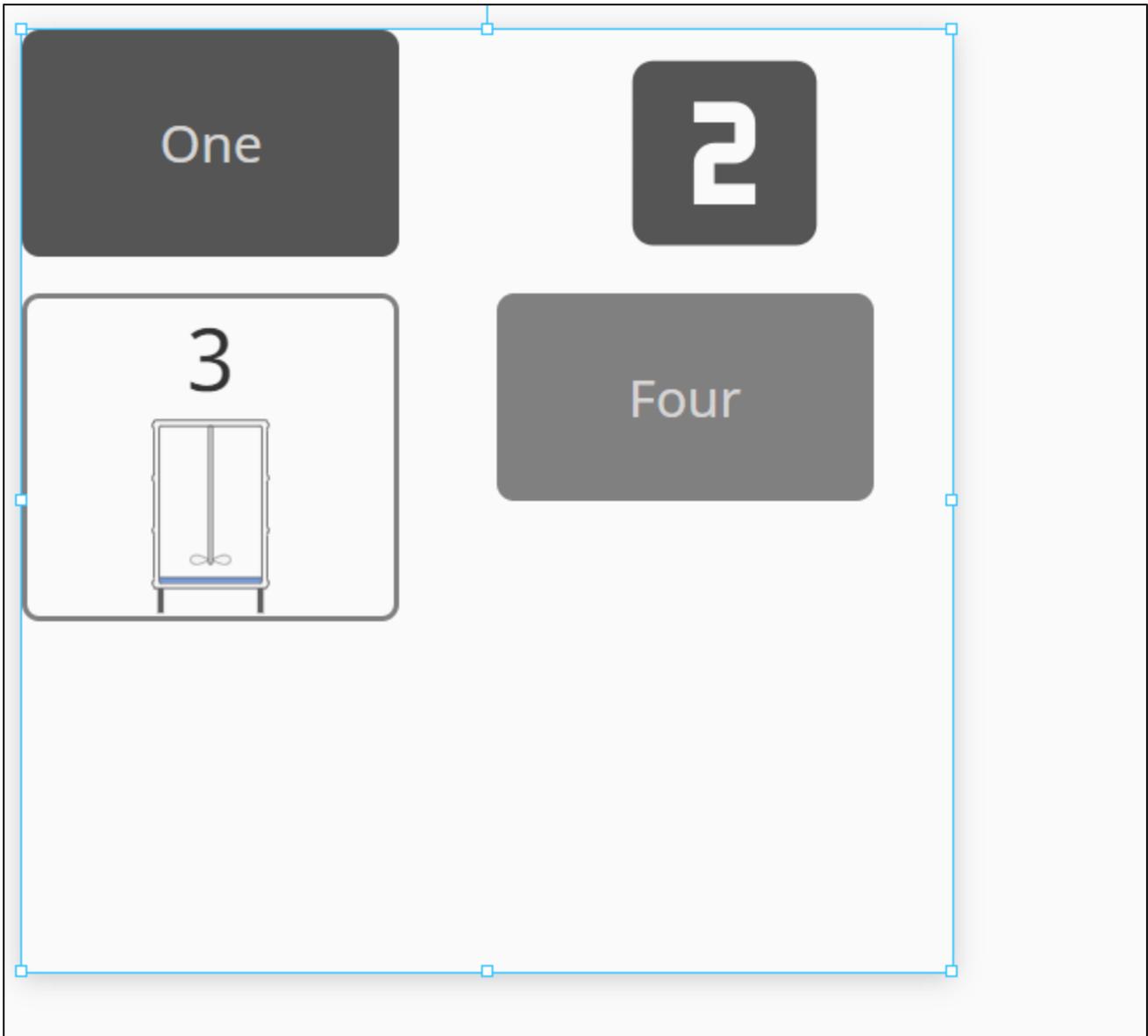
- [User Interface](#)
- [Properties](#)
- [Child Component Position Properties](#)
- [Scripting](#)

Component Palette Icon:



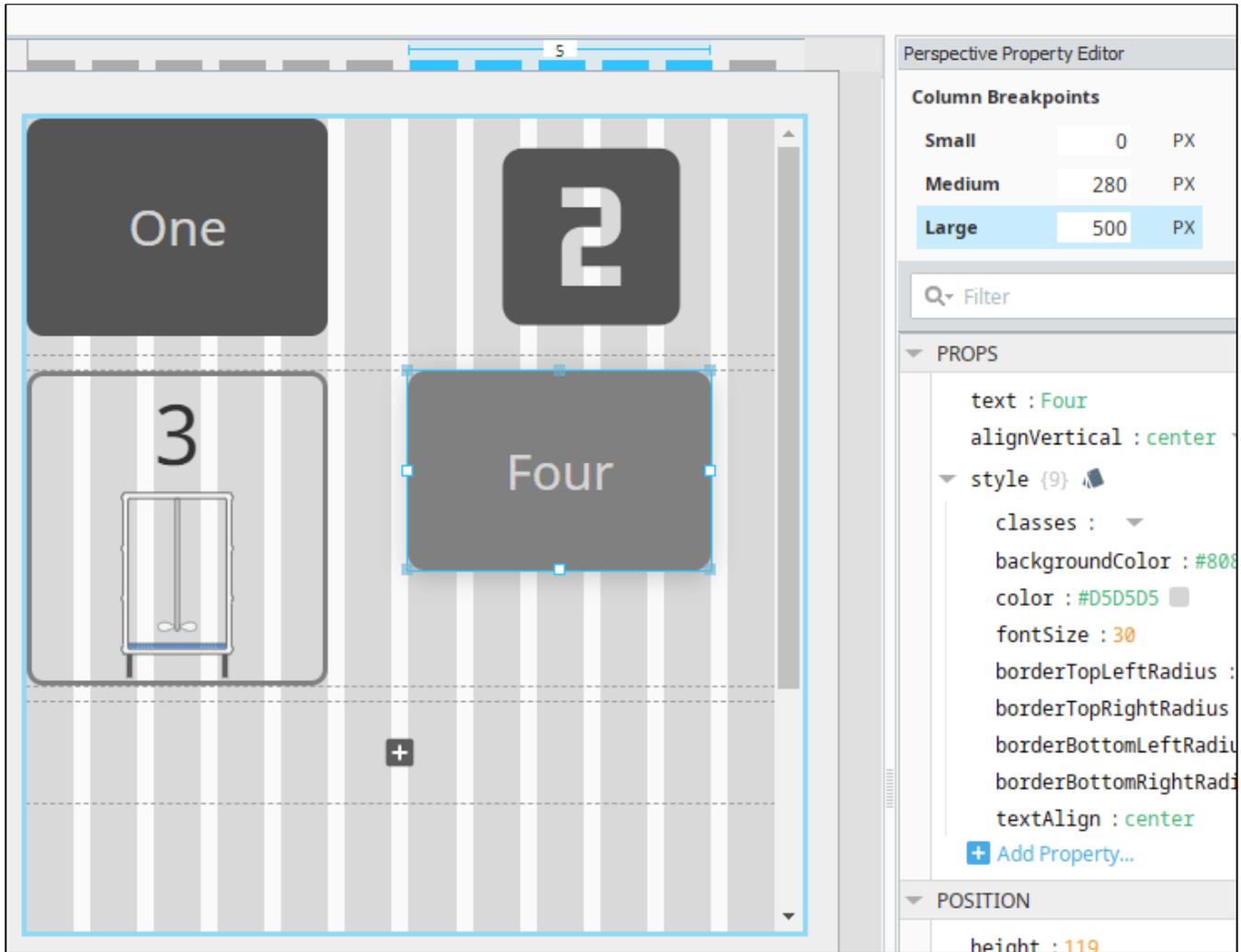
The Column Container is a 12 column grid layout, where components can be organized into columns which alter their layout depending on screen viewport size. The container features three different breakpoints that it can switch between as the viewport changes. Each component inside the container stores positioning and sizing values per breakpoint, allowing you to determine the location and size of each component for each breakpoint.

A column container provides a way to create a single set of components that can be arranged up to three ways depending on the width of the container. Components are able to change how many columns they span when switching between breakpoints, as well as change row and column index altogether. In the GIF below, the components are resizing and repositioning as the viewport on the container changes.



User Interface

While deep selected, the Perspective Property Editor will show a **Column Breakpoints** interface. Clicking on **Small**, **Medium**, or **Large** will change the container workspace, showing you the current configurations for all child components in the container. When a container workspace is selected, you can freely move and resize components for the selected breakpoint, allowing you to determine their placement when the selected breakpoint is active in a session.



Note that switching between breakpoints in the designer does not resize the container. The pixel dimensions represent a minimum width for the viewport, meaning that the container will switch to the breakpoint when the minimum width is satisfied during runtime. For example, if Medium is set to 280px and Large is 500px, then the container will show the medium breakpoint when the viewport is between 280 and 499 pixels in width.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type									
breakpoints	Width breakpoint declarations for child layouts. When the container is sized below minWidth, child position rules will fall back to the next set breakpoint rules.	array									
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>Name of the breakpoint.</td> <td>value: string</td> </tr> <tr> <td>minWidth</td> <td>Minimum width for this breakpoint.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	name	Name of the breakpoint.	value: string	minWidth	Minimum width for this breakpoint.	value: numeric	
Name	Description	Property Type									
name	Name of the breakpoint.	value: string									
minWidth	Minimum width for this breakpoint.	value: numeric									
gutters	Settings for the gutters, which are the amount of space, in pixels, to place between child components.	object									
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>vertical</td> <td>Vertical gutter setting between child components.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	vertical	Vertical gutter setting between child components.	value: numeric				
Name	Description	Property Type									
vertical	Vertical gutter setting between child components.	value: numeric									

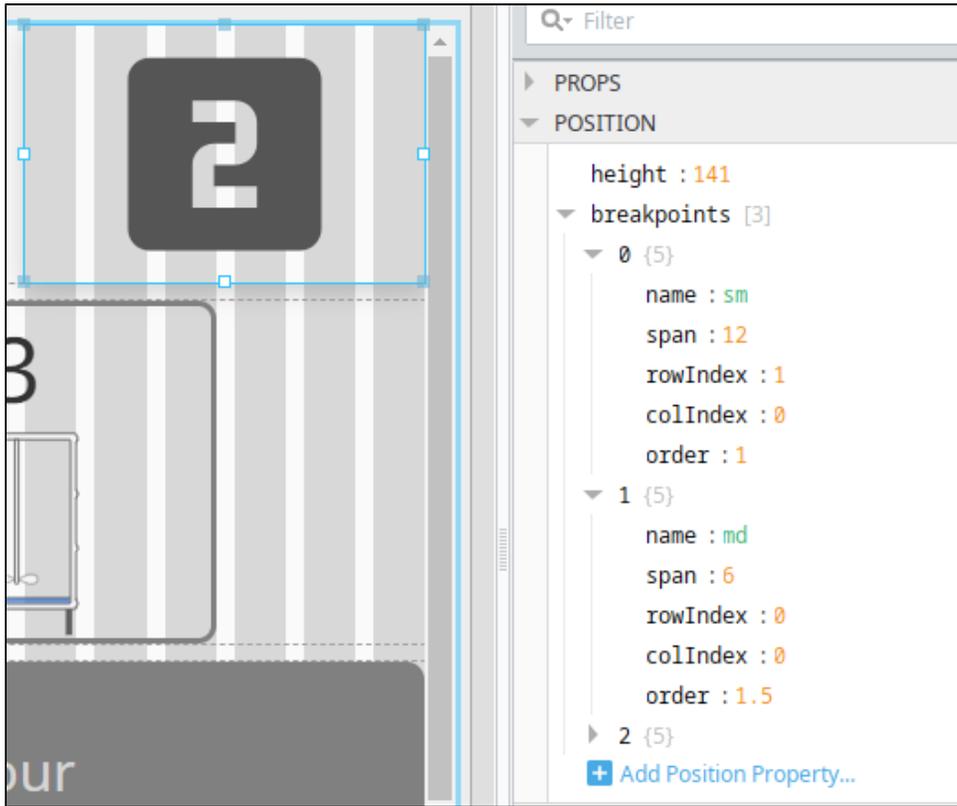
	horizontal	Horizontal gutter setting between child components.	value: numeric	
style	Sets a style for this component. The Style menu contains all the tools for modifying text, background, margins, and borders. You can also specify a style class .			object

Child Component Position Properties

When a component is placed inside of a column container, it will inherit the position properties listed below.

Property	Description	Data Type
height	Specifies the horizontal positioning of the component in pixels.	value: numeric

breakpoints An array of objects, where each object represents a different breakpoint. In the image below, a component in a column container is selected. Each element under **breakpoints** contains positioning information that will be used when the container switches to the named breakpoint.



Each object has the following values:

Property	Description	Data Type
name	Name of a breakpoint defined in container. If this matches the currently applied breakpoint, these rules determine child layout. Options are sm (small), md (medium), or lg (large).	value: string
span	Number of columns the child's width will span.	value: numeric
rowIndex	Row index (starting from 0) in which to place child. Children may wrap lines within a row. Children in separate rows don't affect each other's layout.	value: numeric
colIndex	Column number upon which the child's span should begin unless forced to wrap.	value: numeric
order	Where component is places among its siblings within its row. Ordering is independent per row.	value: numeric

Scripting

See the [Perspective - Column Container Scripting page](#) for the full list of scripting functions available for this component.

Perspective - Column Container Scripting

This page details the various scripting, component, and extension functions available for [Perspective's Column Container](#) component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

On this page ...

- [Component Events](#)
- [Component Functions](#)
 - [.getChildren\(\)](#)
- [Extension Functions](#)

Component Functions

.getChildren()

- Description

Returns an ArrayList, which contains references to all components inside of the container.

- Parameters

None

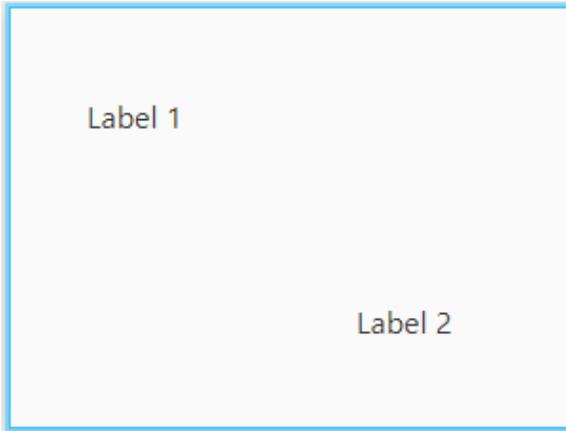
- Return

[Array List](#) - An ArrayList of components in the container. The resulting ArrayList can be iterated over via a for-loop.

Extension Functions

This component does not have extension functions associated with it.

Perspective - Coordinate Container



On this page ...

- [Properties](#)
- [Child Component Position Properties](#)
- [Scripting](#)
- [Example](#)

Component Palette Icon:



The Coordinate Container makes a component's size and location relative to its parent's size and location. Components can be fixed size, or optionally grow/shrink proportionally when the view is stretched.

Coordinate Containers are ideal to use in cases where you need components to overlap each other, such as adding a component on top of another (z-axis) to act as an overlay. They're also useful in cases where you do **not** want components within to resize - for example, building a diagram where each element is a separate component.

Components placed in coordinate containers can be rotated. The Rotate property has been moved to the Position Properties section of the Perspective Property Editor. For more information, see [Working with Perspective Components](#).

The following feature is new in Ignition version **8.1.2**
[Click here](#) to check out the other new features

The Coordinate Container component has two pre-configured [variants](#):

- Fixed - Child layouts will be in fixed coordinate space.
- Percent - Child layout will be stretched to different size containers.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Props		
Name	Description	Property Type
mode	Whether child layouts should always be in fixed coordinate space, or stretched relative to different container sizes: fixed or percent. Fixed mode uses absolute units, which reduces the amount of resizing of components within the container when launched across different display sizes. Percent mode uses relative units, which means child components will be able to resize appropriately when launched on different sized displays.	value: string
aspectRatio	Only applied in percent mode. Optional dimensions, in x:y format to apply to maintain container aspect ratio for different sizes. Empty string (or non x:y input) will disable this mode.	value: string
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object

Child Component Position Properties

When a component is placed inside of a coordinate container, it will inherit the position properties listed below.

Note that the values for x, y, width, and height will differ based on the mode of the coordinate container.

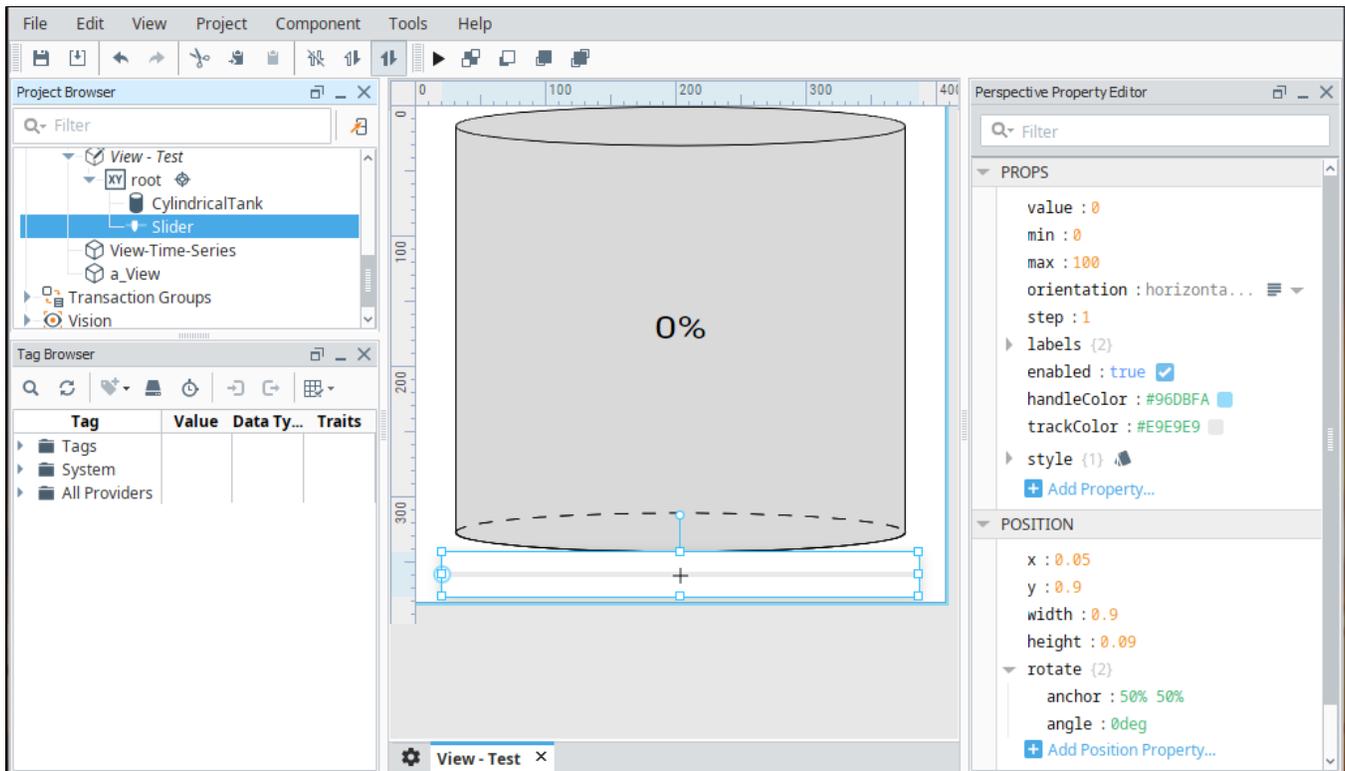
- fixed: Values represent absolute lengths. Example, an x value of 100px means the left edge of the component starts 100 pixels from the left edge of the container.
- percent: Values are in relative lengths. Example, say a component has an x value of 0.25. In this scenario, the left edge of the container would be 0, the right edge would be 1. So 0.25 would be 25% distance from the left edge of the container.

Property	Description	Data Type									
x	Specifies the horizontal positioning of the component in pixels.	value: numeric									
y	Specifies the vertical positioning of the component in pixels.	value: numeric									
width	Specifies the width of the component in pixels.	value: numeric									
height	Specifies the height of the component in pixels.	value: numeric									
rotate	Setting that sets the anchor and angle of rotation for the component.	object									
	<table border="1"><thead><tr><th>Property</th><th>Description</th><th>Data Type</th></tr></thead><tbody><tr><td>anchor</td><td>The point around which the rotation happens. Either as an {x:number, y:number} object where x and y represent percentages such that {x:0, y:0} represents the (0%, 0%) or top-left corner of the component, or as a valid CSS transform-origin string.</td><td>value: string, or object</td></tr><tr><td>angle</td><td>How much to rotate the component. Valid values include numbers (as degrees), and valid CSS angle strings such as '45deg', '2rad', '0.5turn', etc.</td><td>value: numeric or string</td></tr></tbody></table>	Property	Description	Data Type	anchor	The point around which the rotation happens. Either as an {x:number, y:number} object where x and y represent percentages such that {x:0, y:0} represents the (0%, 0%) or top-left corner of the component, or as a valid CSS transform-origin string.	value: string, or object	angle	How much to rotate the component. Valid values include numbers (as degrees), and valid CSS angle strings such as '45deg', '2rad', '0.5turn', etc.	value: numeric or string	
Property	Description	Data Type									
anchor	The point around which the rotation happens. Either as an {x:number, y:number} object where x and y represent percentages such that {x:0, y:0} represents the (0%, 0%) or top-left corner of the component, or as a valid CSS transform-origin string.	value: string, or object									
angle	How much to rotate the component. Valid values include numbers (as degrees), and valid CSS angle strings such as '45deg', '2rad', '0.5turn', etc.	value: numeric or string									

Scripting

See the [Perspective - Coordinate Container Scripting](#) page for the full list of scripting functions available for this component.

Example



In this example, we have a Coordinate container with a Cylindrical Tank component and a Slider component. We've set the container property to percent so that the components will grow and shrink with the container size.

Container properties:

Property	Value
props.mode	percent

Cylindrical Tank properties:

Property	Value
position.x	0.05
position.y	0
position.width	0.9
position.height	0.9

Slider properties:

Property	Value
position.x	0.05
position.y	0.9
position.width	0.9
position.height	0.09

Perspective - Coordinate Container Scripting

This page details the various scripting, component, and extension functions available for [Perspective's Coordinate Container](#) component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

On this page ...

- [Component Events](#)
 - [onPipeClicked](#)
- [Component Functions](#)
 - [.getChildren\(\)](#)
- [Extension Functions](#)

onPipeClicked

This event will trigger when a pipe in this container is clicked.



This component event is designed to be used in tandem with a script runAction. Within the script action, special properties and methods are available on the event object, which is passed to the script action as a parameter.

event.event

- Object Path

event.event

- Type

[Dictionary](#)

- Description

The event object generated from the mouse click.

event.pipeIndex

- Object Path

event.pipeIndex

- Type

[Integer or float](#)

- Description

The array index of the pipe within props.pipes that was clicked.

event.pipeName

- Object Path

event.pipeName

- Type

[String](#)

- Description

The name of the pipe that was clicked.

Component Functions

.getChildren()

- Description

Returns an ArrayList, which contains references to all components inside of the container.

- Parameters

None

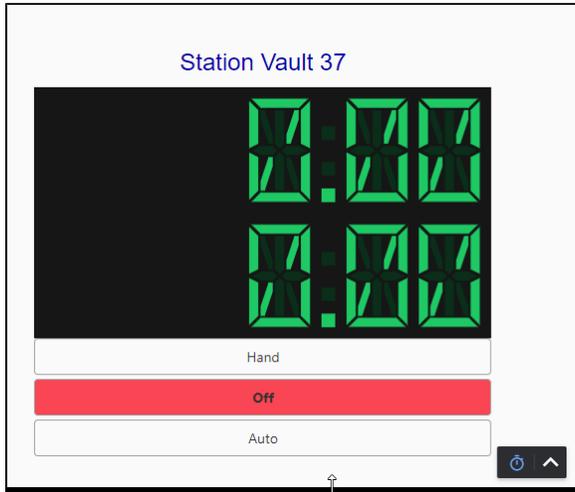
- Return

[Array List](#) - An ArrayList of components in the container. The resulting ArrayList can be iterated over via a for-loop.

Extension Functions

This component does not have extension functions associated with it.

Perspective - Flex Container



On this page ...

- [Properties](#)
- [Child Component Position Properties](#)
- [Growing and Shrinking](#)
- [Static Widths](#)
 - [Even Scaling](#)
- [User Interface](#)
 - [Direction](#)
 - [Direction: Row](#)
 - [Direction: Column](#)
- [Scripting](#)
- [Example](#)

Component Palette Icon:



The flex container provides an efficient way to lay out, align, and distribute space among components in the container particularly when their size is unknown or dynamic. The Flex Container can alter a component's width and height to best fill the available space to accommodate all types of devices and screen sizes. It expands components to fill available free space, or shrinks them to prevent overflow.

The container works off of two axes: main axis and cross axis. Components will fill out the container along the main axis. Additional alignment and justification of the components are determined by the **alignItems**, **alignContent**, and **justify** properties.

The flex container is based off the [CSS flexbox](#).

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
direction	Direction of the child layout. Options are row, row-reverse, column, column-reverse.	value: string
wrap	Whether the container should allow children to wrap to the next line if space has run out. Options are nowrap, wrap, wrap-reverse.	value: string
justify	Adjusts placement of children along the main axis when there is extra space, which may be used to fill areas before, after, or in-between: flex-start, flex-end, center, space-between, space-around, space-evenly.	value: string
alignItems	Adjusts placement of children along the cross axis when there is extra space: flex-start, flex-end, center, baseline, stretch.	value: string
alignContent	Adjusts alignment of wrapped content when there is free space in the cross axis: flex-start, flex-end, center, space-between, space-around, stretch.	value: string
style	Use Style and Classes to customize the visual style of the component. The Style menu contains all the tools for modifying text, background, margins, borders, and more. You can also specify a style class .	object

Child Component Position Properties

When a component is placed inside of a flex container, it will inherit the position properties listed below.

--

Property	Description	Data Type
grow	Ability to grow in the direction dimension as needed, relative to siblings. If space is available and grow is not zero, it may stretch, depending on sibling rules. This value is relative to other components, meaning that two components with the same grow value will grow at the same rate.	value: numeric
shrink	Ability to shrink in direction dimension as needed, relative to siblings. If space is available and grow is not zero, it may stretch, depending on sibling rules. This value is relative to other components, meaning that two components with the same shrink value will shrink at the same rate.	value: numeric
basis	Space filled by component by default, before grow , shrink and sibling considerations are evaluated. This is the component's base width when the direction property is set to row , and it is the component's base height when the direction property is set to column .	value: numeric
display	Determines if the component will be displayed in the container or not. Components that are not displayed won't just be invisible, but will actually be removed from the Flex container, readjusting all other components to fit.	value: boolean

Editor notes are only visible to logged in users
Wait for publish since these changes were reverted

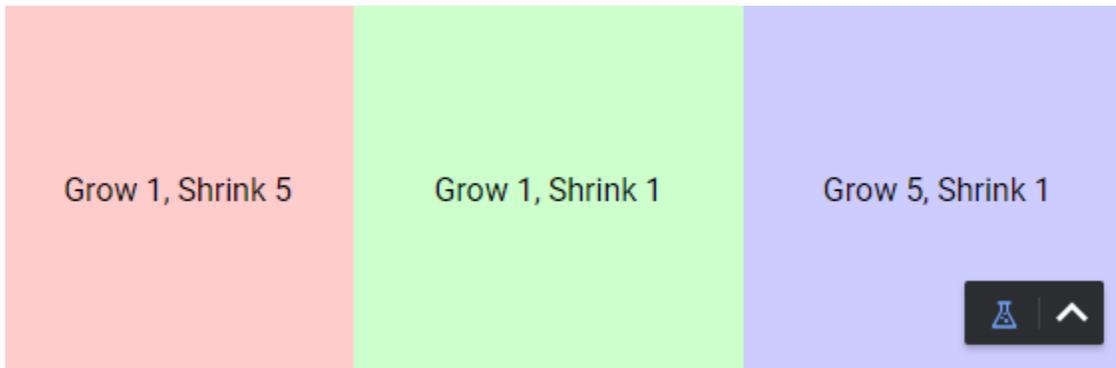
This feature was changed in Ignition version 8.1.19:

The following Perspective Components will no longer have 0 width then they are the child of a Flex Container with `props.alignItems` not equal to stretch:

- Cylindrical Tank
- File Upload
- Flex Repeater
- Icon
- Image
- Map
- Moving Analog Indicator
- LED Display
- Progress Bar
- Slider
- Sparkline
- Table
- Thermometer
- Embedded View
- View Canvas

Growing and Shrinking

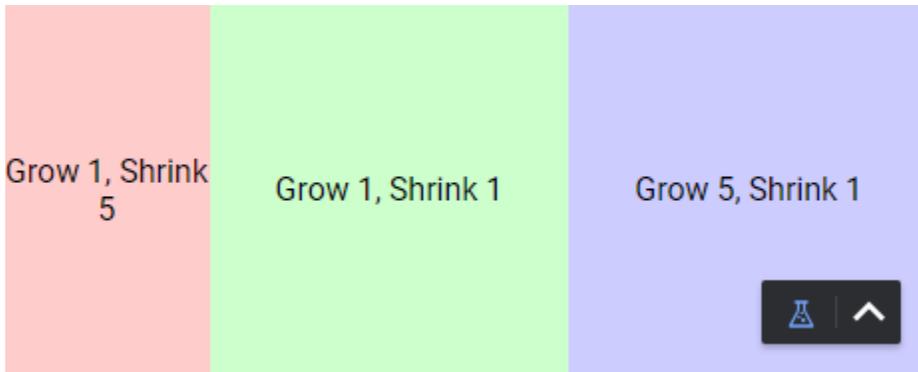
The Flex container's **grow** and **shrink** properties allow a great deal of control over how different components expand and shrink. To demonstrate, here are three components nested inside a Flex container, with a direction of row:



All three components have the same basis, so at this session width, they all have the same size. However, the blue component has a much larger grow value than the other two components. So, when we increase the length of the Flex container along its direction property:



The blue component does most of the stretching. Specifically, since the sum of all grow properties is $1 + 1 + 5 = 7$, and the blue component has a grow property of 5, for every 7 pixels the Flex container grows, the blue component will grow by 5 pixels. Now let's try shrinking the container, noting that the red component has a shrink value of 5:



As you can see, a larger shrink value will make the container shrink more. For every 7 pixels the Flex container loses, the red component will lose 5 pixels.

Static Widths

Now let's try the same example again, but with some different **grow** and **shrink** values:

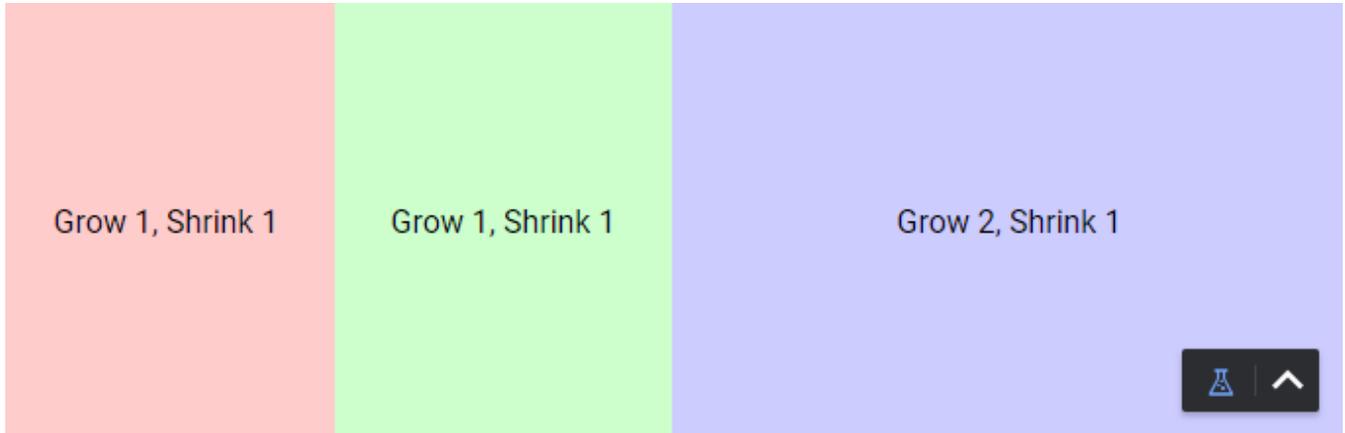


We've given the red and green components identical values, so they should stretch and shrink at the same rate. Meanwhile, the blue component's grow and shrink values are both 0, so when we make the Flex container wider, it stays the same size:

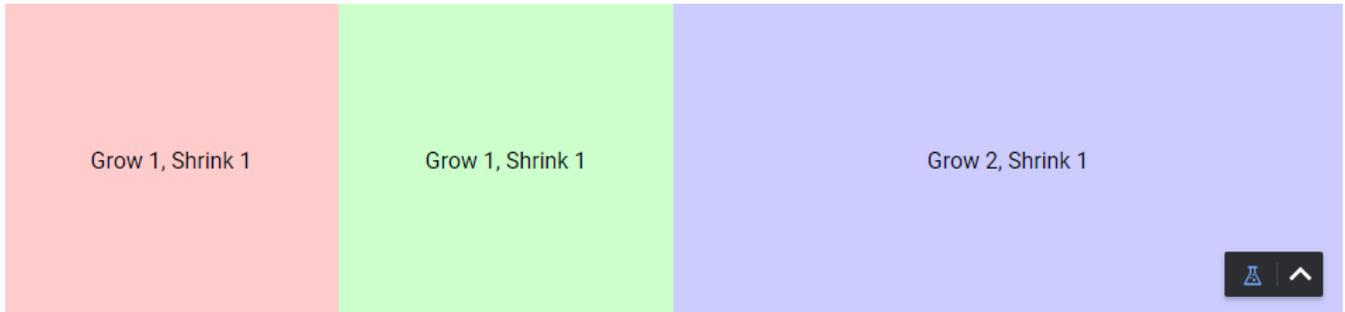


Even Scaling

Now let's say we start off with a blue component twice the **basis** of the others:

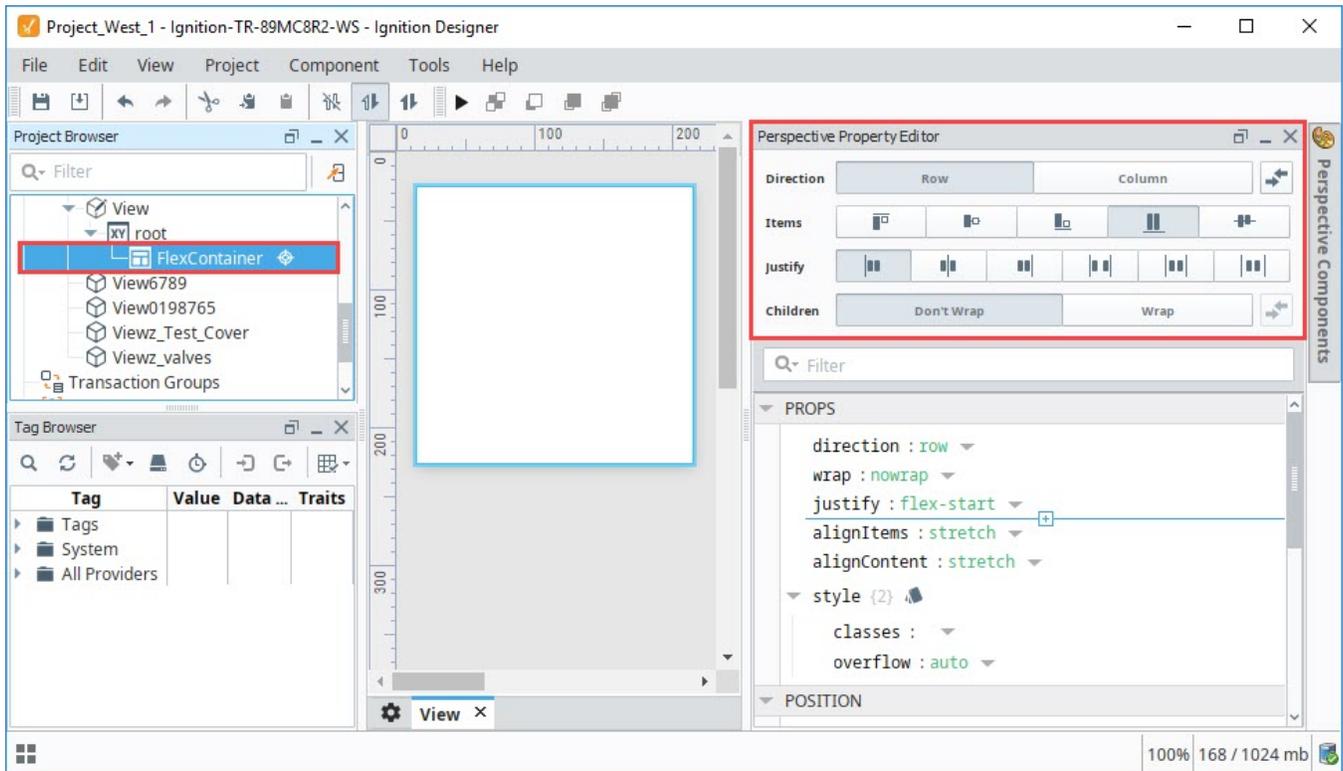


In order to maintain this ratio as the width of the container increases, it must have a **grow** value twice that of the others:



User Interface

When a Flex container is deep selected, there is a Graphical User Interface (GUI) at the top of the Perspective Property Editor that enables you to set the container's properties. Functionality is similar to that of the properties in the Props Tree, but you may find the visual interface easier or quicker to use.



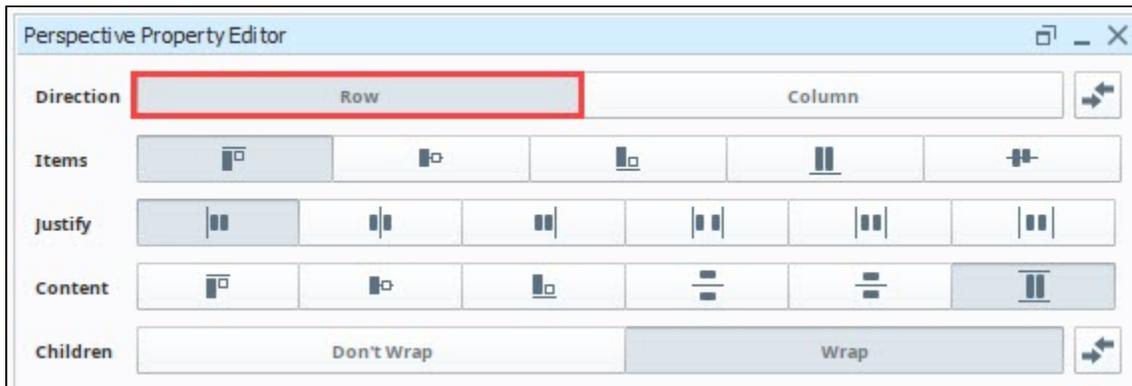
Direction

This sets the direction for the child layout. Options are **Row** or **Column**. When the **Reverse**  icon is selected, the contents of this container are displayed in reverse order.

Direction: Row

The following table shows the icons and properties they represent when **Direction: Row** is selected. The icon that's displayed if Reverse order is selected is also shown.

Note: Left/right/top/bottom notes in the descriptions refer to non-reversed directions. The phrase "when there is extra space" means when no components have are stretching to fill the space. That is, when no components have "grow" greater than 0.



Items			
Row Icon	Row Reversed Icon	Property	Description

		Flex Start	Child items are placed along the start (top) of the container when there is extra space.
		Center	Child items are placed along the center of the container when there is extra space.
		Flex End	Child items are placed along the end (bottom) of the container when there is extra space.
		Stretch	Child items are stretched from top to bottom of the container.
		Baseline	Child items are placed so the baseline of the text matches for all of them when there is extra space.

Justify

Row Icon	Row Reversed Icon	Property	Description
		Flex start	Adjusts placement of children to the start (left) of the container when there is extra space. If reversed, children are placed along the right.
		Center	Adjusts placement of children along the center of the container when there is extra space.
		Flex End	Adjusts placement of children along the end (right) of the container when there is extra space. If reversed, children are placed along the left.
		Space Between	Adjusts placement of children with space in between them reaching to the edges of the container when there is extra space.
		Space Around	Adjusts placement of children with even spacing in between them with some space along the edges when there is extra space.
		Space Evenly	Adjusts placement of children with even spacing in between them and the edges of the container when there is extra space.

Children

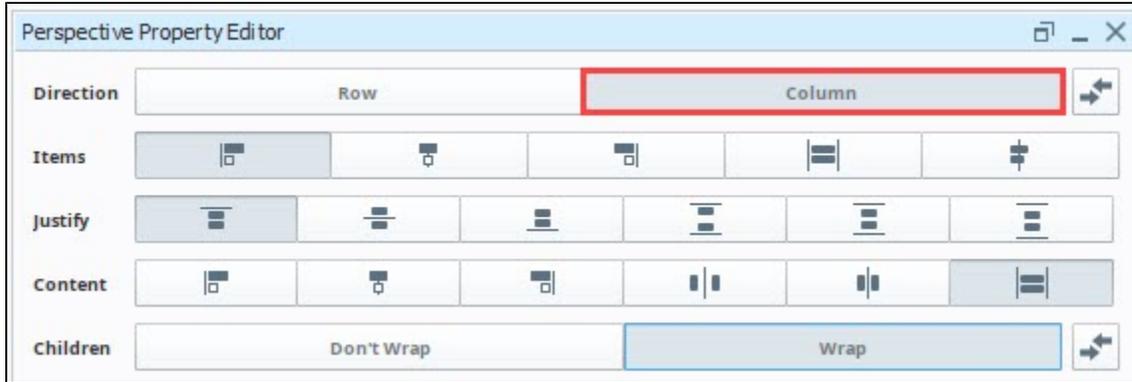
Icon	Property	Description
N/A	Don't Wrap	If there are more components than the width allows, shrink them.
N/A	Wrap	If there are more components than the width allows, wrap onto the next line.
	Reverse Wrap	Toggle to reverse the direction of wrap from top-down to bottom-up

Content (Only applicable when Children:Wrap is selected.)

Row Icon	Row Reversed Icon	Property	Description
		Flex start	Adjusts placement of wrapped content to the start (top) of the container when there is free space.
		Center	Adjusts placement of wrapped content to the middle of the container when there is free space.
		Flex End	Adjusts placement of wrapped content to the end (bottom) of the container when there is free space.
		Space Between	Adjusts placement of wrapped content evenly with space in between each wrapped line, reaching to the edges (top and bottom) of the container when there is extra space.
		Space Around	Adjusts placement of wrapped content evenly with space in between each wrapped line and the edges (top and bottom) of the container when there is extra space.
		Stretch	Adjusts placement of wrapped content evenly with space in between each wrapped line and after the last line (bottom) of the container when there is extra space.

Direction: Column

The following table shows the icons and properties they represent when **Direction: Column** is selected. The icon that's displayed if Reverse order is selected is also shown.



Items			
Column Icon	Column Reversed Icon	Property	Description
		Flex start	Child items are placed along the start (left) of the container when there is extra space.
		Center	Child items are placed along the center of the container when there is extra space.
		Flex End	Child items are placed along the end (right) of the container when there is extra space.
		Stretch	Child items are stretched from left to right of the container.
		Baseline	Child items are placed so the baseline of the text matches for all of them when there is extra space.
Justify			
Column Icon	Column Reversed Icon	Property	Description
		Flex start	Adjusts placement of children to the start (top) of the container when there is extra space. If reversed, children are placed along the bottom.
		Center	Adjusts placement of children along the center of the container when there is extra space.
		Flex End	Adjusts placement of children along the end (bottom) of the container when there is extra space. If reversed, children are placed along the top.
		Space Between	Adjusts placement of children with space in between them reaching to the edges of the container when there is extra space.
		Space Around	Adjusts placement of children with even spacing in between them with some space along the edges when there is extra space.
		Space Evenly	Adjusts placement of children with even spacing in between them and the edges of the container when there is extra space.
Children			
Icon	Description		
N/A	Don't Wrap	If there are more components than the width allows, shrink them.	
N/A	Wrap	If there are more components than the width allows, wrap onto the next line.	
	Reverse	Toggle to reverse the direction of wrap from top-down to bottom-up	

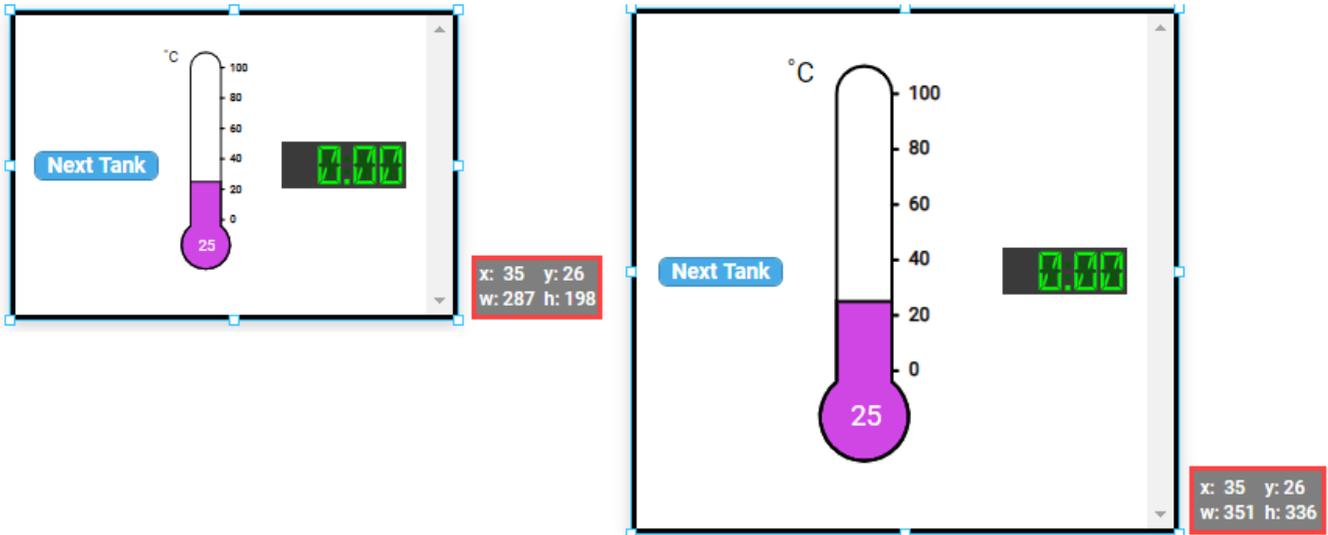
Content (Only applicable when Children:Wrap is selected.)			
Column Icon	Column Reversed Icon	Property	Description
		Flex start	Adjusts placement of wrapped content to the start (left) of the container when there is free space.
		Center	Adjusts placement of wrapped content to the middle of the container when there is free space.
		Flex End	Adjusts placement of wrapped content to the end (bottom) of the container when there is free space.
		Space Between	Adjusts placement of wrapped content evenly with space in between each wrapped line, reaching to the edges (left and right) of the container when there is extra space.
		Space Around	Adjusts placement of wrapped content evenly with space in between each wrapped line and the edges (left and right) of the container when there is extra space.
		Stretch	Adjusts placement of wrapped content evenly with space in between each wrapped line and after the last line of the container when there is extra space.

Scripting

See the [Perspective - Flex Container Scripting page](#) for the full list of scripting functions available for this component.

Example

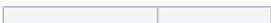
In the following example, we have three components inside a Flex container: Button, Thermometer, and LED Display. The position properties for the Button and LED Display are set so that their size will not change if the Flex container is resized. However, the Thermometer will shrink or grow depending on the Flex container size.



The following properties are set for the Flex Container:

Property	Value
props.direction	row
props.wrap	nowrap
props.justify	space-between
props.alignItems	center
props.alignContent	flex-start

The following properties are set for the Button component within the container:



Property	Value
props.text	Next Tank
position.grow	0
position.shrink	0
position.basis	80px

The following properties are set for the Thermometer component within the container:

Property	Value
props.direction	row
position.grow	1
position.shrink	1
position.basis	50%

The following properties are set for the LED Display component within the container:

Property	Value
position.grow	0
position.shrink	0
position.basis	80px

Perspective - Flex Container Scripting

This page details the various scripting, component, and extension functions available for [Perspective's Flex Container](#) component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

On this page ...

- [Component Events](#)
- [Component Functions](#)
 - [.getChildren\(\)](#)
- [Extension Functions](#)

Component Functions

.getChildren()

- Description

Returns an ArrayList, which contains references to all components inside of the container.

- Parameters

None

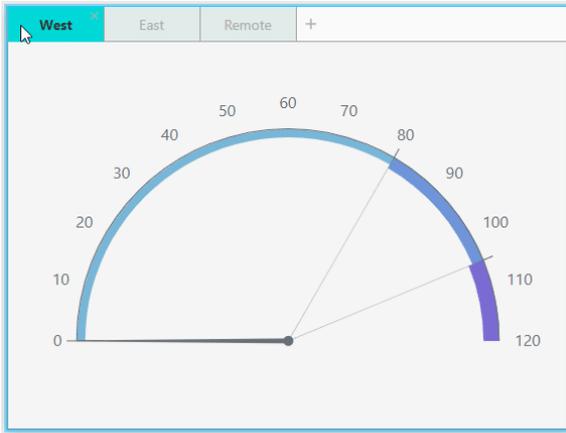
- Return

[Array List](#) - An ArrayList of components in the container. The resulting ArrayList can be iterated over via a for-loop.

Extension Functions

This component does not have extension functions associated with it.

Perspective - Tab Container



Component Palette Icon:



The Tab Container uses tabs as navigation buttons arranged together with the selected tab highlighted. Only one component can be displayed in each tab.

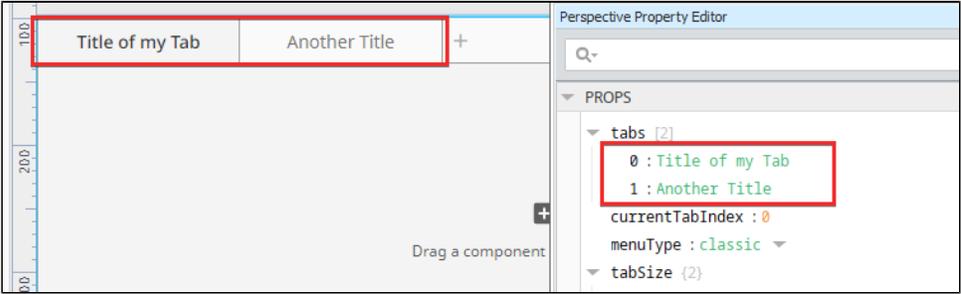
The following feature is new in Ignition version **8.1.2**
[Click here](#) to check out the other new features

The Tab Container component has two pre-configured **variants**:

- Classic - Layout is a traditional menu with boxed tabs.
- Modern - Layout has no borders around each tab and shows selection with an underline.

Properties

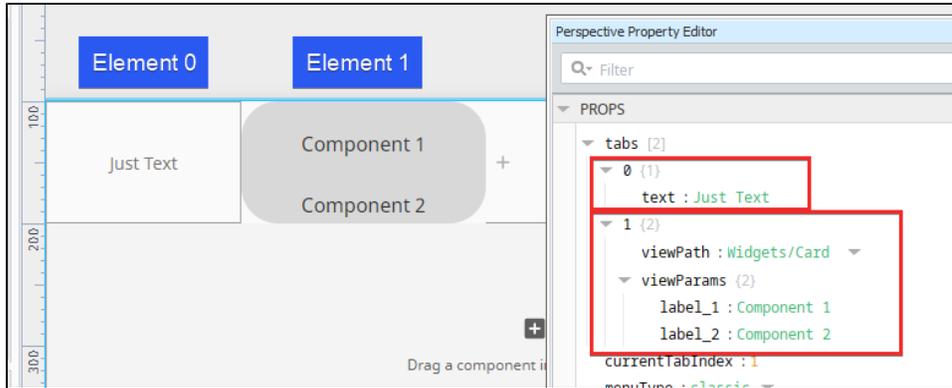
Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description
tabs	<p>Responsible for the number of tabs in the container. Adding additional elements to this array will result in an additional tab being rendered on</p> <p>In addition, each element is responsible for determining the content of the tab: either text or an embedded view. By default, the component <code>str</code> array, set to a value type. Providing a string to the element will set the text on the tab.</p> 

On this page ...

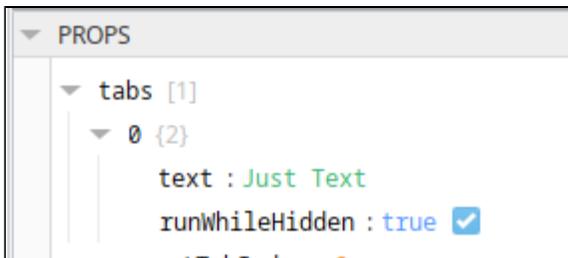
- [Properties](#)
- [User Interaction](#)
 - [Adding Components to Tabs](#)
 - [Adding and Deleting Tabs](#)
- [Scripting](#)
- [Examples](#)
 - [Example 1](#)
 - [Example 2](#)

If the element is changed to an object data type, then the text on the tab can be set by adding a **text** value member to the object and providing below. Alternatively, **viewPath** (value data type) and **viewParams** (object data type) can be added to the element, which allows you to render pass parameters to it. Element 1 below demonstrates the idea.



The following feature is new in Ignition version **8.1.5**
[Click here](#) to check out the other new features

A boolean **runWhileHidden** property can be added to an element that's set to an object data type. This setting determines if contents of this tab are activated, and if it should persist while in the background when the **currentTabIndex** changes.



The following feature is new in Ignition version **8.1.20**
[Click here](#) to check out the other new features

A boolean **disabled** property can be added to an element that's set to an object data type. This setting determines if a tab can be selected. If a tab becomes disabled, users will not be prevented from leaving the tab, but they will not be able to get back to it. The content of the tab is not affected.

currentTabIndex	Which index in tabs array is currently active.									
menuType	If the type is 'classic', a traditional menu with boxed tabs is shown. If the type is 'modern', it has no borders around each tab and shows selected									
tabSize	Default size allotted to a single tab. If a container width does not allow, tab width will shrink from this size accordingly.									
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>width</td> <td>Width in pixels for the tab size.</td> <td>value: numeric</td> </tr> <tr> <td>height</td> <td>Height in pixels for the tab size.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	width	Width in pixels for the tab size.	value: numeric	height	Height in pixels for the tab size.	value: numeric
Name	Description	Property Type								
width	Width in pixels for the tab size.	value: numeric								
height	Height in pixels for the tab size.	value: numeric								
menuStyle	Opens the Style menu to change Tab properties: Text, Background, Margin and Padding, Border, and Misc.									
contentStyle	Sets a style for the content frame component. Full menu of style options is available. You can also specify a style class .									
tabStyle	Additional styling to apply to all tabs depending active (selected) or inactive state.									
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> </tbody> </table>	Name	Description							
Name	Description									

active	Sets a style for a tab when it is the active tab. The Style menu contains all the tools for modifying text, background, margins, borders, and you can also specify a style class .
inactive	Sets a style for tabs that are inactive . The Style menu contains all the tools for modifying text, background, margins, borders, and you can also specify a style class .
style	Sets a style for this component. Full menu of style options is available. You can also specify a style class .

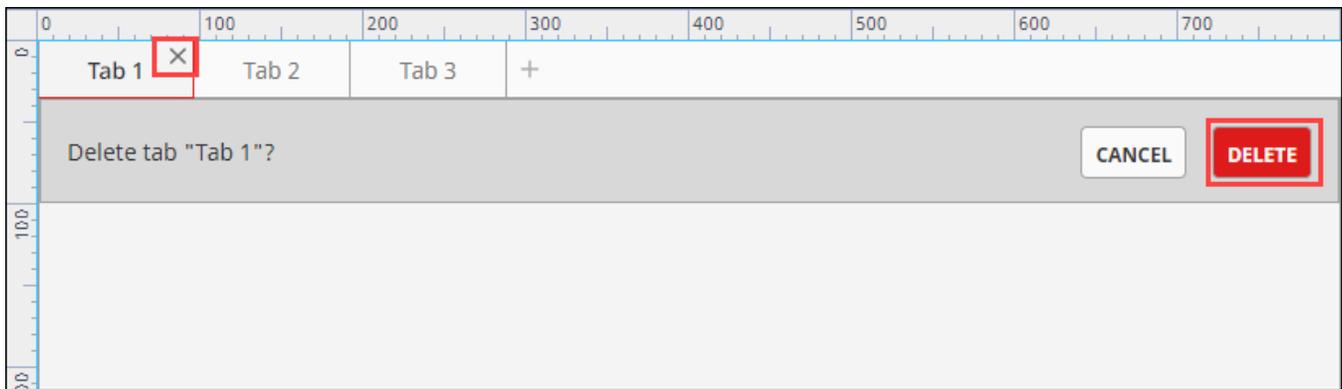
User Interaction

Adding Components to Tabs

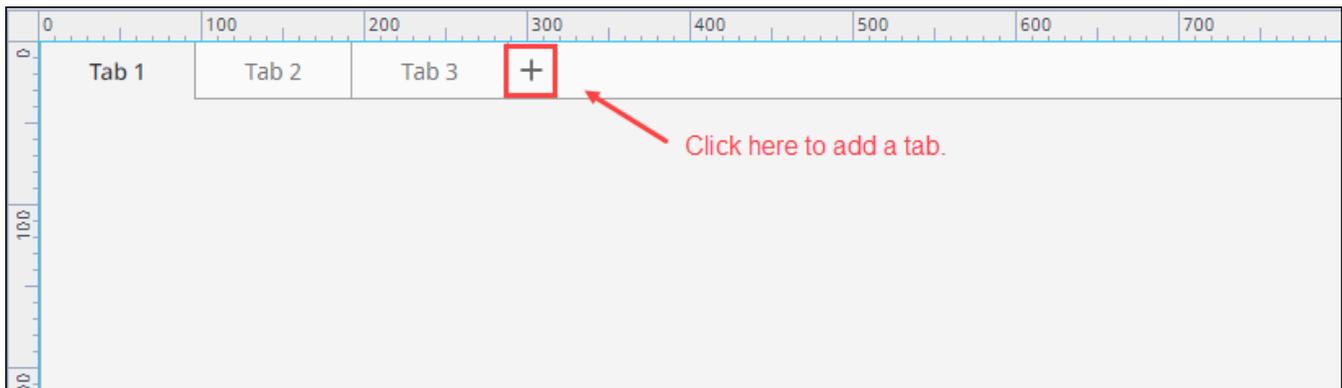
To add a component to a tab, deep select the Tab container, then drag a component onto it.

Adding and Deleting Tabs

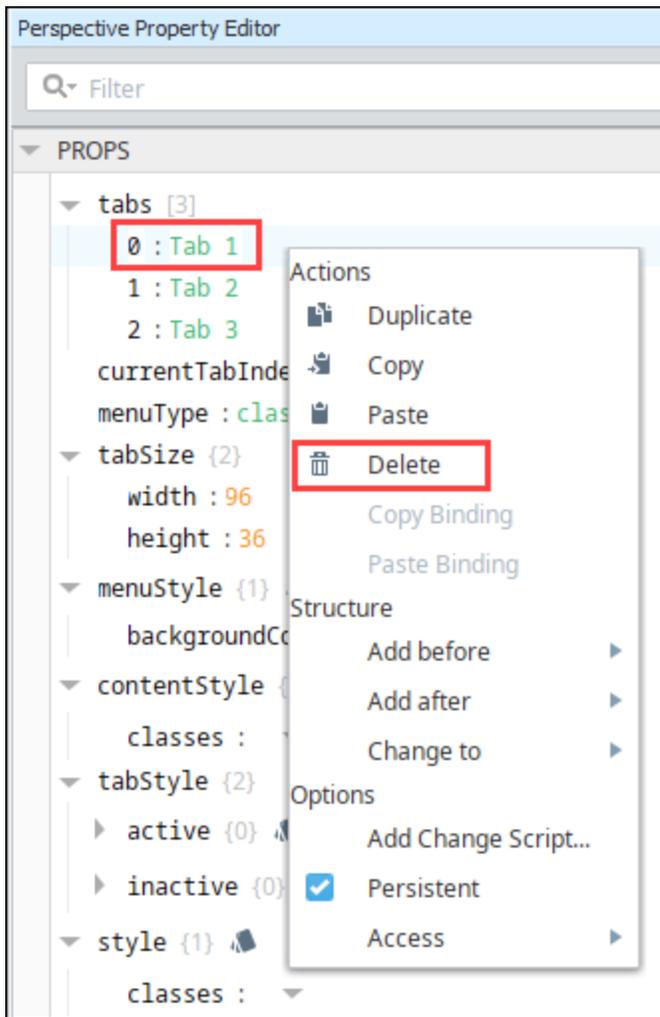
To delete a tab, click on the "X" to the right of the tab itself.



To add a tab, click on the Add icon to the right of the tabs:



You can also use the right-click menu in the Property Editor. Just right click on the tab you want to work with. You'll see options for copying, pasting,



Scripting

See the [Perspective - Tab Container Scripting](#) page for the full list of scripting functions available for this component.

Examples

Example 1

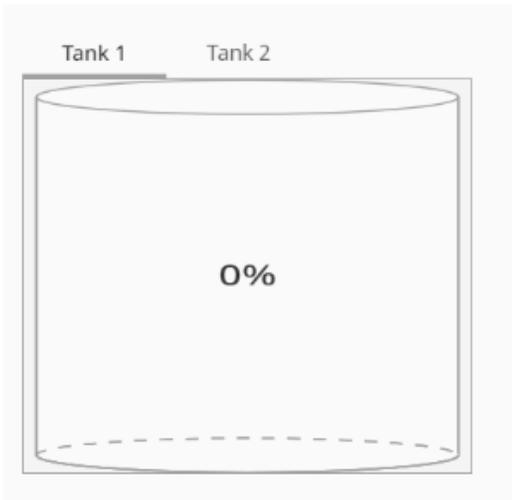


In this example, the default (Classic variant) Tab Container is used. Three tabs are set up in the Tab Container component. Tab 3 is active and contains a Map component.

Property	Value
props.tabs.0	Tab 1
props.tabs.1	Tab 2
props.tabs.2	Tab 3
props.menuStyle.backgroundColor	#D5D5D5
tabStyle.active.backgroundColor	#00FFFF
tabStyle.active.fontWeight	bold
tabStyle.inactive.backgroundColor	#CCFFFF
tabStyle.inactive.fontWeight	lighter

Example 2

In this example, the default (Modern variant) Tab Container is used. Two tabs are set up in the Tab Container component, each with a Cylindrical Tank component.



Property	Value
props.tabs.0	Tank 1
props.tabs.1	Tank 2
props.menuType	modern

Perspective - Tab Container Scripting

This page details the various scripting, component, and extension functions available for [Perspective's Tab Container](#) component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

On this page ...

- [Component Events](#)
- [Component Functions](#)
 - [.getChildren\(\)](#)
- [Extension Functions](#)

Component Functions

.getChildren()

- Description

Returns an ArrayList, which contains references to all components inside of the container.

- Parameters

None

- Return

[Array List](#) - An ArrayList of components in the container. The resulting ArrayList can be iterated over via a for-loop.

Extension Functions

This component does not have extension functions associated with it.

Perspective - Split Container



On this page ...

- [Properties](#)
- [Child Component Position Properties](#)
- [Scripting](#)

Component Palette Icon:



The following feature is new in Ignition version **8.1.18**
[Click here](#) to check out the other new features

The Split container holds two children separated by a draggable "split" that allows the user to resize the two children during the runtime. The Split container supports both horizontal and vertical layouts.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type	
orientation	The orientation in which the container's split is fixed. Horizontal will allow the user to adjust the container from left to right, while Vertical will allow the user to adjust it from top to bottom.	value: numeric	
split	Configuration for the split bar.		
	Name	Description	
	position	Bidirectionally represents the position of the split bar. Numeric values here will be used as pixels.	value: numeric or string
	size	The size of the split bar in pixels.	value: numeric
	visible	Determines if the split bar should be visible.	value: boolean
draggable	Determines if the split bar should be draggable.	value: boolean	
style	Use Style to customize the visual style of the component. The Style menu contains all the tools for modifying text, background, margins, and borders. You can also specify a style class .	object	

Child Component Position Properties

When a component is placed inside of a Split container, it will inherit the position property listed below.

Property	Description	Data
----------	-------------	------

		Type
position	Indicates which side of the split bar the child component is located on. Expected values include "left", "right", "top", and "bottom"	value: string

Scripting

See the [Perspective - Split Container Scripting page](#) for the full list of scripting functions available for this component.

Perspective - Split Container Scripting

This page details the various scripting, component, and extension functions available for [Perspective's Split Container](#) component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

On this page ...

- [Component Events](#)
 - [onMinBoundReached](#)
 - [onMaxBoundReached](#)
- [Component Functions](#)
 - [.getChildren\(\)](#)
- [Extension Functions](#)

onMinBoundReached

This event is fired when the split reaches the minimum bound on the container. For example, when the split bar's position reaches 0 px or 0% of the container's width.

Note: This component event is designed to be used in tandem with a script runAction. Within the script action, special properties and methods are available on the **event** object, which is passed to the script action as a parameter.

onMaxBoundReached

This event is fired when the split reaches the maximum bound on the container. For example, when the split bar's position reaches 100% of the container's width.

Note: This component event is designed to be used in tandem with a script runAction. Within the script action, special properties and methods are available on the **event** object, which is passed to the script action as a parameter.

Component Functions

.getChildren()

- Description

Returns an ArrayList, which contains references to all components inside of the container.

- Parameters

None

- Return

[Array List](#) - An ArrayList of components in the container. The resulting ArrayList can be iterated over via a for-loop.

Extension Functions

This component does not have extension functions associated with it.

Perspective - Display Palette

Display Components

Perspective offers a variety of components that display static and dynamic information.

Here is a complete list of Display components, and a link pointing to a page containing the component's description, properties, and an example of how to configure it.



**Display
Components**

[Watch the Video](#)

In This Section ...

Perspective - Alarm Journal Table

Event Time	Event Id	Source	Event State	Priority
01/13/2020 13:47:17	66464cb1-50d3-43f8-8996-f007...	prov:default:/tag-Motor...	Active	High
01/13/2020 13:47:25	37c85076-b6c8-4b79-8536-e6fff...	prov:default:/tag-Motor...	Active	High
01/13/2020 13:47:35	0beb640-06c7-4f97-bc39-1d92...	prov:default:/tag-Motor...	Active	High
01/13/2020 13:47:45	a213a658-8426-4e46-87a7-157...	prov:default:/tag-Motor...	Active	High
01/13/2020 13:47:56	264550ea-2ff6-4643-b57f-1e367...	prov:default:/tag-Motor...	Active	High
01/13/2020 13:47:17	a2546d50-d94c-4321-b75b-4bc5...	prov:default:/tag-Motor...	Ack	High
01/13/2020 13:47:25	eb6e4e3f-59c3-49a6-bd52-427b9...	prov:default:/tag-Motor...	Ack	High

On this page ...

- [User Interface](#)
- [Properties](#)
- [Scripting](#)

Component Palette Icon:



The Perspective Alarm Journal Table displays the history of the alarm system. It can be configured to show active, cleared, and acknowledged events.

Before the Alarm Journal Table can retrieve alarm data, an [Alarm Journal](#) must first be configured.

The Perspective Alarm Journal Table has a number of configuration options that can be used to filter on realtime and historical alarm data, and change how the component displays those alarms. When you first drag the Alarm Journal Table into the Designer, by default, the table will show you the last 8 hours of alarm journal data. You can interface with the journal table in the Designer, in Preview Mode of the Designer, and in a Perspective Session.

The Alarm Journal Table provides a host of filtering properties that allow you to filter on various parts of alarms and view the details. All the alarm states are visible by clicking the Filter button on the table. There is also a search bar where you can enter text to further refine your filter criteria so you have less alarm events to scroll through. The Alarm Journal Table can filter for alarm events in either Realtime or Historical using the Date Range feature.

You can change the columns that are displayed, the order of the columns, and/or the column width in Preview Mode and in a Perspective Session. Right-click on the table header to show/hide columns. Click and drag the margins of the columns to resize their width. You can also sort table columns in ascending or descending order by simply clicking the up or down arrows next to each column header.

The following feature is new in Ignition version **8.1.12**
[Click here](#) to check out the other new features

The Alarm Journal Table utilizes a shared polling engine when in realtime date range mode to cache and share polling tasks across concurrent sessions. The cache persists for a period of time that matches the configured poll rate. This optimization allows multiple components to poll the alarm system with a reduced impact on overall performance.

User Interface

The following table describes the user interface available for the Alarm Journal Table.

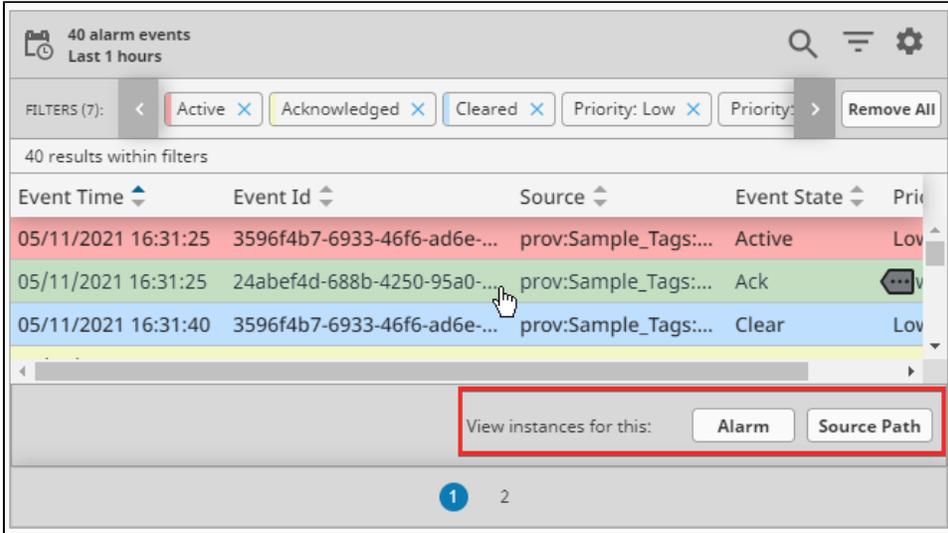
Interaction	Description
Config Settings	Clicking the  icon or right clicking on a column heading will open the Configuration menu, which determines which columns show appear on the component.
Date Range	Clicking on the  icon will open the Date Range window, allowing the user to determine a range of time to filter on. Alarm events with an Event Time that matches the selected range will appear on the table.
Filters	Clicking the  icon opens the Filters menu, allowing you to select event states and priorities to filter on.
Pages	A listing of pages. Long lists of alarm events are spread across multiple pages to improve performance. Clicking on a number will switch which page is shown.
Popup Modal	Hovering over a row in the table will cause the Popup Modal icon to appear. Clicking this icon shows more information about the alarm.
Rows to Display	Determines how many rows are shown per page.

Search

Clicking the  icon will cause a search bar to appear, allowing the user to type in search terms.

Additional interface options become available when selecting a row within the table, which allow the table to search for similar events.

- Alarm: Shows all events for a single alarm instance, meaning the active event, the clearing event, and the acknowledging event (if present).
- Source Path: Shows all events that match the same source path as the selected event, respecting the selected date range.



Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Prop Type
name	The name of the alarm journal to query. Default is "Journal". Note: On Edge Gateways, set this to an empty string value to have the component use the Edge Alarm Journal. Note: In Gateway Network configurations, set this to an empty string OR use the name of the Remote Alarm Journal.	value: string
refresh Rate	The following feature is new in Ignition version 8.1.0 Click here to check out the other new features The rate at which the table will poll for updates in milliseconds.	value: numeri
enable Header	Enables the table header. Default is true.	value: boolea
enable Details	Enables the details action. Default is true.	value: boolea
toolbar	Settings for the toolbar.	object

Name	Description	Property Type
------	-------------	---------------

enabled	Enables the visibility of the table toolbar. Default is true.	value: boolean
enableDateRange	Enables the visibility of the date range toggle. Default is true.	value: boolean
enableFilter	Enables the visibility of the text filter toggle. Default is true.	value: boolean
toggleableFilter	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.18 Click here to check out the other new features</p> </div> <p>If false, the text filter will not require user interaction to open, and instead will remain open. Default is true.</p>	value: boolean
enableFilterResults	Enables the visibility of the filters results count message. Default is true.	value: boolean
enablePreFilters	Enables the visibility of the prefilter toggle. Default is true.	value: boolean
enableConfiguration	Enables the visibility of the configuration toggle. Default is true.	value: boolean

dateFormat A date format string to be applied against dates. Options are none, date in the format "10/15/2018", time in format "3:59:00 PM", or date time in format "10/15/2018 15:59:00". value: string

responsive Responsive layout configuration. Rows are converted to cards. While in responsive layout, disables or removes certain table features that are no longer applicable. object

Name	Description	Property Type
enabled	Enables responsive layout. Default is false.	value: boolean
breakpoint	Width in pixels that triggers change in responsive layout.	value: numeric

dateRange Settings for date range state. object

Name	Description	Property Type									
mode	The current date range mode: realtime or historical.	value: string									
realtime	When mode is set to realtime , this property will appear with the following sub properties available: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>interval</td> <td>The realtime interval as an integer</td> <td>value: numeric</td> </tr> <tr> <td>unit</td> <td>The realtime interval unit: hours, days, months and years.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	interval	The realtime interval as an integer	value: numeric	unit	The realtime interval unit: hours, days, months and years.	value: string	object
Name	Description	Property Type									
interval	The realtime interval as an integer	value: numeric									
unit	The realtime interval unit: hours, days, months and years.	value: string									
historical	When mode is set to historical , this property will appear with the following sub properties available: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>startDate</td> <td>The start date to use, in milliseconds.</td> <td>value: numeric</td> </tr> <tr> <td>endDate</td> <td>The end date to use, in milliseconds.</td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	startDate	The start date to use, in milliseconds.	value: numeric	endDate	The end date to use, in milliseconds.		
Name	Description	Property Type									
startDate	The start date to use, in milliseconds.	value: numeric									
endDate	The end date to use, in milliseconds.										

filter Filter settings. object

Name	Description	Property Type

text	The alarm events filter text.	value: string																					
events	Alarm event types.	object																					
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>active</td> <td>Whether to display alarm events with active states. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>acked</td> <td>Whether to display alarm events with acked states. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>cleared</td> <td>Whether to display alarm events with cleared states. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>enabled</td> <td>Whether to display alarm events that were generated by enabling an alarm.</td> <td>value: boolean</td> </tr> <tr> <td>disabled</td> <td>Whether to display alarm events that were generated by disabling an alarm.</td> <td>value: boolean</td> </tr> <tr> <td>system</td> <td>Whether to display alarm event with system states. Default is false.</td> <td>value: boolean</td> </tr> </tbody> </table>	Name	Description	Property Type	active	Whether to display alarm events with active states. Default is true.	value: boolean	acked	Whether to display alarm events with acked states. Default is true.	value: boolean	cleared	Whether to display alarm events with cleared states. Default is true.	value: boolean	enabled	Whether to display alarm events that were generated by enabling an alarm.	value: boolean	disabled	Whether to display alarm events that were generated by disabling an alarm.	value: boolean	system	Whether to display alarm event with system states. Default is false.	value: boolean	
Name	Description	Property Type																					
active	Whether to display alarm events with active states. Default is true.	value: boolean																					
acked	Whether to display alarm events with acked states. Default is true.	value: boolean																					
cleared	Whether to display alarm events with cleared states. Default is true.	value: boolean																					
enabled	Whether to display alarm events that were generated by enabling an alarm.	value: boolean																					
disabled	Whether to display alarm events that were generated by disabling an alarm.	value: boolean																					
system	Whether to display alarm event with system states. Default is false.	value: boolean																					
priorities	Alarm event priority pre-filters.	object																					
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>diagnostic</td> <td>Whether to display alarms with diagnostic priority. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>low</td> <td>Whether to display alarms with low priority. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>medium</td> <td>Whether to display alarms with medium priority. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>high</td> <td>Whether to display alarms with high priority. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>critical</td> <td>Whether to display alarms with critical priority. Default is true.</td> <td>value: boolean</td> </tr> </tbody> </table>	Name	Description	Property Type	diagnostic	Whether to display alarms with diagnostic priority. Default is false.	value: boolean	low	Whether to display alarms with low priority. Default is true.	value: boolean	medium	Whether to display alarms with medium priority. Default is true.	value: boolean	high	Whether to display alarms with high priority. Default is true.	value: boolean	critical	Whether to display alarms with critical priority. Default is true.	value: boolean				
Name	Description	Property Type																					
diagnostic	Whether to display alarms with diagnostic priority. Default is false.	value: boolean																					
low	Whether to display alarms with low priority. Default is true.	value: boolean																					
medium	Whether to display alarms with medium priority. Default is true.	value: boolean																					
high	Whether to display alarms with high priority. Default is true.	value: boolean																					
critical	Whether to display alarms with critical priority. Default is true.	value: boolean																					
conditions	Gateway side alarm query conditions.	object																					
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>source</td> <td>Filter alarms by alarm source path. Specify multiple paths by separating them with commas. Supports the wildcard *.</td> <td>value: string</td> </tr> <tr> <td>display Path</td> <td>Filter alarms by display path, falling back to the source path if a custom display path is not set. Specify multiple paths by separating them with commas. Supports the wildcard *.</td> <td>value: string</td> </tr> <tr> <td>provider</td> <td>Filter alarms by provider.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	source	Filter alarms by alarm source path. Specify multiple paths by separating them with commas. Supports the wildcard *.	value: string	display Path	Filter alarms by display path, falling back to the source path if a custom display path is not set. Specify multiple paths by separating them with commas. Supports the wildcard *.	value: string	provider	Filter alarms by provider.	value: string										
Name	Description	Property Type																					
source	Filter alarms by alarm source path. Specify multiple paths by separating them with commas. Supports the wildcard *.	value: string																					
display Path	Filter alarms by display path, falling back to the source path if a custom display path is not set. Specify multiple paths by separating them with commas. Supports the wildcard *.	value: string																					
provider	Filter alarms by provider.	value: string																					
results	Alarm event filtering results configuration and data.	object																					
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enable filtering results to be written back to props. Caution: Enabling this property may cause a performance decline.</td> <td>value: boolean</td> </tr> <tr> <td>data</td> <td>An array of objects representing the current filtered data, if enabled and active.</td> <td>array</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Enable filtering results to be written back to props. Caution: Enabling this property may cause a performance decline.	value: boolean	data	An array of objects representing the current filtered data, if enabled and active.	array													
Name	Description	Property Type																					
enabled	Enable filtering results to be written back to props. Caution: Enabling this property may cause a performance decline.	value: boolean																					
data	An array of objects representing the current filtered data, if enabled and active.	array																					

rowStyles

Styles to apply to rows given their alarm event and designated priority.	object						
<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>active</td> <td>Style settings for rows with active alarms.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	active	Style settings for rows with active alarms.	object	
Name	Description	Property Type					
active	Style settings for rows with active alarms.	object					

Name	Description	Property Type																		
base	Base style settings for active alarms. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																		
priorities	Style settings for the alarm row based on priority.	object																		
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>diagnostic</td> <td>Style for active alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class</td> <td>object</td> </tr> <tr> <td>low</td> <td>Style for active alarms with low priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>medium</td> <td>Style for active alarms with medium priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>high</td> <td>Style for active alarms with high priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>critical</td> <td>Style for active alarms with critical priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	diagnostic	Style for active alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class	object	low	Style for active alarms with low priority. Full menu of style options is available. You can also specify a style class .	object	medium	Style for active alarms with medium priority. Full menu of style options is available. You can also specify a style class .	object	high	Style for active alarms with high priority. Full menu of style options is available. You can also specify a style class .	object	critical	Style for active alarms with critical priority. Full menu of style options is available. You can also specify a style class .	object	
Name	Description	Property Type																		
diagnostic	Style for active alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class	object																		
low	Style for active alarms with low priority. Full menu of style options is available. You can also specify a style class .	object																		
medium	Style for active alarms with medium priority. Full menu of style options is available. You can also specify a style class .	object																		
high	Style for active alarms with high priority. Full menu of style options is available. You can also specify a style class .	object																		
critical	Style for active alarms with critical priority. Full menu of style options is available. You can also specify a style class .	object																		

acked Style settings for rows with acked alarms. object

Name	Description	Property Type																		
base	Base style settings for acked alarms.	object																		
priorities	Style settings for the alarm row based on priority.	object																		
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>diagnostic</td> <td>Style for acked alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class .</td> <td>object</td> </tr> <tr> <td>low</td> <td>Style for acked alarms with low priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>medium</td> <td>Style for acked alarms with medium priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>high</td> <td>Style for acked alarms with high priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>critical</td> <td>Style for acked alarms with critical priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	diagnostic	Style for acked alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class .	object	low	Style for acked alarms with low priority. Full menu of style options is available. You can also specify a style class .	object	medium	Style for acked alarms with medium priority. Full menu of style options is available. You can also specify a style class .	object	high	Style for acked alarms with high priority. Full menu of style options is available. You can also specify a style class .	object	critical	Style for acked alarms with critical priority. Full menu of style options is available. You can also specify a style class .	object	
Name	Description	Property Type																		
diagnostic	Style for acked alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class .	object																		
low	Style for acked alarms with low priority. Full menu of style options is available. You can also specify a style class .	object																		
medium	Style for acked alarms with medium priority. Full menu of style options is available. You can also specify a style class .	object																		
high	Style for acked alarms with high priority. Full menu of style options is available. You can also specify a style class .	object																		
critical	Style for acked alarms with critical priority. Full menu of style options is available. You can also specify a style class .	object																		

cleared Style settings for rows with cleared alarms. object

Name	Description	Property Type
base	Base style settings for cleared alarms. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object

priorities Style settings for the alarm row based on priority. object

Name	Description	Property Type
diagnostic	Style for cleared alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class .	object
low	Style for cleared alarms with low priority. Full menu of style options is available. You can also specify a style class .	object
medium	Style for cleared alarms with medium priority. Full menu of style options is available. You can also specify a style class .	object
high	Style for cleared alarms with high priority. Full menu of style options is available. You can also specify a style class .	object
critical	Style for cleared alarms with critical priority. Full menu of style options is available. You can also specify a style class .	object

system Style settings for rows with system alarms. object

Name	Description	Property Type																		
base	Base style settings for system alarms. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																		
priorities	Style settings for the alarm row based on priority. <table border="1" data-bbox="462 1045 1131 1579"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>diagnostic</td> <td>Style for system alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>low</td> <td>Style for system alarms with low priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>medium</td> <td>Style for system alarms with medium priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>high</td> <td>Style for system alarms with high priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>critical</td> <td>Style for system alarms with critical priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	diagnostic	Style for system alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class .	object	low	Style for system alarms with low priority. Full menu of style options is available. You can also specify a style class .	object	medium	Style for system alarms with medium priority. Full menu of style options is available. You can also specify a style class .	object	high	Style for system alarms with high priority. Full menu of style options is available. You can also specify a style class .	object	critical	Style for system alarms with critical priority. Full menu of style options is available. You can also specify a style class .	object	object
Name	Description	Property Type																		
diagnostic	Style for system alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class .	object																		
low	Style for system alarms with low priority. Full menu of style options is available. You can also specify a style class .	object																		
medium	Style for system alarms with medium priority. Full menu of style options is available. You can also specify a style class .	object																		
high	Style for system alarms with high priority. Full menu of style options is available. You can also specify a style class .	object																		
critical	Style for system alarms with critical priority. Full menu of style options is available. You can also specify a style class .	object																		

dragOrderable

The following feature is new in Ignition version **8.1.14**
[Click here](#) to check out the other new features

When enabled, users may drag column headers to reorder columns in the table.

value: boolean

columns Used for determining what column properties to display on the Alarm Journal Table.

object

Name	Description	Property Type															
eventTime	Settings for the eventTime column.	object															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether the column is enabled. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>width</td> <td>The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.</td> <td>value: numeric</td> </tr> <tr> <td>strictWidth</td> <td>If enabled, the width of the column (set with the width property) becomes static. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>sort</td> <td>Default sort order of the column. Options are none, ascending, or descending.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Whether the column is enabled. Default is true.	value: boolean	width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric	strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean	sort	Default sort order of the column. Options are none, ascending, or descending.	value: string	
Name	Description	Property Type															
enabled	Whether the column is enabled. Default is true.	value: boolean															
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric															
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean															
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string															
eventId	Settings for the eventId column.	object															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether the column is enabled. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>width</td> <td>The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.</td> <td>value: numeric</td> </tr> <tr> <td>strictWidth</td> <td>If enabled, the width of the column (set with the width property) becomes static. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>sort</td> <td>Default sort order of the column. Options are none, ascending, or descending.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Whether the column is enabled. Default is true.	value: boolean	width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric	strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean	sort	Default sort order of the column. Options are none, ascending, or descending.	value: string	
Name	Description	Property Type															
enabled	Whether the column is enabled. Default is true.	value: boolean															
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric															
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean															
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string															
name	Settings for the name column.	object															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether the column is enabled. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>width</td> <td>The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.</td> <td>value: numeric</td> </tr> <tr> <td>strictWidth</td> <td>If enabled, the width of the column (set with the width property) becomes static. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>sort</td> <td>Default sort order of the column. Options are none, ascending, or descending.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Whether the column is enabled. Default is true.	value: boolean	width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric	strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean	sort	Default sort order of the column. Options are none, ascending, or descending.	value: string	
Name	Description	Property Type															
enabled	Whether the column is enabled. Default is true.	value: boolean															
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric															
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean															
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string															
source	Settings for the source column.	object															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether the column is enabled. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>width</td> <td>The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.</td> <td>value: numeric</td> </tr> <tr> <td>strictWidth</td> <td>If enabled, the width of the column (set with the width property) becomes static. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>sort</td> <td>Default sort order of the column. Options are none, ascending, or descending.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Whether the column is enabled. Default is true.	value: boolean	width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric	strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean	sort	Default sort order of the column. Options are none, ascending, or descending.	value: string	
Name	Description	Property Type															
enabled	Whether the column is enabled. Default is true.	value: boolean															
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric															
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean															
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string															
eventState	Settings for the eventState column.	object															

ate

Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

priority

Settings for the priority column.

object

Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

state

Settings for the state column.

object

Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

display Path

Settings for the displayPath column.

object

Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

label

Settings for the label column.

object

Name	Description	Property Type
------	-------------	---------------

enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

eventValue Settings for the eventValue column. object

Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

isSystemEvent Settings for the isSystemEvent column. object

Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

ackUser Settings for the ackUser column. object

Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

ackNotes Settings for the ackNotes column. object

Name	Description	Property Type
------	-------------	---------------

	<table border="1"> <tr> <td>enabled</td> <td>Whether the column is enabled. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>width</td> <td>The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.</td> <td>value: numeric</td> </tr> <tr> <td>strictWidth</td> <td>If enabled, the width of the column (set with the width property) becomes static. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>sort</td> <td>Default sort order of the column. Options are none, ascending, or descending.</td> <td>value: string</td> </tr> </table>	enabled	Whether the column is enabled. Default is true.	value: boolean	width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric	strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean	sort	Default sort order of the column. Options are none, ascending, or descending.	value: string										
enabled	Whether the column is enabled. Default is true.	value: boolean																					
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric																					
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean																					
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string																					
columnsAssociated	<p>The following feature is new in Ignition version 8.1.14 Click here to check out the other new features</p> <p>A list of columns used to retrieve and display alarm associated data.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>field</td> <td>Maps to the associated data value represented by the column.</td> <td>value: string</td> </tr> <tr> <td>enabled</td> <td>Whether the column is enabled. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>width</td> <td>The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.</td> <td>value: numeric</td> </tr> <tr> <td>strictWidth</td> <td>If enabled, the width of the column (set with the width property) becomes static. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>sort</td> <td>Default sort order of the column. Options are none, ascending, or descending.</td> <td>value: string</td> </tr> <tr> <td>order</td> <td>Order to display this column in the table.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	field	Maps to the associated data value represented by the column.	value: string	enabled	Whether the column is enabled. Default is true.	value: boolean	width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric	strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean	sort	Default sort order of the column. Options are none, ascending, or descending.	value: string	order	Order to display this column in the table.	value: numeric	object
Name	Description	Property Type																					
field	Maps to the associated data value represented by the column.	value: string																					
enabled	Whether the column is enabled. Default is true.	value: boolean																					
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric																					
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean																					
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string																					
order	Order to display this column in the table.	value: numeric																					
sortOrder	The default weighted order in which columns and their contents are sorted relative to other columns and their contents. Only works if used when the component loads. Columns need to have sort configured in order for this to work.	array																					
pager	Settings for the pager. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enables the pager to be displayed. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>hide</td> <td>Visually hides the pager from view. Useful when pager is manipulated in a controlled fashion via the activePage property. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>options</td> <td>Rows to show per pager option.</td> <td>array</td> </tr> <tr> <td>initialOption</td> <td>The initial option to use when the table first loads. It must exist as an available option.</td> <td>value: numeric</td> </tr> <tr> <td>activePage</td> <td>Represents the current active page and corresponds to the value of the page jump input field.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Enables the pager to be displayed. Default is true.	value: boolean	hide	Visually hides the pager from view. Useful when pager is manipulated in a controlled fashion via the activePage property. Default is false.	value: boolean	options	Rows to show per pager option.	array	initialOption	The initial option to use when the table first loads. It must exist as an available option.	value: numeric	activePage	Represents the current active page and corresponds to the value of the page jump input field.	value: numeric	object			
Name	Description	Property Type																					
enabled	Enables the pager to be displayed. Default is true.	value: boolean																					
hide	Visually hides the pager from view. Useful when pager is manipulated in a controlled fashion via the activePage property. Default is false.	value: boolean																					
options	Rows to show per pager option.	array																					
initialOption	The initial option to use when the table first loads. It must exist as an available option.	value: numeric																					
activePage	Represents the current active page and corresponds to the value of the page jump input field.	value: numeric																					
style	Sets a style that applies to the component. Full menu of style options is available. You can also specify a style class .	object																					

Scripting

See the [Perspective - Alarm Journal Table Scripting page](#) for the full list of scripting functions available for this component.

Perspective - Alarm Journal Table Scripting

This page details the various scripting, component, and extension functions available for [Perspective's Alarm Journal Table](#) component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

On this page ...

- [Component Events](#)
- [Component Functions](#)
 - [.refreshData\(\)](#)
- [Extension Functions](#)
 - [filterAlarm](#)

Component Functions

.refreshData()

The following feature is new in Ignition version **8.1.18**
[Click here](#) to check out the other new features

- Description
Refreshes the data on the Alarm Journal Table.
- Parameters
None
- Return
Nothing

Extension Functions

filterAlarm

The following feature is new in Ignition version **8.1.0**
[Click here](#) to check out the other new features

- Description
Called for each event before it is displayed in the table, allowing you to hide or show each alarm event (row) in the table. Provides an opportunity to write a more complex filter than what's normally provided to the component. Return False to exclude an alarm event from the table.
- Parameters
[ComponentModelScriptWrapper.SafetyWrapper](#) self- A reference to the component that is invoking this function.
[PyAlarmEvent](#) alarmEvent - The alarm event itself. Call `alarmEvent.get('propertyName')` to inspect properties on the event. Common properties: 'name', 'source', 'priority'.
- Return
[Boolean](#) - The function must return either a True or False for every alarm event in the table. True will show the alarm. False will hide the alarm.

Examples

With the built-in alarmEvent object all [alarm event properties](#) are accessible to this function, and can be used to help determine if any given event should appear on the table. Furthermore, Associated Data (also known as custom alarm properties) can be examined from the same event.

```
# Replace Property Name below with a the name of the property you wish to filter on.  
if alarmEvent.get('Property Name'):
```

```
    return True
# It's important that you return a False value for the events you don't want to see in the table.
else:
    return False
```

You could also condense the code example above by using something like the following:

```
return "Low"== str(alarmEvent.get('priority'))
```

Script Configuration on AlarmJournalTable

Scripts

- Custom Methods
 - + Add method...
- Message Handlers
 - + Add handler...
- Extension Functions
 - filterAlarm

Edit Extension Function

enabled

Script

```
1 def filterAlarm(self, alarmEvent):
    """
    Called for each alarm event before it is displayed in the table. Return
    False to exclude the alarm from the table.

    Arguments:
        self: A reference to the component that is invoking this function.
        alarmEvent: The alarm event itself. Call alarmEvent.get('propertyName')
                   to inspect. Common properties: 'name', 'source', 'priority'
    """
2   return "Low"== str(alarmEvent.get('priority'))
```

OK Cancel Apply

Perspective - Alarm Status Table

Active Time	Display Path	Priority	State	Name
02/04/2020 18:34:07	Writeable/WriteableInteger1/Low Tank Level	Critical	Active, Unacknowledged	Low Tank Level
02/06/2020 09:04:07	Speed/High Speed	Critical	Active, Unacknowledged	High Speed
02/04/2020 18:34:07	Tank Level 2/Low SP2	Critical	Active, Acknowledged	Low SP2
02/11/2020 15:17:10	Sine/Sine2/Low Level	Critical	Cleared, Unacknowledged	Low Level
02/11/2020 15:33:49	Sine/Sine2/Low Level	Critical	Cleared, Unacknowledged	Low Level

Component Palette Icon:



On this page ...

- [User Interface](#)
- [Properties](#)
- [Scripting](#)
- [Examples](#)
 - [Example 1](#)
 - [Example 2 - Alarm Status Table Row Styles](#)

The Alarm Status Table allows you to view currently active alarm events in the system, providing an easy way to inspect the alarm details, shelf alarms, and acknowledge them.

Acknowledgement is handled by selecting (checking) alarms and pressing the "Acknowledge" button.

Shelving is supported by pressing the "Shelve" button when an alarm is selected and choosing a time duration.

You can change the columns that are displayed and the column width in Preview Mode and in a Perspective Session. Right-click on the table header to show/hide columns. Click and drag the margins of the columns to resize their width. You can also sort table columns in ascending or descending order by simply clicking the up or down arrows next to each column header. Sorting on alarm State and Priority in the Alarm Status Table sorts in descending order. All the other columns the sort order is alphanumerical.

The following feature is new in Ignition version **8.1.14**
[Click here](#) to check out the other new features

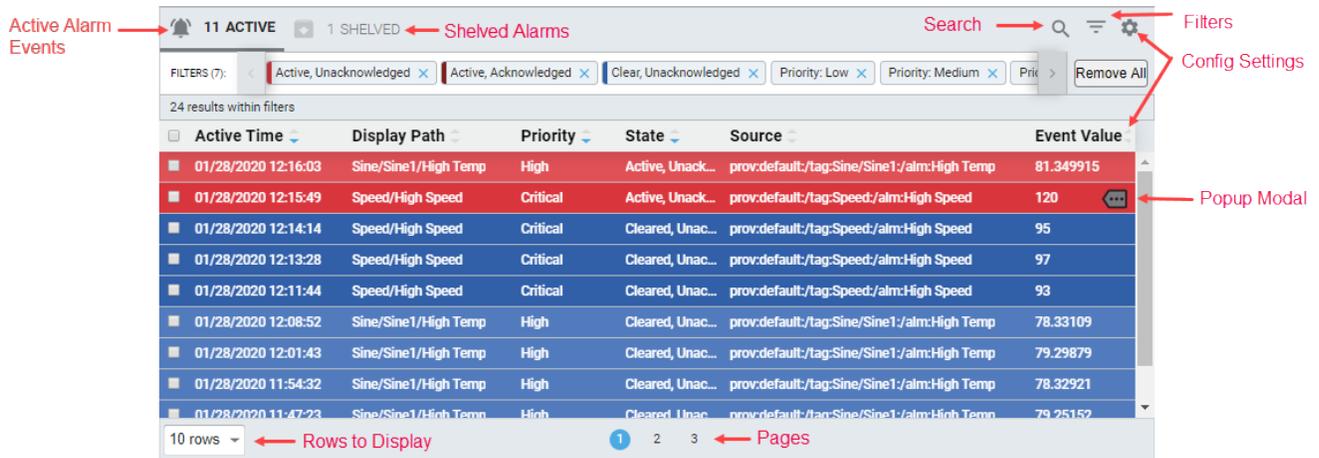
Columns can be reordered in Preview Mode and in a Perspective Session by clicking and dragging when the dragOrderable property is enabled.

Active Time	Priority	Display Path	Source	State
01/06/2022 10:29:26	High	Sensor 3 High Alarm	prov.default/tag:Sensor 3...	Active, Unacknowledged
01/06/2022 10:19:30	High	Sensor 3 High Alarm	prov.default/tag:Sensor 3...	Cleared, Unacknowledged
01/06/2022 10:38:17	High	Sensor 4 High Alarm	prov.default/tag:Sensor 4...	Cleared, Unacknowledged
01/06/2022 10:37:17	High	Sensor 8 High Alarm	prov.default/tag:Sensor 8...	Cleared, Unacknowledged
01/06/2022 10:37:04	High	Sensor 2 High Alarm	prov.default/tag:Sensor 2...	Cleared, Unacknowledged
01/06/2022 10:36:07	High	Sensor 1 High Alarm	prov.default/tag:Sensor 1...	Cleared, Unacknowledged
01/06/2022 10:35:47	High	Sensor 6 High Alarm	prov.default/tag:Sensor 6...	Cleared, Unacknowledged

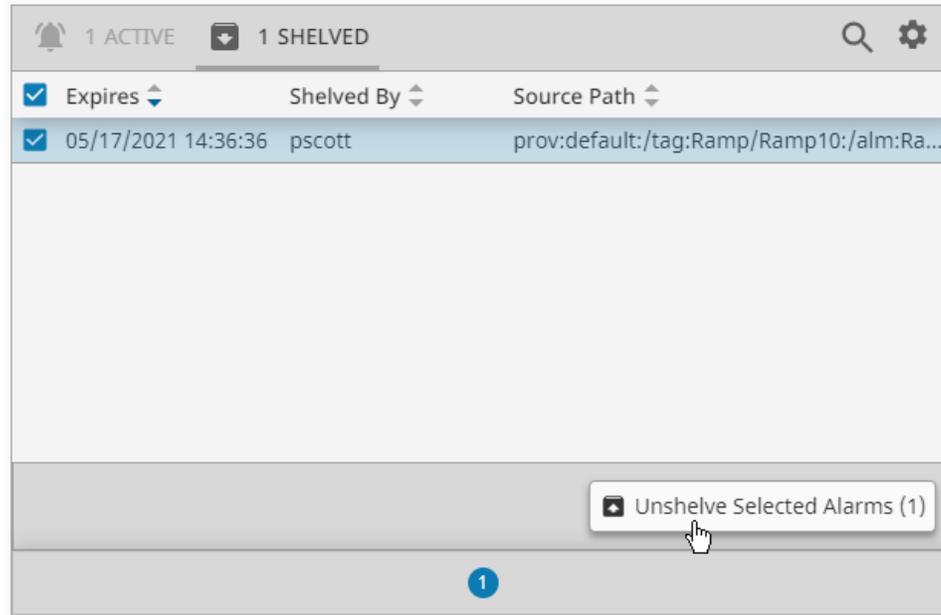
The following feature is new in Ignition version **8.1.12**
[Click here](#) to check out the other new features

The Alarm Status Table utilizes a shared polling engine to cache and share polling tasks across concurrent sessions. The cache persists for a period of time that matches the configured poll rate. This optimization allows multiple components to poll the alarm system with a reduced impact on overall performance.

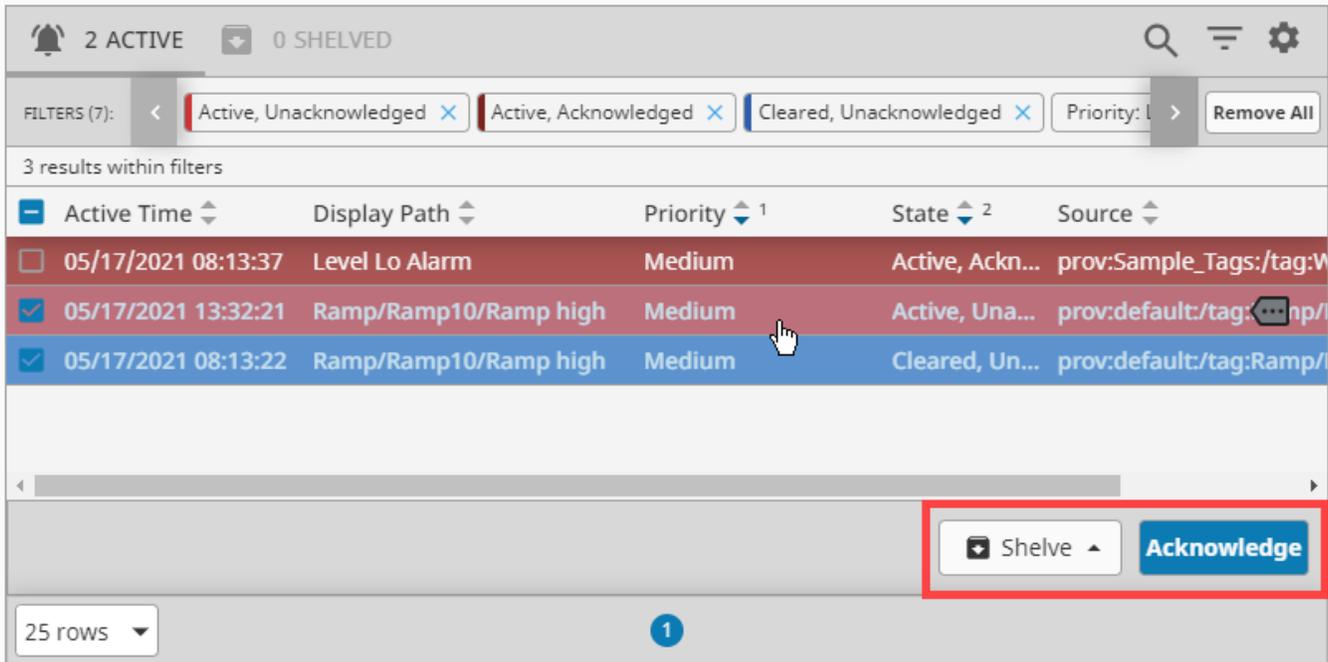
User Interface



Interaction	Description
Active Alarm Events	Shows the number of active alarms in the system. When viewing the shelved alarms view, clicking on the bell icon will switch the component back to the realtime view.
Config Settings	Clicking the  icon or right clicking on a column heading will open the Configuration menu, which determines which columns show appear on the component.
Filters	Clicking the  icon opens the Filters menu, allowing you to select event states and priorities to filter on.
Pages	A listing of pages. Long lists of alarm events are spread across multiple pages to improve performance. Clicking on a number will switch which page is shown.
Popup Modal	Hovering over a row in the table will cause the Popup Modal  icon to appear. Clicking this icon shows more information about the alarm.
Rows to Display	Determines how many rows are shown per page.
Search	Clicking the  icon cause a search bar to appear, allowing the user to type in search terms.
Shelved Alarms	Shows the number of shelved alarms. When viewing the realtime view, clicking on the Shelved Alarms display will switch to the shelved alarms view. While the table is showing shelved alarms, shelved alarms can be selected from the table and unshelved.



In addition, selecting a row within the table shows some additional interaction options.



The **Shelve** button allows you to shelve the selected alarms.

The **Acknowledge** button allows you to mark the selected alarms as "acknowledged".

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Prop Type
refresh Rate		value: numeri

The following feature is new in Ignition version **8.1.0**
[Click here](#) to check out the other new features

The rate at which the table will poll for updates in milliseconds.

enableHeader	Enable table header. Default is true.	value: boolean																											
enableDetails	Enable active events table details action. Default is true.	value: boolean																											
enableAcknowledge	Enable acknowledge action. Default is true.	value: boolean																											
enableShelve	Enable shelve action. Default is true.	value: boolean																											
enableUnshelve	Enable unshelve action. Default is true.	value: boolean																											
toolbar	Settings for the toolbar.	object																											
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enables the visibility of the table toolbar. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>enableActiveTab</td> <td>Enables the visibility of the Active Events tab.</td> <td>value: boolean</td> </tr> <tr> <td>enableShelvedTab</td> <td>Enables the visibility of the Shelved Events tab.</td> <td>value: boolean</td> </tr> <tr> <td>enableFilter</td> <td>Enables the visibility of the text filter toggle. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>toggleableFilter</td> <td> <div data-bbox="360 1108 1269 1176" data-label="Text"> <p>The following feature is new in Ignition version 8.1.18 Click here to check out the other new features</p> </div> <p>If false, the text filter will not require user interaction to open, and instead will remain open. Default is true.</p> </td> <td>value: boolean</td> </tr> <tr> <td>enableFilterResults</td> <td>Enables the visibility of the filters results count message. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>enablePreFilters</td> <td>Enables the visibility of the pre-filter toggle. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>enableConfiguration</td> <td>Enables the visibility of the configuration toggle. Default is true.</td> <td>value: boolean</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Enables the visibility of the table toolbar. Default is true.	value: boolean	enableActiveTab	Enables the visibility of the Active Events tab.	value: boolean	enableShelvedTab	Enables the visibility of the Shelved Events tab.	value: boolean	enableFilter	Enables the visibility of the text filter toggle. Default is true.	value: boolean	toggleableFilter	<div data-bbox="360 1108 1269 1176" data-label="Text"> <p>The following feature is new in Ignition version 8.1.18 Click here to check out the other new features</p> </div> <p>If false, the text filter will not require user interaction to open, and instead will remain open. Default is true.</p>	value: boolean	enableFilterResults	Enables the visibility of the filters results count message. Default is true.	value: boolean	enablePreFilters	Enables the visibility of the pre-filter toggle. Default is true.	value: boolean	enableConfiguration	Enables the visibility of the configuration toggle. Default is true.	value: boolean	
Name	Description	Property Type																											
enabled	Enables the visibility of the table toolbar. Default is true.	value: boolean																											
enableActiveTab	Enables the visibility of the Active Events tab.	value: boolean																											
enableShelvedTab	Enables the visibility of the Shelved Events tab.	value: boolean																											
enableFilter	Enables the visibility of the text filter toggle. Default is true.	value: boolean																											
toggleableFilter	<div data-bbox="360 1108 1269 1176" data-label="Text"> <p>The following feature is new in Ignition version 8.1.18 Click here to check out the other new features</p> </div> <p>If false, the text filter will not require user interaction to open, and instead will remain open. Default is true.</p>	value: boolean																											
enableFilterResults	Enables the visibility of the filters results count message. Default is true.	value: boolean																											
enablePreFilters	Enables the visibility of the pre-filter toggle. Default is true.	value: boolean																											
enableConfiguration	Enables the visibility of the configuration toggle. Default is true.	value: boolean																											
shelvingTimes	Available alarming shelving times in seconds. Shelving times are customizable by editing values for this property in the Property Editor.	array																											
responsive	Responsive layout configuration. Rows are converted to cards. While in responsive layout, disables or removes certain table features that are no longer applicable.	object																											
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enables responsive layout. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>breakpoint</td> <td>Width in pixels that triggers change in responsive layout.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Enables responsive layout. Default is false.	value: boolean	breakpoint	Width in pixels that triggers change in responsive layout.	value: numeric																			
Name	Description	Property Type																											
enabled	Enables responsive layout. Default is false.	value: boolean																											
breakpoint	Width in pixels that triggers change in responsive layout.	value: numeric																											
filters	This is where you configure filtering properties for displaying alarm data in the Alarm Status Table.	object																											
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property</th> </tr> </thead> <tbody> </tbody> </table>	Name	Description	Property																									
Name	Description	Property																											

		Type																																																																																			
active	Settings for active alarms.	object																																																																																			
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>The active alarm events filter text.</td> <td>value: string</td> </tr> <tr> <td>states</td> <td>Pre-filters for filter active alarm events:</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>activeUnack ed</td> <td>Active and unacknowledged. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>activeAcked</td> <td>Active and acknowledged. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>clearUnacked</td> <td>Cleared and unacknowledged. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>clearAcked</td> <td>Active and acknowledged. Default is false.</td> <td>value: boolean</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>priorities</td> <td>Alarm state priority pre-filters.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>diagnos tic</td> <td>Whether to display alarms with diagnostic priority. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>low</td> <td>Whether to display alarms with low priority. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>medium</td> <td>Whether to display alarms with medium priority. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>high</td> <td>Whether to display alarms with high priority. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>critical</td> <td>Whether to display alarms with critical priority. Default is true.</td> <td>value: boolean</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>conditio ns</td> <td>Gateway side alarm query conditions.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>source</td> <td>Filter alarms by alarm source path. Specify multiple paths by separating them with commas. Supports the wildcard *.</td> <td>value: string</td> </tr> <tr> <td>display Path</td> <td>Filters alarms by display path, falling back to the source path if a custom display path isn't set. Specify multiple paths by separating them with commas. Supports the wildcard *.</td> <td>value: string</td> </tr> <tr> <td>provider</td> <td>Filter alarms by alarm provider.</td> <td>value: string</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>results</td> <td>Active alarm filtering results configuration and data.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enable filter results to be written back to props. Warning: Doing so may cause performance decline. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>data</td> <td>An array of objects representing the current filtered data if enabled and active.</td> <td>array</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	text	The active alarm events filter text.	value: string	states	Pre-filters for filter active alarm events:	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>activeUnack ed</td> <td>Active and unacknowledged. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>activeAcked</td> <td>Active and acknowledged. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>clearUnacked</td> <td>Cleared and unacknowledged. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>clearAcked</td> <td>Active and acknowledged. Default is false.</td> <td>value: boolean</td> </tr> </tbody> </table>	Name	Description	Property Type	activeUnack ed	Active and unacknowledged. Default is true.	value: boolean	activeAcked	Active and acknowledged. Default is true.	value: boolean	clearUnacked	Cleared and unacknowledged. Default is true.	value: boolean	clearAcked	Active and acknowledged. Default is false.	value: boolean		priorities	Alarm state priority pre-filters.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>diagnos tic</td> <td>Whether to display alarms with diagnostic priority. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>low</td> <td>Whether to display alarms with low priority. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>medium</td> <td>Whether to display alarms with medium priority. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>high</td> <td>Whether to display alarms with high priority. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>critical</td> <td>Whether to display alarms with critical priority. Default is true.</td> <td>value: boolean</td> </tr> </tbody> </table>	Name	Description	Property Type	diagnos tic	Whether to display alarms with diagnostic priority. Default is false.	value: boolean	low	Whether to display alarms with low priority. Default is true.	value: boolean	medium	Whether to display alarms with medium priority. Default is true.	value: boolean	high	Whether to display alarms with high priority. Default is true.	value: boolean	critical	Whether to display alarms with critical priority. Default is true.	value: boolean		conditio ns	Gateway side alarm query conditions.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>source</td> <td>Filter alarms by alarm source path. Specify multiple paths by separating them with commas. Supports the wildcard *.</td> <td>value: string</td> </tr> <tr> <td>display Path</td> <td>Filters alarms by display path, falling back to the source path if a custom display path isn't set. Specify multiple paths by separating them with commas. Supports the wildcard *.</td> <td>value: string</td> </tr> <tr> <td>provider</td> <td>Filter alarms by alarm provider.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	source	Filter alarms by alarm source path. Specify multiple paths by separating them with commas. Supports the wildcard *.	value: string	display Path	Filters alarms by display path, falling back to the source path if a custom display path isn't set. Specify multiple paths by separating them with commas. Supports the wildcard *.	value: string	provider	Filter alarms by alarm provider.	value: string		results	Active alarm filtering results configuration and data.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enable filter results to be written back to props. Warning: Doing so may cause performance decline. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>data</td> <td>An array of objects representing the current filtered data if enabled and active.</td> <td>array</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Enable filter results to be written back to props. Warning: Doing so may cause performance decline. Default is false.	value: boolean	data	An array of objects representing the current filtered data if enabled and active.	array	
Name	Description	Property Type																																																																																			
text	The active alarm events filter text.	value: string																																																																																			
states	Pre-filters for filter active alarm events:	object																																																																																			
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>activeUnack ed</td> <td>Active and unacknowledged. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>activeAcked</td> <td>Active and acknowledged. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>clearUnacked</td> <td>Cleared and unacknowledged. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>clearAcked</td> <td>Active and acknowledged. Default is false.</td> <td>value: boolean</td> </tr> </tbody> </table>	Name	Description	Property Type	activeUnack ed	Active and unacknowledged. Default is true.	value: boolean	activeAcked	Active and acknowledged. Default is true.	value: boolean	clearUnacked	Cleared and unacknowledged. Default is true.	value: boolean	clearAcked	Active and acknowledged. Default is false.	value: boolean																																																																					
Name	Description	Property Type																																																																																			
activeUnack ed	Active and unacknowledged. Default is true.	value: boolean																																																																																			
activeAcked	Active and acknowledged. Default is true.	value: boolean																																																																																			
clearUnacked	Cleared and unacknowledged. Default is true.	value: boolean																																																																																			
clearAcked	Active and acknowledged. Default is false.	value: boolean																																																																																			
priorities	Alarm state priority pre-filters.	object																																																																																			
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>diagnos tic</td> <td>Whether to display alarms with diagnostic priority. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>low</td> <td>Whether to display alarms with low priority. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>medium</td> <td>Whether to display alarms with medium priority. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>high</td> <td>Whether to display alarms with high priority. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>critical</td> <td>Whether to display alarms with critical priority. Default is true.</td> <td>value: boolean</td> </tr> </tbody> </table>	Name	Description	Property Type	diagnos tic	Whether to display alarms with diagnostic priority. Default is false.	value: boolean	low	Whether to display alarms with low priority. Default is true.	value: boolean	medium	Whether to display alarms with medium priority. Default is true.	value: boolean	high	Whether to display alarms with high priority. Default is true.	value: boolean	critical	Whether to display alarms with critical priority. Default is true.	value: boolean																																																																		
Name	Description	Property Type																																																																																			
diagnos tic	Whether to display alarms with diagnostic priority. Default is false.	value: boolean																																																																																			
low	Whether to display alarms with low priority. Default is true.	value: boolean																																																																																			
medium	Whether to display alarms with medium priority. Default is true.	value: boolean																																																																																			
high	Whether to display alarms with high priority. Default is true.	value: boolean																																																																																			
critical	Whether to display alarms with critical priority. Default is true.	value: boolean																																																																																			
conditio ns	Gateway side alarm query conditions.	object																																																																																			
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>source</td> <td>Filter alarms by alarm source path. Specify multiple paths by separating them with commas. Supports the wildcard *.</td> <td>value: string</td> </tr> <tr> <td>display Path</td> <td>Filters alarms by display path, falling back to the source path if a custom display path isn't set. Specify multiple paths by separating them with commas. Supports the wildcard *.</td> <td>value: string</td> </tr> <tr> <td>provider</td> <td>Filter alarms by alarm provider.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	source	Filter alarms by alarm source path. Specify multiple paths by separating them with commas. Supports the wildcard *.	value: string	display Path	Filters alarms by display path, falling back to the source path if a custom display path isn't set. Specify multiple paths by separating them with commas. Supports the wildcard *.	value: string	provider	Filter alarms by alarm provider.	value: string																																																																								
Name	Description	Property Type																																																																																			
source	Filter alarms by alarm source path. Specify multiple paths by separating them with commas. Supports the wildcard *.	value: string																																																																																			
display Path	Filters alarms by display path, falling back to the source path if a custom display path isn't set. Specify multiple paths by separating them with commas. Supports the wildcard *.	value: string																																																																																			
provider	Filter alarms by alarm provider.	value: string																																																																																			
results	Active alarm filtering results configuration and data.	object																																																																																			
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enable filter results to be written back to props. Warning: Doing so may cause performance decline. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>data</td> <td>An array of objects representing the current filtered data if enabled and active.</td> <td>array</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Enable filter results to be written back to props. Warning: Doing so may cause performance decline. Default is false.	value: boolean	data	An array of objects representing the current filtered data if enabled and active.	array																																																																											
Name	Description	Property Type																																																																																			
enabled	Enable filter results to be written back to props. Warning: Doing so may cause performance decline. Default is false.	value: boolean																																																																																			
data	An array of objects representing the current filtered data if enabled and active.	array																																																																																			

shelved	Temporarily silence an alarm for a fixed period of time while the alarm event issue is worked on.	object																					
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>The filter text for shelved alarms.</td> <td>value: string</td> </tr> <tr> <td>results</td> <td>Shelved alarm filtering results configuration and data.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enable filter results to be written back to props. Warning: Doing so may cause performance decline. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>data</td> <td>An array of objects representing the current filtered data if enabled and active.</td> <td>array</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	text	The filter text for shelved alarms.	value: string	results	Shelved alarm filtering results configuration and data.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enable filter results to be written back to props. Warning: Doing so may cause performance decline. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>data</td> <td>An array of objects representing the current filtered data if enabled and active.</td> <td>array</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Enable filter results to be written back to props. Warning: Doing so may cause performance decline. Default is false.	value: boolean	data	An array of objects representing the current filtered data if enabled and active.	array		
Name	Description	Property Type																					
text	The filter text for shelved alarms.	value: string																					
results	Shelved alarm filtering results configuration and data.	object																					
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enable filter results to be written back to props. Warning: Doing so may cause performance decline. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>data</td> <td>An array of objects representing the current filtered data if enabled and active.</td> <td>array</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Enable filter results to be written back to props. Warning: Doing so may cause performance decline. Default is false.	value: boolean	data	An array of objects representing the current filtered data if enabled and active.	array													
Name	Description	Property Type																					
enabled	Enable filter results to be written back to props. Warning: Doing so may cause performance decline. Default is false.	value: boolean																					
data	An array of objects representing the current filtered data if enabled and active.	array																					

selection	<p>The following feature is new in Ignition version 8.1.4 Click here to check out the other new features</p> <p>Currently selected alarms and alarm selection configuration.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>active</td> <td>Active alarm selection configuration and read-only list of currently selected active alarms.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>mode</td> <td>Active alarm selection configuration. Options are multiple, single, or none.</td> <td>value: string</td> </tr> <tr> <td>data</td> <td>A read-only list of currently selected active alarms.</td> <td>value: array</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>shelved</td> <td>Shelved alarm configuration and read-only list of currently selected shelved alarms</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>mode</td> <td>Shelved alarm selection configuration. Options are multiple, single, or none.</td> <td>value: string</td> </tr> <tr> <td>data</td> <td>A read-only list of currently selected active alarms.</td> <td>value: array</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	active	Active alarm selection configuration and read-only list of currently selected active alarms.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>mode</td> <td>Active alarm selection configuration. Options are multiple, single, or none.</td> <td>value: string</td> </tr> <tr> <td>data</td> <td>A read-only list of currently selected active alarms.</td> <td>value: array</td> </tr> </tbody> </table>	Name	Description	Property Type	mode	Active alarm selection configuration. Options are multiple, single, or none.	value: string	data	A read-only list of currently selected active alarms.	value: array		shelved	Shelved alarm configuration and read-only list of currently selected shelved alarms	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>mode</td> <td>Shelved alarm selection configuration. Options are multiple, single, or none.</td> <td>value: string</td> </tr> <tr> <td>data</td> <td>A read-only list of currently selected active alarms.</td> <td>value: array</td> </tr> </tbody> </table>	Name	Description	Property Type	mode	Shelved alarm selection configuration. Options are multiple, single, or none.	value: string	data	A read-only list of currently selected active alarms.	value: array		object
Name	Description	Property Type																																	
active	Active alarm selection configuration and read-only list of currently selected active alarms.	object																																	
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>mode</td> <td>Active alarm selection configuration. Options are multiple, single, or none.</td> <td>value: string</td> </tr> <tr> <td>data</td> <td>A read-only list of currently selected active alarms.</td> <td>value: array</td> </tr> </tbody> </table>	Name	Description	Property Type	mode	Active alarm selection configuration. Options are multiple, single, or none.	value: string	data	A read-only list of currently selected active alarms.	value: array																									
Name	Description	Property Type																																	
mode	Active alarm selection configuration. Options are multiple, single, or none.	value: string																																	
data	A read-only list of currently selected active alarms.	value: array																																	
shelved	Shelved alarm configuration and read-only list of currently selected shelved alarms	object																																	
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>mode</td> <td>Shelved alarm selection configuration. Options are multiple, single, or none.</td> <td>value: string</td> </tr> <tr> <td>data</td> <td>A read-only list of currently selected active alarms.</td> <td>value: array</td> </tr> </tbody> </table>	Name	Description	Property Type	mode	Shelved alarm selection configuration. Options are multiple, single, or none.	value: string	data	A read-only list of currently selected active alarms.	value: array																									
Name	Description	Property Type																																	
mode	Shelved alarm selection configuration. Options are multiple, single, or none.	value: string																																	
data	A read-only list of currently selected active alarms.	value: array																																	

rowStyles	Styles to apply to rows given their alarm state and designated priority.	object																											
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>activeUnacked</td> <td>Style settings for rows with activeUnacked alarms.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>base</td> <td>Base style settings for activeUnacked alarms. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>priorities</td> <td>Style settings for the alarm row based on priority.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	activeUnacked	Style settings for rows with activeUnacked alarms.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>base</td> <td>Base style settings for activeUnacked alarms. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>priorities</td> <td>Style settings for the alarm row based on priority.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	base	Base style settings for activeUnacked alarms. Full menu of style options is available. You can also specify a style class .	object	priorities	Style settings for the alarm row based on priority.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type						
Name	Description	Property Type																											
activeUnacked	Style settings for rows with activeUnacked alarms.	object																											
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>base</td> <td>Base style settings for activeUnacked alarms. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>priorities</td> <td>Style settings for the alarm row based on priority.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	base	Base style settings for activeUnacked alarms. Full menu of style options is available. You can also specify a style class .	object	priorities	Style settings for the alarm row based on priority.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type														
Name	Description	Property Type																											
base	Base style settings for activeUnacked alarms. Full menu of style options is available. You can also specify a style class .	object																											
priorities	Style settings for the alarm row based on priority.	object																											
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type																									
Name	Description	Property Type																											

diagnostic	Style for activeUnacked alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class .	object
low	Style for activeUnacked alarms with low priority. Full menu of style options is available. You can also specify a style class .	object
medium	Style for activeUnacked alarms with medium priority. Full menu of style options is available. You can also specify a style class .	object
high	Style for activeUnacked alarms with high priority. Full menu of style options is available. You can also specify a style class .	object
critical	Style for activeUnacked alarms with critical priority. Full menu of style options is available. You can also specify a style class .	object

activeA
cked

Style settings for rows with activeAacked alarms.

object

Name	Description	Property Type																		
base	Base style settings for activeAacked alarms. Full menu of style options is available. You can also specify a style class .	object																		
priorities	Style settings for the alarm row based on priority.	object																		
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>diagnostic</td> <td>Style for activeAacked alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>low</td> <td>Style for activeAacked alarms with low priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>medium</td> <td>Style for activeAacked alarms with medium priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>high</td> <td>Style for activeAacked alarms with high priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>critical</td> <td>Style for activeAacked alarms with critical priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	diagnostic	Style for activeAacked alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class .	object	low	Style for activeAacked alarms with low priority. Full menu of style options is available. You can also specify a style class .	object	medium	Style for activeAacked alarms with medium priority. Full menu of style options is available. You can also specify a style class .	object	high	Style for activeAacked alarms with high priority. Full menu of style options is available. You can also specify a style class .	object	critical	Style for activeAacked alarms with critical priority. Full menu of style options is available. You can also specify a style class .	object	
Name	Description	Property Type																		
diagnostic	Style for activeAacked alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class .	object																		
low	Style for activeAacked alarms with low priority. Full menu of style options is available. You can also specify a style class .	object																		
medium	Style for activeAacked alarms with medium priority. Full menu of style options is available. You can also specify a style class .	object																		
high	Style for activeAacked alarms with high priority. Full menu of style options is available. You can also specify a style class .	object																		
critical	Style for activeAacked alarms with critical priority. Full menu of style options is available. You can also specify a style class .	object																		

clearUn
acked

Style settings for rows with clearUnacked alarms.

object

Name	Description	Property Type												
base	Base style settings for clearUnacked alarms. Full menu of style options is available. You can also specify a style class .	object												
priorities	Style settings for the alarm row based on priority.	object												
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>diagnostic</td> <td>Style for clearUnacked alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>low</td> <td>Style for clearUnacked alarms with low priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>medium</td> <td>Style for clearUnacked alarms with medium priority. Full menu of style options is available.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	diagnostic	Style for clearUnacked alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class .	object	low	Style for clearUnacked alarms with low priority. Full menu of style options is available. You can also specify a style class .	object	medium	Style for clearUnacked alarms with medium priority. Full menu of style options is available.	object	
Name	Description	Property Type												
diagnostic	Style for clearUnacked alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class .	object												
low	Style for clearUnacked alarms with low priority. Full menu of style options is available. You can also specify a style class .	object												
medium	Style for clearUnacked alarms with medium priority. Full menu of style options is available.	object												

	You can also specify a style class .	
high	Style for clearUnacked alarms with high priority. Full menu of style options is available. You can also specify a style class .	object
critical	Style for clearUnacked alarms with critical priority. Full menu of style options is available. You can also specify a style class .	object

clearAcked	Style settings for rows with clearAcked alarms.	object																														
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>base</td> <td>Base style settings for clearAcked alarms. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>priorities</td> <td>Style settings for the alarm row based on priority.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>diagnostic</td> <td>Style for clearAcked alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>low</td> <td>Style for clearAcked alarms with low priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>medium</td> <td>Style for clearAcked alarms with medium priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>high</td> <td>Style for clearUnacked alarms with high priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>critical</td> <td>Style for clearAcked alarms with critical priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	base	Base style settings for clearAcked alarms. Full menu of style options is available. You can also specify a style class .	object	priorities	Style settings for the alarm row based on priority.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>diagnostic</td> <td>Style for clearAcked alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>low</td> <td>Style for clearAcked alarms with low priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>medium</td> <td>Style for clearAcked alarms with medium priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>high</td> <td>Style for clearUnacked alarms with high priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>critical</td> <td>Style for clearAcked alarms with critical priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	diagnostic	Style for clearAcked alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class .	object	low	Style for clearAcked alarms with low priority. Full menu of style options is available. You can also specify a style class .	object	medium	Style for clearAcked alarms with medium priority. Full menu of style options is available. You can also specify a style class .	object	high	Style for clearUnacked alarms with high priority. Full menu of style options is available. You can also specify a style class .	object	critical	Style for clearAcked alarms with critical priority. Full menu of style options is available. You can also specify a style class .	object		
Name	Description	Property Type																														
base	Base style settings for clearAcked alarms. Full menu of style options is available. You can also specify a style class .	object																														
priorities	Style settings for the alarm row based on priority.	object																														
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>diagnostic</td> <td>Style for clearAcked alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>low</td> <td>Style for clearAcked alarms with low priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>medium</td> <td>Style for clearAcked alarms with medium priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>high</td> <td>Style for clearUnacked alarms with high priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>critical</td> <td>Style for clearAcked alarms with critical priority. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	diagnostic	Style for clearAcked alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class .	object	low	Style for clearAcked alarms with low priority. Full menu of style options is available. You can also specify a style class .	object	medium	Style for clearAcked alarms with medium priority. Full menu of style options is available. You can also specify a style class .	object	high	Style for clearUnacked alarms with high priority. Full menu of style options is available. You can also specify a style class .	object	critical	Style for clearAcked alarms with critical priority. Full menu of style options is available. You can also specify a style class .	object													
Name	Description	Property Type																														
diagnostic	Style for clearAcked alarms with diagnostic priority. Full menu of style options is available. You can also specify a style class .	object																														
low	Style for clearAcked alarms with low priority. Full menu of style options is available. You can also specify a style class .	object																														
medium	Style for clearAcked alarms with medium priority. Full menu of style options is available. You can also specify a style class .	object																														
high	Style for clearUnacked alarms with high priority. Full menu of style options is available. You can also specify a style class .	object																														
critical	Style for clearAcked alarms with critical priority. Full menu of style options is available. You can also specify a style class .	object																														
dateFormat	A date format string to be applied against dates.	value: string																														

activeSortOrder	The default weighted order in which columns and their contents are sorted relative to other columns and their contents. Used when the component loads. Active event columns need to have sort configured in order for this to work.	array
-----------------	---	-------

shelvedSortOrder	The default weighted order in which columns and their contents are sorted relative to other columns and their contents. Used when the component loads. Shelved event columns need to have sort configured in order for this to work.	array
------------------	--	-------

dragOrderable	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.14 Click here to check out the other new features</p> </div> <p>When enabled, users may drag column headers to reorder columns in the table.</p>	value: boolean
---------------	---	----------------

columns	Used only for determining what columns to show on load.	object															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>active</td> <td>Active alarm event columns to display on load</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>activeTi</td> <td>Settings for the activeTime column.</td> <td>object</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	active	Active alarm event columns to display on load	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>activeTi</td> <td>Settings for the activeTime column.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	activeTi	Settings for the activeTime column.	object		
Name	Description	Property Type															
active	Active alarm event columns to display on load	object															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>activeTi</td> <td>Settings for the activeTime column.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	activeTi	Settings for the activeTime column.	object										
Name	Description	Property Type															
activeTi	Settings for the activeTime column.	object															

me

Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

display Path

Settings for the displayPath column.

object

Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

priority

Settings for the priority column.

object

Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

state

Settings for the state column.

object

Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean

	<table border="1"> <tr> <td>sort</td> <td>Default sort order of the column. Options are none, ascending, or descending.</td> <td>value: string</td> </tr> </table>	sort	Default sort order of the column. Options are none, ascending, or descending.	value: string													
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string															
source	<p>Settings for the source column.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether the column is enabled. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>width</td> <td>The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.</td> <td>value: numeric</td> </tr> <tr> <td>strictWidth</td> <td>If enabled, the width of the column (set with the width property) becomes static. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>sort</td> <td>Default sort order of the column. Options are none, ascending, or descending.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Whether the column is enabled. Default is true.	value: boolean	width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric	strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean	sort	Default sort order of the column. Options are none, ascending, or descending.	value: string	object
Name	Description	Property Type															
enabled	Whether the column is enabled. Default is true.	value: boolean															
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric															
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean															
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string															
label	<p>Settings for the label column.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether the column is enabled. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>width</td> <td>The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.</td> <td>value: numeric</td> </tr> <tr> <td>strictWidth</td> <td>If enabled, the width of the column (set with the width property) becomes static. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>sort</td> <td>Default sort order of the column. Options are none, ascending, or descending.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Whether the column is enabled. Default is true.	value: boolean	width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric	strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean	sort	Default sort order of the column. Options are none, ascending, or descending.	value: string	object
Name	Description	Property Type															
enabled	Whether the column is enabled. Default is true.	value: boolean															
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric															
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean															
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string															
name	<p>Settings for the name column.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether the column is enabled. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>width</td> <td>The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.</td> <td>value: numeric</td> </tr> <tr> <td>strictWidth</td> <td>If enabled, the width of the column (set with the width property) becomes static. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>sort</td> <td>Default sort order of the column. Options are none, ascending, or descending.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Whether the column is enabled. Default is true.	value: boolean	width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric	strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean	sort	Default sort order of the column. Options are none, ascending, or descending.	value: string	object
Name	Description	Property Type															
enabled	Whether the column is enabled. Default is true.	value: boolean															
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric															
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean															
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string															
eventId	<p>Settings for the eventId column.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether the column is enabled. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>width</td> <td>The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Whether the column is enabled. Default is true.	value: boolean	width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric	object						
Name	Description	Property Type															
enabled	Whether the column is enabled. Default is true.	value: boolean															
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric															

strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

eventValue	Settings for the eventValue column.	object
Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

notes	Settings for the notes column.	object
Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

isActive	Settings for the isActive column.	object
Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

isAked	Settings for the isAked column.	object
Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict	value:

	represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

isClear Settings for the isClear column. object

Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

ackTime Settings for the ackTime column. object

Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

ackUser Settings for the ackUser column. object

Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

ackNotes Settings for the ackNotes column. object

Name	Description	Property Type

enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

ackPipeline Settings for the ackPipeline column. object

Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

activePipeline Settings for the activePipeline column. object

Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

clearTime Settings for the clearTime column. object

Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

clearPipeline Settings for the clearPipeline column. object

eline

Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

deadband

Settings for the deadband column.

object

Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

shelved

Shelved alarm columns to display on load.

object

Name	Description	Property Type															
expires	Settings for the expires column.																
	<table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>enabled</td><td>Whether the column is enabled. Default is true.</td><td>value: boolean</td></tr><tr><td>width</td><td>The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.</td><td>value: numeric</td></tr><tr><td>strictWidth</td><td>If enabled, the width of the column (set with the width property) becomes static. Default is false.</td><td>value: boolean</td></tr><tr><td>sort</td><td>Default sort order of the column. Options are none, ascending, or descending.</td><td>value: string</td></tr></tbody></table>	Name	Description	Property Type	enabled	Whether the column is enabled. Default is true.	value: boolean	width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric	strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean	sort	Default sort order of the column. Options are none, ascending, or descending.	value: string	
Name	Description	Property Type															
enabled	Whether the column is enabled. Default is true.	value: boolean															
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric															
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean															
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string															
shelvedBy	Settings for the shelvedBy column.																
	<table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>enabled</td><td>Whether the column is enabled. Default is true.</td><td>value: boolean</td></tr><tr><td>width</td><td>The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.</td><td>value: numeric</td></tr></tbody></table>	Name	Description	Property Type	enabled	Whether the column is enabled. Default is true.	value: boolean	width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric							
Name	Description	Property Type															
enabled	Whether the column is enabled. Default is true.	value: boolean															
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric															

strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

sourcePath	Settings for the sourcePath expires column.	
Name	Description	Property Type
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string

columnsAssociated

The following feature is new in Ignition version **8.1.14**
[Click here](#) to check out the other new features

A list of columns used to retrieve and display alarm associated data.

Name	Description	Property Type
active	Active alarm event associated data columns to display on load.	array
Name	Description	Property Type
field	Maps to the associated data value represented by the column.	value: string
enabled	Whether the column is enabled. Default is true.	value: boolean
width	The column's width, which when not strict represents a proportion of the available space, i.e., flex grow. If strictWidth is enabled, the column will be fixed and static.	value: numeric
strictWidth	If enabled, the width of the column (set with the width property) becomes static. Default is false.	value: boolean
sort	Default sort order of the column. Options are none, ascending, or descending.	value: string
order	Order to display this column in the table.	value: numeric

pager

Settings for the pager.

Name	Description	Property Type
enabled	Enables the pager to be displayed. Default is true.	value: boolean
hide	Visually hides the pager from view. Useful when pager is manipulated in a controlled fashion via the activePage property. Default is false.	value: boolean

object

object

options	Rows to show per pager option.	array
initialOption	The initial option to use when the table first loads. It must exist as an available option.	value: numeric
activePage	Represents the current active page and corresponds to the value of the page jump input field.	value: numeric
shelvedPage	Represents the current shelved page and corresponds to the value of the page jump input field.	value: numeric
style	Sets a style that applies to the component. Full menu of style options is available. You can also specify a style class .	

Scripting

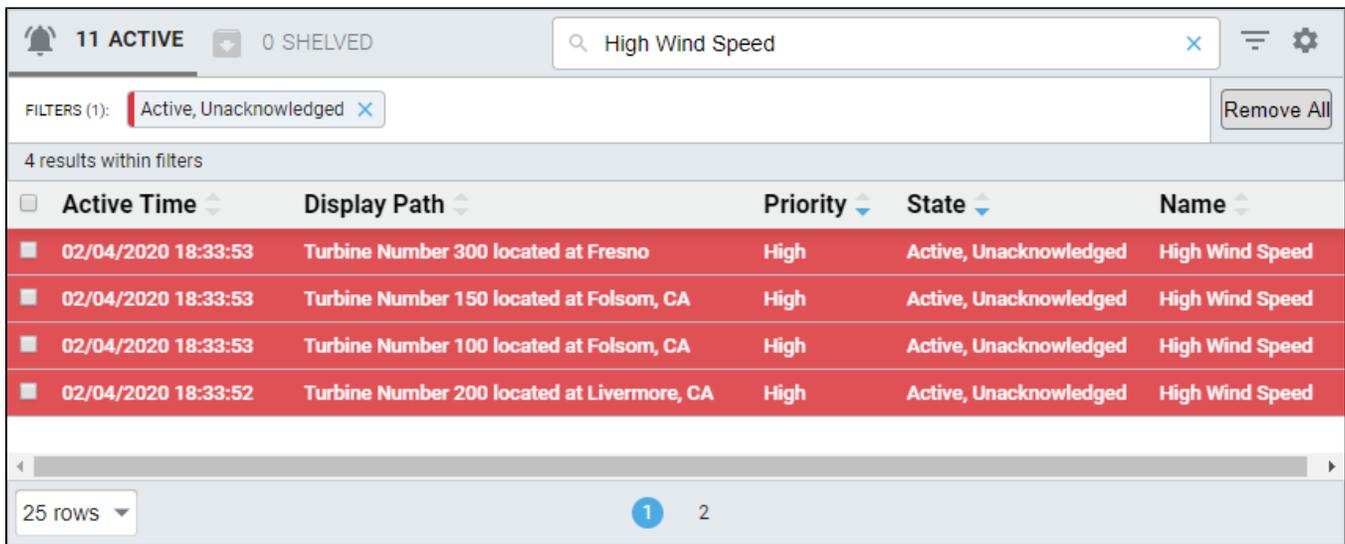
See the [Perspective - Alarm Status Table Scripting](#) page for the full list of scripting functions available for this component.

Examples

Example 1

In a Perspective Session, click on the **Filter** button  to filter on alarm states and/or use the **Search Bar**  to target more specific alarm events. Select from the **Configuration Settings**  to change the column headers to display the alarm event data you're interested in.

This example uses the Search Bar to find any text with 'High Wind Speed' in the Active,Unacknowledged alarm state.



Active Time	Display Path	Priority	State	Name
02/04/2020 18:33:53	Turbine Number 300 located at Fresno	High	Active, Unacknowledged	High Wind Speed
02/04/2020 18:33:53	Turbine Number 150 located at Folsom, CA	High	Active, Unacknowledged	High Wind Speed
02/04/2020 18:33:53	Turbine Number 100 located at Folsom, CA	High	Active, Unacknowledged	High Wind Speed
02/04/2020 18:33:52	Turbine Number 200 located at Livermore, CA	High	Active, Unacknowledged	High Wind Speed

Example 2 - Alarm Status Table Row Styles

In the Designer, you can change row styles to be different colors for the different priorities for each alarm state. In this example, the rowStyle for the Critical priority for the activeAcked alarm state was changed to green.

The image shows a software interface with an Alarms table and a Perspective Property Editor. The Alarms table has columns for Active Time, Display Path, Priority, State, and Source. It contains three rows: a red row for 'High Temp/High Temp', a red row for 'Tank Level 2/Low SP2', and a green row for 'Speed/High Speed'. The Perspective Property Editor shows a tree view of row styles: 'activeUnacked (2)', 'activeAcked (2)', and 'critical (1)'. A color wheel is open, showing the RGB value (0, 128, 0) for a green color, which is highlighted with a red arrow pointing to the 'critical (1)' row style's 'backgroundColor' property in the editor.

Active Time	Display Path	Priority	State	Source
08/02/2019 13:29:13	High Temp/High Temp	Critical	Active, Unacknowledged	prov.default:/tag:High Temp/alm:High Temp
08/02/2019 08:30:54	Tank Level 2/Low SP2	Critical	Active, Unacknowledged	prov.default:/tag:Tank Level 2/alm:Low SP2
08/02/2019 08:30:28	Speed/High Speed	Critical	Active, Acknowledged	prov.default:/tag:Speed/alm:High Speed

Perspective Property Editor

- rowStyles (4)
 - activeUnacked (2)
 - base (4)
 - priorities (5)
 - diagnostic (2)
 - low (1)
 - medium (1)
 - high (1)
 - critical (1)
 - backgroundColor : #...
 - activeAcked (2)
 - base (3)
 - priorities (5)
 - diagnostic (2)
 - low (1)
 - medium (1)
 - high (1)
 - critical (1)
 - backgroundColor : #...

Color Wheel: (0, 128, 0)

Hex: 008000

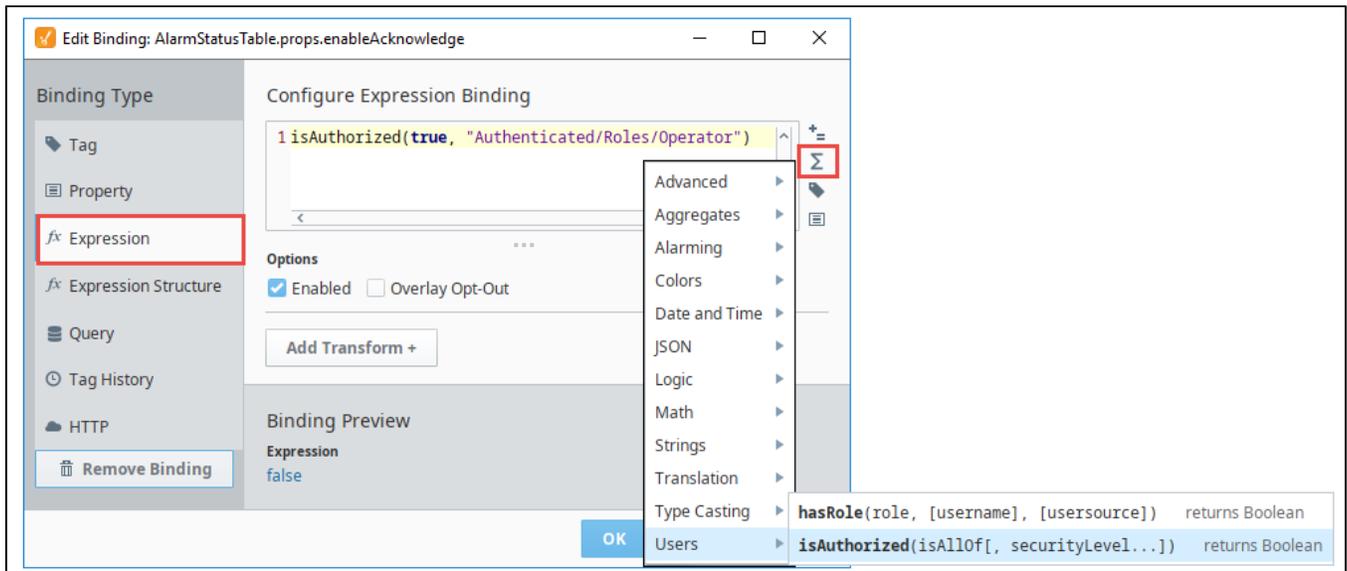
How To Restrict Acknowledgement on the Perspective Alarm Status Table

Security for Alarm Acknowledgement

You can restrict specific users or roles from Acknowledging alarms by setting the **enableAcknowledge** property in the Property Editor to **'false.'** This hides the Acknowledge button on the Alarm Status Table for those users who do not have permission. You can setup permissions for any [role](#), user and [user source](#) in your system.

For example, if you only want those users with the Operator role to acknowledge alarms, the correct permission must be assigned.

1. Select the Alarm Status Table component, and click the **enableAcknowledge binding**  icon to open the Property Binding window.
2. Under **Property Binding Type**, select **Expression**.
3. Click the **Function**  icon and scroll down to **Users**, and select **'isAuthorized.'** This enters the function name.
4. Edit the expression to read: **isAuthorized(true, "Authenticated/Roles/Operator")**
5. Click **OK**.



If you currently have the 'Operator' role, you'll notice in the Property Editor of the Designer that the **enableAcknowledge** property is set to **'true,'** and for other roles, it will be set to **'false.'**

Perspective - Alarm Status Table Scripting

This page details the various scripting, component, and extension functions available for [Perspective's Alarm Status Table](#) component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

On this page ...

- [Component Events](#)
- [Component Functions](#)
 - [.refreshData\(\)](#)
- [Extension Functions](#)
 - [filterAlarm](#)
 - [filterShelvedAlarm](#)

Component Functions

.refreshData()

The following feature is new in Ignition version **8.1.18**
[Click here](#) to check out the other new features

- Description
Refreshes the data on the Alarm Status Table.
- Parameters
None
- Return
Nothing

Extension Functions

filterAlarm

The following feature is new in Ignition version **8.1.0**
[Click here](#) to check out the other new features

- Description
Called for each event before it is displayed in the table, allowing you to hide or show each alarm event (row) in the table. Provides an opportunity to write a more complex filter than what's normally provided to the component. Return False to exclude an alarm event from the table.
- Parameters
[ComponentModelScriptWrapper.SafetyWrapper](#) self- A reference to the component that is invoking this function.
[PyAlarmEvent](#) alarmEvent - The alarm event itself. Call `alarmEvent.get('propertyName')` to inspect properties on the event. Common properties: 'name', 'source', 'priority'.
- Return
[Boolean](#) - The function must return either a True or False for every alarm event in the table. True will show the alarm. False will hide the alarm.

Examples

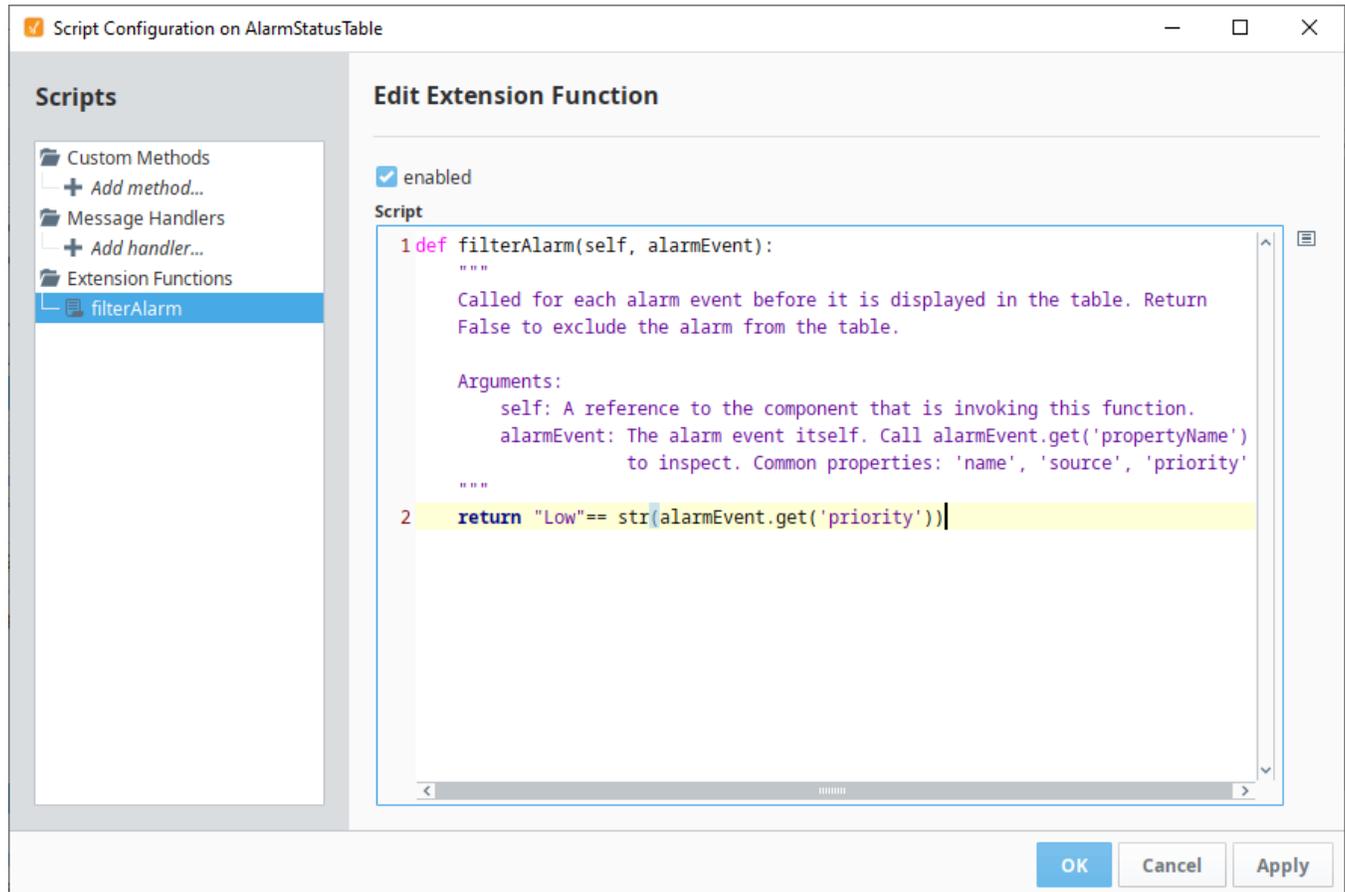
With the built-in `PyAlarmEvent` object all [alarm event properties](#) are accessible to this function, and can be used to help determine if any given event should appear on the table. Furthermore, Associated Data (also known as custom alarm properties) can be examined from the same event.

```
# Replace Property Name below with a the name of the property you wish to filter on.  
if alarmEvent.get('Property Name'):
```

```
    return True
# It's important that you return a False value for the events you don't want to see in the table.
else:
    return False
```

You could also condense the code example above by using something like the following:

```
return "Low"== str(alarmEvent.get('priority'))
```



filterShelvedAlarm

The following feature is new in Ignition version **8.1.10**
[Click here](#) to check out the other new features

- Description

Called for each event before it is displayed in the shelved tab of the table, allowing you to hide or show each alarm event (row) in the table. Return False to exclude a shelved alarm event from the table.

- Parameters

[ComponentModelScriptWrapper.SafetyWrapper](#) self- A reference to the component that is invoking this function.

[PyAlarmEvent](#) shelvedAlarmEvent - The shelved alarm event. Call `shelvedAlarmEvent.get('propertyName')` to inspect properties on the event. Properties: 'sourcePath', 'shelvedBy', 'expires'.

- Return

[Boolean](#) - The function must return either a True or False for every alarm event in the table. True will show the alarm. False will hide the alarm.

Examples

Unlike the alarmEvent object, the shelvedAlarmEvent object may only inspect shelved alarm event properties.

```
# display only shelved alarms from a specific sourcePath:  
  
source = shelvedAlarmEvent.get('sourcePath')  
if source == 'prov:default:/tag:myTag/Mode:/alm:myAlarm':  
    return True  
return False
```

Perspective - Audio



Component Palette Icon:



On this page ...

- [Properties](#)
- [Scripting](#)



The UI for this component is hidden by default and is browser dependent. There is a "hidden" property (`props.display`) which can be used to display UI for this component. In the event a user is displaying the UI (`props.display = true`), the component will be displayed differently based on the browser in use.

The following feature is new in Ignition version **8.1.16**
[Click here](#) to check out the other new features

An Audio component, hidden by default, that designers can use to play and pause sound clips in the browser.

Supported Audio Files

Supported audio file types are browser dependent.

File Type	Browser Compatibility
3GP	Firefox for Android
ADTS	Firefox Available only if available on the underlying operating system's media framework.
FLAC	Chrome 56, Edge 16, Firefox 51, Safari 11
MPEG-4 (MP4)	Chrome 3, Edge 12, Firefox, Internet Explorer 9, Opera 24, Safari 3.1
Ogg	Chrome 3, Firefox 3.5, Edge 17 (desktop only), Internet Explorer 9, Opera 10.50 Edge requires Web Media Extensions to be installed.
QuickTime (MOV)	Only older versions of Safari, plus other browsers that supported Apple's QuickTime plugin
Waveform Audiofile (WAV)	Chrome 8+, Edge 12+, Firefox 4+, Opera 11.5+, Safari 4+
WebM	Chrome 6, Edge 17 (desktop only), Firefox 4, Opera 10.6, Safari 14.1 (macOS), Safari 15 (iOS). Edge requires Web Media Extensions to be installed.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
source	The source URL of the media file.	value: string

play	The play state of the media file. Toggling this property will start or pause the media file.	value: boolean
loop	Determines if the media file should loop after reaching the end.	value: boolean
volume	The percentage of maximum volume (from 0 to 100).	value: numeric
playbackRate	A double that represents the playback rate of the media file.	value: numeric
allowDownload	Determines whether the audio player allows downloading of the media file.	value: boolean
style	Sets a style that applies to the component. Full menu of style options is available. You can also specify a style class .	object

Scripting

See the [Perspective - Audio Scripting page](#) for the full list of scripting functions available for this component

Perspective - Audio Scripting

This page details the various scripting, component, and extension functions available for [Perspective's Audio](#) component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

onPlay

This event is fired when playback has begun.

event

- Object Path

event

- Type

Null

- Description

An empty event object.

onPause

This event is fired when playback has been paused.

event

- Object Path

event

- Type

Null

- Description

An empty event object.

onError

This event is fired when there is an error attempting to play the media file.



This component event is designed to be used in tandem with a script action. Within the script action, special properties and methods are available on the event object, which is passed to the script action as a parameter.

event.errorMessage

- Object Path

event.errorMessage

- Type

String

- Description

On this page ...

- Component Events
 - onPlay
 - onPause
 - onError
 - onEnded
 - onLoaded
 - onRateChanged
- Component Functions
 - .play()
 - .pause()
 - .replay()
- Extension Functions

Error message when attempting to play the media file.

onEnded

This event is fired when playback has ended due to reaching the end of the media.

event

- Object Path

event

- Type

Null

- Description

An empty event object.

onLoaded

This event is fired when the first frame of the media has loaded.

event

- Object Path

event

- Type

Null

- Description

An empty event object.

onRateChanged

This event is fired when the playback rate of the media has changed.

event

- Object Path

event

- Type

Null

- Description

An empty event object.

Component Functions

.play()

- Description

Plays the media file, triggering the onPlay component event.

- Parameters

None

- Return

Nothing

.pause()

- Description

Pauses the media file, triggering the onPause component event.

- Parameters

None

- Return

Nothing

.replay()

- Description

Replays the media file from the beginning.

- Parameters

None

- Return

Nothing

Extension Functions

This component does not have extension functions associated with it.

Perspective - Barcode



Value

Component Palette Icon:



On this page ...

- [Properties](#)
- [Component Events](#)
- [Examples](#)
 - [Example 1](#)
 - [Example 2](#)

The Barcode component enables you to display text as a barcode. The component supports 105 different barcode types including Code 128, QR code, EAN-8, and ISBN.

The following feature is new in Ignition version **8.1.2**
[Click here](#) to check out the other new features

The Barcode component has two pre-configured [variants](#): Code 128 and QR Code.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type																																																																
value	Value to be encoded as a barcode.	value: numeric																																																																
type	What barcode specification to use, currently there are 105 formats supported including Code 128, QR code, EAN-8, and ISBN.	value: string																																																																
	<table border="1"> <thead> <tr> <th>A-B</th> <th>C</th> <th>D</th> <th>E-G</th> </tr> </thead> <tbody> <tr> <td>auspost</td> <td>channelcode,</td> <td>daft</td> <td>ean13</td> </tr> <tr> <td>azteccode</td> <td>codablockf,</td> <td>databarexpanded</td> <td>ean13composite</td> </tr> <tr> <td>azteccodecompact</td> <td>code11</td> <td>databarexpandedcomposite</td> <td>ean14</td> </tr> <tr> <td>aztecrune</td> <td>code128</td> <td>databarexpandedstacked</td> <td>ean2</td> </tr> <tr> <td>bc412</td> <td>code16k</td> <td>databarexpandedstackedcomposite</td> <td>ean5</td> </tr> <tr> <td></td> <td>code2of5</td> <td>databarlimited</td> <td>ean8</td> </tr> <tr> <td></td> <td>code32</td> <td>databarlimitedcomposite</td> <td>ean8composite</td> </tr> <tr> <td></td> <td>code39</td> <td>databaromni</td> <td>flattermarken</td> </tr> <tr> <td></td> <td>code39ext</td> <td>databaromnicomposite</td> <td>gs1-128</td> </tr> <tr> <td></td> <td>code49</td> <td>databarstacked</td> <td>GS1-128</td> </tr> <tr> <td></td> <td>code93</td> <td>databarstackedcomposite</td> <td>gs1-128composite</td> </tr> <tr> <td></td> <td>code93ext</td> <td>databarstackedomni</td> <td>gs1-cc</td> </tr> <tr> <td></td> <td>codeone</td> <td>databarstackedomnicomposite</td> <td>gs1datamatrix</td> </tr> <tr> <td></td> <td>coop2of5</td> <td>databartruncated</td> <td>gs1datamatrixrectangular</td> </tr> <tr> <td></td> <td></td> <td>databaretruncstedcomposite</td> <td>gs1northamericancoupon</td> </tr> </tbody> </table>	A-B	C	D	E-G	auspost	channelcode,	daft	ean13	azteccode	codablockf,	databarexpanded	ean13composite	azteccodecompact	code11	databarexpandedcomposite	ean14	aztecrune	code128	databarexpandedstacked	ean2	bc412	code16k	databarexpandedstackedcomposite	ean5		code2of5	databarlimited	ean8		code32	databarlimitedcomposite	ean8composite		code39	databaromni	flattermarken		code39ext	databaromnicomposite	gs1-128		code49	databarstacked	GS1-128		code93	databarstackedcomposite	gs1-128composite		code93ext	databarstackedomni	gs1-cc		codeone	databarstackedomnicomposite	gs1datamatrix		coop2of5	databartruncated	gs1datamatrixrectangular			databaretruncstedcomposite	gs1northamericancoupon	
A-B	C	D	E-G																																																															
auspost	channelcode,	daft	ean13																																																															
azteccode	codablockf,	databarexpanded	ean13composite																																																															
azteccodecompact	code11	databarexpandedcomposite	ean14																																																															
aztecrune	code128	databarexpandedstacked	ean2																																																															
bc412	code16k	databarexpandedstackedcomposite	ean5																																																															
	code2of5	databarlimited	ean8																																																															
	code32	databarlimitedcomposite	ean8composite																																																															
	code39	databaromni	flattermarken																																																															
	code39ext	databaromnicomposite	gs1-128																																																															
	code49	databarstacked	GS1-128																																																															
	code93	databarstackedcomposite	gs1-128composite																																																															
	code93ext	databarstackedomni	gs1-cc																																																															
	codeone	databarstackedomnicomposite	gs1datamatrix																																																															
	coop2of5	databartruncated	gs1datamatrixrectangular																																																															
		databaretruncstedcomposite	gs1northamericancoupon																																																															

			datalogic2of5 datamatrix datamatrixrectangular datamatrixrectangularextension dotcode	gs1qrbarcode
	H-J	K-P	Q-Z	
	hanxin	kix	qrbarcode	
	hibcazteccode	leitcode	rationalizedCodabar	
	hibccodeablockf	mailmark	raw	
	hibccode128	matrix2of5	royalmail	
	hibccode39	maxicode	sscc18	
	hibcdatamatrix	micropdf417	symbol	
	hibcdatamatrixrectangular	microqrbarcode	telepen	
	hibcmicropdf417	msi	telepennumeric	
	hibcpdf417	onecode	ultracode	
	hibcqrbarcode	pdf417	upca	
	iata2of5	pdf417compact	upcacomposite	
	identcode	pharmacode	upce	
	industrial2of5	pharmacode2	upcecomposite	
	interleaved2of5	planet		
	isbn	plessey		
	ismn	posicode		
	issn	postnet		
	itf14	pzn		
	japanpost			
displayValue	If true, the barcode's value will be displayed as text.			value: boolean
valuePosition	If displayValue is true, this property determines where the value should be displayed. Options are top or bottom; default is bottom.			value: string
valueStyle	Sets a style for the display value for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .			object
errorStyle	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .			object
style	Sets an overall style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .			object

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menu bar or by right clicking on the component.

Examples

Example 1



Property	Value
props.value	014113910613
props.type	upca
props.displayValue	true
props.valuePosition	bottom
props.valueStyle.fontFamily	Verdana
props.valueStyle.fontSize	18px

Example 2

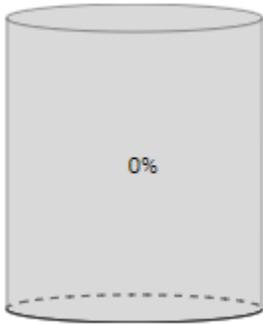


Property	Value
props.value	http://inductiveautomation.com
props.type	qrcode
props.displayValue	true
props.valuePosition	top
props.valueStyle.color	#2747C7
props.valueStyle.fontFamily	sans-serif
props.valueStyle.fontSize	14px
props.valueStyle.fontWeight	bold
props.style.paddingTop	12px

props.style.borderColor

D97700

Perspective - Cylindrical Tank



On this page ...

- [Properties](#)
- [Component Events](#)
- [Example](#)

Component Palette Icon:



A component that looks like a 3D cylindrical tank with some liquid inside. Component can be configured so that the "liquid" rises and falls as the 'value' property changes. Full menu of [style options](#) is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a [style class](#).

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type						
value	Numeric value of the tank's level.	value: numer						
capacity	Total capacity of the tank. Default is 100.	value: numer						
liquidColor	Color used to render the filled part of the tank. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color						
tankColor	Color of the non-filled tank section. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color						
liquidOpacity	The opacity of the liquid in the tank. 0 is fully transparent, 1 is fully opaque. Default is 0.5.	value: numer						
liquidWarningColor	The warning color of the liquid in the tank. See Color Selector .	color						
tankWarningColor	The tank warning color. See Color Selector .	color						
warningThreshold	The warning appearance will be used when value as a percentage of the capacity exceeds this value. Default is 100	value: numer						
strokeWidth	The stroke width, in pixels, for the outside of the tank. Default is 1.	value: numer						
valueDisplay	Value display configuration. Renders and styles a value overlay in the tank.	object						
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type				
Name	Description	Property Type						

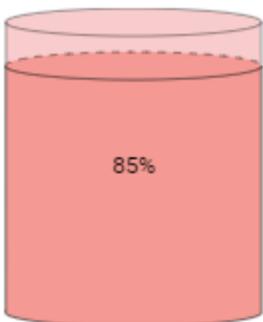
enabled	Whether valueDisplay is shown. Default is true.	value: boolean												
style	Modify the valueDisplay style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object												
format	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.24 Click here to check out the other new features</p> </div> <p>Format to apply to value which is then used as the display value. Available options include:</p> <ul style="list-style-type: none"> • None • Integer • Percent • Currency 	value: string												
unit	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.24 Click here to check out the other new features</p> </div> <p>Unit value to display on value overlay.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Name</th> <th style="text-align: left;">Description</th> <th style="text-align: left;">Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>If true, will show either a prefixed or postfixed unit.</td> <td>value: boolean</td> </tr> <tr> <td>value</td> <td>Unit value to display</td> <td>value: string</td> </tr> <tr> <td>fix</td> <td>Direction in which to place the unit. Either as a prefix or a postfix.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	If true, will show either a prefixed or postfixed unit.	value: boolean	value	Unit value to display	value: string	fix	Direction in which to place the unit. Either as a prefix or a postfix.	value: string	object
Name	Description	Property Type												
enabled	If true, will show either a prefixed or postfixed unit.	value: boolean												
value	Unit value to display	value: string												
fix	Direction in which to place the unit. Either as a prefix or a postfix.	value: string												
style	Sets a style for this cylindrical tank. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object												

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

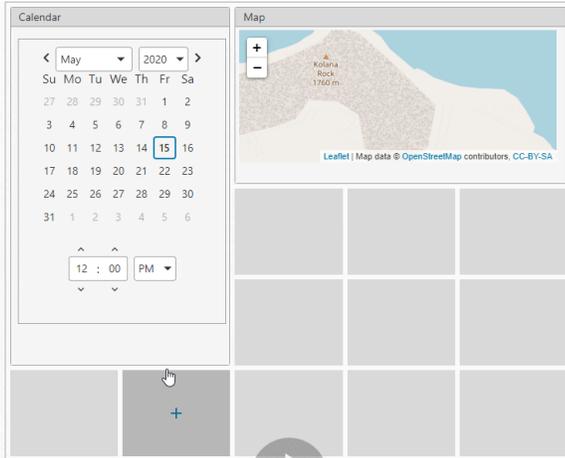
Example



Property	Value
----------	-------

props.value	85
props.capacity	100
props.warningThreshold	80

Perspective - Dashboard



On this page ...

- [Properties](#)
- [Component Events](#)
- [Examples](#)
 - [Demonstration](#)

Component Palette Icon:



The Dashboard component exposes layout capabilities to end users in a Perspective session so they have the ability to customize their dashboard layout for their individual needs. Widgets are configured in the Designer by designers and made available to Perspective session users. The Dashboard component uses a grid system based off of CSS grid specifications to position and place widgets. The Property Editor of the Dashboard component is where the designer controls the general layout of the grid by specifying the responsive mode: fixed or stretch, if the dashboard is editable, and if each widget is configurable and available in a Perspective session.

End users can choose from a list of pre-configured widgets to configure their dashboards in a Perspective session. They can add, remove, resize, move around, and configure widgets, including the ability to interact with widgets in a session such as entering text in a text field, displaying/hiding components in a widget, and even use parameters to pass a property to a specified view.

To learn more, refer to [Configuring a Dashboard](#).

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	FT									
pack	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.4 Click here to check out the other new features</p> </div> <p>Enables widget packing algorithm. When disabled, widgets can be placed anywhere on the Dashboard and the component will not try to rearrange them in an optimal layout.</p>	bc									
grid	<p>The grid layout mode defines the responsive behavior of the grid and its cells: fixed and stretch.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>fixed</td> <td>In fixed mode, the grid's dimensions can be greater or less than the full dimensions of its containing element, and its cells are given a static size, effectively creating a scrollable grid when cells overflow beyond the containing elements dimensions.</td> <td>value: string</td> </tr> <tr> <td>stretch</td> <td>In stretch mode, the grid's dimensions are restricted to the full dimensions of its containing element, and its cells consume one free unit of space, effectively growing and shrinking with the containing element.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	fixed	In fixed mode, the grid's dimensions can be greater or less than the full dimensions of its containing element, and its cells are given a static size, effectively creating a scrollable grid when cells overflow beyond the containing elements dimensions.	value: string	stretch	In stretch mode, the grid's dimensions are restricted to the full dimensions of its containing element, and its cells consume one free unit of space, effectively growing and shrinking with the containing element.	value: string	ot
Name	Description	Property Type									
fixed	In fixed mode, the grid's dimensions can be greater or less than the full dimensions of its containing element, and its cells are given a static size, effectively creating a scrollable grid when cells overflow beyond the containing elements dimensions.	value: string									
stretch	In stretch mode, the grid's dimensions are restricted to the full dimensions of its containing element, and its cells consume one free unit of space, effectively growing and shrinking with the containing element.	value: string									
isEditing	Controls the runtime edit mode of the dashboard component. Stays in sync with the edit/play toggle control located at the bottom of the	bc									

component.

editingToggle

Whether to display the dashboard editing toggle option. When disabled, hides the built in edit/play toggle control located at the bottom of the component. Disable this if you'd like to implement your own toggle that updates the `isEditing` prop in a controlled fashion. Default is true.

boolean

fixed

Visible when the grid mode is `fixed`.

boolean

Name	Description	Property Type
cellSize	Width and height of a grid cell. Exclusively for fixed mode.	numeric
rowCount	The number of rows in the grid.	numeric
columnCount	The number of columns in the grid.	numeric
rowGutterSize	The gap size between grid rows.	numeric
columnGutterSize	The gap size between grid columns.	numeric

stretch

Visible when the grid mode is `stretch`.

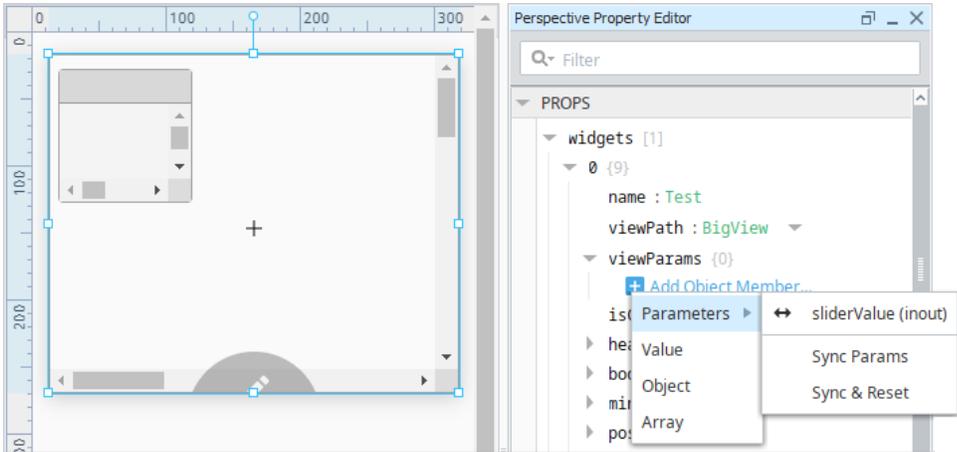
boolean

Name	Description	Property Type
rowCount	The number of rows in the grid.	numeric
columnCount	The number of columns in the grid.	numeric
rowGutterSize	The gap size between grid rows.	numeric
columnGutterSize	The gap size between grid columns.	numeric

widgets

An array of configuration objects for widgets currently in use the dashboard display.

array

Name	Description	Property Type
name	The unique widget name.	value: string
viewPath	The current configuration view path of the widget.	value: string
viewParams	Parameters being passed to the view. <div style="border: 1px solid orange; padding: 5px; margin: 10px 0;">The following feature is new in Ignition version 8.1.4 Click here to check out the other new features</div> <p>As of 8.1.4 a dropdown list of parameters is available when the user clicks the Add Object Member  icon. This makes it easy to add parameters from the rendered view.</p> 	object
isConfigurable	Whether this widget is configurable during runtime. If enabled, dashboard is in edit mode, the toggle becomes available when the widget is selected which is used to configure the widgets view. When toggled on, the configuring view parameter will be true.	value: boolean
header	Configuration object for the widget header.	object

Name	Description	Property Type
enabled	When enabled, renders the widget header.	value: boolean
title	The header title to display.	value: string
style	Style to be applied to the widget header. Full menu of style options is available. You can also specify a style class .	object

body	Configuration object for the widget body.	object
------	---	--------

Name	Description	Property Type
style	Style to be applied the widget body. Full menu of style options is available. You can also specify a style class .	object

minSize	Specifies the widgets minimum allowable size when determining widget layout. Users may not resize widgets below these dimensions.	object
---------	---	--------

Name	Description	Property Type
columnSpan	The minimum allowable columns that this widget may span.	value: numeric
rowSpan	The minimum allowable rows that this widget may span.	value: numeric

position	The widget position in the dashboard. Whenever a widget is added, resized, or moved the widget position object is automatically updated.	object
----------	--	--------

Name	Description	Property Type
rowStart	The top position of the widget.	value: numeric
rowEnd	The bottom position of the widget.	value: numeric
columnStart	The left position of the widget.	value: numeric
columnEnd	The right position of the widget.	value: numeric

style	Style to be applied the widget. Full menu of style options is available. You can also specify a style class .	object
-------	---	--------

availableWidgets An array of widgets as configuration objects that are available to the user. When a widget is added to the dashboard via the add widget modal, this configuration object is copied to the widgets in use array, and act as the widgets defaults. ar

Name	Description	Property Type
viewPath	The current configuration view path of the widget.	string
viewParams	Parameters being passed to the view at the specified path. <div style="border: 1px solid orange; padding: 5px; margin: 5px 0;"> <p>The following feature is new in Ignition version 8.1.4 Click here to check out the other new features</p> </div> <p>As of 8.1.4 a dropdown list of parameters is available when the user clicks the Add Object Member  icon. This makes it easy to add parameters from the rendered view.</p>	object
isConfigurable	Whether this widget is configurable during runtime. If enabled and the dashboard is in edit mode, the toggle becomes available when the widget is selected which is used to configure the widgets view. When toggled on, the configuring view parameter will be 'true.'	value: boolean
defaultSize	Specifies the widgets default size adding a widget with no size specified.	object

Name	Description	Property Type
columnSpan	The default columns that this widget will span.	value: numeric

	rowSpan	The default rows that this widget will span.	value: numeric	
minSize	Specifies the widgets minimum size used when determining widget layout.			object
	Name	Description	Property Type	
	columnSpan	The minimum allowable columns that this widget may span.	value: numeric	
	rowSpan	The minimum allowable rows that this widget may span.	value: numeric	
category	A category in which to group this widget when displayed in the add widgets modal.			value: string
name	A unique name to provide this widget. This is used in the add widget modal. If no name is specified, its value will be blank. This is a required property.			value: string
header	Widget header configuration.			object
	Name	Description	Property Type	
	enabled	Whether the widget header should show.	value: boolean	
	title	The header title to display.	value: string	
	style	Style to be applied the widget. Full menu of style options is available. You can also specify a style class .	object	
body	Widget body configuration.			object
	Name	Description	Property Type	
	style	Style to be applied the widget. Full menu of style options is available. You can also specify a style class .	object	
style	Style to be applied to the widget. Full menu of style options is available. You can also specify a style class .			object

Component Events

Perspective Component Events

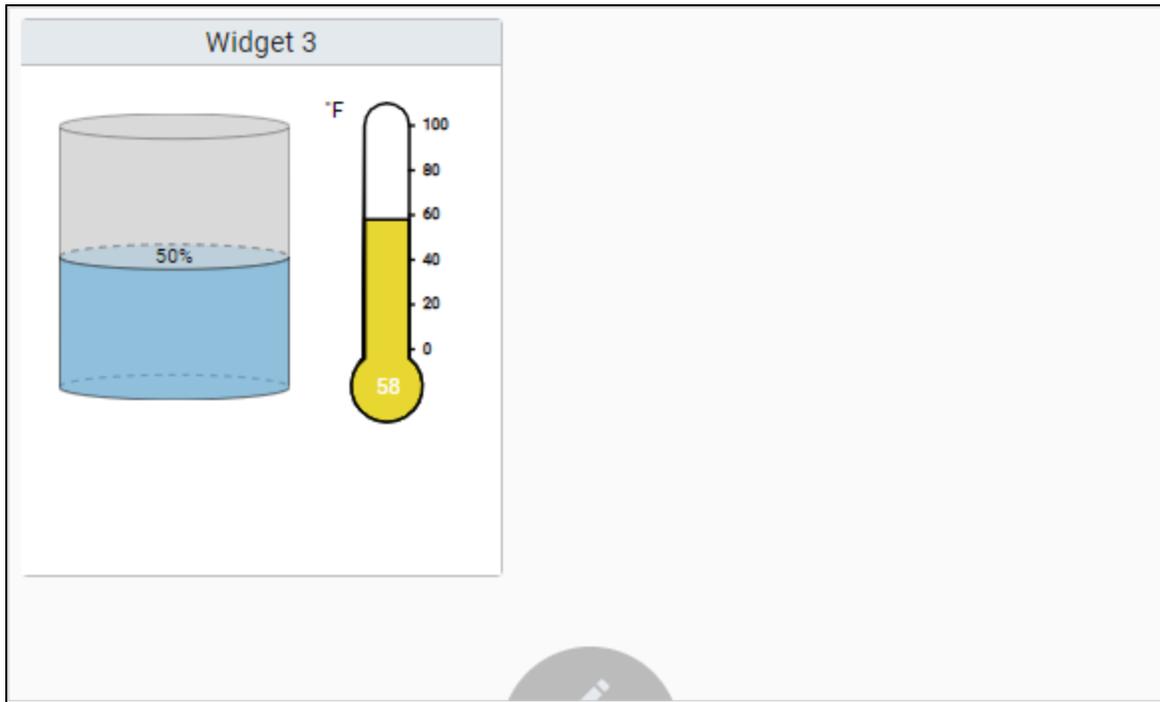
The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Examples

In addition to the demonstration below, learn more about the Dashboard component on the [Configuring a Dashboard](#) page.

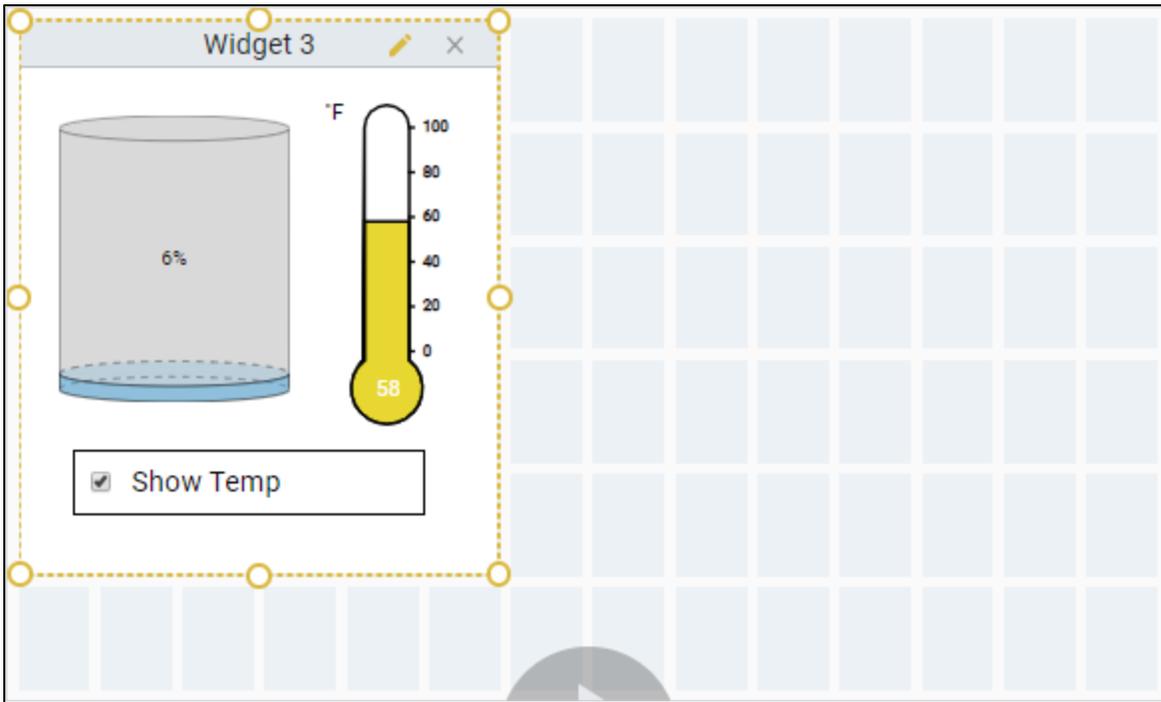
Demonstration

This image shows the dashboard in a Perspective Session with one widget.

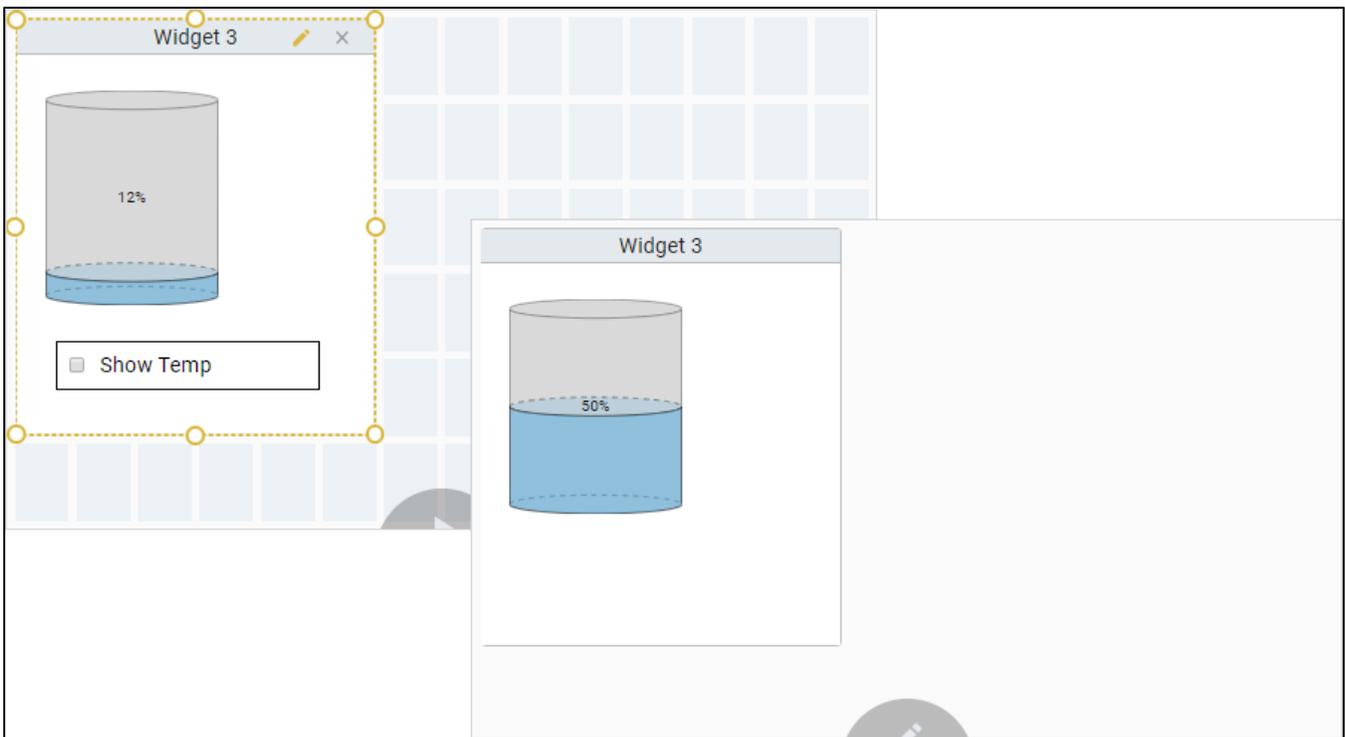


Property	Value
props.grid	stretch
props.isEditing	true
props.editingToggle	true
props.stretch.rowCount	6
props.stretch.columnCount	14
props.widgets.0.name	Widget 3
props.widgets.0.viewPath	Widget 3
props.widgets.0.isConfigurable	true
props.widgets.0.header.enabled	true
props.widgets.0.header.title	Widget 3
props.availableWidgets.2.viewPath	Widget 3
props.availableWidgets.2.isConfigurable	true
props.availableWidgets.2.name	Widget
props.availableWidgets.2.header.enabled	true
props.availableWidgets.2.header.title	Widget 3

This same example also allows the user to interact with the widget in Edit mode when the 'isConfigurable' property is set to 'true.' The view uses a 'configuring' parameter to go into 'configuring' mode allowing users to configure the widget. It allows the user to show/hide the Thermometer showing the temperature of the Tank by simply checking/unchecking the ShowTemp checkbox.



By unchecking Show Temp, the Thermometer component is removed from the widget on the dashboard in a Perspective Session.



Configuring a Dashboard

The Dashboard exposes widgets to end users in a [Perspective Session](#) so they can customize their dashboard layout for their individual needs. Widgets are views that are pre-configured in the Designer and made available to Perspective Session users. End users have the flexibility to add, remove, resize, move around, and even configure widgets in the dashboard of their Perspective Session without having access to the Designer. Users can interact with widgets in a session on both desktop and mobile devices. There may be some minor variances in how a user can interact with their dashboard between desktop and mobile devices, but the principle is still the same.

Configuring a Dashboard Component

Configuring a Dashboard starts with designing widgets and having a selection of pre-configured widgets for users to choose from to configure their individual dashboards. Designers create the widgets and make them available for end users to use in their individual dashboards. By making the widgets available using the 'availableWidgets' property, the widget overlay modal is populated with a searchable list of all the available widgets a user can add to their dashboard. The dashboard component contains a host of additional properties that can be configured based on the end-user requirements.

The [Dashboard component](#) uses a grid system based off of CSS grid specifications to position and place your widgets. The Property Editor settings of the Dashboard component control the general layout of the grid. They specify the responsive mode: fixed or stretch, if the dashboard is editable, and if each widget is configurable and available in a Perspective session. The image below shows one widget on a dashboard in the Designer along with some of its properties.

To learn more about Dashboard properties, refer to the [Dashboard component](#) page.

On this page ...

- [Configuring a Dashboard Component](#)
- [Setting Up a User Dashboard](#)
 - [Adding a Widget](#)
 - [Removing a Widget](#)
 - [Moving a Widget](#)
 - [Resizing a Widget](#)
- [Configuring a Widget](#)
 - [Setting a Widget as Configurable in the Designer](#)
 - [Creating a Configurable View in the Designer](#)
- [Saving Perspective Session Edits and Populating Widgets](#)
- [Saving and Loading Dashboard Component JSON Data](#)

The screenshot displays the Perspective Designer interface. On the left, a dashboard titled 'Dashboard 2' contains a widget named 'Widget 3'. The widget is a container with a grid layout. It features a cylinder representing a tank with '21%' of liquid inside, and a thermometer showing a temperature of '58' degrees Fahrenheit. The background of the widget is a light blue grid. On the right, the 'Perspective Property Editor' is open, showing the configuration for 'Widget 3'. The 'PROPS' section is expanded, showing the following properties:

- `grid`: stretch
- `isEditing`: true
- `editingToggle`: true
- `stretch` (array of 4):
 - `rowCount`: 8
 - `columnCount`: 8
 - `rowGutterSize`: 6
 - `columnGutterSize`: 6
- `widgets` (array of 1):
 - `0` (object):
 - `name`: Widget 3
 - `viewPath`: Widget 3
 - `viewParams`: {1}
 - `isConfigurable`: true
 - `header`: {3}
 - `body`: {1}
 - `minSize`: {2}
 - `position`: {4}
 - `style`: {1}
 - `availableWidgets`: {2}

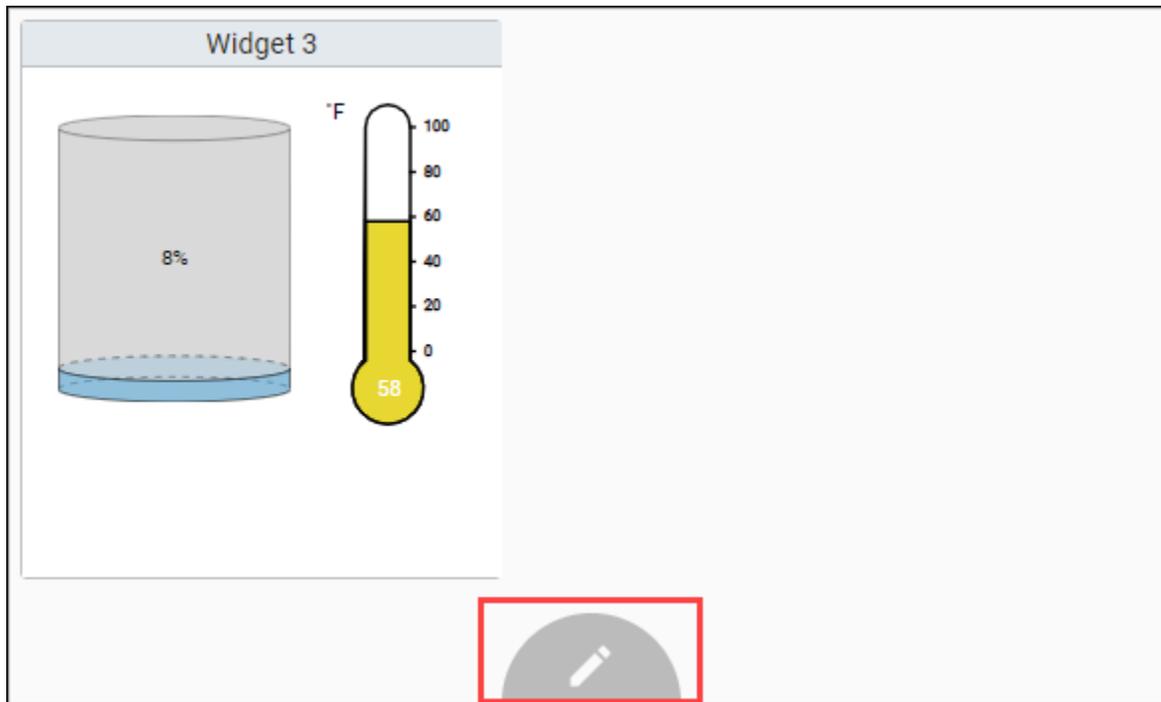
At the bottom of the Property Editor, there are sections for 'POSITION', 'CUSTOM', and 'META' properties.

Setting Up a User Dashboard

Setting up a dashboard starts with users choosing from a list of pre-configured widgets to configure their dashboards in a Perspective Session based on their individual needs.

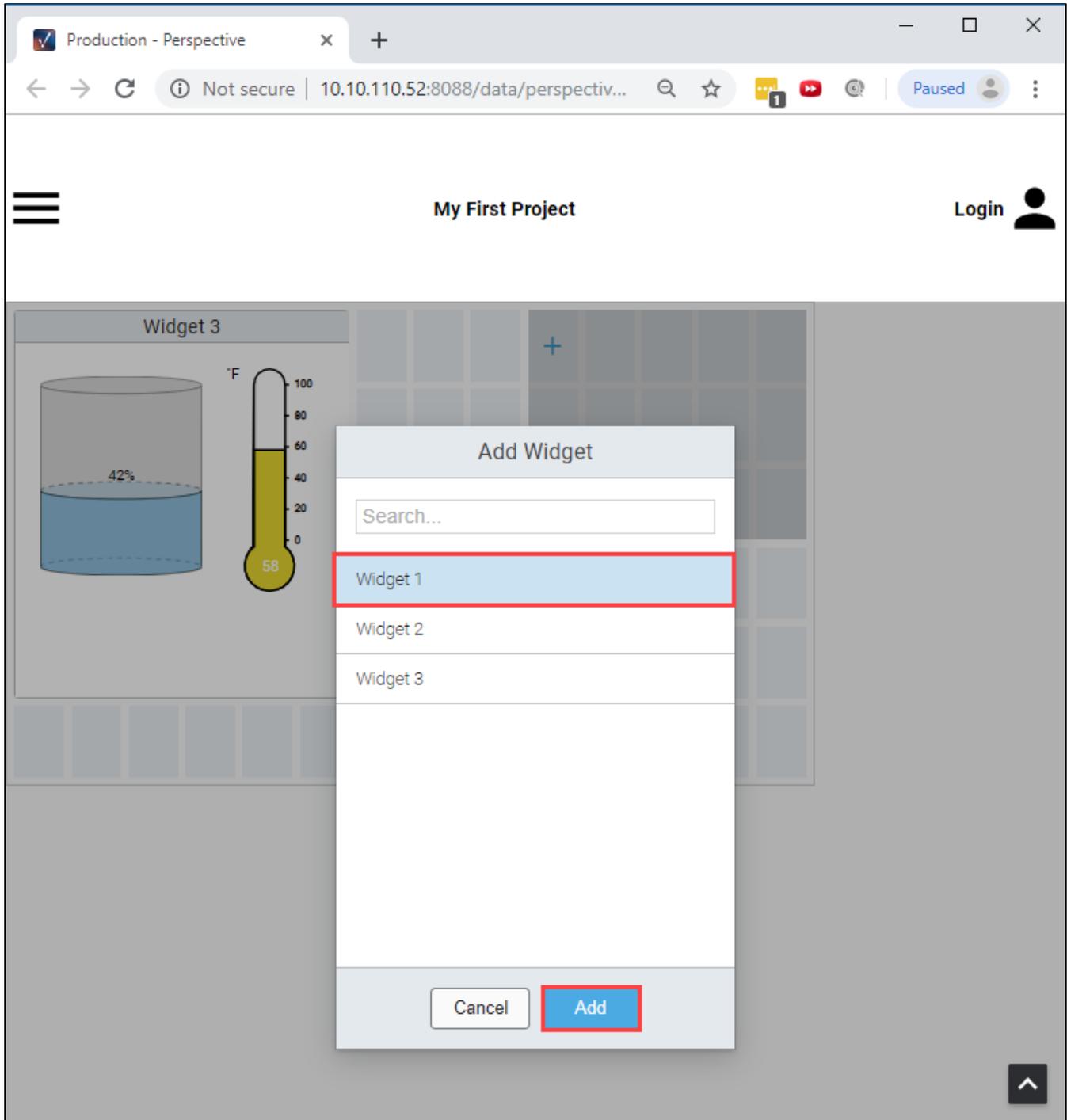
To edit the dashboard in a Perspective Session, the user can put the dashboard into Edit mode by clicking the Edit icon at the bottom of the dashboard and deciding what widgets they want, where they want them, and how they want them configured. They can add, remove, resize and configure widgets, including the ability to interact with widgets such as entering text in a text field or displaying/hiding components in a widget. You can also remove this control entirely and implement your own by configuring the 'editingToggle' property on the component. Refer to the [Dashboard component](#) properties for more details.

The following sections on this page describe how to set up your own dashboard.



Adding a Widget

There are two ways a user can add a widget in a Perspective Session: by clicking on a single grid cell, or by dragging a grid cell over multiple grid cells that opens an add widget overlay as shown in the image below. Both ways result in displaying the add widget modal which provides a searchable list of all of the available widgets a user may add to their dashboard.

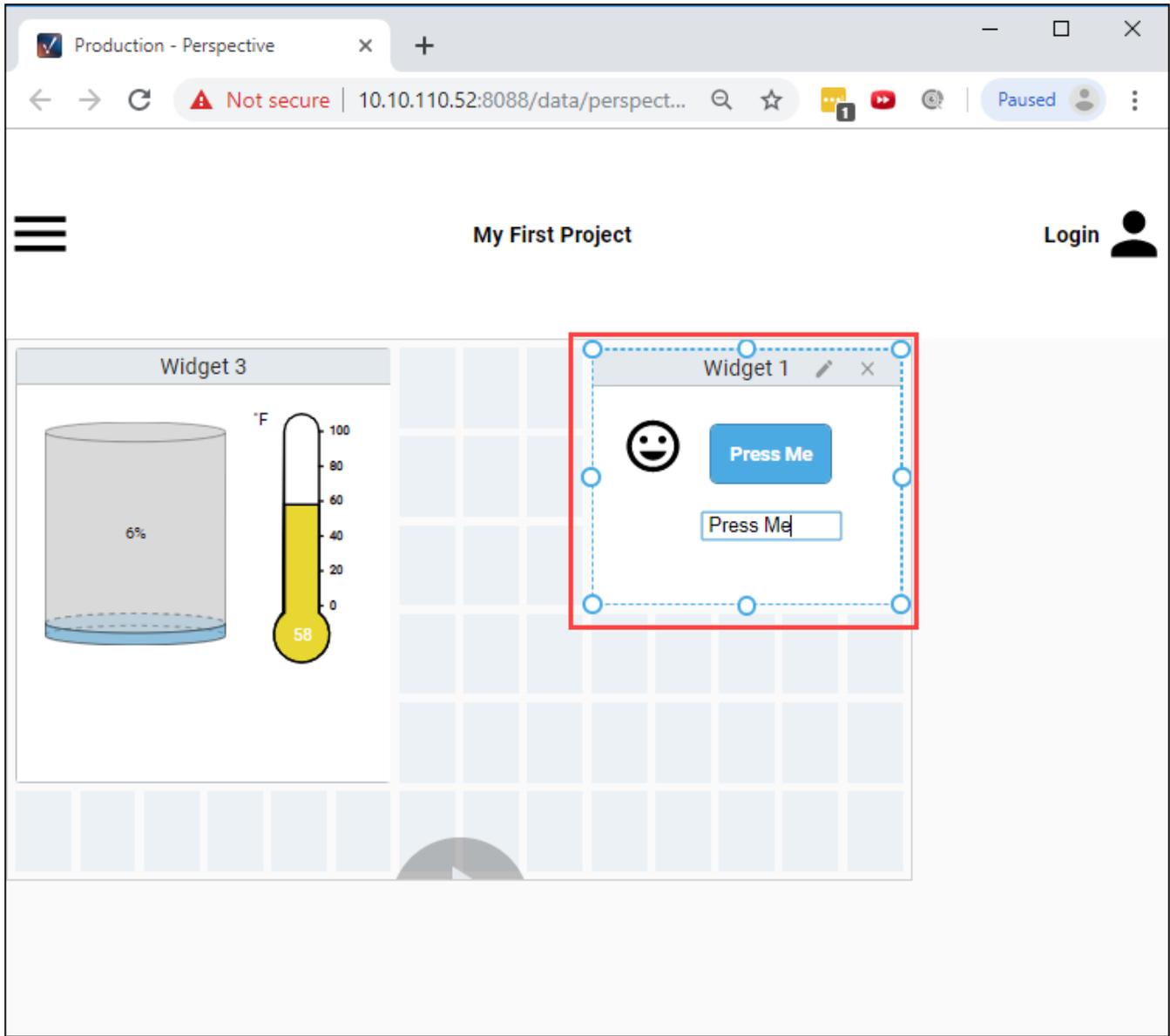


Dragging a grid cell creates an add widget overlay that specifies the desired dimensions of the widget to add. If the desired widget position overlaps other widgets, the overlapped widgets will be moved to any available space on the dashboard. Widgets do not overlap when being added, resized, and moved unless there happens to be no space for a widget so that it is placed within the grid.

Widget's Minimum Dimensions

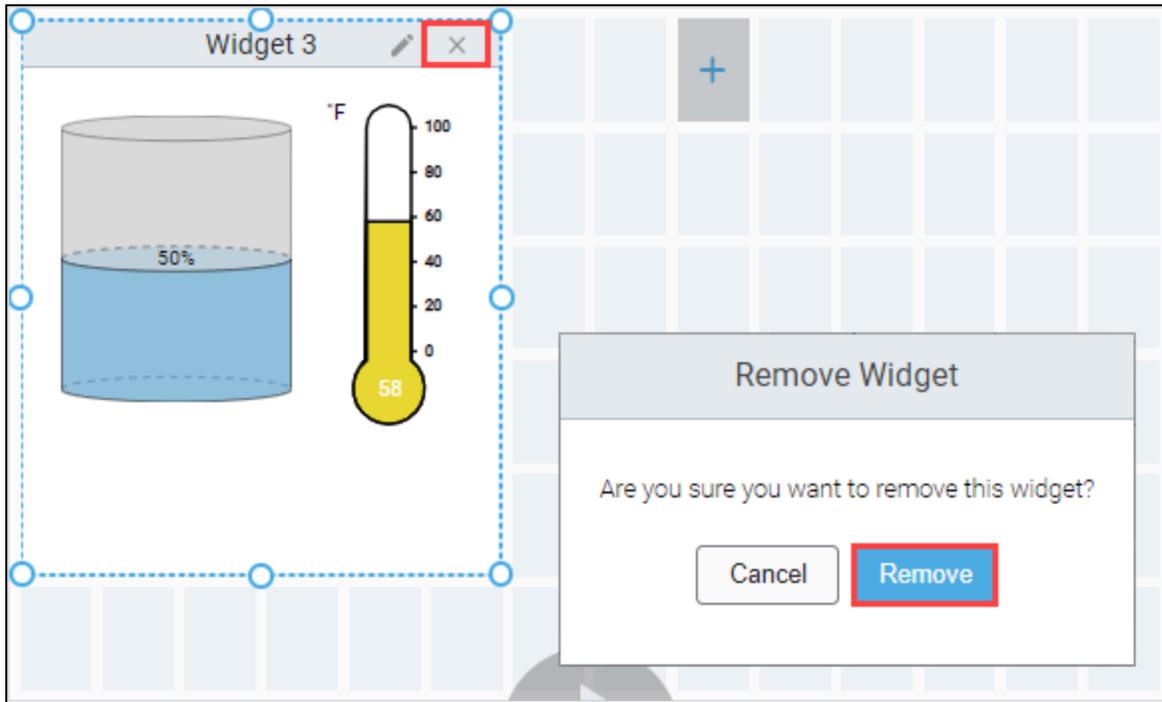
When adding a widget, if the desired dimensions are less than the configured minimum dimensions, the desired dimensions will get overridden by the minimum dimensions. If a single grid cell is clicked, the configured default dimensions will be applied, if and only if, the default meets the required minimum dimensions, otherwise the minimum dimensions are applied. By default, the minimum and default dimensions for a widget are 1x1.

On mobile devices, activating a grid cell requires a long-press of about a second. Once a grid cell is activated, you can then drag to create the add widget overlay. The image below shows Widget 1 dropped over the multiple selected grid cells in the dashboard. You'll notice the active widget has a dashed blue border.



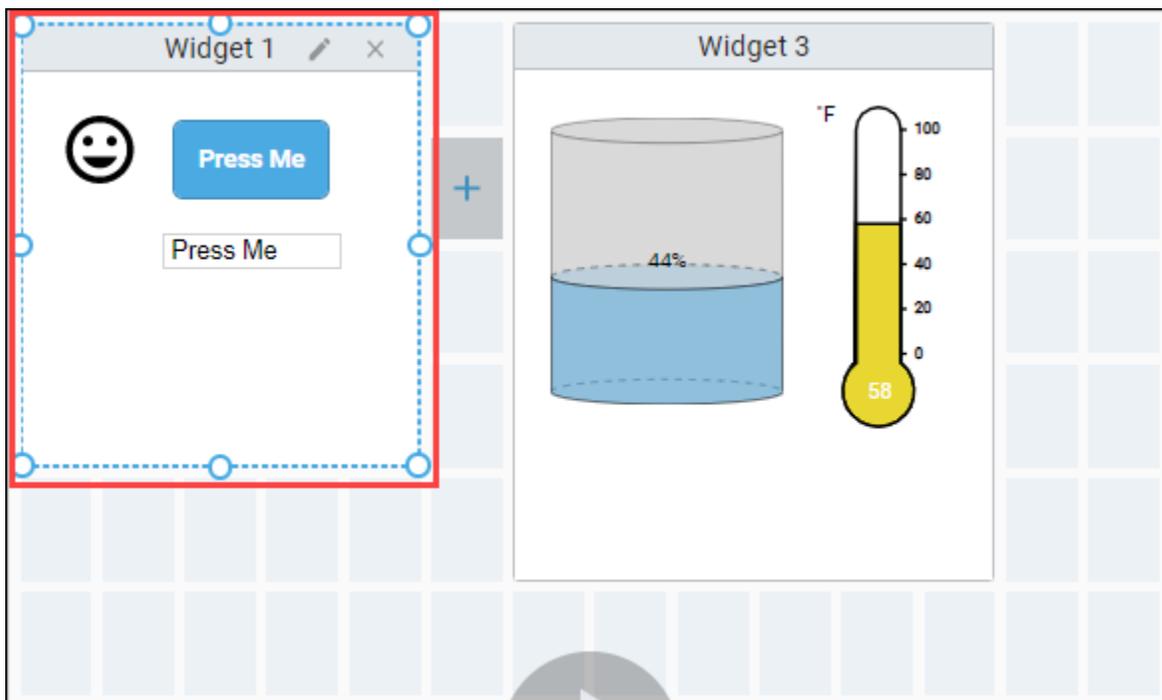
Removing a Widget

Click the **Edit** icon, select the widget, and you'll notice the widget has a dashed blue border indicating the widget is active, then click the **Delete 'x'** icon in the top right corner to remove the widget from the dashboard. You will then be prompted with a confirmation modal to delete the widget. Click **Remove**.



Moving a Widget

Put the dashboard in Edit mode, select the widget so that it becomes active (dashed blue border). Drag the widget to the desired position. As you move the widget, any overlapped widgets will be repositioned into the first available space.



Resizing a Widget

To resize a widget, put the dashboard in Edit mode, then simply select the widget you'd like to resize and drag one of the resize handles. If, while resizing, the widget overlaps other widgets, the overlapped widgets will be repositioned into the first available space.

Configuring a Widget

The dashboard allows your users to configure a widget in a Perspective Session. To do this, you need make a few changes to your view and Dashboard component configuration.

Setting a Widget as Configurable in the Designer

To make a view allow configuration, you need to set the `isConfigurable` property for each widget that needs to be configured. This will set a param value on your view (in the runtime) that you can use to create a configuration display in your view. The purpose of this parameter is to avoid having to make a separate widgets for each possible variation of the same view.

1. Select the Dashboard component.
2. Expand the `availableWidgets` parameter, and expand the array object for the widget that you want to make configurable.
3. Set the `isConfigurable` property to 'true' for this widget.

Creating a Configurable View in the Designer

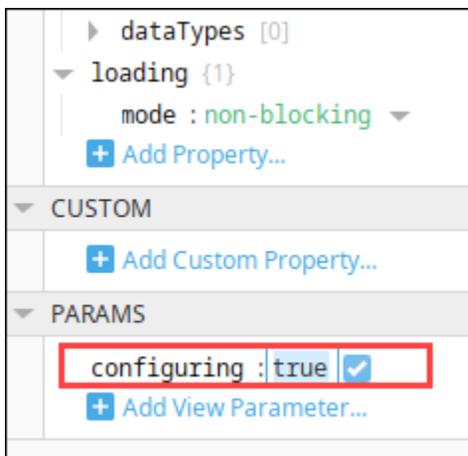
To make a view configurable, you need to do a bit of work to alter what is in the view. This is possible because the Dashboard component was created to use a parameter named `configuring` that is set to 'true' when the widget is in put into configuration mode. The idea here is to have a second 'mode' or 'display' version of the view that has controls on it to effect the primary display. The best way to do this is to create two containers in your view; one for configuration, and one for display. You can then bind the visibility on each container so only one is shown at a time.

To learn more about using parameters to pass properties, refer to the [Perspective Component Properties](#) page. You will not need to pass any value into the param though, it is done automatically for you if you get the param name correct.

1. Create a new Coordinate view. For the example, we named our `Configurable_View`.

2. **Note:** If you use a Flex container, some of the settings will be different further down in the example.

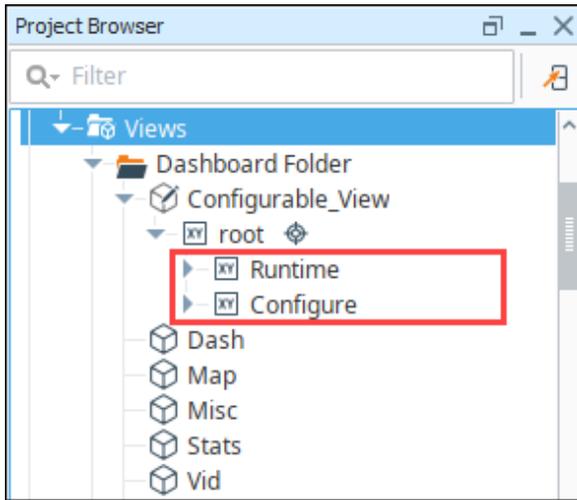
In the Property Editor, add a view param. Name the param '`configuring`' and set the value to 'true'. Note the spelling and (lack of) capitalization.



3. Drag a Coordinate Container component inside your view. Give the container a good name like 'Runtime.'
4. Add any display components you want.

Note: If you started from an existing view, move all existing components into the new container then make the container fill your entire view.

- a. Deep Select the Configure container.
 - b. Drag a Cylindrical Tank component into it.
 - c. Bind the value property to a Tag.
 - d. Drag a Temperature Gauge component into it.
 - e. Bind the value property to a Tag.
5. We need to create a second new container in the view for your configuration. **Duplicate** the Runtime container. Give the container a good name like 'Configure.' This container will be a sibling to the Runtime container, not inside the Runtime container.



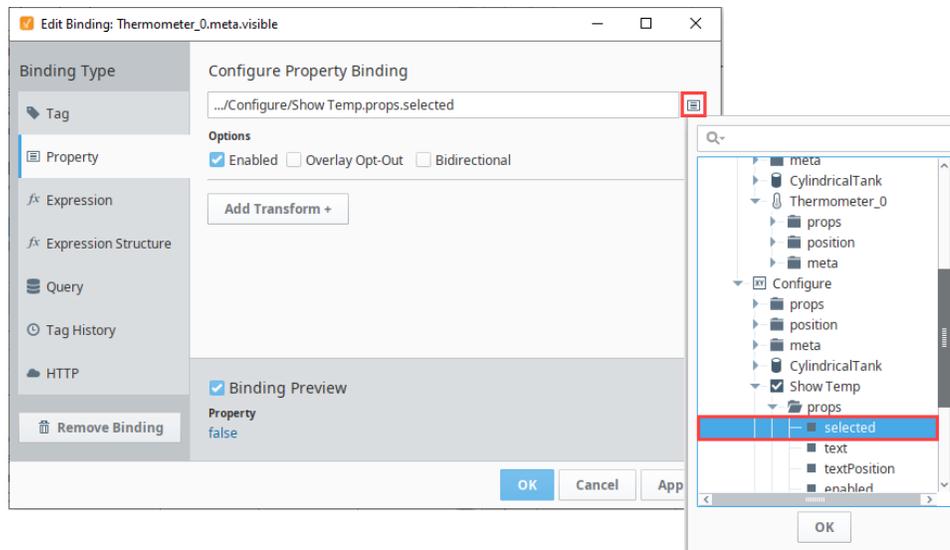
6. Add a Checkbox to toggle the temperature component visibility.

Note: If you started with an existing view, this step is completely up to you. You will decide what should be configurable and create controls for that in your configuration container. For example, you could create a list of Tags for the user to select between and display only the selected Tags on a chart.

- a. Deep Select the Configure container.
- b. Drag a **Checkbox** component into it.
- c. Set the Name and Text to "Show Temp".

7. Now we need to alter the Runtime container components to listen to our new controls.

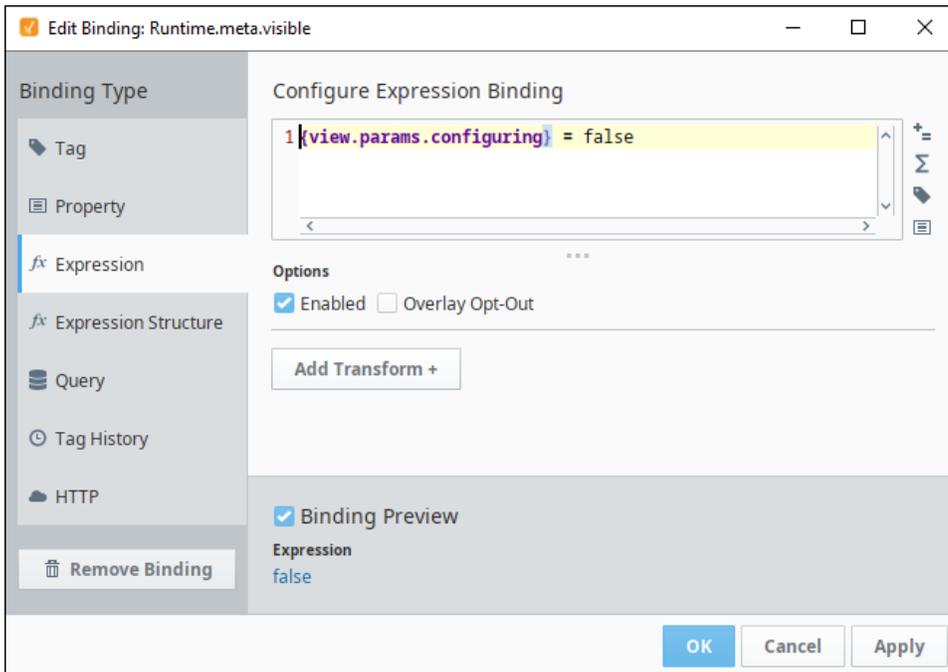
- a. Deep Select the Runtime Container.
- b. Select the Temperature Gauge component.
- c. In the Property Editor under META, bind the **Visible** property to the Selected value of the Show Temp Checkbox component.



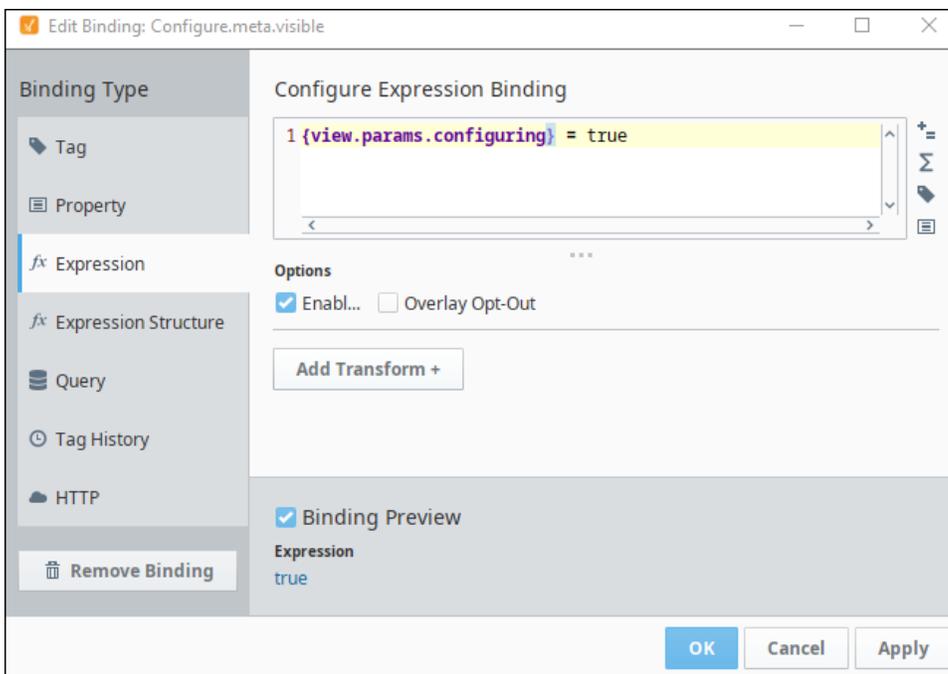
8. Now we just need to show one container at a time.

Note: If you used a Flex container at the start of this example, then use the 'display' property instead of the 'visible' property in the following steps.

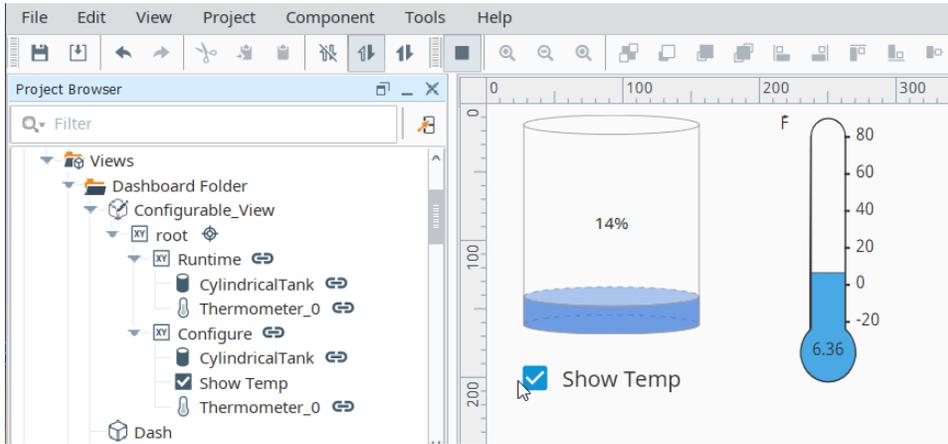
- a. Bind the 'visible' property for the Runtime container using an expression. It should be true when the **configuring** param is false.



- b. Bind the 'visible' property for the Configure container using an expression. It should be true when the `configuring` param is true.



9. Save your project and then put the Designer into Preview mode. When you click on the Temp Show button, you'll see the Temperature component appear or disappear.

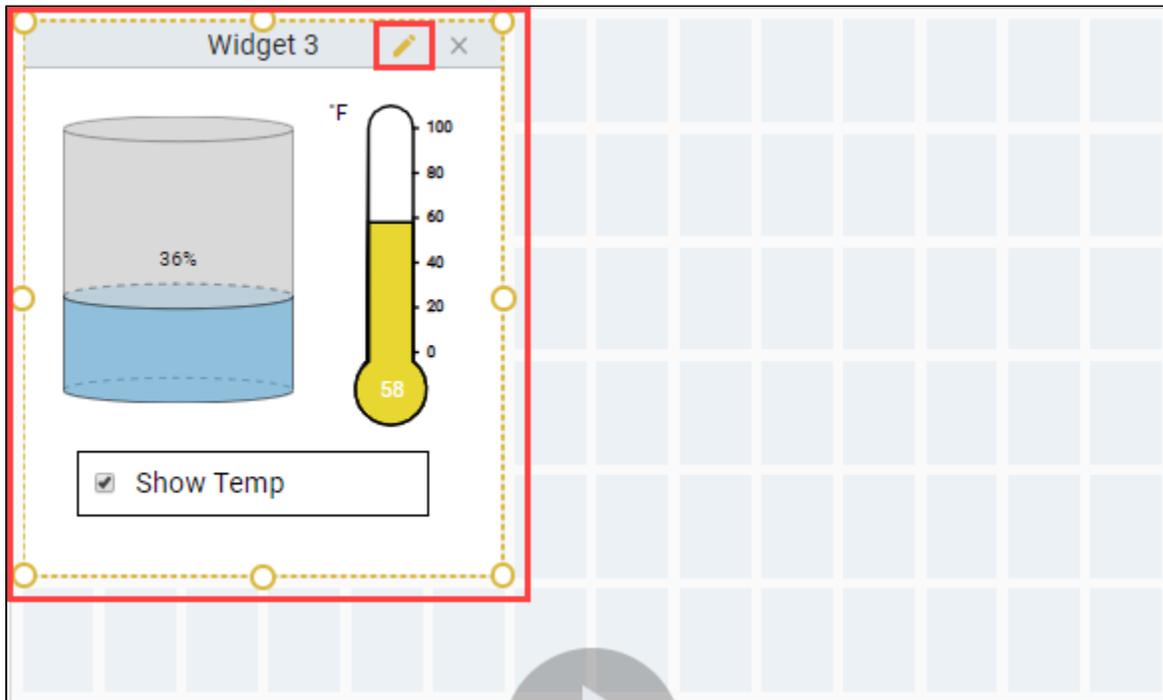


10. Now you can use this view in your dashboard.

Configuring in the Dashboard Component in the Runtime (Perspective Session)

This section is for the people using the Dashboard in a Perspective Session.

1. To use a configurable widget in the runtime, put the dashboard in Edit mode and select the widget you'd like to configure.
2. Click the edit icon (pencil) in the top right corner of the widget. The widget's border will change colors from blue to orange (shown in the image below).
3. The view changes to show the 'configuring' mode you set up previously for the view, allowing users to configure the widget.



Saving Perspective Session Edits and Populating Widgets

Edits that an end user makes in their dashboard in a Perspective Session are not automatically saved and do not persist when the end user's session restarts. A session can be refreshed within the same session. One possible solution for populating widgets for the next editing session is to add a property change script on the 'widgets' prop to listen for changes and then write that value back to a database along with any user information derived from the active session. The value of the 'widgets' prop will be an array of QualifiedValues, which you'll need to handle accordingly. In similar fashion, consider adding an 'onStartup' event action that will query the database and then populate the 'widgets' prop with the users last saved configuration and optionally populate the 'availableWidgets' prop (possibly for varying user roles).

Saving and Loading Dashboard Component JSON Data

The Dashboard Component also has the ability to save and load custom widget configurations. This functionality can be set up by configuring a database table to store widget data, and then writing scripts to save and pull JSON data. We'll set up an example using a MySQL database to demonstrate the basic format of this functionality, which can be expanded to serve many processes.

Once you have a database connection, complete the following steps to create a MySQL database table:

1. Open MySQL Workbench and create a table.
2. Enter a name for the table, in this case we used widgets.
3. Enter name under Column Name in the first row.
4. Use the dropdown to select VARCHAR(45) from the Datatype dropdown.
5. Select UQ to make sure values aren't duplicated.
6. Enter widgetjson under Column Name a second row.
7. Select MEDIUMTEXT from the Datatype dropdown.

Note: Depending on your database, it may be a different datatype name so make sure to select a datatype that can hold long strings.

The screenshot shows the MySQL Workbench Table Designer interface. The table name is 'widgets' and the schema is 'jsondata'. The engine is InnoDB. The table has two columns: 'name' (VARCHAR(45)) and 'widgetjson' (MEDIUMTEXT). The 'name' column has PK, NN, and UQ checked. The 'widgetjson' column has Default/Expression set to NULL. The 'Apply' button is highlighted.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
name	VARCHAR(45)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
widgetjson	MEDIUMTEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

8. Select Apply.
9. Select Apply and Finish on the confirmation pop-up.
10. Add data into your Workbench table query scripting:

```
INSERT INTO test.widgets VALUES ("Widget", "1234")
```

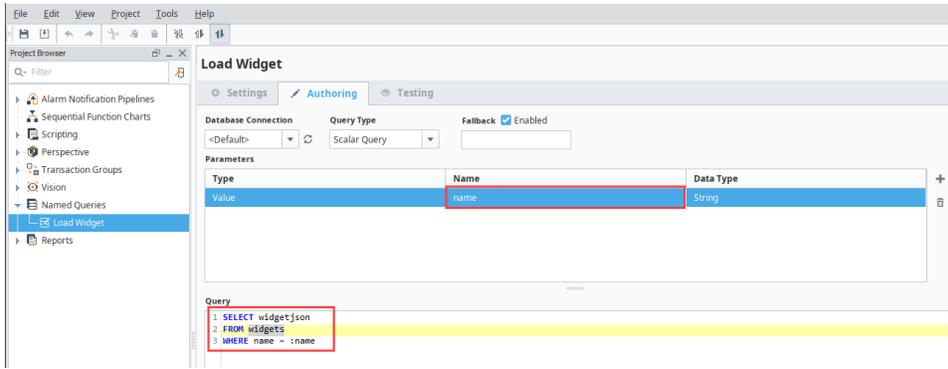
11. Select the Execute  icon to confirm the Action Output.

Next, we will create two named queries in the Designer that will be used in the save and load scripts.

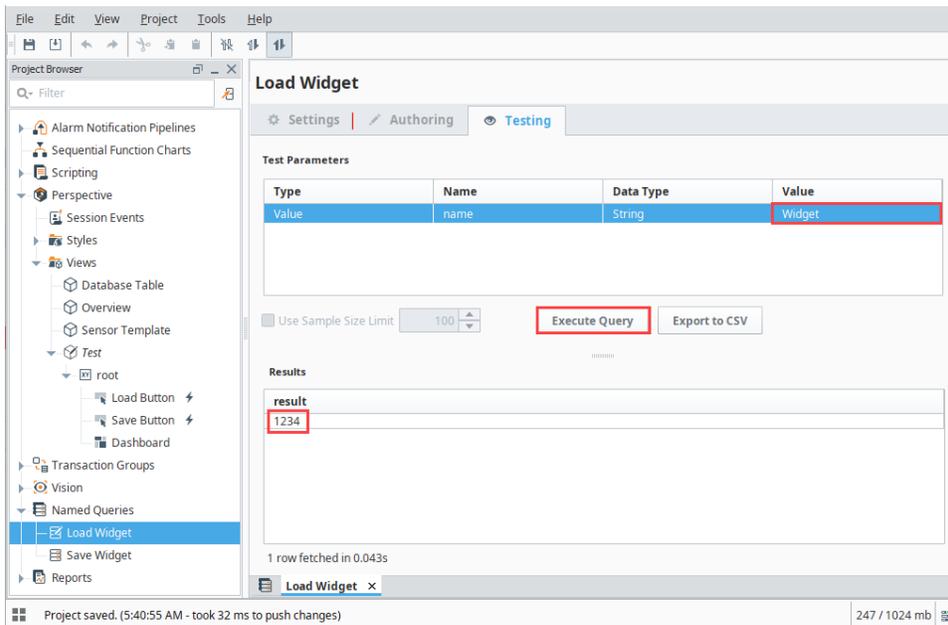
1. Right-Click Named Queries and select New Query in the Designer.
2. Enter Load in the Name field.
3. Select Create Named Query.
4. Select the Authoring Tab.
5. Fill in the Query script:

```
SELECT widgetjson  
FROM widgets  
WHERE name = :name
```

- Enter name in Name column.



- Select the Testing Tab.
- Enter a name value in the Value column, in this example, we used Widget.
- Select Execute Query to confirm the result is as expected, 1234.

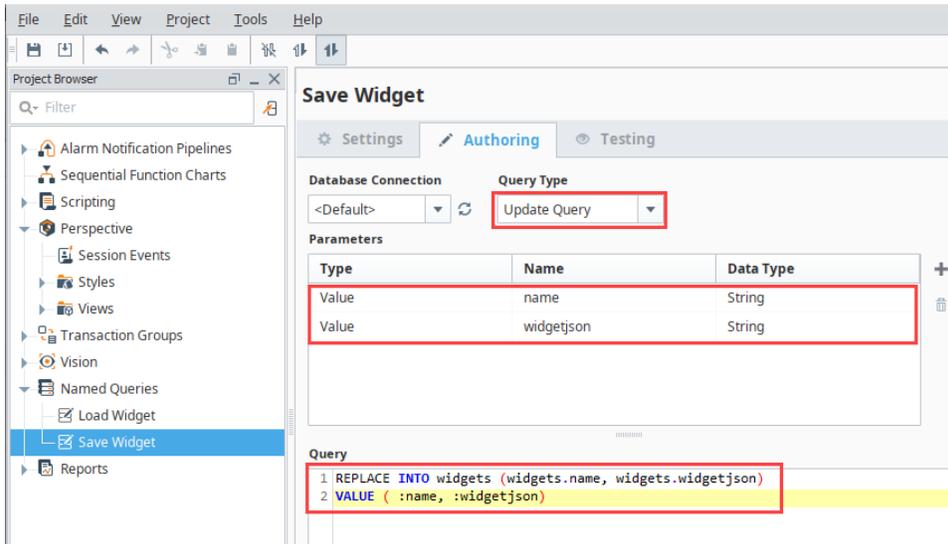


- Select the Save icon to save the Load Widget Query.
- Right-Click Named Queries and select New Query.
- Enter Save in the Name field.
- Select Create Named Query.
- Select the Authoring Tab.
- Fill in the Query script:

```

REPLACE INTO widgets (widgets.name, widgets.widgetjson)
VALUE ( :name, :widgetjson)
  
```

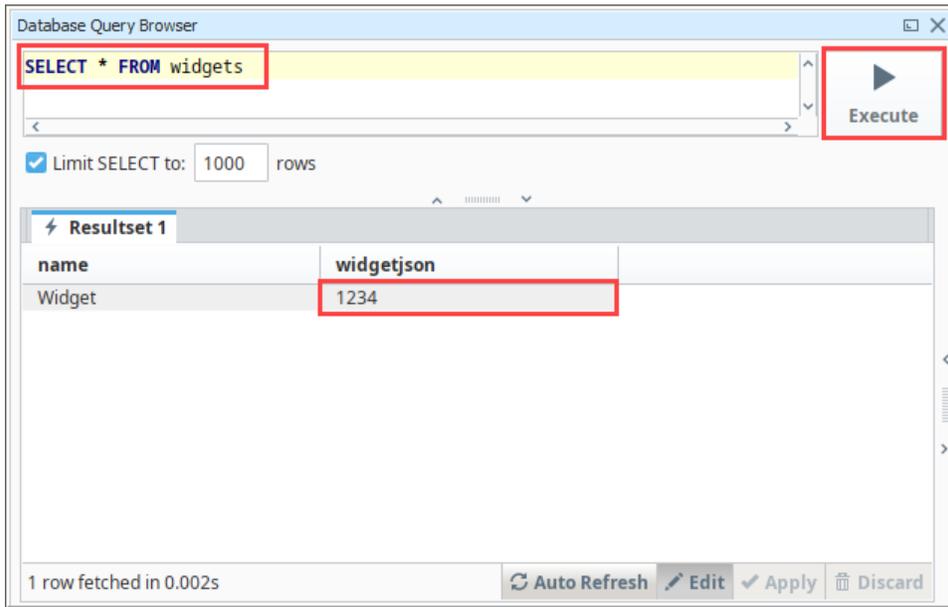
- Select Update Query as the Query Type.
- Enter name into the Name field for the first parameter row.
- Enter widgetjson into the Name field for the second parameter row.
- Confirm both row Data Types are listed as String.



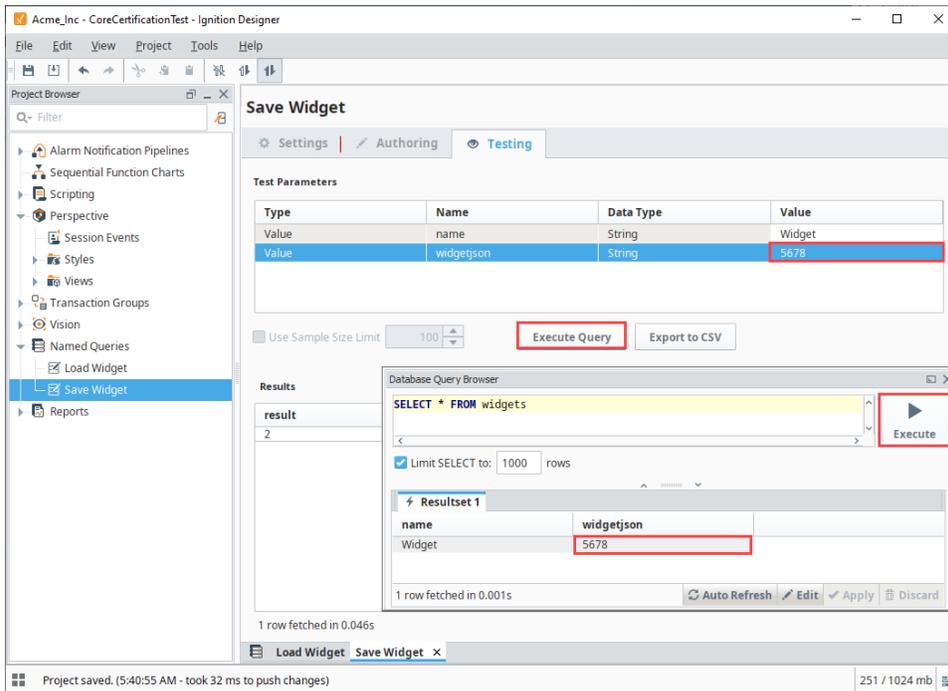
20. Select the Testing tab.
21. Enter Widget into the name row.
22. Enter 5678 into the widgetjson row.
23. Open the Database Query Browser to see how the saved value updates.
24. Enter browsing script to see the current widgets results:

```
SELECT * FROM widgets
```

25. Select Execute to see Widget listed in the name column and 1234 in the widgetjson column.



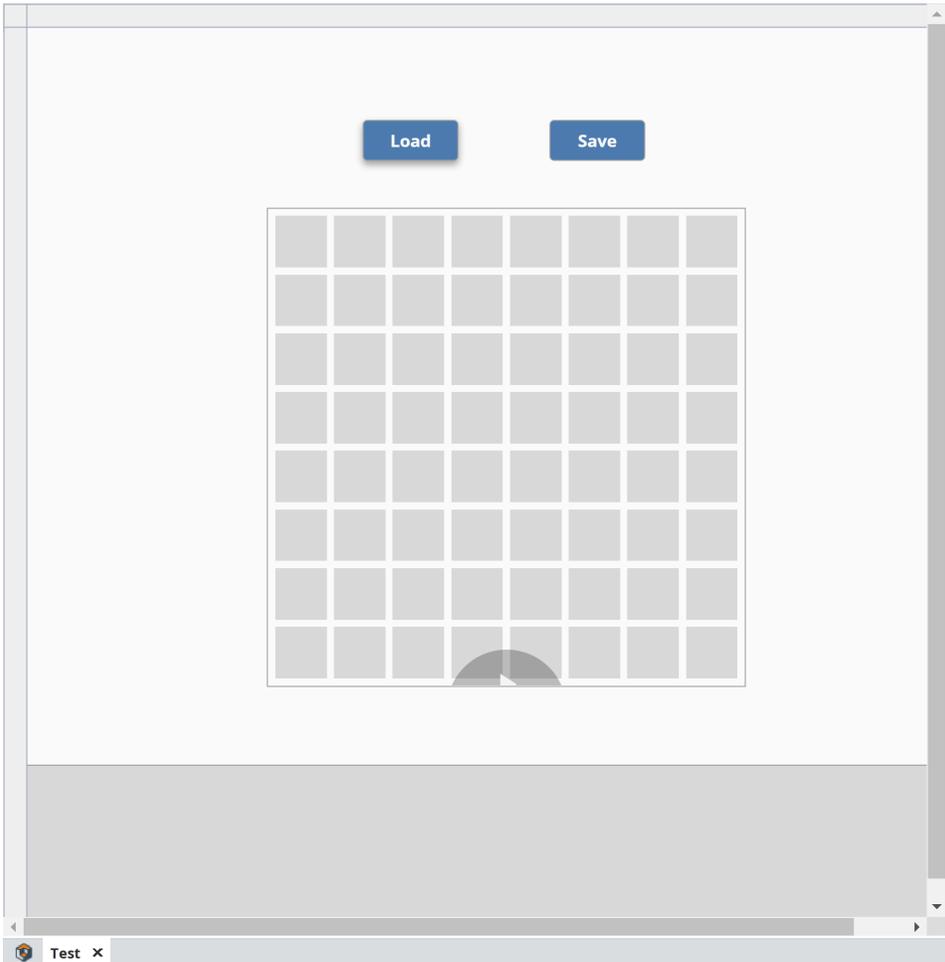
26. Now, select Execute Query for the Save Widget Testing to save a new widgetjson value.
27. Select Execute on the Database Query Browser to see the updated result of 5678 in the widgetjson column.



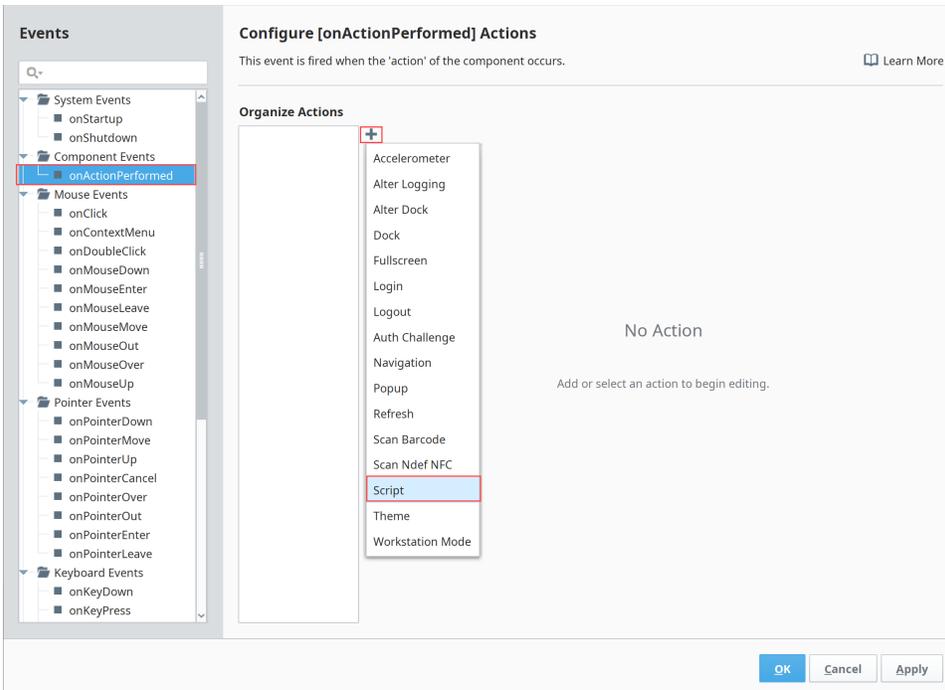
28. Select the Save  icon to save the Save Widget Query.

Next we will create a New View to enable the save and load functionality.

1. Right-click Views and select New View.
2. Enter a name into the Name field. In this example, we used Test as the name. Root Container Type selection does not matter for this example. Selecting Page URL is optional, but can be helpful. In this example, we use the page URL to see the View when we launch a session at the end of the example setup.
3. Drag a Dashboard component onto the view, drag two buttons onto the view.
4. Change one button text to Load and the other to Save.



5. Right-click on the Save button and select Configure Events...
6. Select onActionPerformed.
7. Select the Add  icon.
8. Select Script.



9. Enter scripting to pull name and widgets out of properties:

```
name = "Widget 1"
widgetData = self.getSibling("Dashboard").props.widgets
```

Note: The name can alternatively come from a text entry box.

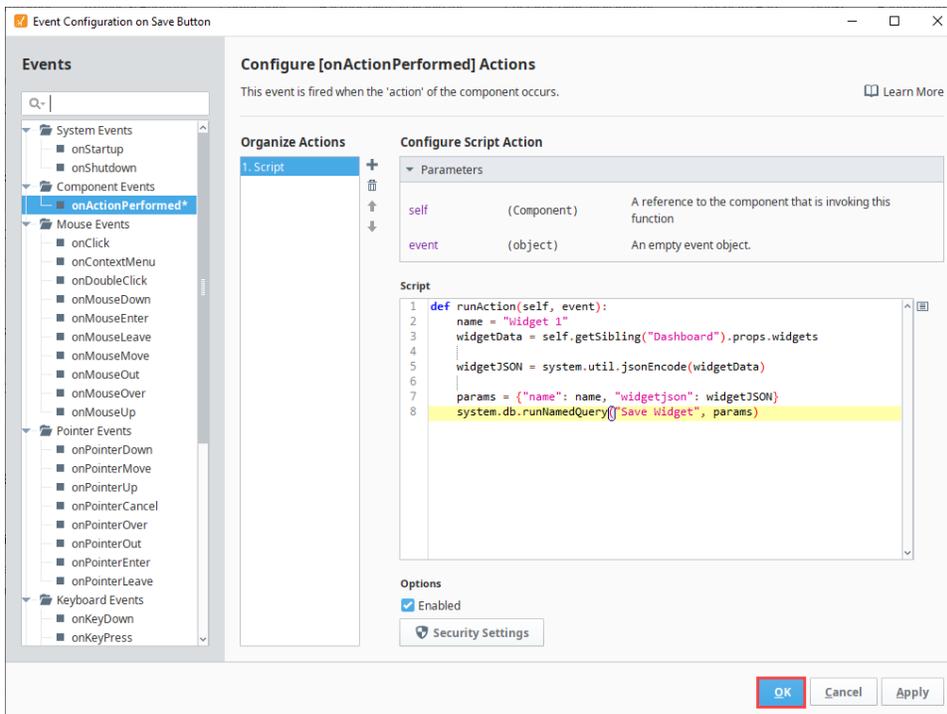
10. Enter scripting to convert the widget json format to a string:

```
widgetJSON = system.util.jsonEncode(widgetData)
```

11. Enter scripting to update variables for the named query and run the query:

```
params = {"name": name, "widgetjson": widgetJSON}
system.db.runNamedQuery("Save Widget", params)
```

12. Select OK.



13. Right-click on the Load button and select Configure Events...

14. Select onActionPerformed.

15. Select the Add  icon.

16. Select Script.

17. Enter scripting to prepare the name of the widgets we want to load:

```
name = "Widget 1"
params = {"name": name}
```

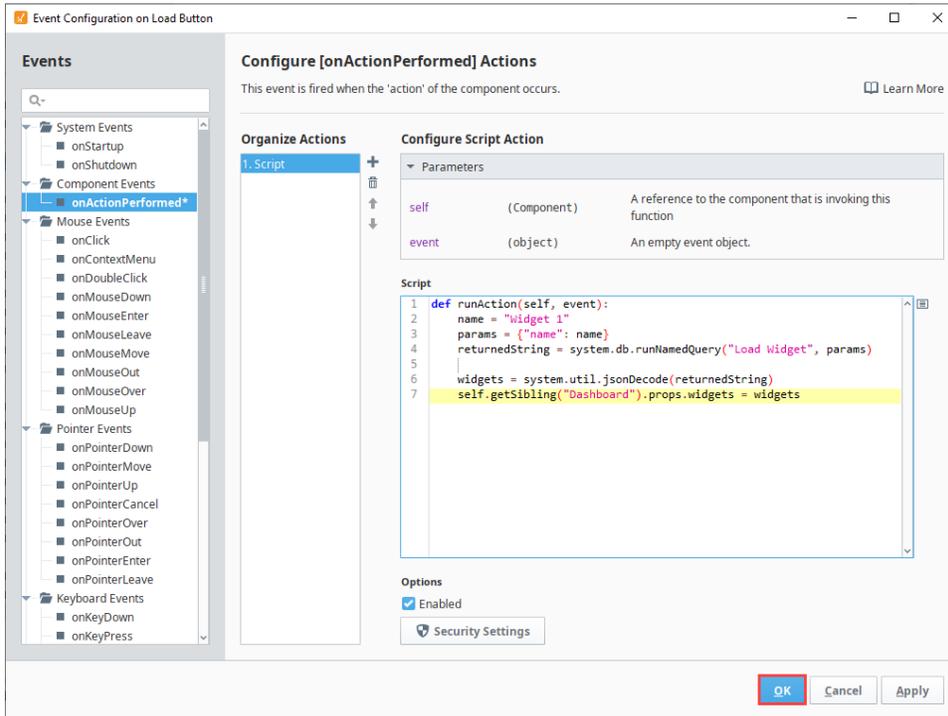
18. Enter scripting to run the Load query:

```
returnedString = system.db.runNamedQuery("Load Widget", params)
```

19. Enter scripting to decode the string into the JSON object and set the Dashboard to the corresponding widgets:

```
widgets = system.util.jsonDecode(returnedString)
self.getSibling("Dashboard").props.widgets = widgets
```

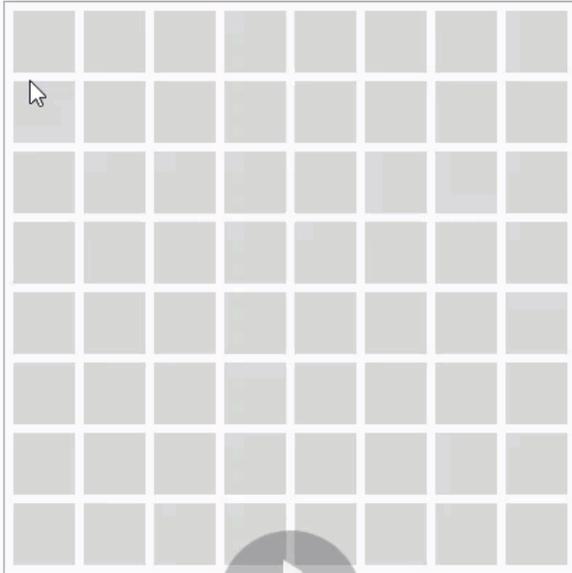
20. Select OK.



Now you can test your Save and Load button functions using the Preview Mode in the Designer or Launching a Perspective Session. Testing will show how users can add, remove, and modify widgets freely, the save the Dashboard configuration so if they continue to modify widgets, the Load button can be used to return the Dashboard to the saved setup.

Load

Save



Perspective - Equipment Schedule



Component Palette Icon:



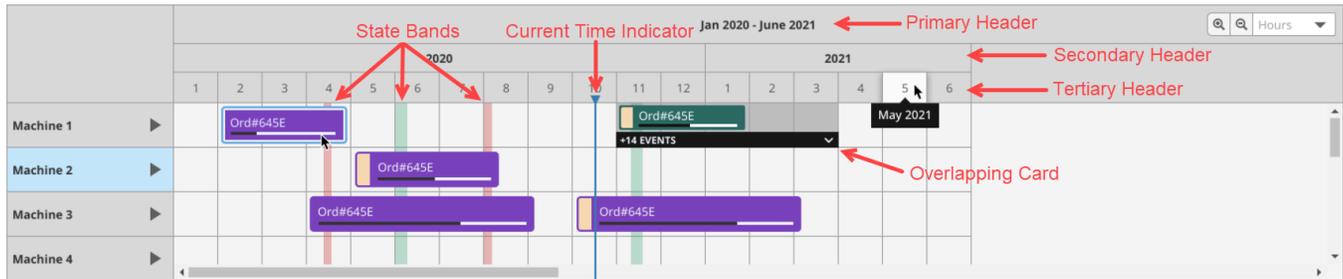
On this page ...

- [Interface Elements](#)
- [Properties](#)
- [Scripting](#)

The following feature is new in Ignition version **8.1.12**
[Click here](#) to check out the other new features

The Equipment Schedule view is a mix between the status chart, gantt chart, and a calendar view. It conveys equipment scheduling information in a concise and easily digestible format.

Interface Elements



Icon	Element	Description
	Current Time Indicator	Marker that indicates the current time, according to the session's timestamp. Style can be configured via <code>currentTimeIndicator</code> .
	State Bands	Bands indicating downtime and break events. Styles can be configured via <code>downtimeEventStyle</code> and <code>breakEventStyle</code> .
	Standard Card	A card on the equipment schedule corresponds to a scheduled event. During a Perspective Session, cards can be moved and resized by clicking and dragging.
	Overlapping Card	When more than one event is scheduled during the same time period for one item, an overlapping card will represent all events scheduled during that time period. Users can click the black drop down bar to select and edit individual events within the overlapping card.
	Zoom level	Sets the zoom level for the chart. Levels can be selected from the drop down menu or by clicking the magnifying glasses to zoom in or out. Valid values include month, day, 12-hr, 8-hr, 6-hr, 3-hr, 15-min, hours, and minutes.
	Primary Header	Header that represents the full range of time displayed on the chart. Styles can be configured via <code>headerStyles.primaryHeaderStyle</code> .
	Secondary Header	Header that represents subsets of time displayed in the Primary Header. Styles can be configured via <code>headerStyles.secondaryHeaderStyle</code> .
	Tertiary Header	Header that represents subsets of time displayed in the Secondary Header. Styles can be configured via <code>headerStyles.tertiaryHeaderStyle</code> .
	Add New Event	Clicking within any empty square on the equipment schedule will allow the user to add a new event onto the chart. Note that Component Events must be configured and the <code>addEnabled</code> property must



be true.

Properties

Name	Description	Data Type																														
addEnabled	If enabled, users can add events to an item's schedule. Note that Component Events must be configured to implement the desired functionality when the events are fired.	boolean																														
resizeEnabled	If enabled, users can resize events. Note that Component Events must be configured to implement the desired functionality when the events are fired.	boolean																														
moveEnabled	If enabled, users can move events. Note that Component Events must be configured to implement the desired functionality when the events are fired.	boolean																														
deleteEnabled	If enabled, users can delete events. Note that Component Events must be configured to implement the desired functionality when the events are fired.	boolean																														
items	The cells, or equipment items, with schedules displayed on the chart. Each row on the equipment schedule corresponds to one item. <table border="1"><thead><tr><th>Name</th><th>Description</th><th>Data Type</th></tr></thead><tbody><tr><td>id</td><td>The ID for the item.</td><td>value: any</td></tr><tr><td>label</td><td>The label for the item displayed on the equipment schedule.</td><td>value: string</td></tr><tr><td>iconConfig</td><td><table border="1"><thead><tr><th>Name</th><th>Description</th><th>Data Type</th></tr></thead><tbody><tr><td>path</td><td>Image path to the icon.</td><td>value: string</td></tr><tr><td>color</td><td>The color of the icon. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td><td>value: string</td></tr><tr><td>style</td><td>Sets a style for the icon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .</td><td>object</td></tr></tbody></table></td><td>object</td></tr><tr><td>headerStyle</td><td>Sets a style for the header. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .</td><td>object</td></tr><tr><td>rowStyle</td><td>Sets a style for the row. Will override default styles set by the component's rowStyle property. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .</td><td>object</td></tr></tbody></table>	Name	Description	Data Type	id	The ID for the item.	value: any	label	The label for the item displayed on the equipment schedule.	value: string	iconConfig	<table border="1"><thead><tr><th>Name</th><th>Description</th><th>Data Type</th></tr></thead><tbody><tr><td>path</td><td>Image path to the icon.</td><td>value: string</td></tr><tr><td>color</td><td>The color of the icon. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td><td>value: string</td></tr><tr><td>style</td><td>Sets a style for the icon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .</td><td>object</td></tr></tbody></table>	Name	Description	Data Type	path	Image path to the icon.	value: string	color	The color of the icon. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	value: string	style	Sets a style for the icon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	object	headerStyle	Sets a style for the header. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	rowStyle	Sets a style for the row. Will override default styles set by the component's rowStyle property. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	array
Name	Description	Data Type																														
id	The ID for the item.	value: any																														
label	The label for the item displayed on the equipment schedule.	value: string																														
iconConfig	<table border="1"><thead><tr><th>Name</th><th>Description</th><th>Data Type</th></tr></thead><tbody><tr><td>path</td><td>Image path to the icon.</td><td>value: string</td></tr><tr><td>color</td><td>The color of the icon. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td><td>value: string</td></tr><tr><td>style</td><td>Sets a style for the icon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .</td><td>object</td></tr></tbody></table>	Name	Description	Data Type	path	Image path to the icon.	value: string	color	The color of the icon. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	value: string	style	Sets a style for the icon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	object																		
Name	Description	Data Type																														
path	Image path to the icon.	value: string																														
color	The color of the icon. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	value: string																														
style	Sets a style for the icon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																														
headerStyle	Sets a style for the header. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																														
rowStyle	Sets a style for the row. Will override default styles set by the component's rowStyle property. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																														
dateRange	<table border="1"><thead><tr><th>Name</th><th>Description</th><th>Data Type</th></tr></thead><tbody><tr><td>startDate</td><td>The beginning of the time range to display.</td><td>value: DateTime</td></tr><tr><td>endDate</td><td>The end of the time range to display.</td><td>value: DateTime</td></tr></tbody></table>	Name	Description	Data Type	startDate	The beginning of the time range to display.	value: DateTime	endDate	The end of the time range to display.	value: DateTime	object																					
Name	Description	Data Type																														
startDate	The beginning of the time range to display.	value: DateTime																														
endDate	The end of the time range to display.	value: DateTime																														
defaultZoom	Default zoom level for the chart. Valid values include month, day, 12-hr, 8-hr, 6-hr, 3-hr, 15-min, hours, and minutes. <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;">Note: The 15-min option is only available in version 8.1.17 and later.</div>	value: string																														
rowHeight	Sets the height for all rows on the equipment schedule.	value: numeric																														
scheduledEvents	The scheduled events for all configured items. <table border="1"><thead><tr><th>Name</th><th>Description</th><th>Data</th></tr></thead></table>	Name	Description	Data	array																											
Name	Description	Data																														

		Type																																							
itemId	The item identifier where this event is scheduled.	value: any																																							
eventID	The event identifier of the scheduled event. The eventID must be unique.	value: any																																							
startDate	The start date of the scheduled event.	value: DateTime																																							
endDate	The end date of the scheduled event.	value: DateTime																																							
label	The label for the event displayed on the equipment schedule.	value: string																																							
leadTime	The amount of lead time to display on the equipment schedule before the scheduled event's startDate.	value: numeric																																							
leadStyle	Sets a style for the leadTime. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																																							
percentDone	If progressBar is enabled, this value will be displayed in the event's progress bar.	value: numeric																																							
style	Sets a style for the event. Will override styles set by scheduledEventStyle. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																																							
downtimeEvents	Downtime events correlated to a specific item.	array																																							
breakEvents	Scheduled breaks, which will appear as downtime for all items.	array																																							
selectedEvent	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td>itemId</td> <td>The ID of the selected item.</td> <td>value: any</td> </tr> <tr> <td>eventID</td> <td>The ID of the selected event.</td> <td>value: any</td> </tr> </tbody> </table>	Name	Description	Data Type	itemId	The ID of the selected item.	value: any	eventID	The ID of the selected event.	value: any	object																														
Name	Description	Data Type																																							
itemId	The ID of the selected item.	value: any																																							
eventID	The ID of the selected event.	value: any																																							
progressBar	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>If enabled, scheduled events will display a progress bar.</td> <td>boolean</td> </tr> <tr> <td>bar</td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>The color of the progress bar. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Sets a style for the progress bar. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .</td> <td>object</td> </tr> </tbody> </table> </td> <td>object</td> </tr> <tr> <td>track</td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>The color of the progress bar track. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Sets a style for the progress bar track. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .</td> <td>object</td> </tr> </tbody> </table> </td> <td>object</td> </tr> <tr> <td>valueDisplay</td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>If enabled, the progress bar will display percentDone as a numeric value.</td> <td>boolean</td> </tr> </tbody> </table> </td> <td>object</td> </tr> </tbody> </table>	Name	Description	Data Type	enabled	If enabled, scheduled events will display a progress bar.	boolean	bar	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>The color of the progress bar. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Sets a style for the progress bar. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Data Type	color	The color of the progress bar. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	value: string	style	Sets a style for the progress bar. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	object	track	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>The color of the progress bar track. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Sets a style for the progress bar track. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Data Type	color	The color of the progress bar track. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	value: string	style	Sets a style for the progress bar track. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	object	valueDisplay	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>If enabled, the progress bar will display percentDone as a numeric value.</td> <td>boolean</td> </tr> </tbody> </table>	Name	Description	Data Type	enabled	If enabled, the progress bar will display percentDone as a numeric value.	boolean	object	object
Name	Description	Data Type																																							
enabled	If enabled, scheduled events will display a progress bar.	boolean																																							
bar	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>The color of the progress bar. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Sets a style for the progress bar. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Data Type	color	The color of the progress bar. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	value: string	style	Sets a style for the progress bar. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	object																														
Name	Description	Data Type																																							
color	The color of the progress bar. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	value: string																																							
style	Sets a style for the progress bar. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																																							
track	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>The color of the progress bar track. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Sets a style for the progress bar track. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Data Type	color	The color of the progress bar track. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	value: string	style	Sets a style for the progress bar track. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	object																														
Name	Description	Data Type																																							
color	The color of the progress bar track. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	value: string																																							
style	Sets a style for the progress bar track. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																																							
valueDisplay	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>If enabled, the progress bar will display percentDone as a numeric value.</td> <td>boolean</td> </tr> </tbody> </table>	Name	Description	Data Type	enabled	If enabled, the progress bar will display percentDone as a numeric value.	boolean	object																																	
Name	Description	Data Type																																							
enabled	If enabled, the progress bar will display percentDone as a numeric value.	boolean																																							

	format	Sets a numeric format for the valueDisplay. Valid values include none, integer, percent, currency, and duration.	value: string	
	justify	Sets alignment for the valueDisplay on the progress bar. Valid values include left, center, and right.	value: string	
	style	Sets a style for the value display. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	
currentTimeIndicator	Name	Description	Data Type	object
	color	The color of the current time indicator. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	value: string	
	opacity	The opacity level for the current time indicator.	value: numeric	
	width	The width of the current time indicator.	value: numeric	
headerStyles	Name	Description	Data Type	object
	primaryHeaderStyle	Sets a style for the primary header. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	
	secondaryHeaderStyle	Sets a style for the secondary header. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	
	tertiaryHeaderStyle	Sets a style for the tertiary header. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	
rowStyle	Sets a style for all rows. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .			object
selectedEventStyle	Sets a style for the selected event. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .			object
scheduledEventStyle	Sets a style for scheduled events. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .			object
downtimeEventStyle	Sets a style for downtime events. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .			object
breakEventStyle	Sets a style for break events. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .			object
bodyStyle	Sets a style for the body. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .			object
style	Sets a style for the component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .			object

Scripting

See the [Perspective - Equipment Schedule Scripting page](#) for the full list of scripting functions available for this component.

Perspective - Equipment Schedule Scripting

This page details the various scripting, component, and extension functions available for [Perspective's Equipment Schedule](#) component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

On this page ...

- [Component Events](#)
 - [onAddEvent](#)
 - [onMoveEvent](#)
 - [onDeleteEvent](#)
 - [onClickEvent](#)
- [Component Functions](#)
- [Extension Functions](#)

onAddEvent

Event is fired after a user adds an event to the schedule.

Note: This component event is designed to be used in tandem with a script runAction. Within the script action, special properties and methods are available on the **event** object, which is passed to the script action as a parameter.

event.end

- Object Path

event.end

- Type

[Integer or float](#)

- Description

The end date of the new event.

event.itemId

- Object Path

event.itemId

- Type

[String, integer, or float](#)

- Description

The item identifier where this event was created.

event.start

- Object Path

event.start

- Type

[Integer or float](#)

- Description

The start date of the new event.

Example

```
from random import random, randint
```

```
format = "yyyy-MM-dd HH:mm:ss z"
locale = "en-us"
scheduled = self.props.scheduledEvents
r = random() * 255
g = random() * 255
b = random() * 255
color = "rgb(" + str(r) + "," + str(g) + "," + str(b) + ")"
percentDone = randint(0, 100)
leadTime = randint(300, 6000)

item = {
  "endDate": system.date.parse(event.end, format, locale),
  "itemId": event.itemId,
  "startDate": system.date.parse(event.start, format, locale),
  "eventId": "event_" + str(randint(1000, 10000)),
  "backgroundColor": color,
  "percentDone": percentDone,
  "leadTime": leadTime
}
scheduled.append(item)
```

onMoveEvent

Event is fired after a user moves an event.

Note: This component event is designed to be used in tandem with a script runAction. Within the script action, special properties and methods are available on the **event** object, which is passed to the script action as a parameter.

event.end

- Object Path

event.end

- Type

Integer or float

- Description

The new end date of the moved event.

event.eventId

- Object Path

event.eventId

- Type

String, integer, or float

- Description

The event identifier of the moved event.

event.itemId

- Object Path

event.itemId

- Type

String, integer, or float

- Description

The item identifier where this event was moved.

event.start

- Object Path

event.start

- Type

Integer or float

- Description

The new start date of the moved event.

Example

```
format = "yyyy-MM-dd HH:mm:ss z"
locale = "en-us"
scheduled = self.props.scheduledEvents
for count, toFind in enumerate(scheduled):
    if toFind.eventId == event.eventId and event.itemId == toFind.itemId:
        toFind.startDate = system.date.parse(event.start, format, locale)
        toFind.endDate = system.date.parse(event.end, format, locale)
```

onResizeEvent

Event is fired after a user resizes an event.

Note: This component event is designed to be used in tandem with a script runAction. Within the script action, special properties and methods are available on the **event** object, which is passed to the script action as a parameter.

event.end

- Object Path

event.end

- Type

Integer or float

- Description

The new end date of the resized event.

event.eventId

- Object Path

event.eventId

- Type

String, integer, or float

- Description

The event identifier of the resized event.

event.itemId

- Object Path

event.itemId

- Type

String, integer, or float

- Description

The item identifier where this event was resized.

event.start

- Object Path

event.start

- Type

Integer or float

- Description

The new start date of the resized event.

Example

```
format = "yyyy-MM-dd HH:mm:ss z"
locale = "en-us"
scheduled = self.props.scheduledEvents
for count, toFind in enumerate(scheduled):
    if toFind.eventId == event.eventId and event.itemId == toFind.itemId:
        toFind.startDate = system.date.parse(event.start, format, locale)
        toFind.endDate = system.date.parse(event.end, format, locale)
```

onDeleteEvent

Event is fired after a user deletes an event.

Note: This component event is designed to be used in tandem with a script runAction. Within the script action, special properties and methods are available on the **event** object, which is passed to the script action as a parameter.

event.end

- Object Path

event.end

- Type

Integer or float

- Description

The end date of the deleted event.

event.eventId

- Object Path

event.eventId

- Type

String, integer, or float

- Description

The event identifier of the deleted event.

event.itemId

- Object Path

event.itemId

- Type

String, integer, or float

- Description

The item identifier where this event was deleted.

event.start

- Object Path

event.start

- Type

Integer or float

- Description

The start date of the deleted event.

Example

```
scheduled = self.props.scheduledEvents
format = "yyyy-MM-dd HH:mm:ss z"
locale = "en-us"
for count, toFind in enumerate(scheduled):
    if toFind.eventId == event.eventId and toFind.itemId == event.itemId:
        del scheduled[count]
        break
```

onClickEvent

Event is fired after a user clicks on an event.

Note: This component event is designed to be used in tandem with a script runAction. Within the script action, special properties and methods are available on the **event** object, which is passed to the script action as a parameter.

event.end

- Object Path

event.end

- Type

Integer or float

- Description

The end date of the clicked event.

event.eventId

- Object Path

event.eventId

- Type

String, integer, or float

- Description

The event identifier of the clicked event.

event.itemId

- Object Path

event.itemId

- Type

String, integer, or float

- Description

The item identifier where this event was clicked.

event.start

- Object Path

event.start

- Type

Integer or float

- Description

The start date of the clicked event.

Example

```
scheduled = self.props.scheduledEvents
format = "yyyy-MM-dd HH:mm:ss z"
locale = "en-us"
for count, toFind in enumerate(scheduled):
    if toFind.eventId == event.eventId and toFind.itemId == event.itemId:
        print scheduled[count]
        break
```

Component Functions

This component does not have component functions associated with it.

Extension Functions

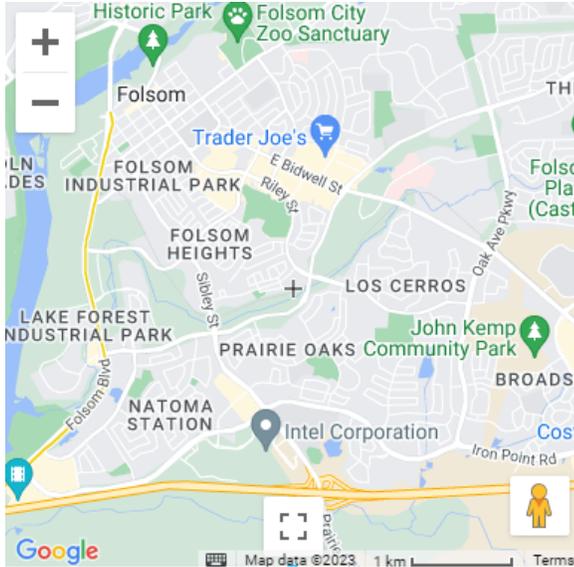
This component does not have extension functions associated with it.

Perspective - Google Map

The following feature is new in Ignition version **8.1.33**
[Click here](#) to check out the other new features

On this page ...

- [Properties](#)
 - [Layers Properties](#)
- [Scripting](#)
- [Example](#)



Component Palette Icon:



The Google Map component provides a new interactive map option to the [Map](#) component. The Google Map component is based on the [LeafletJS](#) plugin to use Google maps basemaps and allows scriptable interactions and various layer configurations including traffic, polygons, markers, and overlays.



A Google Map API Key is required to be entered in the new `googleMapsApiKey` session prop to provide a functional map without a Development Mode watermark. This API key will be exposed on the web application, so it is recommended to add restrictions following Google's [API Security Best Practices](#).

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
init	The map's initial state on load.	object
center	Initial geographic center of the map.	object
lat	Latitudinal coordinate. Default value is 38.660867.	value: numeric

	<table border="1"> <tr> <td>lng</td> <td>Longitude. Default value is -121.159728.</td> <td>value: numeric</td> </tr> <tr> <td>zoom</td> <td>Initial map zoom level. Default is 13.</td> <td>value: numeric</td> </tr> </table>	lng	Longitude. Default value is -121.159728.	value: numeric	zoom	Initial map zoom level. Default is 13.	value: numeric													
lng	Longitude. Default value is -121.159728.	value: numeric																		
zoom	Initial map zoom level. Default is 13.	value: numeric																		
backgroundColor	Color used for the background of the map when the tiles have not yet loaded as the user pans.	value: string																		
clickableIcons	Allow map icons to be clickable. These icons are also known as points of interest (POI) and represent areas like parks, schools, and shops. Default is true.	value: boolean																		
cursor	<p>Cursor related configuration.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>draggable</td> <td>The name or URL of the cursor to display when mousing over a draggable map. This property uses the CSS cursor attribute to change the icon. As with the CSS property, you must specify at least one fallback cursor that is not a URL.</td> <td>value: string</td> </tr> <tr> <td>dragging</td> <td>The name or URL of the cursor to display when the map is being dragged. This property uses the CSS cursor attribute to change the icon. As with the CSS property, you must specify at least one fallback cursor that is not a URL.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	draggable	The name or URL of the cursor to display when mousing over a draggable map. This property uses the CSS cursor attribute to change the icon. As with the CSS property, you must specify at least one fallback cursor that is not a URL.	value: string	dragging	The name or URL of the cursor to display when the map is being dragged. This property uses the CSS cursor attribute to change the icon. As with the CSS property, you must specify at least one fallback cursor that is not a URL.	value: string	object									
Name	Description	Property Type																		
draggable	The name or URL of the cursor to display when mousing over a draggable map. This property uses the CSS cursor attribute to change the icon. As with the CSS property, you must specify at least one fallback cursor that is not a URL.	value: string																		
dragging	The name or URL of the cursor to display when the map is being dragged. This property uses the CSS cursor attribute to change the icon. As with the CSS property, you must specify at least one fallback cursor that is not a URL.	value: string																		
controlsSize	Size in pixels of the controls appearing on the map.	value: numeric																		
disableDefaultUI	Enables/Disables all default UI buttons. May be overridden individually. Does not disable the keyboard controls, which are separately controlled by the keyboardShortcuts property. Does not disable gesture controls, which are separately controlled by the gestureHandling property.	value: boolean																		
fullscreen	<p>Fullscreen related configuration.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>controls</td> <td>Enabled/Disabled state of the Fullscreen control. Default is enabled.</td> <td>value: boolean</td> </tr> <tr> <td>controlPosition</td> <td>Control display position. Default is top_left.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	controls	Enabled/Disabled state of the Fullscreen control. Default is enabled.	value: boolean	controlPosition	Control display position. Default is top_left.	value: string	object									
Name	Description	Property Type																		
controls	Enabled/Disabled state of the Fullscreen control. Default is enabled.	value: boolean																		
controlPosition	Control display position. Default is top_left.	value: string																		
gestureHandling	Controls how the map handles gestures. Options include cooperative, greedy, auto, and none. Default is set to auto.	value: string																		
heading	The heading for aerial imagery in degrees measured clockwise from cardinal direction North.	value: numeric																		
isFractionalZoomEnabled	Whether the map should allow fractional zoom levels.	value: boolean																		
keyboardShortcuts	Allows the map to be controlled by the keyboard when set to true.	value: boolean																		
layers	Map layers configuration. See the Layers section below for more detail on the available map layers.	object																		
mapID	The Map ID of the map.	value: string																		
mapType	<p>MapType related configuration.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>The initial Map mapType id.</td> <td>value: string</td> </tr> <tr> <td>controls</td> <td>The initial enabled/disabled state of the map type control.</td> <td>value: boolean</td> </tr> <tr> <td>controlPosition</td> <td>Control display position. Default is top_right.</td> <td>value: string</td> </tr> <tr> <td>controlMapTypeIds</td> <td>IDs of map types to show in the control.</td> <td>object</td> </tr> <tr> <td>controlStyle</td> <td>Style of map type control to display.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	id	The initial Map mapType id.	value: string	controls	The initial enabled/disabled state of the map type control.	value: boolean	controlPosition	Control display position. Default is top_right.	value: string	controlMapTypeIds	IDs of map types to show in the control.	object	controlStyle	Style of map type control to display.	value: string	object
Name	Description	Property Type																		
id	The initial Map mapType id.	value: string																		
controls	The initial enabled/disabled state of the map type control.	value: boolean																		
controlPosition	Control display position. Default is top_right.	value: string																		
controlMapTypeIds	IDs of map types to show in the control.	object																		
controlStyle	Style of map type control to display.	value: string																		
restriction	Defines a boundary that restricts the area of the map accessible to users. When set, a user can only pan and zoom while the camera view stays inside the limits of the boundary.	object																		

Name	Description	Property Type																
bounds	When set, a user can only pan and zoom inside the given bounds. Bounds can restrict both longitude and latitude, or can restrict latitude only. For latitude-only bounds, use west and east longitudes of -180 and 180, respectively. For example: north: northLat, south: southLat, west: -180, east: 180.	object																
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>north</td> <td>Latitude for Northern boundary.</td> <td>value: string</td> </tr> <tr> <td>east</td> <td>Longitude for the Eastern boundary.</td> <td>value: string</td> </tr> <tr> <td>south</td> <td>Latitude for the Southern boundary.</td> <td>value: string</td> </tr> <tr> <td>west</td> <td>Longitude for the Western boundary.</td> <td>value: string</td> </tr> </tbody> </table>		Name	Description	Property Type	north	Latitude for Northern boundary.	value: string	east	Longitude for the Eastern boundary.	value: string	south	Latitude for the Southern boundary.	value: string	west	Longitude for the Western boundary.	value: string	
	Name		Description	Property Type														
	north		Latitude for Northern boundary.	value: string														
	east		Longitude for the Eastern boundary.	value: string														
south	Latitude for the Southern boundary.	value: string																
west	Longitude for the Western boundary.	value: string																
strict	Bounds can be made more restrictive by setting the strictBounds flag to true. This reduces how far a user can zoom out, ensuring that everything outside of the restricted bounds stays hidden. The default is false, meaning that a user can zoom out until the entire bounded area is in view, possibly including areas outside the bounded area.	value: boolean																
rotate	Rotate related configuration.	object																
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>controls</td> <td>Enabled/Disabled state of the Rotate control. Default is enabled.</td> <td>value: boolean</td> </tr> <tr> <td>controlPosition</td> <td>Control display position. Default is top_left.</td> <td>value: string</td> </tr> </tbody> </table>		Name	Description	Property Type	controls	Enabled/Disabled state of the Rotate control. Default is enabled.	value: boolean	controlPosition	Control display position. Default is top_left.	value: string							
Name	Description		Property Type															
controls	Enabled/Disabled state of the Rotate control. Default is enabled.	value: boolean																
controlPosition	Control display position. Default is top_left.	value: string																
scale	Scale related configuration.	object																
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>controls</td> <td>Enabled/Disabled state of the Scale control.</td> <td>value: boolean</td> </tr> </tbody> </table>		Name	Description	Property Type	controls	Enabled/Disabled state of the Scale control.	value: boolean										
Name	Description	Property Type																
controls	Enabled/Disabled state of the Scale control.	value: boolean																
tilt	Controls the automatic switching behavior for the angle of incidence of the map. The only allowed values are 0 and 45.	value: numeric																
zoom	Zoom related configuration.	object																
			<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>controls</td> <td>Enabled/Disabled state of the Zoom control. Default is enabled.</td> <td>value: boolean</td> </tr> <tr> <td>controlPosition</td> <td>Control display position. Default is top_left.</td> <td>value: string</td> </tr> <tr> <td>max</td> <td>The maximum zoom level which will be displayed on the map.</td> <td>value: numeric</td> </tr> <tr> <td>min</td> <td>The minimum zoom level which will be displayed on the map.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	controls	Enabled/Disabled state of the Zoom control. Default is enabled.	value: boolean	controlPosition	Control display position. Default is top_left.	value: string	max	The maximum zoom level which will be displayed on the map.	value: numeric	min	The minimum zoom level which will be displayed on the map.	value: numeric
	Name		Description	Property Type														
	controls		Enabled/Disabled state of the Zoom control. Default is enabled.	value: boolean														
	controlPosition		Control display position. Default is top_left.	value: string														
max	The maximum zoom level which will be displayed on the map.	value: numeric																
min	The minimum zoom level which will be displayed on the map.	value: numeric																
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																

Layers Properties

Property	Description	Proper Type													
overlayView	Renders instances of Perspective views within the active project.	object													
			<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>A unique name which is used to distinguish this element from others. This name is provided to any applicable interaction event handlers, such as click handlers.</td> <td>value: string</td> </tr> <tr> <td>enabled</td> <td>Enable/disable rendering of the OverlayView layer.</td> <td>value: boolean</td> </tr> <tr> <td>mapPan</td> <td></td> <td>value:</td> </tr> </tbody> </table>	Name	Description	Property Type	name	A unique name which is used to distinguish this element from others. This name is provided to any applicable interaction event handlers, such as click handlers.	value: string	enabled	Enable/disable rendering of the OverlayView layer.	value: boolean	mapPan		value:
	Name		Description	Property Type											
	name		A unique name which is used to distinguish this element from others. This name is provided to any applicable interaction event handlers, such as click handlers.	value: string											
enabled	Enable/disable rendering of the OverlayView layer.	value: boolean													
mapPan		value:													

eName	Map pane in which to render the OverlayView. Options include floatPane, mapPane, markerLayer, overlayLayer, overlayMouseTarget.	string																																	
bounds	Sets the bounds of the overlay. Either of bounds or position are required.	object																																	
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>corner1</td> <td>South-West corner of the overlayView.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>corner2</td> <td>North-East corner of the overlayView.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	corner1	South-West corner of the overlayView.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate.	value: numeric	lng	Longitude coordinate.	value: numeric		corner2	North-East corner of the overlayView.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate.	value: numeric	lng	Longitude coordinate.	value: numeric		
Name	Description	Property Type																																	
corner1	South-West corner of the overlayView.	object																																	
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate.	value: numeric	lng	Longitude coordinate.	value: numeric																									
Name	Description	Property Type																																	
lat	Latitudinal coordinate.	value: numeric																																	
lng	Longitude coordinate.	value: numeric																																	
corner2	North-East corner of the overlayView.	object																																	
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate.	value: numeric	lng	Longitude coordinate.	value: numeric																									
Name	Description	Property Type																																	
lat	Latitudinal coordinate.	value: numeric																																	
lng	Longitude coordinate.	value: numeric																																	
position	OverlayView position. Either of bounds or positions are required.	object																																	
views	OverlayView instance configuration.	object																																	
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Path to the view instance.</td> <td>value: string</td> </tr> <tr> <td>params</td> <td>Params to pass to the view instance.</td> <td>object</td> </tr> <tr> <td>properties</td> <td>Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.</td> <td>object</td> </tr> <tr> <td>style</td> <td>Sets a style for the selected view. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Path to the view instance.	value: string	params	Params to pass to the view instance.	object	properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	object	style	Sets a style for the selected view. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																			
Name	Description	Property Type																																	
path	Path to the view instance.	value: string																																	
params	Params to pass to the view instance.	object																																	
properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	object																																	
style	Sets a style for the selected view. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																																	
transparentBackground	If enabled, disables the default background color.	value: boolean																																	
shadow	If enabled, apply a box shadow around the view.	value: boolean																																	
zIndex	The z-index of the layer.	value: numeric																																	
properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	object																																	
style	Sets a style for this overlayView layer. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																																	
groundOverlay	Binds and scales an image to fit projected ground coordinates.	object																																	
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>A unique name which is used to distinguish this element from others. This name is provided to any applicable interaction event handlers, such as click handlers.</td> <td>value: string</td> </tr> <tr> <td>enabled</td> <td>Enable/disable rendering of GroundOverlay layer.</td> <td>value: boolean</td> </tr> <tr> <td>overlays</td> <td>Overlay related configuration.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	name	A unique name which is used to distinguish this element from others. This name is provided to any applicable interaction event handlers, such as click handlers.	value: string	enabled	Enable/disable rendering of GroundOverlay layer.	value: boolean	overlays	Overlay related configuration.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> </tbody> </table>	Name	Description	Property Type																	
Name	Description	Property Type																																	
name	A unique name which is used to distinguish this element from others. This name is provided to any applicable interaction event handlers, such as click handlers.	value: string																																	
enabled	Enable/disable rendering of GroundOverlay layer.	value: boolean																																	
overlays	Overlay related configuration.	object																																	
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> </tbody> </table>	Name	Description	Property Type																															
Name	Description	Property Type																																	

source	The URL of the image to display.	value: string																																	
bounds	Sets the bounds of the GroundOverlay.	object																																	
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>corner1</td> <td>South-West corner of the GroundOverlay.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>corner2</td> <td>North-East corner of the GroundOverlay.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	corner1	South-West corner of the GroundOverlay.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate.	value: numeric	lng	Longitude coordinate.	value: numeric		corner2	North-East corner of the GroundOverlay.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate.	value: numeric	lng	Longitude coordinate.	value: numeric		
Name	Description	Property Type																																	
corner1	South-West corner of the GroundOverlay.	object																																	
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate.	value: numeric	lng	Longitude coordinate.	value: numeric																									
Name	Description	Property Type																																	
lat	Latitudinal coordinate.	value: numeric																																	
lng	Longitude coordinate.	value: numeric																																	
corner2	North-East corner of the GroundOverlay.	object																																	
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate.	value: numeric	lng	Longitude coordinate.	value: numeric																									
Name	Description	Property Type																																	
lat	Latitudinal coordinate.	value: numeric																																	
lng	Longitude coordinate.	value: numeric																																	
properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	object																																	
clickable	If true, the ground overlay can receive mouse events.	value: boolean																																	
opacity	The opacity of the overlay, expressed as a number between 0 and 1. Optional.	value: numeric																																	
properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	object																																	

marker

A marker identifies a location on a map. By default, a marker uses a standard image. Markers can display custom images

object

Name	Description	Property Type																											
name	A unique name which is used to distinguish this element from others. This name is provided to any applicable interaction event handlers, such as click handlers.	value: string																											
enabled	Enable/disable rendering of the marker layer.	value: boolean																											
visible	If true, the marker is visible.	value: boolean																											
markers	Marker related configuration.	object																											
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>position</td> <td>Marker position configuration.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>icon</td> <td>Marker icon configuration.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property</th> </tr> </thead> <tbody> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	position	Marker position configuration.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate.	value: numeric	lng	Longitude coordinate.	value: numeric		icon	Marker icon configuration.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property</th> </tr> </thead> <tbody> </tbody> </table>	Name	Description	Property		
Name	Description	Property Type																											
position	Marker position configuration.	object																											
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate.	value: numeric	lng	Longitude coordinate.	value: numeric																			
Name	Description	Property Type																											
lat	Latitudinal coordinate.	value: numeric																											
lng	Longitude coordinate.	value: numeric																											
icon	Marker icon configuration.	object																											
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property</th> </tr> </thead> <tbody> </tbody> </table>	Name	Description	Property																									
Name	Description	Property																											

			Type	
	type	The icon type used as to represent the marker. Options include default, image, and symbol.	value: string	
	popup	If enabled, configured popups can be made visible for this marker. Popups do not apply to clusters.	object	
	tooltip	If enabled, configured tooltips can be made visible for this marker. Tooltips do not apply to clusters.	object	
	properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	object	
clusterMarkers	Enables marker clustering.			value: boolean
animation	Starts an animation. Any ongoing animation will be cancelled. Passing in null will cause any animation to stop. Default options include none, bounce, and drop.			value: string
clickable	If true, the marker receives mouse and touch events.			value: boolean
crossOnDrag	If false, disables cross that appears beneath the marker when dragging.			value: boolean
cursor	Mouse cursor to show on hover.			value: string
draggable	If true, the marker can be dragged.			value: boolean
icon	Marker icon configuration.			object
	Name	Description	Property Type	
	type	Selects shape configuration. Options include default, image, and symbol.	value: string	
label	Adds a label to the marker.			value: string
opacity	The marker's opacity between 0.0 and 1.0.			value: numeric
optimized	Optimization enhances performance by rendering many markers as a single static element. This is useful in cases where a large number of markers is required.			value: boolean
popup	If enabled, popup configuration for this marker. Popups do not apply to clusters.			object
	Name	Description	Property Type	
	enabled	Enable marker popup.	value: boolean	
	visible	If true, the popup is visible.	value: boolean	
	content	The popup content to display.		object
	Name	Description	Property Type	
	text	Text to display.	value: numeric	
	view	A view to display as popup content. If configured, overrides the text property.		object
	Name	Description	Property Type	
	path	Path of view	value:	

		to display.	string
	params	Parameters to be passed to the view. Names in this object must match input parameters defined on the view.	object
autoClose	Set it to false if you want to override the default behavior of the popup closing when another popup is opened.		value: boolean
width	Width of the popup. Min and Max values allowed.		object
disableAutoPan	Disable panning the map to make the popup fully visible when it opens.		value: boolean
pixelOffset	The offset, in pixels, of the tip of the popup from the x and y points on the map at whose geographical coordinates the popup is anchored.		object
ariaLabel	AriaLabel to assign to the popup.		value: string
zIndex	All popups are displayed on the map in order of their z-index, with higher values displaying in front of popups with lower values. By default, popups are displayed according to their latitude, with popups of lower latitudes appearing in front of popups at higher latitudes. Popups are always displayed in front of markers.		value: numeric
style	Sets a style for the popup. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .		object

tooltip If enabled, tooltips can be configured for this marker. Tooltips do not apply to clusters. object

Name	Description	Property Type																					
enabled	Enable marker tooltip.	value: boolean																					
content	The tooltip content to display.	object																					
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Text to display.</td> <td>value: numeric</td> </tr> <tr> <td>view</td> <td>A view to display as tooltip content. If configured, overrides the text property.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Path of view to display.</td> <td>value: string</td> </tr> <tr> <td>params</td> <td>Parameters to be passed to the view. Names in this object must match input parameters defined on the view.</td> <td>object</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	text	Text to display.	value: numeric	view	A view to display as tooltip content. If configured, overrides the text property.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Path of view to display.</td> <td>value: string</td> </tr> <tr> <td>params</td> <td>Parameters to be passed to the view. Names in this object must match input parameters defined on the view.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Path of view to display.	value: string	params	Parameters to be passed to the view. Names in this object must match input parameters defined on the view.	object		
Name	Description	Property Type																					
text	Text to display.	value: numeric																					
view	A view to display as tooltip content. If configured, overrides the text property.	object																					
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Path of view to display.</td> <td>value: string</td> </tr> <tr> <td>params</td> <td>Parameters to be passed to the view. Names in this object must match input parameters defined on the view.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Path of view to display.	value: string	params	Parameters to be passed to the view. Names in this object must match input parameters defined on the view.	object													
Name	Description	Property Type																					
path	Path of view to display.	value: string																					
params	Parameters to be passed to the view. Names in this object must match input parameters defined on the view.	object																					
direction	Direction where to open the tooltip. Possible values are right, left, top, bottom, center, and auto. Auto will dynamically switch between right and left according to the tooltip position on the map.	value: string																					
pixelOffset	The offset, in pixels, of the tooltip from the marker's position. For	object																					

set	tooltips configured with auto direction, positive offsets move the tooltip away from the marker's position.							
permanent	If true, the tooltip will display permanently, instead of only on mouseover.	value: boolean						
style	Sets a style for the tooltip. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object						
shape	Image map region definition used for drag/click.	object						
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>type</td> <td>Selects shape configuration. Options include none, circle, poly, rec.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	type	Selects shape configuration. Options include none, circle, poly, rec.	value: string	
Name	Description	Property Type						
type	Selects shape configuration. Options include none, circle, poly, rec.	value: string						
title	Rollover text	value: string						
zIndex	The z-index of the layer.	value: numeric						
properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	object						

circle

An array of circle layers.		object																																																								
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>A unique name which is used to distinguish this element from others. This name is provided to any applicable interaction event handlers, such as click handlers.</td> <td>value: string</td> </tr> <tr> <td>enabled</td> <td>Enable/disable rendering of the circle layer.</td> <td>value: boolean</td> </tr> <tr> <td>visible</td> <td>If true, the circle is visible.</td> <td>value: boolean</td> </tr> <tr> <td>circles</td> <td>Sets the center of the circle.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>center</td> <td>Circle center configuration.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>fill</td> <td>Circle fill configuration. Default properties include enabled, color, and opacity.</td> <td>object</td> </tr> <tr> <td>stroke</td> <td>Stroke configuration. Default properties include enabled, color, opacity, position, and weight.</td> <td>object</td> </tr> <tr> <td>properties</td> <td>Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.</td> <td>object</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>radius</td> <td>Sets the radius of this circle in meters.</td> <td>value: numeric</td> </tr> <tr> <td>clickable</td> <td>Indicates whether this circle handles mouse events.</td> <td>value: boolean</td> </tr> <tr> <td>draggable</td> <td>If set to true, the user can drag this circle over the map.</td> <td>value: boolean</td> </tr> <tr> <td>editable</td> <td>If set to true, the user can edit this circle by dragging the control points shown at the center and around the circumference of the circle.</td> <td>value: boolean</td> </tr> </tbody> </table>	Name	Description	Property Type	name	A unique name which is used to distinguish this element from others. This name is provided to any applicable interaction event handlers, such as click handlers.	value: string	enabled	Enable/disable rendering of the circle layer.	value: boolean	visible	If true, the circle is visible.	value: boolean	circles	Sets the center of the circle.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>center</td> <td>Circle center configuration.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>fill</td> <td>Circle fill configuration. Default properties include enabled, color, and opacity.</td> <td>object</td> </tr> <tr> <td>stroke</td> <td>Stroke configuration. Default properties include enabled, color, opacity, position, and weight.</td> <td>object</td> </tr> <tr> <td>properties</td> <td>Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	center	Circle center configuration.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate.	value: numeric	lng	Longitude coordinate.	value: numeric		fill	Circle fill configuration. Default properties include enabled, color, and opacity.	object	stroke	Stroke configuration. Default properties include enabled, color, opacity, position, and weight.	object	properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	object		radius	Sets the radius of this circle in meters.	value: numeric	clickable	Indicates whether this circle handles mouse events.	value: boolean	draggable	If set to true, the user can drag this circle over the map.	value: boolean	editable	If set to true, the user can edit this circle by dragging the control points shown at the center and around the circumference of the circle.	value: boolean
Name	Description	Property Type																																																								
name	A unique name which is used to distinguish this element from others. This name is provided to any applicable interaction event handlers, such as click handlers.	value: string																																																								
enabled	Enable/disable rendering of the circle layer.	value: boolean																																																								
visible	If true, the circle is visible.	value: boolean																																																								
circles	Sets the center of the circle.	object																																																								
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>center</td> <td>Circle center configuration.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>fill</td> <td>Circle fill configuration. Default properties include enabled, color, and opacity.</td> <td>object</td> </tr> <tr> <td>stroke</td> <td>Stroke configuration. Default properties include enabled, color, opacity, position, and weight.</td> <td>object</td> </tr> <tr> <td>properties</td> <td>Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	center	Circle center configuration.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate.	value: numeric	lng	Longitude coordinate.	value: numeric		fill	Circle fill configuration. Default properties include enabled, color, and opacity.	object	stroke	Stroke configuration. Default properties include enabled, color, opacity, position, and weight.	object	properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	object																														
Name	Description	Property Type																																																								
center	Circle center configuration.	object																																																								
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate.	value: numeric	lng	Longitude coordinate.	value: numeric																																																
Name	Description	Property Type																																																								
lat	Latitudinal coordinate.	value: numeric																																																								
lng	Longitude coordinate.	value: numeric																																																								
fill	Circle fill configuration. Default properties include enabled, color, and opacity.	object																																																								
stroke	Stroke configuration. Default properties include enabled, color, opacity, position, and weight.	object																																																								
properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	object																																																								
radius	Sets the radius of this circle in meters.	value: numeric																																																								
clickable	Indicates whether this circle handles mouse events.	value: boolean																																																								
draggable	If set to true, the user can drag this circle over the map.	value: boolean																																																								
editable	If set to true, the user can edit this circle by dragging the control points shown at the center and around the circumference of the circle.	value: boolean																																																								

fill	Circle fill configuration. Default properties include enabled, color, and opacity.	object
stroke	Stroke configuration. Default properties include enabled, color, opacity, position, and weight.	object
zIndex	The z-index compared to other polys.	value: string
properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	object

polygon

Allows the map to draw arbitrary shapes at specified coordinates. An array of polygon layers.

object

Name	Description	Property Type																											
name	A unique name which is used to distinguish this element from others. This name is provided to any applicable interaction event handlers, such as click handlers.	value: string																											
enabled	Enable/disable rendering of the polygon layer.	value: boolean																											
visible	If true, the polygon is visible.	value: boolean																											
polygons	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>points</td> <td>Sets the path for this polygon. The ordered sequence of coordinates that designates a closed loop. Unlike polylines, a polygon may consist of one or more paths. As a result, the paths property may specify one or more arrays of LatLng coordinates. Paths are closed automatically.</td> <td></td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>fill</td> <td>Polygon fill configuration. Default properties include enabled, color, and opacity.</td> <td>object</td> </tr> <tr> <td>stroke</td> <td>Stroke configuration. Default properties include enabled, color, opacity, position, and weight.</td> <td>object</td> </tr> <tr> <td>properties</td> <td>The z-index compared to other polys.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	points	Sets the path for this polygon. The ordered sequence of coordinates that designates a closed loop. Unlike polylines, a polygon may consist of one or more paths. As a result, the paths property may specify one or more arrays of LatLng coordinates. Paths are closed automatically.			<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate.	value: numeric	lng	Longitude coordinate.	value: numeric		fill	Polygon fill configuration. Default properties include enabled, color, and opacity.	object	stroke	Stroke configuration. Default properties include enabled, color, opacity, position, and weight.	object	properties	The z-index compared to other polys.	value: string	
	Name	Description	Property Type																										
	points	Sets the path for this polygon. The ordered sequence of coordinates that designates a closed loop. Unlike polylines, a polygon may consist of one or more paths. As a result, the paths property may specify one or more arrays of LatLng coordinates. Paths are closed automatically.																											
		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate.	value: numeric	lng	Longitude coordinate.	value: numeric																		
	Name	Description	Property Type																										
	lat	Latitudinal coordinate.	value: numeric																										
	lng	Longitude coordinate.	value: numeric																										
	fill	Polygon fill configuration. Default properties include enabled, color, and opacity.	object																										
	stroke	Stroke configuration. Default properties include enabled, color, opacity, position, and weight.	object																										
	properties	The z-index compared to other polys.	value: string																										
clickable	Indicates whether this polygon handles mouse events.	value: boolean																											
draggable	If set to true, the user can drag this polygon over the map.	value: boolean																											
editable	If set to true, the user can edit this polygon by dragging the control points shown at the center and around the circumference of the circle.	value: boolean																											
fill	Polygon fill configuration. Default properties include enabled, color, and opacity.	object																											
geodesic	When true, edges of the polygon are interpreted as geodesic and will follow the curvature of the Earth. When false, edges of the polygon are rendered as straight lines in screen space. Note that the shape of a geodesic polygon may appear to change when dragged, as the dimensions are maintained relative to the surface of the Earth.	value: boolean																											
stroke	Stroke configuration. Default properties include enabled, color, opacity, position, and weight.	object																											
zIndex	The z-index compared to other polys.	value: string																											
properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	object																											

polyline

An array of polyline layers.

object

Name	Description	Property Type

name	A unique name which is used to distinguish this element from others. This name is provided to any applicable interaction event handlers, such as click handlers.	value: string																																	
enabled	Enable/disable rendering of the polyline layer.	value: boolean																																	
visible	If true, the polyline is visible.	value: boolean																																	
polylines																																			
points	<p>Sets the path for this polyline. The ordered sequence of coordinates of the Polyline. This path may be specified using either a simple array of LatLngs, or an MVCArray of LatLngs. Note that if you pass a simple array, it will be converted to an MVCArray intersterring or removing LatLngs in the MVCArray will automatically update the polyline on the map.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate.	value: numeric	lng	Longitude coordinate.	value: numeric	object																								
Name	Description	Property Type																																	
lat	Latitudinal coordinate.	value: numeric																																	
lng	Longitude coordinate.	value: numeric																																	
icons	<p>The icons to be rendered along the polyline.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>fixedRotation</td> <td>If true, each icon in the sequence has the same fixed rotation regardless of the angle of the edge on which it lies. If false, case each icon in the sequence is rotated to align with its edge.</td> <td>value: boolean</td> </tr> <tr> <td>icon</td> <td>The icon to render on the line.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>The symbol's path, which is a built-in symbol path, or a custom path expressed using SVG path notation.</td> <td>value: string</td> </tr> <tr> <td>anchor</td> <td>The position of the symbol relative to the polyline. The coordinates of the symbol's path are translated left and up by the anchor's x and y coordinates respectively.</td> <td>object</td> </tr> <tr> <td>fill</td> <td>Icon fill configuration. Default properties include enabled, color, and opacity.</td> <td>object</td> </tr> <tr> <td>rotation</td> <td>The angle by which to rotate the symbol, expressed clockwise in degrees. A symbol in an IconSequence where fixedRotation is false is rotated relative to the angle of the edge on which it lies.</td> <td>value: numeric</td> </tr> <tr> <td>scale</td> <td>The amount by which the symbol is scaled in size. Defaults to the stroke weight of the polyline. After scaling, the symbol must lie inside a square 22 pixels in size centered at the symbol's anchor.</td> <td>value: string</td> </tr> <tr> <td>stroke</td> <td>Icon stroke configuration.</td> <td>object</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	fixedRotation	If true, each icon in the sequence has the same fixed rotation regardless of the angle of the edge on which it lies. If false, case each icon in the sequence is rotated to align with its edge.	value: boolean	icon	The icon to render on the line.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>The symbol's path, which is a built-in symbol path, or a custom path expressed using SVG path notation.</td> <td>value: string</td> </tr> <tr> <td>anchor</td> <td>The position of the symbol relative to the polyline. The coordinates of the symbol's path are translated left and up by the anchor's x and y coordinates respectively.</td> <td>object</td> </tr> <tr> <td>fill</td> <td>Icon fill configuration. Default properties include enabled, color, and opacity.</td> <td>object</td> </tr> <tr> <td>rotation</td> <td>The angle by which to rotate the symbol, expressed clockwise in degrees. A symbol in an IconSequence where fixedRotation is false is rotated relative to the angle of the edge on which it lies.</td> <td>value: numeric</td> </tr> <tr> <td>scale</td> <td>The amount by which the symbol is scaled in size. Defaults to the stroke weight of the polyline. After scaling, the symbol must lie inside a square 22 pixels in size centered at the symbol's anchor.</td> <td>value: string</td> </tr> <tr> <td>stroke</td> <td>Icon stroke configuration.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	path	The symbol's path, which is a built-in symbol path, or a custom path expressed using SVG path notation.	value: string	anchor	The position of the symbol relative to the polyline. The coordinates of the symbol's path are translated left and up by the anchor's x and y coordinates respectively.	object	fill	Icon fill configuration. Default properties include enabled, color, and opacity.	object	rotation	The angle by which to rotate the symbol, expressed clockwise in degrees. A symbol in an IconSequence where fixedRotation is false is rotated relative to the angle of the edge on which it lies.	value: numeric	scale	The amount by which the symbol is scaled in size. Defaults to the stroke weight of the polyline. After scaling, the symbol must lie inside a square 22 pixels in size centered at the symbol's anchor.	value: string	stroke	Icon stroke configuration.	object		object
Name	Description	Property Type																																	
fixedRotation	If true, each icon in the sequence has the same fixed rotation regardless of the angle of the edge on which it lies. If false, case each icon in the sequence is rotated to align with its edge.	value: boolean																																	
icon	The icon to render on the line.	object																																	
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>The symbol's path, which is a built-in symbol path, or a custom path expressed using SVG path notation.</td> <td>value: string</td> </tr> <tr> <td>anchor</td> <td>The position of the symbol relative to the polyline. The coordinates of the symbol's path are translated left and up by the anchor's x and y coordinates respectively.</td> <td>object</td> </tr> <tr> <td>fill</td> <td>Icon fill configuration. Default properties include enabled, color, and opacity.</td> <td>object</td> </tr> <tr> <td>rotation</td> <td>The angle by which to rotate the symbol, expressed clockwise in degrees. A symbol in an IconSequence where fixedRotation is false is rotated relative to the angle of the edge on which it lies.</td> <td>value: numeric</td> </tr> <tr> <td>scale</td> <td>The amount by which the symbol is scaled in size. Defaults to the stroke weight of the polyline. After scaling, the symbol must lie inside a square 22 pixels in size centered at the symbol's anchor.</td> <td>value: string</td> </tr> <tr> <td>stroke</td> <td>Icon stroke configuration.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	path	The symbol's path, which is a built-in symbol path, or a custom path expressed using SVG path notation.	value: string	anchor	The position of the symbol relative to the polyline. The coordinates of the symbol's path are translated left and up by the anchor's x and y coordinates respectively.	object	fill	Icon fill configuration. Default properties include enabled, color, and opacity.	object	rotation	The angle by which to rotate the symbol, expressed clockwise in degrees. A symbol in an IconSequence where fixedRotation is false is rotated relative to the angle of the edge on which it lies.	value: numeric	scale	The amount by which the symbol is scaled in size. Defaults to the stroke weight of the polyline. After scaling, the symbol must lie inside a square 22 pixels in size centered at the symbol's anchor.	value: string	stroke	Icon stroke configuration.	object													
Name	Description	Property Type																																	
path	The symbol's path, which is a built-in symbol path, or a custom path expressed using SVG path notation.	value: string																																	
anchor	The position of the symbol relative to the polyline. The coordinates of the symbol's path are translated left and up by the anchor's x and y coordinates respectively.	object																																	
fill	Icon fill configuration. Default properties include enabled, color, and opacity.	object																																	
rotation	The angle by which to rotate the symbol, expressed clockwise in degrees. A symbol in an IconSequence where fixedRotation is false is rotated relative to the angle of the edge on which it lies.	value: numeric																																	
scale	The amount by which the symbol is scaled in size. Defaults to the stroke weight of the polyline. After scaling, the symbol must lie inside a square 22 pixels in size centered at the symbol's anchor.	value: string																																	
stroke	Icon stroke configuration.	object																																	

		Default properties include enabled, color, opacity, position, and weight.	
	offset	The distance from the start of the line at which an icon is to be rendered. This distance may be expressed as a percentage of line's length (%) or in pixels (px).	value: numeric
	repeat	The distance between consecutive icons on the line. This distance may be expressed as a percentage of the line's length (%) or in pixels (px). To disable repeating of the icon, specify 0.	value: numeric
	stroke	Stroke configuration. Default properties include enabled, color, opacity, position, and weight.	object
	properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	object
clickable	Indicates whether this polyline handles mouse events.		value: boolean
draggable	If set to true, the user can drag this polyline over the map.		value: boolean
editable	If set to true, the user can edit this polyline by dragging the control points shown.		value: boolean
geodesic	When true, edges of the polygon are interpreted as geodesic and will follow the curvature of the Earth. When false, edges of the polygon are rendered as straight lines in screen space. Note that the shape of a geodesic polygon may appear to change when dragged, as the dimensions are maintained relative to the surface of the Earth.		value: boolean
stroke	Stroke configuration. Default properties include enabled, color, opacity, position, and weight.		object
zIndex	The zIndex compared to other polys.		value: string
properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.		object

rectangle

An array of rectangle layers.

Name	Description	Property Type																											
name	A unique name which is used to distinguish this element from others. This name is provided to any applicable interaction event handlers, such as click handlers.	value: string																											
enabled	Enable/disable rendering of the rectangle layer.	value: boolean																											
visible	If true, the rectangle is visible.	value: boolean																											
rectangles	Sets rectangle configuration.	object																											
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>bounds</td> <td>Sets the bounds of the rectangle.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>corner1</td> <td>South-West corner of the rectangle.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	bounds	Sets the bounds of the rectangle.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>corner1</td> <td>South-West corner of the rectangle.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	corner1	South-West corner of the rectangle.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate.	value: numeric	lng	Longitude coordinate.	value: numeric			
Name	Description	Property Type																											
bounds	Sets the bounds of the rectangle.	object																											
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>corner1</td> <td>South-West corner of the rectangle.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	corner1	South-West corner of the rectangle.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate.	value: numeric	lng	Longitude coordinate.	value: numeric											
Name	Description	Property Type																											
corner1	South-West corner of the rectangle.	object																											
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate.	value: numeric	lng	Longitude coordinate.	value: numeric																			
Name	Description	Property Type																											
lat	Latitudinal coordinate.	value: numeric																											
lng	Longitude coordinate.	value: numeric																											

object

	<table border="1"> <tr> <td>corner2</td> <td>North-East corner of the rectangle.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> </table>	corner2	North-East corner of the rectangle.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate.	value: numeric	lng	Longitude coordinate.	value: numeric		
corner2	North-East corner of the rectangle.	object															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude coordinate.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate.	value: numeric	lng	Longitude coordinate.	value: numeric							
Name	Description	Property Type															
lat	Latitudinal coordinate.	value: numeric															
lng	Longitude coordinate.	value: numeric															
	fill	Rectangle fill configuration. Default properties include enabled, color, and opacity.	object														
	stroke	Stroke configuration. Default properties include enabled, color, opacity, position, and weight.	object														
	properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	object														
	clickable	Indicates whether this rectangle handles mouse events.	value: boolean														
	draggable	If set to true, the user can drag this rectangle over the map.	value: boolean														
	editable	If set to true, the user can edit this rectangle by dragging the control points shown.	value: boolean														
	fill	Rectangle fill configuration. Default properties include enabled, color, and opacity.	object														
	stroke	Stroke configuration. Default properties include enabled, color, opacity, position, and weight.	object														
	zIndex	The z-index compared to other polys.	value: string														
	properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	object														
bicycling	Shows bicycle route information.		object														
	Name	Description	Property Type														
	name	A unique name which is used to distinguish this element from others. This name is provided to any applicable interaction event handlers, such as click handlers.	value: string														
	enabled	If true, displays bike lanes and paths and demotes large roads.	value: boolean														
	properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	object														
traffic	Shows traffic information		object														
	Name	Description	Property Type														
	name	A unique name which is used to distinguish this element from others. This name is provided to any applicable interaction event handlers, such as click handlers.	value: string														
	enabled	If true, displays current road traffic.	value: boolean														
	autoRefresh	Whether the traffic layer refreshes with updated information automatically.	value: boolean														
	properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	object														
transit	Shows transit information.		object														
	Name	Description	Property Type														

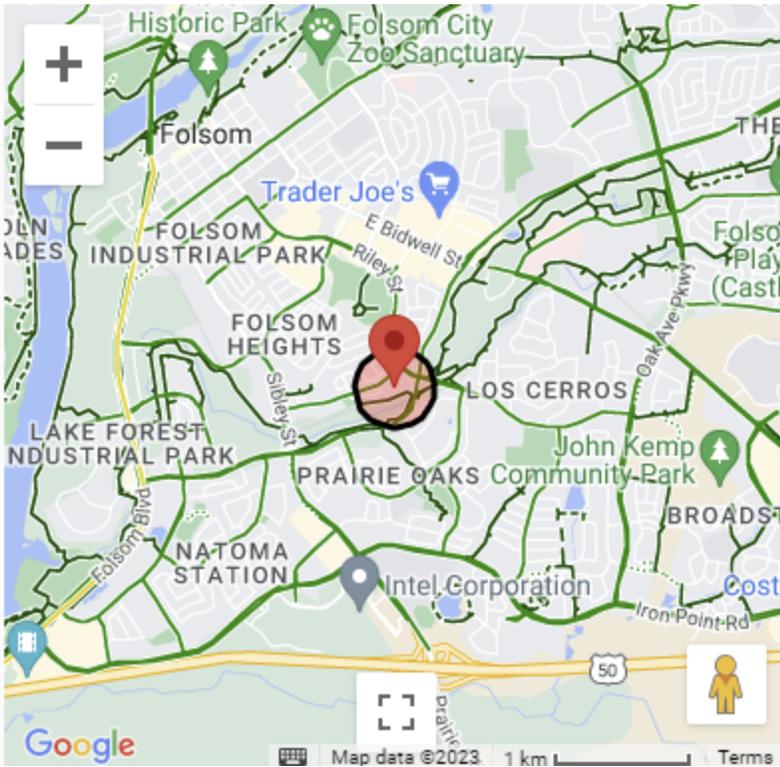
	<table border="1"> <tr> <td>name</td> <td>A unique name which is used to distinguish this element from others. This name is provided to any applicable interaction event handlers, such as click handlers.</td> <td>value: string</td> </tr> <tr> <td>enabled</td> <td>If true, enables layer that displays transit lines.</td> <td>value: boolean</td> </tr> <tr> <td>properties</td> <td>Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.</td> <td>object</td> </tr> </table>	name	A unique name which is used to distinguish this element from others. This name is provided to any applicable interaction event handlers, such as click handlers.	value: string	enabled	If true, enables layer that displays transit lines.	value: boolean	properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	object																						
name	A unique name which is used to distinguish this element from others. This name is provided to any applicable interaction event handlers, such as click handlers.	value: string																														
enabled	If true, enables layer that displays transit lines.	value: boolean																														
properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	object																														
kml	Keyhole Markup Language (KML) configuration to specify a set of geographic visualizations.	object																														
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>A unique name which is used to distinguish this element from others. This name is provided to any applicable interaction event handlers, such as click handlers.</td> <td>value: string</td> </tr> <tr> <td>enabled</td> <td>Enable/disable rendering of a KML layer.</td> <td>value: boolean</td> </tr> <tr> <td>url</td> <td>The URL of the KML document to display</td> <td>object</td> </tr> <tr> <td>clickable</td> <td>If true, the layer receives mouse events</td> <td>value: boolean</td> </tr> <tr> <td>preserveViewport</td> <td>If this option is set to true or if the map's center and zoom were never set, the input map is centered and zoomed to the bounding box of the contents of the layer.</td> <td>value: boolean</td> </tr> <tr> <td>screenOverlays</td> <td>Whether to render the screen overlays.</td> <td>value: boolean</td> </tr> <tr> <td>suppressInfoWindows</td> <td>Suppress the rendering of info windows when layer features are clicked. Default is null.</td> <td>value: string</td> </tr> <tr> <td>zIndex</td> <td>The z-index of the layer. Default is null.</td> <td>value: numeric</td> </tr> <tr> <td>properties</td> <td>Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	name	A unique name which is used to distinguish this element from others. This name is provided to any applicable interaction event handlers, such as click handlers.	value: string	enabled	Enable/disable rendering of a KML layer.	value: boolean	url	The URL of the KML document to display	object	clickable	If true, the layer receives mouse events	value: boolean	preserveViewport	If this option is set to true or if the map's center and zoom were never set, the input map is centered and zoomed to the bounding box of the contents of the layer.	value: boolean	screenOverlays	Whether to render the screen overlays.	value: boolean	suppressInfoWindows	Suppress the rendering of info windows when layer features are clicked. Default is null.	value: string	zIndex	The z-index of the layer. Default is null.	value: numeric	properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	object	
Name	Description	Property Type																														
name	A unique name which is used to distinguish this element from others. This name is provided to any applicable interaction event handlers, such as click handlers.	value: string																														
enabled	Enable/disable rendering of a KML layer.	value: boolean																														
url	The URL of the KML document to display	object																														
clickable	If true, the layer receives mouse events	value: boolean																														
preserveViewport	If this option is set to true or if the map's center and zoom were never set, the input map is centered and zoomed to the bounding box of the contents of the layer.	value: boolean																														
screenOverlays	Whether to render the screen overlays.	value: boolean																														
suppressInfoWindows	Suppress the rendering of info windows when layer features are clicked. Default is null.	value: string																														
zIndex	The z-index of the layer. Default is null.	value: numeric																														
properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	object																														

Scripting

See the [Perspective - Google Map Scripting](#) page for the full list of scripting functions available for this component.

Example

The example below incorporates multiple layers to display bike paths and mark a location at the center of an editable circle using layers.



Property	Value		Description										
layers.marker.0	<table border="1"> <thead> <tr> <th>Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>markers.0.position.lat</td> <td>38.660867</td> </tr> <tr> <td>markers.0.position.lng</td> <td>-121.159728</td> </tr> <tr> <td>markers.0.icon.type</td> <td>default</td> </tr> </tbody> </table>		Property	Value	markers.0.position.lat	38.660867	markers.0.position.lng	-121.159728	markers.0.icon.type	default	Default marker icon		
Property	Value												
markers.0.position.lat	38.660867												
markers.0.position.lng	-121.159728												
markers.0.icon.type	default												
layers.circle.0	<table border="1"> <thead> <tr> <th>Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>circles.0.center.lat</td> <td>38.660867</td> </tr> <tr> <td>circles.0.center.lng</td> <td>-121.159728</td> </tr> <tr> <td>circles.0.fill</td> <td>#FF0000</td> </tr> <tr> <td>radius</td> <td>300</td> </tr> </tbody> </table>		Property	Value	circles.0.center.lat	38.660867	circles.0.center.lng	-121.159728	circles.0.fill	#FF0000	radius	300	Red circle layer centered at the marker icon location
Property	Value												
circles.0.center.lat	38.660867												
circles.0.center.lng	-121.159728												
circles.0.fill	#FF0000												
radius	300												
layers.bicycling	<table border="1"> <thead> <tr> <th>Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>true</td> </tr> </tbody> </table>		Property	Value	enabled	true	Visible bicycle routes						
Property	Value												
enabled	true												

Perspective - Google Map Scripting

This page details the various scripting, component, and extension functions available for Perspective's Google Map component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

MapBounds Events

onMapBoundsChanged

Called when the viewport bounds of the map have changed. Returns the north, south, east, and west bound points.

Object Path	Type	Description
event.north	Numeric	The latitude value for the Northern boundary.
event.east	Numeric	The longitude value for the Eastern boundary.
event.south	Numeric	The latitude value for the Southern boundary.
event.west	Numeric	The longitude value for the Western boundary.

onMapCenterChanged

Called when the center of the map has changed. Returns the new map center as lat and lng.

Object Path	Type	Description
event.lat	Numeric	The latitude value for the center of the map after being changed.
event.lng	Numeric	The longitude value for the center of the map after being changed.

Map Events

onMapClick

Interaction event fired when the map is clicked. Returns the lat and lng of the mouse click as it translates on the map.

Object Path	Type	Description
event.lat	Numeric	The latitude of where on the map the user clicked.
event.lng	Numeric	The longitude of where on the map the user clicked.

onMapDoubleClick

Interaction event fired when the map is double-clicked. Returns the lat and lng of the mouse double-click as it translates on the map.

Object Path	Type	Description
event.lat	Numeric	The latitude of where on the map the user double-clicked.
event.lng	Numeric	The longitude of where on the map the user double-clicked.

onMapDrag

On this page ...

- [Component Events](#)
 - [MapBounds Events](#)
 - [Map Events](#)
 - [BicyclingLayer Events](#)
 - [Circle Events](#)
 - [GroundOverlay Events](#)
 - [KmlLayer Events](#)
 - [Marker Events](#)
 - [OverlayView Events](#)
 - [Polygon Events](#)
 - [Polyline Events](#)
 - [Rectangle Events](#)
 - [TrafficLayer Events](#)
 - [TransitLayer Events](#)
- [Component Functions](#)
 - [fitBounds](#)
 - [panBy](#)
 - [panTo](#)
 - [panToBounds](#)
 - [setCenter](#)
 - [setClickableIcons](#)
 - [setHeading](#)
 - [setMapTypeId](#)
 - [setTilt](#)
 - [setZoom](#)

Interaction event fired while the user drags the map. Returns the lat and lng of the map center while moving.

Object Path	Type	Description
event.lat	Numeric	The latitude of the map center while the user moves the map.
event.lng	Numeric	The longitude of the map center while the user moves the map.

onMapDragEnd

Interaction event fired when the user stops dragging the map. Returns the lat and lng of the new map center.

Object Path	Type	Description
event.lat	Numeric	The latitude of the map center after the map movement is over.
event.lng	Numeric	The longitude of the map center after the map movement is over.

onMapDragStart

Interaction event fired when the user starts dragging the map. Returns the lat and lng of the starting map center.

Object Path	Type	Description
event.lat	Numeric	The latitude of the map center when map dragging begins.
event.lng	Numeric	The longitude of the map center when map dragging begins.

onMapHeadingChanged

Interaction event fired when the map heading property changes. Returns the new map heading value.

Object Path	Type	Description
event.heading	Numeric	The heading value of the map.

onMapMouseMove

Interaction event fired when the user's mouse moves over the map container. Returns the lat and lng of the mouse location on the map.

Object Path	Type	Description
event.lat	Numeric	The latitude of the mouse location on the map.
event.lng	Numeric	The longitude of the mouse location on the map.

onMapMouseOut

Interaction event fired when the user's mouse exits the map container.

Object Path	Type	Description
event.lat	Numeric	The latitude of the mouse location when exiting the map.
event.lng	Numeric	The longitude of the mouse location exiting the map.

onMapMouseOver

Interaction event fired when the user's mouse enters the map container.

Object Path	Type	Description
event.lat	Numeric	The latitude of the mouse location when entering the map.
event.lng	Numeric	The longitude of the mouse location entering the map.

onMapResize

Interaction event fired when the map size has changed. Returns the width and height of the new map size.

Object Path	Type	Description
event.width	Numeric	The width value of the map.
event.height	Numeric	The height value of the map.

onMapRightClick

Interaction event fired when the DOM contextmenu event is fired on the map container.

Object Path	Type	Description
event.lat	Numeric	The latitude of the mouse location.
event.lng	Numeric	The longitude of the mouse location.

onMapTiltChanged

Interaction event fired when the map tilt property changes. Returns the new tilt value.

Object Path	Type	Description
event.tilt	Numeric	The tilt value of the map. Value will be null, 0, or 45.

onMapTypeIdChanged

Interaction event fired when the map type ID property changes. Returns the new map type ID value.

Object Path	Type	Description
event.mapTypeId	String	The mapTypeId value of the map.

onMapZoomChanged

Interaction event fired when the map zoom property changes. Returns the new zoom level.

Object Path	Type	Description
event.zoom	Numeric	The zoom level of the map view.

BicyclingLayer Events

onBicyclingLayerLoad

Interaction event fired when the bicycling layer loads.

Object Path	Type	Description
event.name	String	The name of the bicycling layer.

onBicyclingLayerUnmount

Interaction event fired when the bicycling layer unloads.

Object Path	Type	Description
event.name	String	The name of the bicycling layer.

Circle Events

onCircleCenterChanged

Interaction event fired when the center of the circle changes.

Object Path	Type	Description
event.name	String	The name of the circle.
event.lat	Numeric	The latitude value of the center of the circle
event.lng	Numeric	The longitude value of the center of the circle

onCircleClick

Interaction event fired when the circle is clicked.

Object Path	Type	Description
event.name	String	The name of the circle.
event.lat	Numeric	The latitude of where on the circle the user clicked.
event.lng	Numeric	The longitude of where on the circle the user clicked.

onCircleDoubleClick

Interaction event fired when the circle is double-clicked.

Object Path	Type	Description
event.name	String	The name of the circle.
event.lat	Numeric	The latitude of where on the circle the user double-clicked.
event.lng	Numeric	The longitude of where on the circle the user double-clicked.

onCircleDrag

Interaction event repeatedly fired while the user drags the circle.

Object Path	Type	Description
event.name	String	The name of the circle.
event.lat	Numeric	The latitude of the map center while the user moves the map.
event.lng	Numeric	The longitude of the map center while the user moves the map.

onCircleDragEnd

Interaction event fired when the user stops dragging the circle.

Object Path	Type	Description
event.lat	Numeric	The latitude of the map center while the user moves the map.
event.lng	Numeric	The longitude of the map center while the user moves the map.

onCircleDragStart

Interaction event fired when the user starts dragging the circle.

Object Path	Type	Description
event.name	String	The name of the circle.
event.lat	Numeric	The latitude of the map center when map dragging begins.
event.lng	Numeric	The longitude of the map center when map dragging begins.

onCircleLoad

Interaction event fired when the circle layer loads.

Object Path	Type	Description
event.name	String	The name of the circle layer.

onCircleMouseDown

Interaction event fired on the mousedown of the circle

Object Path	Type	Description
event.name	String	The name of the circle.
event.lat	Numeric	The latitude of the mouse location on the circle layer.
event.lng	Numeric	The longitude of the mouse location on the circle layer.

onCircleMouseMove

Interaction event fired when the mousemove of the circle

Object Path	Type	Description
event.name	String	The name of the circle.
event.lat	Numeric	The latitude of the mouse location on the circle layer.
event.lng	Numeric	The longitude of the mouse location on the circle layer.

onCircleMouseOut

Interaction event fired when the mouseout of the circle

Object Path	Type	Description
event.name	String	The name of the circle.
event.lat	Numeric	The latitude of the mouse location on the circle layer.
event.lng	Numeric	The longitude of the mouse location on the circle layer.

onCircleMouseOver

Interaction event fired when the mouseover of the circle

Object Path	Type	Description
event.name	String	The name of the circle.
event.lat	Numeric	The latitude of the mouse location on the circle layer.
event.lng	Numeric	The longitude of the mouse location on the circle layer.

onCircleMouseUp

Interaction event fired when the mouseup of the circle

Object Path	Type	Description
event.name	String	The name of the circle.
event.lat	Numeric	The latitude of the mouse location on the circle layer.
event.lng	Numeric	The longitude of the mouse location on the circle layer.

onCircleRadiusChanged

Interaction event fired when the radius of the circle is changed

Object Path	Type	Description
-------------	------	-------------

event.name	String	The name of the circle.
event.radius	Numeric	The radius of the circle.

onCircleRightClick

Interaction event fired on right-click of the circle layer.

Object Path	Type	Description
event.name	String	The name of the circle.
event.lat	Numeric	The latitude of the mouse location on the circle layer.
event.lng	Numeric	The longitude of the mouse location on the circle layer.

onCircleUnmount

Interaction event fired when the circle unloads.

Object Path	Type	Description
event.name	String	The name of the circle.

GroundOverlay Events

onGroundOverlayClick

Interaction event fired when the ground overlay is clicked.

Object Path	Type	Description
event.name	String	The name of the ground overlay layer.
event.lat	Numeric	The latitude of where on the ground overlay the user clicked.
event.lng	Numeric	The longitude of where on the ground overlay the user clicked.

onGroundOverlayDoubleClick

Interaction event fired when the ground overlay is double-clicked.

Object Path	Type	Description
event.name	String	The name of the ground overlay layer.
event.lat	Numeric	The latitude of where on the ground overlay the user double-clicked.
event.lng	Numeric	The longitude of where on the ground overlay the user double-clicked.

onGroundOverlayLoad

Interaction event fired when the ground overlay layer is loaded.

Object Path	Type	Description
event.name	String	The name of the ground overlay layer.

onGroundOverlayUnmount

Interaction event fired when the ground overlay layer unloads.

Object Path	Type	Description
event.name	String	The name of the ground overlay layer.

KmlLayer Events

OnKmlLayerClick

Interaction event fired when a feature in the KML layer is clicked.

Object Path	Type	Description
event.name	String	The name of the KML layer.
event.featuredData	Object	Contains information about the clicked feature
event.lat	Numeric	The latitude of where the information pop-up window is anchored on the KML layer.
event.lng	Numeric	The longitude of where the information pop-up window is anchored on the KML layer.
event.pixelOffset	Object	The offset to apply to the anchored information pop-up window on the clicked feature.

OnKmlLayerDefaultViewportChanged

Interaction event fired when the KML layer default viewport has changed.

Object Path	Type	Description
event.name	String	The name of the KML layer.

OnKmlLayerLoad

Interaction event fired when the KML layer loads.

Object Path	Type	Description
event.name	String	The name of the KML layer.

OnKmlLayerStatusChanged

Interaction event fired when the KML layer has finished loading.

Object Path	Type	Description
event.name	String	The name of the KML layer.

OnKmlLayerUnmount

Interaction event fired when the KML layer unloads.

Object Path	Type	Description
event.name	String	The name of the KML layer.

Marker Events

onMarkerClick

Interaction event fired when the marker is clicked. Returns the lat and lng of the marker click.

Object Path	Type	Description
event.name	String	The name of the marker.
event.lat	Numeric	The latitude of the marker click.
event.lng	Numeric	The longitude of the marker click.

onMarkerDoubleClick

Interaction event fired when the marker is double-clicked. Returns the lat and lng of the marker double-click.

Object Path	Type	Description
event.name	String	The name of the marker.
event.lat	Numeric	The latitude of where the marker was double-clicked.
event.lng	Numeric	The longitude of where the marker was double-clicked.

onMarkerDrag

Interaction event repeatedly fired while the user drags the marker. Returns the lat and lng of the marker while moving.

Object Path	Type	Description
event.name	String	The name of the marker.
event.lat	Numeric	The latitude of the marker while being dragged.
event.lng	Numeric	The longitude of the marker while being dragged.

onMarkerDragEnd

Interaction event fired when the user stops dragging the marker. Returns the lat and lng of the new marker position.

Object Path	Type	Description
event.name	String	The name of the marker.
event.lat	Numeric	The latitude of the marker after the marker movement is over.
event.lng	Numeric	The longitude of the marker after the marker movement is over.

onMarkerDragStart

Interaction event fired when the user starts dragging the marker. Returns the lat and lng of the starting marker position.

Object Path	Type	Description
event.name	String	The name of the marker.
event.lat	Numeric	The latitude of the marker when the marker dragging begins.
event.lng	Numeric	The longitude of the marker when the marker dragging begins.

onMarkerLoad

Interaction event fired when the marker layer loads.

Object Path	Type	Description
event.name	String	The name of the marker layer.

onMarkerMouseDown

Interaction event fired on the mousedown of the marker.

Object Path	Type	Description
event.name	String	The name of the marker.
event.lat	Numeric	The latitude of the marker location.
event.lng	Numeric	The longitude of the marker location.

onMarkerMouseOut

Interaction event fired on mouseout of the marker.

Object Path	Type	Description
event.name	String	The name of the marker.
event.lat	Numeric	The latitude of the event.
event.lng	Numeric	The longitude of the event.

onMarkerMouseOver

Interaction event fired on mouseover of the marker.

Object Path	Type	Description
event.name	String	The name of the marker.
event.lat	Numeric	The latitude of the event.
event.lng	Numeric	The longitude of the event.

onMarkerMouseUp

Interaction event fired when the mouseup of the marker.

Object Path	Type	Description
event.name	String	The name of the marker.
event.lat	Numeric	The latitude of the event.
event.lng	Numeric	The longitude of the event.

onMarkerPositionChanged

Interaction event fired when the position of the marker changes.

Object Path	Type	Description
event.name	String	The name of the marker.
event.lat	Numeric	The latitude of the new marker position.
event.lng	Numeric	The longitude of the new marker position.

onMarkerRightClick

Interaction event fired on the right-click of the marker.

Object Path	Type	Description
event.name	String	The name of the marker.
event.lat	Numeric	The latitude of the event location.
event.lng	Numeric	The longitude of the event location.

onMarkerUnmount

Interaction event fired when the marker layer unloads.

Object Path	Type	Description
event.name	String	The name of the marker layer.

onMarkerPopupOpened

Interaction event fired when a marker popup opens.

Object Path	Type	Description
-------------	------	-------------

event.name	String	The name of the marker layer.
------------	--------	-------------------------------

onMarkerPopupClosed

Interaction event fired when a marker popup closes.

Object Path	Type	Description
event.name	String	The name of the marker layer.

OverlayView Events

onOverlayViewLoad

Interaction event fired when the OverlayView layer loads.

Object Path	Type	Description
event.name	String	The name of the OverView layer.

onOverlayViewUnmount

Interaction event fired when the OverlayView layer unloads.

Object Path	Type	Description
event.name	String	The name of the OverView layer.

Polygon Events

onPolygonClick

Interaction event fired when the polygon is clicked. Returns the lat and lng of the mouse click as it translates on the polygon later

Object Path	Type	Description
event.name	String	The name of the polygon layer.
event.lat	Numeric	The latitude of where on the map the user clicked.
event.lng	Numeric	The longitude of where on the map the user clicked.

onPolygonDoubleClick

Interaction event fired when the polygon is double-clicked. Returns the lat and lng of the mouse double-click as it translates on the polygon layer.

Object Path	Type	Description
event.name	String	The name of the polygon layer.
event.lat	Numeric	The latitude of where on the polygon the user double-clicked.
event.lng	Numeric	The longitude of where on the polygon the user double-clicked.

onPolygonDrag

Interaction event repeatedly fired while the user drags the polygon.

Object Path	Type	Description
event.name	String	The name of the polygon layer.
event.lat	Numeric	The latitude of the polygon while the user moves it.
event.lng	Numeric	The longitude of the polygon while the user moves it.

onPolygonDragEnd

Interaction event fired when the user stops dragging the polygon. Returns the lat and lng of the new polygon location.

Object Path	Type	Description
event.name	String	The name of the polygon layer.
event.lat	Numeric	The latitude of the polygon after the polygon movement is over.
event.lng	Numeric	The longitude of the polygon after the polygon movement is over.

onPolygonDragStart

Interaction event fired when the user starts dragging the polygon. Returns the lat and lng of the starting polygon location.

Object Path	Type	Description
event.name	String	The name of the polygon layer.
event.lat	Numeric	The latitude of the polygon when the users starts dragging the polygon.
event.lng	Numeric	The longitude of the polygon when the users starts dragging the polygon.

onPolygonLoad

Interaction event fired when the polygon layer loads.

Object Path	Type	Description
event.name	String	The name of the polygon layer.

onPolygonMouseDown

Interaction event fired on the mousedown of the polygon.

Object Path	Type	Description
event.name	String	The name of the polygon.
event.lat	Numeric	The latitude of the mouse location on the polygon layer.
event.lng	Numeric	The longitude of the mouse location on the polygon layer.

onPolygonMouseMove

Interaction event fired when the user's mouse moves over the polygon layer. Returns the lat and lng of the mouse location on the map.

Object Path	Type	Description
event.name	String	The name of the polygon layer.
event.lat	Numeric	The latitude of the mouse location on the map.
event.lng	Numeric	The longitude of the mouse location on the map.

onPolygonMouseOut

Interaction event fired when the user's mouse exits the polygon layer.

Object Path	Type	Description
event.name	String	The name of the polygon layer.
event.lat	Numeric	The latitude of the mouse location when exiting the map.
event.lng	Numeric	The longitude of the mouse location exiting the map.

onPolygonMouseOver

Interaction event fired when the user's mouse enters the polygon layer.

Object Path	Type	Description
event.name	String	The name of the polygon layer.
event.lat	Numeric	The latitude of the mouse location when entering the polygon.
event.lng	Numeric	The longitude of the mouse location entering the polygon.

onPolygonMouseUp

Interaction event fired when the mouseup of the polygon layer.

Object Path	Type	Description
event.name	String	The name of the polygon layer.
event.lat	Numeric	The latitude of the mouse location on the circle layer.
event.lng	Numeric	The longitude of the mouse location on the circle layer.

onPolygonRightClick

Interaction event fired when the DOM contextmenu event is fired on the polygon layer.

Object Path	Type	Description
event.lat	Numeric	The latitude of the mouse location.
event.lng	Numeric	The longitude of the mouse location.

onPolygonUnmount

Interaction event fired when the polygon layer unloads.

Object Path	Type	Description
event.name	String	The name of the polygon layer.

Polyline Events

onPolylineClick

Interaction event fired when the map is clicked. Returns the lat and lng of the mouse click as it translates on the map.

Object Path	Type	Description
event.lat	Numeric	The latitude of where on the polyline the user clicked.
event.lng	Numeric	The longitude of where on the polyline the user clicked.

onPolylineDoubleClick

Interaction event fired when the map is double-clicked. Returns the lat and lng of the mouse double-click as it translates on the polyline .

Object Path	Type	Description
event.lat	Numeric	The latitude of where on the polyline the user double-clicked.
event.lng	Numeric	The longitude of where on the polyline the user double-clicked.

onPolylineDrag

Interaction event repeatedly fired while the user drags the polyline.

Object Path	Type	Description
event.name	String	The name of the circle.
event.lat	Numeric	The latitude of the polyline while the user moves the map.
event.lng	Numeric	The longitude of the polyline while the user moves the map.

onPolylineDragEnd

Interaction event fired when the user stops dragging the polyline. Returns the lat and lng of the new polyline.

Object Path	Type	Description
event.lat	Numeric	The latitude of the polyline center after the map movement is over.
event.lng	Numeric	The longitude of the polyline center after the map movement is over.

onPolylineDragStart

Interaction event fired when the user starts dragging the polyline. Returns the lat and lng of the starting polyline center.

Object Path	Type	Description
event.lat	Numeric	The latitude of the polyline center when map dragging begins.
event.lng	Numeric	The longitude of the polyline center when map dragging begins.

onPolylineLoad

Interaction event fired when the polyline layer loads.

Object Path	Type	Description
event.name	String	The name of the polyline layer.

onPolylineMouseDown

Interaction event fired on the mousedown of the polyline.

Object Path	Type	Description
event.name	String	The name of the polyline.
event.lat	Numeric	The latitude of the mouse location on the polyline layer.
event.lng	Numeric	The longitude of the mouse location on the polyline layer.

onPolylineMouseMove

Interaction event fired when the user's mouse moves over the polyline layer. Returns the lat and lng of the mouse location on the polyline.

Object Path	Type	Description
event.lat	Numeric	The latitude of the mouse location on the polyline.
event.lng	Numeric	The longitude of the mouse location on the polyline.

onPolylineMouseOut

Interaction event fired when the user's mouse exits the polyline.

Object Path	Type	Description
event.lat	Numeric	The latitude of the mouse location when exiting the polyline.
event.lng	Numeric	The longitude of the mouse location exiting the polyline.

onPolylineMouseOver

Interaction event fired when the user's mouse enters the polyline.

Object Path	Type	Description
event.lat	Numeric	The latitude of the mouse location when entering the polyline.
event.lng	Numeric	The longitude of the mouse location entering the polyline.

onPolylineMouseUp

Interaction event fired when the mouseup of the polyline.

Object Path	Type	Description
event.name	String	The name of the circle.
event.lat	Numeric	The latitude of the mouse location on the polyline layer.
event.lng	Numeric	The longitude of the mouse location on the polyline layer.

onPolylineRightClick

Interaction event fired when the DOM contextmenu event is fired on the polyline.

Object Path	Type	Description
event.lat	Numeric	The latitude of the mouse location.
event.lng	Numeric	The longitude of the mouse location.

onPolylineUnmount

Interaction event fired when the polyline layer unloads.

Object Path	Type	Description
event.name	String	The name of the polyline.

Rectangle Events

onRectangleBoundsChanged

Called when the viewport bounds of the rectangle have changed. Returns the north, south, east, and west bound points.

Object Path	Type	Description
event.north	Numeric	The latitude value for the Northern boundary.
event.east	Numeric	The longitude value for the Eastern boundary.
event.south	Numeric	The latitude value for the Southern boundary.
event.west	Numeric	The longitude value for the Western boundary.

onRectangleClick

Interaction event fired when the rectangle is clicked. Returns the lat and lng of the mouse click.

Object Path	Type	Description
event.lat	Numeric	The latitude of where the user clicked.
event.lng	Numeric	The longitude of where the user clicked.

onRectangleDoubleClick

Interaction event fired when the rectangle is double-clicked. Returns the lat and lng of the mouse double-click as it translates on the map.

Object Path	Type	Description
event.lat	Numeric	The latitude of where the user double-clicked.
event.lng	Numeric	The longitude of where the user double-clicked.

onRectangleDrag

Interaction event repeatedly fired while the user drags the rectangle.

Object Path	Type	Description
event.name	String	The name of the rectangle.
event.lat	Numeric	The latitude of the rectangle center while the user moves the rectangle.
event.lng	Numeric	The longitude of the rectangle center while the user moves the rectangle.

onRectangleDragEnd

Interaction event fired when the user stops dragging the rectangle. Returns the lat and lng of the new rectangle center.

Object Path	Type	Description
event.name	String	The name of the rectangle.
event.lat	Numeric	The latitude of the rectangle center while the user moves the rectangle.
event.lng	Numeric	The longitude of the rectangle center while the user moves the rectangle.

onRectangleDragStart

Interaction event fired when the user starts dragging the rectangle. Returns the lat and lng of the starting rectangle center.

Object Path	Type	Description
event.name	String	The name of the rectangle.
event.lat	Numeric	The latitude of the rectangle center while the user moves the rectangle.
event.lng	Numeric	The longitude of the rectangle center while the user moves the rectangle.

onRectangleLoad

Interaction event fired when the rectangle layer loads.

Object Path	Type	Description
event.name	String	The name of the rectangle layer.

onRectangleMouseDown

Interaction event fired on the mousedown of the rectangle.

Object Path	Type	Description
event.name	String	The name of the rectangle.
event.lat	Numeric	The latitude of the mouse location on the rectangle layer.
event.lng	Numeric	The longitude of the mouse location on the rectangle layer.

onRectangleMouseMove

Interaction event fired when the user's mouse moves over the rectangle. Returns the lat and lng of the mouse location.

Object Path	Type	Description
event.name	String	The name of the rectangle.
event.lat	Numeric	The latitude of the mouse location on the rectangle layer.
event.lng	Numeric	The longitude of the mouse location on the rectangle layer.

onRectangleMouseOut

Interaction event fired when the user's mouse exits the rectangle.

Object Path	Type	Description
event.name	String	The name of the rectangle.
event.lat	Numeric	The latitude of the mouse location on the rectangle layer.
event.lng	Numeric	The longitude of the mouse location on the rectangle layer.

onRectangleMouseOver

Interaction event fired when the user's mouse enters the rectangle.

Object Path	Type	Description
event.name	String	The name of the rectangle.
event.lat	Numeric	The latitude of the mouse location on the rectangle layer.
event.lng	Numeric	The longitude of the mouse location on the rectangle layer.

onRectangleMouseUp

Interaction event fired when the mouseup of the rectangle.

Object Path	Type	Description
event.name	String	The name of the rectangle.
event.lat	Numeric	The latitude of the mouse location on the rectangle layer.
event.lng	Numeric	The longitude of the mouse location on the rectangle layer.

onRectangleRightClick

Interaction event fired when the DOM contextmenu event is fired on the rectangle.

Object Path	Type	Description
event.name	String	The name of the rectangle.
event.lat	Numeric	The latitude of the mouse location.
event.lng	Numeric	The longitude of the mouse location.

onRectangleUnmount

Interaction event fired when the KML layer unloads.

Object Path	Type	Description
event.name	String	The name of the rectangle.

TrafficLayer Events

OnTrafficLayerLoad

Interaction event fired when the traffic layer loads.

Object Path	Type	Description
event.name	String	The name of the traffic layer.

OnTrafficLayerUnmount

Interaction event fired when the traffic layer unloads.

Object Path	Type	Description
event.name	String	The name of the traffic layer.

TransitLayer Events

OnTransitLayerLoad

Interaction event fired when the transit layer loads.

Object Path	Type	Description
event.name	String	The name of the transit layer.

OnTransitLayerUnmount

Interaction event fired when the transit layer unloads.

Object Path	Type	Description
event.name	String	The name of the transit layer.

Component Functions

fitBounds

- Description
 - Sets the viewport to contain the given bounds. When the map is set to display none, the fitBounds function reads the map's size as 0x0 and does not do anything.
- Parameters
 - [Dictionary](#) latLngBounds - A dictionary consisting of two LatLng objects. The LatLng objects combined represent the geographical bounds the map view should be set to.
 - [Numeric](#) padding - Padding in pixels. The value represents the same padding for all four sides of the map. The bounds will be fit in the part of the map that remains after padding is removed. [optional]
- Return
 - None

panBy

- Description
 - Changes the center of the map by the given distance in pixels. If the distance is less than both the width and height of the map, the transition will be smoothly animated. Note that the map coordinate system increases from west to east (for x values) and north to south (for y values).
- Parameters
 - [Numeric](#) x - Number of pixels to move the map in the x direction.
 - [Numeric](#) y - Number of pixels to move the map in the y direction.
- Return
 - None

panTo

- Description
 - Pans the map to a given center. If the change is less than both the width and height of the map, the transition will be smoothly animated.
- Parameters
 - [Dictionary](#) latLng - The geographic point to pan to.
- Return
 - None

panToBounds

- Description
 - Pans the map by the minimum amount necessary to contain the given `LatLngBounds` so that the map will be panned to show as much of the bounds as possible inside `{currentMapSizeInPx}` - `{padding}`. The map's zoom, tilt, and heading will not be changed.
- Parameters
 - [Dictionary](#) `latLngBounds` - A dictionary consisting of two `LatLng` objects. The `LatLng` objects combined represent the geographical bounds the map view will be set to.
 - [Numeric](#) `padding` - Padding in pixels. The value represents the same padding for all four sides of the map. [optional]
- Return
 - None

setCenter

- Description
 - Sets the geographical center of the map in latitude and longitude.
- Parameters
 - [Dictionary](#) `latLngBounds` - A dictionary consisting of two `LatLng` objects as `{ lat: number, lng: number }`.
- Return
 - None

setClickableIcons

- Description
 - Controls whether the map icons are clickable or not. A map icon represents a point of interest (POI).
- Parameters
 - [Boolean](#) `value` - True to enable clickable map icons, false to disable the clickability of map icons.
- Return
 - None

setHeading

- Description
 - Sets the compass heading for map measured in degrees from cardinal direction North. This method only applies to aerial imagery.
- Parameters
 - [Numeric](#) `heading` - The numerical value in degrees to set the compass heading for the map.
- Return
 - None

setMapTypeId

- Description
- Parameters
 - [String](#) `mapTypeId` - A string identifier that is used to associate a `MapType` with a unique value.
- Return
 - None

setTilt

- Description
 - Controls the automatic switching behavior for the angle of incidence of the map. The only allowed values are 0 (default overhead view) and 45. A 45 degree tilt angle will automatically switch to 45 whenever 45° imagery is available for the current zoom level and viewport and switch back when not available.
- Parameters
 - [Numeric](#) `tilt` - The numerical value of the tilt angle.
- Return
 - None

setZoom

- Description
 - Sets the zoom of the map.
- Parameters
 - [Numeric](#) `zoom` - The numerical value to increase the zoom by. Larger zoom values correspond to a higher resolution.
- Return
 - None.

Perspective - Icon



Component Palette Icon:



On this page ...

- [Properties](#)
- [Component Events](#)
- [Examples](#)
 - [Example 1](#)
 - [Example 2](#)

The Icon component provides access to a collection of Scalable Vector Graphic (SVG) images, chosen to be useful as icons in a Perspective session.

The materials icon library is a primary source for icons, see <https://fonts.google.com/icons?selected=Material+Icons>. You can also add your own custom repository of icons. For more information on icons, see [Images and Icons in Perspective](#).

Note: When attempting to change the color of an icon via style class, set the desired color using the Text category's Color property.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
path	Shorthand path to icon source, in format: library/IconName (e.g., material/3d_rotation). See Icons for more information about available icons.	value: string
color	Color of the icon. See Color Selector .	color
style	Sets a style for this component. Full menu of style options is available. You can also specify a style class .	object

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Examples

Example 1



Property	Value
props.color	#00AC00

Example 2



Property	Value
props.path	material/record_voice_over
props.color	#000088
props.style.borderstyle	ridge
props.style.borderColor	#FFAC47
props.style.borderWidth	6px

Perspective - Image



On this page ...

- [Properties](#)
- [Component Events](#)
- [Examples](#)
 - [Example 1](#)
 - [Example 2](#)

Component Palette Icon:



The Image component displays either vector or raster format images, such as a jpeg, gif, png, or svg. For examples see [Images and Icons in Perspective](#).

When attempting to show images from the [Image Management Tool](#) on this component, you'll need to prefix `/system/images/` to the path. For example:

```
/system/images/Builtin/icons/48/about.png
```

The Image component can also be used to show external images stored relative to the local file system on the client. The file path is similar to having your browser view a local document.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type															
source	<p>The image source URL. It can be a URL to an image on the internet or Gateway, or even an embedded image.</p> <p>If you use images in the Image Management tool, simply copy their path for this source property, using the format <code>/system/images/<imagepath></code>. For example, the <code>Builtin/icons/16/about.png</code> image path would be <code>/system/images/Builtin/icons/16/about.png</code></p> <p>Additionally, the source can be set to a Base64 encoded image.</p>	value: string															
altText	An alternate text for the image if the image cannot be displayed because of a slow connection, an error in the source attribute, if the user uses a screen reader, or some other reason.	value: string															
fit	Whether or not the image will size to fit. When in percent mode, the parameters are used to fit based on the percentage of the width and height. When in absolute mode, the image will fit the width and height sizes in pixels.	object															
	<table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>mode</td><td>Can be one of the following modes: none, fill, contain, cover, percent, or absolute.</td><td>value: string</td></tr><tr><td>width</td><td>Width of the image in pixels.</td><td>value: numeric</td></tr><tr><td>height</td><td>Height of the image in pixels.</td><td>value: numeric</td></tr><tr><td>scroll</td><td>If false, scrolling is not enabled.</td><td>value: boolean</td></tr></tbody></table>	Name	Description	Property Type	mode	Can be one of the following modes: none, fill, contain, cover, percent, or absolute.	value: string	width	Width of the image in pixels.	value: numeric	height	Height of the image in pixels.	value: numeric	scroll	If false, scrolling is not enabled.	value: boolean	
Name	Description	Property Type															
mode	Can be one of the following modes: none, fill, contain, cover, percent, or absolute.	value: string															
width	Width of the image in pixels.	value: numeric															
height	Height of the image in pixels.	value: numeric															
scroll	If false, scrolling is not enabled.	value: boolean															

tint	Enables you to tint the entire image a color.		object
	Name	Description	Property Type
	enabled	Turn tint on (true) and off (false).	value: boolean
	color	If the tint filter is on, this is the color of the tint. See Color Selector .	color
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. Classes are predefined styles in a project.		object

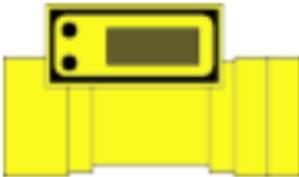
Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Examples

Example 1



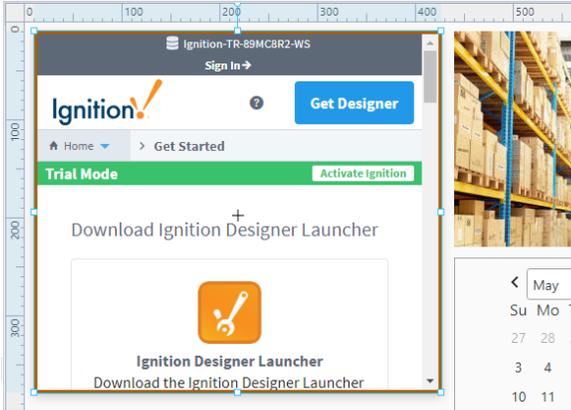
Property	Value
props.source	/system/images/Builtin/Flow/Flow 7.png
props.fit.mode	contain
props.tint.enabled	true
props.tint.color	#FFF00
position.width	150
position.height	115

Example 2



Property	Value	Style Category
props.source	https://inductiveautomation.com/static/images/logos/inductive-automation-logo.png	N/A
props.style.borderStyle	solid	border
props.style.borderWidth	1px	border

Perspective - Inline Frame



On this page ...

- [Properties](#)
- [Component Events](#)
- [Example](#)

Component Palette Icon:



The Inline Frame component enables you to display a webpage within the component, allowing another HTML page to be embedded in the view.

Note that many websites will not support rendering if they're inside a frame, such as this component. Websites choose to opt in to this via the [x-frame-options](#) HTTP header, which all browsers support. The x-frame-options header is designed to help prevent a class of web security attacks called Clickjacking.

Thus, if a given page returns a DENY or SAMEORIGIN value for x-frame-options, then a web browser will refuse to render the content in the iframe on your Perspective page.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

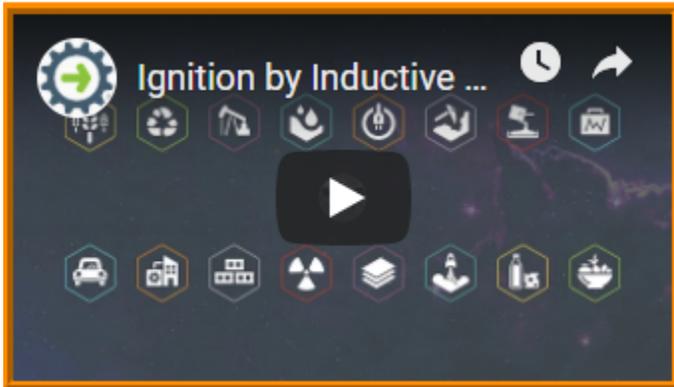
Name	Description	Property Type
src	The source URL of the webpage you want to embed in this frame.	value: string
allowFull IScreen	Whether or not to allow embedded webpage to display full screen. Default is false.	value: boolean
referrer Policy	Referrer Policy is used to determine what information is sent along with the requests. The referral value is stripped when going from a page using HTTPS to a page using the HTTP protocol. This is because this is the default setting for the Referrer Policy if nothing is specified. Technically, this is "no-referrer-when-downgrade," which means it will strip the referral when downgrading to an insecure request like switching from HTTPS to HTTP. You don't have to use the default setting, though. Options as follows: <ul style="list-style-type: none">• no-referrer• no-referrer-when-downgrade• origin• origin-when-cross-origin• unsafe-url	value: string
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object
border	Specifies how the border property is set on the webpage inside the Inline Frame. The default is unset. To set a border, use the borderStyle property in Style on this component. Using the borderStyle property on the component overrides the border property.	value: string

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Example



Property	Value
props.src	https://www.youtube.com/embed/hYXUZeLw5ek
props.allowFullScreen	true
props.referrerPolicy	origin
props.style.borderStyle	ridge
props.style.borderWidth	6px
props.style.borderColor	#FF8C00

Perspective - Label



Component Palette Icon:



On this page ...

- [Properties](#)
- [Component Events](#)
- [Examples](#)
 - [Example 1](#)
 - [Example 2](#)

The Label component displays text and can be customized with a full menu of [style options](#) for the appearance of text, background, border, color, etc. You can use bindings to display additional information on the Label component.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
text	<p>Text to display.</p> <div style="border: 1px solid orange; padding: 5px; margin: 10px 0;"><p>The following feature is new in Ignition version 8.1.26 Click here to check out the other new features</p></div> <p>Text can also be entered directly to the label by deep selecting the component, which enables inline editing. Changes are immediately reflected in the text property field.</p>	value: string
alignVertical	Vertical alignment of the text within the component (top, center, or bottom), based on the dimensions of the component. Default is top.	value: string
textStyle	<div style="border: 1px solid orange; padding: 5px; margin: 10px 0;"><p>The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</p></div> <p>Sets a style for the text within this label. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</p>	object
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Examples

Example 1

The time is: 9:52 AM

Property	Value
props.text	(Expression Binding) <pre>"The time is : "+dateformat(now(),"h:mm a")</pre>
props.style.borderStyle	groove
props.style.padding	12px

Example 2

Lorem ipsum dolor sit amet, consectetur adipiscing ...

Property	Value
props.text	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
props.textStyle.overflow	hidden
props.textStyle.whiteSpace	nowrap
props.textStyle.textOverflow	ellipsis
props.style.borderStyle	double

Perspective - LED Display



Component Palette Icon:



On this page ...

- [Properties](#)
- [Component Events](#)
- [Examples](#)
 - [Example 1](#)
 - [Example 2](#)

The LED Display is a stylized numeric and/or alphanumeric label. It has two visual styles: 7-segment and 14-segment and supports nine common number format patterns. Use the value property to enter numeric and/or alphanumeric characters.

The following feature is new in Ignition version **8.1.2**
[Click here](#) to check out the other new features

TLED Display component has two pre-configured [variants](#):

- 14 Segment - Appearance is that of an LED with 14 light segments.
- 7 Segment - Appearance is that of an LED with 7 light segments.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
value	Value to be displayed.	value: numeric
segmentFormat	Style of each character/digit and the number of segments that compose the character. There are two different visual styles: 7 segment and 14 segment. Default is 14 segment.	value: string
numberFormat	Format of display for numeric characters, including commas, decimal places, percent, etc. There are nine options available from a dropdown list. Default is #,##0.00.	value: string
backgroundColor	Background color of the LED display. Default is #161616. See Color Selector .	color
diodeOnColor	Color of LED segments when switched on. Default is #1EC963. See Color Selector .	color
diodeOffColor	Color of LED segments when switched off. Generally different from display background color to preserve analog look. See Color Selector .	color
locale	Localization code that determines rules for commas, decimals, etc. Default is en-US.	value: string
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Examples

Example 1



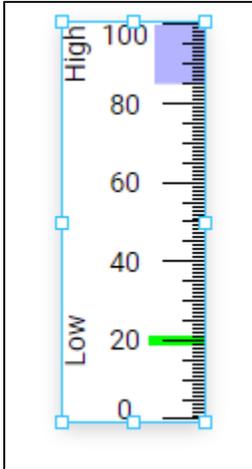
Property	Value
props.value	12.34
props.segmentFormat	7 segment
props.backgroundColor	#D5D5D5
props.diodeOnColor	#0062FF
props.OffColor	#0062FF1A

Example 2



Property	Value	Style Category
props.value	52,345.9	N/A
props.numberFormat	#,##0.00	N/A
props.backgroundColor	#000000	N/A
props.diodeOnColor	#00FF00	N/A
prop.diodeOffColor	#000000	N/A
props.style.borderStyle	groove	border
props.style.padding	2px	margin and padding

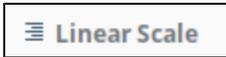
Perspective - Linear Scale



On this page ...

- [Properties](#)
- [Component Events](#)
- [Example](#)

Component Palette Icon:



The Linear Scale component displays a series of tick marks and labels that represent a range between a minimum value and a maximum value. It also displays indicators that represent a value or range of values, correctly positioned on the linear scale.

Linear Scale component allows floating point tick marks such as 0.25, 0.5, 0.75, 1.25, etc.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type															
minValue	The minimum value displayed on the scale.	value: numeric															
maxValue	The maximum value displayed on the scale.	value: numeric															
majorTicks	Settings for the major tick marks on the scale. <table border="1" data-bbox="251 1465 1339 1816"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>span</td> <td>Distance between each tick mark of this type, in pixels. Default is 20.</td> <td>value: numeric</td> </tr> <tr> <td>length</td> <td>Length of each tick mark, in pixels. Default is 20.</td> <td>value: numeric</td> </tr> <tr> <td>color</td> <td>Color for the major ticks. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>color</td> </tr> <tr> <td>stroke</td> <td>Width of each tick mark, in pixels. Default is 1.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	span	Distance between each tick mark of this type, in pixels. Default is 20.	value: numeric	length	Length of each tick mark, in pixels. Default is 20.	value: numeric	color	Color for the major ticks. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color	stroke	Width of each tick mark, in pixels. Default is 1.	value: numeric	object
Name	Description	Property Type															
span	Distance between each tick mark of this type, in pixels. Default is 20.	value: numeric															
length	Length of each tick mark, in pixels. Default is 20.	value: numeric															
color	Color for the major ticks. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color															
stroke	Width of each tick mark, in pixels. Default is 1.	value: numeric															
minorTicks	Settings for the minor tick marks on the scale. <table border="1" data-bbox="251 1879 1339 1965"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> </tbody> </table>	Name	Description	Property Type	object												
Name	Description	Property Type															

	<table border="1"> <tr> <td>span</td> <td>Distance between each tick mark of this type, in pixels. Default is 20.</td> <td>value: numeric</td> </tr> <tr> <td>length</td> <td>Length of each tick mark, in pixels. Default is 20.</td> <td>value: numeric</td> </tr> <tr> <td>color</td> <td>Color for the major ticks. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>color</td> </tr> <tr> <td>stroke</td> <td>Width of each tick mark, in pixels. Default is 1.</td> <td>value: numeric</td> </tr> </table>	span	Distance between each tick mark of this type, in pixels. Default is 20.	value: numeric	length	Length of each tick mark, in pixels. Default is 20.	value: numeric	color	Color for the major ticks. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color	stroke	Width of each tick mark, in pixels. Default is 1.	value: numeric																						
span	Distance between each tick mark of this type, in pixels. Default is 20.	value: numeric																																	
length	Length of each tick mark, in pixels. Default is 20.	value: numeric																																	
color	Color for the major ticks. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color																																	
stroke	Width of each tick mark, in pixels. Default is 1.	value: numeric																																	
fineTicks	Settings for the fine tick marks on the scale.	object																																	
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>span</td> <td>Distance between each tick mark of this type, in pixels. Default is 1.</td> <td>value: numeric</td> </tr> <tr> <td>length</td> <td>Length of each tick mark, in pixels. Default is 5.</td> <td>value: numeric</td> </tr> <tr> <td>color</td> <td>Color for the major ticks. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>color</td> </tr> <tr> <td>stroke</td> <td>Width of each tick mark, in pixels. Default is 1.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	span	Distance between each tick mark of this type, in pixels. Default is 1.	value: numeric	length	Length of each tick mark, in pixels. Default is 5.	value: numeric	color	Color for the major ticks. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color	stroke	Width of each tick mark, in pixels. Default is 1.	value: numeric																			
Name	Description	Property Type																																	
span	Distance between each tick mark of this type, in pixels. Default is 1.	value: numeric																																	
length	Length of each tick mark, in pixels. Default is 5.	value: numeric																																	
color	Color for the major ticks. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color																																	
stroke	Width of each tick mark, in pixels. Default is 1.	value: numeric																																	
labels	Displays of the numeric values of major tick marks. Options are as follows:	object																																	
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>angle</td> <td>Rotation of the numeric labels. Default is 0.</td> <td>value: numeric</td> </tr> <tr> <td>style</td> <td>Sets a style for the label. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	angle	Rotation of the numeric labels. Default is 0.	value: numeric	style	Sets a style for the label. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																									
Name	Description	Property Type																																	
angle	Rotation of the numeric labels. Default is 0.	value: numeric																																	
style	Sets a style for the label. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																																	
indicators	Markers of special significance that can be placed along the scale.	object																																	
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>value</td> <td>Numeric value along the scale where the indicator is placed or started.</td> <td>value: numeric</td> </tr> <tr> <td>label</td> <td>Text to display with the indicator. Default is "High".</td> <td>value: string</td> </tr> <tr> <td>labelColor</td> <td>Color of the indicator label. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>color</td> </tr> <tr> <td>labelAngle</td> <td>Rotational angle of the indicator label.</td> <td>value: numeric</td> </tr> <tr> <td>color</td> <td>Color of the indicator or the area making up the indicator. See Color Selector.</td> <td>color</td> </tr> <tr> <td>stroke</td> <td>If indicatorStyle is set to line or wedge, stroke is the width (in pixels) of the indicator.</td> <td>value: numeric</td> </tr> <tr> <td>opacity</td> <td>Opacity of the indicator. 0 is fully transparent, 1 is fully opaque.</td> <td>value: numeric</td> </tr> <tr> <td>length</td> <td>Length of the indicator as measured by its x value within the scale. Default is 25.</td> <td>value: numeric</td> </tr> <tr> <td>indicatorStyle</td> <td>Indicator style can be set to line, wedge, or range. Line is similar to a tick mark. Wedge displays a triangular shape. Range displays a rectangular range along the scale as measured by the property extent. Default is range.</td> <td>value: string dropdown</td> </tr> <tr> <td>extent</td> <td>If indicatorStyle is set to range, this is the extent along the scale that the indicator is placed. Default is 15.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	value	Numeric value along the scale where the indicator is placed or started.	value: numeric	label	Text to display with the indicator. Default is "High".	value: string	labelColor	Color of the indicator label. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color	labelAngle	Rotational angle of the indicator label.	value: numeric	color	Color of the indicator or the area making up the indicator. See Color Selector .	color	stroke	If indicatorStyle is set to line or wedge, stroke is the width (in pixels) of the indicator.	value: numeric	opacity	Opacity of the indicator. 0 is fully transparent, 1 is fully opaque.	value: numeric	length	Length of the indicator as measured by its x value within the scale. Default is 25.	value: numeric	indicatorStyle	Indicator style can be set to line, wedge, or range. Line is similar to a tick mark. Wedge displays a triangular shape. Range displays a rectangular range along the scale as measured by the property extent. Default is range.	value: string dropdown	extent	If indicatorStyle is set to range, this is the extent along the scale that the indicator is placed. Default is 15.	value: numeric	
Name	Description	Property Type																																	
value	Numeric value along the scale where the indicator is placed or started.	value: numeric																																	
label	Text to display with the indicator. Default is "High".	value: string																																	
labelColor	Color of the indicator label. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color																																	
labelAngle	Rotational angle of the indicator label.	value: numeric																																	
color	Color of the indicator or the area making up the indicator. See Color Selector .	color																																	
stroke	If indicatorStyle is set to line or wedge, stroke is the width (in pixels) of the indicator.	value: numeric																																	
opacity	Opacity of the indicator. 0 is fully transparent, 1 is fully opaque.	value: numeric																																	
length	Length of the indicator as measured by its x value within the scale. Default is 25.	value: numeric																																	
indicatorStyle	Indicator style can be set to line, wedge, or range. Line is similar to a tick mark. Wedge displays a triangular shape. Range displays a rectangular range along the scale as measured by the property extent. Default is range.	value: string dropdown																																	
extent	If indicatorStyle is set to range, this is the extent along the scale that the indicator is placed. Default is 15.	value: numeric																																	

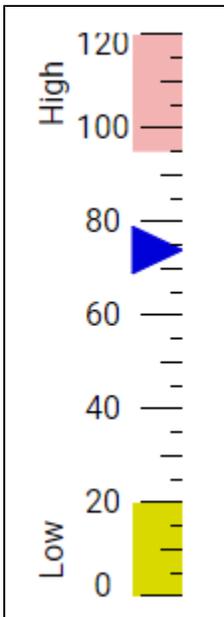
mirror	Aligns the scale to the opposite side of the component. Default is false.	value: boolean
reverse	Inverts the order of the scale values so min to max is ordered in reverse. Default is false (minimum to max).	value: boolean
style	Sets a style for this scale. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Example

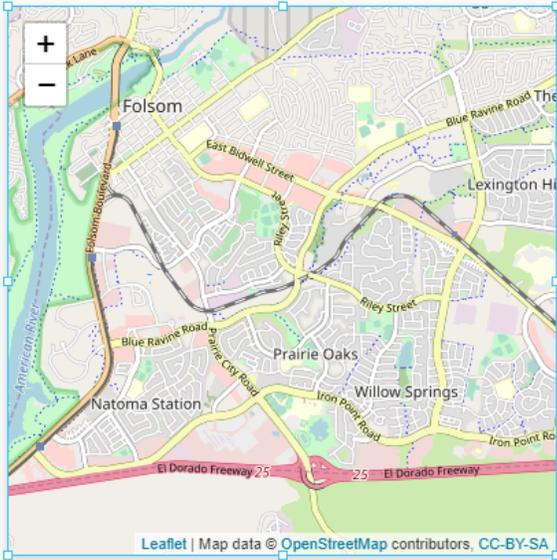


Property	Value
props.maxValue	120
props.minorTicks.span	10
props.fineTicks.span	5
props.indicators.0.value	95
props.indicators.0.color	#D90000
props.indicators.0.extent	25
props.indicators.1.value	00
props.indicators.1.indicatorStyle	range
props.indicators.1.color	#D9D900
props.indicators.1.extent	20
props.indicators.2.value	74
props.indicators.2.indicatorStyle	wedge

props.indicators.2.color

#0000D9

Perspective - Map



On this page ...

- [User Interaction](#)
- [Properties](#)
- [Map Parameters](#)
 - [LatLng](#)
 - [PanOptions](#)
 - [ZoomOptions](#)
 - [FitBounds](#)
- [Scripting](#)
- [Examples](#)
 - [Additional Layers](#)

Component Palette Icon:



The Map component provides a mobile-friendly interactive map. Settings can be customized to control the initial view, zoom, mouse interaction, and keyboard interaction.

The Map component is based on the Leaflet open-source JavaScript library for interactive maps. For more information on Leaflet, see <https://leafletjs.com/reference-1.6.0.html>.

User Interaction

Interaction	Description
Zoom	Depending on the property settings, users can zoom the Map component in several ways: <ul style="list-style-type: none">• Shift and drag the mouse to a rectangular shape.• Double click to zoom in and Shift double-click to zoom out.• Roll the scroll wheel up to zoom in and down to zoom out.• Press the + (plus) key to zoom in and the - (minus) key to zoom out.
Pan	Depending on the property settings, users can pan across the Map component in multiple ways: <ul style="list-style-type: none">• Use the keyboard arrow keys to pan left, right, up, down.• Use the mouse to click and drag the map.
Popups	Depending on property settings, users may see the following popup actions. <ul style="list-style-type: none">• Popups close when they click on the map.• Popups close when they use the escape key.• Popups open as they scroll past certain areas of the map.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description																													
init	<p>Map initial state when loaded. Options as follows:</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>center</td> <td>Sets the latitude and longitude for the initial state of the map.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitude value for the map.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude value for the map.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>zoom</td> <td>Initial map zoom level. Percentage value from 1 to 100. Default is 13.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	center	Sets the latitude and longitude for the initial state of the map.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitude value for the map.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude value for the map.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitude value for the map.	value: numeric	lng	Longitude value for the map.	value: numeric		zoom	Initial map zoom level. Percentage value from 1 to 100. Default is 13.	value: numeric								
Name	Description	Property Type																												
center	Sets the latitude and longitude for the initial state of the map.	object																												
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitude value for the map.</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitude value for the map.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitude value for the map.	value: numeric	lng	Longitude value for the map.	value: numeric																				
Name	Description	Property Type																												
lat	Latitude value for the map.	value: numeric																												
lng	Longitude value for the map.	value: numeric																												
zoom	Initial map zoom level. Percentage value from 1 to 100. Default is 13.	value: numeric																												
location	<p>Map location.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enables geolocation. Default is false (disabled).</td> <td>value: boolean</td> </tr> <tr> <td>showHeadingIndicator</td> <td>Enables the heading indicator. Default is true.</td> <td>value: boolean</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Enables geolocation. Default is false (disabled).	value: boolean	showHeadingIndicator	Enables the heading indicator. Default is true.	value: boolean																				
Name	Description	Property Type																												
enabled	Enables geolocation. Default is false (disabled).	value: boolean																												
showHeadingIndicator	Enables the heading indicator. Default is true.	value: boolean																												
zoom	<p>Zoom properties for the map.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>controls</td> <td>Whether zoom controls are added to the map. Default is true.</td> </tr> <tr> <td>delta</td> <td>Controls how much the map's zoom level will change after a zoomIn(), zoomOut(), pressing + or - on the keyboard, or using the z granularity. Be aware that some combinations of fractional delta and snap values (e.g. delta less than/equal to 0.5 and snap set to rounding).</td> </tr> <tr> <td>onBoxZoom</td> <td>Enables the map to be zoomed to a rectangular area defined by pressing the shift key while dragging the mouse. Default is true.</td> </tr> <tr> <td>onDoubleClick</td> <td>Enables the map to be zoomed in by double-clicking on it and zoomed out by pressing the shift key while double clicking. Default</td> </tr> <tr> <td>onScrollWheel</td> <td>Enables the map to be zoomed in and out using the mouse scroll wheel. Default is true.</td> </tr> <tr> <td>snap</td> <td>Forces the map's zoom level to be a multiple of this value, particularly applicable after fitBounds() or a pinch-zoom. Be aware that g. delta less than/equal to 0.5 and snap set to 1) may cause unresponsive zoom controls due to rounding.</td> </tr> <tr> <td>max</td> <td>Maximum zoom level of the map. Default is null.</td> </tr> <tr> <td>min</td> <td>Minimum zoom level of the map. Default is null.</td> </tr> <tr> <td>animation</td> <td> <p>Animation settings for the map.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether the map zoom animation is enabled.</td> <td>value: boolean</td> </tr> <tr> <td>threshold</td> <td>Won't animate zoom if the zoom difference exceeds this value. Default is 4.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	Name	Description	controls	Whether zoom controls are added to the map. Default is true.	delta	Controls how much the map's zoom level will change after a zoomIn(), zoomOut(), pressing + or - on the keyboard, or using the z granularity. Be aware that some combinations of fractional delta and snap values (e.g. delta less than/equal to 0.5 and snap set to rounding).	onBoxZoom	Enables the map to be zoomed to a rectangular area defined by pressing the shift key while dragging the mouse. Default is true.	onDoubleClick	Enables the map to be zoomed in by double-clicking on it and zoomed out by pressing the shift key while double clicking. Default	onScrollWheel	Enables the map to be zoomed in and out using the mouse scroll wheel. Default is true.	snap	Forces the map's zoom level to be a multiple of this value, particularly applicable after fitBounds() or a pinch-zoom. Be aware that g. delta less than/equal to 0.5 and snap set to 1) may cause unresponsive zoom controls due to rounding.	max	Maximum zoom level of the map. Default is null.	min	Minimum zoom level of the map. Default is null.	animation	<p>Animation settings for the map.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether the map zoom animation is enabled.</td> <td>value: boolean</td> </tr> <tr> <td>threshold</td> <td>Won't animate zoom if the zoom difference exceeds this value. Default is 4.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Whether the map zoom animation is enabled.	value: boolean	threshold	Won't animate zoom if the zoom difference exceeds this value. Default is 4.	value: numeric
Name	Description																													
controls	Whether zoom controls are added to the map. Default is true.																													
delta	Controls how much the map's zoom level will change after a zoomIn(), zoomOut(), pressing + or - on the keyboard, or using the z granularity. Be aware that some combinations of fractional delta and snap values (e.g. delta less than/equal to 0.5 and snap set to rounding).																													
onBoxZoom	Enables the map to be zoomed to a rectangular area defined by pressing the shift key while dragging the mouse. Default is true.																													
onDoubleClick	Enables the map to be zoomed in by double-clicking on it and zoomed out by pressing the shift key while double clicking. Default																													
onScrollWheel	Enables the map to be zoomed in and out using the mouse scroll wheel. Default is true.																													
snap	Forces the map's zoom level to be a multiple of this value, particularly applicable after fitBounds() or a pinch-zoom. Be aware that g. delta less than/equal to 0.5 and snap set to 1) may cause unresponsive zoom controls due to rounding.																													
max	Maximum zoom level of the map. Default is null.																													
min	Minimum zoom level of the map. Default is null.																													
animation	<p>Animation settings for the map.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether the map zoom animation is enabled.</td> <td>value: boolean</td> </tr> <tr> <td>threshold</td> <td>Won't animate zoom if the zoom difference exceeds this value. Default is 4.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Whether the map zoom animation is enabled.	value: boolean	threshold	Won't animate zoom if the zoom difference exceeds this value. Default is 4.	value: numeric																				
Name	Description	Property Type																												
enabled	Whether the map zoom animation is enabled.	value: boolean																												
threshold	Won't animate zoom if the zoom difference exceeds this value. Default is 4.	value: numeric																												
custom Controls	<p>The following feature is new in Ignition version 8.1.6 Click here to check out the other new features</p> <p>Custom map controls displaying views that are shown in the corner of the map.</p> <table border="1"> <thead> <tr> <th>Property</th> <th>Description</th> </tr> </thead> <tbody> </tbody> </table>	Property	Description																											
Property	Description																													

path	Path to the view that will be used as the display for the control. <div style="border: 1px solid orange; padding: 5px; margin: 10px 0;"> The following feature is new in Ignition version 8.1.29 Click here to check out the other new features </div> If a path is present in the path property field, an Open View  icon will appear that will navigate directly to the view when cli															
params	Parameters to be passed to the view within the control. Names in this object must match input parameters defined on the view															
position	Corner of the map where the control will be anchored. Options are: bottom-right, top-right, or bottom left.															
enabled	Whether or not the custom control is enabled.															
attributi on	Enables an attribution control on the map. Default is true.															
closePo pupsOn Click	When set to true, popups will close when a user clicks anywhere else on the map. Default is true.															
trackRe size	Enables the map to automatically handle browser window resizing. Default is false.															
keyboar dNav	Enables navigation of the map with keyboard arrow key and with the + (plus) and - (minus) keys. The + key zooms in; the - key zooms out. De															
keyboar dPanDe lta	The number of pixels to pan when keyboard a arrow key is pressed. Default is 80.															
dragging	Enables the map to be dragged with a mouse/touch. Default is true.															
maxBou nds	When this option is set, the map restricts the view to the given geographical bounds, bouncing the user back if the user tries to pan outside th <table border="1" data-bbox="256 1020 1138 1108" style="width: 100%; border-collapse: collapse;"> <tr> <td data-bbox="256 1020 456 1068">corner1 and corner2</td> <td data-bbox="456 1020 1065 1068">Objects that determine opposing corners of the maximum bounds.</td> <td data-bbox="1065 1020 1138 1068">object</td> </tr> <tr> <td colspan="3" data-bbox="256 1068 1138 1108">Each corner contain a lat and lng value.</td> </tr> </table>		corner1 and corner2	Objects that determine opposing corners of the maximum bounds.	object	Each corner contain a lat and lng value.										
corner1 and corner2	Objects that determine opposing corners of the maximum bounds.	object														
Each corner contain a lat and lng value.																
fadeAni mation	Enables the fade animation. Default is true.															
layers	Settings for the map UI, vector, and raster layers that can be displayed on top of the map. <table border="1" data-bbox="256 1241 1495 1965" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th data-bbox="256 1241 354 1318">Name</th> <th data-bbox="354 1241 1495 1318">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="256 1318 354 1367">raster</td> <td data-bbox="354 1318 1495 1367">Map raster layers.</td> </tr> <tr> <td data-bbox="256 1367 354 1965">tile</td> <td data-bbox="354 1367 1495 1965"> <table border="1" data-bbox="354 1367 1495 1965" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th data-bbox="354 1367 456 1444">Name</th> <th data-bbox="456 1367 1495 1444">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="354 1444 456 1808">url</td> <td data-bbox="456 1444 1495 1808"> Settings enable you to load and display tile layers on the map. The tile provider can be customized to change the app extras.github.io/leaflet-providers/preview/ for some possibilities. URL can be either a URL template if using a standard tile service or a Web Map Services URL. Default is openstreetmap.org/{z}/{x}/{y}.png. In cases where you have the tile files yourself, they need to be hosted on a web server of some sort. Alte WebDev Mounted Folders can be used to host the tiles. </td> </tr> <tr> <td data-bbox="354 1808 456 1898">uriTemp latePar ams</td> <td data-bbox="456 1808 1495 1898">An object of params to use with a URL template string, if the url prop is a template string (i.e. contains {z})</td> </tr> <tr> <td data-bbox="354 1898 456 1965">WMS</td> <td data-bbox="456 1898 1495 1965">Web Map Service tile layer provider configurations. Used only if the url property is not a template string. org/wiki/Web_Map_Service.</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>		Name	Description	raster	Map raster layers.	tile	<table border="1" data-bbox="354 1367 1495 1965" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th data-bbox="354 1367 456 1444">Name</th> <th data-bbox="456 1367 1495 1444">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="354 1444 456 1808">url</td> <td data-bbox="456 1444 1495 1808"> Settings enable you to load and display tile layers on the map. The tile provider can be customized to change the app extras.github.io/leaflet-providers/preview/ for some possibilities. URL can be either a URL template if using a standard tile service or a Web Map Services URL. Default is openstreetmap.org/{z}/{x}/{y}.png. In cases where you have the tile files yourself, they need to be hosted on a web server of some sort. Alte WebDev Mounted Folders can be used to host the tiles. </td> </tr> <tr> <td data-bbox="354 1808 456 1898">uriTemp latePar ams</td> <td data-bbox="456 1808 1495 1898">An object of params to use with a URL template string, if the url prop is a template string (i.e. contains {z})</td> </tr> <tr> <td data-bbox="354 1898 456 1965">WMS</td> <td data-bbox="456 1898 1495 1965">Web Map Service tile layer provider configurations. Used only if the url property is not a template string. org/wiki/Web_Map_Service.</td> </tr> </tbody> </table>	Name	Description	url	Settings enable you to load and display tile layers on the map. The tile provider can be customized to change the app extras.github.io/leaflet-providers/preview/ for some possibilities. URL can be either a URL template if using a standard tile service or a Web Map Services URL. Default is openstreetmap.org/{z}/{x}/{y}.png . In cases where you have the tile files yourself, they need to be hosted on a web server of some sort. Alte WebDev Mounted Folders can be used to host the tiles.	uriTemp latePar ams	An object of params to use with a URL template string, if the url prop is a template string (i.e. contains {z})	WMS	Web Map Service tile layer provider configurations. Used only if the url property is not a template string. org/wiki/Web_Map_Service .
Name	Description															
raster	Map raster layers.															
tile	<table border="1" data-bbox="354 1367 1495 1965" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th data-bbox="354 1367 456 1444">Name</th> <th data-bbox="456 1367 1495 1444">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="354 1444 456 1808">url</td> <td data-bbox="456 1444 1495 1808"> Settings enable you to load and display tile layers on the map. The tile provider can be customized to change the app extras.github.io/leaflet-providers/preview/ for some possibilities. URL can be either a URL template if using a standard tile service or a Web Map Services URL. Default is openstreetmap.org/{z}/{x}/{y}.png. In cases where you have the tile files yourself, they need to be hosted on a web server of some sort. Alte WebDev Mounted Folders can be used to host the tiles. </td> </tr> <tr> <td data-bbox="354 1808 456 1898">uriTemp latePar ams</td> <td data-bbox="456 1808 1495 1898">An object of params to use with a URL template string, if the url prop is a template string (i.e. contains {z})</td> </tr> <tr> <td data-bbox="354 1898 456 1965">WMS</td> <td data-bbox="456 1898 1495 1965">Web Map Service tile layer provider configurations. Used only if the url property is not a template string. org/wiki/Web_Map_Service.</td> </tr> </tbody> </table>	Name	Description	url	Settings enable you to load and display tile layers on the map. The tile provider can be customized to change the app extras.github.io/leaflet-providers/preview/ for some possibilities. URL can be either a URL template if using a standard tile service or a Web Map Services URL. Default is openstreetmap.org/{z}/{x}/{y}.png . In cases where you have the tile files yourself, they need to be hosted on a web server of some sort. Alte WebDev Mounted Folders can be used to host the tiles.	uriTemp latePar ams	An object of params to use with a URL template string, if the url prop is a template string (i.e. contains {z})	WMS	Web Map Service tile layer provider configurations. Used only if the url property is not a template string. org/wiki/Web_Map_Service .							
Name	Description															
url	Settings enable you to load and display tile layers on the map. The tile provider can be customized to change the app extras.github.io/leaflet-providers/preview/ for some possibilities. URL can be either a URL template if using a standard tile service or a Web Map Services URL. Default is openstreetmap.org/{z}/{x}/{y}.png . In cases where you have the tile files yourself, they need to be hosted on a web server of some sort. Alte WebDev Mounted Folders can be used to host the tiles.															
uriTemp latePar ams	An object of params to use with a URL template string, if the url prop is a template string (i.e. contains {z})															
WMS	Web Map Service tile layer provider configurations. Used only if the url property is not a template string. org/wiki/Web_Map_Service .															

Name	Description	Prop
layers	Comma-separated list of WMS layers to display. (Required)	value
styles	Comma-separated list of WMS styles.	value
format	WMS image format (use 'image/png' for layers with transparency).	value
transparent	If enabled, the WMS service will return images with transparency. Default is false.	value
version	Version of the WMS service to use.	value
uppercase	If enabled, WMS request parameters keys will be uppercase. Default is false.	value

options Options for standard tile layer creation.

Name	Description									
attribution	This tile layers attribution.									
opacity	Opacity of tiles.									
zIndex	The z-index of tiles in the grid.									
tileSize	Width and height of tiles in the grid.									
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>width</td> <td>Width of tiles in the grid.</td> <td>value: numeric</td> </tr> <tr> <td>height</td> <td>Height of tiles in the grid.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	width	Width of tiles in the grid.	value: numeric	height	Height of tiles in the grid.	value: numeric
	Name	Description	Property Type							
	width	Width of tiles in the grid.	value: numeric							
height	Height of tiles in the grid.	value: numeric								

update Tile update options.

Name	Description	P T
whenZooming	By default, a smooth animation will update grid layers at every integer zoom level. Setting this to false will update the grid layer only when the smooth animation ends. Default is true.	va bo
whenIdle	Load new tiles only when panning ends. True by default on mobile browsers, in order to avoid too many requests and keep smooth navigation. Default is false otherwise, in order to display new tiles during panning.	va bo
interval	Tiles will not update more than once every update interval in milliseconds when panning.	va nu

zoom Zoom options.

Name	Description	P T						
max	Maximum zoom level up to which this layer will be displayed (inclusive). Default is 18.	va nu						
min	Minimum zoom level down to which this layer will be displayed (inclusive). Default is 0.	va nu						
offset	Zoom number used in the tile URLs will be offset with this value.	va nu						
reverse	If set to true, the zoom number used in the tile URLs will be reversed (maxZoom - zoom instead of zoom to maxZoom). Default is false.	va bo						
native	Native zoom levels.	ob						
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Description	Property				
Name	Description	Property						

		Type
max	Maximum zoom number the tile source has available. If specified, the tiles on all zoom levels higher than maxNativeZoom will be loaded from maxNativeZoom level and auto-scaled. Default is null.	Value: string
min	Minimum zoom number the tile source has available. If specified, the tiles on all zoom levels lower than minNativeZoom will be loaded from minNativeZoom level and auto-scaled. Default is null.	Value: string

subdomains Subdomains of the tile service. Passed in the form of an array of strings (where each string is subdomain name). For example, ['a','b','c'].

Name	Description	P T
errorTileUrl	URL to the tile image to show in place of the tile that failed to load.	va str
tms	If true, inverses Y axis numbering for tiles (turn this on for Tile Map Service services). Default is false.	va bo
detectRetina	If true and user is on a retina display, it will request four tiles of half the specified size and a bigger zoom level in place of one to utilize the high resolution. Default is false.	va bo
crossOrigin	Enables the crossOrigin attribute to be added to the tiles. If a string is provided, all tiles will have their crossOrigin attribute set to the String provided. Default is false.	va bo

image Settings for images displayed over specific bounds of the map.

Name	Description																											
source	Image url.																											
bounds	Rectangle bounds. <table border="1" data-bbox="573 1354 1365 1822"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>corner1</td> <td>Settings for lat and lng bounds for corner1. <table border="1" data-bbox="678 1459 1179 1606"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitudinal coordinate</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td>object</td> </tr> <tr> <td>corner2</td> <td>Settings for lat and lng bounds for corner2. <table border="1" data-bbox="678 1669 1179 1816"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitudinal coordinate</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	corner1	Settings for lat and lng bounds for corner1. <table border="1" data-bbox="678 1459 1179 1606"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitudinal coordinate</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate	value: numeric	lng	Longitudinal coordinate	value: numeric	object	corner2	Settings for lat and lng bounds for corner2. <table border="1" data-bbox="678 1669 1179 1816"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitudinal coordinate</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate	value: numeric	lng	Longitudinal coordinate	value: numeric	object
Name	Description	Property Type																										
corner1	Settings for lat and lng bounds for corner1. <table border="1" data-bbox="678 1459 1179 1606"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitudinal coordinate</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate	value: numeric	lng	Longitudinal coordinate	value: numeric	object																	
Name	Description	Property Type																										
lat	Latitudinal coordinate	value: numeric																										
lng	Longitudinal coordinate	value: numeric																										
corner2	Settings for lat and lng bounds for corner2. <table border="1" data-bbox="678 1669 1179 1816"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitudinal coordinate</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate	value: numeric	lng	Longitudinal coordinate	value: numeric	object																	
Name	Description	Property Type																										
lat	Latitudinal coordinate	value: numeric																										
lng	Longitudinal coordinate	value: numeric																										
options	Options for the image overlay. <table border="1" data-bbox="573 1890 1500 1963"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table>	Name	Description																									
Name	Description																											

opacity	The opacity of the image overlay.
alt	Text for the alt attribute of the image (useful for accessibility).
crossOrigin	Whether the crossOrigin attribute will be added to the image. If a string is provided, the image have its crossOrigin attribute set to the String provided.
errorOverlayUrl	URL to the overlay image to show in place of the overlay that failed to load.
zIndex	The explicit zIndex of the image layer.

vector Map vector layers. Allows the map to draw arbitrary shapes at specified coordinates.

Name	Description																			
polygons	An array of polygons each consisting of an array of points that create a single polygon.																			
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitudinal coordinate</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate	value: numeric	lng	Longitudinal coordinate	value: numeric										
Name	Description	Property Type																		
lat	Latitudinal coordinate	value: numeric																		
lng	Longitudinal coordinate	value: numeric																		
	<p><u>This feature was changed in Ignition version 8.1.15:</u></p> <p>In 8.1.15 the default shape of polygons changed to the following:</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>A unique name which is used to distinguish this elements from others. This name is provided applicable interaction event handlers, such as click handlers.</td> </tr> <tr> <td>properties</td> <td>Properties of this element. These are provided to any applicable interaction event handlers, s click handlers.</td> </tr> <tr> <td>points</td> <td>An array of points that make up this polygon.</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitudinal coordinate</td> <td>value: numeric</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	Name	Description	name	A unique name which is used to distinguish this elements from others. This name is provided applicable interaction event handlers, such as click handlers.	properties	Properties of this element. These are provided to any applicable interaction event handlers, s click handlers.	points	An array of points that make up this polygon.		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitudinal coordinate</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate	value: numeric	lng	Longitudinal coordinate	value: numeric
Name	Description																			
name	A unique name which is used to distinguish this elements from others. This name is provided applicable interaction event handlers, such as click handlers.																			
properties	Properties of this element. These are provided to any applicable interaction event handlers, s click handlers.																			
points	An array of points that make up this polygon.																			
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitudinal coordinate</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate	value: numeric	lng	Longitudinal coordinate	value: numeric										
Name	Description	Property Type																		
lat	Latitudinal coordinate	value: numeric																		
lng	Longitudinal coordinate	value: numeric																		
smoothFactor	How much to simplify each vector on each zoom level. Higher the number means better performance and lower the number means more accurate representation.																			
noClip	Disables polyline clipping.																			
event	Event settings for the individual polygon layer.																			
	<table border="1"> <tr> <td>stopPropagation</td> <td>When enabled, this prevents other mouse event interactions from happening when this layer is clicked.</td> </tr> </table>	stopPropagation	When enabled, this prevents other mouse event interactions from happening when this layer is clicked.																	
stopPropagation	When enabled, this prevents other mouse event interactions from happening when this layer is clicked.																			
stroke	Stroke settings for the individual polygon layer.																			
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> </table>	Name	Description																	
Name	Description																			

enabled	Whether to draw stroke along the path. Set it to false to disable borders.
color	Stroke color.
weight	Stroke weight in pixels.
opacity	Stroke opacity (0-1).
dashArray	Stroke dash array
dashOffset	Defines the distance in the dash pattern to start the dash.
lineCap	A string that defines shape to be used at the end of the stroke. Options are round, butt, or square.
lineJoin	A string that defines shape to be used at the corners of the stroke. Options are round, arc, bevel, miter, or miter-clip.

fill	Fill settings for the individual polygon layer.										
<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether to fill the pattern with color.</td> </tr> <tr> <td>color</td> <td>Fill color.</td> </tr> <tr> <td>opacity</td> <td>Fill opacity (0-1).</td> </tr> <tr> <td>rule</td> <td>Presentation attribute defining the algorithm to use to determine the inside part of the shape. Options are nonzero or evenodd.</td> </tr> </tbody> </table>		Name	Description	enabled	Whether to fill the pattern with color.	color	Fill color.	opacity	Fill opacity (0-1).	rule	Presentation attribute defining the algorithm to use to determine the inside part of the shape. Options are nonzero or evenodd.
Name	Description										
enabled	Whether to fill the pattern with color.										
color	Fill color.										
opacity	Fill opacity (0-1).										
rule	Presentation attribute defining the algorithm to use to determine the inside part of the shape. Options are nonzero or evenodd.										

polyline An array of polyline layers.

Name	Description																			
polylines	An array of polylines each consisting of an array of points that create a single polyline.																			
<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitudinal coordinate</td> <td>value: numeric</td> </tr> </tbody> </table>		Name	Description	Property Type	lat	Latitudinal coordinate	value: numeric	lng	Longitudinal coordinate	value: numeric										
Name	Description	Property Type																		
lat	Latitudinal coordinate	value: numeric																		
lng	Longitudinal coordinate	value: numeric																		
<div style="border: 1px solid black; padding: 5px;"> <p>This feature was changed in Ignition version 8.1.15:</p> </div>																				
<p>In 8.1.15 the default shape of polylines changed to the following:</p>																				
<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>A unique name which is used to distinguish this elements from others. This name is provided to any applicable interaction event handlers, such as click handlers.</td> </tr> <tr> <td>properties</td> <td>Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.</td> </tr> <tr> <td>points</td> <td>An array of points along this polyline. Each point must contain a minimum of two elements, each with the following properties:</td> </tr> <tr> <td colspan="2"> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitudinal coordinate</td> <td>value: numeric</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>		Name	Description	name	A unique name which is used to distinguish this elements from others. This name is provided to any applicable interaction event handlers, such as click handlers.	properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.	points	An array of points along this polyline. Each point must contain a minimum of two elements, each with the following properties:	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitudinal coordinate</td> <td>value: numeric</td> </tr> </tbody> </table>		Name	Description	Property Type	lat	Latitudinal coordinate	value: numeric	lng	Longitudinal coordinate	value: numeric
Name	Description																			
name	A unique name which is used to distinguish this elements from others. This name is provided to any applicable interaction event handlers, such as click handlers.																			
properties	Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.																			
points	An array of points along this polyline. Each point must contain a minimum of two elements, each with the following properties:																			
<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitudinal coordinate</td> <td>value: numeric</td> </tr> </tbody> </table>		Name	Description	Property Type	lat	Latitudinal coordinate	value: numeric	lng	Longitudinal coordinate	value: numeric										
Name	Description	Property Type																		
lat	Latitudinal coordinate	value: numeric																		
lng	Longitudinal coordinate	value: numeric																		

smoothFactor	How much to simplify each polyline on the zoom level.																		
noClip	Disables polyline clipping.																		
event	Event settings for the individual polyline layer.																		
	<table border="1"> <tr> <td>stopPropagation</td> <td>When enabled, this prevents other mouse event interactions from happening when this layer is clicked.</td> </tr> </table>	stopPropagation	When enabled, this prevents other mouse event interactions from happening when this layer is clicked.																
stopPropagation	When enabled, this prevents other mouse event interactions from happening when this layer is clicked.																		
stroke	Stroke settings for the individual polyline layer.																		
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether to draw stroke along the path. Set it to false to disable borders.</td> </tr> <tr> <td>color</td> <td>Stroke color.</td> </tr> <tr> <td>weight</td> <td>Stroke weight.</td> </tr> <tr> <td>opacity</td> <td>Stroke opacity.</td> </tr> <tr> <td>dashArray</td> <td>Stroke dash array</td> </tr> <tr> <td>dashOffset</td> <td>Defines the distance in the dash pattern to start the dash.</td> </tr> <tr> <td>lineCap</td> <td>Shape to be used at the end of the stroke. Options are round, butt, or square.</td> </tr> <tr> <td>lineJoin</td> <td>Shape to be used at the corners of the stroke. Options are round, arc, bevel, miter, or miter clip.</td> </tr> </tbody> </table>	Name	Description	enabled	Whether to draw stroke along the path. Set it to false to disable borders.	color	Stroke color.	weight	Stroke weight.	opacity	Stroke opacity.	dashArray	Stroke dash array	dashOffset	Defines the distance in the dash pattern to start the dash.	lineCap	Shape to be used at the end of the stroke. Options are round, butt, or square.	lineJoin	Shape to be used at the corners of the stroke. Options are round, arc, bevel, miter, or miter clip.
	Name	Description																	
	enabled	Whether to draw stroke along the path. Set it to false to disable borders.																	
	color	Stroke color.																	
	weight	Stroke weight.																	
	opacity	Stroke opacity.																	
	dashArray	Stroke dash array																	
	dashOffset	Defines the distance in the dash pattern to start the dash.																	
	lineCap	Shape to be used at the end of the stroke. Options are round, butt, or square.																	
lineJoin	Shape to be used at the corners of the stroke. Options are round, arc, bevel, miter, or miter clip.																		
fill	Fill settings for the individual polyline layer.																		
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether to fill the pattern with color.</td> </tr> <tr> <td>color</td> <td>Fill color.</td> </tr> <tr> <td>opacity</td> <td>Fill opacity.</td> </tr> <tr> <td>rule</td> <td>Presentation attribute defining the algorithm to use to determine the inside part of the shape. are nonzero or evenodd.</td> </tr> </tbody> </table>	Name	Description	enabled	Whether to fill the pattern with color.	color	Fill color.	opacity	Fill opacity.	rule	Presentation attribute defining the algorithm to use to determine the inside part of the shape. are nonzero or evenodd.								
	Name	Description																	
	enabled	Whether to fill the pattern with color.																	
	color	Fill color.																	
	opacity	Fill opacity.																	
rule	Presentation attribute defining the algorithm to use to determine the inside part of the shape. are nonzero or evenodd.																		

rectangle An array of rectangle layers.

Name	Description						
rectangles	An array of rectangle bounds each consisting of two corners that create a single rectangle.						
properties	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>name</td> <td> <p>The following feature is new in Ignition version 8.1.15 Click here to check out the other new features</p> <p>A unique name which is used to distinguish this elements from others. This name is provided applicable interaction event handlers, such as click handlers.</p> </td> </tr> <tr> <td>properties</td> <td> <p>The following feature is new in Ignition version 8.1.15 Click here to check out the other new features</p> </td> </tr> </tbody> </table>	Name	Description	name	<p>The following feature is new in Ignition version 8.1.15 Click here to check out the other new features</p> <p>A unique name which is used to distinguish this elements from others. This name is provided applicable interaction event handlers, such as click handlers.</p>	properties	<p>The following feature is new in Ignition version 8.1.15 Click here to check out the other new features</p>
	Name	Description					
name	<p>The following feature is new in Ignition version 8.1.15 Click here to check out the other new features</p> <p>A unique name which is used to distinguish this elements from others. This name is provided applicable interaction event handlers, such as click handlers.</p>						
properties	<p>The following feature is new in Ignition version 8.1.15 Click here to check out the other new features</p>						

	Properties of this element. These are provided to any applicable interaction event handlers, s click handlers.																		
corner1 and corner2	<p>An object with two elements, each element represents an opposing corner of the rectangle. E corner contains the following properties:</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>lat</td> <td>Latitudinal coordinate</td> <td>value: numeric</td> </tr> <tr> <td>lng</td> <td>Longitudinal coordinate</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	lat	Latitudinal coordinate	value: numeric	lng	Longitudinal coordinate	value: numeric									
Name	Description	Property Type																	
lat	Latitudinal coordinate	value: numeric																	
lng	Longitudinal coordinate	value: numeric																	
smoothFactor	How much to simplify each rectangle on the zoom level.																		
noClip	Disables rectangle clipping.																		
event	Event settings for the individual rectangle layer.																		
	<table border="1"> <tr> <td>stopPropagation</td> <td>When enabled, this prevents other mouse event interactions from happening when this layer is clicked.</td> </tr> </table>	stopPropagation	When enabled, this prevents other mouse event interactions from happening when this layer is clicked.																
stopPropagation	When enabled, this prevents other mouse event interactions from happening when this layer is clicked.																		
stroke	Stroke settings.																		
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether to draw stroke along the path. Set it to false to disable borders.</td> </tr> <tr> <td>color</td> <td>Stroke color.</td> </tr> <tr> <td>weight</td> <td>Stroke width in pixels.</td> </tr> <tr> <td>opacity</td> <td>Stroke opacity (0-1).</td> </tr> <tr> <td>dashArray</td> <td>Stroke dash array.</td> </tr> <tr> <td>dashOffset</td> <td>Defines the distance in the dash patter to start the dash.</td> </tr> <tr> <td>lineCap</td> <td>A string that defines the shape to be used at the end of the stroke. Options are round, butt, c square.</td> </tr> <tr> <td>lineJoin</td> <td>A string that defines the shape to be used at the corners of the stroke. Options are round, an bevel, miter, or miter-clip.</td> </tr> </tbody> </table>	Name	Description	enabled	Whether to draw stroke along the path. Set it to false to disable borders.	color	Stroke color.	weight	Stroke width in pixels.	opacity	Stroke opacity (0-1).	dashArray	Stroke dash array.	dashOffset	Defines the distance in the dash patter to start the dash.	lineCap	A string that defines the shape to be used at the end of the stroke. Options are round, butt, c square.	lineJoin	A string that defines the shape to be used at the corners of the stroke. Options are round, an bevel, miter, or miter-clip.
Name	Description																		
enabled	Whether to draw stroke along the path. Set it to false to disable borders.																		
color	Stroke color.																		
weight	Stroke width in pixels.																		
opacity	Stroke opacity (0-1).																		
dashArray	Stroke dash array.																		
dashOffset	Defines the distance in the dash patter to start the dash.																		
lineCap	A string that defines the shape to be used at the end of the stroke. Options are round, butt, c square.																		
lineJoin	A string that defines the shape to be used at the corners of the stroke. Options are round, an bevel, miter, or miter-clip.																		
fill	Fill settings for the individual rectangle layer.																		
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether to fill the pattern with color.</td> </tr> <tr> <td>color</td> <td>Fill color.</td> </tr> <tr> <td>opacity</td> <td>Fill opacity (0-1).</td> </tr> <tr> <td>rule</td> <td>Presentation attribute defining the algorithm to use to determine the inside part of the shape. are nonzero or evenodd.</td> </tr> </tbody> </table>	Name	Description	enabled	Whether to fill the pattern with color.	color	Fill color.	opacity	Fill opacity (0-1).	rule	Presentation attribute defining the algorithm to use to determine the inside part of the shape. are nonzero or evenodd.								
Name	Description																		
enabled	Whether to fill the pattern with color.																		
color	Fill color.																		
opacity	Fill opacity (0-1).																		
rule	Presentation attribute defining the algorithm to use to determine the inside part of the shape. are nonzero or evenodd.																		
circle	An array of circle layers.																		
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>circles</td> <td>An array of circles each consisting of center point corners that creates a single circle.</td> </tr> </tbody> </table>	Name	Description	circles	An array of circles each consisting of center point corners that creates a single circle.														
Name	Description																		
circles	An array of circles each consisting of center point corners that creates a single circle.																		

Name	Description																		
name	<p>The following feature is new in Ignition version 8.1.15 Click here to check out the other new features</p> <p>A unique name which is used to distinguish this elements from others. This name is provided applicable interaction event handlers, such as click handlers.</p>																		
properties	<p>The following feature is new in Ignition version 8.1.15 Click here to check out the other new features</p> <p>Properties of this element. These are provided to any applicable interaction event handlers, such as click handlers.</p>																		
lat	Latitudinal coordinate																		
lng	Longitudinal coordinate																		
radius	Radius of the circle marker, in pixels.																		
event	Event settings for the individual circle layer.																		
stopPropagation	When enabled, this prevents other mouse event interactions from happening when this layer is clicked.																		
stroke	Stroke settings for individual circle layer.																		
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether to draw stroke along the path. Set it to false to disable borders.</td> </tr> <tr> <td>color</td> <td>Stroke color.</td> </tr> <tr> <td>weight</td> <td>Stroke width in pixels.</td> </tr> <tr> <td>opacity</td> <td>Stroke opacity.</td> </tr> <tr> <td>dashArray</td> <td>Stroke dash array</td> </tr> <tr> <td>dashOffset</td> <td>Defines the distance in the dash pattern to start the dash.</td> </tr> <tr> <td>lineCap</td> <td>Shape to be used at the end of the stroke. Options are round, butt, or square.</td> </tr> <tr> <td>lineJoin</td> <td>Shape to be used at the corners of the stroke. Options are round, arcs, bevel, miter, or mit clip.</td> </tr> </tbody> </table>	Name	Description	enabled	Whether to draw stroke along the path. Set it to false to disable borders.	color	Stroke color.	weight	Stroke width in pixels.	opacity	Stroke opacity.	dashArray	Stroke dash array	dashOffset	Defines the distance in the dash pattern to start the dash.	lineCap	Shape to be used at the end of the stroke. Options are round, butt, or square.	lineJoin	Shape to be used at the corners of the stroke. Options are round, arcs, bevel, miter, or mit clip.
Name	Description																		
enabled	Whether to draw stroke along the path. Set it to false to disable borders.																		
color	Stroke color.																		
weight	Stroke width in pixels.																		
opacity	Stroke opacity.																		
dashArray	Stroke dash array																		
dashOffset	Defines the distance in the dash pattern to start the dash.																		
lineCap	Shape to be used at the end of the stroke. Options are round, butt, or square.																		
lineJoin	Shape to be used at the corners of the stroke. Options are round, arcs, bevel, miter, or mit clip.																		
fill	Fill settings for individual circle layer.																		
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether to fill the pattern with color.</td> </tr> <tr> <td>color</td> <td>Fill color.</td> </tr> <tr> <td>opacity</td> <td>Fill opacity.</td> </tr> <tr> <td>rule</td> <td>Presentation attribute defining the algorithm to use to determine the inside part of the shape. are nonzero or evenodd.</td> </tr> </tbody> </table>	Name	Description	enabled	Whether to fill the pattern with color.	color	Fill color.	opacity	Fill opacity.	rule	Presentation attribute defining the algorithm to use to determine the inside part of the shape. are nonzero or evenodd.								
Name	Description																		
enabled	Whether to fill the pattern with color.																		
color	Fill color.																		
opacity	Fill opacity.																		
rule	Presentation attribute defining the algorithm to use to determine the inside part of the shape. are nonzero or evenodd.																		

ui

Map user interface (UI) layers. An array of popup layer configurations for this map. Includes popup location, width, height, and co

Name	Description																																																															
marker	Map marker layers.																																																															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>A unique name that is used to distinguish this marker from another.</td> </tr> <tr> <td>enabled</td> <td> <div style="border: 1px solid orange; padding: 5px; margin-bottom: 5px;"> <p>The following feature is new in Ignition version 8.1.16 Click here to check out the other new features</p> </div> <p>Determines if the marker is shown on the map. Default value is true.</p> </td> </tr> <tr> <td>lat</td> <td>Latitudinal coordinate.</td> </tr> <tr> <td>lng</td> <td>Longitudinal coordinate.</td> </tr> <tr> <td>opacity</td> <td>Marker opacity.</td> </tr> <tr> <td>icon</td> <td>Marker icon configuration.</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Icon path. Otherwise uses default icon.</td> </tr> <tr> <td>color</td> <td>Fill color for the icon.</td> </tr> <tr> <td>size</td> <td>Size settings.</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>width</td> <td>Width in pixels.</td> <td>value: numeric</td> </tr> <tr> <td>height</td> <td>Height in pixels.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> </tr> <tr> <td>style</td> <td>Style for the icon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> </tr> <tr> <td>rotate</td> <td> <div style="border: 1px solid orange; padding: 5px; margin-bottom: 5px;"> <p>The following feature is new in Ignition version 8.1.21 Click here to check out the other new features</p> </div> <p>Rotation degree of the specified map marker</p> </td> </tr> </tbody> </table> </td> </tr> <tr> <td>tooltip</td> <td>This marker's tooltip configuration, if tooltip is enabled.</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>P</th> <th>T</th> </tr> </thead> <tbody> <tr> <td>content</td> <td>The tooltip content to display.</td> <td></td> <td></td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>P</th> <th>T</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Text to display.</td> <td>va</td> <td>str</td> </tr> </tbody> </table> </td> <td></td> <td></td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	Name	Description	name	A unique name that is used to distinguish this marker from another.	enabled	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 5px;"> <p>The following feature is new in Ignition version 8.1.16 Click here to check out the other new features</p> </div> <p>Determines if the marker is shown on the map. Default value is true.</p>	lat	Latitudinal coordinate.	lng	Longitudinal coordinate.	opacity	Marker opacity.	icon	Marker icon configuration.		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Icon path. Otherwise uses default icon.</td> </tr> <tr> <td>color</td> <td>Fill color for the icon.</td> </tr> <tr> <td>size</td> <td>Size settings.</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>width</td> <td>Width in pixels.</td> <td>value: numeric</td> </tr> <tr> <td>height</td> <td>Height in pixels.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> </tr> <tr> <td>style</td> <td>Style for the icon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> </tr> <tr> <td>rotate</td> <td> <div style="border: 1px solid orange; padding: 5px; margin-bottom: 5px;"> <p>The following feature is new in Ignition version 8.1.21 Click here to check out the other new features</p> </div> <p>Rotation degree of the specified map marker</p> </td> </tr> </tbody> </table>	Name	Description	path	Icon path. Otherwise uses default icon.	color	Fill color for the icon.	size	Size settings.		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>width</td> <td>Width in pixels.</td> <td>value: numeric</td> </tr> <tr> <td>height</td> <td>Height in pixels.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	width	Width in pixels.	value: numeric	height	Height in pixels.	value: numeric	style	Style for the icon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	rotate	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 5px;"> <p>The following feature is new in Ignition version 8.1.21 Click here to check out the other new features</p> </div> <p>Rotation degree of the specified map marker</p>	tooltip	This marker's tooltip configuration, if tooltip is enabled.		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>P</th> <th>T</th> </tr> </thead> <tbody> <tr> <td>content</td> <td>The tooltip content to display.</td> <td></td> <td></td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>P</th> <th>T</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Text to display.</td> <td>va</td> <td>str</td> </tr> </tbody> </table> </td> <td></td> <td></td> </tr> </tbody> </table>	Name	Description	P	T	content	The tooltip content to display.				<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>P</th> <th>T</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Text to display.</td> <td>va</td> <td>str</td> </tr> </tbody> </table>	Name	Description	P	T	text	Text to display.	va	str		
Name	Description																																																															
name	A unique name that is used to distinguish this marker from another.																																																															
enabled	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 5px;"> <p>The following feature is new in Ignition version 8.1.16 Click here to check out the other new features</p> </div> <p>Determines if the marker is shown on the map. Default value is true.</p>																																																															
lat	Latitudinal coordinate.																																																															
lng	Longitudinal coordinate.																																																															
opacity	Marker opacity.																																																															
icon	Marker icon configuration.																																																															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Icon path. Otherwise uses default icon.</td> </tr> <tr> <td>color</td> <td>Fill color for the icon.</td> </tr> <tr> <td>size</td> <td>Size settings.</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>width</td> <td>Width in pixels.</td> <td>value: numeric</td> </tr> <tr> <td>height</td> <td>Height in pixels.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> </tr> <tr> <td>style</td> <td>Style for the icon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> </tr> <tr> <td>rotate</td> <td> <div style="border: 1px solid orange; padding: 5px; margin-bottom: 5px;"> <p>The following feature is new in Ignition version 8.1.21 Click here to check out the other new features</p> </div> <p>Rotation degree of the specified map marker</p> </td> </tr> </tbody> </table>	Name	Description	path	Icon path. Otherwise uses default icon.	color	Fill color for the icon.	size	Size settings.		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>width</td> <td>Width in pixels.</td> <td>value: numeric</td> </tr> <tr> <td>height</td> <td>Height in pixels.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	width	Width in pixels.	value: numeric	height	Height in pixels.	value: numeric	style	Style for the icon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	rotate	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 5px;"> <p>The following feature is new in Ignition version 8.1.21 Click here to check out the other new features</p> </div> <p>Rotation degree of the specified map marker</p>																																								
Name	Description																																																															
path	Icon path. Otherwise uses default icon.																																																															
color	Fill color for the icon.																																																															
size	Size settings.																																																															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>width</td> <td>Width in pixels.</td> <td>value: numeric</td> </tr> <tr> <td>height</td> <td>Height in pixels.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	width	Width in pixels.	value: numeric	height	Height in pixels.	value: numeric																																																						
Name	Description	Property Type																																																														
width	Width in pixels.	value: numeric																																																														
height	Height in pixels.	value: numeric																																																														
style	Style for the icon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .																																																															
rotate	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 5px;"> <p>The following feature is new in Ignition version 8.1.21 Click here to check out the other new features</p> </div> <p>Rotation degree of the specified map marker</p>																																																															
tooltip	This marker's tooltip configuration, if tooltip is enabled.																																																															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>P</th> <th>T</th> </tr> </thead> <tbody> <tr> <td>content</td> <td>The tooltip content to display.</td> <td></td> <td></td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>P</th> <th>T</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Text to display.</td> <td>va</td> <td>str</td> </tr> </tbody> </table> </td> <td></td> <td></td> </tr> </tbody> </table>	Name	Description	P	T	content	The tooltip content to display.				<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>P</th> <th>T</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Text to display.</td> <td>va</td> <td>str</td> </tr> </tbody> </table>	Name	Description	P	T	text	Text to display.	va	str																																													
Name	Description	P	T																																																													
content	The tooltip content to display.																																																															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>P</th> <th>T</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Text to display.</td> <td>va</td> <td>str</td> </tr> </tbody> </table>	Name	Description	P	T	text	Text to display.	va	str																																																							
Name	Description	P	T																																																													
text	Text to display.	va	str																																																													

view	The view to display as popup content. If configured, this overrides the text property.		ob
	Name	Description	Property Type
	path	Path of view to display.	value: string
	params	Parameters to be passed to the view. Names in this object must match input parameters defined on the view.	object

direction	Direction where to open the tooltip. Possible values are: right, left, top, bottom, center, auto. Dynamically switch between right and left according to the tooltip position on the map.
permanent	Whether to open the tooltip permanently or on a mouseover.
sticky	If true, the tooltip will follow the mouse instead of being fixed at the feature center.
opacity	Tooltip opacity.

popup Popup configuration for this marker.

Name	Description	Property Type
enabled	Enable marker popup.	
content	The popup content to display.	
	Name	Description
	text	Text to display.
	view	The view to display as popup content. If configured, this overrides the text property.
		Name
		Description
		Property Type
	path	Path of view to display.
	params	Parameters to be passed to the view. Names in this object must match input parameters defined on the view.
width		
	Name	Description
	max	Maximum popup width in pixels.
	min	Minimum popup width in pixels.
height	Maximum popup height in pixels.	
pan	Popup pan configuration.	
	Name	Description
	auto	Set it to false if you don't want the map to do panning animation to fit the opened popup.
closeButton	Controls the presence of a close button in the popup.	

autoClose	Set to false if you want to override the default behavior of the popup closing when another opened.
closeOnEscapeKey	Set to false if you want to override the default behavior of the escape key for closing the pc
closeOnClick	Set if you want to override the default behavior of the popup closing when user clicks on th

popup Array of popup layer configurations for this map.

Name	Description																		
enabled	Show popup.																		
lat	Latitudinal coordinate.																		
lng	Longitudinal coordinate.																		
content	The popup content to display. <table border="1" data-bbox="618 747 1495 1167"> <thead> <tr> <th>Name</th> <th>Description</th> <th>P T</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Text to display.</td> <td></td> </tr> <tr> <td>view</td> <td>The view to display as popup content. If configured, this overrides the text property. <table border="1" data-bbox="724 947 1495 1167"> <thead> <tr> <th>Name</th> <th>Description</th> <th>P T</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Path of view to display.</td> <td>va str</td> </tr> <tr> <td>params</td> <td>Parameters to be passed to the view. Names in this object must match input parameters defined on the view.</td> <td>ob</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	P T	text	Text to display.		view	The view to display as popup content. If configured, this overrides the text property. <table border="1" data-bbox="724 947 1495 1167"> <thead> <tr> <th>Name</th> <th>Description</th> <th>P T</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Path of view to display.</td> <td>va str</td> </tr> <tr> <td>params</td> <td>Parameters to be passed to the view. Names in this object must match input parameters defined on the view.</td> <td>ob</td> </tr> </tbody> </table>	Name	Description	P T	path	Path of view to display.	va str	params	Parameters to be passed to the view. Names in this object must match input parameters defined on the view.	ob	
Name	Description	P T																	
text	Text to display.																		
view	The view to display as popup content. If configured, this overrides the text property. <table border="1" data-bbox="724 947 1495 1167"> <thead> <tr> <th>Name</th> <th>Description</th> <th>P T</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Path of view to display.</td> <td>va str</td> </tr> <tr> <td>params</td> <td>Parameters to be passed to the view. Names in this object must match input parameters defined on the view.</td> <td>ob</td> </tr> </tbody> </table>	Name	Description	P T	path	Path of view to display.	va str	params	Parameters to be passed to the view. Names in this object must match input parameters defined on the view.	ob									
Name	Description	P T																	
path	Path of view to display.	va str																	
params	Parameters to be passed to the view. Names in this object must match input parameters defined on the view.	ob																	
width	Width settings for the popup. <table border="1" data-bbox="618 1234 1195 1377"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>max</td> <td>Maximum popup width in pixels.</td> <td>value: numeric</td> </tr> <tr> <td>min</td> <td>Minimum popup width in pixels.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	max	Maximum popup width in pixels.	value: numeric	min	Minimum popup width in pixels.	value: numeric									
Name	Description	Property Type																	
max	Maximum popup width in pixels.	value: numeric																	
min	Minimum popup width in pixels.	value: numeric																	
height	Maximum popup height in pixels.																		
pan	Popup pan configuration. <table border="1" data-bbox="618 1514 1495 1608"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>auto</td> <td>Set it to false if you don't want the map to do panning animation to fit the opened popup</td> </tr> </tbody> </table>	Name	Description	auto	Set it to false if you don't want the map to do panning animation to fit the opened popup														
Name	Description																		
auto	Set it to false if you don't want the map to do panning animation to fit the opened popup																		
closeButton	Controls the presence of a close button in the popup.																		
autoClose	Set to false if you want to override the default behavior of the popup closing when another popup is																		
closeOnEscapeKey	Set to false if you want to override the default behavior of the escape key for closing the popup.																		
closeOnClick	Set if you want to override the default behavior of the popup closing when user clicks on the map.																		

view Array of view layer configurations for this map.

Name	Description
path	Path of view to display.
params	Parameters to be passed to the view. Names in this object must match input parameters c
lat	Latitudinal coordinate.
lng	Longitudinal coordinate.
transparentBackground	If enabled, disables the default background color.
shadow	If enabled, displays a box shadow around the view.

other Other map layers.

Name	Description
geoJSON	<p>GeoJSON objects to include as a feature layer. Each element should be a geoJSON FeatureCollection. This property binding to a URL containing the desired geoJSON data, and a map transform to process the results.</p> <p>For more information on the geoJSON format, see http://geojson.org/.</p> <p>It is not the intent that geoJSON shapes are manually added to the component. Instead they In addition, the following geoJSON layer: https://geojson.io/.</p> <p>See Perspective Map - Adding GeoJSON Shapes for more information.</p> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p>The following feature is new in Ignition version 8.1.15 Click here to check out the other new features</p> </div>

Name	Description																												
styleOptions	<p>Style options for a FeatureCollection layer. Additional styleOptions objects can be added to individual fea FeatureCollection.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>stroke</td> <td>Whether to draw stroke along the path. Set it to <code>false</code> to disable borders on polygons or circ</td> </tr> <tr> <td>color</td> <td>Stroke color.</td> </tr> <tr> <td>weight</td> <td>Stroke width in pixels.</td> </tr> <tr> <td>opacity</td> <td>Stroke opacity.</td> </tr> <tr> <td>lineCap</td> <td>A string that defines the shape to be used at the end of the stroke. Options include butt, round square, and inherit. Default value is inherit.</td> </tr> <tr> <td>lineJoin</td> <td>A string that defines shape to be used at the corners of the stroke. Options include miter, round bevel, and inherit. Default value is inherit.</td> </tr> <tr> <td>dashArray</td> <td>A string that defines the stroke dash pattern.</td> </tr> <tr> <td>dashOffset</td> <td>A string that defines the distance into the dash pattern to start the dash.</td> </tr> <tr> <td>fill</td> <td>Whether to fill the path with color. Set it to <code>false</code> to disable filling on polygons or circles.</td> </tr> <tr> <td>fillColor</td> <td>Fill color. Defaults to the value of the <code>color</code> property.</td> </tr> <tr> <td>fillOpacity</td> <td>Fill opacity.</td> </tr> <tr> <td>fillRule</td> <td>A string that defines how the inside of a shape is determined.</td> </tr> <tr> <td>interact</td> <td>If <code>false</code>, the layer will not emit mouse events and will act as a part of the underlying map.</td> </tr> </tbody> </table>	Name	Description	stroke	Whether to draw stroke along the path. Set it to <code>false</code> to disable borders on polygons or circ	color	Stroke color.	weight	Stroke width in pixels.	opacity	Stroke opacity.	lineCap	A string that defines the shape to be used at the end of the stroke. Options include butt, round square, and inherit. Default value is inherit.	lineJoin	A string that defines shape to be used at the corners of the stroke. Options include miter, round bevel, and inherit. Default value is inherit.	dashArray	A string that defines the stroke dash pattern .	dashOffset	A string that defines the distance into the dash pattern to start the dash .	fill	Whether to fill the path with color. Set it to <code>false</code> to disable filling on polygons or circles.	fillColor	Fill color. Defaults to the value of the <code>color</code> property.	fillOpacity	Fill opacity.	fillRule	A string that defines how the inside of a shape is determined.	interact	If <code>false</code> , the layer will not emit mouse events and will act as a part of the underlying map.
Name	Description																												
stroke	Whether to draw stroke along the path. Set it to <code>false</code> to disable borders on polygons or circ																												
color	Stroke color.																												
weight	Stroke width in pixels.																												
opacity	Stroke opacity.																												
lineCap	A string that defines the shape to be used at the end of the stroke. Options include butt, round square, and inherit. Default value is inherit.																												
lineJoin	A string that defines shape to be used at the corners of the stroke. Options include miter, round bevel, and inherit. Default value is inherit.																												
dashArray	A string that defines the stroke dash pattern .																												
dashOffset	A string that defines the distance into the dash pattern to start the dash .																												
fill	Whether to fill the path with color. Set it to <code>false</code> to disable filling on polygons or circles.																												
fillColor	Fill color. Defaults to the value of the <code>color</code> property.																												
fillOpacity	Fill opacity.																												
fillRule	A string that defines how the inside of a shape is determined.																												
interact	If <code>false</code> , the layer will not emit mouse events and will act as a part of the underlying map.																												

				ve
	hideViewMarkersZoom	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 5px;"> <p>The following feature is new in Ignition version 8.1.16 Click here to check out the other new features</p> </div> <p>Hides view markers while they are being repositioned while zooming in or out. Default value is <code>true</code>.</p>		
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellan			

Map Parameters

Because this component uses the Leaflet library, there are several common objects you'll want to be aware of when using the the callable methods on this component. Each of these "objects" is just a Python dictionary with specific keys. Documentation for these objects are provided for convenience.

LatLng

An object representing a point at a certain latitude and longitude. For the purposes of interacting with the map component, this object is typically a Python dictionary.

```
# Example
{ lat:50, lng:30 }
```

Option	Type	Default	Description
lat	Numeric	None	Numerical value representing a latitude value.
lng	Numeric	None	Numerical value representing a longitude value.

PanOptions

A Python dictionary containing keys that can modify the panning behavior on the Map component. The contents of this dictionary are listed below, but the original explanation can be [found in Leaflet's documentation](#).

Option	Type	Default	Description
animate	Boolean	false	If <code>true</code> , panning will always be animated if possible. If <code>false</code> , it will not animate panning, either resetting the map view if panning more than a screen away, or just setting a new offset for the map pane (except for <code>panBy</code> which always does the latter).
duration	Numeric	0.25	Duration of animated panning, in seconds.
easing	Numeric	0.25	The curvature factor of panning animation easing (third parameter of the Cubic Bezier curve). 1.0 means linear animation, and the smaller this number, the more bowed the curve.
noMoveStart	Boolean	false	If <code>true</code> , panning won't fire <code>movestart</code> event on start (used internally for panning inertia).

ZoomOptions

A Python dictionary containing keys that can modify the zooming behavior on the Map. The contents of this dictionary are listed below, but the original explanation can be [found in Leaflet's documentation](#).

Option	Type	Default	Description
animate	Boolean	false	If <code>true</code> , panning will always be animated if possible. If <code>false</code> , it will not animate panning, either resetting the map view if panning more than a screen away, or just setting a new offset for the map pane (except for <code>panBy</code> which always does the latter).

FitBounds

A Python dictionary containing keys that can modify the zooming behavior on the Map. The contents of this dictionary are listed below, but the original explanation can be [found in Leaflet's documentation](#).

--	--	--	--

Option	Type	Default	Description
paddingTopLeft	Point	[0,0]	Sets the amount of padding in the top left corner of a map container that shouldn't be accounted for when setting the view to fit bounds. Useful if you have some control overlays on the map like a sidebar and you don't want them to obscure objects you're zooming to.
paddingBottomRight	Point	[0,0]	Similar to paddingTopLeft , except for the bottom right corner of the map.
padding	Point	[0,0]	A setting that allows both the top left and bottom right padding to use the same value.
maxZoom	Numeric	null	The maximum possible zoom to use.

Scripting

See the [Perspective - Map Scripting page](#) for the full list of scripting functions available for this component.

Examples

Additional Layers

You can add any number of layers to the Perspective Map component by adding tiles under **layers.raster.tile**.

You will need to obtain an API key to call most maps. Many API keys are available for free or with a subscription to a service such as Google Maps Platform. The example below makes use of the [Open Weather Map API](#).

Open Weather Map API Endpoint

```
https://tile.openweathermap.org/map/{layer}/{z}/{x}/{y}.png?appid={API key}
```

For this example, we only need to update two parameters in the endpoint URL:

Parameter	Description
{layer}	The name of the layer to display, such as <code>clouds_new</code> or <code>temp_new</code> .
{API key}	Your unique API key.



Property	Value	Description												
layers.raster. tile.0	<table border="1"> <thead> <tr> <th>Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>url</td> <td><code>https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png</code></td> </tr> <tr> <td>options</td> <td> <table border="1"> <thead> <tr> <th>Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>opacity</td> <td>1</td> </tr> <tr> <td>zIndex</td> <td>0</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	Property	Value	url	<code>https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png</code>	options	<table border="1"> <thead> <tr> <th>Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>opacity</td> <td>1</td> </tr> <tr> <td>zIndex</td> <td>0</td> </tr> </tbody> </table>	Property	Value	opacity	1	zIndex	0	Default map tile
	Property	Value												
	url	<code>https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png</code>												
	options	<table border="1"> <thead> <tr> <th>Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>opacity</td> <td>1</td> </tr> <tr> <td>zIndex</td> <td>0</td> </tr> </tbody> </table>	Property	Value	opacity	1	zIndex	0						
Property	Value													
opacity	1													
zIndex	0													
layers.raster. tile.1	<table border="1"> <thead> <tr> <th>Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>url</td> <td><code>https://tile.openweathermap.org/map/clouds_new/{z}/{x}/{y}.png?appid=YOUR_API_KEY</code></td> </tr> <tr> <td>options</td> <td> <table border="1"> <thead> <tr> <th>Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>opacity</td> <td>1</td> </tr> <tr> <td>zIndex</td> <td>1</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	Property	Value	url	<code>https://tile.openweathermap.org/map/clouds_new/{z}/{x}/{y}.png?appid=YOUR_API_KEY</code>	options	<table border="1"> <thead> <tr> <th>Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>opacity</td> <td>1</td> </tr> <tr> <td>zIndex</td> <td>1</td> </tr> </tbody> </table>	Property	Value	opacity	1	zIndex	1	Clouds map tile
	Property	Value												
	url	<code>https://tile.openweathermap.org/map/clouds_new/{z}/{x}/{y}.png?appid=YOUR_API_KEY</code>												
	options	<table border="1"> <thead> <tr> <th>Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>opacity</td> <td>1</td> </tr> <tr> <td>zIndex</td> <td>1</td> </tr> </tbody> </table>	Property	Value	opacity	1	zIndex	1						
Property	Value													
opacity	1													
zIndex	1													
layers.raster. tile.2	<table border="1"> <thead> <tr> <th>Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>url</td> <td><code>https://tile.openweathermap.org/map/temp_new/{z}/{x}/{y}.png?appid=YOUR_API_KEY</code></td> </tr> <tr> <td>options</td> <td> <table border="1"> <thead> <tr> <th>Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>opacity</td> <td>0.6</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	Property	Value	url	<code>https://tile.openweathermap.org/map/temp_new/{z}/{x}/{y}.png?appid=YOUR_API_KEY</code>	options	<table border="1"> <thead> <tr> <th>Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>opacity</td> <td>0.6</td> </tr> </tbody> </table>	Property	Value	opacity	0.6	Temperature map tile		
	Property	Value												
	url	<code>https://tile.openweathermap.org/map/temp_new/{z}/{x}/{y}.png?appid=YOUR_API_KEY</code>												
	options	<table border="1"> <thead> <tr> <th>Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>opacity</td> <td>0.6</td> </tr> </tbody> </table>	Property	Value	opacity	0.6								
Property	Value													
opacity	0.6													

The code block below contains the data we'll use for this example.

Example GeoJSON Data

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {},
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [
            -121.14997386932373,
            38.655488159953
          ],
          [
            -121.15048885345459,
            38.655689239321134
          ],
          [
            -121.15031719207764,
            38.65699624145518
          ],
          [
            -121.14920139312744,
            38.65699624145518
          ],
          [
            -121.14877223968504,
            38.65816917178303
          ],
          [
            -121.14551067352294,
            38.658135659754485
          ],
          [
            -121.14551067352294,
            38.655521673220235
          ],
          [
            -121.14684104919432,
            38.65434869953815
          ],
          [
            -121.14834308624268,
            38.6554546466701
          ],
          [
            -121.14975929260254,
            38.655521673220235
          ],
          [
            -121.14817142486571,
            38.6554546466701
          ]
        ]
      }
    },
    {
      "type": "Feature",
      "properties": {
      },
      "geometry": {
        "type": "LineString",
        "coordinates": [

```

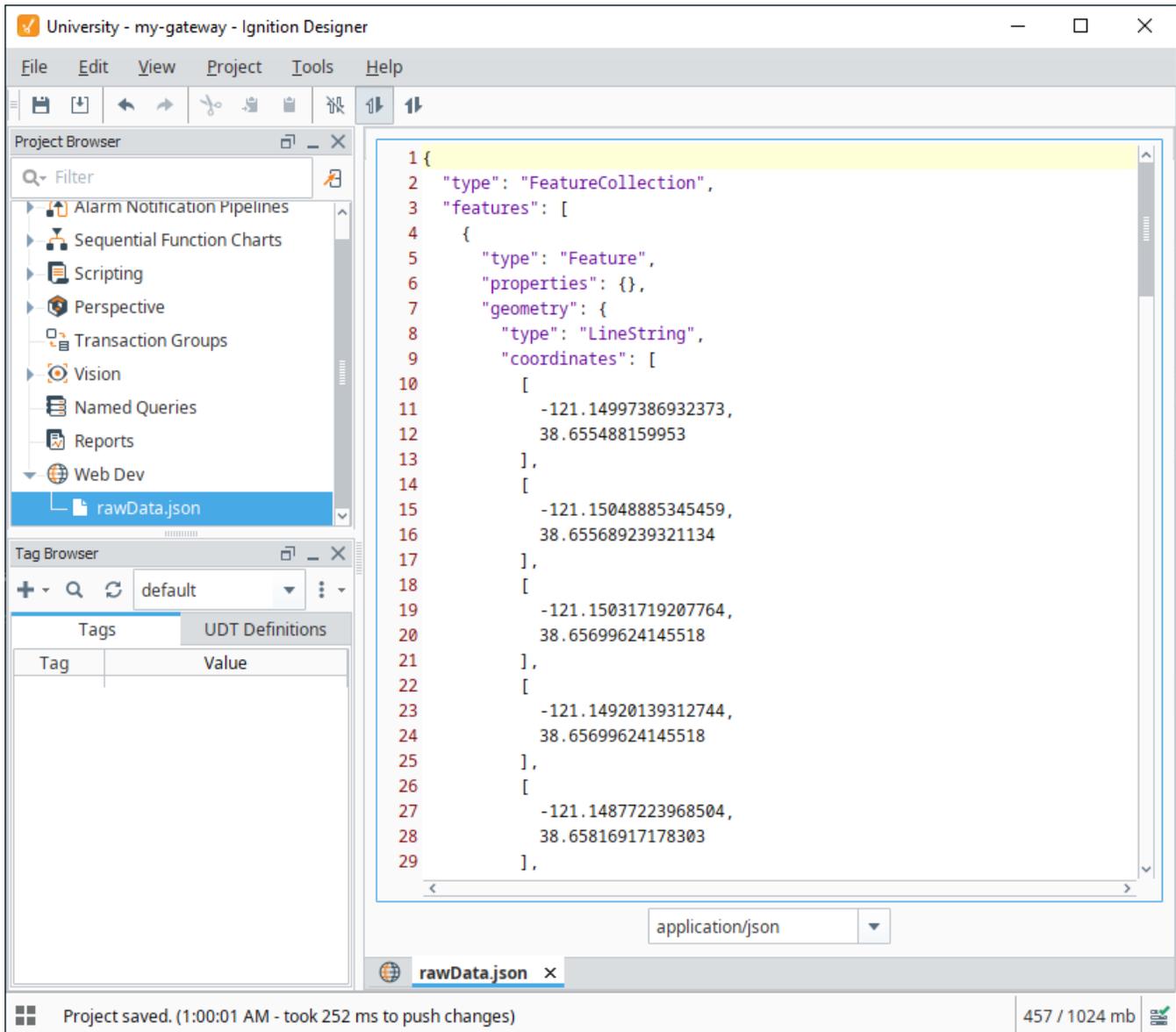
```
[
  [-121.16469383239745,
    38.66983040550309
  ],
  [
    [-121.16443634033203,
      38.670265989507826
    ],
    [
      [-121.16044521331787,
        38.66895922954322
      ],
      [
        [-121.15975856781006,
          38.66755192290758
        ],
        [
          [-121.15967273712158,
            38.666580195039295
          ],
          [
            [-121.16053104400635,
              38.66490477119321
            ],
            [
              [-121.16224765777588,
                38.665005297729465
              ],
              [
                [-121.16340637207033,
                  38.66580950493899
                ],
                [
                  [-121.16310596466064,
                    38.66664721117775
                  ],
                  [
                    [-121.16383552551268,
                      38.66684825921672
                    ],
                    [
                      [-121.16336345672606,
                        38.66785349094385
                      ],
                      [
                        [-121.16293430328369,
                          38.66781998344699
                        ],
                        [
                          [-121.16284847259521,
                            38.66889221556877
                          ],
                          [
                            [-121.16482257843018,
                              38.669662872487926
                            ]
                          ]
                        ]
                      ]
                    ]
                  ]
                ]
              ]
            ]
          ]
        ]
      ]
    ]
  ],
  [
    [
      [-121.17366313934326,
        38.65706326656277
      ],
      ]
    ]
  ]
],
{
  "type": "Feature",
  "properties": {},
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [
          [-121.17366313934326,
            38.65706326656277
          ],
          ]
        ]
      ]
    ]
  }
}
```

```
        -121.17001533508302,  
        38.65706326656277  
    ],  
    [  
        -121.17001533508302,  
        38.659844753217264  
    ],  
    [  
        -121.17366313934326,  
        38.659844753217264  
    ],  
    [  
        -121.17366313934326,  
        38.65706326656277  
    ]  
    ]  
    ]  
    }  
    }  
    ]  
    }
```

Store the Data

Next we can store the data as a [Web Dev](#) resource. If your data is being provided by an existing endpoint, then skip this section.

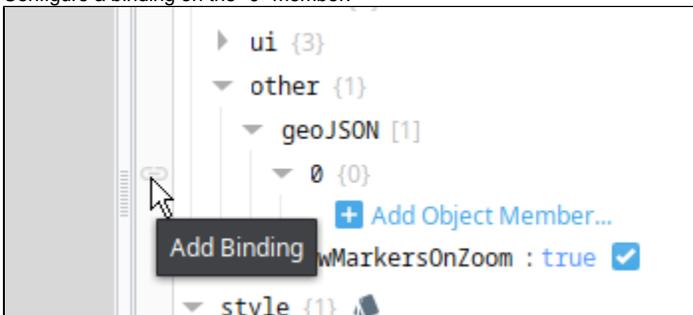
1. In the Designer's Project Browser, navigate to the Web Dev section and create a new **Text Resource**. Name this resource **rawData.json**.
2. At the bottom of the workspace, change the dropdown to **application/json**.
3. Paste your geoJSON data into the text resource.
4. Save the Project.



Pass the Data to the Map

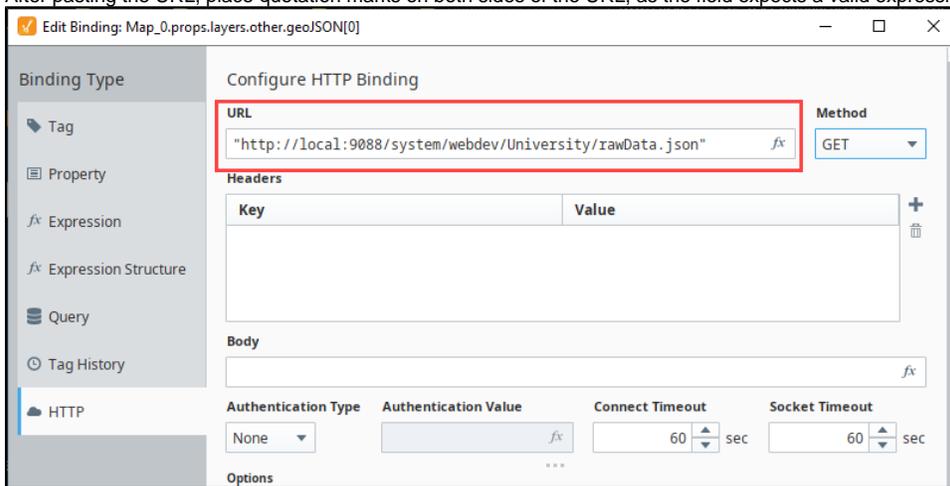
Now that we have some data we can hand it off to our map component. We'll use a HTTP binding.

1. In the Designer, navigate to, or create, a Perspective view.
2. On the view, create a Map component.
3. With the Map selected, locate the **props.layers.other.geoJSON** property. Click the **Add Array Element** button, which adds a "0" member.
4. Configure a binding on the "0" member.



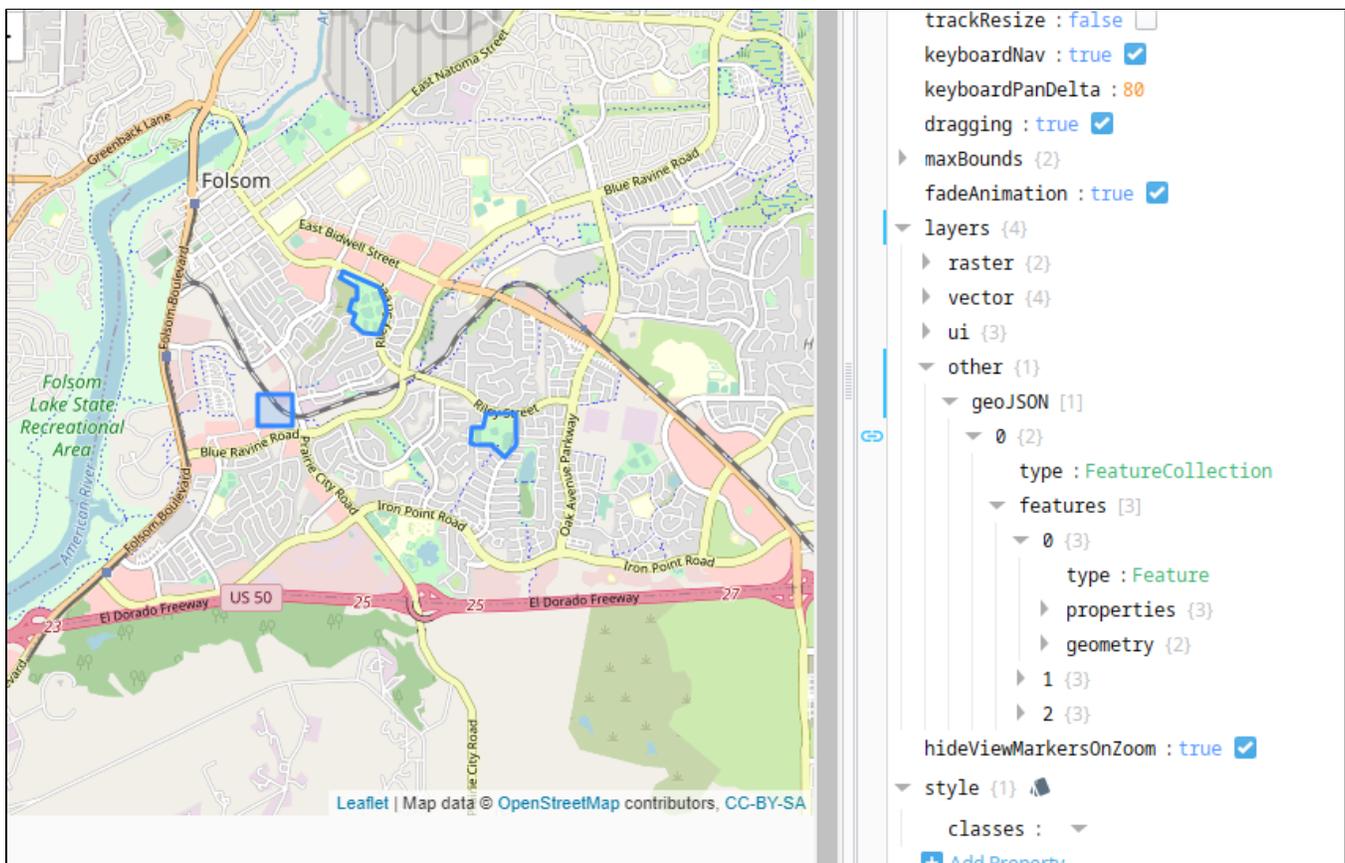
5. Select **HTTP** as the binding type.
6. For the URL property, enter the endpoint housing your data. If you're using a Web Dev resource as demonstrated earlier on this page then you can [copy the mounted path](#) from the Project Browser, but make sure to use a full URL by adding in "http://" or "https://".

7. After pasting the URL, place quotation marks on both sides of the URL, as the field expects a valid expression.



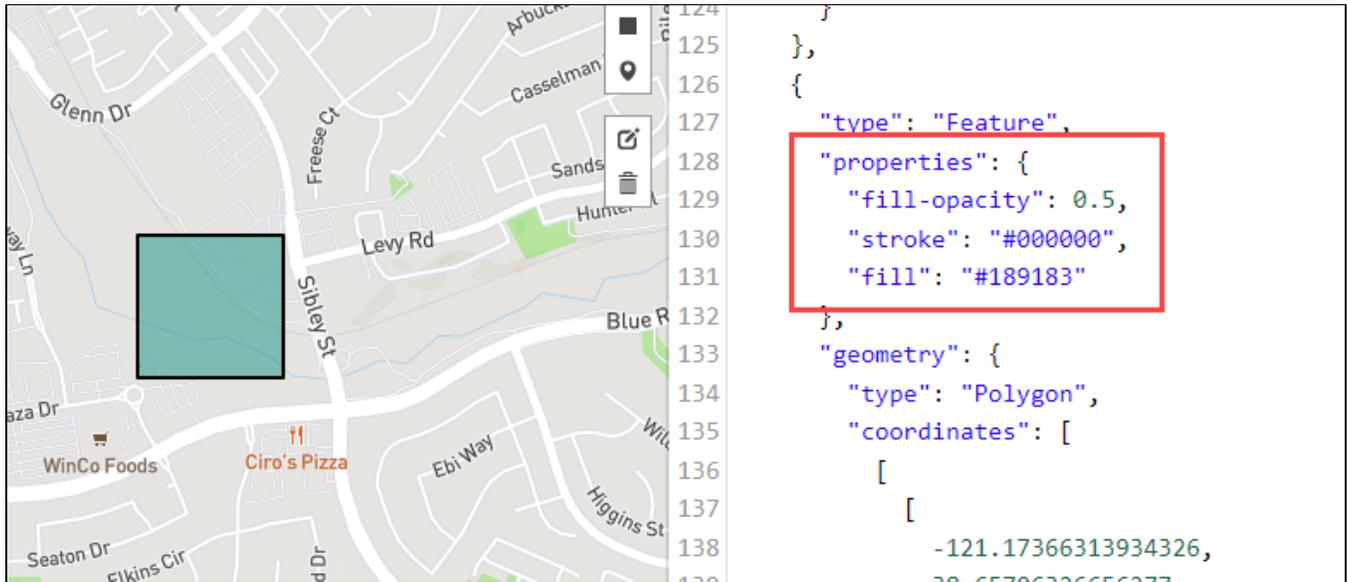
8. Click **OK** to close the binding window.

Now we have some features on our map that are driven by our geoJSON data.



Modify the Incoming Data

Our Map is now showing features. In a real world scenario you may need to modify the incoming data so that it is formatted in a way that the map component expects. For example, the [geojson.io](#) tool allows you to define colors for the strokes and fill on each feature. However, it defines these features under a "properties" object, which our Map component cannot utilize since it is expecting any styling attributes to be listed under a "styleOptions" object.



For this next example, we will demonstrate how to modify incoming data with a script transform. This example will focus on styling attributes, but could be expanded to account for other arbitrary geoJSON property data.

Update our GeoJSON Data

The code block below contains the new geoJSON data this example will use. Copy the contents of the code block, and paste them into our Web Dev text resource. Save the project after pasting.

Updated GeoJSON Data

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {
        "stroke": "#c7170a"
      },
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [
            -121.14997386932373,
            38.655488159953
          ],
          [
            -121.15048885345459,
            38.655689239321134
          ],
          [
            -121.15031719207764,
            38.65699624145518
          ],
          [
            -121.14920139312744,
            38.65699624145518
          ],
          [
            -121.14877223968504,
            38.65816917178303
          ],
          [
            -121.14551067352294,
            38.658135659754485
          ]
        ]
      }
    }
  ]
}
```

```
[
  [-121.14551067352294,
    38.655521673220235
  ],
  [
    [-121.14684104919432,
      38.65434869953815
    ],
    [
      [-121.14834308624268,
        38.6554546466701
      ],
      [
        [-121.14975929260254,
          38.655521673220235
        ],
        [
          [-121.14817142486571,
            38.6554546466701
          ]
        ]
      ]
    ]
  ],
  {
    "type": "Feature",
    "properties": {
      "stroke": "#0f3df5"
    },
    "geometry": {
      "type": "LineString",
      "coordinates": [
        [
          [-121.16469383239745,
            38.66983040550309
          ],
          [
            [-121.16443634033203,
              38.670265989507826
            ],
            [
              [-121.16044521331787,
                38.66895922954322
              ],
              [
                [-121.15975856781006,
                  38.66755192290758
                ],
                [
                  [-121.15967273712158,
                    38.666580195039295
                  ],
                  [
                    [-121.16053104400635,
                      38.66490477119321
                    ],
                    [
                      [-121.16224765777588,
                        38.665005297729465
                      ],
                      [
                        [-121.16340637207033,
                          38.66580950493899
                        ],
                        [
                          [-121.16310596466064,
                            38.66664721117775
                          ],
                          [
                            [-121.16383552551268,
                              38.66684825921672
                            ]
                          ]
                        ]
                      ]
                    ]
                  ]
                ]
              ]
            ]
          ]
        ]
      ]
    }
  ],
  {
    "type": "Feature",
    "properties": {
      "stroke": "#0f3df5"
    },
    "geometry": {
      "type": "LineString",
      "coordinates": [
        [
          [-121.16469383239745,
            38.66983040550309
          ],
          [
            [-121.16443634033203,
              38.670265989507826
            ],
            [
              [-121.16044521331787,
                38.66895922954322
              ],
              [
                [-121.15975856781006,
                  38.66755192290758
                ],
                [
                  [-121.15967273712158,
                    38.666580195039295
                  ],
                  [
                    [-121.16053104400635,
                      38.66490477119321
                    ],
                    [
                      [-121.16224765777588,
                        38.665005297729465
                      ],
                      [
                        [-121.16340637207033,
                          38.66580950493899
                        ],
                        [
                          [-121.16310596466064,
                            38.66664721117775
                          ],
                          [
                            [-121.16383552551268,
                              38.66684825921672
                            ]
                          ]
                        ]
                      ]
                    ]
                  ]
                ]
              ]
            ]
          ]
        ]
      ]
    }
  ]
],
```

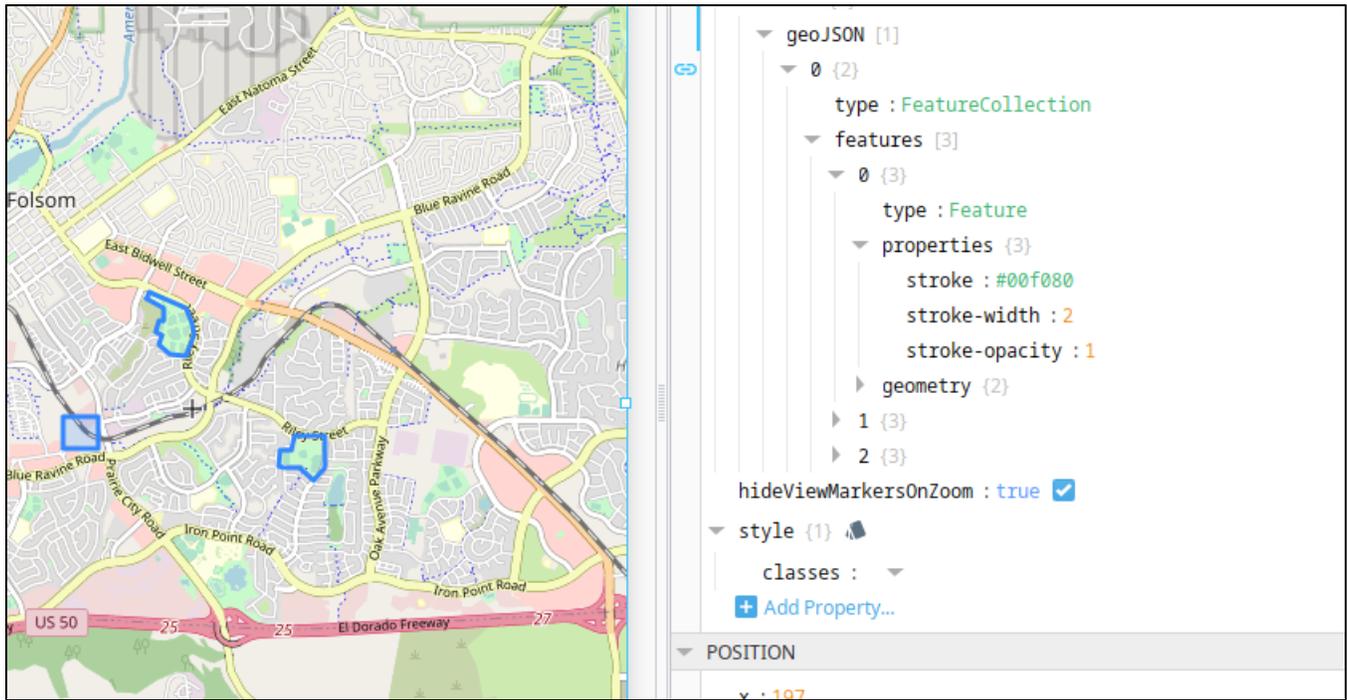
```

    [
      -121.16336345672606,
      38.66785349094385
    ],
    [
      -121.16293430328369,
      38.66781998344699
    ],
    [
      -121.16284847259521,
      38.66889221556877
    ],
    [
      -121.16482257843018,
      38.669662872487926
    ]
  ]
}
},
{
  "type": "Feature",
  "properties": {
    "fill-opacity": 0.5,
    "stroke": "#000000",
    "fill": "#189183"
  },
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [
          -121.17366313934326,
          38.65706326656277
        ],
        [
          -121.17001533508302,
          38.65706326656277
        ],
        [
          -121.17001533508302,
          38.659844753217264
        ],
        [
          -121.17366313934326,
          38.659844753217264
        ],
        [
          -121.17366313934326,
          38.65706326656277
        ]
      ]
    ]
  }
}
]
}
}
}

```

Update our Binding

Now that our map is using the updated data, you should see that the "properties" object under each feature contains some styling properties. However, they are being ignored. This is because we need to add a "styleOptions" object in each feature, and relocate the styling configurations so they're under the new object.



In addition, we should also add a shared styleOptions object as a peer to all of our features. This allows each feature to only override the style settings they need to. In other words, we want something like this:

The image shows a JSON tree structure for a geoJSON object. The root is `geoJSON [1]`, which contains an array of features. The first feature (index 0) is a `FeatureCollection` with three features. The first feature (index 0) is a `Feature` with `properties {3}` and `geometry {2}`. Its `styleOptions {0}` is highlighted with a red box and an arrow pointing to the text "Applies overrides to just feature '0'". Below this, the `styleOptions {13}` for the entire collection is shown, also highlighted with a red box and an arrow pointing to "Applies base styling to ALL features".

```
geoJSON [1]
├── 0 {3}
│   ├── type : FeatureCollection
│   └── features [3]
│       ├── 0 {4}
│       │   ├── type : Feature
│       │   ├── properties {3}
│       │   ├── geometry {2}
│       │   └── styleOptions {0}
│       │       └── + Add Object Member...
│       ├── 1 {3}
│       └── 2 {3}
└── styleOptions {13}
    ├── stroke : true 
    ├── color : 
    ├── weight : 1
    ├── opacity : 1
    ├── lineCap : inherit ▾
    ├── lineJoin : inherit ▾
    ├── dashArray :
    ├── dashOffset :
    ├── fill : true 
    ├── fillColor : 
    ├── fillOpacity : 1
    ├── fillRule : inherit ▾
    └── interactive : true 
```

We can do this automatically with a Script Transform.

1. Select the Map component.
2. Locate the binding on `props.layers.other.geoJSON.0`, and click the link icon to open the binding window

3. Click the **Add Transform** button

The screenshot shows the 'Configure HTTP Binding' dialog box. On the left, there is a sidebar with 'Binding Type' options: Tag, Property, Expression, Expression Structure, Query, Tag History, and HTTP (selected). Below the sidebar is a 'Remove Binding' button. The main area is titled 'Configure HTTP Binding' and contains the following fields:

- URL:** "http://host.docker.internal:9088/system/webdev/University/rawData.jsc" with an 'fx' icon.
- Method:** A dropdown menu set to 'GET'.
- Headers:** A table with 'Key' and 'Value' columns, currently empty.
- Body:** A text input field with an 'fx' icon.
- Authentication Type:** A dropdown menu set to 'None'.
- Authentication Value:** A text input field with an 'fx' icon.
- Connect Timeout:** A spinner set to '60' with 'sec' next to it.
- Socket Timeout:** A spinner set to '60' with 'sec' next to it.
- Options:** A group of checkboxes: 'Enabled' (checked), 'Overlay Opt-Out' (unchecked), 'Allow Cookies' (checked), 'Cache & Share' (checked), and 'Polling' (unchecked) with a text input field and 'sec' next to it.

At the bottom of the main area, there is a button labeled 'Add Transform +' which is highlighted with a red rectangle. Below this is a 'Binding Preview' section with a checked checkbox and the text 'HTTP' and '({\"type\":\"FeatureCollection\"...}'. At the very bottom of the dialog are 'OK', 'Cancel', and 'Apply' buttons.

4. Select "Script" and click **Add Transform >**

The screenshot shows the 'Select Transform' dialog box. On the left, there is a sidebar with four options: 'Map', 'Format', 'Script' (selected), and 'Expression'. The main area is titled 'Script Transform' and contains the following text:

Script Transform

This transform will take the input value and call a scripting function with the input value as arguments to the function.

At the bottom of the dialog are 'Cancel' and 'Add Transform >' buttons. The 'Add Transform >' button is highlighted.

5. Locate the first line of editable code. Replace that with the code below.

```
# Define a base style for all features, and add them as an object at the root of our results.
# The keys below are based upon the expected keys that the Map component supports.
value["styleOptions"] = {
  "stroke": True,
  "color": "",
  "weight": 2,
  "opacity": 1,
  "lineCap": "inherit",
  "lineJoin": "inherit",
  "dashArray": "",
  "dashOffset": "",
  "fill": False,
```

```

        "fillColor": "",
        "fillOpacity": 1,
        "fillRule": "inherit",
        "interactive": True
    }

#Here we iterate all of the features within our data.
for i in value["features"]:

    # Inside each feature, we can examine each key in "properties", and add custom
styling
    i["styleOptions"] = {}
    if i["properties"].has_key("stroke"):
        i["styleOptions"]["color"] = i["properties"]["stroke"]

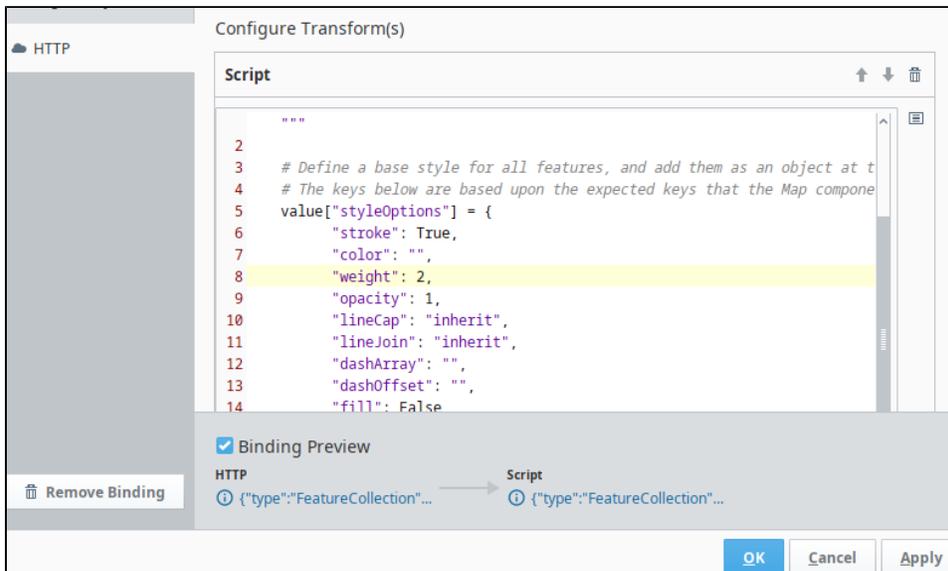
    # Note that we need to both enable fill and specify the fill color if the feature has
"fill"
    # under its raw properties key. This is because our base style above set "fill" to
False.
    if i["properties"].has_key("fill"):
        i["styleOptions"]["fillColor"] = i["properties"]["fill"]
        i["styleOptions"]["fill"] = True

    # Here we also need to change "fill-opacity" to "fillOpacity", to match the map's
default styleOptions keys.
    if i["properties"].has_key("fill-opacity"):
        i["styleOptions"]["fillOpacity"] = i["properties"]["fill-opacity"]

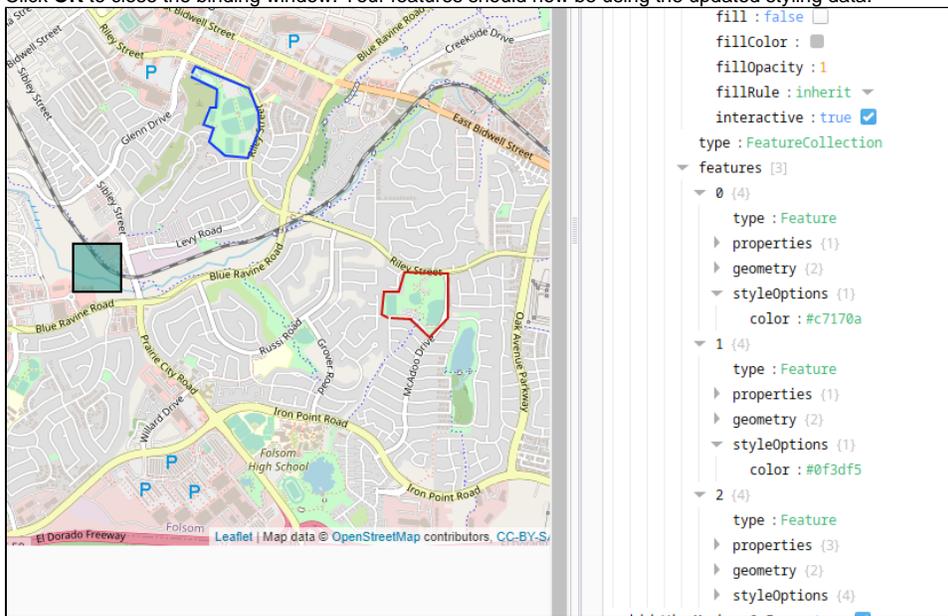
return value

```

Note: Make sure the code is properly indented after pasting. No visible characters should appear on the far left of any line.



6. Click **OK** to close the binding window. Your features should now be using the updated styling data.



Editor notes are only visible to logged in users

Original code example from the PR. I tried to come up with a simplified version in the example above, but figured we could keep this one hidden in case we need it.

```
from com.inductiveautomation.ignition.common.script.adapters import PyJsonObjectAdapter
from org.python.core import PyArray

# This function is how the Map component and Leaflet traverses and determines if a GeoJSON object
# qualifies as a Feature object. It is generic. If you know
# what the layer looks like, i.e. a FeatureCollection of Point objects, you definitely don't need this.
def walkFeatures(geojson, operate):
    features = geojson if isinstance(geojson, PyArray) else geojson['features'] if isinstance(geojson,
PyJsonObjectAdapter) and isinstance(geojson['features'], PyArray) else None
    if features:
        for idx in range(len(features)):
            feature = features[idx]
            if isinstance(feature, PyJsonObjectAdapter):
                if feature.has_key('features'):
                    walkFeatures(feature['features'], operate)
                elif feature.has_key('geometry') or feature.has_key('geometries') or feature.has_key
('coordinates'):
                    operate(feature)

            elif isinstance(geojson, PyJsonObjectAdapter):
                if feature.has_key('geometry') or feature.has_key('geometries') or feature.has_key
('coordinates'):
                    operate(geojson)

# Operates on a feature, in this example, all of the features of this layer are expected to be Point
# objects, which we configure as Leaflet markers
def operateOnFeature(feature):
    if isinstance(feature, PyJsonObjectAdapter):

        marker = {
            'render': 'icon',
            'icon': {
                'path': 'material/location_on',
                'size': {
                    'width': 36,
                    'height': 36
                },
            },
            'color': '#9bfa03'
```

```

    }
  }

  if feature.has_key('properties'):
    properties = feature['properties']
    LOCATION_TYPE = properties.get('LOCATION_TYPE')
    LOCATION_NAME = properties.get('LOCATION_NAME')
    ADDRESS_LINE1 = properties.get('ADDRESS_LINE1')
    if LOCATION_TYPE == 'Headquarters':
      marker['icon']['color'] = '#ff0000'
    marker['tooltip'] = {
      'content': {
        'text': LOCATION_NAME
      }
    }
    marker['popup'] = {
      'content': {
        'text': properties
      }
    }
    feature['marker'] = marker
  walkFeatures(value, operateOnFeature)
  return value

```

Adding Points

The geoJSON standard accounts for "points", which can be rendered on the map as icon based markers or simple circles. The following JSON demonstrates how to define a simple point. Note that **geometry.type** is set to "Point".

```

{
  "type": "Feature",
  "properties": {},
  "geometry": {
    "type": "Point",
    "coordinates": [
      "-121.16945743560791",
      "38.6632293081453"
    ]
  }
}

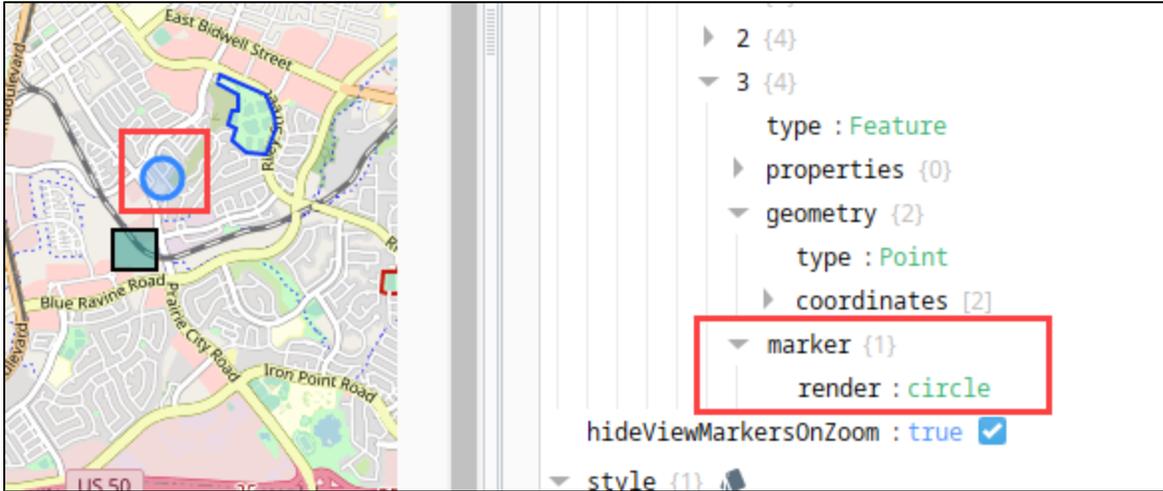
```

A "marker" object can be added to points which allows for additional configuration options. For example, the following causes the point to be rendered as a circle instead of an icon.

```

{
  "type": "Feature",
  "properties": {},
  "geometry": {
    "type": "Point",
    "coordinates": [
      "-121.16945743560791",
      "38.6632293081453"
    ]
  },
  "marker": {
    "render": "circle"
  }
}

```



Customizing Points

Points are rendered using the leaflet library, and can utilize options available in the leaflet documentation. The sections below demonstrate some available functionality. Each sub heading describes an additional object that can be added under a marker object.

Options Object

Additional settings for the marker can be applied on an **options** object under the **marker** object. A list of available options can be found in [leaflet's documentation](#). The example below simply changes the opacity on the marker.

```
[
  {
    "type": "Feature",
    "properties": {},
    "geometry": {
      "type": "Point",
      "coordinates": [
        "-121.16945743560791",
        "38.6632293081453"
      ]
    },
    "marker": {
      "options": {
        "opacity": 0.5
      }
    }
  }
]
```

Icon Object

When rendered as an icon, a **icon** object can be added to configure the icon.

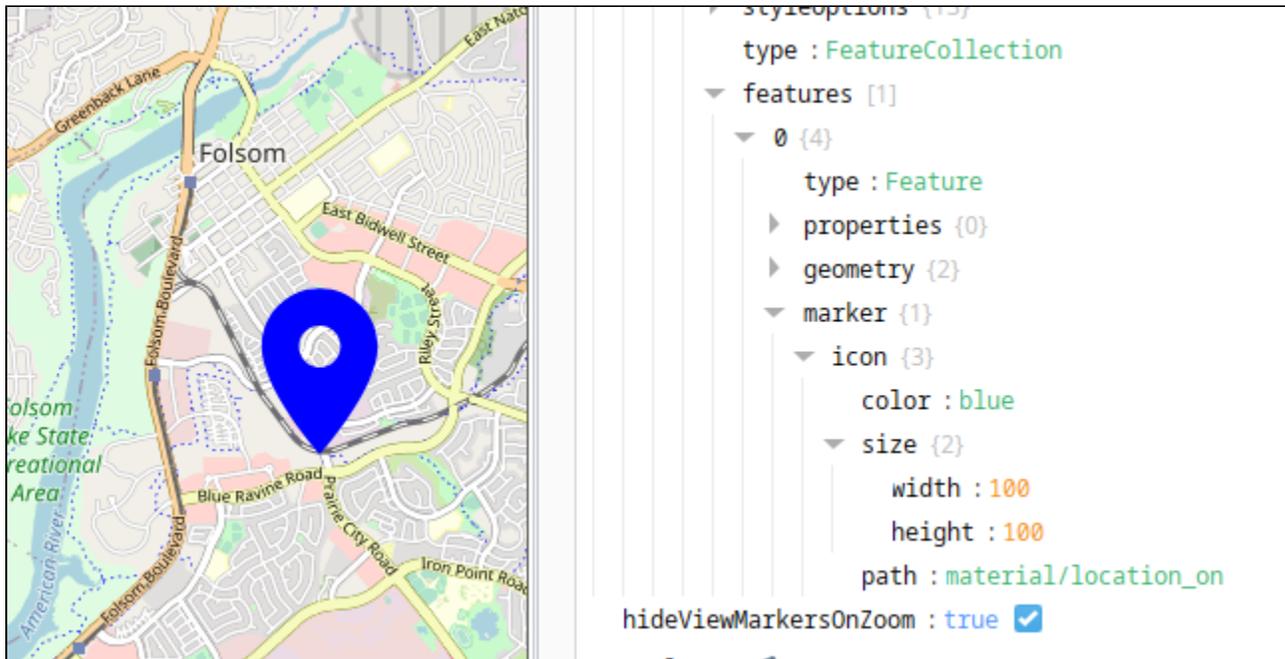
Setting	Description
color	A color, as represented by a string. Determines the color of the icon.
size	An array with two values: width and height. Determines the size of the icon.
path	A path to an icon. Uses the same icon library and pathing notation as the Icon component .

```
{
  "type": "Feature",
  "properties": {},
  "geometry": {
    "type": "Point",
    "coordinates": [
```

```

    -121.16945743560791,
    38.6632293081453
  ],
  "marker": {
    "icon": {
      "color": "blue",
      "size": {
        "width": 100,
        "height": 100
      },
    },
    "path": "material/location_on"
  },
  "tooltip": {
    "content": {
      "text": "Some text"
    }
  }
}

```



Tooltip Object

Tools tips can shown while hovering over a point. The text of the tooltip is driven by `marker.tooltip.content.text`.

```

[
  {
    "type": "Feature",
    "properties": {},
    "geometry": {
      "type": "Point",
      "coordinates": [
        "-121.16945743560791",
        "38.6632293081453"
      ]
    }
  },
  "marker": {
    "tooltip": {
      "content": {
        "text": "This is my tooltip"
      }
    }
  }
}

```

```
}  
]
```

In addition, extra options can be added. See leaflet's documentation for a list of [tooltip options](#). The example below changes the opacity on the tooltip to .5.

```
{  
  "type": "Feature",  
  "properties": {},  
  "geometry": {  
    "type": "Point",  
    "coordinates": [  
      "-121.16945743560791",  
      "38.6632293081453"  
    ]  
  },  
  "marker": {  
    "tooltip": {  
      "content": {  
        "text": "This is my tooltip"  
      },  
      "options": {  
        "opacity": ".5"  
      }  
    }  
  }  
}
```

Popup Object

Simplistic popups can be added to points by adding a **marker.popup.content.text** property. Clicking such a marker will cause a popup to appear.

```
{  
  "type": "Feature",  
  "properties": {},  
  "geometry": {  
    "type": "Point",  
    "coordinates": [  
      "-121.16945743560791",  
      "38.6632293081453"  
    ]  
  },  
  "marker": {  
    "popup": {  
      "content": {  
        "text": "Hello from my popup"  
      }  
    }  
  }  
}
```



Like tooltips popups have additional options that can be specified in an **options** object. See leaflet's [popup documentation](#) for a list of possible options.

```
{
  "type": "Feature",
  "properties": {},
  "geometry": {
    "type": "Point",
    "coordinates": [
      "-121.16945743560791",
      "38.6632293081453"
    ]
  },
  "marker": {
    "popup": {
      "content": {
        "text": "Hello from my popup"
      },
      "options": {
        "minWidth": 400
      }
    }
  }
}
```

All Together

Each of these marker objects can be applied simultaneously, each adding additional functionality.



```
geometry (2)  
  type : Point  
  ▶ coordinates [2]  
styleOptions {1}  
  color : Red  
marker {3}  
  icon {3}  
    color : blue  
    ▶ size {2}  
    path : material/location_on  
  options {1}  
    opacity : .8  
  tooltip {1}  
    ▶ content {1}
```

hideViewMarkersOnZoom : true

Perspective Map - Custom Controls Example

In this example, we'll set up three custom controls on a map component. The controls will be icons in the top right corner. When a user clicks on the home icon, the map will return to its default settings. When they click on the waves icon, the map navigates to the big island of Hawaii. When the snowflake icon is clicked, the map navigates to the Sierra Mountain range near South Lake Tahoe California.

1. Create a folder under Views called Components.
2. Next we'll create the view that will have the custom controls. In the Components folder, create a flex view called **MapLocations**. The MapLocations view will be embedded directly into the map component, and will provide some icons that users can click on to modify the map.
3. Set the Direction to "Row", and Justify to "Space Evenly".
4. You'll want to resize MapLocations so that it's close to the size you'd like to see it. Using the guides in the Designer, we'll set the width to about 200px and the height to about 50px.
5. To help the controls stand out from the map, let's give them a background. Select the root container, open the Style Editor, and apply the following changes:

backgroundColor: #D5D5D5

BoarderRadius: Set to 10px on each corner

< Applied Styles

Background

background-color : #D5D5D5

Border

border-top-left-radius : 10px

border-top-right-radius : 10px

border-bottom-left-radius : 10px

border-bottom-right-radius : 10...

Misc.

overflow : auto

Text

Background

Margin and Padding

Border

Border

Border style **Border width** **Border color**

Border radius

10px All corners 10px

10px 10px

Outline

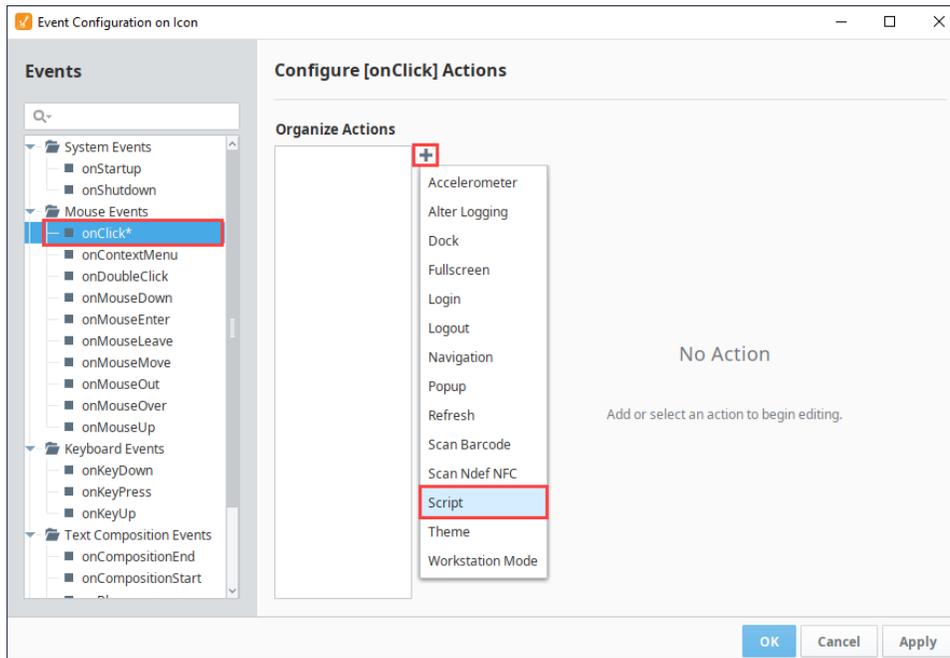
Outline style **Outline width** **Outline color**

Shape

Misc.

OK Cancel

6. Drag an icon component onto the view.
 - a. Set the following properties for the icon:
 - color:** #3CACF9
 - path:** material/home
 - style:** Open the style editor. Under **Misc**, set **Cursor** to "Pointer"
 - b. Right-click on the icon and select **Configure Events**.
 - c. Select the Mouse Events, onClick event.
 - d. Click the **Add +** icon and select the Script action:



- e. In the Configure Script Action window, scroll down to the second line and enter the following:

```
system.perspective.sendMessage("NavigateMap", payload = { "lat": 38.660867, "lng":
-121.159728, "zoom": 13 }, scope = "session")
```

The script above will attempt to invoke a message handler named "NavigateMap", and pass some coordinates to it. At this point, we have not configured NavigateMap yet. We will do so in a later step.

- f. Click **OK** to save the script.
- g. Next, duplicate the icon. We'll configure this icon to represent a different location. Select the first icon, copy it, and paste it. Make the following property change to it:
path : material/ waves
- h. Right-click on the second icon and select **Configure Events**.
- i. Select the Mouse Events, onClick event.
- j. There should already be an action defined on this event, since our copy also has the script we wrote earlier. Select the script action.
- k. In the Configure Script Action window, scroll down to the second line and replace the "lat" and "lng" values in the payload with the following:
lat: 19.6089562306
lng: 155.385131836

The script should look like the following:

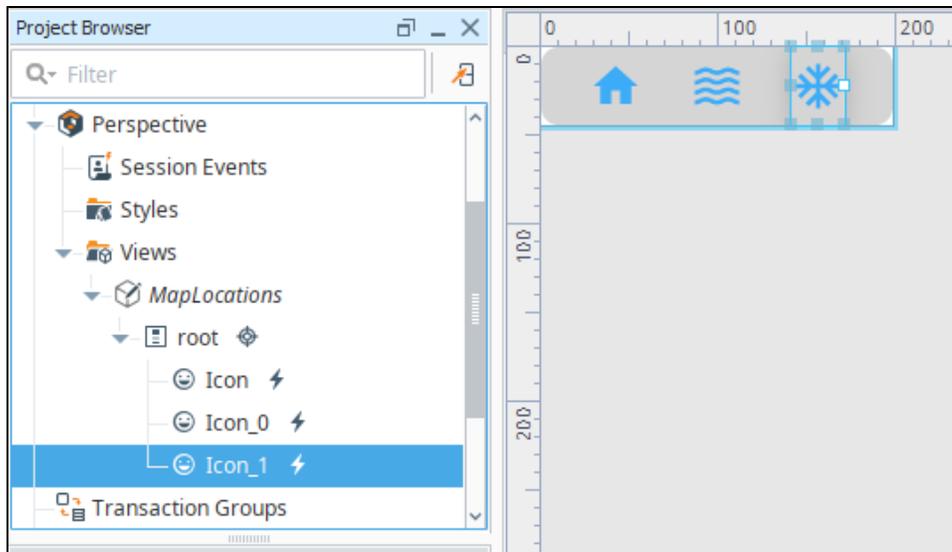
```
system.perspective.sendMessage("NavigateMap", payload = { "lat": 19.6089562306, "lng":
-155.385131836, "zoom": 13 }, scope = "session")
```

- l. Click **OK** to save the script.
- m. Next, duplicate either of the icons again, and change the following property:
path : material/ac_unit
- n. Right-click on the third icon and select **Configure Events**.
- o. Select the Mouse Events, onClick event.
- p. Select the Script action.
- q. In the Configure Script Action window, scroll down to the second line and replace the "lat" and "lng" values in the payload with the following:
lat: 38.8665777761
lng: -119.848823547

The script should look like the following:

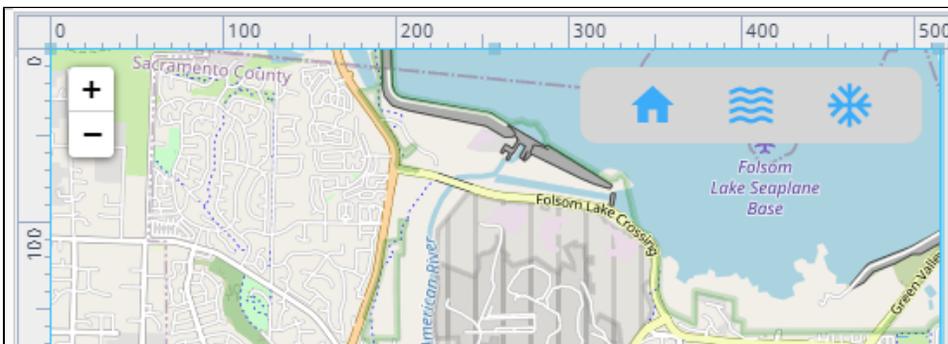
```
system.perspective.sendMessage("NavigateMap", payload = { "lat": 38.8665777761, "lng":
-119.848823547 }, scope = "session")
```

r. Click **OK** to save the script.



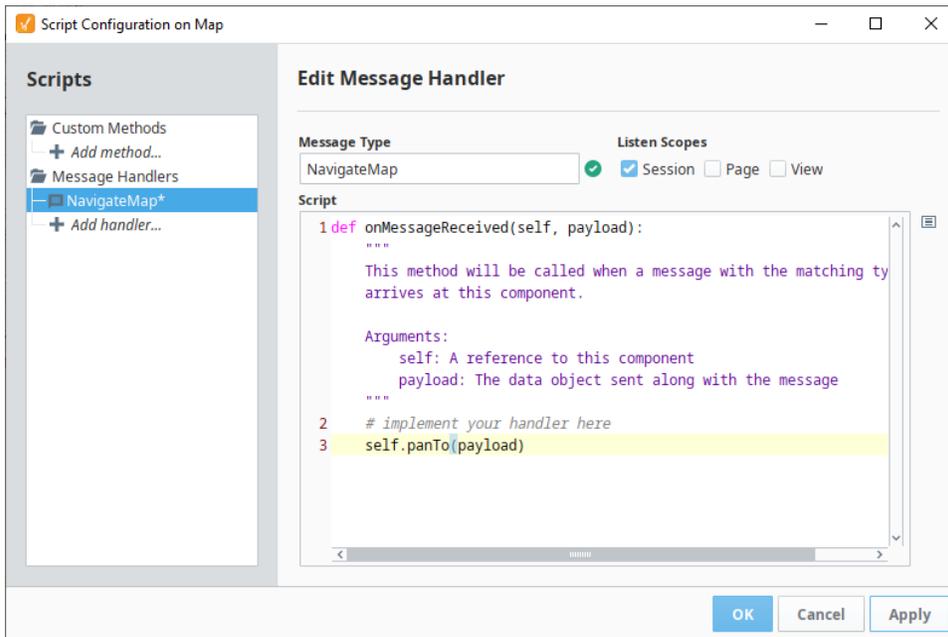
7. Next we'll set up the view with the Map component. In the Components folder, create a new Flex view and name it **Map**.
8. Drag a Map component onto the view.
9. In the Perspective Property editor, set **position.grow** on the map to 1, so the map takes up all available space.
10. Also in the Perspective Property editor, expand the **props.customControls property**.
11. Click the **Add +** icon to add an array element.
12. For the **props.customControls.0** properties, set the following:
enabled: true
path: Components/MapLocations
position: top-right

You'll now see the icons from the MapLocations view appear in the upper right corner of the Map component.



13. Next, we need to configure a message handler on the Map, so when our icons the map will respond. With the Map component selected, right-click and select **Configure Scripts**.
14. Double click on **Add handler** to add a new message handler
15. Set the **Message Type** to the handler name we used in the icon scripts: **NavigateMap**
16. Set **Listen Scopes** to only "Session"
17. Type the following script:

```
self.panTo(payload)
```



18. Click **OK** to commit the script.
19. Save your project.
20. Put your project into Preview mode. Now test it out. Click each icon and see how the location on the Map changes.

Perspective - Map Scripting

This page details the various scripting, component, and extension functions available for [Perspective's Map](#) component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

onMarkerClick

Interaction event. Fired when the marker is clicked. Returns the unique name of the marker.

Object Path	Type	Description
event.name	String	The name of the marker.

onMapClick

Interaction event. Fired when the map is clicked. Returns the lat and lng of the mouse click as it translates on the map.

Object Path	Type	Description
event.lat	Numeric	The latitude of where on the map the user clicked.
event.lng	Numeric	The longitude of where on the map the user clicked.

onMapMouseMove

Interaction event. Fires as the mouse moves over the map. Returns lat and lng of mouse as it translates on the map.

Object Path	Type	Description
event.lat	Numeric	The latitude of where the mouse moved.
event.lng	Numeric	The longitude of where the mouse moved.

onVectorClick

Interaction event. Fired when a vector is clicked. Returns the name of the vector, the type of the vector, and the coordinates of the vector.

Object Path	Type	Description
event.name	Any	The given name of the vector. Returns an empty string if a name is not defined.
event.type	String	The type of the vector clicked. Possible values are: <ul style="list-style-type: none">• polygon• circle• polyline• rectangle
event.lat	Numeric	The latitude of the clicked vector.
event.lng	Numeric	The longitude of the clicked vector.

On this page ...

- [Component Events](#)
 - [onMarkerClick](#)
 - [onMapClick](#)
 - [onMapMouseMove](#)
 - [onVectorClick](#)
 - [onZoomStart](#)
 - [onZoom](#)
 - [onZoomEnd](#)
 - [onMoveStart](#)
 - [onMove](#)
 - [onMoveEnd](#)
 - [onResize](#)
- [Component Functions](#)
 - [.getCenter\(\)](#)
 - [.getZoom\(\)](#)
 - [.getBounds\(\)](#)
 - [.getBoundsAsBBoxString\(\)](#)
 - [.zoomIn\(\[delta, options\]\)](#)
 - [.zoomOut\(\[delta, options\]\)](#)
 - [.setZoomAround\(point, zoom, \[options\]\)](#)
 - [.fitBounds\(latLngBounds, \[options\]\)](#)
 - [.fitWorld\(\[options\]\)](#)
 - [.panTo\(latLng, \[options\]\)](#)
 - [.panBy\(point, \[options\]\)](#)
 - [.flyTo\(latLng, \[zoom, options\]\)](#)
 - [.flyToBounds\(latLngBounds, \[options\]\)](#)
 - [.panInsideBounds\(latLngBounds, \[options\]\)](#)
 - [.panInside\(latLng, \[options\]\)](#)
 - [.getSize\(\)](#)
- [Extension Functions](#)

onZoomStart

Map state event. Fired when the map zoom is about to change (e.g. before zoom animation). Returns the zoom level.

Object Path	Type	Description
event.zoom	Numeric	What the zoom level on the map was <i>before</i> zooming.

onZoom

Map state event. Fired repeatedly during any change in zoom level, included zoom and fly animations. Returns the zoom level.

Object Path	Type	Description
event.zoom	Numeric	What the zoom level was changed to.

onZoomEnd

Map state event. Fired when the map has changed, after any animations. Returns the zoom level.

Object Path	Type	Description
event.zoom	Numeric	What the zoom level on the map was after zooming.

onMoveStart

Map state event. Fired when the view of the map starts changing (e.g., user starts dragging the map). Returns the map center as lat and lng.

Object Path	Type	Description
event.lat	Numeric	The latitude of the center of the map before moving.
event.lng	Numeric	The longitude value for the center of the map before moving.

onMove

Map state event. Fires repeatedly during any movement on the map, include pan and fly animations. Returns the map center as lat and lng.

Object Path	Type	Description
event.lat	Numeric	The latitude value for the center of the map during moving.
event.lng	Numeric	The longitude value for the center of the map during moving.

onMoveEnd

Map state event. Fired when the center of the map stops changing (e.g. user stopped dragging the map). Returns the new map center as lat and lng.

Object Path	Type	Description
event.lat	Numeric	The latitude value for the center of the map after moving.
event.lng	Numeric	The longitude value for the center of the map after moving.

onResize

Map state event. Fires when the map size has changed. Returns the map size as oldSize and newSize.

Object Path	Type	Description
event.oldSize	PyObjectAdapter	Returns the starting size of the map before the resize. Values are returned in an object that's functionally similar to a Python dictionary. The object contains two values: height and width.
event.	PyObjectAdapter	Returns the starting size of the map after the resize. Values are returned in an object that's functionally similar to a

Component Functions

.getCenter()

- Description

Returns the geographical center of the map in latitude and longitude.

- Parameters

Nothing

- Return

[LatLng](#) Returns the geographical center of the map view as { lat: number, lng: number } .

.getZoom()

- Description

Returns the current zoom level of the map view as a number.

- Parameters

Nothing

- Return

[Numeric](#) Returns the current zoom level of the map view as a number.

.getBounds()

- Description

Returns the geographical bound of the map as a dictionary.

- Parameters

Nothing

- Return

[Dictionary](#) A dictionary containing the following keys:

```
north: number
northEast: LatLng
LatLngeast: number
southEast: LatLng
south: number
southWest: LatLng
west: number
northWest: LatLng
```

.getBoundsAsBBoxString()

- Description

Returns a string with bounding box coordinates in a '*South West longitude, South West latitude , North East longitude, North East latitude*' format.

- Parameters

Nothing

- Return

[String](#) Returns the bounding box of the map as a string.

.zoomIn([delta, options])

- Description

Increases the zoom of the map by delta.

- Parameters

[Numeric](#) delta - The numerical value to increase the zoom by. If omitted, uses the value of props.zoom.delta. [optional]

[Dictionary](#) options - A dictionary of parameters to use during the zoom, typically containing a single key, animate. See [Perspective - Map#ZoomOptions](#). [optional]

- Return

Nothing

.zoomOut([delta, options])

- Description

Decreases the zoom of the map by delta.

- Parameters

[Numeric](#) delta - The numerical value to increase the zoom by. If omitted, uses the value of props.zoom.delta. [optional]

[Dictionary](#) options - A dictionary of parameters to use during the zoom, typically containing a single key, animate. See [Perspective - Map#ZoomOptions](#). [optional]

- Return

Nothing

.setZoomAround(point, zoom, [options])

- Description

Zooms the map while keeping a specified geographical point on the map stationary (e.g. used internally for scroll zoom and double-click zoom)

- Parameters

[Dictionary](#) point - The geographic point that the map will zoom around. See [Perspective - Map#LatLng](#). [required]

[Numeric](#) zoom - The numerical value to increase the zoom by. If omitted, uses the value of props.zoom.delta. [optional]

[Dictionary](#) options - A dictionary of parameters to use during the zoom, typically containing a single key, animate. See [Perspective - Map#ZoomOptions](#). [optional]

- Return

Nothing

.fitBounds(latLngBounds, [options])

- Description

Sets a map view that contains the given geographical bounds with the maximum zoom level possible

- Parameters

[Dictionary](#) latLngBounds - A dictionary consisting of two LatLng objects. The LatLng objects combined represent the geographical bounds the map view should be set to. [required]

[Dictionary](#) options - A dictionary of parameters used to manipulate the FitBound settings. See [Perspective - Map#FitBounds](#). [optional]

- Return

Nothing

.fitWorld([options])

- Description

Sets a map view that mostly contains the whole world with the maximum zoom level possible.

- Parameters

[Dictionary](#) options - A dictionary of parameters used to manipulate the FitBound settings. See [Perspective - Map#FitBounds](#). [optional]

- Return
Nothing

.panTo(latLng, [options])

- Description
Pans the map to a given center.
- Parameters
[Dictionary](#) latLng - The geographic point to pan to. [required]
[Dictionary](#) options - A dictionary of parameters used to modify the panning behavior. See [Perspective - Map#PanOptions](#). [optional]
- Return
Nothing

.panBy(point, [options])

- Description
Pans the map by a given number of pixels (animated).
- Parameters
[Dictionary](#) point - The geographic point to pan to. The dictionary should contain an 'x' and 'y' key, both with numeric values. [required]
[Dictionary](#) options - A dictionary of parameters used to modify the panning behavior. See [Perspective - Map#PanOptions](#). [optional]
- Return
Nothing

.flyTo(latLng, [zoom, options])

- Description
Sets the view of the map (geographical center and zoom) performing a smooth pan-zoom animation.
- Parameters
[Dictionary](#) latLng - A Python Dictionary representing the coordinates to fly to. [required]
[Numeric](#) zoom - Sets the zoom level to transition to during the flight. If omitted, uses the value on props.zoom.delta. [optional]
[Dictionary](#) options - A dictionary of panning options to use. See [Perspective - Map#PanOptions](#). [optional]
- Return
Nothing

.flyToBounds(latLngBounds, [options])

- Description
Sets the view of the map with a smooth animation like flyTo, but takes a bounds parameter like fitBounds.
- Parameters
[Dictionary](#) latLngBounds - A dictionary consisting of two LatLng objects. The LatLng objects combined represent the geographical bounds the map view should be set to. [required]
[Dictionary](#) options - A dictionary of panning options to use. See [Perspective - Map#PanOptions](#). [optional]
- Return
Nothing

.panInsideBounds(latLngBounds, [options])

- Description
Pans the map to the closest view that would lie inside the given bounds (if it's not already), controlling the animation using the options specific, if any.

- Parameters

[Dictionary](#) latLngBounds - A dictionary consisting of two LatLng objects. The LatLng objects combined represent the geographical bounds the map view should be set to. [required]

[Dictionary](#) options - A dictionary of panning options to use. See [Perspective - Map#PanOptions](#) . [optional]

- Return

Nothing

.panInside(latLng, [options])

- Description

Pans the map the minimum amount to make the latLng visible.

- Parameters

[Dictionary](#) latLng - A Python dictionary representing the coordinates to pan to. [required]

[Dictionary](#) options - A dictionary of panning options to use. See [Perspective - Map#PanOptions](#) . [optional]

- Return

Nothing

.getSize()

- Description

Returns height and width of the Map component.

- Parameters

Nothing

- Return

[JSON Object](#) Returns a Python dictionary. Contains two items: height and width .

Scripting Example

```
def doMapStuff(self):
    map = self.getSibling("Map")
    coordinateBounds = {'corner1': {'lat': 39.086798, 'lng': -120.069014}, 'corner2': { 'lat':
38.815319, 'lng': -119.787519 }}
    latLngTahoe = {'lat': 39.086798, 'lng': -120.069014 }
    latLngInductive = {'lat': 38.652511, 'lng': -121.189438 }
    zoomPanOptions = { 'animate': True, 'duration': 3, 'easeLinearity': 0.25, 'noMoveStart': False }
    fitBoundsOptions = { 'padding': { 'x': 100, 'y': 100 }, 'animate': True, 'duration': 3 }
    panPixels = { 'x': 200, 'y': 200 }

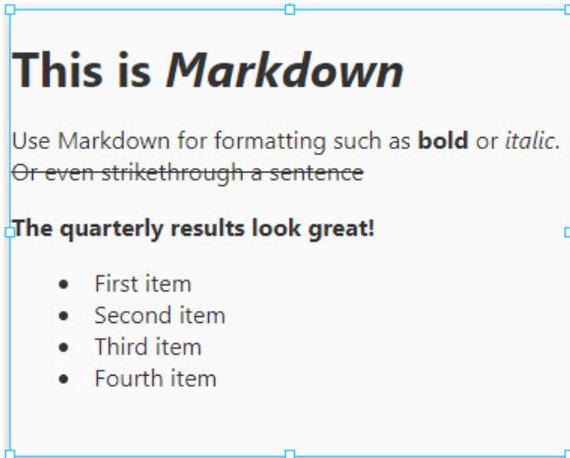
    print map.getCenter()
    # Returns the geographical bounds visible in the current map view as latLngBounds

    map.flyTo(latLng = latLngTahoe, options = zoomPanOptions)
    # Sets the view of the map with a smooth animation like flyTo, but takes a bounds parameter like
fitBounds
```

Extension Functions

This component does not have extension functions associated with it.

Perspective - Markdown



Component Palette Icon:



The Markdown component allows users to format any type of text so it is publishable as plain text without looking like it's been marked up with tags or formatting instructions. The Markdown component provides is a lightweight formatting language which is easy to write and easy to read. Markdown's formatting syntax only addresses issues that can be conveyed in plain text. For any marked up content that is not covered by Markdown's syntax, you can use HTML. You can even change the color of text in the component with HTML tags using the 'source' prop in the Property Editor as shown in Example 2.

To learn more about how to use Markdown component for publishing plain text, refer to the following articles: [Markdown Basics](#) and [Markdown Node Types](#).

If you would like to manually add line breaks, the most direct approach would be to disable the escapeHTML property, and add your own paragraph and line break elements. Once disabled, you can add line breaks as seen below:

```
Value of "source"
<p>
one <br>
two <br>
three <br>
four
five
</p>
```

Because there isn't a line break between "four" and "five", they're on the same line.

On this page ...

- [Properties](#)
- [Component Events](#)
- [Examples](#)
 - [Example 1](#)
 - [Example 2](#)
 - [Example 3](#)

renders

Renders in a browser as plain text.

object

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Examples

Example 1

Markdown [basic syntax](#) link.
From <https://www.markdownguide.org>.
Text formatting examples:

Heading 1

Heading 2

Heading 3

Bold text

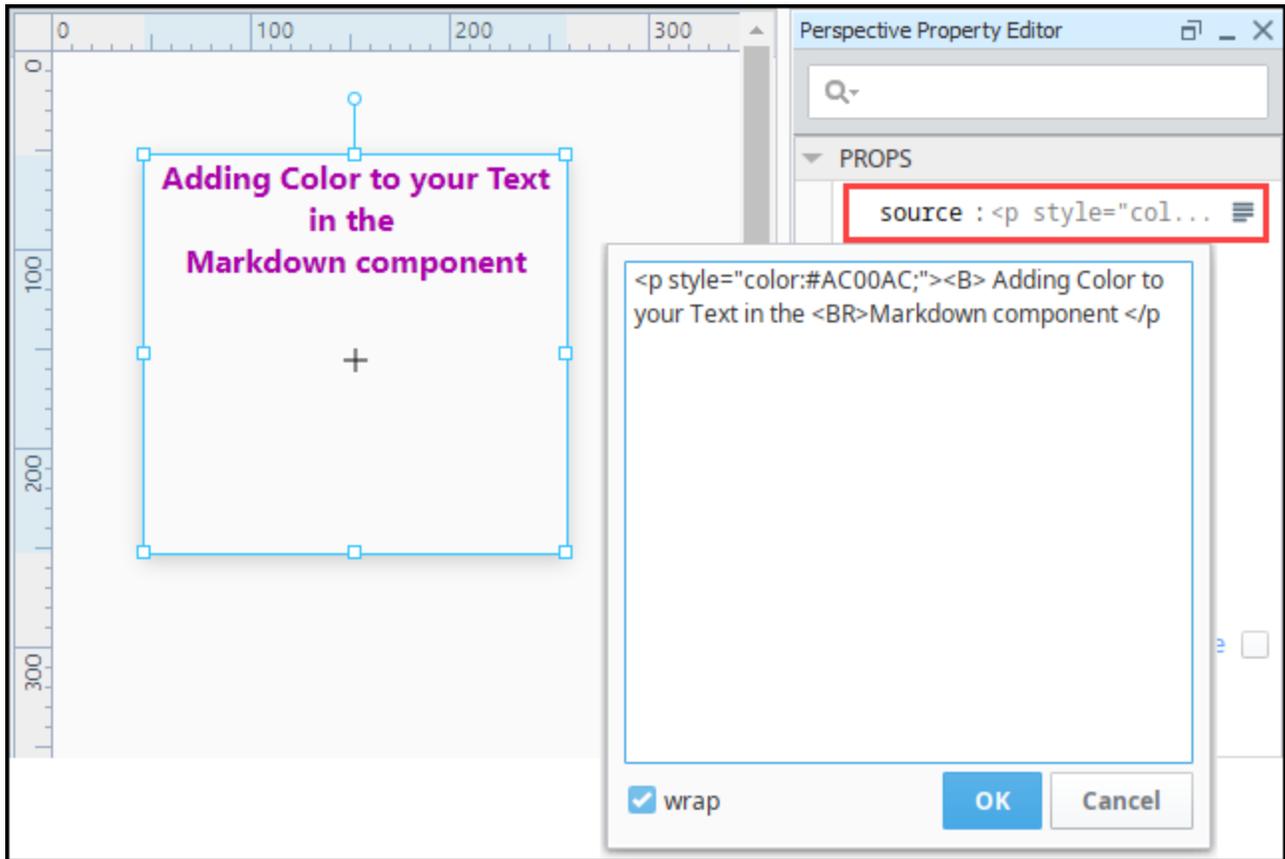
Italic text

Bold and Italic text

Line of plain text

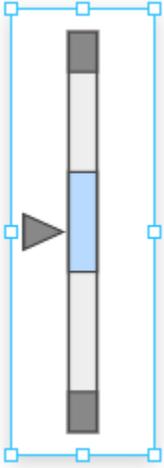
Many other formatting options are listed online.

Property	Value
props.source	MARKDOWN source: <pre>Markdown [basic syntax](https://www.markdownguide.org/basic-syntax/) link. From https://www.markdownguide.org. Text formatting examples: # Heading 1 ## Heading 2</pre>



**Adding Color to your Text
in the
Markdown component**

Perspective - Moving Analog Indicator



On this page ...

- [Properties](#)
- [Component Events](#)
- [Examples](#)
 - [Example 1](#)
 - [Example 2](#)

Component Palette Icon:

▶ **Moving Analog Indicator**

The Moving Analog Indicator displays an analog value in context with other information about that value so that you can visually and quickly see if the value is in the normal range or not. The current value is shown as an arrow pointing at a bar with segments showing the desired operating range, low and high alarm ranges, and interlock ranges.

The Moving Analog Indicator component allows for extremely fast information delivery. At a glance, it is obvious to an operator whether or not the value is where it should be, or if it needs attention. If the value is in one of its alarm ranges, then that range can be set to change color to get attention.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
processValue	Current value of the process.	value: numeric
setpointValue	Current value of the setpoint.	value: numeric
minValue	The minimum value shown on the indicator. Default is 0.	value: numeric
maxValue	The maximum value shown on the indicator. Default is 100.	value: numeric
desiredHigh	The upper limit of the desired range. Default is 65.	value: numeric
desiredLow	The lower limit of the desired range. Default is 40.	value: numeric
highAlarm	Value above indicating a high alarm. Default is 90.	value: numeric
highHighAlarm	Value above indicating a high-high alarm. Default is null.	value: numeric
highInterlock	Value above when an interlock will be activated. Default is null.	value: numeric

lowAlarm	Value below indicating a low alarm. Default is 10.	value: numeric												
lowLowAlarm	Value below indicating a low-low alarm. Default is null.	integer												
lowInterlock	Value below when an interlock will be activated. Default is null.	integer												
desiredRangeColor	Color for the area in the desired range. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color												
defaultRangeColor	Color for the area not defined as a range. See Color Selector .	color												
inactiveAlarmColor	Color for the inactive alarm range. See Color Selector .	color												
level2AlarmColor	Color for the active level 2 alarm (high or low). See Color Selector .	color												
level1AlarmColor	Color for the active level 1 alarm (high-high or low-low). See Color Selector .	color												
interlockColor	Color for the interlock range. See Color Selector .	color												
indicatorColor	Color for the process indicator value. See Color Selector .	color												
setpointColor	Color for setpoint value marker. See Color Selector .	color												
label	Numeric value displayed as text next to the indicator.	object												
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>visible</td> <td>Whether to display the label. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>format</td> <td>Format of numeric value in label, including commas, decimal places, etc. Options as follows: <ul style="list-style-type: none"> • #,##0 • #,##0.0 • #,##0.00 • 0 • 0.0 • 0.00 • #,##0% </td> <td>value: string dropdown</td> </tr> <tr> <td>style</td> <td>Sets a style for the label property. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	visible	Whether to display the label. Default is false.	value: boolean	format	Format of numeric value in label, including commas, decimal places, etc. Options as follows: <ul style="list-style-type: none"> • #,##0 • #,##0.0 • #,##0.00 • 0 • 0.0 • 0.00 • #,##0% 	value: string dropdown	style	Sets a style for the label property. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	
Name	Description	Property Type												
visible	Whether to display the label. Default is false.	value: boolean												
format	Format of numeric value in label, including commas, decimal places, etc. Options as follows: <ul style="list-style-type: none"> • #,##0 • #,##0.0 • #,##0.00 • 0 • 0.0 • 0.00 • #,##0% 	value: string dropdown												
style	Sets a style for the label property. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object												
selectOutline	Border settings for the outline surrounding each range area. Options as follows:	object												
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Sets the color for the outline surrounding the range borders. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>color</td> </tr> <tr> <td>width</td> <td>Sets the width of the outline surrounding the range borders in pixels.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Sets the color for the outline surrounding the range borders. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color	width	Sets the width of the outline surrounding the range borders in pixels.	value: numeric				
Name	Description	Property Type												
color	Sets the color for the outline surrounding the range borders. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color												
width	Sets the width of the outline surrounding the range borders in pixels.	value: numeric												
reverseIndicator	Displays the process value indicator on the opposite side of the scale. Default is false.	value: boolean												
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object												

Component Events

Perspective Component Events

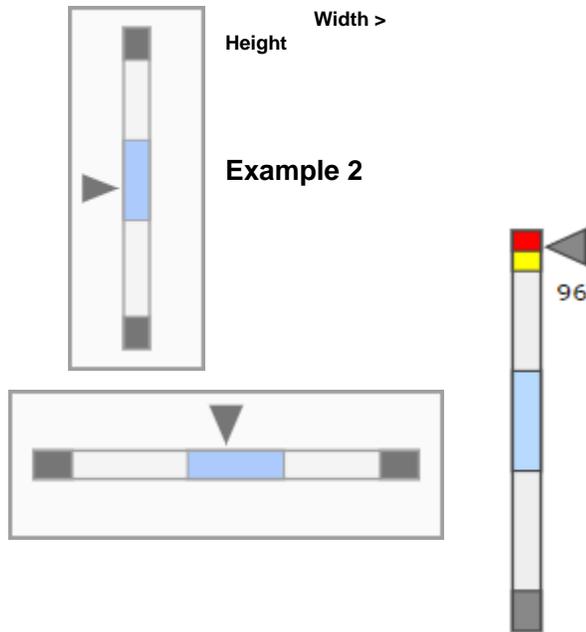
The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Examples

Example 1

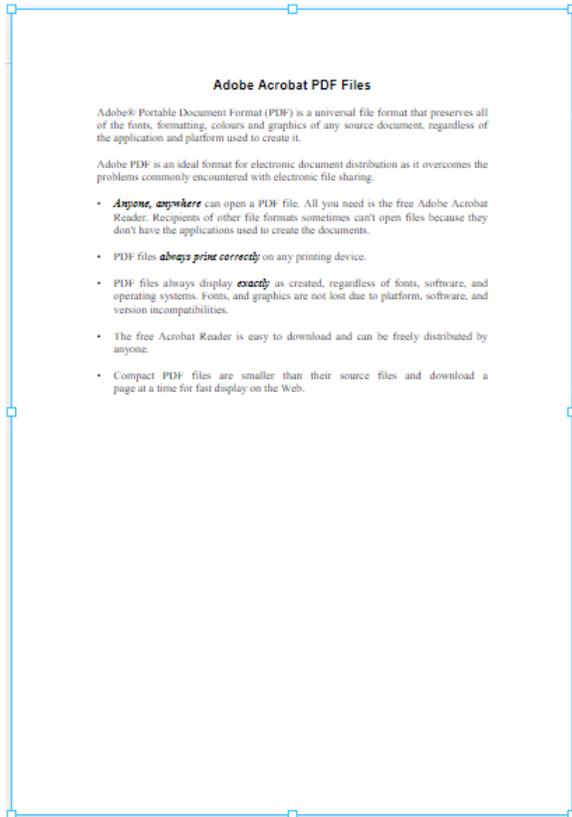
The alignment of the component is based on the height and width of the component.

Height > Width



Property	Value
props.processValue	96
props.highHighAlarm	96
props.reverseIndicator	true
props.label.visible	true

Perspective - PDF Viewer



On this page ...

- [Properties](#)
- [Scripting](#)

Component Palette Icon:



The PDF Viewer component displays a PDF that's hosted on a web server by providing a URL to the source property. A simple approach is to create either a File Resource or Mounted Folder within [Web Dev](#), and set the source on the component to the resource's endpoint.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
source	Path to the .pdf file to be displayed. Expects a URL to a PDF hosted on a web server.	value: string
page	The current page being displayed.	value: numeric
pageCount	The number of pages the pdf contains. Read only.	value: numeric
showPage Number	If true, the current page number and page count will be shown at the bottom of the component.	value: boolean
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object

Scripting

See the [Perspective - PDF Viewer Scripting page](#) for the full list of scripting functions available for this component.

Perspective - PDF Viewer Scripting

This page details the various scripting, component, and extension functions available for [Perspective's PDF Viewer](#) component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

On this page ...

- [Component Events](#)
- [Component Functions](#)
 - [.reload\(\)](#)
- [Extension Functions](#)

Component Functions

.reload()

- Description

This function will reload the PDF in the PDF Viewer component.

- Parameters

[String](#) name - The name of the PDF.

- Return

Nothing

Extension Functions

This component does not have extension functions associated with it.

Perspective - Progress



Component Palette Icon:



On this page ...

- [Properties](#)
- [Component Events](#)
- [Examples](#)
 - [Example 1](#)
 - [Example 2](#)

The Progress bar visually indicates the progress of a task. It is used to display any value that has an upper and lower bound. Custom settings are available for the track and the bar.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type																																			
min	The minimum value of the progress indicator. If the value reaches the max, the progress indicator will be completely filled. Must be less than the max. Default is 0.	value: numeric																																			
max	The maximum value of the progress indicator. If the value reaches the max, the progress indicator will be completely filled. Must be greater than 0.0. Default is 100.	value: numeric																																			
value	The current value representing the current progress. Must be greater than 0.0 and less than the value set in max. Default is 50.	value: numeric																																			
mode	Determines if the component should show a determinate state, or an indeterminate loading state. When set to determinate, shows the progress of the value relative to the min and max properties.	value: string																																			
bar	Settings for the bar.	object																																			
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>A convenience property for setting the base color of the bar. This can also be accomplished by using the prop.bar.style to set the background color.</td> <td>object</td> </tr> <tr> <td>style</td> <td>Sets a style for the bar. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>determinate</td> <td>Determinate bar configuration.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>A convenience property for setting the color of the bar when mode is determinate. This can also be accomplished by using the prop.bar.determinate.style to set the background color.</td> <td>color</td> </tr> <tr> <td>style</td> <td>Sets a style for the bar when mode is determinate. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>indeterminate</td> <td>Indeterminate bar configuration.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>A convenience property for setting the color of the bar when mode</td> <td>color</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	color	A convenience property for setting the base color of the bar. This can also be accomplished by using the prop.bar.style to set the background color.	object	style	Sets a style for the bar. Full menu of style options is available. You can also specify a style class .	object	determinate	Determinate bar configuration.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>A convenience property for setting the color of the bar when mode is determinate. This can also be accomplished by using the prop.bar.determinate.style to set the background color.</td> <td>color</td> </tr> <tr> <td>style</td> <td>Sets a style for the bar when mode is determinate. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	color	A convenience property for setting the color of the bar when mode is determinate. This can also be accomplished by using the prop.bar.determinate.style to set the background color.	color	style	Sets a style for the bar when mode is determinate. Full menu of style options is available. You can also specify a style class .	object		indeterminate	Indeterminate bar configuration.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>A convenience property for setting the color of the bar when mode</td> <td>color</td> </tr> </tbody> </table>	Name	Description	Property Type	color	A convenience property for setting the color of the bar when mode	color	
Name	Description	Property Type																																			
color	A convenience property for setting the base color of the bar. This can also be accomplished by using the prop.bar.style to set the background color.	object																																			
style	Sets a style for the bar. Full menu of style options is available. You can also specify a style class .	object																																			
determinate	Determinate bar configuration.	object																																			
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>A convenience property for setting the color of the bar when mode is determinate. This can also be accomplished by using the prop.bar.determinate.style to set the background color.</td> <td>color</td> </tr> <tr> <td>style</td> <td>Sets a style for the bar when mode is determinate. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	color	A convenience property for setting the color of the bar when mode is determinate. This can also be accomplished by using the prop.bar.determinate.style to set the background color.	color	style	Sets a style for the bar when mode is determinate. Full menu of style options is available. You can also specify a style class .	object																											
Name	Description	Property Type																																			
color	A convenience property for setting the color of the bar when mode is determinate. This can also be accomplished by using the prop.bar.determinate.style to set the background color.	color																																			
style	Sets a style for the bar when mode is determinate. Full menu of style options is available. You can also specify a style class .	object																																			
indeterminate	Indeterminate bar configuration.	object																																			
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>A convenience property for setting the color of the bar when mode</td> <td>color</td> </tr> </tbody> </table>	Name	Description	Property Type	color	A convenience property for setting the color of the bar when mode	color																														
Name	Description	Property Type																																			
color	A convenience property for setting the color of the bar when mode	color																																			

	<table border="1"> <tr> <td></td> <td>is indeterminate. This can also be accomplished by using the prop. <code>bar.indeterminate.style</code> to set the background color.</td> <td></td> </tr> <tr> <td>style</td> <td>Sets a style for the bar when mode is determinate. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </table>		is indeterminate. This can also be accomplished by using the prop. <code>bar.indeterminate.style</code> to set the background color.		style	Sets a style for the bar when mode is determinate. Full menu of style options is available. You can also specify a style class .	object																																		
	is indeterminate. This can also be accomplished by using the prop. <code>bar.indeterminate.style</code> to set the background color.																																								
style	Sets a style for the bar when mode is determinate. Full menu of style options is available. You can also specify a style class .	object																																							
track	<p>Settings for the track.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>A convenience property for setting the base color of the track. This can also be accomplished by using the prop. <code>track.style</code> to set the background color.</td> <td>object</td> </tr> <tr> <td>style</td> <td>Sets a style for the track. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>determinate</td> <td>Determinate track configuration.</td> <td></td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Table</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>A convenience property for setting the color of the track when mode is determinate. This can also be accomplished by using the prop. <code>track.determinate.style</code> to set the background color.</td> <td></td> </tr> <tr> <td>style</td> <td>Sets a style for the track when mode is determinate. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>indeterminate</td> <td>Indeterminate track configuration.</td> <td></td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>A convenience property for setting the color of the track when mode is indeterminate. This can also be accomplished by using the prop. <code>track.indeterminate.style</code> to set the background color.</td> <td></td> </tr> <tr> <td>style</td> <td>Sets a style for the track when mode is indeterminate. Full menu of style options is available.</td> <td>object</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	color	A convenience property for setting the base color of the track. This can also be accomplished by using the prop. <code>track.style</code> to set the background color.	object	style	Sets a style for the track. Full menu of style options is available. You can also specify a style class .	object	determinate	Determinate track configuration.			<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Table</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>A convenience property for setting the color of the track when mode is determinate. This can also be accomplished by using the prop. <code>track.determinate.style</code> to set the background color.</td> <td></td> </tr> <tr> <td>style</td> <td>Sets a style for the track when mode is determinate. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Table	color	A convenience property for setting the color of the track when mode is determinate. This can also be accomplished by using the prop. <code>track.determinate.style</code> to set the background color.		style	Sets a style for the track when mode is determinate. Full menu of style options is available. You can also specify a style class .	object		indeterminate	Indeterminate track configuration.			<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>A convenience property for setting the color of the track when mode is indeterminate. This can also be accomplished by using the prop. <code>track.indeterminate.style</code> to set the background color.</td> <td></td> </tr> <tr> <td>style</td> <td>Sets a style for the track when mode is indeterminate. Full menu of style options is available.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	color	A convenience property for setting the color of the track when mode is indeterminate. This can also be accomplished by using the prop. <code>track.indeterminate.style</code> to set the background color.		style	Sets a style for the track when mode is indeterminate. Full menu of style options is available.	object		object
Name	Description	Property Type																																							
color	A convenience property for setting the base color of the track. This can also be accomplished by using the prop. <code>track.style</code> to set the background color.	object																																							
style	Sets a style for the track. Full menu of style options is available. You can also specify a style class .	object																																							
determinate	Determinate track configuration.																																								
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Table</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>A convenience property for setting the color of the track when mode is determinate. This can also be accomplished by using the prop. <code>track.determinate.style</code> to set the background color.</td> <td></td> </tr> <tr> <td>style</td> <td>Sets a style for the track when mode is determinate. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Table	color	A convenience property for setting the color of the track when mode is determinate. This can also be accomplished by using the prop. <code>track.determinate.style</code> to set the background color.		style	Sets a style for the track when mode is determinate. Full menu of style options is available. You can also specify a style class .	object																															
Name	Description	Property Table																																							
color	A convenience property for setting the color of the track when mode is determinate. This can also be accomplished by using the prop. <code>track.determinate.style</code> to set the background color.																																								
style	Sets a style for the track when mode is determinate. Full menu of style options is available. You can also specify a style class .	object																																							
indeterminate	Indeterminate track configuration.																																								
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>A convenience property for setting the color of the track when mode is indeterminate. This can also be accomplished by using the prop. <code>track.indeterminate.style</code> to set the background color.</td> <td></td> </tr> <tr> <td>style</td> <td>Sets a style for the track when mode is indeterminate. Full menu of style options is available.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	color	A convenience property for setting the color of the track when mode is indeterminate. This can also be accomplished by using the prop. <code>track.indeterminate.style</code> to set the background color.		style	Sets a style for the track when mode is indeterminate. Full menu of style options is available.	object																															
Name	Description	Property Type																																							
color	A convenience property for setting the color of the track when mode is indeterminate. This can also be accomplished by using the prop. <code>track.indeterminate.style</code> to set the background color.																																								
style	Sets a style for the track when mode is indeterminate. Full menu of style options is available.	object																																							
valueDisplay	<p>Value display configuration. Renders and styles a value overlay above the progress bar.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>If true, will show the value display.</td> <td>value: boolean</td> </tr> <tr> <td>format</td> <td>Format to apply to value, which is then used in the value display.</td> <td>value: string</td> </tr> <tr> <td>justify</td> <td>Horizontal alignment of the displayed value.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Sets a style for the track. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	If true, will show the value display.	value: boolean	format	Format to apply to value, which is then used in the value display.	value: string	justify	Horizontal alignment of the displayed value.	value: string	style	Sets a style for the track. Full menu of style options is available. You can also specify a style class .	object	object																								
Name	Description	Property Type																																							
enabled	If true, will show the value display.	value: boolean																																							
format	Format to apply to value, which is then used in the value display.	value: string																																							
justify	Horizontal alignment of the displayed value.	value: string																																							
style	Sets a style for the track. Full menu of style options is available. You can also specify a style class .	object																																							
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																																							

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

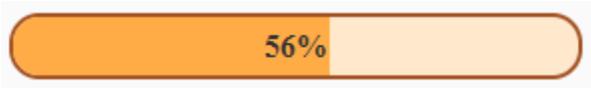
Examples

Example 1



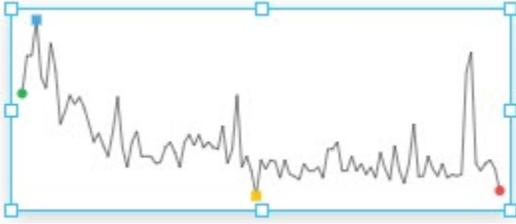
Property	Value
max	100
value	80

Example 2



Property	Value
max	100
value	56
mode	determinate
bar.color	#FFAC47
track.color	#FFE8CC
track.borderStyle	solid
track.borderWidth	2
track.borderBottomLeftRadius	15
track.borderBottomRightRadius	15
track.borderTopRightRadius	15
track.borderTopLeftRadius	15
track.borderColor	#A45324
valueDisplay.enabled	true
valueDisplay.format	percent
valueDisplay.justify	center
valueDisplay.fontFamily	Merriweather
valueDisplay.fontWeight	bold

Perspective - Sparkline



Component Palette Icon:



On this page ...
<ul style="list-style-type: none"> • Properties • Component Events • Example

The Sparkline chart is a minimalistic chart component that displays a line-chart history for a single datapoint. Sparklines were invented by Edward Tufte as a way to show a great deal of contextual information in a very small amount of space. Sparklines are typically used to display the recent history (up to current time) of a datapoint so that the viewer can quickly discern the recent trend of a datapoint: is it rising? falling? oscillating? etc..

To use a sparkline, bind its Data property either to a Tag Historian realtime query, or to a database query. There should be two columns in this dataset: the first one a date column, the second a number. Each row will become a datapoint on the chart, and the dataset must be sorted by time in ascending order.

Instead of using axes to convey scale, the Sparkline can display a band of color across the back of the chart which indicates the desired operating range of the datapoint. In this way, it is instantly obvious when a value is in its expected range, above that range, or below. The sparkline automatically configures its internal axes based on the data given to it. To display a desired range, fill in the **props.desired.high** and **props.desired.low** properties.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type						
points	<p>Data points to plot. Accepts the following: May be a dataset, or an array of values or of objects containing X or Y coordinates. Also may be a string formatted with X and Y values separated by a comma.</p> <ul style="list-style-type: none"> • An array of numbers containing X or Y coordinates. • A string of space delimited points where x and y are separated by a comma. For example: 0,20 1,35 2,15 • An array of objects, where each object contains an x and a y property, and where each property's value is a number. • A dataset of a single column of number type. • A dataset of two columns, the first representing the x value and the second column representing the y value. The first column can be either of type number or type date. Dates and Timestamps are converted to unix timestamps which is used as the x value. 	array or dataset						
color	Color of the line. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color						
width	Thickness of the line, in pixels.	value: numeric						
opacity	The opacity of the line ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric						
dashArray	The pattern of dashes and gaps used to paint the stroke. It's a list of comma separated lengths (in pixels) and percentages (percentage of the total stroke length) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5,3,2" is equivalent to "5,3,2,5,3,2".	array						
range	Settings for the upper and lower edge of the chart.	object						
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>high</td> <td>A fixed value for the upper edge of the chart as a number.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	high	A fixed value for the upper edge of the chart as a number.	value: numeric	
Name	Description	Property Type						
high	A fixed value for the upper edge of the chart as a number.	value: numeric						

low	A fixed value for the lower edge of the chart as a number.	value: numeric
-----	--	----------------

desired The desired operating range. Settings for the desired properties operating range.

Name	Description	Property Type															
high	The high value of the desired operating range.	value: numeric															
low	The low value of the desired operating range.	value: numeric															
stroke	Settings for the stroke. Options as follows:	object															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color of the line. See Color Selector.</td> <td>color</td> </tr> <tr> <td>width</td> <td>Width of the line in pixels.</td> <td>value: numeric</td> </tr> <tr> <td>opacity</td> <td>Opacity of the line ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td>value: numeric</td> </tr> <tr> <td>dashArray</td> <td>The pattern of dashes and gaps used to paint the stroke. It's a list of space separated lengths (in pixels) and percentages (percentage of the stroke length for the desired operating range) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5 3 2" is equivalent to "5 3 2 5 3 2".</td> <td>array</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color of the line. See Color Selector .	color	width	Width of the line in pixels.	value: numeric	opacity	Opacity of the line ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric	dashArray	The pattern of dashes and gaps used to paint the stroke. It's a list of space separated lengths (in pixels) and percentages (percentage of the stroke length for the desired operating range) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5 3 2" is equivalent to "5 3 2 5 3 2".	array	
Name	Description	Property Type															
color	Color of the line. See Color Selector .	color															
width	Width of the line in pixels.	value: numeric															
opacity	Opacity of the line ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric															
dashArray	The pattern of dashes and gaps used to paint the stroke. It's a list of space separated lengths (in pixels) and percentages (percentage of the stroke length for the desired operating range) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5 3 2" is equivalent to "5 3 2 5 3 2".	array															
fill	Settings for the fill for the desired range. Options as follows:	object															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Fill color of the desired range. See Color Selector.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the fill ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Fill color of the desired range. See Color Selector .	color	opacity	Opacity of the fill ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric							
Name	Description	Property Type															
color	Fill color of the desired range. See Color Selector .	color															
opacity	Opacity of the fill ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric															

marker Settings for the first, last, high, and low markers on the chart.

object

Name	Description	Property Type																											
first	Settings for the first marker on the chart.	object																											
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>shape</td> <td>Shape of the marker used to indicate the first point. Options are circle, triangle, or square. Default is circle.</td> <td>value: string dropdown</td> </tr> <tr> <td>size</td> <td>Size of the marker, in pixels.</td> <td>value: numeric</td> </tr> <tr> <td>stroke</td> <td>Settings for the stroke for the first marker. Options as follows:</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color of the line. See Color Selector.</td> <td>color</td> </tr> <tr> <td>width</td> <td>Width of the line in pixels. .</td> <td>value: numeric</td> </tr> <tr> <td>opacity</td> <td>Opacity of the line ranging from 0 to</td> <td>value:</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	shape	Shape of the marker used to indicate the first point. Options are circle, triangle, or square. Default is circle.	value: string dropdown	size	Size of the marker, in pixels.	value: numeric	stroke	Settings for the stroke for the first marker. Options as follows:	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color of the line. See Color Selector.</td> <td>color</td> </tr> <tr> <td>width</td> <td>Width of the line in pixels. .</td> <td>value: numeric</td> </tr> <tr> <td>opacity</td> <td>Opacity of the line ranging from 0 to</td> <td>value:</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color of the line. See Color Selector .	color	width	Width of the line in pixels. .	value: numeric	opacity	Opacity of the line ranging from 0 to	value:		
Name	Description	Property Type																											
shape	Shape of the marker used to indicate the first point. Options are circle, triangle, or square. Default is circle.	value: string dropdown																											
size	Size of the marker, in pixels.	value: numeric																											
stroke	Settings for the stroke for the first marker. Options as follows:	object																											
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color of the line. See Color Selector.</td> <td>color</td> </tr> <tr> <td>width</td> <td>Width of the line in pixels. .</td> <td>value: numeric</td> </tr> <tr> <td>opacity</td> <td>Opacity of the line ranging from 0 to</td> <td>value:</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color of the line. See Color Selector .	color	width	Width of the line in pixels. .	value: numeric	opacity	Opacity of the line ranging from 0 to	value:																
Name	Description	Property Type																											
color	Color of the line. See Color Selector .	color																											
width	Width of the line in pixels. .	value: numeric																											
opacity	Opacity of the line ranging from 0 to	value:																											

	1. 0 is fully transparent, 1 is fully opaque.	numeric									
dashArray	The pattern of dashes and gaps used to paint the stroke. It's a list of space separated lengths (in pixels) and percentages (percentage of the stroke length for the desired operating range) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5 3 2" is equivalent to "5 3 2 5 3 2".	array									
fill	Settings for the fill for the first marker. Options as follows:	object									
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color of the line. See Color Selector.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Cursor line opacity. 0 is fully transparent, 1 is fully opaque.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color of the line. See Color Selector .	color	opacity	Cursor line opacity. 0 is fully transparent, 1 is fully opaque.	value: numeric	
Name	Description	Property Type									
color	Color of the line. See Color Selector .	color									
opacity	Cursor line opacity. 0 is fully transparent, 1 is fully opaque.	value: numeric									
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object									

last Settings for the last marker on the chart. object

Name	Description	Property Type															
shape	Shape of the last marker. Options are circle, triangle, or square. Default is circle.	value: string dropdown															
size	Size of the last marker, in pixels.	value: numeric															
stroke	Settings for the stroke for the last marker. Options as follows:	object															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color of the line. See Color Selector.</td> <td>color</td> </tr> <tr> <td>width</td> <td>Width of the line in pixels.</td> <td>value: numeric</td> </tr> <tr> <td>opacity</td> <td>Opacity of the line ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td>value: numeric</td> </tr> <tr> <td>dashArray</td> <td>The pattern of dashes and gaps used to paint the stroke. It's a list of space separated lengths (in pixels) and percentages (percentage of the stroke length for the desired operating range) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5 3 2" is equivalent to "5 3 2 5 3 2".</td> <td>array</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color of the line. See Color Selector .	color	width	Width of the line in pixels.	value: numeric	opacity	Opacity of the line ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric	dashArray	The pattern of dashes and gaps used to paint the stroke. It's a list of space separated lengths (in pixels) and percentages (percentage of the stroke length for the desired operating range) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5 3 2" is equivalent to "5 3 2 5 3 2".	array	
Name	Description	Property Type															
color	Color of the line. See Color Selector .	color															
width	Width of the line in pixels.	value: numeric															
opacity	Opacity of the line ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric															
dashArray	The pattern of dashes and gaps used to paint the stroke. It's a list of space separated lengths (in pixels) and percentages (percentage of the stroke length for the desired operating range) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5 3 2" is equivalent to "5 3 2 5 3 2".	array															
fill	Settings for the fill for the last marker. Options as follows:	object															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property</th> </tr> </thead> <tbody> </tbody> </table>	Name	Description	Property													
Name	Description	Property															

		Type
color	Color of the line. See Color Selector .	color
opacity	Opacity of the line ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object

high Settings for the high marker on the chart. object

Name	Description	Property Type															
shape	Shape of the high marker. Options are circle, triangle, or square. Default is square.	value: string dropdown															
size	Size of the high marker, in pixels.	value: numeric															
stroke	Settings for the stroke for the high marker. Options as follows:	object															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color of the line. See Color Selector.</td> <td>color</td> </tr> <tr> <td>width</td> <td>Width of the line in pixels.</td> <td>value: numeric</td> </tr> <tr> <td>opacity</td> <td>Opacity of the line ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td>value: numeric</td> </tr> <tr> <td>dashArray</td> <td>The pattern of dashes and gaps used to paint the stroke. It's a list of comma separated lengths (in pixels) and percentages (percentage of the stroke length for the desired operating range) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5,3,2" is equivalent to "5,3,2,5,3,2".</td> <td>array</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color of the line. See Color Selector .	color	width	Width of the line in pixels.	value: numeric	opacity	Opacity of the line ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric	dashArray	The pattern of dashes and gaps used to paint the stroke. It's a list of comma separated lengths (in pixels) and percentages (percentage of the stroke length for the desired operating range) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5,3,2" is equivalent to "5,3,2,5,3,2".	array	
Name	Description	Property Type															
color	Color of the line. See Color Selector .	color															
width	Width of the line in pixels.	value: numeric															
opacity	Opacity of the line ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric															
dashArray	The pattern of dashes and gaps used to paint the stroke. It's a list of comma separated lengths (in pixels) and percentages (percentage of the stroke length for the desired operating range) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5,3,2" is equivalent to "5,3,2,5,3,2".	array															
fill	Settings for the fill for the high marker. Options as follows:	object															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color of the line. See Color Selector.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the line ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color of the line. See Color Selector .	color	opacity	Opacity of the line ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric							
Name	Description	Property Type															
color	Color of the line. See Color Selector .	color															
opacity	Opacity of the line ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric															
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															

low Settings for the low marker on the chart. object

Name	Description	Property Type

	shape	Shape of the low marker. Options are circle, triangle, or square. Default is square.	value: string dropdown															
	size	Size of the marker, in pixels.	value: numeric															
	stroke	Settings for the stroke for the low marker. Options as follows:	object															
		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color of the line. See Color Selector.</td> <td>color</td> </tr> <tr> <td>width</td> <td>Width of the line in pixels.</td> <td>value: numeric</td> </tr> <tr> <td>opacity</td> <td>Opacity of the line ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td>value: numeric</td> </tr> <tr> <td>dashArray</td> <td>The pattern of dashes and gaps used to paint the stroke. It's a list of space separated lengths (in pixels) and percentages (percentage of the stroke length for the desired operating range) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5 3 2" is equivalent to "5 3 2 5 3 2".</td> <td>array</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color of the line. See Color Selector .	color	width	Width of the line in pixels.	value: numeric	opacity	Opacity of the line ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric	dashArray	The pattern of dashes and gaps used to paint the stroke. It's a list of space separated lengths (in pixels) and percentages (percentage of the stroke length for the desired operating range) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5 3 2" is equivalent to "5 3 2 5 3 2".	array	
Name	Description	Property Type																
color	Color of the line. See Color Selector .	color																
width	Width of the line in pixels.	value: numeric																
opacity	Opacity of the line ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric																
dashArray	The pattern of dashes and gaps used to paint the stroke. It's a list of space separated lengths (in pixels) and percentages (percentage of the stroke length for the desired operating range) that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, "5 3 2" is equivalent to "5 3 2 5 3 2".	array																
	fill	Settings for the fill for the low marker. Options as follows:	object															
		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color of the line. See Color Selector.</td> <td>color</td> </tr> <tr> <td>opacity</td> <td>Opacity of the line ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color of the line. See Color Selector .	color	opacity	Opacity of the line ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric							
Name	Description	Property Type																
color	Color of the line. See Color Selector .	color																
opacity	Opacity of the line ranging from 0 to 1. 0 is fully transparent, 1 is fully opaque.	value: numeric																
	style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															
style		Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															

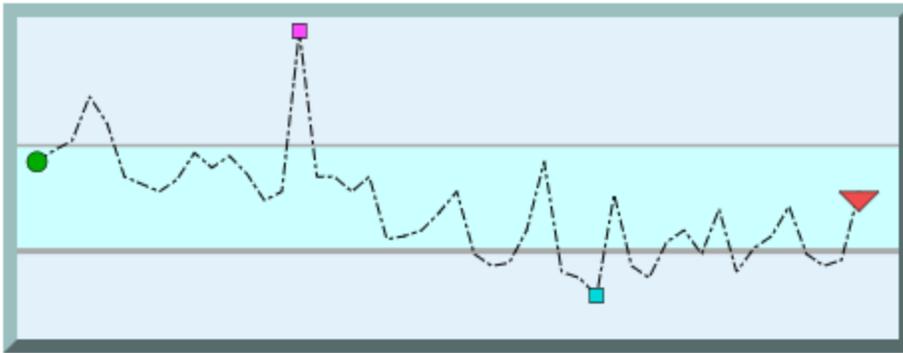
Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Example

This example demonstrates what changes to the various marker properties can do. In addition, the desired range has been enabled.



Property	Value
props.width	1
props.dashArray	6 2 2 2
props.desired.high	60
props.desired.low	25
props.desired.stroke.color	#AAAAAA
props.desired.stroke.width	3
props.desired.stroke.opacity	3
props.desired.fill.color	#CCFFF
props.desired.fill.opacity	4
props.marker.first.shape	circle
props.marker.first.size	10
props.marker.first.stroke.color	#000000
props.marker.first.fill.color	#00AC00
props.marker.last.shape	triangle
props.marker.last.size	20
props.marker.last.stroke.color	#000000
props.marker.last.fill.color	#EF4D4D
props.marker.low.shape	square
props.marker.low.size	7
props.marker.low.stroke.color	#000000
props.marker.low.fill.color	#0D9D9
props.marker.high.shape	square
props.marker.high.size	7
props.marker.high.stroke.color	#000000
props.marker.high.fill.color	#FF47FF
props.style.backgroundColor	#E2F1Fa
props.style.borderStyle	outset
props.style.borderWidth	7px
props.style.borderColor	#9BBFBF

Perspective - Table

city	country	population
Folsom	United States	77,271
Helsinki	Finland	635,591
Jakarta	Indonesia	10,187,595
Madrid	Spain	3,233,527
Prague	Czech Republic	1,241,664
San Diego	United States	1,406,630
San Francisco	United States	884,363
Shanghai	China	24,153,000
Tokyo	Japan	13,617,000
Washington, DC	United States	658,893
Wellington	New Zealand	405,000

On this page ...

- [User Interaction](#)
- [Properties](#)
- [Scripting](#)
- [Examples](#)
 - [Example 1: Styling Rows Based on Value - JSON Data](#)
 - [Example 2: Styling Rows Based on Value - Dataset Data](#)
 - [More Examples](#)

Component Palette Icon:



The Table component displays database data in tabular form. Properties enable you to customize the data content, style, navigation, and user interaction of your table.

User Interaction

Interaction	Description
Column Resizing	When configured through the designer via the corresponding column config, a column can be resized during runtime. The resize handle exists in a 36px swath centered on the end of the header cell. Hovering over this area will change the mouse cursor to column resizing. Dragging the resize handle will display a resize guide effectively providing a visual for the new column position as the user drags. These changes in width will not persist, and are merely for the convenience of the user.
Sorting	<p>When sorting is enabled on a column and the table head is enabled, a sort indicator will display to the right of the header cell content. The sort indicator will display the sort direction.</p> <ul style="list-style-type: none"> • Single Sort - Enabled by double clicking on a header cell. • Multi Sort - Enabled by holding down Shift then double-clicking on multiple header cells. <div style="border: 1px solid orange; padding: 5px; margin: 10px 0;"> <p>The following feature is new in Ignition version 8.1.6 Click here to check out the other new features</p> </div> <p>As of 8.1.6 the Table component now sorts columns based on the underlying data type in the column, instead of sorting alphanumerically as if all values were string formatted.</p>
Selection	<p>When selection is enabled, a user may select table data based upon the table's selection configuration. In the browser, selection is indicated by a light blue overlay rendered on cells. The root selection, or most recently selected cell has a light blue border. The root selection corresponds with the selected column and selected row properties of the table component's selection configuration.</p> <ul style="list-style-type: none"> • Single - Single mouse click enabled. • Single Interval - Shift and single mouse click enabled. • Multiple Interval - Command/Ctrl + shift and single mouse click enabled.
Editing	

	When editing is enabled on an individual cell, a user can edit a cell by performing the interaction specified by the 'allow edit on' property of the table component. When in edit mode, an editing cell with the corresponding cells content will be presented for edit. To commit this edit, the user must press the return or enter key. To exit the edit, the user may either press the escape key or select another table cell. When an edit is committed, the edited data is sent to the cell edit component event of the Table component.
Paging	When paging is enabled, a user may use the provided buttons to navigate between available pages and also jump to a specific page within range. 
Filtering	When filtering is enabled, a user may filter all of the data, not just the data being displayed when pagers are enabled, of the table component. If paging also happens to be enabled, the table will automatically page jump if it becomes necessary so that it does not display an empty page.
Coloring /Look	The table is made up of various subareas (rows, cells, etc). To aid with styling the component, these subareas have dedicated style objects that can be used to change the look. Furthermore, some parts of the table's property model allow for more fine tuned control of the look. For example, changing the color of all the rows on table can be accomplished by setting a background color on the <code>rows.style</code> object. However, if you wanted to alternate colors on each row, you could instead look towards the <code>rows.striped.color</code> object, which allows you to pick colors for even and odd rows separately.

Editor notes are only visible to logged in users

Removed the following, since it looks like there were never implemented. IGN-215

Freeze Column	If the table head property is enabled, a user can freeze a column by holding down Alt and double clicking the column header. This action "freezes" the column within the bounds of the table so that the user may scroll to perform data comparisons. To unfreeze a column, hold down Alt and double click on the column header of the frozen column or of the source column. A frozen column can be dragged horizontally within the bounds of the table by selecting and dragging with the mouse. It is possible to freeze as many columns as a user may like. The user is not confined to freezing columns that are only visible when at scroll start position.
Freeze Row	A user may freeze an individual row by holding down 'alt' and double clicking on the desired row within the table body. This will fix the table row within the bounds of the table. To unfreeze, perform the same operation on either the frozen row or the source row. A frozen row can be dragged vertically within the bounds of the table by selecting and dragging with the mouse. The user is not confined to freezing rows that are only visible when at scroll start position.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Prop Type
data	Can be a dataset, an array of arrays, or an array of objects. The preferred (recommended) data is either a dataset or an array of objects. Individual data items can be a string, a number, or an object with reserved keys. Object data items must have a value property. Optionally they can also have properties to indicate the style for the object and whether it is editable. <pre>city: { value: 'Folsom', editable: true,</pre>	array

```

style: {
  backgroundColor: 'grey',
  classes: [] | ''
}
}

```

virtualized

Enables virtualization of table rows, which is an optimization method that only shows a portion of the underlying data on the chart at a time.

While enabled, the table will only be populated with a smaller subset of data: just the visible rows, and a few rows above and below. The idea being the component will be populated with new records as the user scrolls down the listing, assuming there are enough records to necessitate a scrollbar.

Enabling virtualization generally results in a performance gain in the session, in cases where the data property is populated with a large amount of content, as the table will only have to "load" a small subset of content. The trade off is that the table will need to load records as the user scrolls, so scrolling quickly may not feel as "smooth" when compared to disabling virtualization.

value: boolean

selection

When Selection is configured, a user will be able to select table data based upon the table's selection configuration. Similar to Vision module, you can select single, single interval, and multiple interval selection modes. The current selection and selection data is written back to the table components property tree. With the exception of the selection data property, the selection properties are bidirectional, meaning that if you were to change the value of the selected column property, it should be reflected in the table component.

object

Name	Description	Property Type
mode	This option determines if only one row, cell, or column can be selected at once. Options are single, single interval, and multiple interval.	value: string
enable Row Selection	This option is used in conjunction with the Column Selection allowed flag in order to determine whether whole rows, whole columns, or both (single cells) are selectable. Can be set to true or false. Default is true.	value: boolean
enable Column Selection	This option is used in conjunction with the Row Selection allowed flag in order to determine whether whole rows, whole columns, or both (single cells) are selectable. Can be set to true or false. Default is false.	value: boolean
selectedColumn	The index of the first selected column, or null if none.	value: numeric
selectedRow	The index of the first selected row, or null if none.	value: numeric
data	An array of objects representing the current selection.	array
style	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</p> </div> <p>Sets a style that applies to the visual selection. This does not impact the highlight style when the mouse is hovered over the table. Full menu of style options is available. You can also specify a style class.</p>	object

filter

Where Table filtering is configured, as well as the filtered data.

object

Name	Description	Property Type						
enabled	Enables filtering. Default is false.	value: boolean						
text	Contains the text you want to filter on.	value: string						
results	The filtered data.	object						
<table border="1" style="width: 100%;"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enables the filter results to be written back to the props. Doing so may</td> <td>value:</td> </tr> </tbody> </table>			Name	Description	Property Type	enabled	Enables the filter results to be written back to the props. Doing so may	value:
Name	Description	Property Type						
enabled	Enables the filter results to be written back to the props. Doing so may	value:						

	<table border="1"> <tr> <td></td> <td>cause performance decline. Default is false.</td> <td>boolean</td> </tr> <tr> <td>data</td> <td>An array of objects representing the current filtered data if filtering is enabled. Each object represents a row of the table.</td> <td>array</td> </tr> </table>		cause performance decline. Default is false.	boolean	data	An array of objects representing the current filtered data if filtering is enabled. Each object represents a row of the table.	array													
	cause performance decline. Default is false.	boolean																		
data	An array of objects representing the current filtered data if filtering is enabled. Each object represents a row of the table.	array																		
style	<p>The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</p> <p>Sets a style that applies to the filter display. Full menu of style options is available. You can also specify a style class.</p>	object																		
enableHeader	When enabled, the table header is displayed including the main table header along with the Header Groups. Default is true.	value: boolean																		
enableFooter	When selected, this enables the table footer, including the main table footer along with the Footer Groups. Default is false.	value: boolean																		
enableHeaderGroups	Enable table header groups if available. Default is false.	value: boolean																		
enableFooterGroups	Enable table footer groups if available. Default is false.	value: boolean																		
headerGroups	Header Groups are additional headers that are displayed above the main table header. Each header group equates to a single row with individual cells containing title text.	object																		
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>title</td> <td>Text displayed in the header.</td> <td>value: string</td> </tr> <tr> <td>span</td> <td>You can use the span property to instruct individual cells to span multiple columns of the table. However, header group cells cannot span more than the available columns. If you specify more columns in the span property than are actually available in the table, the cell will only span the remaining space.</td> <td>value: numeric</td> </tr> <tr> <td>justify</td> <td>Justify content horizontally. Options are left, right, and center.</td> <td>value: string</td> </tr> <tr> <td>align</td> <td>Aligns the content vertically. Options are top, center, or bottom.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Sets a style that applies to header. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	title	Text displayed in the header.	value: string	span	You can use the span property to instruct individual cells to span multiple columns of the table. However, header group cells cannot span more than the available columns. If you specify more columns in the span property than are actually available in the table, the cell will only span the remaining space.	value: numeric	justify	Justify content horizontally. Options are left, right, and center.	value: string	align	Aligns the content vertically. Options are top, center, or bottom.	value: string	style	Sets a style that applies to header. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	
Name	Description	Property Type																		
title	Text displayed in the header.	value: string																		
span	You can use the span property to instruct individual cells to span multiple columns of the table. However, header group cells cannot span more than the available columns. If you specify more columns in the span property than are actually available in the table, the cell will only span the remaining space.	value: numeric																		
justify	Justify content horizontally. Options are left, right, and center.	value: string																		
align	Aligns the content vertically. Options are top, center, or bottom.	value: string																		
style	Sets a style that applies to header. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																		
footerGroups	Footer Groups are additional footers that display above the main table footer. Each footer group equates to a single row which consists of individual cells containing title text.	object																		
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>title</td> <td>Text displayed in the footer group.</td> <td>value: string</td> </tr> <tr> <td>span</td> <td>You can use the span property to instruct individual cells to span multiple columns of the table. However, footer group cells cannot span more than the available columns. If you specify more columns in the span property than are actually available in the table, the cell will only span the remaining space.</td> <td>value: numeric</td> </tr> <tr> <td>justify</td> <td>Justify content horizontally. Options are left, right, and center.</td> <td>value: string</td> </tr> <tr> <td>align</td> <td>Aligns the content vertically. Options are top, center, or bottom.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Sets a style that applies to footer. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	title	Text displayed in the footer group.	value: string	span	You can use the span property to instruct individual cells to span multiple columns of the table. However, footer group cells cannot span more than the available columns. If you specify more columns in the span property than are actually available in the table, the cell will only span the remaining space.	value: numeric	justify	Justify content horizontally. Options are left, right, and center.	value: string	align	Aligns the content vertically. Options are top, center, or bottom.	value: string	style	Sets a style that applies to footer. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	
Name	Description	Property Type																		
title	Text displayed in the footer group.	value: string																		
span	You can use the span property to instruct individual cells to span multiple columns of the table. However, footer group cells cannot span more than the available columns. If you specify more columns in the span property than are actually available in the table, the cell will only span the remaining space.	value: numeric																		
justify	Justify content horizontally. Options are left, right, and center.	value: string																		
align	Aligns the content vertically. Options are top, center, or bottom.	value: string																		
style	Sets a style that applies to footer. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																		
columns		array																		

The Columns property allows for granular column-by-column configurations, indicating how each column should be displayed in the table. Column configs enable the you to customize the table component's display and how users will be able to interact with the table in runtime. When a Column Config option is present, the table reflects that custom configuration, such as a single cell of a Table component being changed from a simple value (like a string) to an embedded view.

You can add Column Config options by selecting **Add Array Element** under the Columns property. By default, the Table component displays all available data in columns, however choosing to customize column configuration will reset the table to a

single column display. Columns will have to be manually added back into the table using the Add Array Element icon  on the right of the Columns property.

For examples on how column configurations work, see [Table Column Configurations](#).

Name	Description	Property Type																								
field	A string that matches this column config with a table column. This string must correspond to the default column name of the column.	value: string																								
visible	Toggles column visibility. Allows table columns to be invisible to users, but data will be available to view params and selection.	value: boolean																								
editable	Enables editing of all cells within this column. This can be overridden if the Editable property is set to false on an individual cell.	value: boolean																								
render	The default render setting is auto. Can be auto, number, date, boolean, string, or view. When set to "view", the adjacent viewPath and viewParams properties can be used to specify which view, and set values on view parameters for the nested view.	value: string																								
justify	Sets the justification for the content of the column. Options are left, center, right, or auto. The default setting is auto.	value: string dropdown																								
align	Sets the alignment for the content of the column. Options are top, center, or bottom. The default alignment is center.	value: string																								
resizable	Enables columns to be resized. When enabled, users can resize columns in the runtime by hovering over the edge of the column header then dragging the  cursor.	value: boolean																								
sortable	Enables the column to be sorted. When enabled, users can double click on the column header in the run time to sort by ascending or descending order.	value: boolean																								
filter	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.22 Click here to check out the other new features</p> </div> <p>Column-specific filtering configuration.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>When true, will apply any valid column filters configured here.</td> <td>boolean</td> </tr> <tr> <td>visible</td> <td>Specifies how the filter icon in the column header is visible to the user. Unless "never", will always be shown if a mobile device is connected.</td> <td>value: string dropdown</td> </tr> <tr> <td>string</td> <td>String type column filter.</td> <td>object</td> </tr> <tr> <td colspan="3" style="text-align: center;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>condition</td> <td>The conditions by which the string filter will apply. Possible values include: <ul style="list-style-type: none"> contains equals starts with ends with </td> <td>value: string dropdown</td> </tr> <tr> <td>value</td> <td>The specific string value that will be used for the filter.</td> <td>value: string</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	When true, will apply any valid column filters configured here.	boolean	visible	Specifies how the filter icon in the column header is visible to the user. Unless "never", will always be shown if a mobile device is connected.	value: string dropdown	string	String type column filter.	object	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>condition</td> <td>The conditions by which the string filter will apply. Possible values include: <ul style="list-style-type: none"> contains equals starts with ends with </td> <td>value: string dropdown</td> </tr> <tr> <td>value</td> <td>The specific string value that will be used for the filter.</td> <td>value: string</td> </tr> </tbody> </table>			Name	Description	Property Type	condition	The conditions by which the string filter will apply. Possible values include: <ul style="list-style-type: none"> contains equals starts with ends with 	value: string dropdown	value	The specific string value that will be used for the filter.	value: string	object
Name	Description	Property Type																								
enabled	When true, will apply any valid column filters configured here.	boolean																								
visible	Specifies how the filter icon in the column header is visible to the user. Unless "never", will always be shown if a mobile device is connected.	value: string dropdown																								
string	String type column filter.	object																								
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>condition</td> <td>The conditions by which the string filter will apply. Possible values include: <ul style="list-style-type: none"> contains equals starts with ends with </td> <td>value: string dropdown</td> </tr> <tr> <td>value</td> <td>The specific string value that will be used for the filter.</td> <td>value: string</td> </tr> </tbody> </table>			Name	Description	Property Type	condition	The conditions by which the string filter will apply. Possible values include: <ul style="list-style-type: none"> contains equals starts with ends with 	value: string dropdown	value	The specific string value that will be used for the filter.	value: string															
Name	Description	Property Type																								
condition	The conditions by which the string filter will apply. Possible values include: <ul style="list-style-type: none"> contains equals starts with ends with 	value: string dropdown																								
value	The specific string value that will be used for the filter.	value: string																								

number	Number type column filter.	object									
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>condition</td> <td>The conditions by which the number filter will apply. Possible values include: <ul style="list-style-type: none"> greater than greater than or equal to less than less than or equal to equals between </td> <td>value: string dropdown</td> </tr> <tr> <td>value</td> <td>The specific number value that will be used for the filter.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	condition	The conditions by which the number filter will apply. Possible values include: <ul style="list-style-type: none"> greater than greater than or equal to less than less than or equal to equals between 	value: string dropdown	value	The specific number value that will be used for the filter.	value: string	
Name	Description	Property Type									
condition	The conditions by which the number filter will apply. Possible values include: <ul style="list-style-type: none"> greater than greater than or equal to less than less than or equal to equals between 	value: string dropdown									
value	The specific number value that will be used for the filter.	value: string									

boolean	Boolean type column filter.	object						
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>condition</td> <td>The condition by which the boolean filter will apply. Possible values include: <ul style="list-style-type: none"> true false </td> <td>value: string dropdown</td> </tr> </tbody> </table>	Name	Description	Property Type	condition	The condition by which the boolean filter will apply. Possible values include: <ul style="list-style-type: none"> true false 	value: string dropdown	
Name	Description	Property Type						
condition	The condition by which the boolean filter will apply. Possible values include: <ul style="list-style-type: none"> true false 	value: string dropdown						

date	Date type column filter.	object									
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>condition</td> <td>The condition by which the date filter will apply. Possible values include: <ul style="list-style-type: none"> date equals date time equals before date before date time before or equal to date before or equal to date time after date after date time after or equal to date after or equal to date time between dates between date times </td> <td>value: string dropdown</td> </tr> <tr> <td>value</td> <td>The specific date time that will be used for the filter.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	condition	The condition by which the date filter will apply. Possible values include: <ul style="list-style-type: none"> date equals date time equals before date before date time before or equal to date before or equal to date time after date after date time after or equal to date after or equal to date time between dates between date times 	value: string dropdown	value	The specific date time that will be used for the filter.	value: string	
Name	Description	Property Type									
condition	The condition by which the date filter will apply. Possible values include: <ul style="list-style-type: none"> date equals date time equals before date before date time before or equal to date before or equal to date time after date after date time after or equal to date after or equal to date time between dates between date times 	value: string dropdown									
value	The specific date time that will be used for the filter.	value: string									

viewPath	When render mode is set to View, the table will display the view found at the view path within each cell of this column.	value: string
----------	--	---------------

viewParams	Parameters to feed the configured view. Will be added to implicit parameters as follows: {row:number; rowIndex:number;value:PlainObject;...viewParams}	object
------------	--	--------

boolean	When render mode is set to Boolean, you can then specify how the boolean is represented in the runtime, for example, as a checkbox, toggle switch, value, and so forth. See Toggle Switch below.	value: string
---------	--	---------------

number	Type of component to render for boolean. Options are number or progress. When render mode is set to Number, you can then specify whether the number is represented in the runtime as value or as progress.	value: string
--------	--	---------------

progressBar	A progress bar configuration that is used when Number property is set to progress bar. You can specify the maximum value of the progress bar, as well as configure the following:	object						
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>max</td> <td>Progress bar maximum value.</td> <td>value:</td> </tr> </tbody> </table>	Name	Description	Property Type	max	Progress bar maximum value.	value:	
Name	Description	Property Type						
max	Progress bar maximum value.	value:						

		numeric															
bar	Settings for the bar. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color of the bar and the track. See Color Selector.</td> <td>color</td> </tr> <tr> <td>style</td> <td>Sets a style for the progress bar. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color of the bar and the track. See Color Selector .	color	style	Sets a style for the progress bar. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	object						
Name	Description	Property Type															
color	Color of the bar and the track. See Color Selector .	color															
style	Sets a style for the progress bar. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															
track	Settings for the track. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color of the bar and the track. See Color Selector.</td> <td>color</td> </tr> <tr> <td>style</td> <td>Sets a style for the progress bar. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color of the bar and the track. See Color Selector .	color	style	Sets a style for the progress bar. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	object						
Name	Description	Property Type															
color	Color of the bar and the track. See Color Selector .	color															
style	Sets a style for the progress bar. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															
value	Settings for the value on the Progress Bar. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether or not to show the value.</td> <td>value: boolean</td> </tr> <tr> <td>format</td> <td>Format to apply to the value.</td> <td>value: string</td> </tr> <tr> <td>justify</td> <td>Horizontal alignment of the value.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Sets a style for the value. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Whether or not to show the value.	value: boolean	format	Format to apply to the value.	value: string	justify	Horizontal alignment of the value.	value: string	style	Sets a style for the value. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	object
Name	Description	Property Type															
enabled	Whether or not to show the value.	value: boolean															
format	Format to apply to the value.	value: string															
justify	Horizontal alignment of the value.	value: string															
style	Sets a style for the value. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															
toggleSwitch	Toggle switch configuration used when boolean is set to display as a toggle switch. Can specify selected and unselected color. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color.selected</td> <td>Color when the toggle switch is selected. See Color Selector.</td> <td>color</td> </tr> <tr> <td>color.unselected</td> <td>Color when the toggle switch is not selected. See Color Selector.</td> <td>color</td> </tr> </tbody> </table>	Name	Description	Property Type	color.selected	Color when the toggle switch is selected. See Color Selector .	color	color.unselected	Color when the toggle switch is not selected. See Color Selector .	color	value: boolean						
Name	Description	Property Type															
color.selected	Color when the toggle switch is selected. See Color Selector .	color															
color.unselected	Color when the toggle switch is not selected. See Color Selector .	color															
numberFormat	A number format string when render mode is set to number. Options are none, number [1,000.12], integer [1,200], four decimal precision [1.1200], percent [10.12%], scientific [1.01E+03], accounting [\$(1,000.12)], financial [(1,000.12)], currency [\$1,000.12], currency (rounded) [\$1,012], duration [24:01:00], abbreviation [1.2k], or ordinal [100th].	value: string															
dateFormat	Date format string used when render mode is set to date. Options are none, date [10/15/1018], time [3:59:00 PM], or date time [10/15/2018 15:59:00]	value: string															
width	The width of this column. If resize is enabled, specifies the column width on initial load. User can override this in the runtime if the Resizable option is enabled.	value: numeric															
strictWidth	If enabled, the width of the column becomes fixed.	value: boolean															

style	<p>Sets a style for this individual column. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p><u>This feature was changed in Ignition version 8.1.11:</u></p> </div> <p>As of 8.1.11, this style is applied to the header, cells, and footer of the entire column.</p>	object															
header	<p>Header cell configuration.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Name</th> <th style="width: 60%;">Description</th> <th style="width: 25%;">Property Type</th> </tr> </thead> <tbody> <tr> <td>title</td> <td>Text for title of the column.</td> <td>value: string</td> </tr> <tr> <td>justify</td> <td>Setting for justification of the title. Options are right, left, and center.</td> <td>value: string dropdown</td> </tr> <tr> <td>align</td> <td>Setting for alignment of the title. Options are top, center, and bottom.</td> <td>value: string dropdown</td> </tr> <tr> <td>style</td> <td>Sets a style for this header. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	title	Text for title of the column.	value: string	justify	Setting for justification of the title. Options are right, left, and center.	value: string dropdown	align	Setting for alignment of the title. Options are top, center, and bottom.	value: string dropdown	style	Sets a style for this header. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	value: object
Name	Description	Property Type															
title	Text for title of the column.	value: string															
justify	Setting for justification of the title. Options are right, left, and center.	value: string dropdown															
align	Setting for alignment of the title. Options are top, center, and bottom.	value: string dropdown															
style	Sets a style for this header. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															
footer	<p>Footer cell configuration.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Name</th> <th style="width: 60%;">Description</th> <th style="width: 25%;">Property Type</th> </tr> </thead> <tbody> <tr> <td>title</td> <td>Text for title of the footer.</td> <td>value: string</td> </tr> <tr> <td>justify</td> <td>Setting for justification of the title. Options are right, left, and center.</td> <td>value: string dropdown</td> </tr> <tr> <td>align</td> <td>Setting for alignment of the title. Options are top, center, and bottom.</td> <td>value: string dropdown</td> </tr> <tr> <td>style</td> <td>Sets a style for this footer. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	title	Text for title of the footer.	value: string	justify	Setting for justification of the title. Options are right, left, and center.	value: string dropdown	align	Setting for alignment of the title. Options are top, center, and bottom.	value: string dropdown	style	Sets a style for this footer. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	value: string
Name	Description	Property Type															
title	Text for title of the footer.	value: string															
justify	Setting for justification of the title. Options are right, left, and center.	value: string dropdown															
align	Setting for alignment of the title. Options are top, center, and bottom.	value: string dropdown															
style	Sets a style for this footer. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															
Implicit Properties																	
rowData	Used to map parameters on a view cell to an entire row of data. The view must have a rowData object input parameter, with sub values that match the names of the columns. Then add the new view to the props.columns.0.viewPath property, and the input parameter as the props.columns.0.field property.	value: string															

Editor notes are only visible to logged in users
Add the following row above progressBar upon 8.1.25 release

nullFormat	<div style="border: 1px solid orange; padding: 5px; margin: 10px 0;"> <p>The following feature is new in Ignition version 8.1.25 Click here to check out the other new features</p> </div> <p>The column null format configuration used when a column contains either a "null" string value or blank cell data. These property settings override the table nullFormat property.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tbody> <tr> <td style="width: 15%;">includeNullStrings</td> <td style="width: 60%;">Toggles inclusion of "null" strings in null format logic. Default value is false.</td> <td style="width: 25%;">value: boolean</td> </tr> <tr> <td>strict</td> <td>Overrides render mode and apply nullFormatValue when enabled.</td> <td>value: boolean</td> </tr> </tbody> </table>	includeNullStrings	Toggles inclusion of "null" strings in null format logic. Default value is false.	value: boolean	strict	Overrides render mode and apply nullFormatValue when enabled.	value: boolean	object
includeNullStrings	Toggles inclusion of "null" strings in null format logic. Default value is false.	value: boolean						
strict	Overrides render mode and apply nullFormatValue when enabled.	value: boolean						

nullForm atValue	Value to be applied against null values (or "null" strings if includeNullStrings is set to true), and includes three build-in options: blank, N/A, and null.	object
------------------	--	--------

dragOrderable

The following feature is new in Ignition version **8.1.14**
[Click here](#) to check out the other new features

When enabled, users may drag column headers to reorder columns in the table if [Column Config](#) options are present.

value: boolean

sortOrder

The default weighted order in which columns and their contents are sorted relative to other columns and their contents. Used when the component loads.

For **sortOrder** to be applied, the table must meet the following requirements

- Objects under the **columns** array must be defined for each column in the table's underlying data property you wish to display and sort on.
 - In addition, each object under columns must have the **field** setting set to the data item under the data property (for example, "population" in the table's default data set).
 - **sortable** must be enabled
 - **sort** must be set to something other than none

Once all columns have been configured, the sortOrder can be configured.

Each element in the **sortOrder** array is expected to be a string value representing the name of the column (as determined by **field** value in the columns array). For example, sorting by population first, city second, and country last, would look like the following:

```

sortOrder [3]
  0 : population
  1 : city
  2 : country

```

array

rows

Configures all rows in the table component. Includes settings for expanding rows into subviews.

object

Name	Description	Property Type												
height	<p>The following feature is new in Ignition version 8.1.6 Click here to check out the other new features</p> <p>A minimum height value applied to all rows. A row cannot be shorter than this value, but it can be taller based on the height of the content it displays. This property can be set to "auto" (the default value) or given a numerical value that will correspond to a minimum row height in pixels.</p>	value: string numeric												
subviewExpansionMode	Specifies how many subviews can be expanded at any given time. Options are multiple or single. Default is multiple.	value: string												
subview	When enabled, each table row can be expanded into a subview. The Expandable Arrow  opens the subview. Content of the subview is determined by the View Path property.	object												
<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enable each row to allow toggling of the specified view.</td> <td>value: boolean</td> </tr> <tr> <td>viewPath</td> <td>A viewpath used to display a view as an expanded row</td> <td>value: string</td> </tr> <tr> <td>viewParams</td> <td>Parameters to feed the configured view. Parameters specified here will be passed to the root of the "params" category of properties on the sub view.</td> <td>object</td> </tr> </tbody> </table>			Name	Description	Property Type	enabled	Enable each row to allow toggling of the specified view.	value: boolean	viewPath	A viewpath used to display a view as an expanded row	value: string	viewParams	Parameters to feed the configured view. Parameters specified here will be passed to the root of the "params" category of properties on the sub view.	object
Name	Description	Property Type												
enabled	Enable each row to allow toggling of the specified view.	value: boolean												
viewPath	A viewpath used to display a view as an expanded row	value: string												
viewParams	Parameters to feed the configured view. Parameters specified here will be passed to the root of the "params" category of properties on the sub view.	object												

In addition to the properties above, subviews will be passed implicit parameters provided by the row:

- row - a number representing the row
- rowIndex - a number representing the index of the current row
- value - JSON Object representing the contents of the table. The value of each column in the table will be a value under this object. Example: value.population.

striped	Settings for setting the striping (alternating background color) to the rows of the table.	object																					
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>When enabled (true), the table will be displayed with alternating background color to the rows of the table.</td> <td>value: boolean</td> </tr> <tr> <td>color</td> <td>Color settings</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>even</td> <td>Background color for the even rows. See Color Selector.</td> <td>color</td> </tr> <tr> <td>odd</td> <td>Background color for the odd rows. See Color Selector.</td> <td>color</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	When enabled (true), the table will be displayed with alternating background color to the rows of the table.	value: boolean	color	Color settings	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>even</td> <td>Background color for the even rows. See Color Selector.</td> <td>color</td> </tr> <tr> <td>odd</td> <td>Background color for the odd rows. See Color Selector.</td> <td>color</td> </tr> </tbody> </table>	Name	Description	Property Type	even	Background color for the even rows. See Color Selector .	color	odd	Background color for the odd rows. See Color Selector .	color		
Name	Description	Property Type																					
enabled	When enabled (true), the table will be displayed with alternating background color to the rows of the table.	value: boolean																					
color	Color settings	object																					
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>even</td> <td>Background color for the even rows. See Color Selector.</td> <td>color</td> </tr> <tr> <td>odd</td> <td>Background color for the odd rows. See Color Selector.</td> <td>color</td> </tr> </tbody> </table>	Name	Description	Property Type	even	Background color for the even rows. See Color Selector .	color	odd	Background color for the odd rows. See Color Selector .	color													
Name	Description	Property Type																					
even	Background color for the even rows. See Color Selector .	color																					
odd	Background color for the odd rows. See Color Selector .	color																					
highlight	Highlight settings.	object																					
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>When enabled (true), this feature will highlight the row that is currently selected or moused over.</td> <td>value: boolean</td> </tr> <tr> <td>color</td> <td>Highlight color for the row. See Color Selector.</td> <td>color</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	When enabled (true), this feature will highlight the row that is currently selected or moused over.	value: boolean	color	Highlight color for the row. See Color Selector .	color													
Name	Description	Property Type																					
enabled	When enabled (true), this feature will highlight the row that is currently selected or moused over.	value: boolean																					
color	Highlight color for the row. See Color Selector .	color																					
style	Sets a style that applies to every row in the table. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																					

cells	Configures all cells in the table component.	object									
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>allowEditOn</td> <td>Enables the table cells to be edited on a single click, double click, or long press.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Sets a style that applies to every cell in the table. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	allowEditOn	Enables the table cells to be edited on a single click, double click, or long press.	value: string	style	Sets a style that applies to every cell in the table. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	
Name	Description	Property Type									
allowEditOn	Enables the table cells to be edited on a single click, double click, or long press.	value: string									
style	Sets a style that applies to every cell in the table. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object									

editingCell	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.37 Click here to check out the other new features</p> </div> <p>Allows a user to specify the column and row of the current editing cell when editing is enabled.</p>	object
-------------	---	--------

nullFormat	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.25 Click here to check out the other new features</p> </div> <p>The table null format configuration used when a table contains either a "null" string value or blank cell data. Can be overridden by individual column nullFormat configuration.</p>	object									
	<table border="1"> <tbody> <tr> <td>includeNullStrings</td> <td>Toggles inclusion of "null" strings in null format logic. Default value is false.</td> <td>value: boolean</td> </tr> <tr> <td>strict</td> <td>Overrides render mode and apply nullFormatValue when enabled.</td> <td>value: boolean</td> </tr> <tr> <td>nullFormat</td> <td>Value to be applied against null values (or "null" strings if includeNullStrings is set to true), and</td> <td>object</td> </tr> </tbody> </table>	includeNullStrings	Toggles inclusion of "null" strings in null format logic. Default value is false.	value: boolean	strict	Overrides render mode and apply nullFormatValue when enabled.	value: boolean	nullFormat	Value to be applied against null values (or "null" strings if includeNullStrings is set to true), and	object	
includeNullStrings	Toggles inclusion of "null" strings in null format logic. Default value is false.	value: boolean									
strict	Overrides render mode and apply nullFormatValue when enabled.	value: boolean									
nullFormat	Value to be applied against null values (or "null" strings if includeNullStrings is set to true), and	object									

Value	includes three build-in options: blank, N/A, and null.
-------	--

pager

Enables table pagination. Pagination improves performance and appearance on large tables, over 1000 rows.

Name	Description	Property Type
options	Rows to show per pager option.	array
initialOption	Initial option to use when the table first loads. Must exist as an available option.	value: numeric
top	Enables top pager. The pager is a menu that displays the current page and Previous < and Next > icons for navigation.	value: boolean
bottom	Enables bottom pager. The pager is a menu that displays the current page and Previous < and Next > icons for navigation.	value: boolean
activePage	Represents the current active page and corresponds to the value of the page jump input field.	value: numeric
style	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</p> </div> <p>Sets a style that applies to the pager container. Full menu of style options is available. You can also specify a style class.</p>	object

resizeMode

Specifies whether the table resize mode is either Fill or Fixed. In Fill resized mode, the total width of all the columns cannot be less than the width of the table. In Fixed resized mode, the total width of all the columns can be less than the width of the table.

value:
boolean

style

Sets a style that applies to the component. Full menu of [style options](#) is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a [style class](#).

object

emptyMessage

Empty message configuration.

noData	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.2 Click here to check out the other new features</p> </div> <p>Empty message configuration for when there is either no data source or the data source is empty.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Text to display when there is no data source or the data source is empty.</td> <td>value: string</td> </tr> <tr> <td>textStyle</td> <td>Sets a style that applies to the text. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>icon</td> <td>Settings for the icon to be displayed when there is no data source or the data source is empty.</td> <td>object</td> </tr> <tr> <td colspan="2"> <table border="1"> <tr> <td>path</td> <td>Shorthand path to the icon source, in format: library/iconName.</td> <td>value: string</td> </tr> <tr> <td>color</td> <td>Color of the icon. Alternatively, you can use fill settings in the style property.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Sets a style that applies to the icon. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </table> </td> <td></td> </tr> <tr> <td>bottom</td> <td>Enables bottom pager. The pager is a menu that displays the current page and Previous < and Next > icons for navigation.</td> <td>value: boolean</td> </tr> <tr> <td>activePage</td> <td>Represents the current active page and corresponds to the value of the page jump input field.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	text	Text to display when there is no data source or the data source is empty.	value: string	textStyle	Sets a style that applies to the text. Full menu of style options is available. You can also specify a style class .	object	icon	Settings for the icon to be displayed when there is no data source or the data source is empty.	object	<table border="1"> <tr> <td>path</td> <td>Shorthand path to the icon source, in format: library/iconName.</td> <td>value: string</td> </tr> <tr> <td>color</td> <td>Color of the icon. Alternatively, you can use fill settings in the style property.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Sets a style that applies to the icon. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </table>		path	Shorthand path to the icon source, in format: library/iconName.	value: string	color	Color of the icon. Alternatively, you can use fill settings in the style property.	value: string	style	Sets a style that applies to the icon. Full menu of style options is available. You can also specify a style class .	object		bottom	Enables bottom pager. The pager is a menu that displays the current page and Previous < and Next > icons for navigation.	value: boolean	activePage	Represents the current active page and corresponds to the value of the page jump input field.	value: numeric	object
Name	Description	Property Type																														
text	Text to display when there is no data source or the data source is empty.	value: string																														
textStyle	Sets a style that applies to the text. Full menu of style options is available. You can also specify a style class .	object																														
icon	Settings for the icon to be displayed when there is no data source or the data source is empty.	object																														
<table border="1"> <tr> <td>path</td> <td>Shorthand path to the icon source, in format: library/iconName.</td> <td>value: string</td> </tr> <tr> <td>color</td> <td>Color of the icon. Alternatively, you can use fill settings in the style property.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Sets a style that applies to the icon. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </table>		path	Shorthand path to the icon source, in format: library/iconName.	value: string	color	Color of the icon. Alternatively, you can use fill settings in the style property.	value: string	style	Sets a style that applies to the icon. Full menu of style options is available. You can also specify a style class .	object																						
path	Shorthand path to the icon source, in format: library/iconName.	value: string																														
color	Color of the icon. Alternatively, you can use fill settings in the style property.	value: string																														
style	Sets a style that applies to the icon. Full menu of style options is available. You can also specify a style class .	object																														
bottom	Enables bottom pager. The pager is a menu that displays the current page and Previous < and Next > icons for navigation.	value: boolean																														
activePage	Represents the current active page and corresponds to the value of the page jump input field.	value: numeric																														

noFilter Results	<div data-bbox="370 132 1279 220" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.2 Click here to check out the other new features</p> </div> <p>Empty message configuration for when a filter returns no results.</p> <table border="1" data-bbox="370 283 1352 787"> <thead> <tr> <th data-bbox="370 283 462 357">Name</th> <th data-bbox="462 283 1221 357">Description</th> <th data-bbox="1221 283 1352 357">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="370 357 462 430">text</td> <td data-bbox="462 357 1221 430">Text to display when a filter returns no results.</td> <td data-bbox="1221 357 1352 430">value: string</td> </tr> <tr> <td data-bbox="370 430 462 493">textStyle</td> <td data-bbox="462 430 1221 493">Sets a style that applies to the text. Full menu of style options is available. You can also specify a style class.</td> <td data-bbox="1221 430 1352 493">object</td> </tr> <tr> <td data-bbox="370 493 462 787" rowspan="4">icon</td> <td data-bbox="462 493 1221 546">Settings for the icon to be displayed when a filter returns no results.</td> <td data-bbox="1221 493 1352 787" rowspan="4">object</td> </tr> <tr> <td data-bbox="462 546 1128 619"> <table border="1" data-bbox="470 556 1209 619"> <tr> <td data-bbox="470 556 511 619">p ath</td> <td data-bbox="511 556 1128 619">Shorthand path to the icon source, in format: library/iconName.</td> <td data-bbox="1128 556 1209 619">value: string</td> </tr> </table> </td> </tr> <tr> <td data-bbox="470 619 1128 703"> <table border="1" data-bbox="470 619 1209 703"> <tr> <td data-bbox="470 619 511 703">c ol or</td> <td data-bbox="511 619 1128 703">Color of the icon. Alternatively, you can use fill settings in the style property.</td> <td data-bbox="1128 619 1209 703">value: string</td> </tr> </table> </td> </tr> <tr> <td data-bbox="470 703 1128 787"> <table border="1" data-bbox="470 703 1209 787"> <tr> <td data-bbox="470 703 511 787">st yle</td> <td data-bbox="511 703 1128 787">Sets a style that applies to the icon. Full menu of style options is available. You can also specify a style class.</td> <td data-bbox="1128 703 1209 787">object</td> </tr> </table> </td> </tr> </tbody> </table>	Name	Description	Property Type	text	Text to display when a filter returns no results.	value: string	textStyle	Sets a style that applies to the text. Full menu of style options is available. You can also specify a style class .	object	icon	Settings for the icon to be displayed when a filter returns no results.	object	<table border="1" data-bbox="470 556 1209 619"> <tr> <td data-bbox="470 556 511 619">p ath</td> <td data-bbox="511 556 1128 619">Shorthand path to the icon source, in format: library/iconName.</td> <td data-bbox="1128 556 1209 619">value: string</td> </tr> </table>	p ath	Shorthand path to the icon source, in format: library/iconName.	value: string	<table border="1" data-bbox="470 619 1209 703"> <tr> <td data-bbox="470 619 511 703">c ol or</td> <td data-bbox="511 619 1128 703">Color of the icon. Alternatively, you can use fill settings in the style property.</td> <td data-bbox="1128 619 1209 703">value: string</td> </tr> </table>	c ol or	Color of the icon. Alternatively, you can use fill settings in the style property.	value: string	<table border="1" data-bbox="470 703 1209 787"> <tr> <td data-bbox="470 703 511 787">st yle</td> <td data-bbox="511 703 1128 787">Sets a style that applies to the icon. Full menu of style options is available. You can also specify a style class.</td> <td data-bbox="1128 703 1209 787">object</td> </tr> </table>	st yle	Sets a style that applies to the icon. Full menu of style options is available. You can also specify a style class .	object	obj ect
Name	Description	Property Type																								
text	Text to display when a filter returns no results.	value: string																								
textStyle	Sets a style that applies to the text. Full menu of style options is available. You can also specify a style class .	object																								
icon	Settings for the icon to be displayed when a filter returns no results.	object																								
	<table border="1" data-bbox="470 556 1209 619"> <tr> <td data-bbox="470 556 511 619">p ath</td> <td data-bbox="511 556 1128 619">Shorthand path to the icon source, in format: library/iconName.</td> <td data-bbox="1128 556 1209 619">value: string</td> </tr> </table>		p ath	Shorthand path to the icon source, in format: library/iconName.	value: string																					
	p ath		Shorthand path to the icon source, in format: library/iconName.	value: string																						
	<table border="1" data-bbox="470 619 1209 703"> <tr> <td data-bbox="470 619 511 703">c ol or</td> <td data-bbox="511 619 1128 703">Color of the icon. Alternatively, you can use fill settings in the style property.</td> <td data-bbox="1128 619 1209 703">value: string</td> </tr> </table>		c ol or	Color of the icon. Alternatively, you can use fill settings in the style property.	value: string																					
c ol or	Color of the icon. Alternatively, you can use fill settings in the style property.	value: string																								
<table border="1" data-bbox="470 703 1209 787"> <tr> <td data-bbox="470 703 511 787">st yle</td> <td data-bbox="511 703 1128 787">Sets a style that applies to the icon. Full menu of style options is available. You can also specify a style class.</td> <td data-bbox="1128 703 1209 787">object</td> </tr> </table>	st yle	Sets a style that applies to the icon. Full menu of style options is available. You can also specify a style class .	object																							
st yle	Sets a style that applies to the icon. Full menu of style options is available. You can also specify a style class .	object																								
style	<div data-bbox="370 829 1279 917" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</p> </div> <p>Sets a style that applies to the empty message display area. Full menu of style options is available. You can also specify a style class.</p>	obj ect																								
header Style	<div data-bbox="256 1081 1166 1169" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</p> </div> <p>Sets a style that applies to all column headers. Can be overridden by both columns.style and columns.header.style properties. Full menu of style options is available. You can also specify a style class.</p>	object																								
header GroupStyle	<div data-bbox="256 1323 1166 1411" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</p> </div> <p>Sets a style that applies to all headerGroups. Full menu of style options is available. You can also specify a style class.</p>	object																								
bodyStyle	<div data-bbox="256 1512 1166 1600" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</p> </div> <p>Sets a style that applies to the table body. Full menu of style options is available. You can also specify a style class.</p>	object																								
footerStyle	<div data-bbox="256 1701 1166 1789" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</p> </div> <p>Sets a style that applies to all column footers. Can be overridden by both columns.style and columns.footer.style properties. Full menu of style options is available. You can also specify a style class.</p>	object																								
footerGroupStyle	<div data-bbox="256 1921 1166 1969" style="border: 1px solid orange; padding: 5px;"> <p>The following feature is new in Ignition version 8.1.11 Click here to check out the other new features</p> </div>	object																								

The following feature is new in Ignition version **8.1.11**
[Click here](#) to check out the other new features

Sets a style that applies to all footerGroups. Full menu of [style options](#) is available. You can also specify a [style class](#).

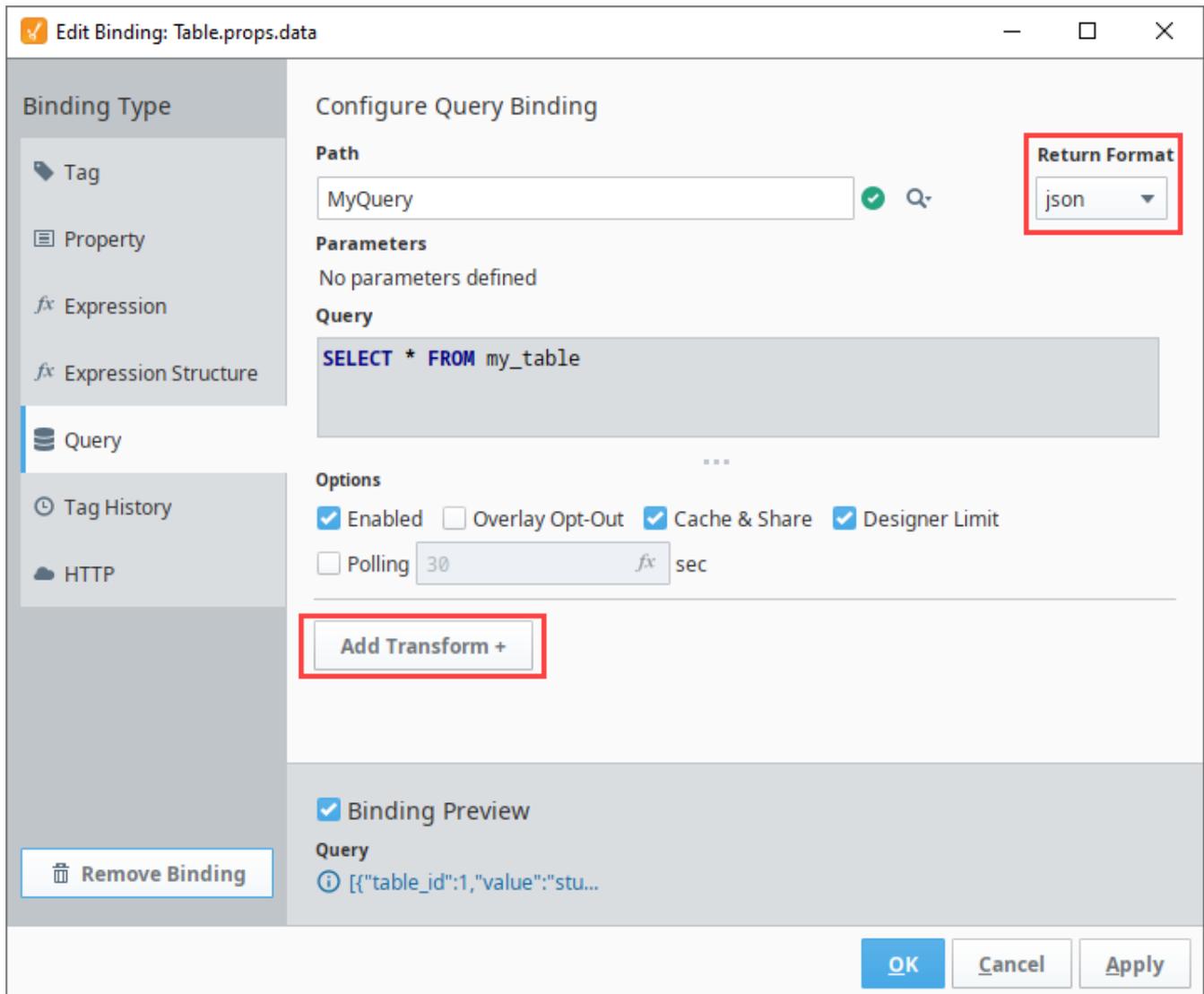
Scripting

See the [Perspective - Table Scripting page](#) for the full list of scripting functions available for this component.

Examples

Example 1: Styling Rows Based on Value - JSON Data

It is possible to change the styling on rows based upon the value in a row with a Script Transform. For example, if the data property on the table component has a Query binding configured, we can set the Return Format to JSON, and then add a Script Transform:



The screenshot shows the 'Edit Binding: Table.props.data' dialog box. The 'Return Format' dropdown is set to 'json' and is highlighted with a red box. The 'Add Transform +' button is also highlighted with a red box. The dialog shows a query 'SELECT * FROM my_table' and options for 'Enabled', 'Cache & Share', and 'Designer Limit'.

In the Script Transform, we can add code that adds a styling to each value in the Named Query results:

Add Styling to each Value Returned by a Named Query

```
# Create a new list to store a modified result set from our query
newData = []
```

```

# Iterate over each row in 'value', which is the original result set
for row in value:

    # Within the row, iterate over each column
    for col in row:

        # Create a variable to store the contents of the original "cell"
        cell = row[col]

        # Create a dictionary containing the original value, and some styling information
        row[col] = {"value": cell, "style": {"backgroundColor": "#00FF00"}}

    # Add the modified row to the list we initialized earlier
    newData.append(row)

#Return the list
return newData

```

Example 2: Styling Rows Based on Value - Dataset Data

You can also change the color of rows in a table when the source of data is a dataset. This approach involves recreating the original data as a new JSON document that contains a style object for each row.

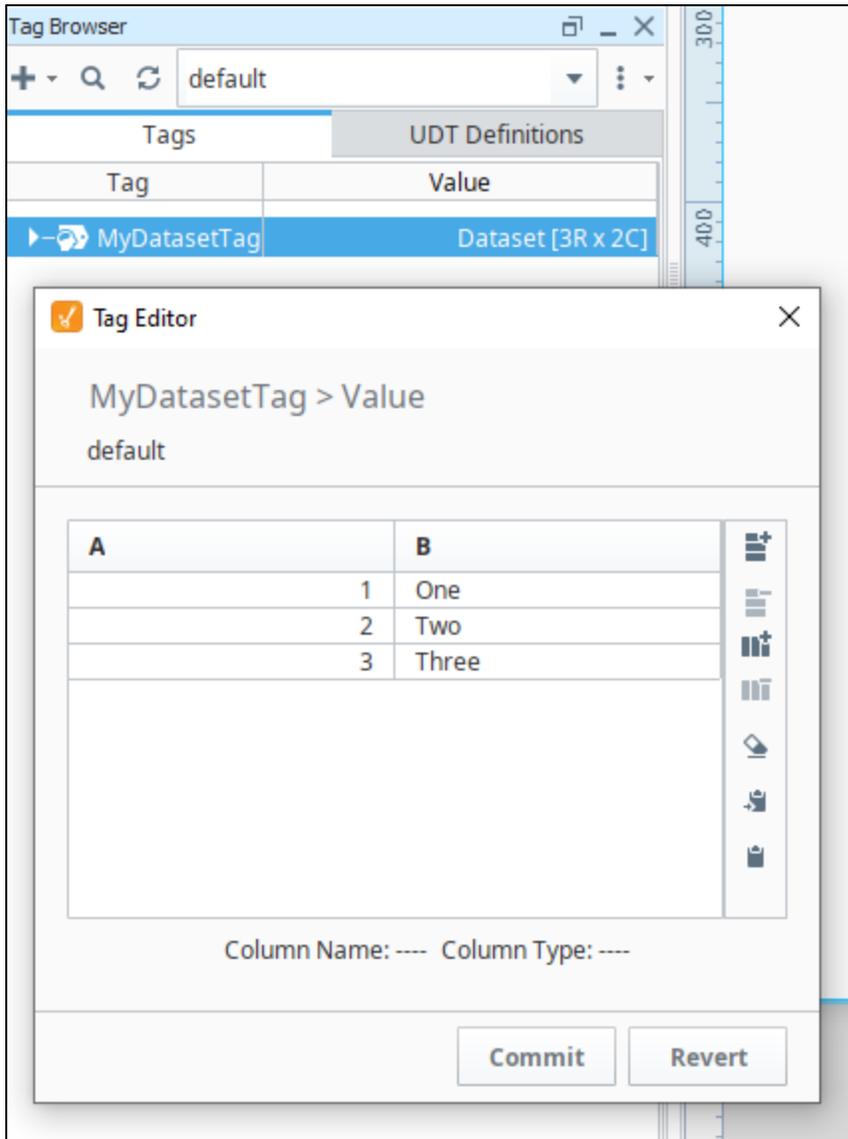
For example, say there's a memory tag set to a dataset type, which contains the following data:

A	B
1	One
2	Two
3	Three

```

{
  "valueSource": "memory",
  "dataType": "DataSet",
  "name": "MyDataSetTag",
  "value": "{ \"columns\": [{ \"name\": \"A\", \"type\": \"java.lang.Integer\" }, { \"name\": \"B\", \"type\": \"java.lang.String\" } ], \"rows\": [[ 1, \"One\" ], [ 2, \"Two\" ], [ 3, \"Three\" ] ] }",
  "tagType": "AtomicTag"
}
{
  "valueSource": "memory",
  "dataType": "DataSet",
  "name": "MyDataSetTag",
  "value": "{ \"columns\": [{ \"name\": \"A\", \"type\": \"java.lang.Integer\" }, { \"name\": \"B\", \"type\": \"java.lang.String\" } ], \"rows\": [[ 1, \"One\" ], [ 2, \"Two\" ], [ 3, \"Three\" ] ] }",
  "tagType": "AtomicTag"
}

```



We can bind the table component's props.data property to this tag with a Tag Binding. From here we can add a script transform with the following:

```

# This list will be used to create a JSON like structure that will insert rows for our styles
output_json = []

# Here we can define what styling on our rows will be.
style_orange = {"backgroundColor": "#F7901D"}
style_green = {"backgroundColor": "#00AA00"}

# You could change more than just the background color, for example:
# style_another_example {"backgroundColor": "#00AA00", "font-weight": "bold"}

for row in range(value.getRowCount()):
    row_object = {}
    row_value = {}
    row_style = {}
    for col in range(value.getColumnCount()):
        row_value[value.getColumnName(col)] = value.getValueAt(row, col)
        row_object['value'] = row_value

        # Here we're checking the name of the column that we want to base our styling on.
        if value.getColumnName(col) == 'B':

            # Here we're checking for individual values within the column, and applying
styling

```

```

        if value.getValueAt(row, col) == 'One':
            row_style = style_orange
        elif value.getValueAt(row, col) == 'Two':
            row_style = style_green

        row_object['style'] = row_style
        output_json.append(row_object)
    return output_json

```

Edit Binding: Table.props.data

Binding Type

- Tag
- Property
- Expression
- Expression Structure
- Query
- Tag History
- HTTP

Configure Tag Binding

Direct Indirect Expression

Tag Path: [default]MyDatasetTag

Options

Enabled Overlay Opt-Out Bidirectional Publish Initial Uncertain Value

Fallback Delay: 2.5

Configure Transform(s)

Script

```

self: A reference to the component this binding is co
value: The incoming value from the binding or the pre
quality: The quality code of the incoming value.
timestamp: The timestamp of the incoming value as a j
...
2
3 output_json = []
4 style_orange = {"backgroundColor": "#F7901D"}
5 style_green = {"backgroundColor": "#00AA00"}
6
7

```

Binding Preview

Tag: Dataset[3 rows, 2 cols] → Script: [{style: {backgroundColor: ...}}

OK Cancel Apply

This would result in coloring rows where the "B" column has distinct values of "One" or "Two".

Displaying a Subview in a Table Row

In a Perspective Table component, you have the option to enable subviews. When a subview is set up, you can click on the Expand icon and have another view display without closing the first view. This example sets up a table with several city statistics and when the Expand icon is selected for a city, a Map component will be displayed showing the location of the city on the map.

This example focuses on using a Map component in a subview on the table, but the larger implication here is that subviews in table rows can receive values from each row and utilize them with property bindings, allowing each subview to contain data unique to the row. The image below shows what our finished view will look like.

city	country	population	lat	lng
▶ Folsom	United States	77,271	38.68	-121.18
▶ Jakarta	Indonesia	10,187,595	-6.21	106.85
▼ Madrid	Spain	3,233,527	40.41	-3.70
				
▶ Prague	Czech Republic	1,241,664	50.07	14.45
▶ San Diego	United States	1,406,630	32.71	-117.16
▶ San Francisco	United States	884,363	37.78	-122.42
▶ Santiago	Chile	5,415,000	-33.33	-70.66

On this page ...

- [Summary of Subviews](#)
- [Create a View for the Table Data](#)
- [Create a View for Displaying the Map](#)
- [Use the Maps View as a Subview for the Table](#)

Summary of Subviews

When subviews are enabled for each row in a table, the row will implicitly pass some parameters over to the subview. The passed parameters are:

- row - a number representing the row
- rowIndex - a number representing the index of the current row
- value - JSON Object representing the contents of the row. The value of each column in the row will be a value under this object.

Thus, in the image above where our table has a **country** column, the subview will receive the value of country at `params.value.country`. So you can create a component binding on the subview's configuration that references a value at `params.value.country`.

Create a View for the Table Data

We'll start by creating a view that will contain the table.

1. Right click on Views to create a view. In the example, we named the new view **CityStats**, set it as a **Coordinate** Root Container Type, and checked the Page URL option. Pages can be added later as well.

New View
✕

Name

 ✔

Root Container Type

Coordinate
▼

Page URL

 ✔

Cancel

Create View

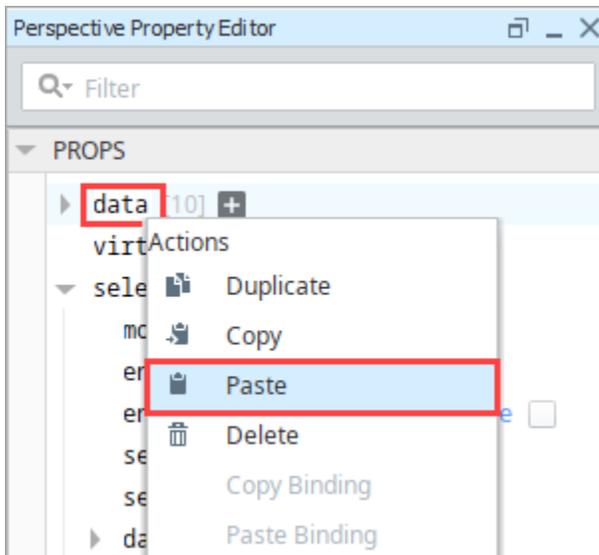
2. Drag a Table component onto the view.
3. The table needs to have Latitude and Longitude data for the map to show that location. Highlight and copy the following data:

```
[
  {
    "city": "Folsom",
    "country": "United States",
    "population": 77271,
    "lat": 38.678287,
    "lng": -121.177318
  },
  {
    "city": "Jakarta",
    "country": "Indonesia",
    "population": 10187595,
    "lat": -6.208404,
    "lng": 106.849087
  },
  {
    "city": "Madrid",
    "country": "Spain",
    "population": 3233527,
    "lat": 40.41498,
    "lng": -3.702002
  },
  {
    "city": "Prague",
    "country": "Czech Republic",
    "population": 1241664,
    "lat": 50.073453,
    "lng": 14.450091
  },
  {
    "city": "San Diego",
    "country": "United States",
    "population": 1406630,
    "lat": 32.713832,
    "lng": -117.158616
  },
  {
    "city": "San Francisco",
    "country": "United States",
    "population": 884363,
    "lat": 37.776379,
    "lng": -122.423501
  },
  {
    "city": "Shanghai",
    "country": "China",

```

```
"population": 24153000,
"lat": 31.227167,
"lng": 121.498839
},
{
  "city": "Tokyo",
  "country": "Japan",
  "population": 13617000,
  "lat": 35.69042,
  "lng": 139.746457
},
{
  "city": "Washington, DC",
  "country": "United States",
  "population": 658893,
  "lat": 38.90598,
  "lng": -77.04882
},
{
  "city": "Wellington",
  "country": "New Zealand",
  "population": 405000,
  "lat": -41.284336,
  "lng": 174.770488
}
]
```

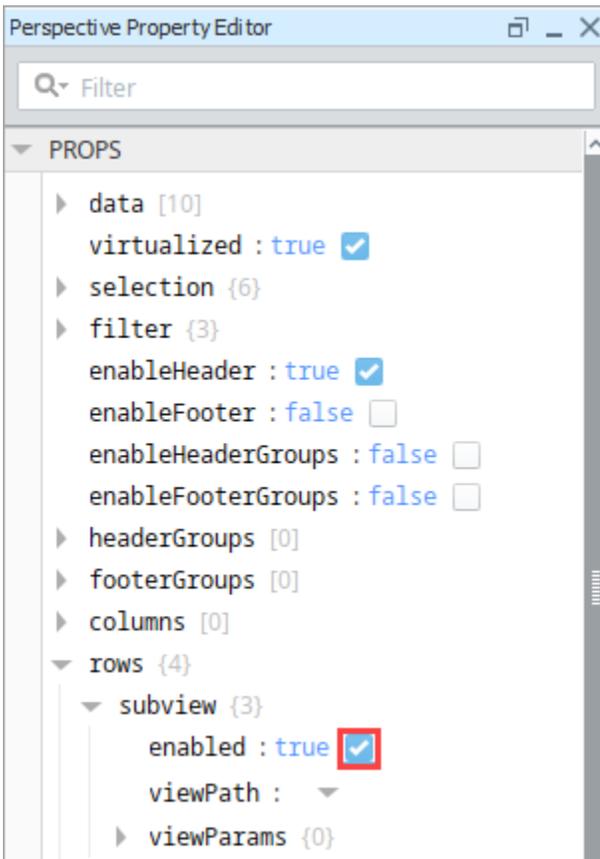
4. Right click on the data property of the Table component and select **Paste**.



5. Your table will now display the data for 10 rows and 5 columns.

city	country	population	lat	lng
Folsom	United States	77,271	38.68	-121.18
Jakarta	Indonesia	10,187,595	-6.21	106.85
Madrid	Spain	3,233,527	40.41	-3.7
Prague	Czech Republic	1,241,664	50.07	14.45
San Diego	United States	1,406,630	32.71	-117.16
San Francisco	United States	884,363	37.78	-122.42
Shanghai	China	24,153,000	31.23	121.5
Tokyo	Japan	13,617,000	35.69	139.75
Washington, DC	United States	658,893	38.91	-77.05
Wellington	New Zealand	405,000	-41.28	174.77

6. Next, enable the **subview** property under props.rows.



7. The table now has **Expand** ▶ icons for each row.

	city	country	population	lat	lng
▶	Folsom	United States	77,271	38.68	-121.18
▶	Jakarta	Indonesia	10,187,595	-6.21	106.85
▶	Madrid	Spain	3,233,527	40.41	-3.70
▶	Prague	Czech Republic	1,241,664	50.07	14.45
▶	San Diego	United States	1,406,630	37.71	-117.14

Create a View for Displaying the Map

Next we'll make the view that will be display a map of the cities in our table.

1. In the Project Browser, right click on Views to create a view. Name the new view **CityMaps**. Set it as a **Flex** layout, so the map easily takes up all available space. Lastly, do not check the Page URL option, as we don't need a corresponding page.

New View ×

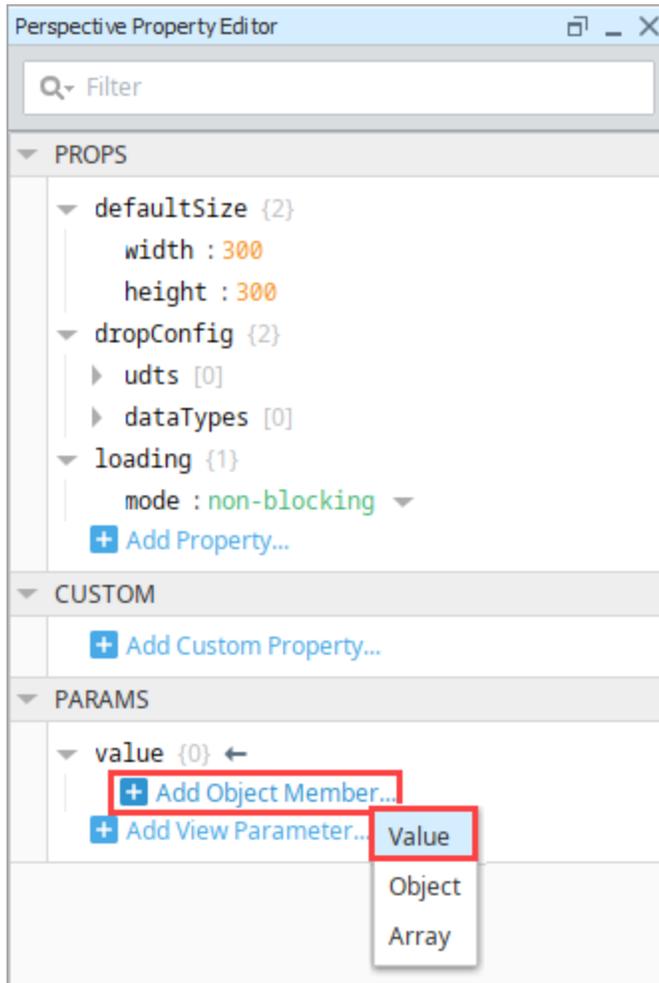
Name
CityMaps ✓

Root Container Type
Flex

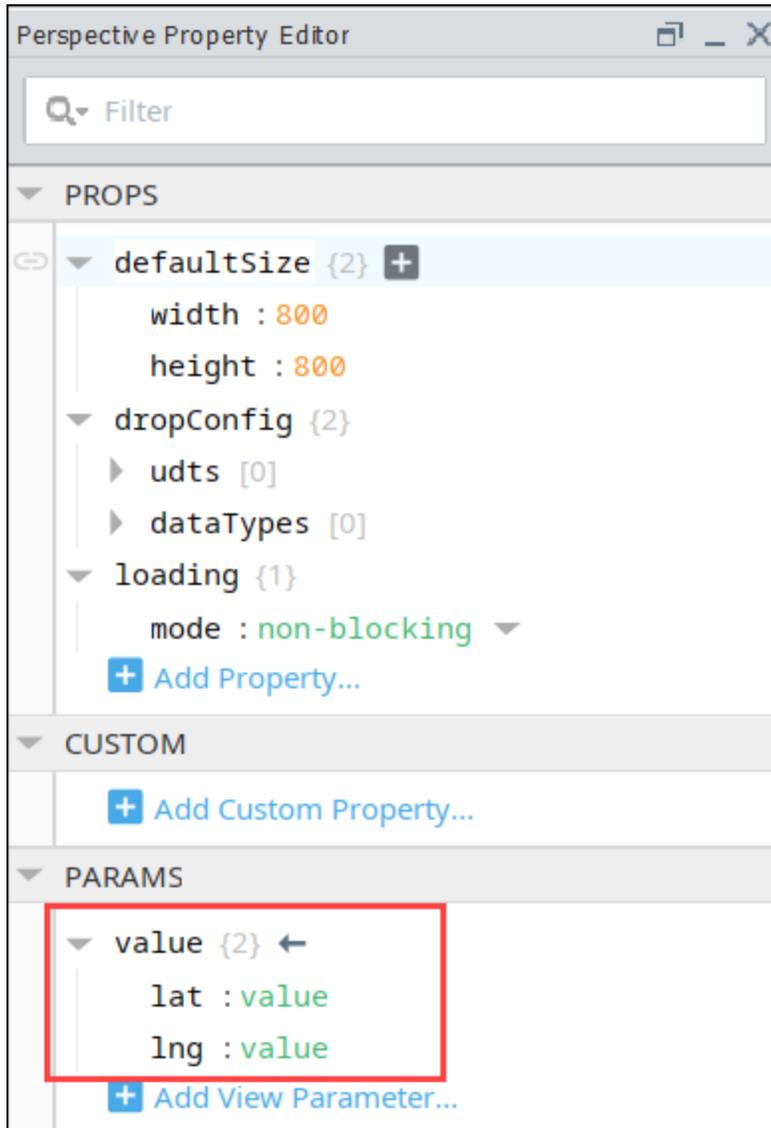
Page URL
/citymaps ✓

Cancel Create View

2. Drag a Map component onto the **CityMaps** View.
3. Set the Map's **Position.Grow** property to "1" so it resizes to take up the entire view.
4. Click on the **CityMaps** view in the Project Browser. In the Property Editor, click the **Add View Parameter** link under **Params** and choose **Object**.
5. Double click on key, and enter **value** as the object name. Note that the object **must** be named "value".
6. Next, we'll add two parameters to that value object.
 - a. Click **Add Object Member** link under Params and choose **Value**.



- b. Double click on key, and enter "**lat**". This matches the lat (latitude) column from the Table on the CityStats view. This name must *exactly* match the column name in the table.
- c. Click the **Add Object Member** icon next to value and choose **Value**.
- d. Double click on key, and enter "**lng**". This matches the lng (longitude) column from the Table on the CityStats view. This name must *exactly* match the column name in the table.



Note: It's important that the object here is named "value". The perspective table we're going to embed the map into has a feature where content from each row in the table can be passed to its subview. This mechanism requires that the subview contain an object named "value". In So if you wanted the subview to be passed any of the values from it's parent row, simply make a "value" object, and add values to it where the keys match the name of the column on the table.

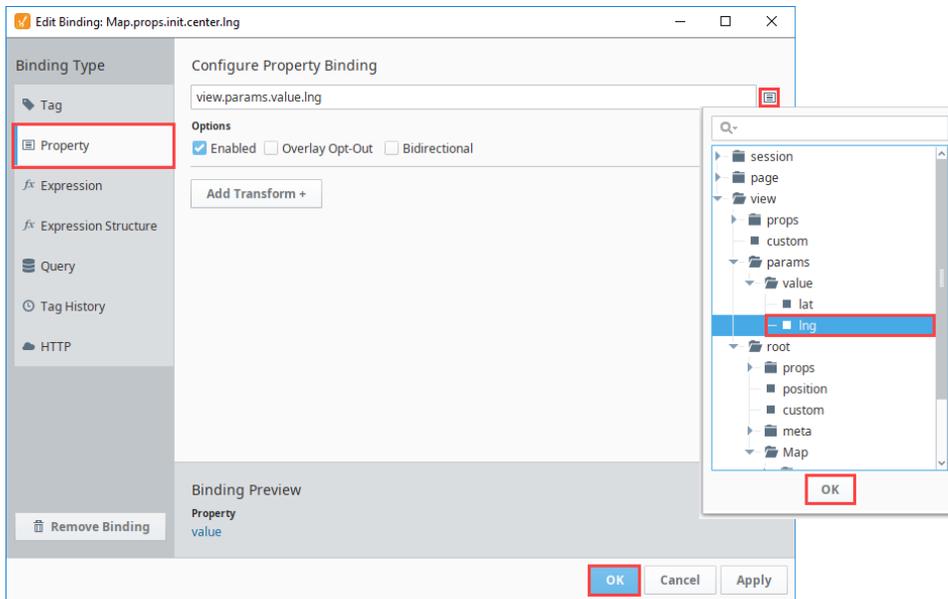
7. Next select the Map component. We need to set the map's initial geographic center to the view parameters. In the Property Editor, expand the **init.center** property.

- Click on the **Binding**  icon next to the **lat** property.
- On the Edit Binding screen, select **Property** as the binding type.
- Click the **Browse Properties**  icon. Navigate to view, the params, the value, and then the **lat** property.
- Click **OK**, then click **OK** again to save the binding.

Note: At this point, the `init.center.lat` property is bound to `view.params.value.lat` where `view.params.value.lat`'s value is "value" instead of a valid latitude number. This will cause a Component Error which is expected.

- Click on the **binding**  icon next to the **lng** property.
- On the Edit Binding screen, select **Property** as the binding type.
- Click the **Browse Properties**  icon. Navigate to view, the params, the value, and then the **lng** property.
- Click **OK**, then click **OK** again to save the binding.

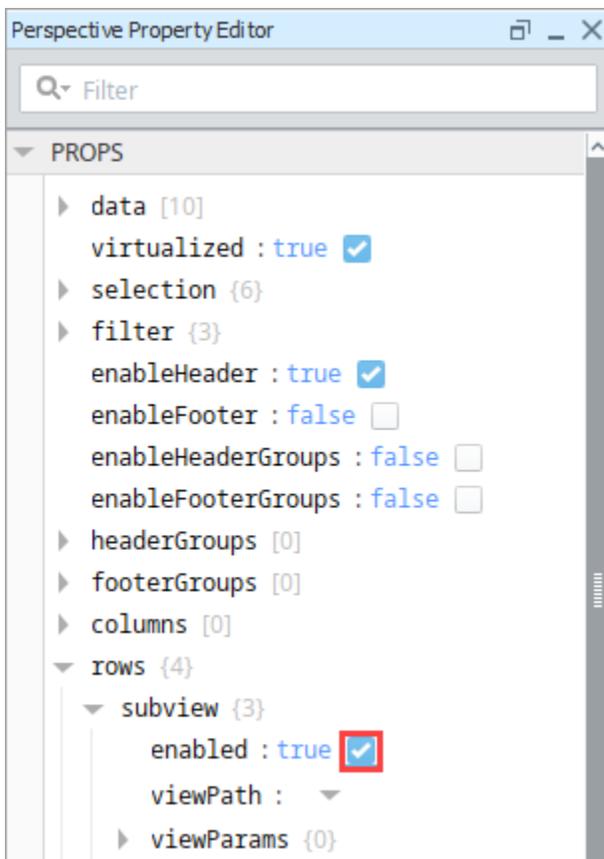
Note: At this point, the `init.center.lat` property is bound to `view.params.value.lat` where `view.params.value.lat`'s value is "value" instead of a valid latitude number. This will cause a Component Error which is expected.



Use the Maps View as a Subview for the Table

Lastly, we need to tell the CityStats View to use CityMaps as its subview.

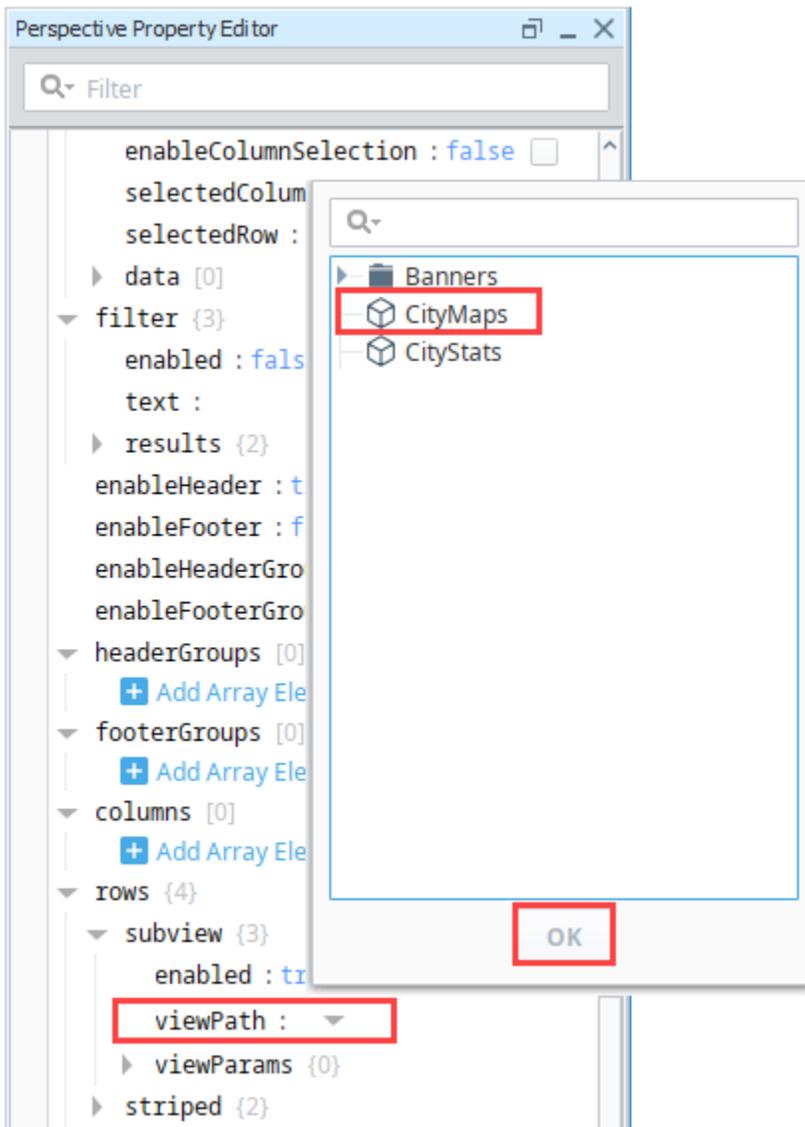
1. On the CityStats View, select the **Table** component.
2. In the Property Editor, scroll down to the **rows.subview.enabled** property.
3. Next, enable the "enabled" property.



As a result, you'll notice the table now has **Expand** ▶ icons for each row.

	city	country	population	lat	lng
▶	Folsom	United States	77,271	38.68	-121.18
▶	Jakarta	Indonesia	10,187,595	-6.21	106.85
▶	Madrid	Spain	3,233,527	40.41	-3.70
▶	Prague	Czech Republic	1,241,664	50.07	14.45
▶	San Diego	United States	1,406,630	37.71	-117.14

4. Find `rows.subview.viewPath`, and click the dropdown to see the a list of possible views. Choose **CityMaps** from the list and click **OK**.



5. Save your project.
6. Put the Designer into **Preview** mode. Click on the **Expand** ▶ icon next to one of the cities. You'll see a map of the city appear underneath the table row for that city. To close the map, click the **Collapse** ▲ icon.

city	country	population	lat	lng
▶ Folsom	United States	77,271	38.68	-121.18
▶ Jakarta	Indonesia	10,187,595	-6.21	106.85
▼ Madrid	Spain	3,233,527	40.41	-3.70

Leaflet | Map data © OpenStreetMap contributors, CC-BY-SA

▶ Prague	Czech Republic	1,241,664	50.07	14.45
▶ San Diego	United States	1,406,630	37.71	-117.14
▶ San Francisco	United States	884,363	37.78	-122.42
▶ Shanghai	China	24,153,000	31.23	121.50
▶ Tokyo	Japan	13,617,000	35.69	139.75
▼ Washington, DC	United States	658,893	38.91	-77.05

1

Perspective - Table Scripting

This page details the various scripting, component, and extension functions available for [Perspective's Table](#) component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

onEditCellCommit

This onEditCellCommit event is used with a runAction script on a table to take user entry and store it in the table or a database.

Provides a chance do something once a user has typed something into a cell. The user must commit the new value before the event will trigger. "Committing" a value depends on the type of value and how it's rendered. Numerical and text values can be committed by pressing "Enter" after typing a new value. Boolean values are typically committed via a click (such as in cases when the cell is rendered as a checkbox or toggle switches).

Additionally, the cell must first be editable (*props.data.[rowNumber].[columnName].editable* is set to true). The first cell in the default dataset on a newly created instance of the component demonstrates where the *editable* property must be positioned.

Note: This component event is designed to be used in tandem with a script runAction. Within the script action, special properties and methods are available on the **event** object, which is passed to the script action as a parameter.

event.column

- Object Path

event.column

- Type

String

- Description

The name of the column under which the cell was edited.

event.row

- Object Path

event.row

- Type

Number

- Description

The unique row index as it is represented in the source data. Also known as the row ID.

event.value

- Object Path

event.value

- Type

On this page ...

- [Component Events](#)
 - [onEditCellCommit](#)
 - [onSelectionChange](#)
 - [onEditCellStart](#)
 - [onEditCellCancel](#)
 - [onRowClick](#)
 - [onRowDoubleClick](#)
 - [onSubviewExpand](#)
 - [onSubviewCollapse](#)
- [Component Functions](#)
 - [.collapseSubviews\(\)](#)
 - [.expandSubviews\(\)](#)
- [Extension Functions](#)

Any

- Description

The value that was typed into the cell.

Example - Change the value in a cell

```
# This example will set the value of a cell, based on what the user typed into it.

# Get the value that was typed into the cell
valueToSet = event.value

# We need to set a value in a particular cell. The event object contains row and column properties
# that report the position of the cell that was edited.

# If the data property contains an array, you would use the line below
self.props.data[event.row][event.column] = valueToSet

# If the data property contains a dataset, then you would want to use the following line instead
#self.props.data = system.dataset.setValue(self.props.data, event.row, event.column, valueToSet)
```

onSelectionChange

This onSelectionChange event will trigger when the selection in the chart changes.

Note:

The onSelectionChange event will fire on startup or mount if props do not equal the table components default selection config.

This component event is designed to be used in tandem with a script runAction. Within the script action, special properties and methods are available on the **event** object, which is passed to the script action as a parameter.

event.selectedColumn

- Object Path

event.selectedColumn

- Type

String or null

- Description

The name of the column that the selected cell is located under.

event.selectedRow

- Object Path

event.selectedRow

- Type

Number or null

- Description

The unique row index as it is represented in the source data. Also known as the row ID.

event.data

- Object Path

event.data

- Type

Array

- Description

Represents the currently selected entries. The contents of the array is based on the `enabledRowSelection` and `enableColumnSelection` properties as represented on the table below. The actual resulting value may include additional values if the selection mode on the table is set to "single interval" or "multiple interval".

<code>enabledRowSelection</code>	<code>enableColumnSelection</code>	Resulting return type	Example Output
True	False	An array containing a number of JSON objects that each represent a single row. Each JSON object contains one key-value pair for each column on the table.	<pre>[{"city": "Folsom", "country": "United States", "population": 77271}]</pre>
False	True	An array of JSON objects, where each object represents a separate row in the selected column. Each object contains a single key-value pair, where the key is the column name and the value is the value of the cell.	<pre>[{"city": "Folsom"}, {"city": "Helsinki"}, {"city": "Jakarta"}]</pre>
True	True	An array containing a single JSON object, which can be treated like a Python dictionary.	<pre>[{"city": "Folsom"}]</pre>

onEditCellStart

This `onEditCellStart` event fires when the user starts editing a cell. For `onEditCellStart`, the value is the initial value before any edits.

Note: This component event is designed to be used in tandem with a script `runAction`. Within the script action, special properties and methods are available on the **event** object, which is passed to the script action as a parameter.

event.column

- Object Path

`event.column`

- Type

[String or number](#)

- Description

The column the editing cell is positioned under.

event.row

- Object Path

`event.row`

- Type

[Number](#)

- Description

The unique row index as it is represented in the source data. Also known as the row ID.

event.rowIndex

- Object Path

`event.rowIndex`

- Type

[Number](#)

- Description

The row index as it is represented in the current visible data. Useful in cases where some of the rows are hidden, such as when filtering.

event.value

- Object Path

event.value

- Type

Any

- Description

The value of the cell before editing began.

onEditCellCancel

This onEditCellCancel event is fired when the user has canceled a cell edit and has exited editing mode by effectively pressing the escape key.

Note: This component event is designed to be used in tandem with a script runAction. Within the script action, special properties and methods are available on the **event** object, which is passed to the script action as a parameter.

event.column

- Object Path

event.column

- Type

String

- Description

The column name of the cell being edited.

event.row

- Object Path

event.row

- Type

Number

- Description

The unique row index as it is represented in the source data. Also known as the row ID.

event.rowIndex

- Object Path

event.rowIndex

- Type

Number

- Description

The row index as it is represented in the current visible data. Useful in cases where some of the rows are hidden, such as when filtering.

event.value

- Object Path

event.value

- Type

Any

- Description

The value of the cell before editing began.

onRowClick

This onRowClick event is fired when a row in the table is clicked.

Note: This component event is designed to be used in tandem with a script runAction. Within the script action, special properties and methods are available on the **event** object, which is passed to the script action as a parameter.

event.row

- Object Path

event.row

- Type

Number

- Description

The unique row index as it is represented in the source data. Also known as the row ID.

event.rowIndex

- Object Path

event.rowIndex

- Type

Number

- Description

The row index as it is represented in the current visible data. Useful in cases where some of the rows are hidden, such as when filtering.

event.value

- Object Path

event.value

- Type

PlainObject

- Description

The rows value as a JSON object.

onRowDoubleClick

This onRowDoubleClick event is triggered when a row in the table is double clicked.

Note: This component event is designed to be used in tandem with a script runAction. Within the script action, special properties and methods are available on the **event** object, which is passed to the script action as a parameter.

event.row

- Object Path

event.row

- Type

Number

- Description

The unique row index as it is represented in the source data. Also known as the row ID.

event.rowIndex

- Object Path

event.rowIndex

- Type

Number

- Description

The row index as it is represented in the current visible data. Useful in cases where some of the rows are hidden, such as when filtering.

event.value

- Object Path

event.value

- Type

PlainObject

- Description

The rows value as a JSON object.

onSubviewExpand

This onSubviewExpand event is triggered when a row subview is expanded.

Note: This component event is designed to be used in tandem with a script runAction. Within the script action, special properties and methods are available on the **event** object, which is passed to the script action as a parameter.

event.row

- Object Path

event.row

- Type

Number

- Description

The unique row index as it is represented in the source data. Also known as the row ID.

event.rowIndex

- Object Path

event.rowIndex

- Type

Number

- Description

The row index as it is represented in the current visible data. Useful in cases where some of the rows are hidden, such as when filtering.

event.value

- Object Path

event.value

- Type

PlainObject

- Description

The rows value as a JSON object.

onSubviewCollapse

This onSubviewCollapse event is triggered when a row subview is collapsed.

Note: This component event is designed to be used in tandem with a script runAction. Within the script action, special properties and methods are available on the **event** object, which is passed to the script action as a parameter.

event.row

- Object Path

event.row

- Type

Number

- Description

The unique row index as it is represented in the source data. Also known as the row ID.

event.rowIndex

- Object Path

event.rowIndex

- Type

Number

- Description

The row index as it is represented in the current visible data. Useful in cases where some of the rows are hidden, such as when filtering.

event.value

- Object Path

event.value

- Type

PlainObject

- Description

The rows value as a JSON object.

Component Functions

.collapseSubviews()

The following feature is new in Ignition version **8.1.17**
[Click here](#) to check out the other new features

- Description

This function will collapse the specified row subviews. If no parameter is specified, this function will collapse all expanded subviews on the current page.

- Parameters

array rows - An optional array of indices of rows to collapse. Any argument that is not a list will throw an exception. A list of invalid indices will not throw an exception. Omitting this parameter will collapse all subviews.

- Return

Nothing

Note: This function only operates on rows which are on the currently displayed page. For example, if you have a table that displays 25 rows per page and invoke `self.collapseSubviews([100])` on page one, nothing will happen.

This feature was changed in Ignition version 8.1.28:

Note:

Specifying a number of rows to collapse when using this function will now affect the actual number specified, instead of stopping at the end of the currently displayed page.

Additionally, there is a distinction between using `row` and `RowIndex`. `Row` refers to the true index of the row as it exists in the data, and is not affected by paging, sorting, or searching. `RowIndex` refers to the visual index of the row as it appears on the table and is affected by paging, sorting, and searching.

Example

```
# Collapse subviews for rows 1 and 3 if they exist. If the list does not match any indices that exist,
nothing will happen.
self.getSibling('Table').collapseSubviews([1, 3])

# Collapse all expanded subviews.
self.getSibling('Table').collapseSubviews()

# The following lines are invalid and will throw an exception:
self.getSibling('Table').collapseSubviews(None)
self.getSibling('Table').collapseSubviews(3)
```

.expandSubviews()

The following feature is new in Ignition version **8.1.17**
[Click here](#) to check out the other new features

- Description

This function will expand the specified row subviews. This will only expand rows that are visible on the current page.

- Parameters

array rows - An array of indices of rows to expand. Any argument that is not a list will throw an exception. A list of invalid indices will not throw an exception. Omitting this parameter will expand all subviews.

- Return

Nothing

Note: This function only operates on rows which are on the currently displayed page. For example, if you have a table that displays 25 rows per page and invoke `self.expandSubviews([100])` on page one, nothing will happen.

This feature was changed in Ignition version 8.1.28:

Note:

Specifying a number of rows to expand when using this function will now affect the actual number specified, instead of stopping at the end of the currently displayed page.

Additionally, there is a distinction between using `row` and `RowIndex`. `Row` refers to the true index of the row as it exists in the data, and is not affected by paging, sorting, or searching. `RowIndex` refers to the visual index of the row as it appears on the table and is affected by paging, sorting, and searching.

Example

```
# Expand subviews for rows 1 and 3 if they exist. If the list does not match any indices that exist, nothing
will happen.
self.getSibling('Table').expandSubviews([1, 3])

# Expand all subviews.
self.getSibling('Table').expandSubviews()

# The following lines are invalid and will throw an exception:
self.getSibling('Table').expandSubviews(None)
self.getSibling('Table').expandSubviews(3)
```

Extension Functions

This component does not have extension functions associated with it.

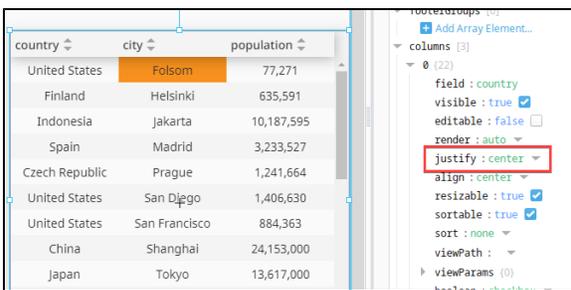
Table Column Configurations

This page contains examples that involve making changes to a table component's columns property.

Aligning Text in a Column

The table component can apply different text alignment for each column. In this example we will demonstrate the concept with the default dataset on the component.

1. Create a new table component, or select an existing one.
2. Add an element for each column you wish to display on the tag to the props.columns property by either clicking the **Add Array Element** button or the **+** icon right of the **columns** property. Repeat this process until you see all columns on the table again.
3. Map each props.column element to a column in the underlying data (props.data). For each **props.columns[#].field** property, set the field property to the name of a column/series in our props.data property in the desired column order:
 - props.columns[0].field to **country**
 - props.columns[1].field to **city**
 - props.columns[2].field to **population**
4. Change the alignment on each column by changing the **props.columns[#].justify** property. In our case, we set the justify property on all columns elements to **center** to center the text.



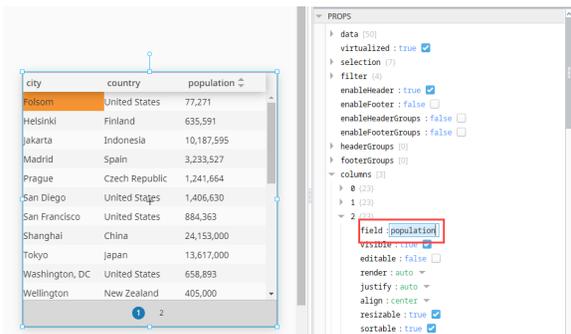
On this page ...

- [Aligning Text in a Column](#)
- [Changing Applied Formatting](#)
- [Replacing a Value in a Cell with a Progress Bar](#)
- [Embedding a View in a Table Cell](#)

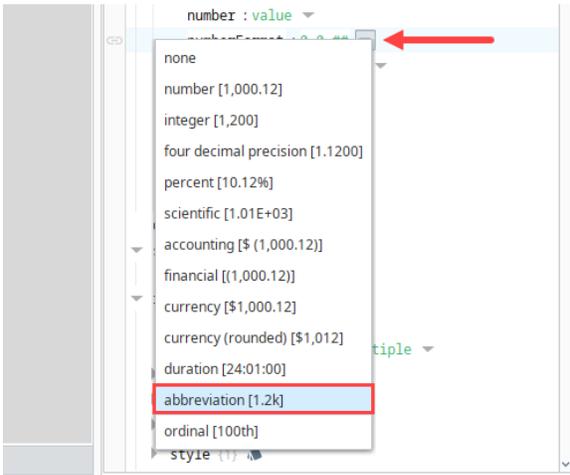
Changing Applied Formatting

In a Perspective Table, you can adjust your column format to display data as desired. In this example, instead of using the default **number [1,000.12]** format to display population counts, we will select **abbreviation [1.2k]** using the **numberFormat** dropdown.

1. Drag a Perspective Table component onto your view. We'll use the default population information.
2. Add three array elements to the columns property of your table.
3. Set the **field** property value for props.columns[2] to **population** to associate the column data in the table to the column configuration.



4. Select the **numberFormat** dropdown to view all format options.



5. Select **abbreviation [1.2k]** to adjust the displayed data in the table's population column.

city	country	population
Folsom	United States	77.3k
Helsinki	Finland	635.6k
Jakarta	Indonesia	10.2m
Madrid	Spain	3.2m
Prague	Czech Republic	1.2m
San Diego	United States	1.4m
San Francisco	United States	884.4k
Shanghai	China	24.2m
Tokyo	Japan	13.6m
Washington, DC	United States	658.9k
Wellington	New Zealand	405.0k

Replacing a Value in a Cell with a Progress Bar

Column configurations can be customized to display a progress bar on the table to show a column's value. We will use the default population information initially configured in the Table component to demonstrate this option.

1. In the Project Browser, right click on Views to create a view. In this example, the view will be named **table**. Set it to have a **Coordinate Root Container Type**.

✓ New View
✕

Name

Root Container Type

Coordinate

Page URL

Cancel

Create View

2. Drag a Table component onto your table view.
3. Add three array elements to the columns property of your table.
4. **Set the field** property values inside each of the three columns array elements to match each of the column names in your table to associate a column in the table and the custom column configurations.
 - props.columns[0].field to **country**
 - props.columns[1].field to **city**
 - props.columns[2].field to **population**
5. Access the props.columns[2] and set the **render** property to **number**.
6. Then, set the **number** property to **progress**.

```

columns [3]
  0 {22}
  1 {22}
  2 {22}
    field : population
    visible : true ✓
    editable : false
    render : number
    justify : auto
    align : center
    resizable : true ✓
    sortable : true ✓
    sort : none
    number : progress
  progressBar {5}
    max : 30,000,000
    min : 0
    bar {2}
    track {2}
    value {4}

```

7. Set the `progressBar.max` value to 30,000,000 to account for cities with a large population.

city	country	population
Folsom	United States	77,271
Helsinki	Finland	635,591
Jakarta	Indonesia	10,187,595
Madrid	Spain	3,233,527
Prague	Czech Republic	1,241,664
San Diego	United States	1,406,630
San Francisco	United States	884,363
Shanghai	China	24,153,000
Tokyo	Japan	13,617,000
Washington, DC	United States	658,893
Wellington	New Zealand	405,000
Delhi	India	11,034,555
Dhaka	Bangladesh	14,399,000
Lagos	Nigeria	16,060,303
Karachi	Pakistan	14,910,352
Istanbul	Turkey	14,025,000
Cairo	Egypt	10,230,350
Mexico City	Mexico	8,974,724
London	United Kingdom	8,825,001
New York City	United States	8,622,698
Tehran	Iran	8,154,051

Embedding a View in a Table Cell

In a Perspective Table, it is possible to embed a view inside a table cell. In this example, instead of using the table's built-in progress bar, we'll embed a view that contains a custom progress bar using the [Progress](#) component. The default information that comes on the factory configured table component will be used.

1. Create the table view first. In the Project Browser, right click on Views to create a view. In this example, name the view **table**. Set it to a **Coordinate** Root Container Type.

New View
✕

Name

 ✓

Root Container Type

Coordinate
▾

Page URL

 ✓

Cancel

Create View

2. Drag a [Perspective Table](#) onto the table view.
3. Add three array elements to the columns property of the table like as shown in Property Editor as shown in the image below.

city	country	population
▶ Tunis	Tunisia	1,056,247
▶ Yerevan	Armenia	1,060,138
▶ Prague	Czech Republic	1,241,664
▶ Dallas	United States	1,317,929
▶ Milan	Italy	1,359,905
▶ San Diego	United States	1,406,630
▶ Guadalajara	Mexico	1,495,189
▶ Montreal	Canada	1,649,519
▶ Manila	Philippines	1,780,148
▶ Shiraz	Iran	1,869,001
▶ Jakarta	Indonesia	10,187,595
▶ Cairo	Egypt	10,230,350
▶ Delhi	India	11,034,555
▶ Tokyo	Japan	13,617,000
▶ Istanbul	Turkey	14,025,000
▶ Dhaka	Banladesh	14,399,000

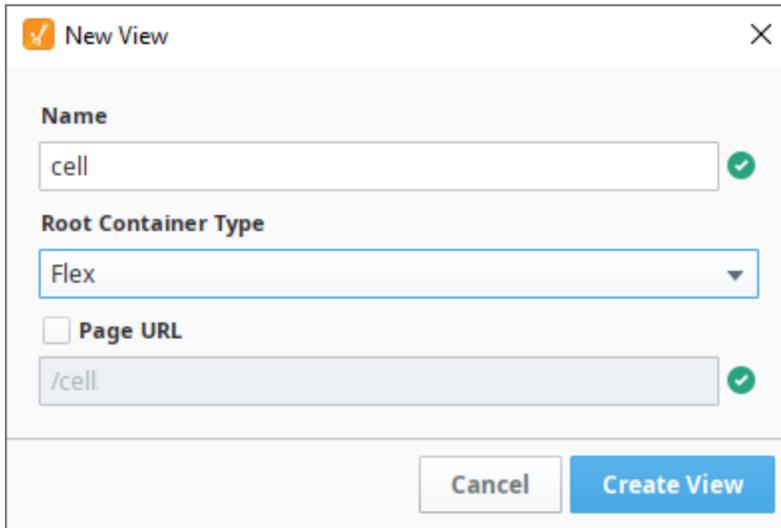
Perspective Property Editor

Filter

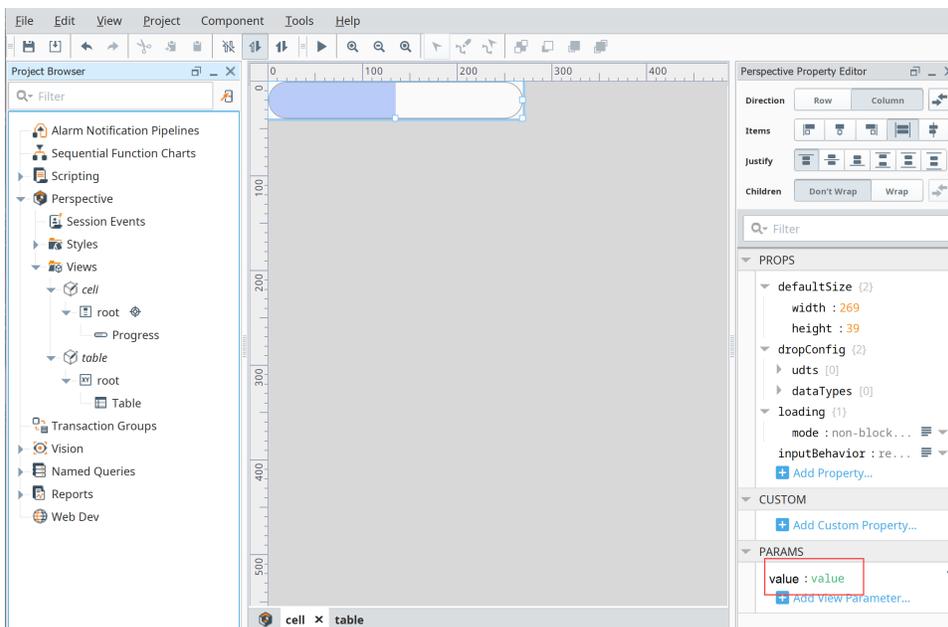
PROPS

- ▶ data [50]
- virtualized : true
- selection [6]
 - mode : single ▾
 - enableRowSelection : true
 - enableColumnSelection : false
 - selectedColumn : city
 - selectedRow : 16
- data [1]
- filter [3]
 - enabled : false
 - text :
 - results [2]
 - enableHeader : true
 - enableFooter : false
 - enableHeaderGroups : false
 - enableFooterGroups : false
- headerGroups [0]
 - Add Array Element...
- footerGroups [0]
 - Add Array Element...
- columns [3]**
 - ▶ 0 [22]
 - ▶ 1 [22]
 - ▶ 2 [22]
- sortOrder [0]
 - Add Array Element...
- rows [3]
 - subviewExpansionMode : single ▾

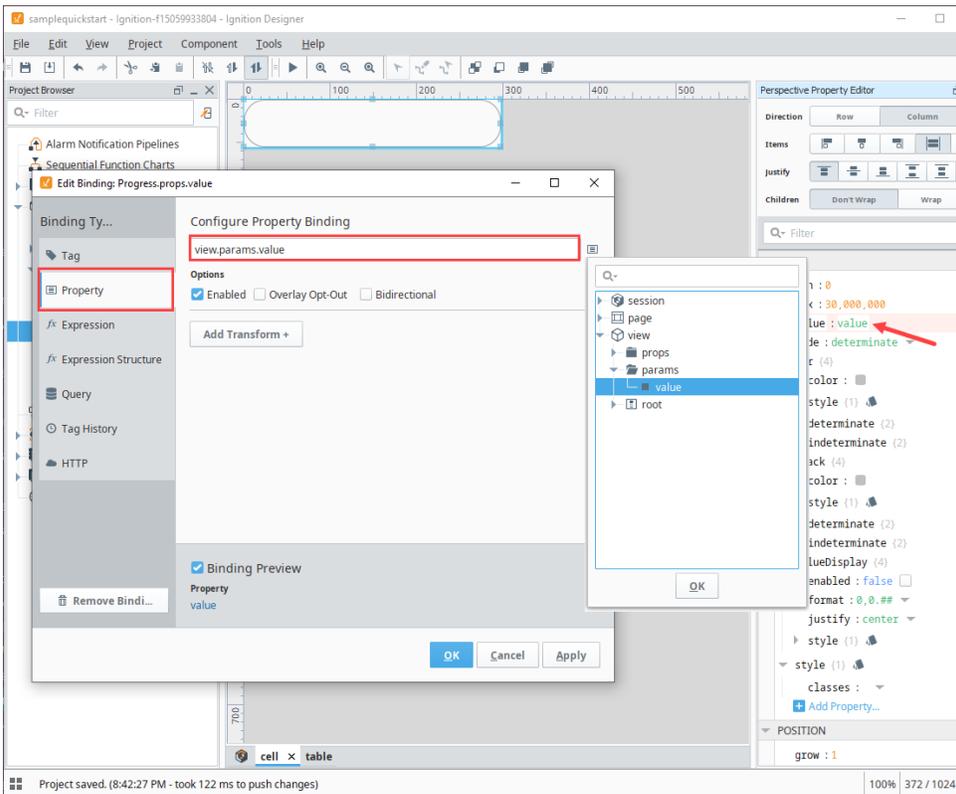
4. There is a **field** property inside each of the three column array elements. Set the field values to match each of the column names in your table
 - props.columns[0].field to **country**
 - props.columns[1].field to **city**
 - props.columns[2].field to **population**
5. Now, create the cell view. Right click on Views to create a view. In this example, the view will be named **cell**. Set it to a **Flex** Root Container Type.



6. Drag and drop a [Progress](#) component onto the cell view.
7. Select the Progress Bar component and set the **position.grow** property to 1 so that the bar takes up as much space in the container as possible.
8. In the Project Browser, select the **cell** view, and resize the view so it is closer in size to the population column in the table on the **table** view.
9. Click **Add View Parameter** while your view is selected in the Project Browser.
10. Select **Value** from the listed type options.
11. Enter **value** in place of **key**.



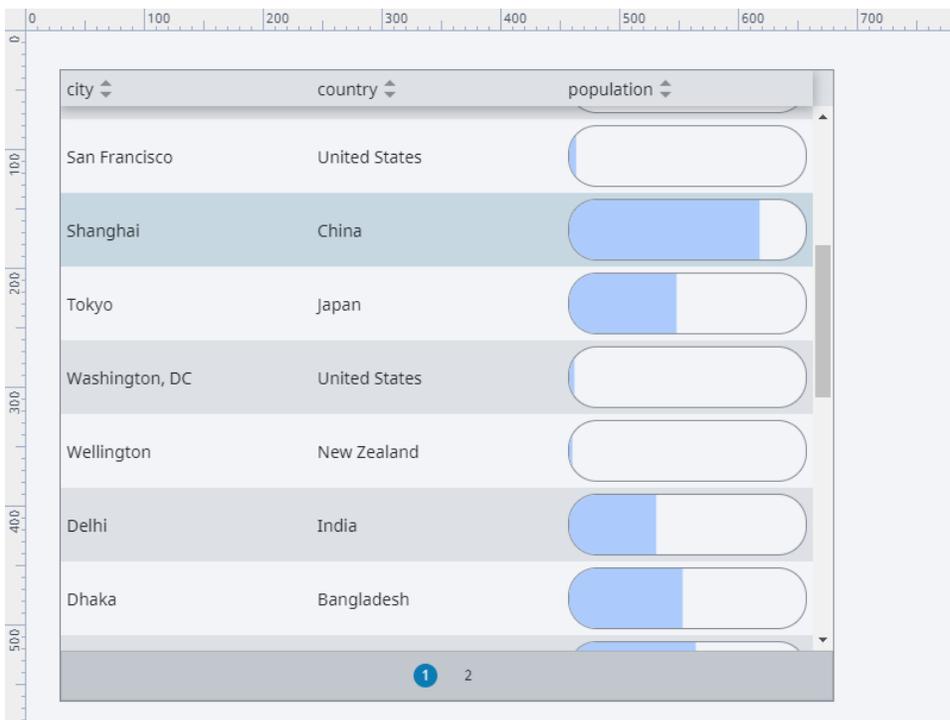
12. Set the **max** property value to 30,000,000 on the Progress component to account for large values.
13. Bind the Progress's **value** property to the view parameter created in Step 10 as shown below.



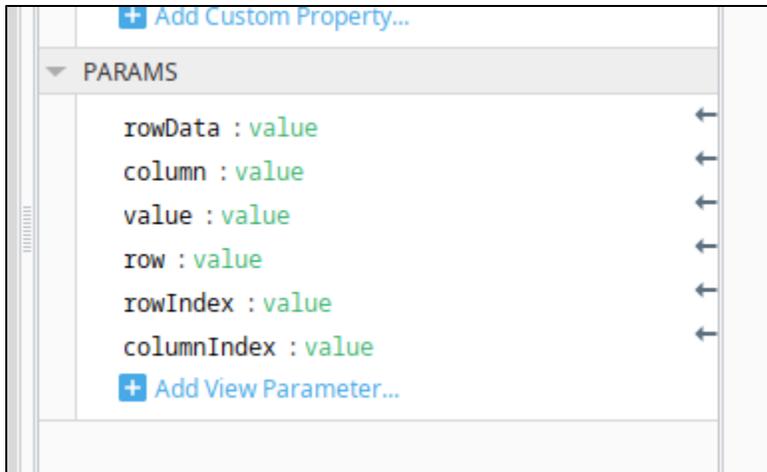
14. Go to the column array element with the **field** value of **population** and set the **render** value to **view**.
15. Set the **viewPath** to **cell** to embed **cell view** into the population cell values.



16. After the population column is pointed to the cell view, the population number from the table cell will be passed to the cell view. Since the cell view's Progress Bar has its value property bound to the cell view's input parameter, the population value will then be displayed on the table by the Progress Bar in the cell view. If you wanted to resize the progress bar, simply change the **height** and **width** properties under the **defaultSize** property on the **cell** view.



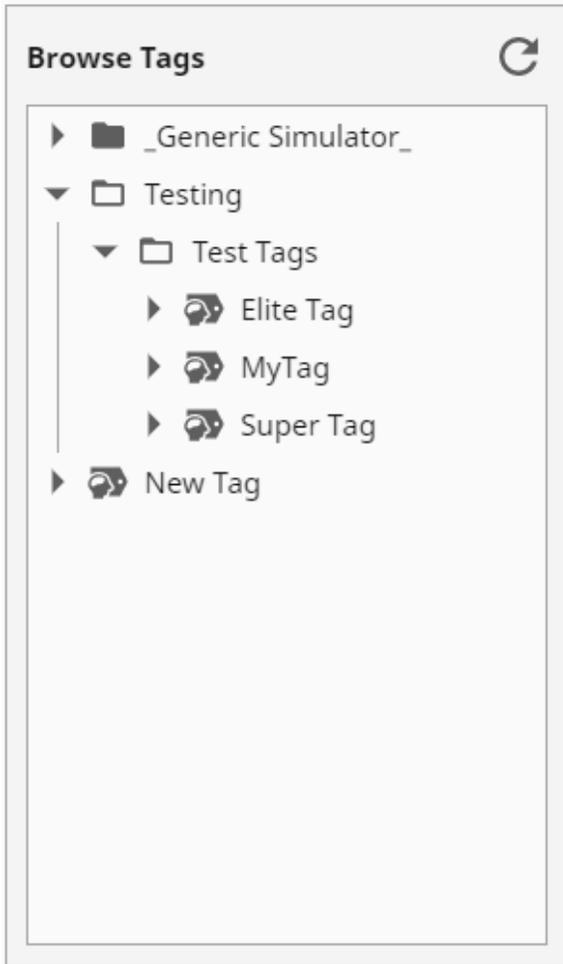
Once a column is configured to render each cell as a view, the table component will automatically pass each view a number of contextual view parameters. Components on the rendered view can access these property values by first configuring view params that match the passed keys.



Property Key	Description
column	A string representing the column name of the column, as defined by <code>props.columns.[columnNumber].field</code> .
columnIndex	A number representing the index of the current column. Similar to <code>columnNumber</code> in <code>props.columns.[columnNumber]</code> .
row	A number representing the index of the row. Note that this property does not adjust for hidden rows, or rows that are otherwise not displayed.
rowData	A JSON object representing table data for the current row. This is equivalent to accessing <code>props.data[row]</code> on the table.
rowIndex	A number representing the displayed index of the row. This property differs from <code>row</code> in that <code>rowIndex</code> does not count rows that aren't displayed.

	For example, assuming a table is only showing 25 rows per page, then the first row on the second page of results would display a rowIndex of 0, while row would provide a value of 25.
value	Represents the value of the cell being rendered as a view.

Perspective - Tag Browse Tree



On this page ...

- [Properties](#)
- [Scripting](#)
- [Example](#)

Component Palette Icon:



The following feature is new in Ignition version **8.1.16**
[Click here](#) to check out the other new features

The Tag Browse Tree component displays a tree hierarchy based on an array of objects. Icons can be chosen for the nodes of the tree, and different icons can be used when an node is expanded or collapsed.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

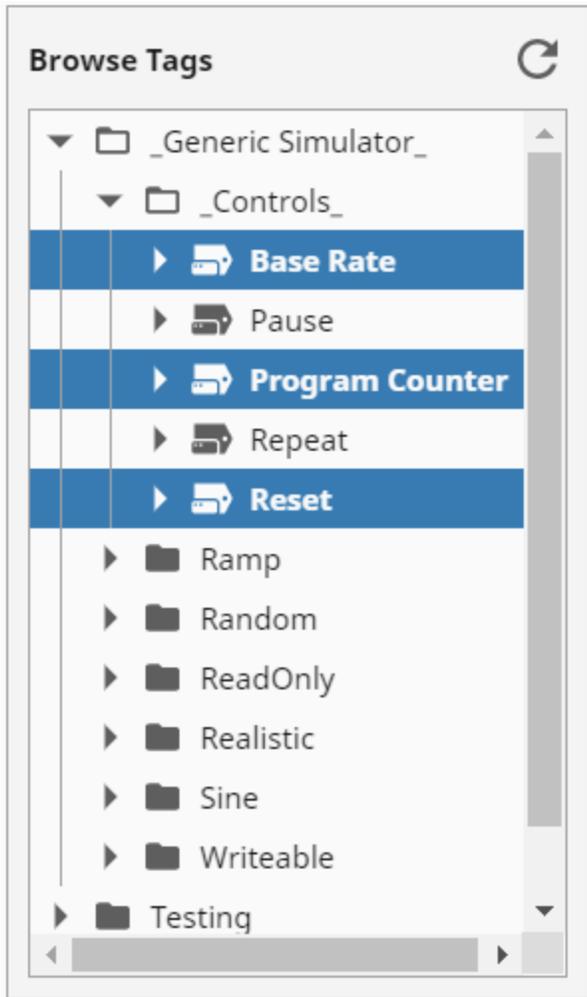
Name	Description	Property Type
root	Configuration for the path from which the displaying folder/Tag structure will start.	object

	Name	Description	Property Type									
	path	String value representing the "starting path" from which the Tag structure will begin displaying.	string									
filter	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.32 Click here to check out the other new features</p> </div> <p>Tree filtering configuration.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enables the visibility of the filter.</td> <td>boolean</td> </tr> <tr> <td>text</td> <td>The filter text.</td> <td>string</td> </tr> </tbody> </table>		Name	Description	Property Type	enabled	Enables the visibility of the filter.	boolean	text	The filter text.	string	object
Name	Description	Property Type										
enabled	Enables the visibility of the filter.	boolean										
text	The filter text.	string										
selection	Configuration for the selected Tag.		object									
	Name	Description	Property Type									
	mode	Mode used when selecting Tags. Choose between single to limit selection to only one Tag, or multiple to allow selection of multiple Tags at once.	string									
	values	List of the selected Tag paths in the order in which selection occurred.	array									
display	Display settings for the component.		object									
	Name	Description	Property Type									
	refreshIcon	Display settings for the refresh icon.	object									
	Name	Description		Property Type								
	visible	Visibility setting for the refresh icon.		boolean								
	path	Path to the icon used to represent the "refresh" action.		string								
	style	Sets a style for the refresh icon. Full menu of style options is available. You can also specify a style class .		object								
style	Sets a style for this component. Full menu of style options is available. You can also specify a style class .		object									

Scripting

See the [Perspective - Tag Browse Tree Scripting page](#) for the full list of scripting functions available for this component.

Example



Property	Value
selection.mode	multiple
selection.values.0	_Generic Simulator./_Controls_/Base Rate
selection.values.1	_Generic Simulator./_Controls_/Program Counter
selection.values.2	_Generic Simulator./_Controls_/Reset

Perspective - Tag Browse Tree Scripting

This page details the various scripting, component, and extension functions available for [Perspective's Tag Browse Tree](#) component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

On this page ...

- [Component Events](#)
 - [onNodeClick](#)
 - [onNodeDoubleClick](#)
 - [onNodeContextMenu](#)
- [Component Functions](#)
- [Extension Functions](#)
 - [filterBrowseNode](#)

onNodeClick

Fires whenever a node is clicked.

Object Path	Type	Description
event.name	String	The name of the node that was clicked.
event.path	String	The Tag path of the node that was clicked.

onNodeDoubleClick

Fires whenever a node is double-clicked.

Object Path	Type	Description
event.name	String	The name of the node that was clicked.
event.path	String	The Tag path of the node that was clicked.

onNodeContextMenu

Fires whenever a node is *right-clicked*.

Object Path	Type	Description
event.name	String	The name of the node that was clicked.
event.path	String	The Tag path of the node that was clicked.

Component Functions

This component does not have component functions associated with it.

Extension Functions

filterBrowseNode

- Description

Called for each Tag before it is displayed in the Tag Browse Tree. Provides an opportunity to create a complex filter for the Tags in the Tag Browse Tree. *filterBrowseNode* is best used alongside the Tag Browse Tree's `root.path` property to specify where the filter should begin filtering. Return `False` to exclude the Tag from displayed results.

- Parameters

[ComponentModelScriptWrapper.SafetyWrapper](#) self - A reference to the component that is invoking this function.

[NodeBrowseInfo](#) node - The Tag returned as type `NodeBrowseInfo`. See the [Ignition JavaDocs](#) for usage.

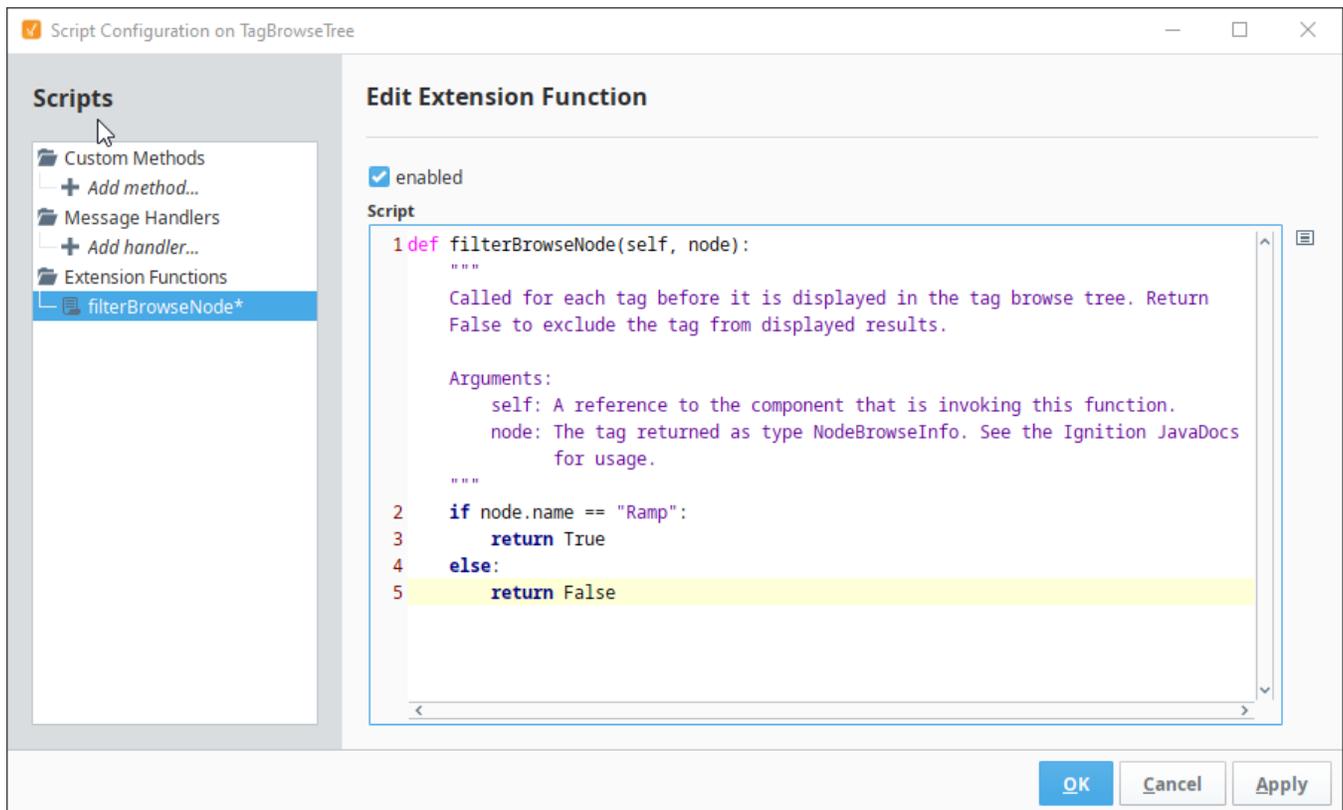
- Return

Boolean - The function must return either a True or False.

Example

filterBrowseNode Example

```
# This example will filter out any nodes (both Tags and folders included) that do not match the string Ramp.
if node.name == "Ramp":
    return True
else:
    return False
```



Script Configuration on TagBrowseTree

Scripts

- Custom Methods
 - + Add method...
- Message Handlers
 - + Add handler...
- Extension Functions
 - filterBrowseNode*

Edit Extension Function

enabled

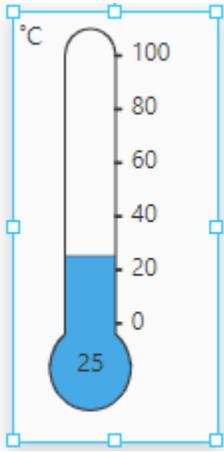
Script

```
1 def filterBrowseNode(self, node):
    """
    Called for each tag before it is displayed in the tag browse tree. Return
    False to exclude the tag from displayed results.

    Arguments:
        self: A reference to the component that is invoking this function.
        node: The tag returned as type NodeBrowseInfo. See the Ignition JavaDocs
            for usage.
    """
2   if node.name == "Ramp":
3       return True
4   else:
5       return False
```

OK Cancel Apply

Perspective - Thermometer



Component Palette Icon:



The Thermometer component displays a temperature value depicted as a level in a mercury thermometer. Temperature intervals can be defined with their own colors so that the mercury color changes based on the temperature range. Full menu of [style options](#) is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a [style class](#).

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
thermometerColor	Color of the outline of the thermometer. Default is black.	color
mercuryColor	Color of the mercury. See Color Selector .	color
axisLabelColor	The color of the thermometer's y-axis label. Default is black. See Color Selector .	color
strokeWidth	Width of the lines used to draw the thermometer in pixels.	value: numeric
highBound	The high boundary value for the whole thermometer.	value: numeric
lowBound	The lower boundary value for the whole thermometer.	value: numeric
value	The value to display in the thermometer. The mercury level and value label will change to reflect this.	value: numeric
unit	A string to describe the units for the current value label. Options are "F" for Fahrenheit or "C" for Celsius.	value: string dropdown
valueFontColor	The color of the current value. See Color Selector .	color
valueFont	The font to use for the current value label.	object

Name	Description	Property Type
------	-------------	---------------

On this page ...

- [Properties](#)
- [Component Events](#)
- [Example](#)

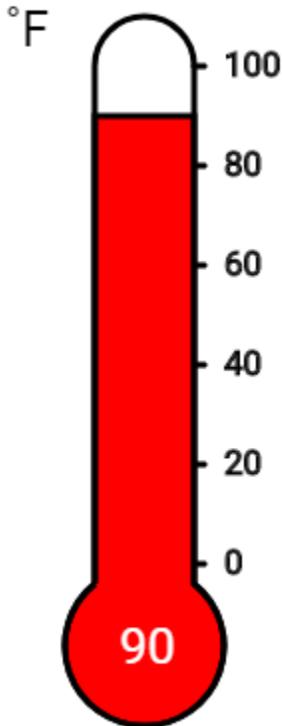
	<table border="1"> <tr> <td>fontSize</td> <td>Size of the font for the current value.</td> <td>value: numeric</td> </tr> </table>	fontSize	Size of the font for the current value.	value: numeric										
fontSize	Size of the font for the current value.	value: numeric												
intervals	<p>Defines the upper and lower temperature range for each interval.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color of the mercury in the thermometer. See Color Selector.</td> <td>color</td> </tr> <tr> <td>high</td> <td>High bound value for the interval.</td> <td>value: numeric</td> </tr> <tr> <td>low</td> <td>Low bound value for the interval.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color of the mercury in the thermometer. See Color Selector .	color	high	High bound value for the interval.	value: numeric	low	Low bound value for the interval.	value: numeric	object
Name	Description	Property Type												
color	Color of the mercury in the thermometer. See Color Selector .	color												
high	High bound value for the interval.	value: numeric												
low	Low bound value for the interval.	value: numeric												
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object												

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

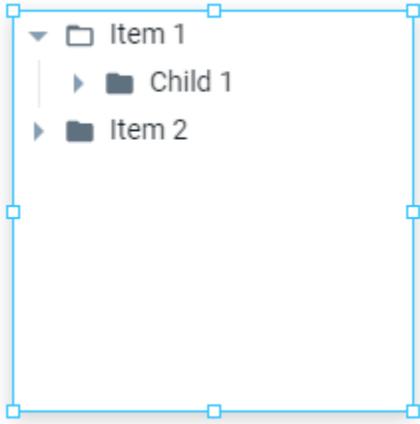
Example



Property	Value
props.unit	F
props.mercuryColor	#8AFF8A
props.intervals.1.high	45

props.intervals.0.color	#0000FF
props.intervals.1.color	#CCCCFF
props.intervals.2.color	#FF0000
props.intervals.2.low	85
props.value	90

Perspective - Tree



On this page ...

- [Properties](#)
- [Scripting](#)
- [Example](#)

Component Palette Icon:



The Tree component displays a tree hierarchy based on an array of objects. Icons can be chosen for the nodes of the tree, and different icons can be used when an node is expanded or collapsed.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type															
items	An array of objects, each of which represents a node on the tree. <table border="1" data-bbox="253 1247 1177 1482"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>label</td> <td>Label text for the list item.</td> <td>value: string</td> </tr> <tr> <td>expanded</td> <td>Whether or not the tree appears with all levels expanded.</td> <td>value: boolean</td> </tr> <tr> <td>data</td> <td>String data for list item.</td> <td>value: string</td> </tr> <tr> <td>items</td> <td>An array of objects, each of which represents a child node on the tree.</td> <td>array</td> </tr> </tbody> </table>	Name	Description	Property Type	label	Label text for the list item.	value: string	expanded	Whether or not the tree appears with all levels expanded.	value: boolean	data	String data for list item.	value: string	items	An array of objects, each of which represents a child node on the tree.	array	array
Name	Description	Property Type															
label	Label text for the list item.	value: string															
expanded	Whether or not the tree appears with all levels expanded.	value: boolean															
data	String data for list item.	value: string															
items	An array of objects, each of which represents a child node on the tree.	array															
interactable	If set to false, the tree is displayed but the user can't interact with it in the runtime. Default is true.	value: boolean															
selection	Holds the item index path of the current selection.	value: string															
selectionData	Array of objects containing the data and index path for all currently selected nodes. <table border="1" data-bbox="253 1682 1026 1827"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>itemPath</td> <td>Index path.</td> <td>value: numeric</td> </tr> <tr> <td>value</td> <td>The value of the 'data' property for the selected node.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	itemPath	Index path.	value: numeric	value	The value of the 'data' property for the selected node.	value: string	array						
Name	Description	Property Type															
itemPath	Index path.	value: numeric															
value	The value of the 'data' property for the selected node.	value: string															
appearance	Settings for the appearance of the tree. <table border="1" data-bbox="253 1892 1336 1965"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> </tbody> </table>	Name	Description	Property Type	object												
Name	Description	Property Type															

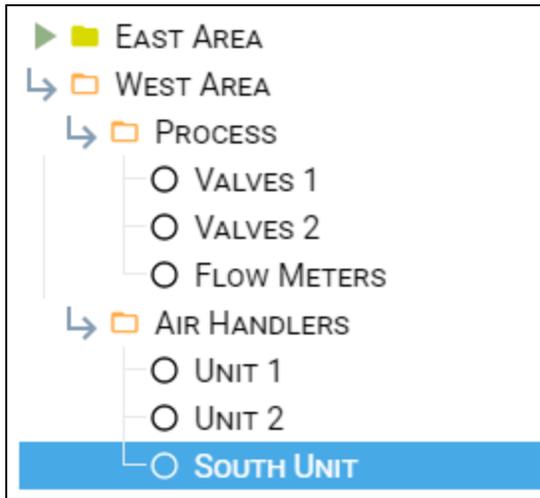
textOverflow	Setting indicating whether overflowing text should cause the entire tree to scroll horizontally or whether the text should be truncated with an ellipsis. Default is scroll.	value: string dropdown																																																
expandIcons	<p>Settings for the expand icons . Options as follows:</p> <table border="1" data-bbox="358 268 1193 1772"> <thead> <tr> <th data-bbox="358 268 456 344">Name</th> <th data-bbox="456 268 1065 344">Description</th> <th data-bbox="1065 268 1193 344">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="358 344 456 848">collapsed</td> <td data-bbox="456 344 1065 848"> <p>Icon appearance when path is collapsed.</p> <table border="1" data-bbox="466 401 1053 795"> <thead> <tr> <th data-bbox="466 401 563 476">Name</th> <th data-bbox="563 401 927 476">Description</th> <th data-bbox="927 401 1053 476">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="466 476 563 615">path</td> <td data-bbox="563 476 927 615">Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons.</td> <td data-bbox="927 476 1053 615">value: string</td> </tr> <tr> <td data-bbox="466 615 563 661">Color</td> <td data-bbox="563 615 927 661">Fill color to apply to the icon.</td> <td data-bbox="927 615 1053 661">string</td> </tr> <tr> <td data-bbox="466 661 563 795">style</td> <td data-bbox="563 661 927 795">Sets a style for the icon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td data-bbox="927 661 1053 795">object</td> </tr> </tbody> </table> </td> <td data-bbox="1065 344 1193 848">object</td> </tr> <tr> <td data-bbox="358 848 456 1310">expanded</td> <td data-bbox="456 848 1065 1310"> <p>Icon appearance when path is expanded.</p> <table border="1" data-bbox="466 905 1053 1255"> <thead> <tr> <th data-bbox="466 905 563 980">Name</th> <th data-bbox="563 905 927 980">Description</th> <th data-bbox="927 905 1053 980">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="466 980 563 1119">path</td> <td data-bbox="563 980 927 1119">Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons.</td> <td data-bbox="927 980 1053 1119">value: string</td> </tr> <tr> <td data-bbox="466 1119 563 1165">Color</td> <td data-bbox="563 1119 927 1165">Fill color to apply to the icon.</td> <td data-bbox="927 1119 1053 1165">string</td> </tr> <tr> <td data-bbox="466 1165 563 1255">style</td> <td data-bbox="563 1165 927 1255">Sets a style for the icon. Full menu of style options is available. You can also specify a style class.</td> <td data-bbox="927 1165 1053 1255">object</td> </tr> </tbody> </table> </td> <td data-bbox="1065 848 1193 1310">object</td> </tr> <tr> <td data-bbox="358 1310 456 1772">empty</td> <td data-bbox="456 1310 1065 1772"> <p>Icon appearance when path is empty.</p> <table border="1" data-bbox="466 1367 1053 1717"> <thead> <tr> <th data-bbox="466 1367 563 1442">Name</th> <th data-bbox="563 1367 927 1442">Description</th> <th data-bbox="927 1367 1053 1442">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="466 1442 563 1581">path</td> <td data-bbox="563 1442 927 1581">Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons.</td> <td data-bbox="927 1442 1053 1581">value: string</td> </tr> <tr> <td data-bbox="466 1581 563 1627">Color</td> <td data-bbox="563 1581 927 1627">Fill color to apply to the icon.</td> <td data-bbox="927 1581 1053 1627">string</td> </tr> <tr> <td data-bbox="466 1627 563 1717">style</td> <td data-bbox="563 1627 927 1717">Sets a style for the icon. Full menu of style options is available. You can also specify a style class.</td> <td data-bbox="927 1627 1053 1717">object</td> </tr> </tbody> </table> </td> <td data-bbox="1065 1310 1193 1772">object</td> </tr> </tbody> </table>	Name	Description	Property Type	collapsed	<p>Icon appearance when path is collapsed.</p> <table border="1" data-bbox="466 401 1053 795"> <thead> <tr> <th data-bbox="466 401 563 476">Name</th> <th data-bbox="563 401 927 476">Description</th> <th data-bbox="927 401 1053 476">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="466 476 563 615">path</td> <td data-bbox="563 476 927 615">Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons.</td> <td data-bbox="927 476 1053 615">value: string</td> </tr> <tr> <td data-bbox="466 615 563 661">Color</td> <td data-bbox="563 615 927 661">Fill color to apply to the icon.</td> <td data-bbox="927 615 1053 661">string</td> </tr> <tr> <td data-bbox="466 661 563 795">style</td> <td data-bbox="563 661 927 795">Sets a style for the icon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td data-bbox="927 661 1053 795">object</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string	Color	Fill color to apply to the icon.	string	style	Sets a style for the icon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	object	expanded	<p>Icon appearance when path is expanded.</p> <table border="1" data-bbox="466 905 1053 1255"> <thead> <tr> <th data-bbox="466 905 563 980">Name</th> <th data-bbox="563 905 927 980">Description</th> <th data-bbox="927 905 1053 980">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="466 980 563 1119">path</td> <td data-bbox="563 980 927 1119">Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons.</td> <td data-bbox="927 980 1053 1119">value: string</td> </tr> <tr> <td data-bbox="466 1119 563 1165">Color</td> <td data-bbox="563 1119 927 1165">Fill color to apply to the icon.</td> <td data-bbox="927 1119 1053 1165">string</td> </tr> <tr> <td data-bbox="466 1165 563 1255">style</td> <td data-bbox="563 1165 927 1255">Sets a style for the icon. Full menu of style options is available. You can also specify a style class.</td> <td data-bbox="927 1165 1053 1255">object</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string	Color	Fill color to apply to the icon.	string	style	Sets a style for the icon. Full menu of style options is available. You can also specify a style class .	object	object	empty	<p>Icon appearance when path is empty.</p> <table border="1" data-bbox="466 1367 1053 1717"> <thead> <tr> <th data-bbox="466 1367 563 1442">Name</th> <th data-bbox="563 1367 927 1442">Description</th> <th data-bbox="927 1367 1053 1442">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="466 1442 563 1581">path</td> <td data-bbox="563 1442 927 1581">Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons.</td> <td data-bbox="927 1442 1053 1581">value: string</td> </tr> <tr> <td data-bbox="466 1581 563 1627">Color</td> <td data-bbox="563 1581 927 1627">Fill color to apply to the icon.</td> <td data-bbox="927 1581 1053 1627">string</td> </tr> <tr> <td data-bbox="466 1627 563 1717">style</td> <td data-bbox="563 1627 927 1717">Sets a style for the icon. Full menu of style options is available. You can also specify a style class.</td> <td data-bbox="927 1627 1053 1717">object</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string	Color	Fill color to apply to the icon.	string	style	Sets a style for the icon. Full menu of style options is available. You can also specify a style class .	object	object	object
Name	Description	Property Type																																																
collapsed	<p>Icon appearance when path is collapsed.</p> <table border="1" data-bbox="466 401 1053 795"> <thead> <tr> <th data-bbox="466 401 563 476">Name</th> <th data-bbox="563 401 927 476">Description</th> <th data-bbox="927 401 1053 476">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="466 476 563 615">path</td> <td data-bbox="563 476 927 615">Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons.</td> <td data-bbox="927 476 1053 615">value: string</td> </tr> <tr> <td data-bbox="466 615 563 661">Color</td> <td data-bbox="563 615 927 661">Fill color to apply to the icon.</td> <td data-bbox="927 615 1053 661">string</td> </tr> <tr> <td data-bbox="466 661 563 795">style</td> <td data-bbox="563 661 927 795">Sets a style for the icon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td data-bbox="927 661 1053 795">object</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string	Color	Fill color to apply to the icon.	string	style	Sets a style for the icon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	object																																				
Name	Description	Property Type																																																
path	Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string																																																
Color	Fill color to apply to the icon.	string																																																
style	Sets a style for the icon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																																																
expanded	<p>Icon appearance when path is expanded.</p> <table border="1" data-bbox="466 905 1053 1255"> <thead> <tr> <th data-bbox="466 905 563 980">Name</th> <th data-bbox="563 905 927 980">Description</th> <th data-bbox="927 905 1053 980">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="466 980 563 1119">path</td> <td data-bbox="563 980 927 1119">Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons.</td> <td data-bbox="927 980 1053 1119">value: string</td> </tr> <tr> <td data-bbox="466 1119 563 1165">Color</td> <td data-bbox="563 1119 927 1165">Fill color to apply to the icon.</td> <td data-bbox="927 1119 1053 1165">string</td> </tr> <tr> <td data-bbox="466 1165 563 1255">style</td> <td data-bbox="563 1165 927 1255">Sets a style for the icon. Full menu of style options is available. You can also specify a style class.</td> <td data-bbox="927 1165 1053 1255">object</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string	Color	Fill color to apply to the icon.	string	style	Sets a style for the icon. Full menu of style options is available. You can also specify a style class .	object	object																																				
Name	Description	Property Type																																																
path	Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string																																																
Color	Fill color to apply to the icon.	string																																																
style	Sets a style for the icon. Full menu of style options is available. You can also specify a style class .	object																																																
empty	<p>Icon appearance when path is empty.</p> <table border="1" data-bbox="466 1367 1053 1717"> <thead> <tr> <th data-bbox="466 1367 563 1442">Name</th> <th data-bbox="563 1367 927 1442">Description</th> <th data-bbox="927 1367 1053 1442">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="466 1442 563 1581">path</td> <td data-bbox="563 1442 927 1581">Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons.</td> <td data-bbox="927 1442 1053 1581">value: string</td> </tr> <tr> <td data-bbox="466 1581 563 1627">Color</td> <td data-bbox="563 1581 927 1627">Fill color to apply to the icon.</td> <td data-bbox="927 1581 1053 1627">string</td> </tr> <tr> <td data-bbox="466 1627 563 1717">style</td> <td data-bbox="563 1627 927 1717">Sets a style for the icon. Full menu of style options is available. You can also specify a style class.</td> <td data-bbox="927 1627 1053 1717">object</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string	Color	Fill color to apply to the icon.	string	style	Sets a style for the icon. Full menu of style options is available. You can also specify a style class .	object	object																																				
Name	Description	Property Type																																																
path	Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string																																																
Color	Fill color to apply to the icon.	string																																																
style	Sets a style for the icon. Full menu of style options is available. You can also specify a style class .	object																																																
defaultNodeIcons	<p>Settings for the node icons. Options as follows:</p> <table border="1" data-bbox="358 1841 1193 1959"> <thead> <tr> <th data-bbox="358 1841 456 1917">Name</th> <th data-bbox="456 1841 1065 1917">Description</th> <th data-bbox="1065 1841 1193 1917">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="358 1917 456 1959">expanded</td> <td data-bbox="456 1917 1065 1959">Icon appearance when path is expanded.</td> <td data-bbox="1065 1917 1193 1959">object</td> </tr> </tbody> </table>	Name	Description	Property Type	expanded	Icon appearance when path is expanded.	object	object																																										
Name	Description	Property Type																																																
expanded	Icon appearance when path is expanded.	object																																																

	ed	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons.</td> <td>value: string</td> </tr> <tr> <td>Color</td> <td>Fill color to apply to the icon.</td> <td>string</td> </tr> <tr> <td>style</td> <td>Sets a style for the icon. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string	Color	Fill color to apply to the icon.	string	style	Sets a style for the icon. Full menu of style options is available. You can also specify a style class .	object	
	Name	Description	Property Type												
	path	Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string												
	Color	Fill color to apply to the icon.	string												
	style	Sets a style for the icon. Full menu of style options is available. You can also specify a style class .	object												
	collapsed	Icon appearance when path is collapsed.	object												
		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons.</td> <td>value: string</td> </tr> <tr> <td>Color</td> <td>Fill color to apply to the icon.</td> <td>string</td> </tr> <tr> <td>style</td> <td>Sets a style for the icon. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string	Color	Fill color to apply to the icon.	string	style	Sets a style for the icon. Full menu of style options is available. You can also specify a style class .	object	
	Name	Description	Property Type												
	path	Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string												
	Color	Fill color to apply to the icon.	string												
style	Sets a style for the icon. Full menu of style options is available. You can also specify a style class .	object													
empty	Icon appearance when path is empty.	object													
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons.</td> <td>value: string</td> </tr> <tr> <td>Color</td> <td>Fill color to apply to the icon.</td> <td>string</td> </tr> <tr> <td>style</td> <td>Sets a style for the icon. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string	Color	Fill color to apply to the icon.	string	style	Sets a style for the icon. Full menu of style options is available. You can also specify a style class .	object		
Name	Description	Property Type													
path	Path to the icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string													
Color	Fill color to apply to the icon.	string													
style	Sets a style for the icon. Full menu of style options is available. You can also specify a style class .	object													
selectedStyle	Sets a style for when nodes are selected. Full menu of style options is available. You can also specify a style class .	object													
unselectedStyle	Sets a style for when nodes are unselected. Full menu of style options is available. You can also specify a style class .	object													
rowHeight	Height, in pixels, of each row/node of the tree. Default is 24.	value: numeric													
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object													

Scripting

See the [Perspective - Tree Scripting page](#) for the full list of scripting functions available for this component.

Example



Property	Value
appearance.defaultNodelcons.collapsed.path	material/Play_arrow
appearance.defaultNodelcons.collapsed.color	#D9D900
appearance.defaultNodelcons.expanded.path	material/subdirectory_arrow_right
appearance.defaultNodelcons.expanded.color	#FFAC47
appearance.defaultNodelcons.empty.path	material/panorama_fish_eye
appearance.defaultNodelcons.empty.color	#000000

Perspective - Tree Scripting

This page details the various scripting, component, and extension functions available for [Perspective's Tree](#) component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

On this page ...

- [Component Events](#)
 - [onItemClicked](#)
- [Component Functions](#)
- [Extension Functions](#)

onItemClicked

Fires whenever a node is clicked.

Object Path	Type	Description
event.data	Any	The value of the contextual 'data' object on the clicked node.
event.itemPath	List	A list containing the item indexes leading to the item that was clicked.
event.label	String	The displayed text on the clicked item.

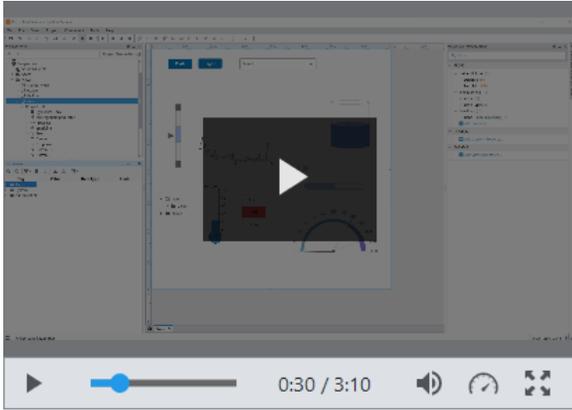
Component Functions

This component does not have component functions associated with it.

Extension Functions

This component does not have extension functions associated with it.

Perspective - Video Player



Component Palette Icon:



On this page ...

- [Designer Playback](#)
- [Mobile Platform Restrictions](#)
 - [iOS](#)
 - [iOS and Android Tablet:](#)
 - [Fullscreen Mode \(all desktop and mobile platforms\):](#)
- [User Interaction](#)
 - [User Interface](#)
 - [Controls](#)
- [Properties](#)
- [Component Events](#)

The Video Player component enables you to embed video or a live feed in Perspective views. In a view, the component displays either a live feed from an IP camera or a web hosted video file that is accessible from your gateway. The component is wrapped in a skin that gives you control over the style of the video controls and a uniform experience across browsers. Video controls can also be hidden (available on hover) to allow for a simple, clean video display.

The component requires a URL to a video or live feed. This also includes files placed on a [WebDev](#) mounted folder or file resource, which can be used to serve video files.

Designer Playback

The Designer contains an instance of JxBrowser to display your views as you build them. There are a [few codecs that JxBrowser does not support](#). Because of this, you may find that some videos do not play or display correctly while in the Designer. This is *only* a limitation of the codecs available to the Designer. The video will work as expected in a client session assuming it supports the required codec.

Note: This component plays embedded media files, which is not supported by the Safari 14 web browser. As a result, Sessions running in Safari 14 will not be able to utilize video playback on this component.

Mobile Platform Restrictions

Due to security restrictions on some mobile platforms (and in certain use cases), there are some special behaviors to be aware of when using this component.

iOS

All iOS devices require user interaction (touch, click, etc) to play the video. For this reason, the `controls.play` parameter will not play or pause the video. That must be done by the user clicking the play button. Because of this restriction, this platform also will only use the native look of the player (as determined by the web browser), rather than the custom look that is provided by the Perspective module.

iOS and Android Tablet:

On these platforms, security restrictions surround the use of the `controls.autoplay` property. Video content can only be automatically played if there is no audio. Because of this restriction, the `controls.mute` property must also be set to true.

Fullscreen Mode (all desktop and mobile platforms):

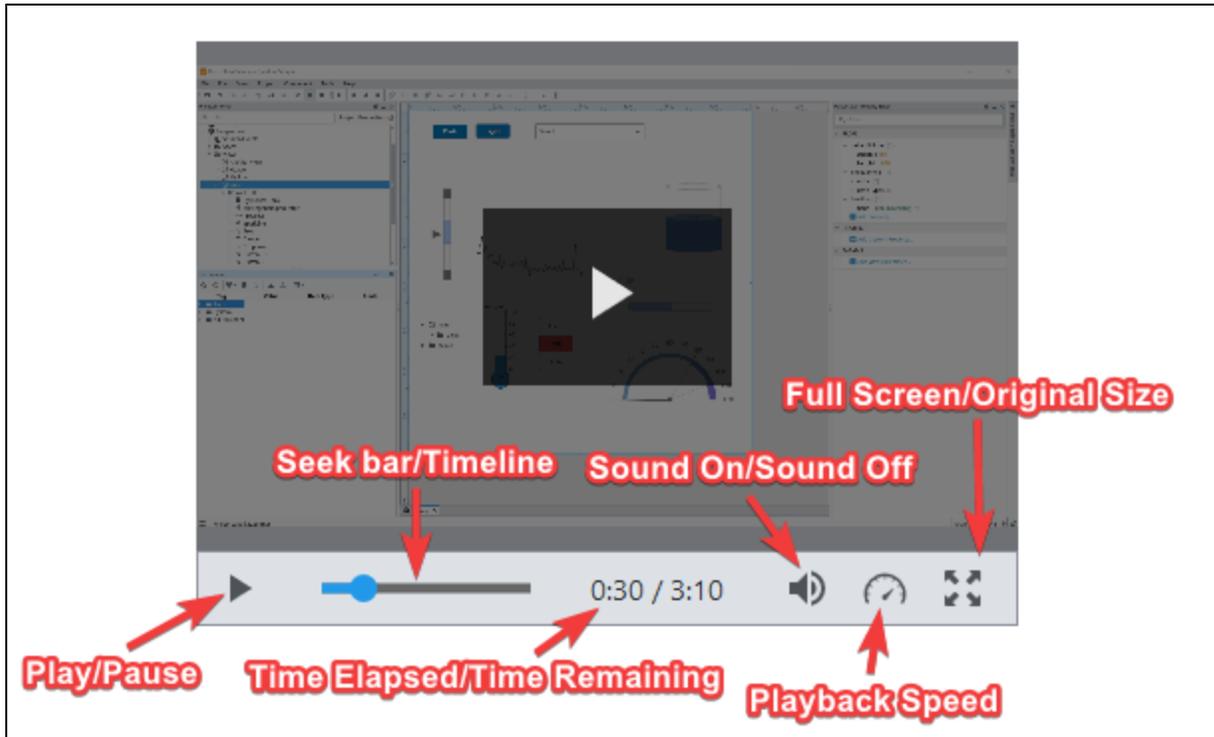
When in fullscreen mode, the native look of the player (determined by the web browser) is used as opposed to the custom look provided by the Perspective module. Because of this, the `controls.play` parameter will not play or pause the video. That must be done via user action (click, touch, etc).

User Interaction

The Video Player component properties have impact on the way a user can interact with it in the runtime.

Interaction	Description
Viewing on a Mobile Device with Android	On Android, you'll get the same experience as the desktop display with one exception; when going into fullscreen mode, you'll be presented with the native video control for a cleaner fullscreen experience on that platform.
Viewing on a Mobile Device with iOS	On iOS, you'll get the native video control for standard and fullscreen mode.

User Interface



Controls

The following controls are available to the user in a session.

Icon	Definition	Description
	Play	Starts the video play.
	Pause	Pauses the video play.
	Playback Speed	Sets the speed of the playback. Options are .25, .5, Normal, 1.25, 1.5, 2, 5, and 10 (for example, .5 is half speed, 2 is double speed, etc.).
	Seek bar/Timeline	Interactable slider representation of the time elapsed and time remaining.
	Sound On	Sound is turned on for the video. Clicking on this icon brings up a sliding bar with which you can adjust the volume.
	Sound Off	Sound is turned off for the video.
	Time elapsed/Time remaining	Displays the time elapsed in the video and the time remaining.
	Full Screen	Expands the video to full screen.



Original size

Returns the video to original screen size. You can also press the Esc key to return to original size.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type																								
source	The path to the source of the video or live feed.	value: string																								
liveFeed	Used to toggle the component to display a live feed. If set to true, the poster, autohideControls, controls, and status properties will be hidden as they pertain only to a video file.	value: boolean																								
poster	The path to an image that will display as the background image of the video file when the video has not yet loaded. (Hidden if props.liveFeed is set to true.)	value: string																								
autohideControls	Used to toggle the visible state of the control bar when displaying a video file. If set to true, the control bar will be displayed only when the mouse is hovered over the video. (Hidden if props.liveFeed is set to true.)	value: boolean																								
controls	Properties that are used to provide settings and interaction points with a video file. (Hidden if props.liveFeed is set to true.)	object																								
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>autoplay</td> <td>If set to true, the video will begin playing when the client session loads. While autoplay is enabled, the video will be muted while playing initially.</td> <td>value: boolean</td> </tr> <tr> <td>loop</td> <td>If enabled, the video file will play over indefinitely once it has completed.</td> <td>value: boolean</td> </tr> <tr> <td>mute</td> <td>If enabled, the video will show as being in a muted state and will have no volume.</td> <td>value: boolean</td> </tr> <tr> <td>play</td> <td>If enabled, the video will play. If disabled, the video will be paused.</td> <td>value: boolean</td> </tr> <tr> <td>seek</td> <td>The time (in seconds) from which the video should start playing.</td> <td>value: numeric</td> </tr> <tr> <td>volume</td> <td>A number (percentage value) representing the current volume of the video file.</td> <td>value: numeric</td> </tr> <tr> <td>playRate</td> <td>The speed at which the video will be played (where 1 is normal speed, .5 is half speed, 2 is double speed, etc.).</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	autoplay	If set to true, the video will begin playing when the client session loads. While autoplay is enabled, the video will be muted while playing initially.	value: boolean	loop	If enabled, the video file will play over indefinitely once it has completed.	value: boolean	mute	If enabled, the video will show as being in a muted state and will have no volume.	value: boolean	play	If enabled, the video will play. If disabled, the video will be paused.	value: boolean	seek	The time (in seconds) from which the video should start playing.	value: numeric	volume	A number (percentage value) representing the current volume of the video file.	value: numeric	playRate	The speed at which the video will be played (where 1 is normal speed, .5 is half speed, 2 is double speed, etc.).	value: numeric	
Name	Description	Property Type																								
autoplay	If set to true, the video will begin playing when the client session loads. While autoplay is enabled, the video will be muted while playing initially.	value: boolean																								
loop	If enabled, the video file will play over indefinitely once it has completed.	value: boolean																								
mute	If enabled, the video will show as being in a muted state and will have no volume.	value: boolean																								
play	If enabled, the video will play. If disabled, the video will be paused.	value: boolean																								
seek	The time (in seconds) from which the video should start playing.	value: numeric																								
volume	A number (percentage value) representing the current volume of the video file.	value: numeric																								
playRate	The speed at which the video will be played (where 1 is normal speed, .5 is half speed, 2 is double speed, etc.).	value: numeric																								
status	This property holds several sub-properties that are used to provide status updates while the video file goes through the playback process. These sub-properties should not be set as they are constantly overwritten during the playback process. (Hidden if props.liveFeed is set to true.)	object																								
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>loadedData</td> <td>True when the current playback position of the media has finished loading; often the first frame.</td> <td>value: boolean</td> </tr> <tr> <td>playing</td> <td>True when playback is ready to start after having been paused or delayed due to lack of data.</td> <td>value: boolean</td> </tr> <tr> <td>paused</td> <td>True when playback has been paused.</td> <td>value: boolean</td> </tr> <tr> <td>progress</td> <td>A number representing the time (in seconds) where playback has occurred.</td> <td>value: numeric</td> </tr> <tr> <td>rateChanged</td> <td>A number representing the current playback rate (1 being normal speed).</td> <td>value: numeric</td> </tr> <tr> <td>seeking</td> <td>True when a seek operation is in progress.</td> <td>value: boolean</td> </tr> </tbody> </table>	Name	Description	Property Type	loadedData	True when the current playback position of the media has finished loading; often the first frame.	value: boolean	playing	True when playback is ready to start after having been paused or delayed due to lack of data.	value: boolean	paused	True when playback has been paused.	value: boolean	progress	A number representing the time (in seconds) where playback has occurred.	value: numeric	rateChanged	A number representing the current playback rate (1 being normal speed).	value: numeric	seeking	True when a seek operation is in progress.	value: boolean				
Name	Description	Property Type																								
loadedData	True when the current playback position of the media has finished loading; often the first frame.	value: boolean																								
playing	True when playback is ready to start after having been paused or delayed due to lack of data.	value: boolean																								
paused	True when playback has been paused.	value: boolean																								
progress	A number representing the time (in seconds) where playback has occurred.	value: numeric																								
rateChanged	A number representing the current playback rate (1 being normal speed).	value: numeric																								
seeking	True when a seek operation is in progress.	value: boolean																								

	seeked	A number representing the time (in seconds) where the seek operation was completed.	value: numeric	
	waiting	True when playback has stopped because of temporary lack of data.	value: boolean	
	ended	True when playback has stopped because the end of the media was reached.	value: boolean	
controls	Sets a style for the controls on this component: the control bar, all controls, error messaging, context menus, and control popups. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. Classes are predefined styles in a project.			object
style	Sets a style for the background display of the component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .			object

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Perspective - Embedding Palette

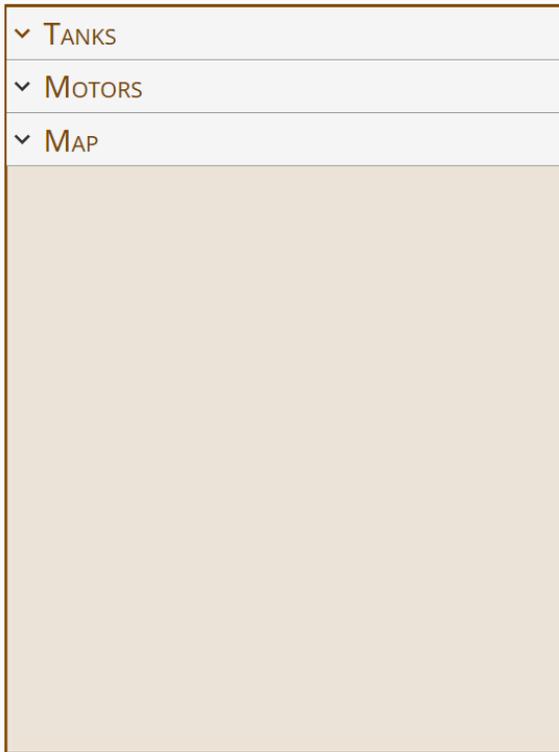
The following components function a bit differently, but what they all have in common is that each component can be embedded in multiple views of a project.

The Carousel component allows you to display a selection of rotating views. An Embedded View is an instance of a view that is used as a component within another view. The Flex Repeater component lets you easily create multiple instances of components for display in another view, each having the same look, feel, and functionality of the original components.

Here is a complete list of the embedding components, and a link pointing to a page containing the component's description, properties, and an example of how to configure it.

[In This Section ...](#)

Perspective - Accordion



On this page ...

- [Properties](#)
- [Scripting](#)
- [Example](#)

Component Palette Icon:



The Accordion Component allows the embedding of expandable/collapsible views which can be toggled with a click or a tap of their headers. The headers may contain text or a view.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description										
expansionMode	Determines how many items can be expanded at a given time. Options are 'single' and 'multiple'. When using 'single' any item that's currently clicked. When using 'multiple', items that are expanded will remain open until they are collapsed.										
items	An array of items in the accordion. Each item has a separate header and body configurations. <table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>expanded</td><td>Determines if the the accordion body expanded. Set to true to expand, false to collapse.</td></tr><tr><td>header</td><td>An object containing configuration options for the toggle icon.<table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>toggle</td><td>An object containing configuration options for the toggle icon.</td></tr></tbody></table></td></tr></tbody></table>	Name	Description	expanded	Determines if the the accordion body expanded. Set to true to expand, false to collapse.	header	An object containing configuration options for the toggle icon. <table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>toggle</td><td>An object containing configuration options for the toggle icon.</td></tr></tbody></table>	Name	Description	toggle	An object containing configuration options for the toggle icon.
Name	Description										
expanded	Determines if the the accordion body expanded. Set to true to expand, false to collapse.										
header	An object containing configuration options for the toggle icon. <table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>toggle</td><td>An object containing configuration options for the toggle icon.</td></tr></tbody></table>	Name	Description	toggle	An object containing configuration options for the toggle icon.						
Name	Description										
toggle	An object containing configuration options for the toggle icon.										

Name	Description																
enabled	Enables the collapse and expand toggle.																
expandedIcon	<p>An object containing configuration options for the header icon while the item body is expanded.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Shorthand path to the icon source, in format: library/iconName.</td> <td>value: string</td> </tr> <tr> <td>color</td> <td>Color of the icon. May instead 'fill' in the styles prop.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Sets a style for the expandedIcon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Shorthand path to the icon source, in format: library/iconName.	value: string	color	Color of the icon. May instead 'fill' in the styles prop.	value: string	style	Sets a style for the expandedIcon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object				
Name	Description	Property Type															
path	Shorthand path to the icon source, in format: library/iconName.	value: string															
color	Color of the icon. May instead 'fill' in the styles prop.	value: string															
style	Sets a style for the expandedIcon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															
collapsedIcon	<p>An object containing configuration options for the header icon while the item body is collapsed.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Shorthand path to icon source, in format: library/iconName</td> <td>value: string</td> </tr> <tr> <td>color</td> <td>Color of the icon. If deleted, the Shape "fill" property in the adjacent style object will determine the color of the icon.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Sets a style for the collapsedIcon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Shorthand path to icon source, in format: library/iconName	value: string	color	Color of the icon. If deleted, the Shape "fill" property in the adjacent style object will determine the color of the icon.	value: string	style	Sets a style for the collapsedIcon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object				
Name	Description	Property Type															
path	Shorthand path to icon source, in format: library/iconName	value: string															
color	Color of the icon. If deleted, the Shape "fill" property in the adjacent style object will determine the color of the icon.	value: string															
style	Sets a style for the collapsedIcon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															
content	<p>An object containing configuration options for the content.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>type</td> <td>Whether text or a view will be rendered in this accordion header. Set this property to either 'text' or 'view'.</td> </tr> <tr> <td>text</td> <td>Text to display for this accordion header.</td> </tr> <tr> <td>useDefaultViewWidth</td> <td>Use of view's default width instead of adjusting based on the content's width.</td> </tr> <tr> <td>useDefaultViewHeight</td> <td>Use of view's default height instead of adjusting based on the content's height.</td> </tr> <tr> <td>viewPath</td> <td>Path to view to render in this accordion header.</td> </tr> <tr> <td>viewParams</td> <td> <p>Params to pass to this view rendered in this accordion header.</p> <div style="border: 1px solid orange; padding: 5px; margin: 5px 0;"> <p>The following feature is new in Ignition version 8.1.4 Click here to check out the other new features</p> </div> <p>As of 8.1.4 a dropdown list of parameters is available when the user clicks the Add Object Member  icon. This makes it easy to add parameters from the rendered view.</p> </td> </tr> <tr> <td>style</td> <td>Sets a style for the content. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> </tr> </tbody> </table>	Name	Description	type	Whether text or a view will be rendered in this accordion header. Set this property to either 'text' or 'view'.	text	Text to display for this accordion header.	useDefaultViewWidth	Use of view's default width instead of adjusting based on the content's width.	useDefaultViewHeight	Use of view's default height instead of adjusting based on the content's height.	viewPath	Path to view to render in this accordion header.	viewParams	<p>Params to pass to this view rendered in this accordion header.</p> <div style="border: 1px solid orange; padding: 5px; margin: 5px 0;"> <p>The following feature is new in Ignition version 8.1.4 Click here to check out the other new features</p> </div> <p>As of 8.1.4 a dropdown list of parameters is available when the user clicks the Add Object Member  icon. This makes it easy to add parameters from the rendered view.</p>	style	Sets a style for the content. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .
Name	Description																
type	Whether text or a view will be rendered in this accordion header. Set this property to either 'text' or 'view'.																
text	Text to display for this accordion header.																
useDefaultViewWidth	Use of view's default width instead of adjusting based on the content's width.																
useDefaultViewHeight	Use of view's default height instead of adjusting based on the content's height.																
viewPath	Path to view to render in this accordion header.																
viewParams	<p>Params to pass to this view rendered in this accordion header.</p> <div style="border: 1px solid orange; padding: 5px; margin: 5px 0;"> <p>The following feature is new in Ignition version 8.1.4 Click here to check out the other new features</p> </div> <p>As of 8.1.4 a dropdown list of parameters is available when the user clicks the Add Object Member  icon. This makes it easy to add parameters from the rendered view.</p>																
style	Sets a style for the content. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .																
height	The height of the header.																
reverse	Reverses the order of the toggle and header content, (i.e., from left side to the right side).																

	style	Sets a style for the header. Full menu of style options is available for text, background, margin and padding, border, s miscellaneous. You can also specify a style class .
body	An object containing configuration options for the body.	
	Name	Description
	viewPath	<p>Path of the view to display.</p> <p>The following feature is new in Ignition version 8.1.29 Click here to check out the other new features</p> <p>If a path is present in the viewPath property, an Open View  icon will appear that will navigate directly to th clicked.</p>
	viewParams	<p>Parameters to be passed to the view. Names in this object must match input parameters defined on the view.</p> <p>The following feature is new in Ignition version 8.1.4 Click here to check out the other new features</p> <p>As of 8.1.4 a dropdown list of parameters is available when the user clicks the Add Object Member  icon. easy to add parameters from the rendered view.</p>
	useDefaultViewWidth	Use of view's default width instead of adjusting based on the content's width.
	useDefaultViewHeight	Use of view's default height instead of adjusting based on the content's height.
	height	The height of the body.
	style	Sets a style for the body. Full menu of style options is available for text, background, margin and padding, border and miscellaneous. You can also specify a style class .
unused SpaceStyle	Sets a style for the empty area at the bottom of the accordion. Full menu of style options is available. You can also specify a style class .	
style	Sets a style for this component. Full menu of style options is available. You can also specify a style class .	

Scripting

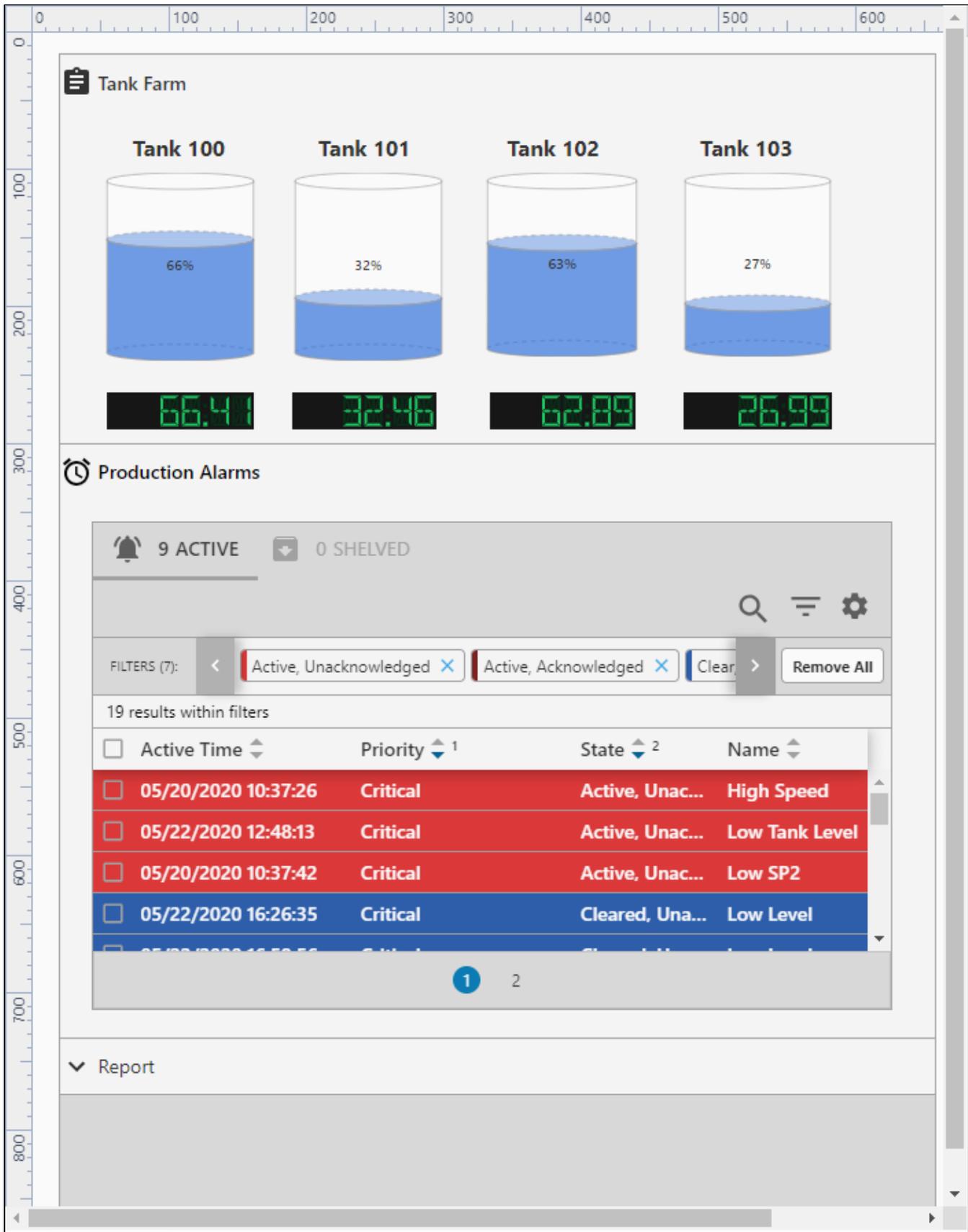
See the [Perspective - Accordion Scripting page](#) for the full list of scripting functions available for this component.

Example

This Accordion example has three multiple embedded expandable and collapsible views. Each view can be expanded or collapsed by clicking on their headers. The Tank Farm and the Production Alarms views are both expanded while the Report view is collapsed.

The three views used in the accordion example are working views and the component was configured to use these existing views. Configuring the accordion component is just a matter of how you want to present the information on the component, and then to configure its properties.

Preview Mode



Property Settings

Property	Value

expansionMode	multiple
props.items.0.header.toggle.expandedIcon.path	material/assignment
props.items.0.header.toggle.collapsedIcon.path	material/expand_more
props.items.0.header.content.type	text
props.items.0.header.content.text	Tank Farm
props.items.0.body.viewPath	Tank Farm
props.items.0.body.style.marginTop	3px
props.items.0.body.style.margin	10px
props.items.1.header.toggle.expandedIcon.path	material/alarm
props.items.1.header.toggle.collapsedIcon.path	material/expand_more
props.items.1.header.content.type	text
props.items.1.header.content.text	Production Alarms
props.items.1.body.viewPath	Production Alarms
props.items.1.body.style.marginTop	3px
props.items.1.body.style.margin	10px
props.items.2.header.toggle.expandedIcon.path	material/info
props.items.2.header.toggle.collapsedIcon.path	material/expand_more
props.items.2.header.content.type	text
props.items.2.header.content.text	Report
props.items.2.body.viewPath	Report4
props.items.2.body.style.marginTop	3px
props.items.2.body.style.margin	10px

Perspective - Accordion Scripting

This page details the various scripting, component, and extension functions available for [Perspective's Accordion](#) component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

On this page ...

- [Component Events](#)
 - [onItemExpanded](#)
 - [onItemCollapsed](#)
- [Component Functions](#)
- [Extension Functions](#)

onItemExpanded

This event is fired when an item is expanded.

Object Path	Type	Description
event.index	Numeric	The index of the item that was expanded.

onItemCollapsed

This event is fired when an item is expanded.

Object Path	Type	Description
event.index	Numeric	The index of the item that was collapsed.

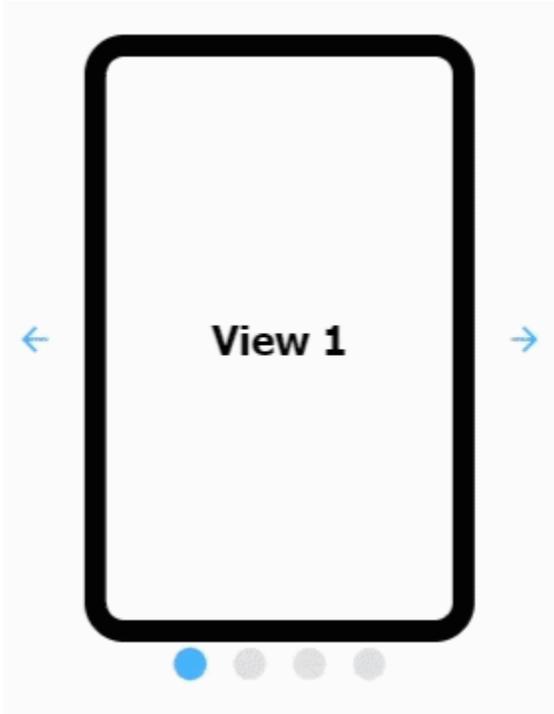
Component Functions

This component does not have component functions associated with it.

Extension Functions

This component does not have extension functions associated with it.

Perspective - Carousel



On this page ...

- [Properties](#)
- [Component Events](#)
- [Examples](#)

Component Palette Icon:



The Carousel component allows you to display a selection of rotating views at a defined rate with a link to the view on a page in your project. The Carousel component can automatically cycle through the views or a user can click through the views on demand, either way, still providing a link to the view on a page.

This version updates how the component handles drag transition ("swiping" across embedded views). Only common rotational angles are supported (90, 180, 270, 360) for drag transitions. If the Carousel's rotation doesn't match one of those angles, then drag transition is disabled.

Here are a few best practices when working with the Carousel component.

- Components such as the Video Player and Map are performance intensive components and should not be embedded in the Carousel since they can hurt session performance.
- Avoid embedding views containing carousels in a carousel. This can become confusing for users.
- Avoid embedding views that contain iFrame components. It's easy for content embedded in an iFrame to steal focus from other components. Also, depending on the content in the iFrame, it may impact performance.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
views	Visible area of a page. Can have multiple views in the carousel component.	
viewPath	The path of the view to render in this carousel.	value: string

The following feature is new in Ignition version **8.1.29**
[Click here](#) to check out the other new features

If a path is present in the viewPath property field, an Open View  icon will appear that will navigate directly to the view when clicked.

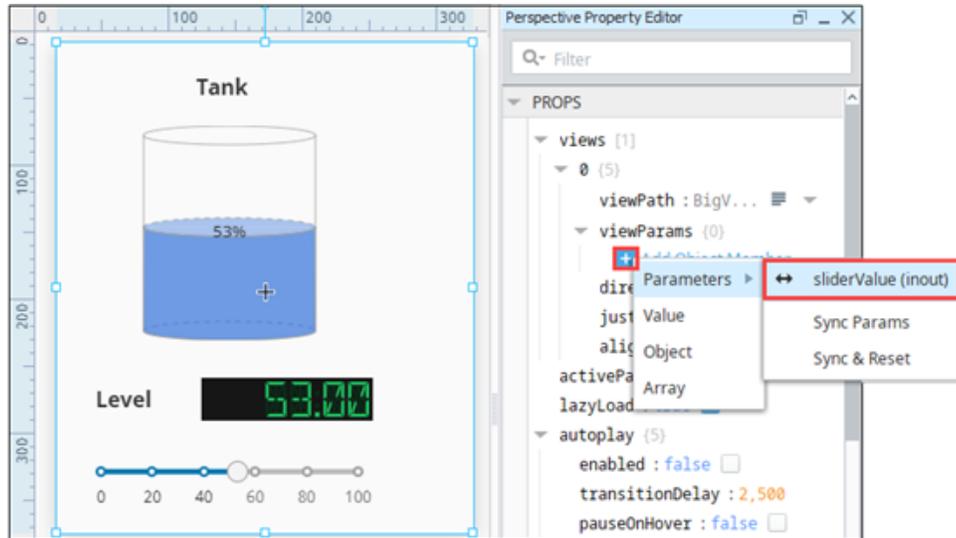
viewParams

Parameters to provide to this rendered view.

object

The following feature is new in Ignition version **8.1.4**
[Click here](#) to check out the other new features

As of 8.1.4 a dropdown list of parameters is available when the user clicks the **Add Object Member**  icon. This makes it easy to add parameters from the rendered view.



direction

Direction of the child layout. Options are row or column.

value: string

justify

Adjusts placement of view along the main axis. Options are flex-start, flex-end, or center.

value: string

alignItems

Adjusts placement of view along the cross axis. Options are flex-start, flex-end, or center.

value: string

activePane

Active pane being displayed.

value: number

lazyLoad

Load views on demand or progressively.

value: boolean

autoplay

Settings controlling the rotation of views in the carousel.

object

Name	Description	Property Type
enabled	If true, the carousel will automatically rotate the views according to the transitionDelay.	value: boolean
transitionDelay	Delay (in ms) at which slides scroll through the carousel when autoplay is true.	value: numeric
pauseOnHover	Pauses autoplay when user hovers the mouse over the view.	value: boolean
pauseOnFocus	Pauses autoplay on focus.	value: boolean
pauseOnDotHover	Pauses autoplay when user hovers the mouse over the dot for the view.	value: boolean

behavior

Behavior and interaction related carousel options.

object

Name	Description	Property Type
transitionSpeed	The speed (in ms) at which the carousel transitions between slides.	value: numeric

fade	Enables slides to fade in and out of view on transition	value: boolean
mobileSwipeable	Enables swiping on mobile devices to change slides.	value: boolean
desktopDraggable	Enables scrolling via drag the desktop.	value: boolean

appearance

Appearance related carousel options.

ot

Name	Description	Property Type																																										
dots	Carousel dots configuration.	object																																										
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enables dots at the bottom of the carousel component.</td> <td>value: boolean</td> </tr> <tr> <td>iconPath</td> <td>Path to the icon that will be used.</td> <td>value: string</td> </tr> <tr> <td>styles</td> <td>Configure active and inactive styles for the dot icon.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>active</td> <td>Sets a style for the dot icon when active. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>inactive</td> <td>Sets a style for the dot icon when inactive. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Enables dots at the bottom of the carousel component.	value: boolean	iconPath	Path to the icon that will be used.	value: string	styles	Configure active and inactive styles for the dot icon.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>active</td> <td>Sets a style for the dot icon when active. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>inactive</td> <td>Sets a style for the dot icon when inactive. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	active	Sets a style for the dot icon when active. Full menu of style options is available. You can also specify a style class .	object	inactive	Sets a style for the dot icon when inactive. Full menu of style options is available. You can also specify a style class .	object																				
Name	Description	Property Type																																										
enabled	Enables dots at the bottom of the carousel component.	value: boolean																																										
iconPath	Path to the icon that will be used.	value: string																																										
styles	Configure active and inactive styles for the dot icon.	object																																										
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>active</td> <td>Sets a style for the dot icon when active. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>inactive</td> <td>Sets a style for the dot icon when inactive. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	active	Sets a style for the dot icon when active. Full menu of style options is available. You can also specify a style class .	object	inactive	Sets a style for the dot icon when inactive. Full menu of style options is available. You can also specify a style class .	object																																		
Name	Description	Property Type																																										
active	Sets a style for the dot icon when active. Full menu of style options is available. You can also specify a style class .	object																																										
inactive	Sets a style for the dot icon when inactive. Full menu of style options is available. You can also specify a style class .	object																																										
arrows	Carousel arrows configuration.	object																																										
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Enables arrows at the sides of the carousel component.</td> <td>value: boolean</td> </tr> <tr> <td>next</td> <td>Next arrow icon configuration.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>iconPath</td> <td>Path to the icon that will be used for the next arrow, if provided.</td> <td>value: string</td> </tr> <tr> <td>fillColor</td> <td>Fill color to apply to the icon.</td> <td>string</td> </tr> <tr> <td>style</td> <td>Sets a style for the next arrow. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>previous</td> <td>Previous arrow icon configuration.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>iconPath</td> <td>Path to the icon that will be used for the next arrow, if provided.</td> <td>value: string</td> </tr> <tr> <td>fillColor</td> <td>Fill color to apply to the icon.</td> <td>string</td> </tr> <tr> <td>style</td> <td>Sets a style for the next arrow. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Enables arrows at the sides of the carousel component.	value: boolean	next	Next arrow icon configuration.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>iconPath</td> <td>Path to the icon that will be used for the next arrow, if provided.</td> <td>value: string</td> </tr> <tr> <td>fillColor</td> <td>Fill color to apply to the icon.</td> <td>string</td> </tr> <tr> <td>style</td> <td>Sets a style for the next arrow. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	iconPath	Path to the icon that will be used for the next arrow, if provided.	value: string	fillColor	Fill color to apply to the icon.	string	style	Sets a style for the next arrow. Full menu of style options is available. You can also specify a style class .	object		previous	Previous arrow icon configuration.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>iconPath</td> <td>Path to the icon that will be used for the next arrow, if provided.</td> <td>value: string</td> </tr> <tr> <td>fillColor</td> <td>Fill color to apply to the icon.</td> <td>string</td> </tr> <tr> <td>style</td> <td>Sets a style for the next arrow. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	iconPath	Path to the icon that will be used for the next arrow, if provided.	value: string	fillColor	Fill color to apply to the icon.	string	style	Sets a style for the next arrow. Full menu of style options is available. You can also specify a style class .	object		
Name	Description	Property Type																																										
enabled	Enables arrows at the sides of the carousel component.	value: boolean																																										
next	Next arrow icon configuration.	object																																										
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>iconPath</td> <td>Path to the icon that will be used for the next arrow, if provided.</td> <td>value: string</td> </tr> <tr> <td>fillColor</td> <td>Fill color to apply to the icon.</td> <td>string</td> </tr> <tr> <td>style</td> <td>Sets a style for the next arrow. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	iconPath	Path to the icon that will be used for the next arrow, if provided.	value: string	fillColor	Fill color to apply to the icon.	string	style	Sets a style for the next arrow. Full menu of style options is available. You can also specify a style class .	object																															
Name	Description	Property Type																																										
iconPath	Path to the icon that will be used for the next arrow, if provided.	value: string																																										
fillColor	Fill color to apply to the icon.	string																																										
style	Sets a style for the next arrow. Full menu of style options is available. You can also specify a style class .	object																																										
previous	Previous arrow icon configuration.	object																																										
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>iconPath</td> <td>Path to the icon that will be used for the next arrow, if provided.</td> <td>value: string</td> </tr> <tr> <td>fillColor</td> <td>Fill color to apply to the icon.</td> <td>string</td> </tr> <tr> <td>style</td> <td>Sets a style for the next arrow. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	iconPath	Path to the icon that will be used for the next arrow, if provided.	value: string	fillColor	Fill color to apply to the icon.	string	style	Sets a style for the next arrow. Full menu of style options is available. You can also specify a style class .	object																															
Name	Description	Property Type																																										
iconPath	Path to the icon that will be used for the next arrow, if provided.	value: string																																										
fillColor	Fill color to apply to the icon.	string																																										
style	Sets a style for the next arrow. Full menu of style options is available. You can also specify a style class .	object																																										
useDefaultViewWidth	Enables the use of the view's default width instead of dynamically adjusting based on the available width.	value: boolean																																										

	useDefaultViewHeight	Enables the use of the view's default height instead of dynamically adjusting based on the available height.	value: boolean
	slidesToShow	Number of views to show on each carousel page.	value: numeric
	slidePadding	Applies padding between slides.	value: numeric
	reverse	Reverses the slide order. Meaning, the first view rendered on the component will be the last element in the views array property.	value: boolean
style	Sets a style for this component. Full menu of style options is available. You can also specify a style class .		ot

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

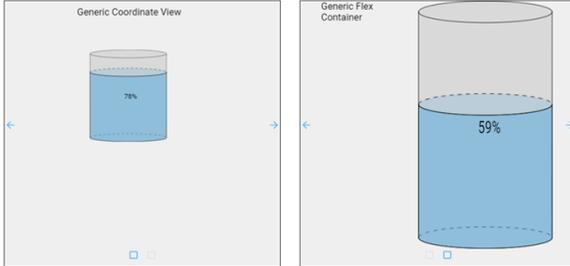
Examples

See the [Carousel Component Examples](#) page for more examples of using the Carousel component.

Carousel Component Examples

Example 1

In this example we made two view containers. One is a coordinate container and the other is flex container, named **generic_coordinate** and **generic_flex**, respectively. Each has a Label component and a Cylindrical Tank component. The Label's text property shows the name of the view it is in. The Cylindrical Tank's value property shows a level above 20. The **generic_flex** view is set to display as a row. We did not adjust any other settings on these two generic screens.



Once the two view containers are set up, we can create the Carousel with the following properties and values:

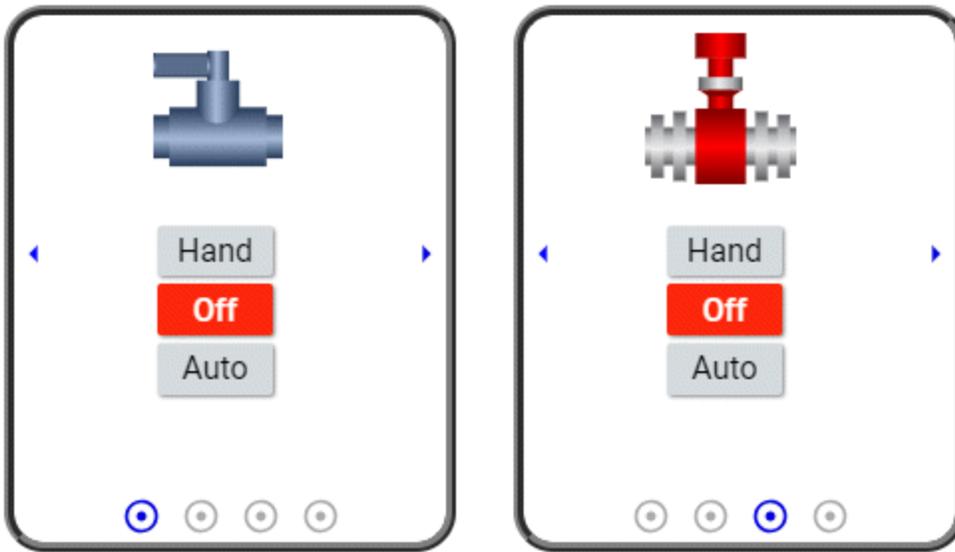
Property	Value	Style Category
props.views.0.viewPath	generic_coordinate	N/A
props.view.1.viewPath	generic_flex	N/A
props.autoplay.enabled	true	N/A
props.autoplay.pauseOnHover	true	N/A
props.appearance.useDefaultViewWidth	false	N/A
props.appearance.useDefaultViewHeight	false	N/A
props.appearance.dots.iconPath	/system/images/material/crop_square	N/A
props.appearance.arrows.enabled	true	N/A
props.style.borderStyle	solid	border
props.style.borderWidth	1px	border
props.style.backgroundColor	#EFEFEF	background

Example 2

In this example, we have set up a Carousel component that enables users to quickly move between four views showing valves and multistate buttons. The views must be created before the carousel, but it doesn't matter what the contents of the views are. You could use any combination of views including duplicates. Example 3 below describes setting up the views to scroll through in more detail.

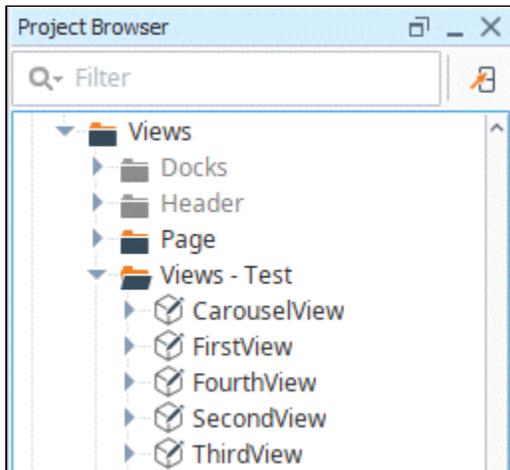
On this page ...

- [Example 1](#)
- [Example 2](#)
- [Example 3](#)
 - [Initial Project Setup](#)
 - [Set Up the Carousel View](#)



They are named as follows in a **Views - Test** folder:

- CarouselView
- FirstView
- SecondView
- ThirdView
- FourthView



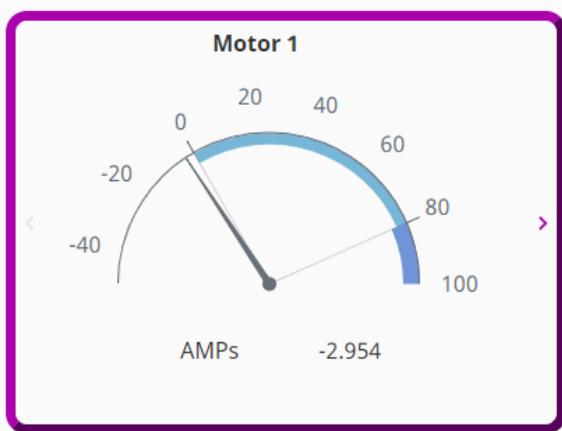
The properties used for the Carousel component are as follows:

Property	Value	Style Category
props.views.0.viewPath	Views - Test/FirstView	N/A
props.views.1.viewPath	Views - Test/SecondView	N/A
props.views.2.viewPath	Views - Test/ThirdView	N/A
props.views.3.viewPath	Views - Test/FourthView	N/A
props.autoplay.enabled	false	N/A
props.appearance.useDefaultViewWidth	true	N/A
props.appearance.useDefaultViewHeight	true	N/A
props.appearance.dots.enabled	true	N/A
props.appearance.dots.iconPath	material/adjust	N/A
props.appearance.dots.styles.active.fillColor	#0000D9	N/A

props.appearance.arrows.enabled	true	N/A
props.appearance.arrows.next.iconPath	material/arrow_right	N/A
props.appearance.arrows.next.fillColor	#0000D9	N/A
props.appearance.arrows.previous.iconPath	material/arrow_left	N/A
props.appearance.arrows.previous.fillColor	#0000D9	N/A
props.style.borderStyle	ridge	border
props.style.borderWidth	5px	border
props.style.borderRadius	20	border
props.style.borderColor	#808080	border

Example 3

This example demonstrates a more complex case of how to configure side scrolling using the Carousel component starting with the initial project setup.



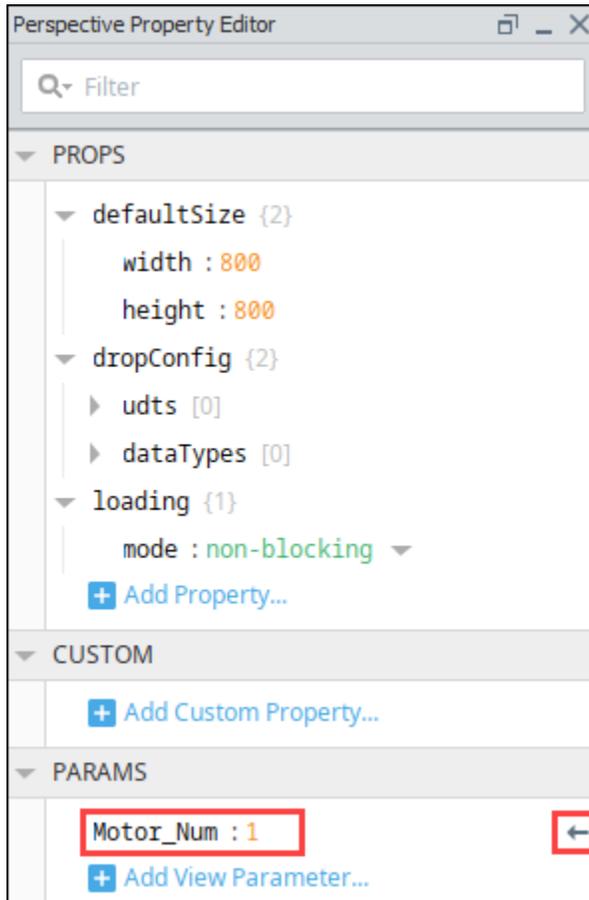
Initial Project Setup

First we'll create three views for the carousel to scroll through.

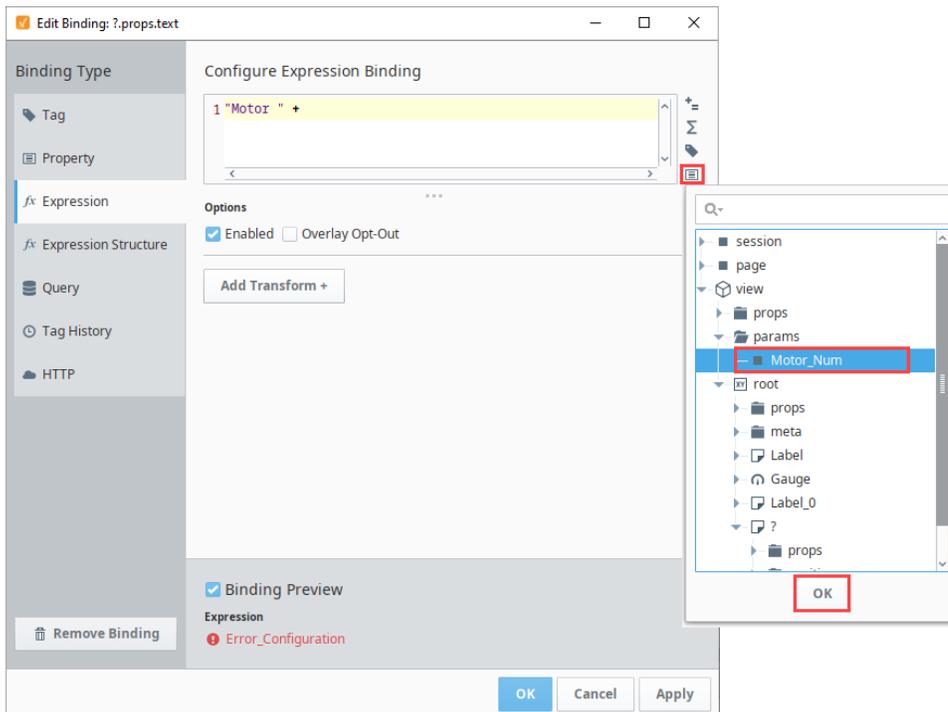
1. In the Project Browser, right-click on Views and select the **NewFolder**  option. We named our folder **Carousel Example**.
2. Right-click on the **Carousel Example** folder and select the **NewView**  option.
Name: **Motor**
Layout: **Coordinate**
Page URL: unchecked

3. Click **Create View**.

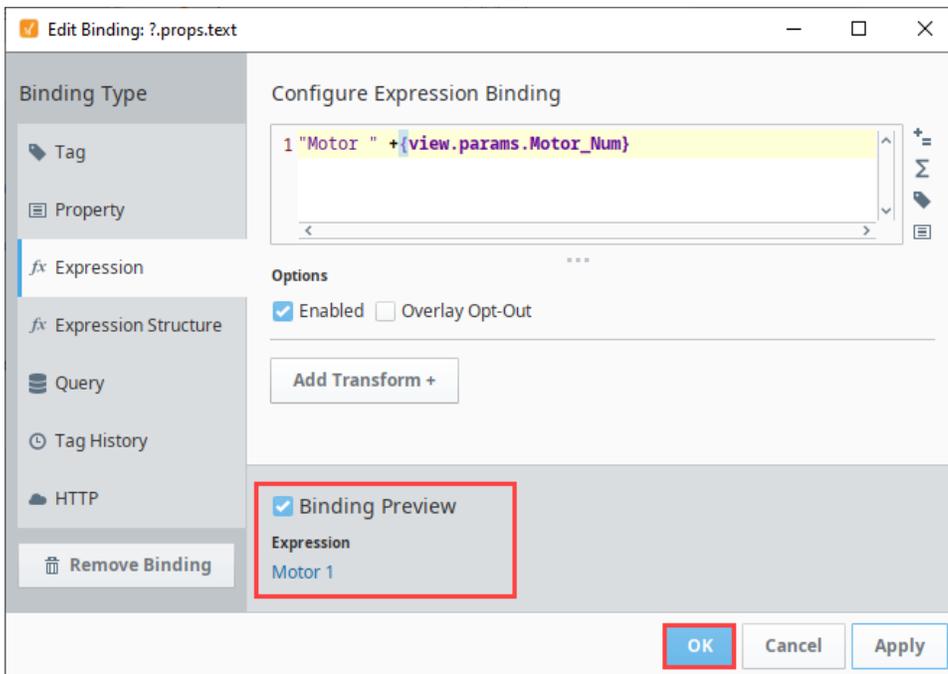
4. Next, create a view parameter that we can use for the motor number. In the Property Editor under PARAMS, click on the **Add View Parameter** icon.
 - a. Select **value**.
 - b. Change key to **Motor_Num**.
 - c. Change value to **1**.
 - d. Make sure the arrow icon is facing to the left, indicating this is an input parameter.



5. Next, make a title that will change depending on the Motor that's being displayed. Drag a Label component onto the view.
6. Bind the Label's text property to the view parameter as follows:
 - a. Click the **binding**  icon next to the text property.
 - b. Select **Expression** as the binding type.
 - c. Enter **"Motor " +**.
 - d. Click the **Property**  icon then scroll down and select the Motor_Num view parameter.

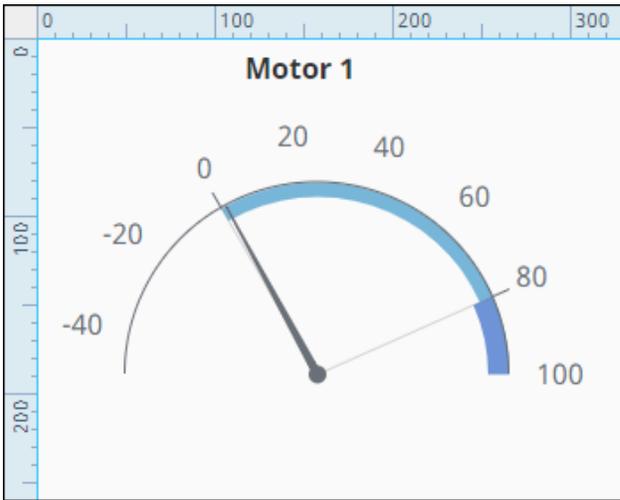


- e. Click **OK**. You'll now see the binding preview shows "Motor 1."

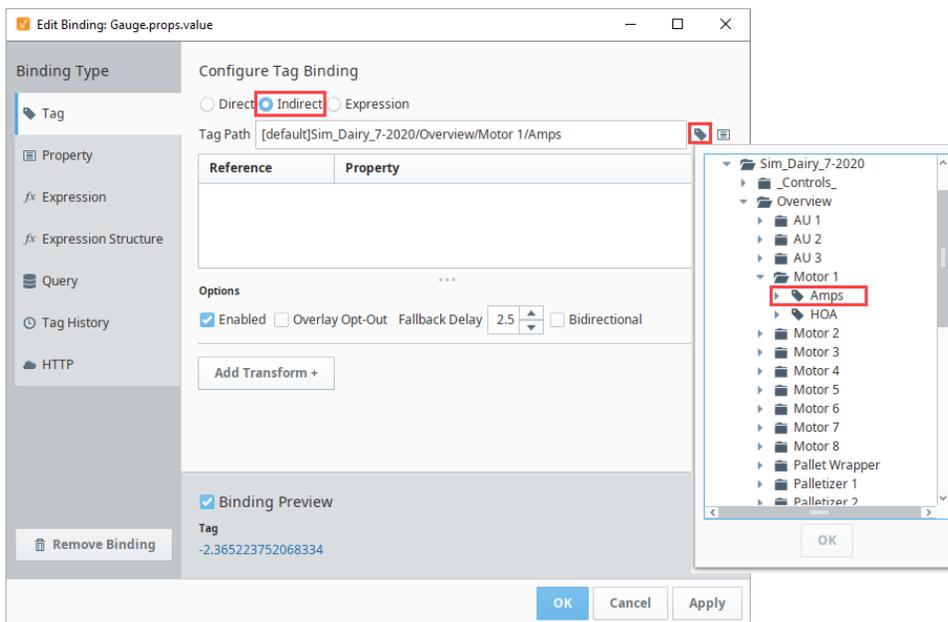


- f. Click **OK** to save the binding.

- Next, drag a Gauge component onto the view to display motor Amps.
- Align the Motor label centered above the Gauge component.
- In the Property Editor, expand the outerAxis properties. Change the properties as follows:
 minValue: **-50**
 maxValue: **100**

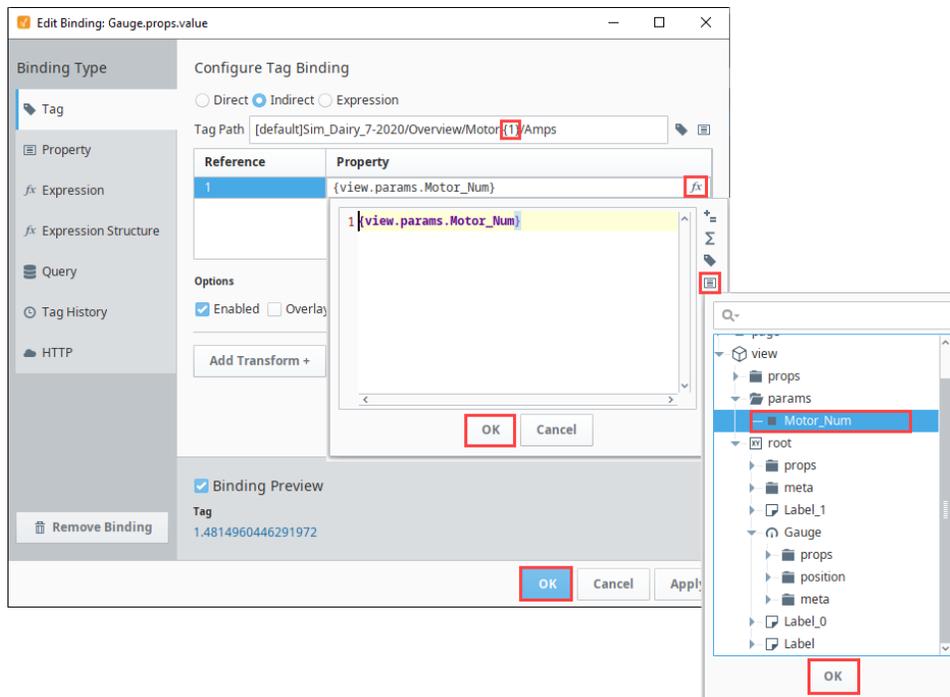


10. Now we'll set up an indirect tag binding using one of the motor tags in the Dairy simulator. (For more information on the Dairy simulator, see [Programmable Device Simulator](#).) Select the Gauge component.
11. In the Property Editor, click the **binding**  icon next to the value property. On the Configure Tag Binding page set the following:
 - a. Choose **Tag** as the binding type.
 - b. Select the **Indirect** option.
 - c. Next to the Tag Path field, click on the **Tag**  icon and navigate to the the **Motor 1/Amps** tag in the [Dairy simulator program](#).
 - d. Click **OK**.

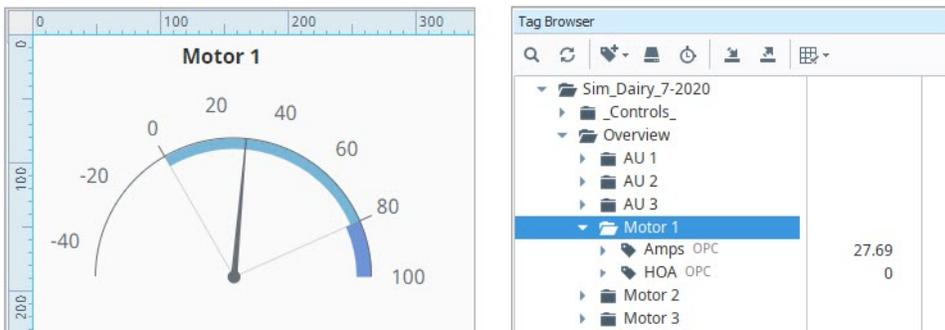


- e. In the Tag Path field, replace the 1 with **{1}**.
- f. In the References list, under Property, click on the **Functions**  icon.
- g. Click the **Properties**  icon.
- h. Scroll to the view params folder and select **Motor_Num**. Click **OK**.
- i. Click **OK** again. You'll see the binding in the preview area.

j. Click **OK** to save the binding.



12. Confirm the Gauge now displays the value of the Amps tag.



13. Drag a Label component onto your view. Place it under the Gauge component and change the text property to **AMPS**.

14. Drag another Label component onto your view and place it next to the AMPS label. We'll set up a similar indirect tag binding on this label.

15. Select the second Label click the **binding** icon next to the value property.

a. Select **Tag** as the binding type and click the **Indirect** radio button.

b. Next to the Tag Path field, click on the **Tag** icon and navigate to the the Motor 1/Amps Tag in the Dairy simulator program.

c. Click **OK**.

d. In the Tag Path field, replace the 1 with **{1}**.

e. In the References list, under Property, click on the **Functions** icon.

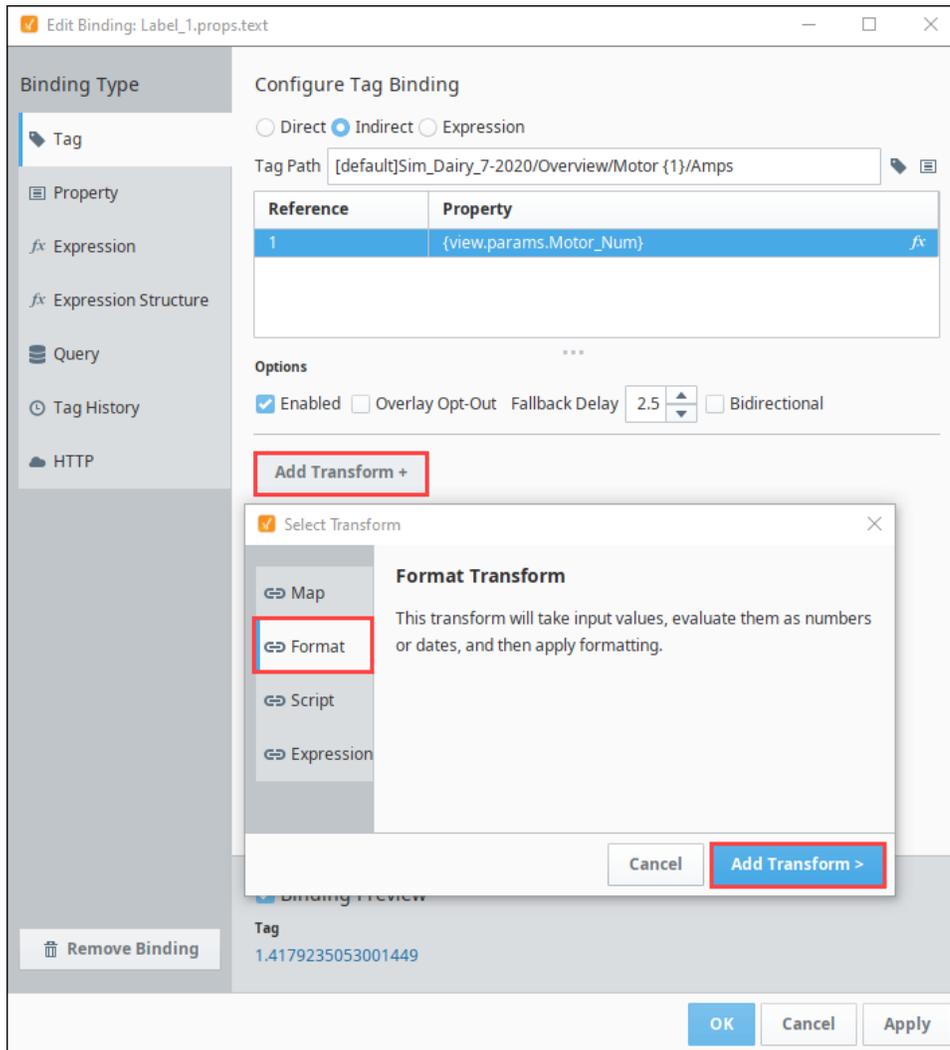
f. Click the **Properties** icon.

g. Scroll to the view params folder and select **Motor_Num**. Click **OK**.

h. Click **Apply**. You'll see the binding in the preview area.

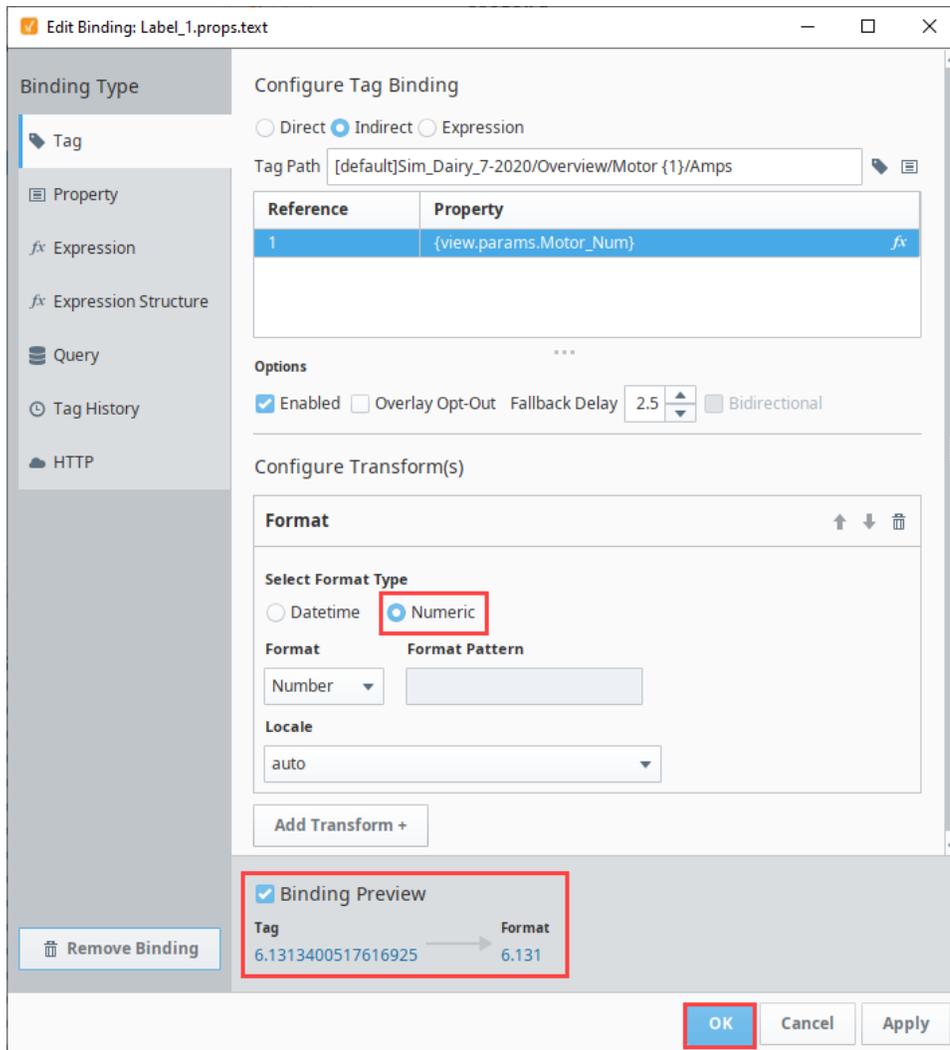
i. For this Label component we don't want the full tag value displayed. So we'll add a Transform to limit it to two decimal points. Click on **Add Transform**.

j. Select **Format**, then click **Add Transform**.



- k. Select **Numeric** as the Format type. The displayed value will now be shortened to just two decimal points. You can see the format in the Binding Preview.

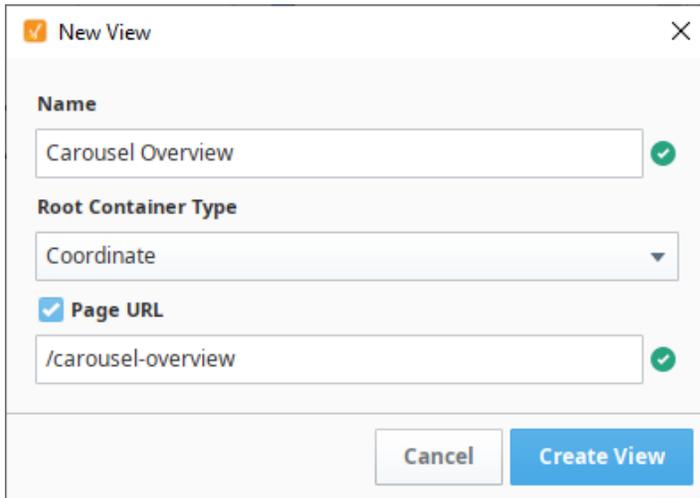
I. Click **OK** to save the binding.



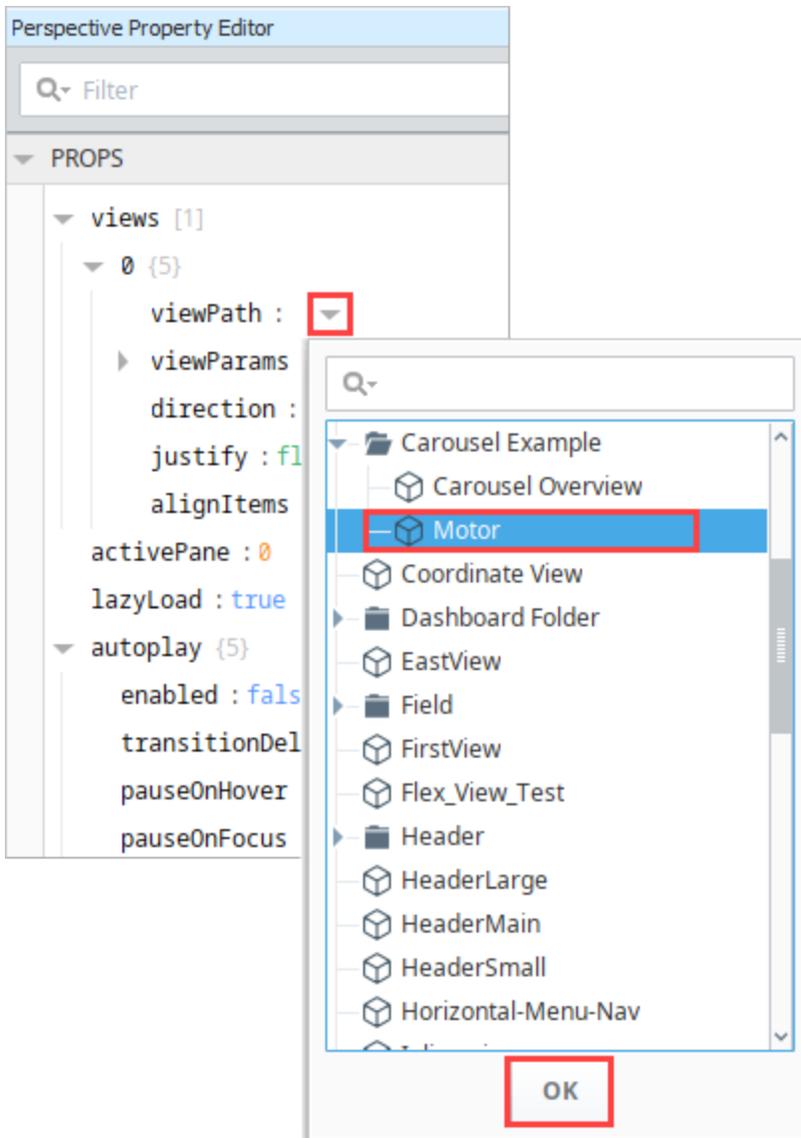
Set Up the Carousel View

Now we'll set up a view that holds the Carousel component.

1. Right-click on the Carousel Example folder and select the **NewView**  option
2. Name the view **Carousel Overview**. Check the Page URL option.
3. Click **Create View**.

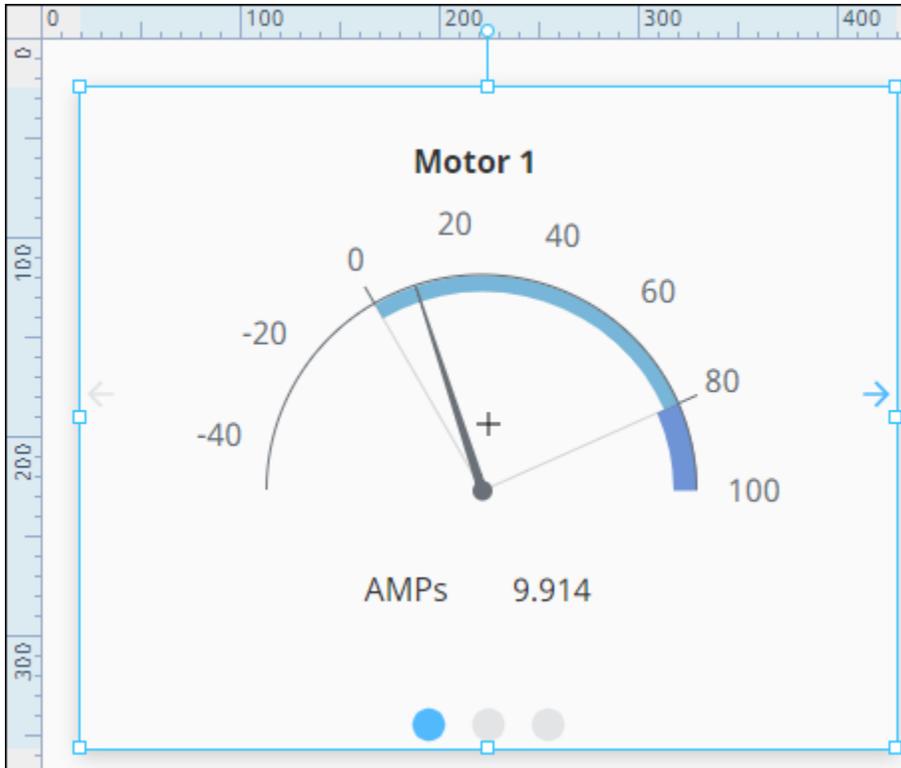


4. Drag a Carousel component onto the view.
5. In the Property Editor, click the **Expand** icon in expand the props.views.0.viewPath property and select the **Motor** view.

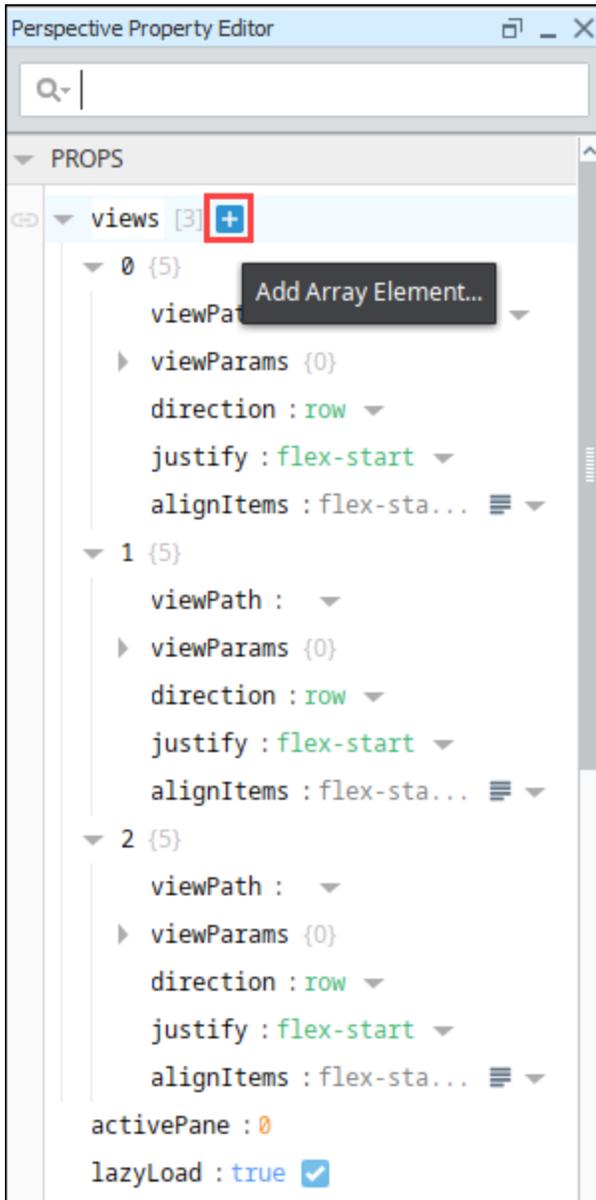


6. Under viewParams, click the **Add** icon then choose **value**.
7. Replace key with **Motor_Num** and replace value with **1**.

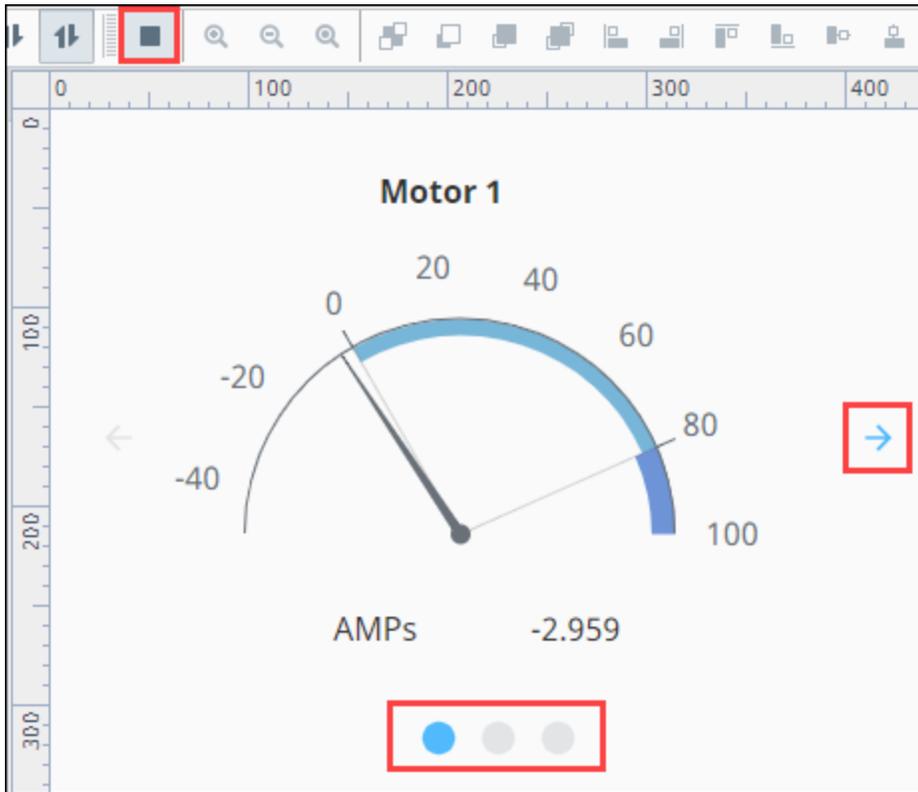
8. Click **OK**. The Motor view now appears in the Carousel component. You may need to expand your Carousel component slightly to fit the Motor view in without scrollbars.



9. Next, we'll add two more views. Click on the **Add +** icon two times for the Property Editor's views property.



10. Under the props.views.1.viewPath property, choose the **Motor** view.
 - a. Under viewParams, click the **Add**  icon then choose value.
 - b. Replace key with **Motor_Num** and replace value with **2**. This will point this instance to the Motor 2/AMPS Tag.
 - c. Click **OK**.
11. Under the props.views.2.viewPath property, choose the **Motor** view.
 - a. Under viewParams, click the **Add**  icon then choose value.
 - b. Replace key with **Motor_Num** and replace value with **3**. This will point this instance to the Motor 3/AMPS Tag.
 - c. Click **OK**.
12. Save your project.
13. Put the Designer into Preview mode. Click the left and right arrows or the dots to scroll between the three Motor views.



14. Lastly, we decided to change a few properties on the Carousel to update the appearance. Here are the settings we used:

Property	Value
props.appearance.dots.enabled	false
props.appearance.arrows.next.iconPath	material/navigate_next
props.appearance.arrows.next.fillColor	#AC00AC
props.appearance.arrows.previous.iconPath	material/navigate_before
props.appearance.arrows.previous.fillColor	#AC00AC
props.style.borderStyle	outset
props.style.borderColor	#AC00AC
props.style.borderWidth	7
props.style.borderTopLeftRadius	15
props.style.borderTopRightRadius	15
props.style.borderBottomLeftRadius	15
props.style.borderBottomRightRadius	15

Perspective - Embedded View



Component Palette Icon:



On this page ...

- [Properties](#)
- [Component Events](#)
- [Example](#)

The Embedded view component allows you to include an entire view inside another. Using this component allows you to select a view to display, and to pass parameters into the view. Because of this, views can easily act as templates for information.

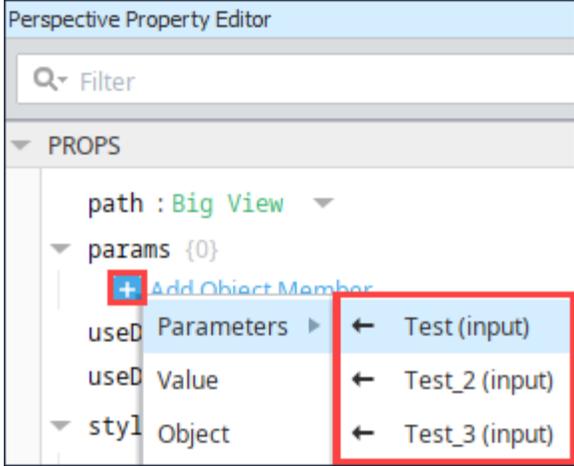
For example, you could create a tank view and embed several into another, larger view that shows an overview of the facility.

The embedded view is different than a container because you cannot alter the contents of a view using the Embedded View. A new container would allow you to create a new grouped set of components.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
path	<p>Path of the view to load in wrapper.</p> <p>The following feature is new in Ignition version 8.1.29 Click here to check out the other new features</p> <p>If a path is present in the path property field, an Open View  icon will appear that will navigate directly to the view when clicked.</p>	value: string
params	<p>Parameters for the view. If passing parameters into the embedded view, the names here must match the parameters on that view.</p> <p>The following feature is new in Ignition version 8.1.4 Click here to check out the other new features</p> <p>As of 8.1.4 a dropdown list of parameters is available when the user clicks the Add Object Member  icon. This makes it easy to add parameters from the embedded view. See also Embedded Views.</p>	object

								
useDefaultViewWidth	Use of view's default width instead of adjusting based on the content's width.	value: boolean						
useDefaultViewHeight	Use of view's default height instead of adjusting based on the content's width.	value: boolean						
loading	<p>The following feature is new in Ignition version 8.1.5 Click here to check out the other new features</p> <p>View loading settings.</p> <table border="1" data-bbox="250 1066 1339 1333"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>order</td> <td>Controls when the embedded view loads: alongside the parent view loading (<i>with-parent</i>), or after the parent view finishes loading (<i>after-parent</i>). Default is <i>with-parent</i>. Generally speaking, <i>with-parent</i> is more efficient for the browser, but in some cases can feel slower overall. Alternatively, <i>after-parent</i> is generally less efficient for the browser and can add to the overall load time. However, since it allows the parent view to load first, <i>after-parent</i> may feel quicker since the topmost layer of views get started up sooner.</td> <td>value: boolean</td> </tr> </tbody> </table>	Name	Description	Property Type	order	Controls when the embedded view loads: alongside the parent view loading (<i>with-parent</i>), or after the parent view finishes loading (<i>after-parent</i>). Default is <i>with-parent</i> . Generally speaking, <i>with-parent</i> is more efficient for the browser, but in some cases can feel slower overall. Alternatively, <i>after-parent</i> is generally less efficient for the browser and can add to the overall load time. However, since it allows the parent view to load first, <i>after-parent</i> may feel quicker since the topmost layer of views get started up sooner.	value: boolean	object
Name	Description	Property Type						
order	Controls when the embedded view loads: alongside the parent view loading (<i>with-parent</i>), or after the parent view finishes loading (<i>after-parent</i>). Default is <i>with-parent</i> . Generally speaking, <i>with-parent</i> is more efficient for the browser, but in some cases can feel slower overall. Alternatively, <i>after-parent</i> is generally less efficient for the browser and can add to the overall load time. However, since it allows the parent view to load first, <i>after-parent</i> may feel quicker since the topmost layer of views get started up sooner.	value: boolean						
style	Sets a style for this component. Full menu of style options is available. You can also specify a style class .	object						

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

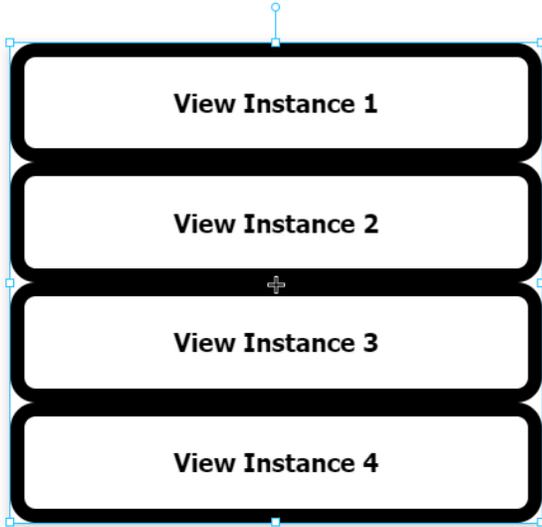
Example

The image displays a user interface layout. At the top left is a map component showing a street map with a blue dot labeled 'Valve 1'. To its right is a control panel with a valve icon and buttons for 'Hand', 'Off', and 'Auto'. Below these is an 'Embedded View' component, which is a chart titled 'Pressure Trending - Overview' showing a fluctuating line graph with a green dot, a pink square, and a red triangle. A red arrow points to the 'Embedded View' label.

In this example, there is a Map component and a Carousel component on the top of the page. Underneath them, we've placed an Embedded View component. The idea was to embed this overview to give users a quick visual reference to bigger picture trends for the site. This example assumes you have a view already created named "Pressure Trend Overview". Here are the properties for just the Embedded View:

Property	Value	Style Category
props.path	Pressure Trend Overview	N/A
props.style.backgroundColor	#FFE8CC	background
props.style.borderStyle	groove	border
props.style.borderWidth	6px	border

Perspective - Flex Repeater



On this page ...

- [Properties](#)
- [User Interface](#)
- [Component Events](#)
- [Example](#)

Component Palette Icon:



The Flex Repeater component lets you easily create multiple instances of views for display in another view.

When first dropped on a view, the Flex Repeater looks like any other empty container. Set the **path** to the component that you want to create multiple instances of, and then under **instances** add an object for each instance that you want to create. The **object** will usually contain one or more parameters, including the instance's own **index**, that will be passed into that particular instance. As a side note, overwriting the index parameter is not recommended.

The flex repeater is functionally very similar to the [Flex Container component](#). Both components are based off of the [CSS flexbox](#), and both abide by similar rules in regards to how child objects are positions. However the Flex Repeater differs from the contain in two notable ways:

1. The Flex Repeater can only have embedded views as direct children, where as the Flex Container can have an type of component.
2. The Flex Repeater can create instances of views from the runtime, by adding additional elements to the **instances** array. The flex container does not have anything resembling this functionality.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
path	<p>Path to the desired view to display.</p> <div style="border: 1px solid orange; padding: 5px; margin: 10px 0;"><p>The following feature is new in Ignition version 8.1.29 Click here to check out the other new features</p></div> <p>If a path is present in the path property field, an Open View  icon will appear that will navigate directly to the view when clicked.</p>	value: string
instances	<p>Number of instances of the view that you want to display in the container. Each instance can contain an instanceStyle and instancePosition property. Changing these properties will override any styles and position settings applied by elementStyle and elementPosition.</p> <p>This is where a value property can be added that matches up with a parameter in the view to pass in a value.</p>	array

Name	Description	Property Type
instanceStyle	Sets a style for this instance of a view. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object
instancePosition	Position properties such as grow, shrink, or basis that would apply to all instances.	object

The following feature is new in Ignition version **8.1.4**
[Click here](#) to check out the other new features

As of 8.1.4 a dropdown list of parameters is available when the view being displayed has view parameters. This makes it easy to add, delete, or synchronize parameters from that view.

direction	Direction of layout of repeated views. Options are row, row-reverse, column and column-reverse.	value: string												
wrap	Whether the container should allow instances to wrap to the next line if space has run out. Options are nowrap, wrap, wrap-reverse.	value: string												
justify	Adjusts placement of instances along the main axis when there is extra space, which may be used to fill areas before, after, or in-between: flex-start, flex-end, center, space-between, space-around, space-evenly.	value: string												
alignItems	Adjusts placement of instances along the cross axis when there is extra space: flex-start, flex-end, center, baseline, stretch.	value: string												
alignContent	Adjusts alignment of wrapped content when there is free space in the cross axis: flex-start, flex-end, center, space-between, space-around, stretch.	value: string												
useDefaultViewWidth	Use view's default width instead of adjusting based on the content's width.	value: boolean												
useDefaultViewHeight	Use view's default height instead of adjusting based on the content's height.	value: boolean												
elementStyle	Sets a style for this element. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object												
elementPosition	Sets a position for this element. Element position properties that apply to all instances, unless overridden by instancePosition.	object												
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>grow</td> <td>grow and shrink control the way that a component responds to changes in the flex container's width or height. For columns row controls what happens when additional space is available.</td> <td>value: numeric</td> </tr> <tr> <td>shrink</td> <td>shrink controls what happens when the component does not have enough space to fulfill its basis.</td> <td>value: numeric</td> </tr> <tr> <td>basis</td> <td>Controls the default size of a component along the flex repeater's direction. You can enter the value in pixels (e.g. 75px), as a percentage of the total length of the container (e.g. 50%), or you can use auto. All components configured to auto will equally share the available space in the container.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	grow	grow and shrink control the way that a component responds to changes in the flex container's width or height. For columns row controls what happens when additional space is available.	value: numeric	shrink	shrink controls what happens when the component does not have enough space to fulfill its basis.	value: numeric	basis	Controls the default size of a component along the flex repeater's direction. You can enter the value in pixels (e.g. 75px), as a percentage of the total length of the container (e.g. 50%), or you can use auto. All components configured to auto will equally share the available space in the container.	value: numeric	
Name	Description	Property Type												
grow	grow and shrink control the way that a component responds to changes in the flex container's width or height. For columns row controls what happens when additional space is available.	value: numeric												
shrink	shrink controls what happens when the component does not have enough space to fulfill its basis.	value: numeric												
basis	Controls the default size of a component along the flex repeater's direction. You can enter the value in pixels (e.g. 75px), as a percentage of the total length of the container (e.g. 50%), or you can use auto. All components configured to auto will equally share the available space in the container.	value: numeric												

loading	<p>The following feature is new in Ignition version 8.1.5 Click here to check out the other new features</p> <p>View loading settings.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>order</td> <td>Controls when the embedded views load: alongside the parent view loading (<code>with-parent</code>), or after the parent view finishes loading (<code>after-parent</code>). Default is after-parent.</td> <td>value: boolean</td> </tr> </tbody> </table>	Name	Description	Property Type	order	Controls when the embedded views load: alongside the parent view loading (<code>with-parent</code>), or after the parent view finishes loading (<code>after-parent</code>). Default is after-parent.	value: boolean	object
Name	Description	Property Type						
order	Controls when the embedded views load: alongside the parent view loading (<code>with-parent</code>), or after the parent view finishes loading (<code>after-parent</code>). Default is after-parent.	value: boolean						

Generally speaking, `with-parent` is more efficient for the browser, but in some cases can feel slower overall. Alternatively, `after-parent` is generally less efficient for the browser and can add to the overall load time. However, since it allows the parent view to load first, `after-parent` may feel quicker since the topmost layer of views get started up sooner.

style

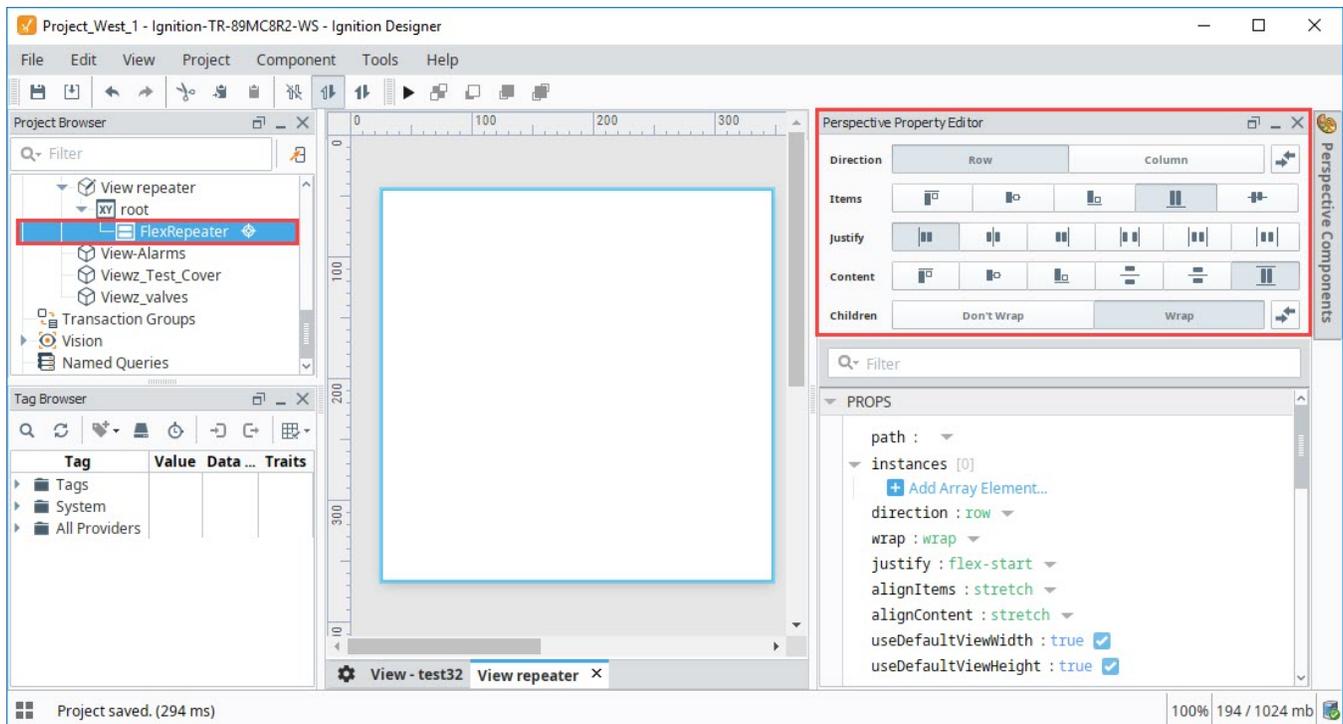
Sets a style for the Flex Repeater. Full menu of [style options](#) is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a [style class](#).

object

Name	Description	Property Type
classes	Sets a style class for the Flex Repeater.	object
overflow	Options are auto, visible, scroll, or hidden.	value: string

User Interface

When a Flex Repeater is deep selected, there is an interface at the top of the Perspective Property Editor that enables you to set the container's properties. Functionality is similar to that of the Flex Container component. See [Perspective - Flex Container](#).

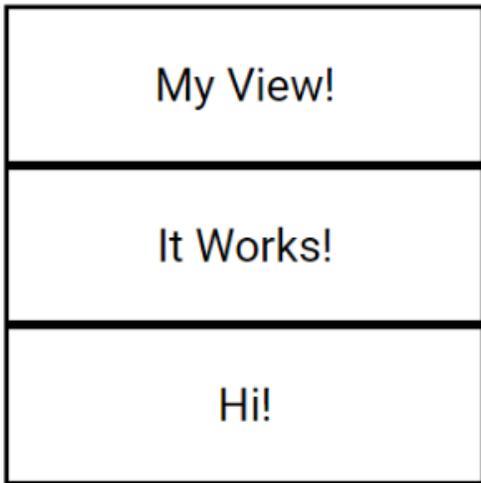


Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Example



1. In order to use the Flex Repeater, you need a view that will be shown inside the repeater. To do this, we made a new View with a Coordinate layout called "RepeatedView".

- a. In the Project Browser, select the new View
- b. In the Property Editor. Under **PARAMS**, select **Add View Parameter**, select a property type of **Value**.
- c. Rename the parameter from "key" to "labelText".
- d. Select the **root** container for the view, and set the **mode** property to "percent".

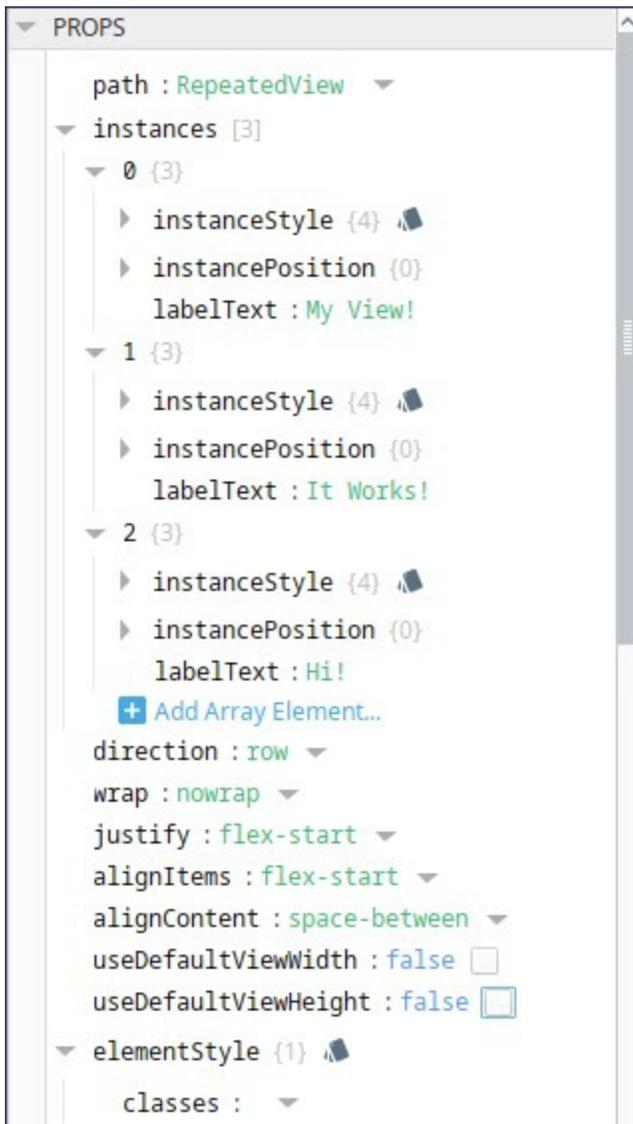
2. Add a Label component to the view, and configure a property binding on its **text** property to the **labelText** parameter we just created. We set the component's **alignVertical** property to "center", and we stretched the label to fill the entire view. We also configured some styling on the Label:

Style Category	Value
borderStyle	solid
fontSize	30px
textAlign	center

3. Now we can configure our Flex Repeater. Drag a Flex Repeater component onto the view, then set the following properties:

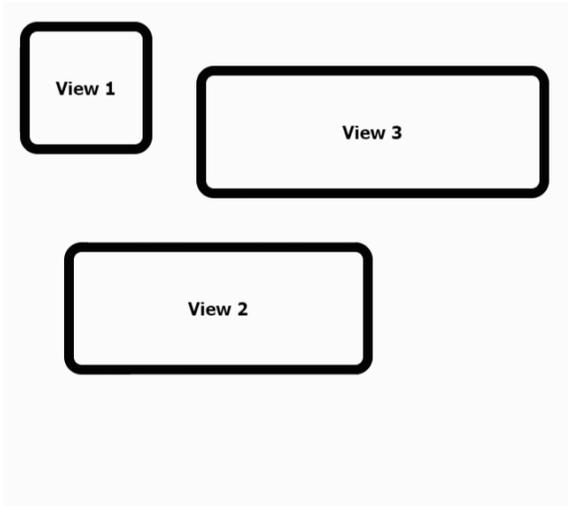
Property Name	Value
path	RepeatedView
direction	Column
useDefaultViewWidth	false
useDefaultViewHeight	false

4. Finally, create three **object** members in the **instances** array. Add a **labelText** property to each object of type **value** , and replace the value strings to the desired strings to show. Here's how the property editor looks for our Flex Repeater:



5. You should now see the Flex repeater populated as shown in the image above.

Perspective - View Canvas



On this page ...

- [Properties](#)
- [Scripting](#)

Component Palette Icon:



The View Canvas component can display multiple Perspective views, each positioned on a coordinate based system. The component offers smooth transition animations when views are relocated. Familiarity with CSS is helpful in taking full advantage of the this component.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Prope Type																								
instanc es	Array of views to display in the canvas.	array																								
	<table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>position</td><td>Mode that defines how the element is positioned within the canvas. For position absolute, the view is placed within the canvas based on its top, left, bottom, right positions. Views that are positioned absolute do not participate in the flow of the document. For position relative, the view is placed within the canvas placed in the normal document flow and then offset by its top, left values. This is the same with left and right. Options are relative or absolute. Default is absolute.</td><td>value: string</td></tr><tr><td>top</td><td>The top position of the view.</td><td>value: numeric</td></tr><tr><td>left</td><td>The left position of the view.</td><td>value: numeric</td></tr><tr><td>bottom</td><td>The bottom position of the view. Note: If both top and bottom are set, bottom is respected only if position is set to absolute and height is unspecified.</td><td>value: numeric</td></tr><tr><td>right</td><td>The right position of the view. Note: If both left and right are set, left is respected only if position is set to absolute and width is unspecified.</td><td>value: numeric</td></tr><tr><td>zIndex</td><td>The z-order position of the view.</td><td>value: numeric</td></tr><tr><td>width</td><td>The width of the view.</td><td>value: numeric</td></tr></tbody></table>	Name	Description	Property Type	position	Mode that defines how the element is positioned within the canvas. For position absolute, the view is placed within the canvas based on its top, left, bottom, right positions. Views that are positioned absolute do not participate in the flow of the document. For position relative, the view is placed within the canvas placed in the normal document flow and then offset by its top, left values. This is the same with left and right. Options are relative or absolute. Default is absolute.	value: string	top	The top position of the view.	value: numeric	left	The left position of the view.	value: numeric	bottom	The bottom position of the view. Note: If both top and bottom are set, bottom is respected only if position is set to absolute and height is unspecified.	value: numeric	right	The right position of the view. Note: If both left and right are set, left is respected only if position is set to absolute and width is unspecified.	value: numeric	zIndex	The z-order position of the view.	value: numeric	width	The width of the view.	value: numeric	
Name	Description	Property Type																								
position	Mode that defines how the element is positioned within the canvas. For position absolute, the view is placed within the canvas based on its top, left, bottom, right positions. Views that are positioned absolute do not participate in the flow of the document. For position relative, the view is placed within the canvas placed in the normal document flow and then offset by its top, left values. This is the same with left and right. Options are relative or absolute. Default is absolute.	value: string																								
top	The top position of the view.	value: numeric																								
left	The left position of the view.	value: numeric																								
bottom	The bottom position of the view. Note: If both top and bottom are set, bottom is respected only if position is set to absolute and height is unspecified.	value: numeric																								
right	The right position of the view. Note: If both left and right are set, left is respected only if position is set to absolute and width is unspecified.	value: numeric																								
zIndex	The z-order position of the view.	value: numeric																								
width	The width of the view.	value: numeric																								

height	The height of the view.	value: numeric									
viewPath	<p>Path to the view you want to display.</p> <div style="border: 1px solid orange; padding: 5px; margin: 5px 0;"> <p>The following feature is new in Ignition version 8.1.29 Click here to check out the other new features</p> </div> <p>If a path is present in the viewPath property field, an Open View  icon will appear that will navigate directly to the view when clicked.</p>	value: string									
viewParams	<p>The parameters of the view.</p> <div style="border: 1px solid orange; padding: 5px; margin: 5px 0;"> <p>The following feature is new in Ignition version 8.1.4 Click here to check out the other new features</p> </div> <p>As of 8.1.4 a dropdown list of parameters is available when the user clicks the AddObject Member  icon. This makes it easy to add parameters from the rendered view.</p>	object									
style	Sets a style for this view. Full menu of style options is available. You can also specify a style class .	object									
transitionSettings	<p>Transition settings on each view. The properties affected by transition settings are top, left, bottom, right, and zIndex.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Name</th> <th style="width: 60%;">Description</th> <th style="width: 25%;">Property Type</th> </tr> </thead> <tbody> <tr> <td>duration</td> <td>Duration of the transition. Units are seconds or milliseconds.</td> <td>value: numeric</td> </tr> <tr> <td>timingFunction</td> <td>Mathematical function that defines how fast one-dimensional values change during the transition. The transition can be described as a cubic Bezier or steps function. The presets for cubic Bezier functions are linear, ease, ease-in, ease-in-out, and ease-out. The presets for steps functions are step-start and step-end.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	duration	Duration of the transition. Units are seconds or milliseconds.	value: numeric	timingFunction	Mathematical function that defines how fast one-dimensional values change during the transition. The transition can be described as a cubic Bezier or steps function. The presets for cubic Bezier functions are linear, ease, ease-in, ease-in-out, and ease-out. The presets for steps functions are step-start and step-end.	value: string	value: numerical
Name	Description	Property Type									
duration	Duration of the transition. Units are seconds or milliseconds.	value: numeric									
timingFunction	Mathematical function that defines how fast one-dimensional values change during the transition. The transition can be described as a cubic Bezier or steps function. The presets for cubic Bezier functions are linear, ease, ease-in, ease-in-out, and ease-out. The presets for steps functions are step-start and step-end.	value: string									
enableTransitions	Determines whether transitions should play when transitions are defined.	value: boolean									
defaultStyle	Sets a style for all views. Full menu of style options is available. You can also specify a style class .	object									
style	Sets a style for the canvas. Full menu of style options is available. You can also specify a style class .	object									

Scripting

See the [Perspective - View Canvas Scripting page](#) for the full list of scripting functions available for this component.

Perspective - View Canvas Scripting

This page details the various scripting, component, and extension functions available for [Perspective's View Canvas](#) component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

On this page ...

- [Component Events](#)
 - [onInstanceClicked](#)
- [Component Functions](#)
- [Extension Functions](#)

onInstanceClicked

Event is fired when a view instance is clicked.

event.index

- Object Path

event.index

- Type

[Number](#)

- Description

The index of the view instance.

event.params

- Object Path

event.params

- Type

[Dictionary](#)

- Description

The position of the view instance in relation to the canvas.

event.path

- Object Path

event.path

- Type

[String](#)

- Description

The path of the view instance.

event.position

- Object Path

event.position

- Type

[JSON Object](#)

- Description

A JSON Object representing the current position values.

event.position.top

- Object Path

event.position.top

- Type

Number

- Description

The top position of the view instance.

event.position.left

- Object Path

event.position.left

- Type

Number

- Description

The left position of the view instance.

event.position.bottom

- Object Path

event.position.bottom

- Type

Number

- Description

The bottom position of the view instance.

event.position.right

- Object Path

event.position.right

- Type

Number

- Description

The right position of the view instance.

event.size

- Object Path

event.size

- Type

JSON Object

- Description

A JSON Object representing the current size.

event.size.width

- Object Path

event.size.width

- Type

Number

- Description

The width of the view instance.

event.size.height

- Object Path

event.size.height

- Type

Number

- Description

The height of the view instance.

Component Functions

This component does not have component functions associated with it.

Extension Functions

This component does not have extension functions associated with it.

Perspective - Input Palette

Input Components

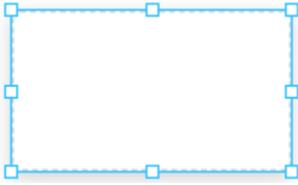
Perspective provides a host of Input components that allow users to enter or select data, and even control a device.

Here is a complete list of Input components, and a link pointing to a page containing the component's description, properties, and an example of how to configure it.



[In This Section ...](#)

Perspective - Barcode Scanner Input



Component Palette Icon:



On this page ...

- [Properties](#)
- [Component Events](#)

The Barcode Scanner Input component awaits for input from a barcode scanner. The component was designed for keyboard wedge scanners, as the component provides dedicated prefix and suffix properties to define scanner input. As such, it can be useful to think of the Barcode Scanner Input component as a specialized text field that does not require focus, and uses characters to decide when to accept and reject text input.

The scanner component is continuously listening, waiting for either the prefix and suffix characters to be entered, or the regex pattern to find a match. Once triggered, the component will load the scanned barcode string (excluding the prefix and suffix) into the data property for processing. The regex property can be used to extract specific fields from a scan, or validate data from the scan.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
prefix	String value that mark the start of the barcode scan capture. If a value is provided to this property, then the regex property will be ignored.	value: string
suffix	String value that marks the end of the barcode scan capture. If a value is provided to this property, then the regex property will be ignored.	value: string
regex	Regex describing the format of scans. The first capture will be used as barcode. When used, this property will pull out the first group of any regex provided. Note that this setting uses Javascript regex, as opposed to other flavors of regex. If either a prefix or suffix value is specified with a non-empty string, then this property will be ignored.	object
window	Length of buffer to monitor for regex match.	value: numeric
capture Mode	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 5px;">The following feature is new in Ignition version 8.1.16 Click here to check out the other new features</div> Indicates which key event the component will listen for to start the barcode scan capture. Values include <code>keypress</code> , <code>keyup</code> , and <code>keydown</code> . Default value is <code>keypress</code> .	value: string
data	Barcode scans returned from scanner.	array
dataStyle	Sets a style for data returned to this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object
style	Sets a style for this component. Full menu of style options is available text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class . Styles can can be set on the component before a value is scanned.	object

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Perspective - Button



Component Palette Icon:



On this page ...

- [Properties](#)
- [Component Events](#)
- [Examples](#)
 - [Example 1](#)
 - [Example 2](#)

The Button component is a versatile component, that is used to initiate some sort of action in response to being pressed. It can be used for showing status, as well. For example, you can configure buttons to be active or inactive, change color, text or any other property, and you can alter these configurations in response to conditions in your project. Button components support icons as well. For an example, see Example 2 below.

To get buttons to do things, you configure one or more Actions that occur following an Event. For instance, you might call a [Script action](#) on the onActionPerformed component event, which triggers when the button is pressed.

The following feature is new in Ignition version **8.1.2**
[Click here](#) to check out the other new features

The Button component has two pre-configured [variants](#): Primary and Secondary.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type						
text	<p>Text to display on a button.</p> <p>The following feature is new in Ignition version 8.1.26 Click here to check out the other new features</p> <p>Text can also be entered directly on the button by deep selecting the component, which enables inline editing. Changes are immediately reflected in the text property field.</p>	value: string						
textStyle	Style properties that are directly applied to the text within the Button component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object						
primary	Toggle between default primary and secondary button style. Default is true.	value: boolean						
enabled	Enables button interaction. Default is true.	value: boolean						
image	An optional image embedded in the button.	object						
	<table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>source</td><td><p>The image source URL. It could be a URL to an image on the internet or Gateway, or even an embedded image.</p><p>For an image already in the Image Management console, use /system/images/{path to your image}. For example:</p><p><code>/system/images/Builtin/icons/24/lightbulb_on.png</code></p></td><td>value: string</td></tr></tbody></table>	Name	Description	Property Type	source	<p>The image source URL. It could be a URL to an image on the internet or Gateway, or even an embedded image.</p> <p>For an image already in the Image Management console, use /system/images/{path to your image}. For example:</p> <p><code>/system/images/Builtin/icons/24/lightbulb_on.png</code></p>	value: string	
Name	Description	Property Type						
source	<p>The image source URL. It could be a URL to an image on the internet or Gateway, or even an embedded image.</p> <p>For an image already in the Image Management console, use /system/images/{path to your image}. For example:</p> <p><code>/system/images/Builtin/icons/24/lightbulb_on.png</code></p>	value: string						

icon	An image path used to augment the writingState of the component by placing an image next to it.	object									
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Shorthand path to icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons.</td> <td>value: string</td> </tr> <tr> <td>color</td> <td>Color of the icon. Can also assign color in "fill" of the style property . Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>color</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Shorthand path to icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string	color	Color of the icon. Can also assign color in "fill" of the style property . Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color	
Name	Description	Property Type									
path	Shorthand path to icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string									
color	Color of the icon. Can also assign color in "fill" of the style property . Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color									
width	Width of the button image in pixels.	value: numeric									
height	Height of the button image in pixels.	value: numeric									
position	Horizontal position of the image within the button relative to the text: left, center, right, top, or bottom.	value: string									
style	Sets a style for the image. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object									
align	Aligns the text and image (if present) along the cross axis. Vertical if imagePosition is top or bottom, otherwise it's horizontal. Options are start, center, end, and stretch. Default is center.	value: string									
justify	Justifies the text and image (if present) along the main axis. Horizontal if the imagePosition is top or bottom, otherwise it's vertical. Options are start, center, end, space-around, space-between, and space-evenly. Default is center.	value: string									
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object									

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Examples

Example 1



Property	Value
props.text	Complete
props.image.source	/system/images/Builtin/icons/48/check2.png
props.image.position	right
props.textStyle.color	#000000
props.justify	space-evenly
props.style.backgroundColor	#D5D5D5
props.style.borderColor	solid

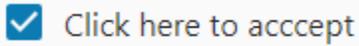
props.style.borderColor	#000000
props.style.borderWidth	2px

Example 2



Property	Value
props.text	Save
props.textStyle.color	#00AC00
props.image.source	/system/images/Builtin/icons/48/disk_green.png
props.image.position	top
props.image.width	40
props.image.height	40
props.align	end
props.justify	space-between
props.style.backgroundColor	#FFFFFF
props.style.borderStyle	inset
props.style.borderWidth	5px
props.style.borderColor	#00AC00

Perspective - Checkbox



Component Palette Icon:



On this page ...

- [Properties](#)
- [Component Events](#)
- [Examples](#)
 - [Example 1](#)
 - [Example 2](#)
 - [Example 3](#)

A Checkbox is a familiar component that represents a bit - it is either on (selected) or off (not selected). In addition, the 'triState' property can be enabled, adding a third state to represent an indeterminate value. It is functionally equivalent to the Toggle Switch component.

The following feature is new in Ignition version **8.1.2**
[Click here](#) to check out the other new features

The Checkbox component has two pre-configured [variants](#): Text Right and Text left.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type									
selected	Output value for checkbox.	value: boolean									
text	Label for the checkbox. <p>The following feature is new in Ignition version 8.1.26 Click here to check out the other new features</p> Label text can also be entered directly by deep selecting the checkbox component, which enables inline editing. Changes are immediately reflected in the text property field.	value: string									
textPosition	Where to place the label text in relation to the checkbox: top, right, bottom, or left.	value: string									
enabled	Whether the user can currently interact with the checkbox. Note: If the component is disabled, scripts can still run on the component. For example, if you add a script action to the onClick event, the script will fire when the user clicks on the Checkbox.	value: boolean									
triState	Whether the checkbox supports a third state of "indeterminate" - effectively 'null' or 'no choice'.	value: boolean									
checkedIcon	Settings for the appearance of the check box's icon when it is selected (checked).	object									
	<table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>path</td><td>Path to the icon source, in format: library/IconName. For more information on icons, see the Images, and Icons in Perspective page.</td><td>value: string</td></tr><tr><td>color</td><td>Settings for the fill color for the checked icon.</td><td>object</td></tr></tbody></table>	Name	Description	Property Type	path	Path to the icon source, in format: library/IconName. For more information on icons, see the Images, and Icons in Perspective page.	value: string	color	Settings for the fill color for the checked icon.	object	
Name	Description	Property Type									
path	Path to the icon source, in format: library/IconName. For more information on icons, see the Images, and Icons in Perspective page.	value: string									
color	Settings for the fill color for the checked icon.	object									

	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Color of the icon when enabled. Can be a named color.</td> <td>value: string</td> </tr> <tr> <td>disabled</td> <td>Color of the icon when disabled. Can be a named color.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Color of the icon when enabled. Can be a named color.	value: string	disabled	Color of the icon when disabled. Can be a named color.	value: string													
Name	Description	Property Type																					
enabled	Color of the icon when enabled. Can be a named color.	value: string																					
disabled	Color of the icon when disabled. Can be a named color.	value: string																					
	<table border="1"> <tr> <td>style</td> <td>Sets a style for the icon. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </table>	style	Sets a style for the icon. Full menu of style options is available. You can also specify a style class .	object																			
style	Sets a style for the icon. Full menu of style options is available. You can also specify a style class .	object																					
uncheckedIcon	<p>Settings for the appearance of the check box's icon when it is not selected.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Path to the icon source, in format: library/IconName. For more information on icons, see the Images, and Icons in Perspective page.</td> <td>value: string</td> </tr> <tr> <td>color</td> <td>Settings for the fill color for the unchecked icon. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Color of the icon when enabled. Can be a named color.</td> <td>value: string</td> </tr> <tr> <td>disabled</td> <td>Color of the icon when disabled. Can be a named color.</td> <td>value: string</td> </tr> </tbody> </table> </td> <td>object</td> </tr> <tr> <td>style</td> <td>Sets a style for the icon. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table> <p>Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</p>	Name	Description	Property Type	path	Path to the icon source, in format: library/IconName. For more information on icons, see the Images, and Icons in Perspective page.	value: string	color	Settings for the fill color for the unchecked icon. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Color of the icon when enabled. Can be a named color.</td> <td>value: string</td> </tr> <tr> <td>disabled</td> <td>Color of the icon when disabled. Can be a named color.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Color of the icon when enabled. Can be a named color.	value: string	disabled	Color of the icon when disabled. Can be a named color.	value: string	object	style	Sets a style for the icon. Full menu of style options is available. You can also specify a style class .	object	object
Name	Description	Property Type																					
path	Path to the icon source, in format: library/IconName. For more information on icons, see the Images, and Icons in Perspective page.	value: string																					
color	Settings for the fill color for the unchecked icon. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Color of the icon when enabled. Can be a named color.</td> <td>value: string</td> </tr> <tr> <td>disabled</td> <td>Color of the icon when disabled. Can be a named color.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Color of the icon when enabled. Can be a named color.	value: string	disabled	Color of the icon when disabled. Can be a named color.	value: string	object												
Name	Description	Property Type																					
enabled	Color of the icon when enabled. Can be a named color.	value: string																					
disabled	Color of the icon when disabled. Can be a named color.	value: string																					
style	Sets a style for the icon. Full menu of style options is available. You can also specify a style class .	object																					
indeterminateIcon	<p>Settings for the appearance of the check box's icon when it is in the indeterminate state. Only applies if props.triState is set to true.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Path to the icon source, in format: library/IconName. For more information on icons, see the Images, and Icons in Perspective page.</td> <td>value: string</td> </tr> <tr> <td>color</td> <td>Settings for the fill color for the icon. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Color of the icon when enabled. Can be a named color.</td> <td>value: string</td> </tr> <tr> <td>disabled</td> <td>Color of the icon when disabled. Can be a named color.</td> <td>value: string</td> </tr> </tbody> </table> </td> <td>object</td> </tr> <tr> <td>style</td> <td>Sets a style for the icon. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Path to the icon source, in format: library/IconName. For more information on icons, see the Images, and Icons in Perspective page.	value: string	color	Settings for the fill color for the icon. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Color of the icon when enabled. Can be a named color.</td> <td>value: string</td> </tr> <tr> <td>disabled</td> <td>Color of the icon when disabled. Can be a named color.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Color of the icon when enabled. Can be a named color.	value: string	disabled	Color of the icon when disabled. Can be a named color.	value: string	object	style	Sets a style for the icon. Full menu of style options is available. You can also specify a style class .	object	object
Name	Description	Property Type																					
path	Path to the icon source, in format: library/IconName. For more information on icons, see the Images, and Icons in Perspective page.	value: string																					
color	Settings for the fill color for the icon. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Color of the icon when enabled. Can be a named color.</td> <td>value: string</td> </tr> <tr> <td>disabled</td> <td>Color of the icon when disabled. Can be a named color.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Color of the icon when enabled. Can be a named color.	value: string	disabled	Color of the icon when disabled. Can be a named color.	value: string	object												
Name	Description	Property Type																					
enabled	Color of the icon when enabled. Can be a named color.	value: string																					
disabled	Color of the icon when disabled. Can be a named color.	value: string																					
style	Sets a style for the icon. Full menu of style options is available. You can also specify a style class .	object																					
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																					

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Examples

Example 1

Checkbox 1

Property	Value
props.text	Checkbox 1
props.textPosition	left
props.triState	true

Example 2

Checkbox 2

Property	Value
props.text	Checkbox 2
props.textPosition	left
props.selected	null

Example 3

Checkbox 3

Property	Value
props.text	Checkbox 3
props.textPosition	top
props.enabled	false
props.selected	false

Perspective - DateTime Input

02/25/2018 10:55 pm

Component Palette Icon:



On this page ...

- [Properties](#)
- [Component Events](#)
- [Example](#)

A DateTime Input is an easy way to select a date from a popup calendar. Similar to the DateTime Picker component, it takes up much less real estate on the screen. Configure the date and time format in the Property Editor using the 'formattedValue' property.

The following feature is new in Ignition version **8.1.2**
[Click here](#) to check out the other new features

The DateTime Input component has two pre-configured [variants](#):

- Date and Time - Opens a calendar from which users can select a date and time.
- Time - Enables users to set a time using the up and down arrows on the component.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
value	Current date/time as a Date object or timestamp in milliseconds.	value: string
formattedValue	Date and time in configured format.	value: string
inputProps	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous.	object
modalStyle	Style applied to the Date picker modal (popup). Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object
pickerType	Whether to display and enable picker for date only, time only, or both date and time.	value: string
minDate	Minimum date/time as a Date object or timestamp in milliseconds. If null, the minimum date is 10 years in the past from today.	value: string
maxDate	Maximum date/time as a Date object or timestamp in milliseconds. If null, the maximum date is 10 years in the future from today	value: string
format	Template for formatting date display - must be valid moment.js format, e.g., 'MM/DD/YYYY h:mm a'.	value: string
enabled	'False' will disable any interaction with the calendar. Note: If the component is disabled, scripts can still run on the component. For example, if you add a script action to a System Event, such as an onStartUp event, the script will fire when the page is loaded. Events that require user interaction, such as onClick events, will not fire with the exception of Pointer Events.	value: boolean
placeholder	Text for input field to display when no date/time is selected.	value: boolean
locale	Code for localization of language and formatting. Use the dropdown to select language.	value:

		string
dismissOnSelect	Determines if the date picker should be dismissed when a date is selected.	value: boolean
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object

Component Events

Perspective Component Events

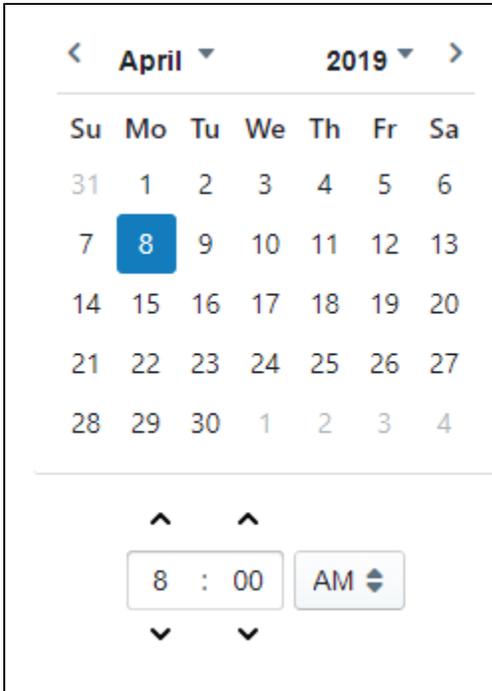
The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Example



Property	Value	Style Category
props.pickerType	date	N/A
props.format	MM/DD/YYYY	N/A
props.style.borderStyle	solid	border
props.style.borderColor	#00AC00	border
props.style.borderWidth	2px	border

Perspective - DateTime Picker



On this page ...

- [User Interface](#)
- [Properties](#)
- [Component Events](#)
- [Example](#)

Component Palette Icon:



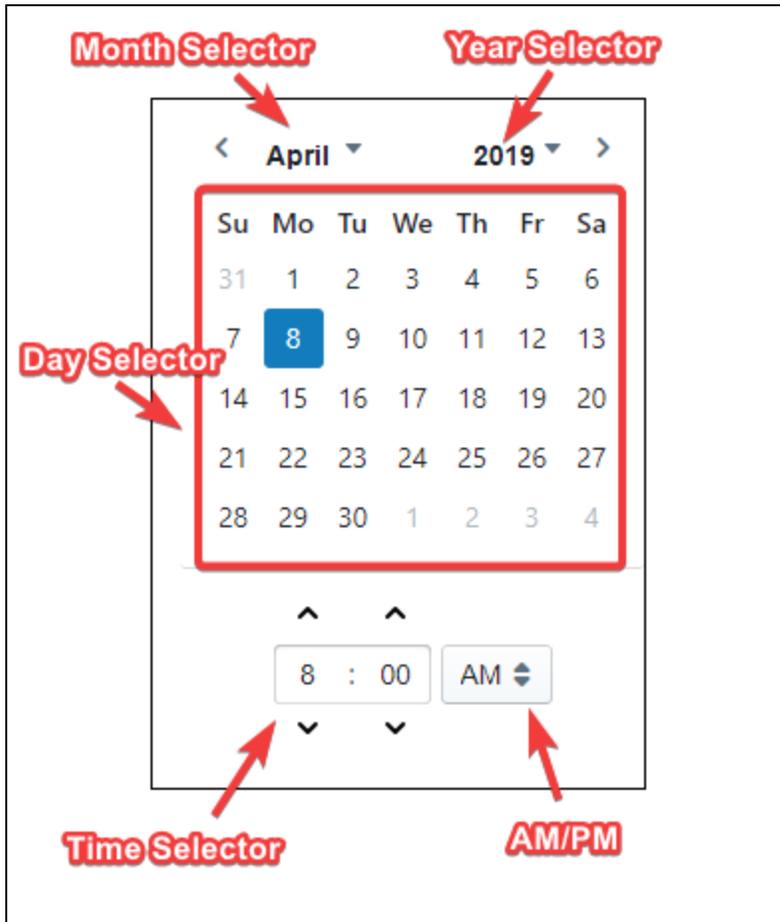
The DateTime Picker component uses the calendar to select the date and time. You can choose the "pickerType" to set both date and time, or just date. Configure the date and time format in the Property Editor using the formattedValue [property](#). To use the DateTime Picker, select the month, date, and time on the component.

The following feature is new in Ignition version **8.1.2**
[Click here](#) to check out the other new features

The DateTime Picker component has two pre-configured [variants](#):

- Date and Time - Enables users can select a date and time on a calendar.
- Date - Enables users can select a date on a calendar.

User Interface



Interaction	Description
AM/PM	Allows users to toggle between AM or PM.
Day Selector	Allows users to choose a specific day.
Month Selector	Allows users to choose a specific month.
Time Selector	Allows users to choose a specific time.
Year Selector	Allows users to choose a specific year.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
value	Current date/time as a Date object or timestamp in milliseconds.	value: dropdown
formattedValue	Date and time in configured format.	value: string
pickerType	Whether to display and enable picker for date only or for both date and time.	value: string
minDate	Minimum date/time as a Date object or timestamp in milliseconds. If null, the minimum date is 10 years in the past from today.	value: string

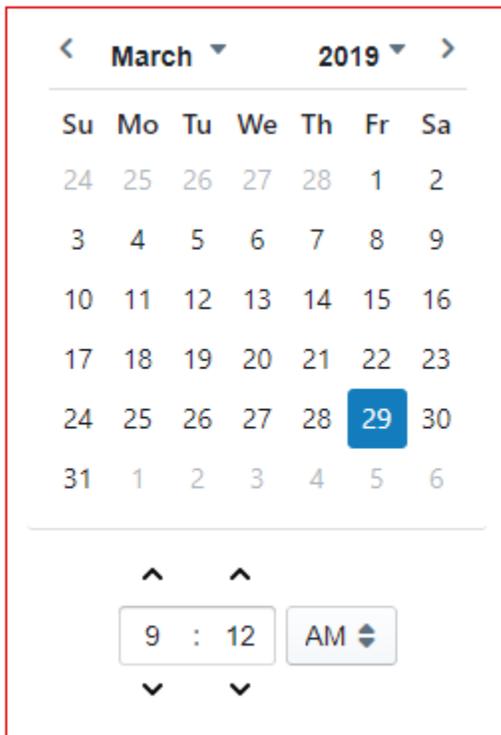
maxDate	Maximum date/time as a Date object or timestamp in milliseconds. If null, the minimum date is 10 years in the future from today.	value: string
format	Template for formatting date display - must be valid moment.js format, e.g., 'MM/DD/YYYY h:mm a'.	value: string
locale	Code for localization of language and formatting. Use the dropdown to select language.	value: string
enabled	'False' will disable any interaction with the calendar. Note: If the component is disabled, scripts can still run on the component. For example, if you add a script action to the onClick event, the script will fire when the user clicks on the DateTime Picker.	value: boolean
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Example



Property	Value	Style Category
props.format	MM/DD/YY hh:mm:ss a	N/A
props.style.borderStyle	solid	border
props.style.borderWidth	1px	border
props.style.borderColor	#D90000	border

Perspective - Dropdown



Component Palette Icon:



On this page ...

- [Properties](#)
- [Component Events](#)
- [Examples](#)
 - [Example 1](#)
 - [Example 2](#)

The Dropdown component is a great way to display a list of choices in a limited amount of space. The current selection is shown, and the choices are presented when the user clicks on the dropdown button. There is also the capability to search for an element by typing the name of that element in the dropdown field. If the element is present, it will appear as you are typing and you can select it. If the element doesn't exist, you can define text to display that the text is not found by configuring the **noResultsText** property.

The choices that are displayed in the Dropdown depend on what Elements are defined in the **options** property of the Property Editor. The **placeholder** property defines what text is shown before any of the choices are selected. For example, you can have the word 'Select...' to inform the user to select any of the Elements from the dropdown list.

There is also a **multiSelect** property which allows the user to select multiple elements from the dropdown. Selected elements can be deleted by clicking the 'x' icon.

The following feature is new in Ignition version **8.1.2**
[Click here](#) to check out the other new features

The Dropdown component has two pre-configured [variants](#):

- Single Selection - Default layout that displays a list of choices of which the user can select one.
- Multi-Selection - Layout with a list of choices of which the user can select more than one.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type									
value	The result of current selections (input) after any processing.	variable, based on which item in props.options property is selected.									
options	<p>And array of objects for each dropdown option.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>value</td> <td> <p>Actual value to be matched by the input or selection.</p> <p>The type of this property is initially a Value-type, but it can be converted to an Object-type or Array-type. Doing so will populate the PROPS.value property with the entire object/array, allowing a single selection on the dropdown to return multiple values</p> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> Each option must have a unique value. </div> </td> <td>variable</td> </tr> <tr> <td>label</td> <td>Text to display in the menu representing this option.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	value	<p>Actual value to be matched by the input or selection.</p> <p>The type of this property is initially a Value-type, but it can be converted to an Object-type or Array-type. Doing so will populate the PROPS.value property with the entire object/array, allowing a single selection on the dropdown to return multiple values</p> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> Each option must have a unique value. </div>	variable	label	Text to display in the menu representing this option.	value: string	array
Name	Description	Property Type									
value	<p>Actual value to be matched by the input or selection.</p> <p>The type of this property is initially a Value-type, but it can be converted to an Object-type or Array-type. Doing so will populate the PROPS.value property with the entire object/array, allowing a single selection on the dropdown to return multiple values</p> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> Each option must have a unique value. </div>	variable									
label	Text to display in the menu representing this option.	value: string									

	isDisabled	Whether this option is currently disabled from selection. If set to true, option will not be selectable, and will use a grey font (assuming another text color isn't being applied)	value: boolean																								
multiSelect	Enable multiple selected values. Default is false.		value: boolean																								
placeholder	Settings for the text displayed when value is empty.		object																								
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Prompt text to display when no options are selected.</td> <td>value: string</td> </tr> <tr> <td>color</td> <td>Color of placeholder text. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>color</td> </tr> <tr> <td>icon</td> <td>Settings for an icon used as a placeholder.</td> <td>object</td> </tr> </tbody> </table>		Name	Description	Property Type	text	Prompt text to display when no options are selected.	value: string	color	Color of placeholder text. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color	icon	Settings for an icon used as a placeholder.	object													
Name	Description	Property Type																									
text	Prompt text to display when no options are selected.	value: string																									
color	Color of placeholder text. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color																									
icon	Settings for an icon used as a placeholder.	object																									
		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Shorthand path to the icon source. Format is library /iconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons.</td> <td>value: string</td> </tr> <tr> <td>color</td> <td>Color of the placeholder icon, if it exists. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>color</td> </tr> <tr> <td>style</td> <td>Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. Three additional available settings are as follows: <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>margin</td> <td>Margin around the icon, in pixels. Default is 0px 8px -3px 0px</td> <td>value: numeric</td> </tr> <tr> <td>width</td> <td>Width of the icon in pixels. Default is 16px.</td> <td>value: numeric</td> </tr> <tr> <td>height</td> <td>Height of the icon in pixels. Default is 16px.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Shorthand path to the icon source. Format is library /iconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string	color	Color of the placeholder icon, if it exists. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color	style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. Three additional available settings are as follows: <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>margin</td> <td>Margin around the icon, in pixels. Default is 0px 8px -3px 0px</td> <td>value: numeric</td> </tr> <tr> <td>width</td> <td>Width of the icon in pixels. Default is 16px.</td> <td>value: numeric</td> </tr> <tr> <td>height</td> <td>Height of the icon in pixels. Default is 16px.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	margin	Margin around the icon, in pixels. Default is 0px 8px -3px 0px	value: numeric	width	Width of the icon in pixels. Default is 16px.	value: numeric	height	Height of the icon in pixels. Default is 16px.	value: numeric	object	
Name	Description	Property Type																									
path	Shorthand path to the icon source. Format is library /iconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string																									
color	Color of the placeholder icon, if it exists. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color																									
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. Three additional available settings are as follows: <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>margin</td> <td>Margin around the icon, in pixels. Default is 0px 8px -3px 0px</td> <td>value: numeric</td> </tr> <tr> <td>width</td> <td>Width of the icon in pixels. Default is 16px.</td> <td>value: numeric</td> </tr> <tr> <td>height</td> <td>Height of the icon in pixels. Default is 16px.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	margin	Margin around the icon, in pixels. Default is 0px 8px -3px 0px	value: numeric	width	Width of the icon in pixels. Default is 16px.	value: numeric	height	Height of the icon in pixels. Default is 16px.	value: numeric	object													
Name	Description	Property Type																									
margin	Margin around the icon, in pixels. Default is 0px 8px -3px 0px	value: numeric																									
width	Width of the icon in pixels. Default is 16px.	value: numeric																									
height	Height of the icon in pixels. Default is 16px.	value: numeric																									
enabled	If set to false, component is disabled. Field will not focus and dropdown will not be interactive.		value: boolean																								
	<div style="border: 1px solid #ccc; padding: 5px;"> <p>Note: If the component is disabled, scripts can still run on the component. For example, if you add a script action to the onClick event, the script will fire when the user clicks on the Dropdown.</p> </div>																										
search	Enter text to start search.		object																								
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether options are searchable by typing text into the field. Default is true.</td> <td>value: boolean</td> </tr> <tr> <td>matching</td> <td>Whether search string must match from the start or may match any position of an option: start or any.</td> <td>value: string</td> </tr> <tr> <td>noResultsText</td> <td>Text to display in dropdown when no option matches the search. Default is "No results found."</td> <td>value: string</td> </tr> </tbody> </table>		Name	Description	Property Type	enabled	Whether options are searchable by typing text into the field. Default is true.	value: boolean	matching	Whether search string must match from the start or may match any position of an option: start or any.	value: string	noResultsText	Text to display in dropdown when no option matches the search. Default is "No results found."	value: string													
Name	Description	Property Type																									
enabled	Whether options are searchable by typing text into the field. Default is true.	value: boolean																									
matching	Whether search string must match from the start or may match any position of an option: start or any.	value: string																									
noResultsText	Text to display in dropdown when no option matches the search. Default is "No results found."	value: string																									

	searchParam	The text being searched for.	value: string
showClearIcon		Whether to display a button that the user can use to clear the selection. Default is false.	value: boolean
allowCustomOptions		Whether a user may enter a custom value to be submitted. Default is false. While set to True, typing a value that doesn't match one of the existing options from the session will provide the user with a "Create" option. Selecting the Create option will set <code>props.value</code> to the new option. Creating a custom option in this way does not change the value of <code>props.options</code> . The custom option is simply a means to allow the user to type a custom value into the component.	value: boolean
textAlign		<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.19 Click here to check out the other new features</p> </div> <p>Aligns the value(s) and/or placeholder text displayed within the dropdown. Valid values are 'left', 'center', and 'right'. <code>textAlign</code> within the dropdown modal itself may be overridden or set separately using <code>props.dropdownOptionStyle.textAlign</code>.</p>	value: string
minMenuHeight		<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.29 Click here to check out the other new features</p> </div> <p>Minimum height of the dropdown menu. If the minimum height is not available under the component, and space is available above the component, the dropdown will flip to display options above the component. Default value is 150.</p>	value: numeric
maxMenuHeight		<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.29 Click here to check out the other new features</p> </div> <p>Maximum height of the dropdown menu before it becomes scrollable. Default value is 350.</p>	value: numeric
style		Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object
dropdownOptionStyle		Sets a style for the dropdown options. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Examples

Example 1



Property	Value
props.options.0.value	[default]_Dairy_/Bldg25/valve1
props.options.0.label	Valve 1
props.options.1.value	[default]_Dairy_/Bldg25/valve2
props.options.1.label	Valve 2
props.options.2.value	[default]_Dairy_/Bldg25/valve3
props.options.2.label	Valve 3
props.options.3.value	[default]_Dairy_/Bldg25/valve4
props.options.3.label	Valve 4
props.placeholder.text	Select a Valve...
props.placeholder.color	#0000AC
props.placeholder.icon.path	material/grade
props.placeholder.icon.color	#008000
props.style.borderStyle	solid
props.style.color	#0000D9
props.style.fontFamily	garamound
props.style.fontSize	16px
props.style.fontWeight	bold
props.style.borderWidth	2px
props.style.borderColor	#008000
props.style.borderRadius	8px
props.dropdownOptionStyle.color	#008000
props.dropdownOptionStyle.fontSize	14px
props.dropdownOptionStyle.fontWeight	bold
props.dropdownOptionStyle.textAlign	right

Example 2

In this example, we have a dropdown list with an expression binding on the options property. There is also a label on the view with the word "Email" as its text.

A default email address of j_smith@companyname.com is set as the starting value for the component.

Email

As the user starts entering characters for an email address, the dropdown list provides typeahead options of the entered text plus three possible email options, '@cn.com', '@companyname.com', or '@gmail.com'.

Email

- s_jones@cn.com
- s_jones@companyname.com
- s_jones@gmail.com
- Create "s_jones"

Property	Value
value	j_smith@companyname.com
options	(Bound to an expression. See example below.)
props.search.enabled	true

transform code

```
# suggest auto-completed options for an email address
options = []
# skip if blank
if value:
    # check for @ symbol and suggest email address if not present
    if "@" not in value:
        options.append({ "value":value+"@cn.com", "label":value+"@cn.com"})
        options.append({ "value":value+"@gmail.com", "label":value+"@gmail.com"})
        options.append({ "value":value+"@companyname.com", "label":value+"@companyname.com"})
    # check for extension (.com) and suggest extensions if not present
    elif ".com" not in value and ".net" not in value and ".org" not in value:
        options.append({ "value":value+".com", "label":value+".com"})
        options.append({ "value":value+".net", "label":value+".net"})
        options.append({ "value":value+".org", "label":value+".org"})
# return a list of suggested options
return options
```

Edit Binding: Dropdown.props.options

Binding Type

- Tag
- Property
- Expression
- Expression Structure
- Query
- Tag History
- HTTP

Remove Binding

Configure Expression Binding

```
1 {{this.props.search.searchParam}}
```

Options

Enabled Overlay Opt-Out

Configure Transform(s)

Script

```
2 # suggest auto-completed options for an email address
3 options = []
4 # skip if blank
5 if value:
6     # check for @ symbol and suggest email address if not present
7     if "@" not in value:
8         options.append({ "value":value+"@cn.com", "label":value+"@cn.com"})
9         options.append({ "value":value+"@gmail.com", "label":value+"@gmail.com"})
10        options.append({ "value":value+"@companyname.com", "label":value+"@companyname.com"})
11    # check for extension (.com) and suggest extensions if not present
12    elif ".com" not in value and ".net" not in value and ".org" not in value:
13        options.append({ "value":value+".com", "label":value+".com"})
14        options.append({ "value":value+".net", "label":value+".net"})
15        options.append({ "value":value+".org", "label":value+".org"})
16    # return a list of suggested options
17    return options
```

Add Transform +

Binding Preview

Expression → Script

OK Cancel Apply

Perspective - File Upload



[Browse](#) or Drag files here

On this page ...

- [Properties](#)
- [Scripting](#)

Component Palette Icon:



The File Upload component allows users to upload files to the Gateway or other locations from a Perspective session using a script action on the onFileReceived component event. For an example, see [Download and Upload Files](#) page.

The component has different appearances based on its width. When initially dragged onto your View, a Browse button will be visible and paired with

"Drag files here" text. At smaller widths, the component defaults to a simple cloud  icon and no text. This icon can be changed using the fileUploadIcon property settings.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type									
maxUploads	The maximum number of concurrent (simultaneous) uploads to allow. Default is 5.	value: integer									
supportedFileTypes	An array of string values, indicating what file types are allowed to be uploaded. Example values are "pdf" or "txt".	array									
fileSizeLimit	Specifies the maximum size of each uploaded file, in megabytes (MB). Default is 10 MB.	value: integer									
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object									
fileUploadIcon	Determines the icon used when the File Upload component is small. <table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>path</td><td>Shorthand path to icon source, in format: library/iconName (i.e., material/arrow_right). The materials icon library is the default source for icons in Ignition. See https://fonts.google.com/icons?selected=Material+Icons.</td><td>value: string</td></tr><tr><td>color</td><td>Color of the icon. Here for convenience, may instead assign 'fill' in the styles property. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td><td>color</td></tr></tbody></table>	Name	Description	Property Type	path	Shorthand path to icon source, in format: library/iconName (i.e., material/arrow_right). The materials icon library is the default source for icons in Ignition. See https://fonts.google.com/icons?selected=Material+Icons .	value: string	color	Color of the icon. Here for convenience, may instead assign 'fill' in the styles property. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color	object
Name	Description	Property Type									
path	Shorthand path to icon source, in format: library/iconName (i.e., material/arrow_right). The materials icon library is the default source for icons in Ignition. See https://fonts.google.com/icons?selected=Material+Icons .	value: string									
color	Color of the icon. Here for convenience, may instead assign 'fill' in the styles property. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color									

Scripting

See the [Perspective - File Upload Scripting](#) page for the full list of scripting functions available for this component.

Download and Upload Files

Downloading and uploading files from a Perspective session typically involves storing and retrieving files from a database. A table will store all of the available files, and each row of the table represents a new file. This allows for long term storage that is accessible from any project.

Query Examples

The examples on this page show suggested methods of uploading files from a session, as well as how to download them. Before following along with the examples on this page, you'll need to create a table in the database that will hold the files. This process can vary by database, along with the column datatypes.

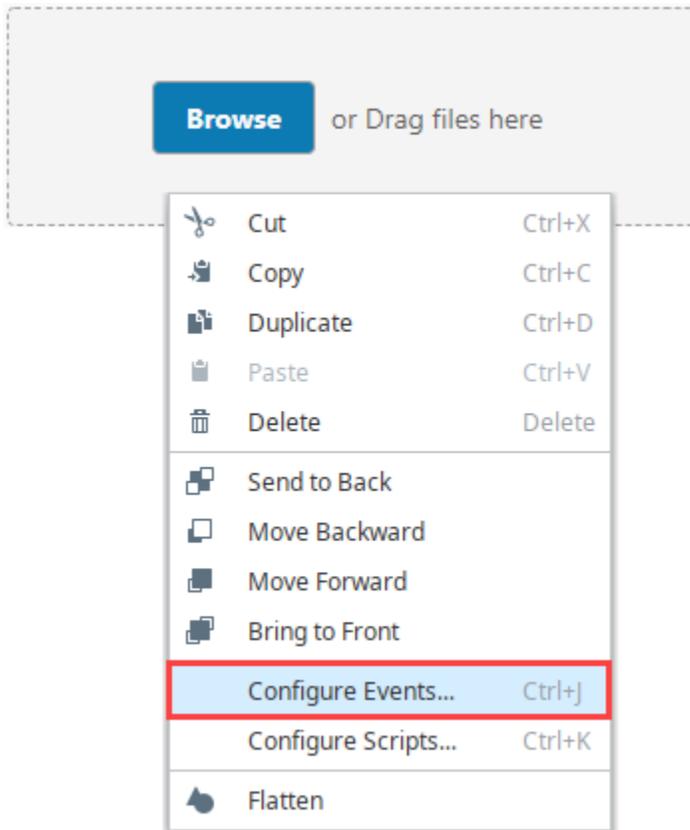
For the sake of brevity, the example assumes the files will be stored and retrieved from a SQL Server database. You may need to modify the query examples on this page if using a different database. The "files" database table used by these examples contains the following columns:

- **id** - integer, primary key, identity
- **filename** - varchar (255)
- **filedata** - varbinary (MAX)

Uploading a File

To upload a file in Perspective, we will want to use the [File Upload component](#). This allows us an easy way to manage the upload.

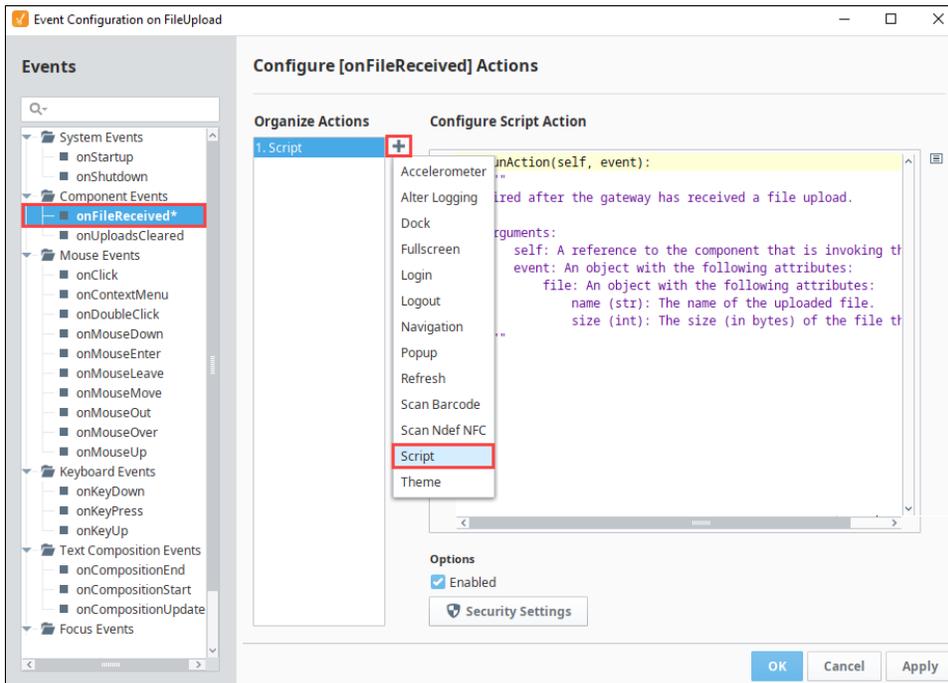
1. Add the File Upload component to a view. The File Upload component has everything we need to upload a file into the database.
2. Right-click on the File Upload component and select **Configure Events**.



3. Select the **onFileReceived** event and click the **Add**  icon to add a **script** action to it.

On this page ...

- [Query Examples](#)
- [Uploading a File](#)
- [Downloading a File](#)



4. Add the following script to the script action:

```
# Grab the file name and data
filename = event.file.name
filedata = event.file.getBytes()

# Use a query to insert the file
query = "INSERT INTO files (filename, filedata) VALUES(?, CONVERT(varbinary(MAX), ?))"
args = [filename, filedata]
db = "myDatabase"
system.db.runPrepUpdate(query, args, db)
```



As mentioned above, the query will vary based on the database used.

5. Click **OK**.

You can test out the upload functionality by dragging a file onto the File Upload component, either from a session, or the designer while it's in preview mode. After the file is uploaded, a successful upload message will appear and your file will be present in the created database table.

Downloading a File

To download a file that is stored in the database in Perspective, we will want to use the `system.perspective.download` function. This will allow us to download the file data that we receive from the database.

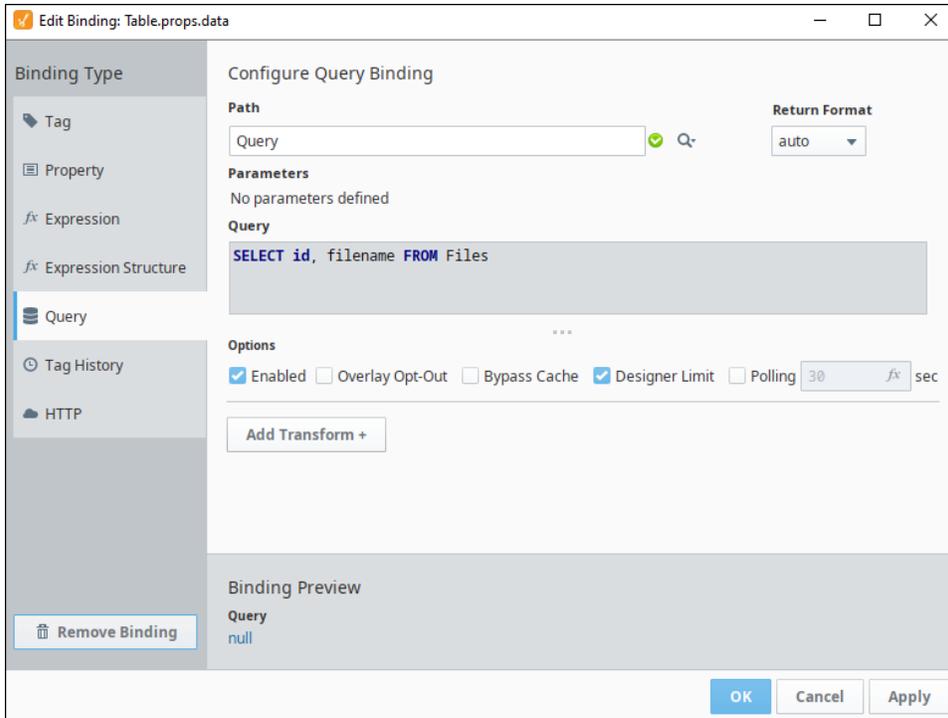
This example will show you how to do several things:

- Create a [named query](#), that will return the contents of our file database table
 - Create a table component, that shows a listing of potential files to download, using the named query above in conjunction with a Named Query Binding.
 - Add a button component, that will allow users to download a file, assuming one of the rows in the table component are selected.
1. Create a Named Query that we will use to pull a list of files out of the database table. We're using a named query here since a named query binding is the easiest way to run a query when the view loads. The query should pull out the id of the row which we can use to later query the data, as well as the filename which the user can use to identify the file.

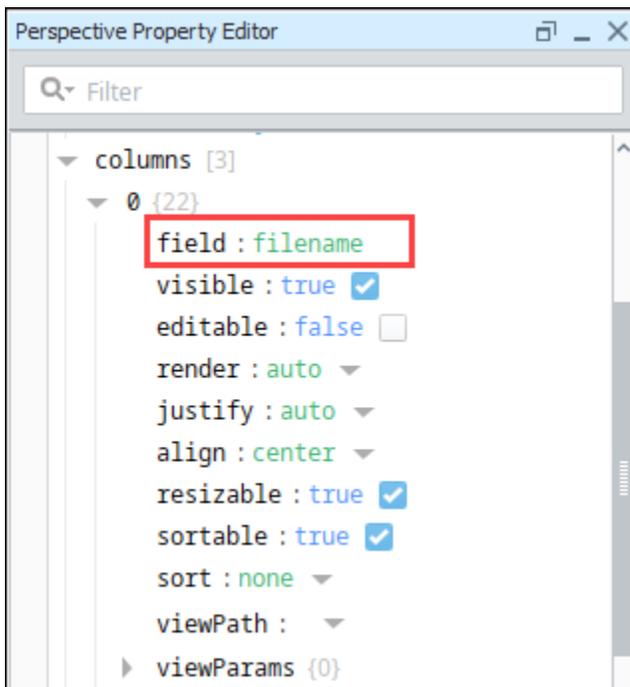
```
SELECT id, filename FROM Files
```

2. On a view, add a Table component. This will display a list of all files we currently have in the database table.

- Set up a binding on the Table's **data** property. The binding should be a Query type, and it should use the query that we just made. We want to return the data in a JSON format, and you can enable polling so that it automatically updates if new files get uploaded.



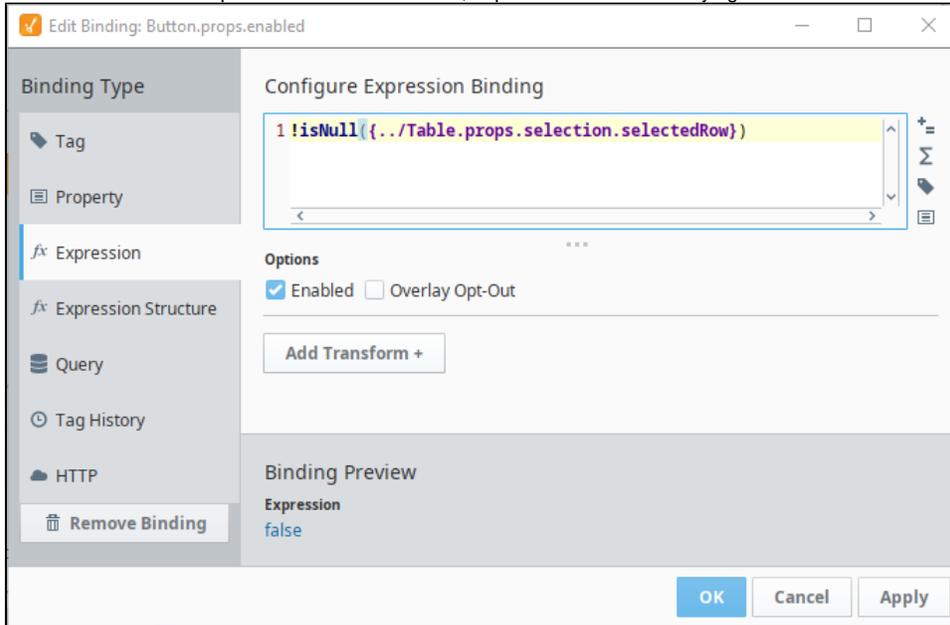
- On the Table's columns property, add an array element. Set **columns.0.field** to the name of the column that holds the filename. This will display only the filename column, as the id column does not need to be visible.



- Add a Button to the view. This button will be used to download the file after the user has made a selection. However, we also want to make sure the user can't press the button unless a row in the table is selected.
- On the Button's enabled property, configure a binding. The binding type should be an expression. The expression should check to see if the Table's selected row is null, and invert it.

```
!isNull({../Table.props.selection.selectedRow})
```

This will disable the component if no row is selected, to prevent the user from trying to download without making a selection.



7. Right-click on the Button and go to **Configure Events**.
8. Select the **onActionPerformed** event, click the **Add +** icon to add a **script** action to it.
9. Add the following script to the script action:

```
# Grab the selected row
selectedRow = self.getSibling("Table").props.selection.selectedRow

# Use the selected row to grab the id of the file at that row
id = self.getSibling("Table").props.data[selectedRow].id

# Use the id to grab the file data out of the database, along with its corresponding name.
query = "SELECT filename, filedata FROM Files WHERE id = ?"
args = [id]
data = system.db.runPrepQuery(query, args)

# Pull out the file name and data
filename = data[0][0]
filedata = data[0][1]

# Download the file
system.perspective.download(filename, filedata)
```

10. Test the script by selecting a row in the table and clicking on the button while in Preview mode.

Perspective - File Upload Scripting

This page details the various scripting, component, and extension functions available for [Perspective's File Upload](#) component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

On this page ...

- [Component Events](#)
 - [onFileReceived](#)
 - [onUploadsCleared](#)
- [Component Functions](#)
 - [.clearUploads\(\)](#)
- [Extension Functions](#)

onFileReceived

Provides a chance to handle file data uploaded to the component.

Note: This component event is designed to be used in tandem with a script action. Within the script action, special properties and methods are available on the **event** object, which is passed to the script action as a parameter.

event.file.name

- Object Path

event.file.name

- Type

[String](#)

- Description

The name of the uploaded file.

event.file.size

- Object Path

event.file.size

- Type

[Integer](#)

- Description

The size of the uploaded file in bytes.

event.file.copyTo(filePath)

- Object Path

event.file.copyTo()

- Description

Saves the uploaded file at a location accessible to the Gateway.

- Parameters

[String](#) filePath - The path to where the file should be saved on the Gateway.

- Return

None

event.file.getBytes()

- Object Path

`event.file.getBytes()`

- Description

Fetches the incoming file data. Suitable for further data processing.

- Parameters

None

- Return

[byteArray](#) - The raw data of the incoming file.

event.file.getString()

- Object Path

`event.file.getString()`

- Description

Fetches the incoming file data and attempts to parse it as a string via UTF-8 (Eight-bit UCS Transformation Format) encoding. Defaults to UTF-8 (super common), but can use other character sets. Passed as a string, for example `getString("UTF_16BE")`.

- Parameters

None

- Return

[String](#) - The raw data of the incoming file as a string.

onUploadsCleared

This event is fired when the user has cleared all uploads, but not while uploads are still in progress.

Note: This component event is designed to be used in tandem with a script action. Within the script action, special properties and methods are available on the **event** object, which is passed to the script action as a parameter.

event.files.name

- Object Path

`event.files.name`

- Type

[String](#)

- Description

The name of the uploaded file.

event.files.size

- Object Path

`event.files.size`

- Type

[Integer](#)

- Description

The size (in bytes) of the uploaded file.

Component Functions

.clearUploads()



The following feature is new in Ignition version **8.1.18**
[Click here](#) to check out the other new features

- Description

Resets the File Upload component to its default state

Note: clearUploads() does not remove or delete uploaded files from the Gateway. Clearing uploads does not undo any actions triggered by the onFileReceived() Component Event.

- Parameters

None

- Return

Nothing

Extension Functions

This component does not have extension functions associated with it.

Perspective - Multi-State Button



On this page ...

- [Properties](#)
- [Component Events](#)
- [Example](#)

Component Palette Icon:



The Multi-State button is really a series of two or more buttons, arranged in a column or row. Each button represents an integer-valued state. Each state defines two styles for a button: the selected style, and the unselected style. Each button is automatically displayed with the correct style based on the current state (the value of Indicator Value). When a button is pressed, its state's value is written to the Control Value.

When the Multi-State Button is dragged to a container, it is pre-configured with 'defaultSelectedStyle' and 'defaultUnselectedStyle' properties'. These styles can be changed or deleted.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type																		
controlValue	Bind this to the Tag that controls the state. (Typically, this is bound to the same location as the indicatorValue property.)	value: numeric																		
indicatorValue	Bind this to the Tag that indicates the current state. (Typically, this is bound to the same location as the controlValue property).	value: numeric																		
states	The value that will be written to controlValue when any of the buttons are clicked. Shows a list of the possible states for the component. You can add, remove, and the change the order of each state listed. Each state has two default visual styles applied for each button: Selected Style and Unselected Style. The Multi-State Button has default visual styles defined for both the selectedStyle and unselectedStyle. (Refer to 'defaultSelectedStyle' and 'defaultUnselectedStyle' properties in this table).	object																		
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Text displayed on the button.</td> <td>string</td> </tr> <tr> <td>value</td> <td>Value assigned to the state.</td> <td>value: numeric</td> </tr> <tr> <td>selectedStyle</td> <td>Style settings for the button when it is selected. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>unselectedStyle</td> <td>Style settings for the button when it is not selected. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>tooltipText</td> <td>Determines what text should appear when mouse cursor hovers over the button associated with this property. If blank, no tooltip will appear.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	text	Text displayed on the button.	string	value	Value assigned to the state.	value: numeric	selectedStyle	Style settings for the button when it is selected. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	unselectedStyle	Style settings for the button when it is not selected. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	tooltipText	Determines what text should appear when mouse cursor hovers over the button associated with this property. If blank, no tooltip will appear.	value: string	
Name	Description	Property Type																		
text	Text displayed on the button.	string																		
value	Value assigned to the state.	value: numeric																		
selectedStyle	Style settings for the button when it is selected. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																		
unselectedStyle	Style settings for the button when it is not selected. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																		
tooltipText	Determines what text should appear when mouse cursor hovers over the button associated with this property. If blank, no tooltip will appear.	value: string																		
	<div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>Note: This property is not present by default, and must be added manually.</p> </div>																			

orientation	Physical position of the button: Column or Row.	boolean
defaultSelectedStyle	Default selected style. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object
defaultUnselectedStyle	Default styles for unselectedStyles when any of the buttons are <i>not</i> selected. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object
primary	Toggles between the default primary and secondary button style.	value: boolean
enabled	If true, the user is able to interact with the buttons.	value: boolean
buttonGap	Space, in pixels, between each button in a group.	value: numeric
endButtonCornerRadius	Amount to round the end of the corners of the first and last button.	value: numeric
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Example



Property	Value
props.indicatorValue	(property binding)
props.states.0.text	Open
props.states.0.value	2
props.states.0.selectedStyle.backgroundColor	#FFF809
props.states.1.text	Close
props.states.1.value	0
props.states.1.selectedStyle.backgroundColor	#FF8C00
props.states.2.text	Auto
props.states.2.value	1
props.states.2.selectedStyle.backgroundColor	#62ED2A
props.states.3.text	Bypass
props.states.3.value	4
props.states.3.selectedStyle.backgroundColor	#FF0000

props.orientation	row
props.buttonGap	8
props.endButtonCornerRadius	2

Perspective - Numeric Entry Field

1,000.50

Component Palette Icon:

 Numeric Entry Field

On this page ...

- [Properties](#)
- [Component Events](#)
- [Examples](#)
 - [Example 1](#)
 - [Example 2](#)
 - [Example 3](#)

The Numeric Entry Field is similar to the standard Text Field, except that it is specialized for use with numbers. When the 'enabled' property is set to true, it allows users to alter the value on the component. There are three different modes for how users can edit the value in the component: direct, protected or by clicking an edit button. To change the value, click once in the field for 'direct' mode, double click for 'protected' mode, and click on the Edit icon for the 'button' mode. When done, press enter or leave the field, and the field becomes editable again. When the 'enabled' property is false, the field is not editable even when it receives input focus.

The following feature is new in Ignition version **8.1.2**
[Click here](#) to check out the other new features

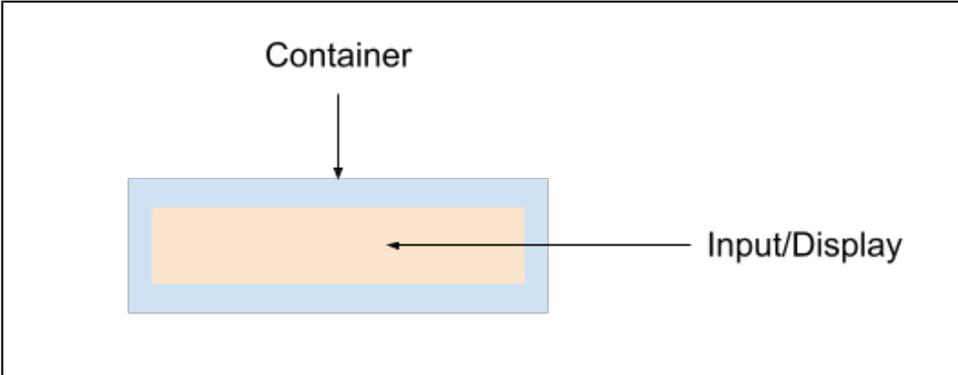
The Numeric Entry Field component has three pre-configured [variants](#):

- Direct - Default design of the field.
- Protect - Requires a double-click or long-press to enter edit mode.
- Button - Clicking the button brings up a popup window, allowing the user to edit the value from the popup, or cancel the edit.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type								
value	Value as number or numeric string to display.	value: numeric								
format	<p>The formatting string to be applied to the input value. Options are currency, number, integer, four decimal precision, percent, scientific, accounting, financial, currency, currency (rounded), duration, abbreviation, or ordinal. A list of format specifiers can be found here.</p> <p>The following feature is new in Ignition version 8.1.2 Click here to check out the other new features</p> <p>The Numeric Entry Field supports locale-specific formatting, allowing changes to the session's locale to update how numbers are formatted on the Numeric Entry Field. Note that the localization conversion occurs automatically <i>after</i> the initial format specifiers are applied.</p>	value: string								
mode	<p>Determines how users will edit the value in the component. The following modes are available:</p> <table border="1"><thead><tr><th>Mode</th><th>Description</th></tr></thead><tbody><tr><td>Direct</td><td>Requires no special action to enter edit mode simply click in the field.</td></tr><tr><td>Protected</td><td>Requires a double-click or long-press to enter edit mode.</td></tr><tr><td>Button</td><td>Places an Edit  icon next to the Numeric Entry Field. Clicking the button brings up a popup window, allowing the user to edit the value from the popup, or cancel the edit.</td></tr></tbody></table>	Mode	Description	Direct	Requires no special action to enter edit mode simply click in the field.	Protected	Requires a double-click or long-press to enter edit mode.	Button	Places an Edit  icon next to the Numeric Entry Field. Clicking the button brings up a popup window, allowing the user to edit the value from the popup, or cancel the edit.	value: string
Mode	Description									
Direct	Requires no special action to enter edit mode simply click in the field.									
Protected	Requires a double-click or long-press to enter edit mode.									
Button	Places an Edit  icon next to the Numeric Entry Field. Clicking the button brings up a popup window, allowing the user to edit the value from the popup, or cancel the edit.									

	The value in the component may only be edited via the popup.													
align	Aligns the input value right or left.	value: boolean												
inputBo unds	Max and min bounds configuration. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>maximum</td> <td>The max allowable value.</td> <td>value: numeric</td> </tr> <tr> <td>minimum</td> <td>The min allowable value.</td> <td>value: numeric</td> </tr> <tr> <td>invalidS tyle</td> <td>Sets an invalid style when the min or max values are out of bounds for this component. Modify the invalidStyle using the style properties. Full menu of style options is available. You can also specify a style class as an invalid style.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	maximum	The max allowable value.	value: numeric	minimum	The min allowable value.	value: numeric	invalidS tyle	Sets an invalid style when the min or max values are out of bounds for this component. Modify the invalidStyle using the style properties. Full menu of style options is available. You can also specify a style class as an invalid style.	object	object
Name	Description	Property Type												
maximum	The max allowable value.	value: numeric												
minimum	The min allowable value.	value: numeric												
invalidS tyle	Sets an invalid style when the min or max values are out of bounds for this component. Modify the invalidStyle using the style properties. Full menu of style options is available. You can also specify a style class as an invalid style.	object												
placeho lder	Text to be displayed when value is empty.	value: string												
spinner	<p>The following feature is new in Ignition version 8.1.12 Click here to check out the other new features</p> <p>Optional spinner configuration.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>If enabled, a spinner will appear when the field is selected by the user.</td> <td>value: boolean</td> </tr> <tr> <td>increment</td> <td>The increment the spinner uses to increase or decrease the value.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	If enabled, a spinner will appear when the field is selected by the user.	value: boolean	increment	The increment the spinner uses to increase or decrease the value.	value: numeric	object			
Name	Description	Property Type												
enabled	If enabled, a spinner will appear when the field is selected by the user.	value: boolean												
increment	The increment the spinner uses to increase or decrease the value.	value: numeric												
tooltipT ext	Mousing over this button will display a tooltip with this text, if present.	value: string												
enabled	Indicates if user should be allowed to alter the value. <div style="border: 1px solid #add8e6; padding: 5px;"> <p>Note: If the component is disabled, scripts can still run on the component. For example, if you add a script action to a System Event, such as an onStartup event, the script will fire when the page is loaded. Events that require user interaction, such as onClick events, will not fire with the exception of Pointer Events.</p> </div>	value: boolean												
contain erStyle	<p>The following feature is new in Ignition version 8.1.2 Click here to check out the other new features</p> <p>Sets a style for the outer area of the component. The image below represents a low fidelity representation of the component. The containerStyle property determines the look of the outer "container" area of the component, making it ideal for adding borders, margins around the entire component, and padding between the container and input/display area.</p> 	object												

	Changes that should be made to the input/display area, such as changing the font on the displayed value, should be made on the style property.	
style	Sets a style for the "inner" numeric display/input in this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object

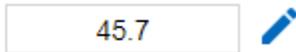
Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Examples

Example 1



This example shows the component set to "button" mode, providing a button to click on when entering a new value.

Property	Value
props.value	45.678
props.format	#0.0
props.mode	button
props.align	center
props.placeholder	setpoint

Example 2



This examples demonstrates the placeholder property, showing a default entry in cases where the value is null.

Property	Value
props.value	null
props.format	#0.0
props.mode	button
props.align	center
props.placeholder	setpoint

Example 3



This example demonstrates the format property, allowing custom formatting to be applied to the value in the component.

Property	Value

props.value	0.2345
props.format	0.0#%
props.mode	direct
props.align	right
props.placeholder	setpoint

Perspective - One-Shot Button

One-Shot Button

Component Palette Icon:



On this page ...

- [Properties](#)
- [Component Events](#)
- [Example](#)

The One-Shot Button is designed to send off a write request, and wait for a response, disabling the button until something resets the 'value' property on the component.

When the 'value' property and the 'setValue' property are equal, the component transitions to the writing state. Once 'value' and 'setValue' are no longer equal, the button returns to the ready state.

The following feature is new in Ignition version **8.1.2**
[Click here](#) to check out the other new features

The One-Shot Button component has three pre-configured [variants](#):

- Primary - Default design of the button.
- Secondary - A secondary design for the button.
- Require Confirm - Default design of the button but requires confirmation from user before action is submitted.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type						
value	<p>The current value displayed on the component. Should be bound bi-directionally to a Tag. Default is 0.</p> <p>The following feature is new in Ignition version 8.1.4 Click here to check out the other new features</p> <p>As of 8.1.4 the property type for value can be numeric, boolean, string, or null.</p>	value: numeric, boolean, string, or null.						
setValue	<p>The value to set when the button is pushed. Default is 1.</p> <p>The following feature is new in Ignition version 8.1.4 Click here to check out the other new features</p> <p>As of 8.1.4 the property type for setValue can be numeric, boolean, string, or null.</p>	value: numeric, boolean, string, or null.						
primary	Toggle between the default primary and secondary button style. Default is true.	value: boolean						
enabled	Whether the user can interact with the One-Shot Button. If disabled, the component cannot be used. Default is true.	value: boolean						
readyState	Displays the readyState value on the component.	object						
	<table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>text</td><td>The text of the button while it's value is not being written.</td><td>value: string</td></tr></tbody></table>	Name	Description	Property Type	text	The text of the button while it's value is not being written.	value: string	
Name	Description	Property Type						
text	The text of the button while it's value is not being written.	value: string						

style	Modify readyState style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																	
icon	The Icon is an image path used to augment the readyState of the component by placing an image next to it.	object																	
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Name</th> <th style="width: 55%;">Description</th> <th style="width: 30%;">Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Shorthand path to icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons.</td> <td>value: string</td> </tr> <tr> <td>color</td> <td>Color of the icon. Can also assign color in "fill" of the style property. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>color</td> </tr> <tr> <td>style</td> <td>Modify the readyState style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>		Name	Description	Property Type	path	Shorthand path to icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string	color	Color of the icon. Can also assign color in "fill" of the style property. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color	style	Modify the readyState style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object					
	Name		Description	Property Type															
	path		Shorthand path to icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string															
	color		Color of the icon. Can also assign color in "fill" of the style property. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color															
style	Modify the readyState style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																	
writingState	Displays the writingState value on the component.	object																	
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Name</th> <th style="width: 55%;">Description</th> <th style="width: 30%;">Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>The text of the button while its value is being written. Default is "Writing..."</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Modify the writingState style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> <tr> <td>icon</td> <td>An image path used to augment the writingState of the component by placing an image next to it.</td> <td rowspan="4">object</td> </tr> <tr> <td></td> <td></td> </tr> <tr> <td></td> <td></td> </tr> <tr> <td></td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	text	The text of the button while its value is being written. Default is "Writing..."	value: string	style	Modify the writingState style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	icon	An image path used to augment the writingState of the component by placing an image next to it.	object						
	Name	Description	Property Type																
	text	The text of the button while its value is being written. Default is "Writing..."	value: string																
	style	Modify the writingState style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																
	icon	An image path used to augment the writingState of the component by placing an image next to it.	object																
confirm	<p>If enabled, a message that asks the user to approve the requested operation. If the user doesn't confirm, then the value property will not update.</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p>Note: When configuring a script on this component while confirm is enabled, it is generally recommend that the script is placed on the <code>onActionPerformed</code> component event, since that event will wait for user confirmation.</p> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Name</th> <th style="width: 55%;">Description</th> <th style="width: 30%;">Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>If true, a confirmation box will be shown. Default is false.</td> <td>value: boolean</td> </tr> <tr> <td>text</td> <td>Message to show user if confirmation is enabled. Default is "Are you sure?"</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	If true, a confirmation box will be shown. Default is false.	value: boolean	text	Message to show user if confirmation is enabled. Default is "Are you sure?"	value: string	object								
Name	Description	Property Type																	
enabled	If true, a confirmation box will be shown. Default is false.	value: boolean																	
text	Message to show user if confirmation is enabled. Default is "Are you sure?"	value: string																	
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																	
disabledStyle		object																	

The following feature is new in Ignition version **8.1.25**
[Click here](#) to check out the other new features

Sets a style for this component when it is **disabled**. Full menu of [style options](#) is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a [style class](#).

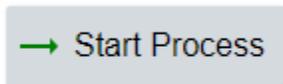
Note: If you are using both the style and disabledStyle properties, keep in mind that CSS dictates that style classes will be rendered in alphabetical order. In other words, specific elements of a style class will be overwritten if a later style class modifies the same element.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Example

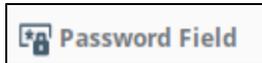


Property	Value	Style Category
props.readyState.text	Start Process	N/A
props.readyState.icon.path	material/trending_flat	N/A
props.readyState.icon.color	#008000	N/A
props.writingState.text	Starting	N/A
props.writingState.style.backgroundColor	#8AFF8A	background

Perspective - Password Field



Component Palette Icon:



On this page ...

- [Properties](#)
- [Component Events](#)
- [Example](#)

The Password Field component is similar to a Text Field component. It allows users to enter their password text. When the Password field is empty, you can create a placeholder that informs user to "Login". You can also enable the "allowReveal" property to allow users to view their password entry.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
text	Password text.	value: string
placeholder	Text displayed when password text is empty.	value: string
enabled	Whether the user can alter the password text. Default is true. Note: If the component is disabled, scripts can still run on the component. For example, if you add a script action to a System Event, such as an onStartup event, the script will fire when the page is loaded. Events that require user interaction, such as onClick events, will not fire with the exception of Pointer Events.	value: boolean
allowReveal	Whether the user can temporarily remove the password mask, revealing the password.	value: boolean
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Example



Property	Value	Style Category
props.placeholder	Enter Password	N/A

props.borderStyle	solid	border
props.borderColor	#D90000	border
props.borderWidth	3px	border
props.fontSize	14px	text
props.fontWeight	bold	text

Perspective - Radio Group



Component Palette Icon:



On this page ...

- [Properties](#)
- [Component Events](#)

The Radio Group allows you to create multiple radio buttons in a single container. The number of radio buttons in the group is determined by the number of elements in the "radios" object. Only one radio button in a group may be selected at a time. Radio groups are a good way to let the user choose just one of a number of options. If multiple selections are expected, the [Checkbox](#) or [Dropdown](#) components can be used.

The following feature is new in Ignition version **8.1.2**
[Click here](#) to check out the other new features

The Radio Group component has three pre-configured [variants](#):

- Text Right - Default layout with text on the right of the radio button.
- Text Left - Layout with text on the left of the radio button.
- Multiple - Layout with multiple radio buttons and text on the right.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type													
value	The value of the selected radio.	value: string, integer, boolean, or null													
index	The index of the selected node.	value: integer													
radios	List of radios that make up this group.	array													
	<table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>text</td><td>Text to pair with this radio.</td><td>value: string</td></tr><tr><td>selected</td><td>If 'true,' this radio is selected.</td><td>value: boolean</td></tr><tr><td>value</td><td>The value of the radio to be evaluated when selected.</td><td>value: numeric</td></tr></tbody></table>		Name	Description	Property Type	text	Text to pair with this radio.	value: string	selected	If 'true,' this radio is selected.	value: boolean	value	The value of the radio to be evaluated when selected.	value: numeric	
	Name		Description	Property Type											
	text		Text to pair with this radio.	value: string											
selected	If 'true,' this radio is selected.	value: boolean													
value	The value of the radio to be evaluated when selected.	value: numeric													
orientation	Placement of the Radio Button: row or column. Default is row.	value: string													
align	Align radios along the cross axis. Vertical if orientation is set to row; horizontal if orientation is set to column.	value: string													
justify	Justify radios along the main axis. Horizontal if orientation is set to row; vertical if orientation is set to column.	value: string													
textPosition	Where to place the label text in relation to the Radio Button: top, right, bottom, or left. Default is right.	value: string													
enabled	If true, user is allowed to select a radio. Default is true.	value: boolean													

Note: If the component is disabled, scripts can still run on the component. For example, if you add a script action to the onClick event, the script will fire when the user clicks on the Radio Button.

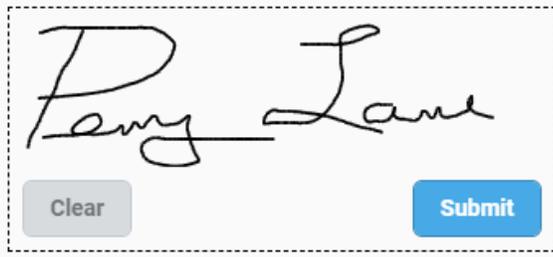
selectedIcon	<p>Settings for the appearance of the radio's icon when it is selected.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Path to the icon source, in format: library/IconName. For more information on icons, see the Images and Icons in Perspective page.</td> <td>value: string</td> </tr> <tr> <td>color</td> <td>Fill color settings to apply to the icon.</td> <td>object</td> </tr> <tr> <td colspan="3"> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Color of the icon when enabled. Can be a named color.</td> <td>value: string</td> </tr> <tr> <td>disabled</td> <td>Color of the icon when disabled. Can be a named color.</td> <td>value: string</td> </tr> </tbody> </table> </td> </tr> <tr> <td>style</td> <td>Sets a style for the icon. Full menu of style options is available. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Path to the icon source, in format: library/IconName. For more information on icons, see the Images and Icons in Perspective page.	value: string	color	Fill color settings to apply to the icon.	object	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Color of the icon when enabled. Can be a named color.</td> <td>value: string</td> </tr> <tr> <td>disabled</td> <td>Color of the icon when disabled. Can be a named color.</td> <td>value: string</td> </tr> </tbody> </table>			Name	Description	Property Type	enabled	Color of the icon when enabled. Can be a named color.	value: string	disabled	Color of the icon when disabled. Can be a named color.	value: string	style	Sets a style for the icon. Full menu of style options is available. You can also specify a style class .	object	object
Name	Description	Property Type																								
path	Path to the icon source, in format: library/IconName. For more information on icons, see the Images and Icons in Perspective page.	value: string																								
color	Fill color settings to apply to the icon.	object																								
<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Color of the icon when enabled. Can be a named color.</td> <td>value: string</td> </tr> <tr> <td>disabled</td> <td>Color of the icon when disabled. Can be a named color.</td> <td>value: string</td> </tr> </tbody> </table>			Name	Description	Property Type	enabled	Color of the icon when enabled. Can be a named color.	value: string	disabled	Color of the icon when disabled. Can be a named color.	value: string															
Name	Description	Property Type																								
enabled	Color of the icon when enabled. Can be a named color.	value: string																								
disabled	Color of the icon when disabled. Can be a named color.	value: string																								
style	Sets a style for the icon. Full menu of style options is available. You can also specify a style class .	object																								
unselectedIcon	<p>Settings for the appearance of the radio's icon when it is not selected.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Path to the icon source, in format: library/IconName. For more information on icons, see the Images, SVGs, and Icons in Perspective page.</td> <td>value: string</td> </tr> <tr> <td>color</td> <td>Fill color settings to apply to the icon.</td> <td>object</td> </tr> <tr> <td colspan="3"> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Color of the icon when enabled. Can be a named color.</td> <td>value: string</td> </tr> <tr> <td>disabled</td> <td>Color of the icon when disabled. Can be a named color.</td> <td>value: string</td> </tr> </tbody> </table> </td> </tr> <tr> <td>style</td> <td>Sets a style for the icon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Path to the icon source, in format: library/IconName. For more information on icons, see the Images, SVGs, and Icons in Perspective page.	value: string	color	Fill color settings to apply to the icon.	object	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Color of the icon when enabled. Can be a named color.</td> <td>value: string</td> </tr> <tr> <td>disabled</td> <td>Color of the icon when disabled. Can be a named color.</td> <td>value: string</td> </tr> </tbody> </table>			Name	Description	Property Type	enabled	Color of the icon when enabled. Can be a named color.	value: string	disabled	Color of the icon when disabled. Can be a named color.	value: string	style	Sets a style for the icon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	object
Name	Description	Property Type																								
path	Path to the icon source, in format: library/IconName. For more information on icons, see the Images, SVGs, and Icons in Perspective page.	value: string																								
color	Fill color settings to apply to the icon.	object																								
<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Color of the icon when enabled. Can be a named color.</td> <td>value: string</td> </tr> <tr> <td>disabled</td> <td>Color of the icon when disabled. Can be a named color.</td> <td>value: string</td> </tr> </tbody> </table>			Name	Description	Property Type	enabled	Color of the icon when enabled. Can be a named color.	value: string	disabled	Color of the icon when disabled. Can be a named color.	value: string															
Name	Description	Property Type																								
enabled	Color of the icon when enabled. Can be a named color.	value: string																								
disabled	Color of the icon when disabled. Can be a named color.	value: string																								
style	Sets a style for the icon. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																								
radioStyle	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.4 Click here to check out the other new features</p> </div> <p>Sets a style for the radio buttons. Full menu of style options is available including margin and padding, border, shape and miscellaneous.</p>	object																								
style	<p>Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</p>	object																								

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menu bar or by right clicking on the component.

Perspective - Signature Pad



On this page ...

- [Properties](#)
- [Scripting](#)
- [Examples](#)
 - [Example 1](#)
 - [Example 2](#)

Component Palette Icon:



The Signature Pad component enables users to draw a signature and “submit” it. Submitting a signature triggers a component event, enabling Ignition to do something with the signature.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type																											
enabled	Enables the canvas, clear button, and submit button. When enabled, component scripting functions for clearSignature and submitSignature are also enabled.	value: boolean																											
pad	Settings for the pad. <table border="1" data-bbox="253 1150 1338 1854"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>pen</td> <td>Settings for the pen. <table border="1" data-bbox="358 1283 1195 1520"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color used to draw the lines with the pen. You can set the color with a HEX, RGB, or HSL value. See also Color Selector Reference.</td> <td>value: color</td> </tr> <tr> <td>width</td> <td>Width (in pixels) of the line drawn by the pen.</td> <td>value: numeric</td> </tr> </tbody> </table> </td> <td>value: object</td> </tr> <tr> <td>canvas</td> <td>Settings for the canvas. <table border="1" data-bbox="358 1583 1195 1854"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>clearColor</td> <td>Color used to paint over the signature pad when cleared. Default is transparent. You can set the color with a HEX, RGB, or HSL value. See also Color Selector Reference.</td> <td>value: color</td> </tr> <tr> <td>style</td> <td>Sets a style for this property. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table> </td> <td>value: object</td> </tr> </tbody> </table>	Name	Description	Property Type	pen	Settings for the pen. <table border="1" data-bbox="358 1283 1195 1520"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color used to draw the lines with the pen. You can set the color with a HEX, RGB, or HSL value. See also Color Selector Reference.</td> <td>value: color</td> </tr> <tr> <td>width</td> <td>Width (in pixels) of the line drawn by the pen.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color used to draw the lines with the pen. You can set the color with a HEX, RGB , or HSL value. See also Color Selector Reference .	value: color	width	Width (in pixels) of the line drawn by the pen.	value: numeric	value: object	canvas	Settings for the canvas. <table border="1" data-bbox="358 1583 1195 1854"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>clearColor</td> <td>Color used to paint over the signature pad when cleared. Default is transparent. You can set the color with a HEX, RGB, or HSL value. See also Color Selector Reference.</td> <td>value: color</td> </tr> <tr> <td>style</td> <td>Sets a style for this property. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	clearColor	Color used to paint over the signature pad when cleared. Default is transparent. You can set the color with a HEX, RGB , or HSL value. See also Color Selector Reference .	value: color	style	Sets a style for this property. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	value: object	value: object
Name	Description	Property Type																											
pen	Settings for the pen. <table border="1" data-bbox="358 1283 1195 1520"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>color</td> <td>Color used to draw the lines with the pen. You can set the color with a HEX, RGB, or HSL value. See also Color Selector Reference.</td> <td>value: color</td> </tr> <tr> <td>width</td> <td>Width (in pixels) of the line drawn by the pen.</td> <td>value: numeric</td> </tr> </tbody> </table>	Name	Description	Property Type	color	Color used to draw the lines with the pen. You can set the color with a HEX, RGB , or HSL value. See also Color Selector Reference .	value: color	width	Width (in pixels) of the line drawn by the pen.	value: numeric	value: object																		
Name	Description	Property Type																											
color	Color used to draw the lines with the pen. You can set the color with a HEX, RGB , or HSL value. See also Color Selector Reference .	value: color																											
width	Width (in pixels) of the line drawn by the pen.	value: numeric																											
canvas	Settings for the canvas. <table border="1" data-bbox="358 1583 1195 1854"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>clearColor</td> <td>Color used to paint over the signature pad when cleared. Default is transparent. You can set the color with a HEX, RGB, or HSL value. See also Color Selector Reference.</td> <td>value: color</td> </tr> <tr> <td>style</td> <td>Sets a style for this property. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	clearColor	Color used to paint over the signature pad when cleared. Default is transparent. You can set the color with a HEX, RGB , or HSL value. See also Color Selector Reference .	value: color	style	Sets a style for this property. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	value: object																		
Name	Description	Property Type																											
clearColor	Color used to paint over the signature pad when cleared. Default is transparent. You can set the color with a HEX, RGB , or HSL value. See also Color Selector Reference .	value: color																											
style	Sets a style for this property. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																											
actionBar	Settings for the actionBar. <table border="1" data-bbox="253 1919 1338 1967"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property</th> </tr> </thead> <tbody> </tbody> </table>	Name	Description	Property	value: numeric																								
Name	Description	Property																											

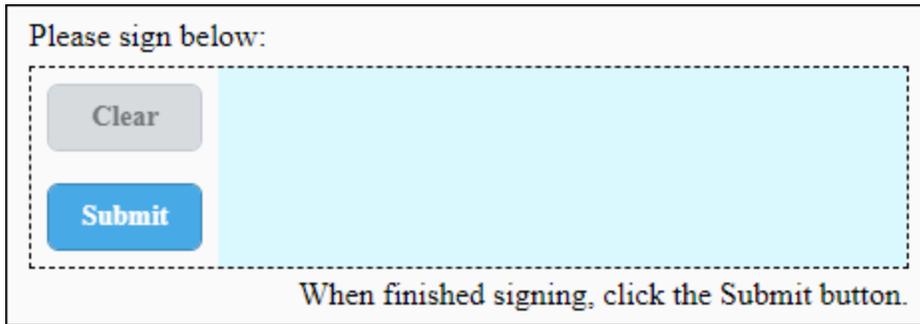
		Type															
position	Action bar position relative to the canvas. Options are top, bottom, left, or right. Default is bottom.	value: sting															
submitButton	Settings for the submit button.	object															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Text to display on the button. Default is submit.</td> <td>value: string</td> </tr> <tr> <td>enabled</td> <td>Enables submit button interaction. This does not affect the submitSignature component scripting function.</td> <td>value: boolean</td> </tr> <tr> <td>primary</td> <td>Toggle between the default primary and secondary button style.</td> <td>value: boolean</td> </tr> <tr> <td>style</td> <td>Sets a style for this property. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	text	Text to display on the button. Default is submit.	value: string	enabled	Enables submit button interaction. This does not affect the submitSignature component scripting function.	value: boolean	primary	Toggle between the default primary and secondary button style.	value: boolean	style	Sets a style for this property. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	
Name	Description	Property Type															
text	Text to display on the button. Default is submit.	value: string															
enabled	Enables submit button interaction. This does not affect the submitSignature component scripting function.	value: boolean															
primary	Toggle between the default primary and secondary button style.	value: boolean															
style	Sets a style for this property. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															
clearButton	Settings for the clear button.	object															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Text to display on the button. Default is clear.</td> <td>value: string</td> </tr> <tr> <td>enabled</td> <td>Enables clear button interaction. This does not affect the submitSignature component scripting function.</td> <td>value: boolean</td> </tr> <tr> <td>primary</td> <td>Toggle between the default primary and secondary button style.</td> <td>value: boolean</td> </tr> <tr> <td>style</td> <td>Sets a style for this property. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	text	Text to display on the button. Default is clear.	value: string	enabled	Enables clear button interaction. This does not affect the submitSignature component scripting function.	value: boolean	primary	Toggle between the default primary and secondary button style.	value: boolean	style	Sets a style for this property. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	
Name	Description	Property Type															
text	Text to display on the button. Default is clear.	value: string															
enabled	Enables clear button interaction. This does not affect the submitSignature component scripting function.	value: boolean															
primary	Toggle between the default primary and secondary button style.	value: boolean															
style	Sets a style for this property. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															
style	Sets a style for this property. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															
status	Settings for the status of the component.	object															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>touched</td> <td>True when the signature pad contains a signature.</td> <td>value: boolean</td> </tr> </tbody> </table>	Name	Description	Property Type	touched	True when the signature pad contains a signature.	value: boolean										
Name	Description	Property Type															
touched	True when the signature pad contains a signature.	value: boolean															
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															

Scripting

See the [Perspective - Signature Pad Scripting page](#) for the full list of scripting functions available for this component.

Examples

Example 1



In this example we set a few properties to customize the look of the Signature Pad. The buttons are on the left. The blue background is set with the `prop.canvas.clearColor` property, which enables the color to show up in our project but not get saved as part of the signature. Lastly, we put two Label components above and below the Signature Pad with signing instructions.

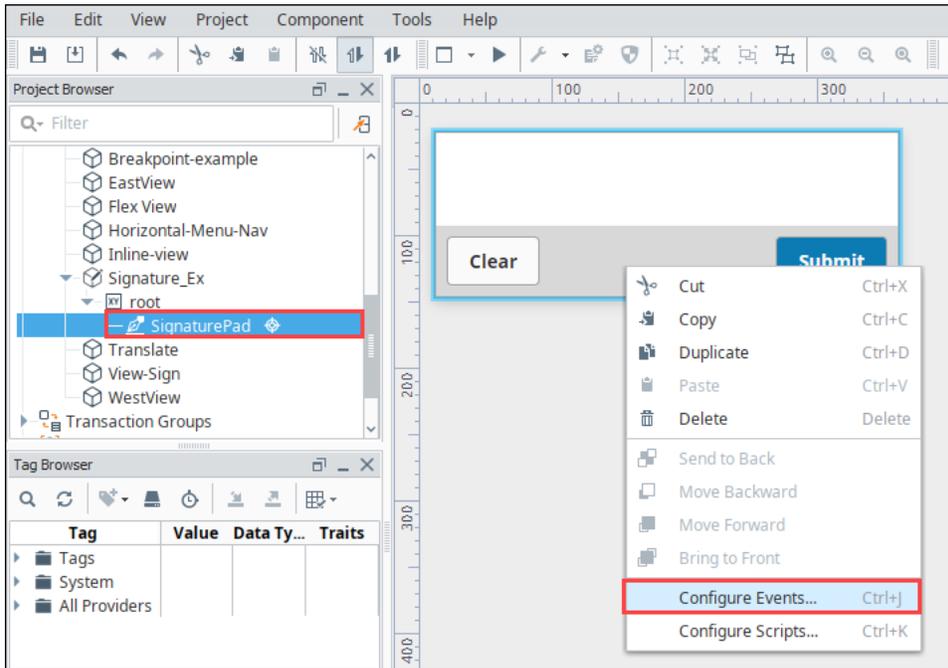
Property	Value
<code>props.enabled</code>	<code>true</code>
<code>props.canvas.clearColor</code>	<code>#DAF9FF</code>
<code>props.actionBar.position</code>	<code>left</code>
<code>props.actionBar.submitButton.text</code>	<code>Submit</code>
<code>props.actionBar.submitButton.enabled</code>	<code>true</code>
<code>props.actionBar.submitButton.primary</code>	<code>true</code>
<code>props.actionBar.submitButton.style.fontFamily</code>	<code>Merriweather</code>
<code>props.actionBar.clearButton.text</code>	<code>Clear</code>
<code>props.actionBar.clearButton.enabled</code>	<code>true</code>
<code>props.actionBar.clearButton.primary</code>	<code>true</code>
<code>props.actionBar.clearButton.style.fontFamily</code>	<code>Merriweather</code>
<code>props.style.borderStyle</code>	<code>dashed</code>
<code>props.style.borderWidth</code>	<code>1pt</code>

Example 2

The following example downloads the signature image when a user clicks the Submit button on the component.

To set this up, do the following:

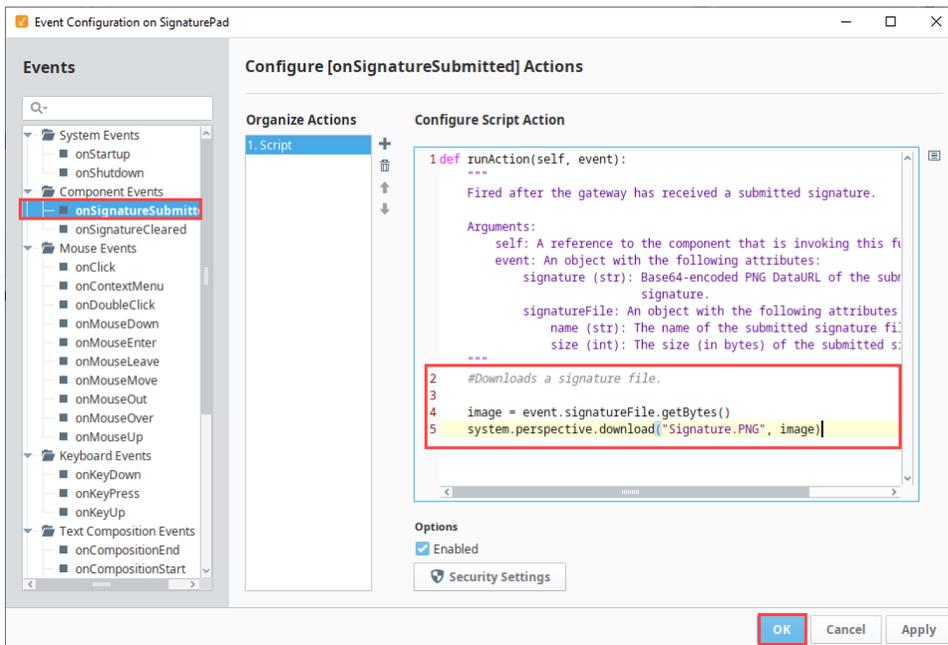
1. Drag a Signature component onto a Perspective view. Make sure it's a view that has a URL.
2. Right click on the component and select **Configure Events**.



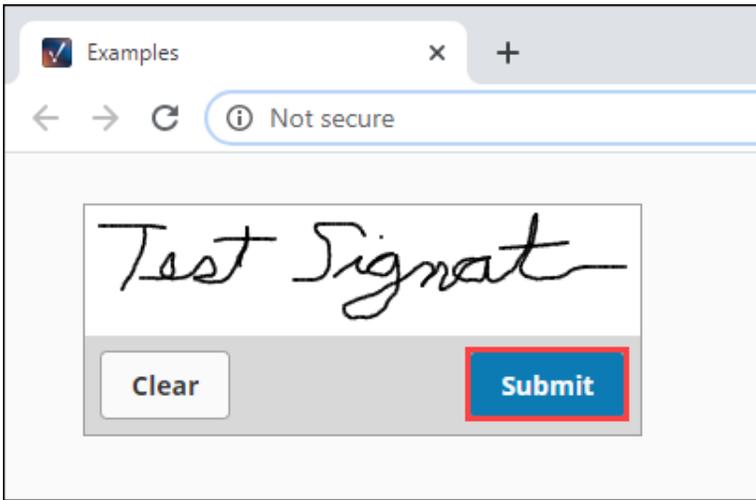
3. Select the **OnSignatureSubmitted** event .
4. Click the **Add** icon and select **Script**.
5. In the Configure Script Action box, add the following script:

```
#Downloads a signature file.

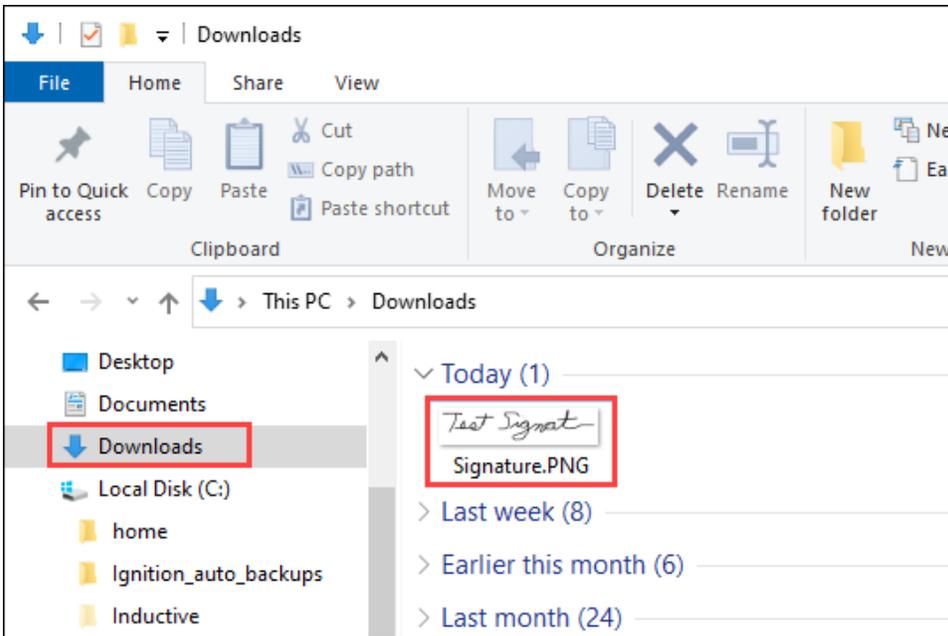
image = event.signatureFile.getBytes()
system.perspective.download("Signature.PNG", image)
```



6. Click **OK** to save the script.
7. Save your project.
8. Open a Perspective Session with the view that has the Signature Pad component.
9. Sign the component and click Submit.



10. An image file is saved to your computer. In this example, we're running Ignition on Windows. The file Signature.PNG appears in our Downloads folder:



Perspective - Signature Pad Scripting

This page details the various scripting, component, and extension functions available for [Perspective's Signature Pad](#) component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

On this page ...

- [Component Events](#)
 - [onSignatureSubmitted](#)
 - [onSignatureCleared](#)
- [Component Functions](#)
 - [.clearSignature\(\)](#)
 - [.submitSignature\(\)](#)
- [Extension Functions](#)

onSignatureSubmitted

Event is fired after the Gateway has received a submitted signature.

Note: This component event is designed to be used in tandem with a run action script. Within the script action, special properties and methods are available on the event object, which is passed to the script action as a parameter.

event.signature

- Object Path

event.signature

- Type

[String](#)

- Description

Base64-encoded PNG **DataURL** of the submitted signature.

event.signatureFile.name

- Object Path

event.signatureFile.name

- Type

[String](#)

- Description

A name for the signature file.

event.signatureFile.size

- Object Path

event.signatureFile.size

- Type

[Integer](#)

- Description

The size of the signature image file in bytes.

event.signatureFile.copyTo(filePath)

- Object Path

event.signatureFile.copyTo()

- Description

Saves the uploaded signature file at a location accessible to the Gateway.

- Parameters

`String` filePath - The path to where the file should be saved on the Gateway.

- Return

None

event.signatureFile.getBytes()

- Object Path

`event.signatureFile.getBytes()`

- Description

Returns a bytearray of the image, allowing the signature file to be saved from the session (with `system.perspective.download()`).

- Parameters

None

- Return

`byteArray` - The raw data of the incoming file.

event.signatureFile.getString()

- Object Path

`event.file.getString()`

- Description

Fetches the incoming file data and attempts to parse it as a string via UTF-8 (Eight-bit UCS Transformation Format) encoding. Defaults to UTF-8 (super common), but can use other character sets. Passed as a string, for example `getString("UTF_16BE")`.

- Parameters

None

- Return

`byteArray` - The raw data of the incoming signature file.

onSignatureCleared

This event is fired when the Gateway has received a signal that the signature has been cleared.

Note: This component event is designed to be used in tandem with a run action script. Within the script action, special properties and methods are available on the **event** object, which is passed to the script action as a parameter.

event

- Object Path

`event`

- Type

`Null`

- Description

An empty event object.

Component Functions

.clearSignature()

- Description

Clears the current signature on the component

- Parameters

None

- Return

Nothing

.submitSignature()

- Description

Submits the signature, triggering the onSignatureSubmitted component event.

- Parameters

None

- Return

Nothing

Extension Functions

This component does not have extension functions associated with it.

Perspective - Slider



Component Palette Icon:



On this page ...

- [Properties](#)
- [Component Events](#)
- [Example](#)

The Slider component lets the user drag an indicator along a scale to choose a value in a range. Enable the "show" and "interval" properties under "labels" to visually display the values within a range. The slider can be configured to orient horizontally or vertically with the "orientation" property.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type									
value	Value represented by slider handle. Current value of the slider.	value: numeric									
min	The minimum value for the slider scale: all the way left or down.	value: numeric									
max	The maximum value for the slider scale: all the way right or up.	value: numeric									
orientation	Specifies whether the slider track is aligned vertically or horizontally.	value: boolean									
step	Intervals along track at which a value may be set. Specifies the size of increments between values of the slider. Note: This does not force the value into that step size. Setting the slider value manually or through a binding will cause it to show the actual value.	value: numeric									
labels	Label settings along the track. <table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>show</td><td>If true, displays labels at periodic values along track.</td><td>value: boolean</td></tr><tr><td>interval</td><td>Interval at which to display periodic labels along track.</td><td>value: numeric</td></tr></tbody></table>	Name	Description	Property Type	show	If true, displays labels at periodic values along track.	value: boolean	interval	Interval at which to display periodic labels along track.	value: numeric	object
Name	Description	Property Type									
show	If true, displays labels at periodic values along track.	value: boolean									
interval	Interval at which to display periodic labels along track.	value: numeric									
enabled	Whether slider interaction is currently active. Note: If the component is disabled, scripts can still run on the component. For example, if you add a script action to the onClick event, the script will fire when the user clicks on the Slider.	value: boolean									
handleColor	Color of slider handle. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color									
trackColor	Color of slider track. See Color Selector .	color									
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object									

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menu bar or by right clicking on the component.

Example



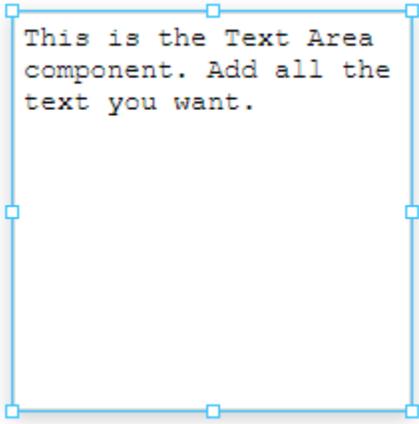
Property	Value
props.value	65
props.orientation	vertical
props.step	5
props.labels.show	true
props.labels.interval	10
props.handleColor	#8AFF8A
props.trackColor	#CCFFFF

Editor notes are only visible to logged in users

Was going through DOC-574 when I saw the thing below...not sure what it is, so I'm going to leave it alone for now.

[Component Test - WIP](#)

Perspective - Text Area



On this page ...

- [Properties](#)
- [Component Events](#)
- [Example](#)

Component Palette Icon:



Suitable for multi-line text display and editing. Will scroll vertically on demand. Horizontal scroll is determined by the "wrap" property.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
text	Text to display.	value: string
placeholder	Text displayed when Text Area is empty.	value: string
enabled	If true, user is allowed to alter text.	value: boolean
deferUpdates	When true, updates to props.text will be deferred until focus is lost or enter is pressed.	value: boolean
rejectUpdatesWhileFocused	When true, props.text will not accept updates from external sources while focused.	value: boolean
resize	Sets whether text is resizable, and if so, in which direction: none, both, horizontal, or vertical.	value: string
wrap	Specifies how to wrap text: hard, soft, or off. (Soft wrap is the break resulting from a line wrap or word wrap. Hard wrap is an intentional break, which moves text to the next line, or creates a new paragraph).	value: string
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object
spellcheck	<div style="border: 1px solid orange; padding: 5px; margin-top: 10px;">The following feature is new in Ignition version 8.1.27 Click here to check out the other new features</div>	value: boolean

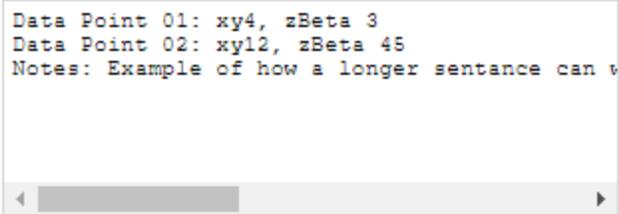
When true, text containing potential spelling errors will be underlined in red while the text is being edited in a launched session. Be aware there may be some slight behavioral differences in spellcheck error detection depending on the session browser.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Example



Data Point 01: xy4, zBeta 3
Data Point 02: xy12, zBeta 45
Notes: Example of how a longer sentence can wrap



Lab Data Entry:

This example shows the Text Area in two states: the top image is with text entered and the bottom image is without text entered.

Property	Value
props.text	Data Point 01: xy4, zBeta 3 Data Point 02: xy12, zBeta 45 Notes: Example of how a longer sentence can wrap (or not) around to the next line. Enter was pressed between each of these three lines.
props.placeholder	Lab Data Entry
props.wrap	off

Perspective - Text Field

This is a Text Field

Component Palette Icon:

 Text Field

On this page ...

- [User Interaction](#)
- [Properties](#)
- [Component Events](#)
- [Example](#)

The Text Field component is used for input of any single-line text. This component will accept any alpha-numeric input.

If you need a field that accepts multiple lines of text, see the [Perspective - Text Area](#) component. If you're looking for a numeric field, see the [Perspective - Numeric Entry Field](#) component.

User Interaction

The Text Field component properties have impact on the way a user can interact with a table in the runtime.

Interaction	Description
Enabled	When the enabled property is set to true, a user can edit the text field in the runtime. The user must double click on the field or press enter in order to edit the field. When done, press enter or leave the field, and the text field becomes non-editable again. When the enabled property is set to false, it is not editable even when it receives input focus.
General	The Text Field also supports the reject updates during edit feature. This feature ignores updates coming from property bindings while the component is being edited by a user.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
text	Text to display.	value: string
placeholder	Text displayed when Text Field is empty.	value: string
enabled	If true, user is allowed to alter text.	value: boolean
deferUpdates	When true, updates to props.text will be deferred until focus is lost or enter is pressed.	value: boolean
rejectUpdatesWhileFocused	When true, props.text will not accept updates from external sources while focused.	value: boolean
spellcheck	<div data-bbox="284 1785 1193 1869" style="border: 1px solid orange; padding: 5px;"><p>The following feature is new in Ignition version 8.1.27 Click here to check out the other new features</p></div> When true, text containing potential spelling errors will be underlined in red while the text is being edited in a launched session. Be aware there may be some slight behavioral differences in spellcheck error detection depending on the session browser.	value: boolean

styles	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object
--------	--	--------

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Example

A rectangular text input field with a light gray border. The text "Lab Results from QA: Chem analysis OK" is entered in a dark gray font.A rectangular text input field with a light gray border. The text "Single Line Entry" is displayed in a light gray font, serving as a placeholder.

This example shows the Text Field in two states: the top image is with text entered and the bottom image is without text entered.

Property	Value
props.text	Lab Results from QA: Chem Analysis OK
props.placeholder	Single Line Entry

Perspective - Toggle Switch



Component Palette Icon:



On this page ...

- [Properties](#)
- [Component Events](#)
- [Example](#)

The Toggle Switch represents a bit: on (selected) or off (not selected). By default, when the switch is selected the color is blue. It is gray when it is not selected. Logically, this component is very similar to the [Checkbox](#) component.

The following feature is new in Ignition version **8.1.2**
[Click here](#) to check out the other new features

The Toggle Switch component has three pre-configured [variants](#):

- No Text - Default layout with no text.
- Text Right - Layout with text on the right of the Toggle Switch.
- Text Left - Layout with text on the left of the Toggle Switch.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type												
selected	The selected state of the Toggle Switch.	value: boolean												
label	Settings for the label for the Toggle Switch. <table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody><tr><td>text</td><td>Text for the Toggle Switch. <div data-bbox="360 1499 1269 1583" data-label="Text"><p>The following feature is new in Ignition version 8.1.26 Click here to check out the other new features</p></div><p>Text can also be entered directly by deep selecting the Toggle Switch component, which enables inline editing. Changes are immediately reflected in the text property field.</p></td><td>string</td></tr><tr><td>position</td><td>Text position relative to the Toggle Switch: right or left.</td><td>value: boolean</td></tr><tr><td>style</td><td>Modify text style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td><td>object</td></tr></tbody></table>	Name	Description	Property Type	text	Text for the Toggle Switch. <div data-bbox="360 1499 1269 1583" data-label="Text"><p>The following feature is new in Ignition version 8.1.26 Click here to check out the other new features</p></div> <p>Text can also be entered directly by deep selecting the Toggle Switch component, which enables inline editing. Changes are immediately reflected in the text property field.</p>	string	position	Text position relative to the Toggle Switch: right or left.	value: boolean	style	Modify text style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	object
Name	Description	Property Type												
text	Text for the Toggle Switch. <div data-bbox="360 1499 1269 1583" data-label="Text"><p>The following feature is new in Ignition version 8.1.26 Click here to check out the other new features</p></div> <p>Text can also be entered directly by deep selecting the Toggle Switch component, which enables inline editing. Changes are immediately reflected in the text property field.</p>	string												
position	Text position relative to the Toggle Switch: right or left.	value: boolean												
style	Modify text style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object												
color	Color settings for the Toggle Switch when it is selected and unselected. <table border="1"><thead><tr><th>Name</th><th>Description</th><th>Property Type</th></tr></thead><tbody></tbody></table>	Name	Description	Property Type	object									
Name	Description	Property Type												

	selected	Color of the Toggle Switch when selected (on). Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color
	unselected	Color of the Toggle Switch when unselected (off). See Color Selector .	color
enabled	Whether the user should be allowed to alter the Toggle Switch's selected state. Default is true.		value: boolean
	<p>Note: If the component is disabled, scripts can still run on the component. For example, if you add a script action to the <code>onClick</code> event, the script will fire when the user clicks on the Toggle Switch.</p>		
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .		object

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Example

Off / On 

Off / On 

Image above shows both selected and deselected positions for the following properties.

Property	Value
props.label.text	Off / On
props.label.position	left
props.color.selected	#FF0000
props.color.unselected	#8AFF8A

Perspective - Navigation Palette

One of the most important aspects to consider when developing a Perspective application is a solid navigation design so the user knows where they are, where they've been, and where they are going.

The following navigational components provide you with design strategy options to navigate within a Perspective Session, a link pointing to a page containing the component's description, properties, and an example of how to configure it.

[In This Section ...](#)

Perspective - Horizontal Menu



Component Palette Icon:



On this page ...

- [Properties](#)
- [Scripting](#)
- [Examples](#)
 - [Example 1](#)
 - [Example 2](#)
 - [Example 3](#)

Horizontal Menu component enables you to build a menu structure by setting up multiple links to different page URLs from the component. The Horizontal Menu occupies a large amount of horizontal space and a comparatively small amount of vertical space. The menu starts with a list of root-level menu items that make up the main display area of the component.

If you have more menu items than will fit the width of the component, arrow buttons appear to enable you to scroll left and right through all menu items. The control can also be disabled as a whole.

Each menu item can be configured with a target that will serve as either a link to a page that should be shown (e.g. `"/my-page"`), or a link to an external web page (e.g. `"http://www.inductiveautomation.com/"`). They can also be given a list of child menu items that will show temporarily in a popup as the user is interacting with them. Additionally, menu items can be labeled, disabled, and be given an icon to show to the left of their label.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type																																	
items	Configure items representing the main menu items.	array																																	
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>Whether this option is currently enabled to perform its action or render its submenu.</td> <td>value: boolean</td> </tr> <tr> <td>target</td> <td>A URL (http://www.inductiveautomation.com/) or mounted path to a page (<code>/my-page</code>). If "items" is empty (no subtree to this item), this will navigate to that location.</td> <td>value: string</td> </tr> <tr> <td>items</td> <td>Configure items representing child menu items from this option. If defined, a submenu will branch from here with these options.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <tbody> <tr> <td>enabled</td> <td>Whether this option is currently enabled to perform its action or render its submenu.</td> <td>value: boolean</td> </tr> <tr> <td>target</td> <td>A URL (external) or mounted path to a page. If "items" is empty (no subtree to this item), this will navigate to that location.</td> <td>value: string</td> </tr> <tr> <td>items</td> <td>Configure items representing child menu items from this option. If defined, a submenu will branch from here with these options.</td> <td>array</td> </tr> <tr> <td>icon</td> <td>Icon image appended to the left of the menu item.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> </table> </td> <td></td> </tr> </tbody> </table> </td> <td></td> </tr> </tbody> </table>	Name	Description	Property Type	enabled	Whether this option is currently enabled to perform its action or render its submenu.	value: boolean	target	A URL (http://www.inductiveautomation.com/) or mounted path to a page (<code>/my-page</code>). If "items" is empty (no subtree to this item), this will navigate to that location.	value: string	items	Configure items representing child menu items from this option. If defined, a submenu will branch from here with these options.	object		<table border="1"> <tbody> <tr> <td>enabled</td> <td>Whether this option is currently enabled to perform its action or render its submenu.</td> <td>value: boolean</td> </tr> <tr> <td>target</td> <td>A URL (external) or mounted path to a page. If "items" is empty (no subtree to this item), this will navigate to that location.</td> <td>value: string</td> </tr> <tr> <td>items</td> <td>Configure items representing child menu items from this option. If defined, a submenu will branch from here with these options.</td> <td>array</td> </tr> <tr> <td>icon</td> <td>Icon image appended to the left of the menu item.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> </table> </td> <td></td> </tr> </tbody> </table>	enabled	Whether this option is currently enabled to perform its action or render its submenu.	value: boolean	target	A URL (external) or mounted path to a page. If "items" is empty (no subtree to this item), this will navigate to that location.	value: string	items	Configure items representing child menu items from this option. If defined, a submenu will branch from here with these options.	array	icon	Icon image appended to the left of the menu item.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> </table>	Name	Description	Property Type			
Name	Description	Property Type																																	
enabled	Whether this option is currently enabled to perform its action or render its submenu.	value: boolean																																	
target	A URL (http://www.inductiveautomation.com/) or mounted path to a page (<code>/my-page</code>). If "items" is empty (no subtree to this item), this will navigate to that location.	value: string																																	
items	Configure items representing child menu items from this option. If defined, a submenu will branch from here with these options.	object																																	
	<table border="1"> <tbody> <tr> <td>enabled</td> <td>Whether this option is currently enabled to perform its action or render its submenu.</td> <td>value: boolean</td> </tr> <tr> <td>target</td> <td>A URL (external) or mounted path to a page. If "items" is empty (no subtree to this item), this will navigate to that location.</td> <td>value: string</td> </tr> <tr> <td>items</td> <td>Configure items representing child menu items from this option. If defined, a submenu will branch from here with these options.</td> <td>array</td> </tr> <tr> <td>icon</td> <td>Icon image appended to the left of the menu item.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> </table> </td> <td></td> </tr> </tbody> </table>	enabled	Whether this option is currently enabled to perform its action or render its submenu.	value: boolean	target	A URL (external) or mounted path to a page. If "items" is empty (no subtree to this item), this will navigate to that location.	value: string	items	Configure items representing child menu items from this option. If defined, a submenu will branch from here with these options.	array	icon	Icon image appended to the left of the menu item.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> </table>	Name	Description	Property Type																	
enabled	Whether this option is currently enabled to perform its action or render its submenu.	value: boolean																																	
target	A URL (external) or mounted path to a page. If "items" is empty (no subtree to this item), this will navigate to that location.	value: string																																	
items	Configure items representing child menu items from this option. If defined, a submenu will branch from here with these options.	array																																	
icon	Icon image appended to the left of the menu item.	object																																	
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> </table>	Name	Description	Property Type																															
Name	Description	Property Type																																	

	<table border="1"> <tr> <td>path</td> <td>Shorthand path to icon source, in format: library /iconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons.</td> <td>value: string</td> </tr> <tr> <td>color</td> <td>Color of the icon. Can also assign the "fill" property in styles. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>color</td> </tr> </table>	path	Shorthand path to icon source, in format: library /iconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string	color	Color of the icon. Can also assign the "fill" property in styles. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color	
path	Shorthand path to icon source, in format: library /iconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string						
color	Color of the icon. Can also assign the "fill" property in styles. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color						
	<table border="1"> <tr> <td>label</td> <td>Text to display for this menu item label.</td> <td>value: string</td> </tr> <tr> <td>Style</td> <td>Sets a style for this item. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </table>	label	Text to display for this menu item label.	value: string	Style	Sets a style for this item. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	
label	Text to display for this menu item label.	value: string						
Style	Sets a style for this item. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object						
enabled	If true, the component is enabled/active. Default is true.	value: boolean						
itemStyle	Sets a style for all the items in the component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object						
style	Sets a style for the entire component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object						

Scripting

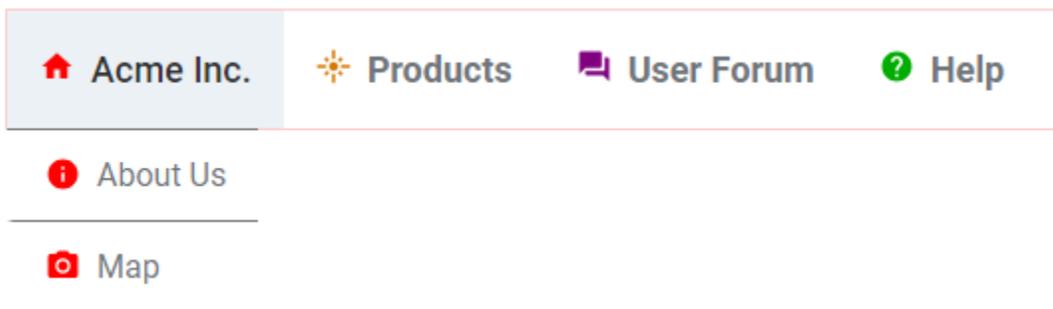
See the [Perspective - Horizontal Menu Scripting](#) page for the full list of scripting functions available for this component.

Examples

In addition to the examples below, learn more about the Horizontal Menu component on the [Navigating with the Horizontal Menu Component](#) page.

Example 1

This example shows a Horizontal Menu with four items. The first item also has two subitems. Each item links to a webpage. Icons are taken from the Material Design icons that can be found here: <https://material.io/tools/icons/>.



Top Level Properties

Property	Value
props.items.0.enabled	true
props.items.0.target	https:// (link to Acme home)
props.items.0.icon.path	material/home
props.items.0.icon.color	#FF0000
props.items.0.label	Acme Inc.

props.items.1.enabled	true
props.items.1.target	https:// (link to products)
props.items.1.icon	material/flare
props.items.1.color	#D97700
props.items.1.label	Products
props.items.2.enabled	true
props.items.2.target	https:// (link to forum)
props.items.2.icon.path	material/forum
props.items.2.icon.color	#800080
props.items.2.label	User Forum
props.items.3.enabled	true
props.items.3.target	https:// (link to help)
props.items.3.icon.path	material/help
props.items.3.icon.color	#00AC00
props.items.3.label	Help
props.style.borderStyle	solid
props.style.color	#FF4747
props.style.fontSize	18px
props.style.fontWeight	bold
props.style.borderWidth	1pc
props.style.borderColor	#FFCCCC

Sub Level Properties of Item 0

Property	Value
props.items.0.items.0.target	https:// (link to about)
props.items.0.items.0.icon.path	material/info
props.items.0.items.0.icon.color	#FF0000
props.items.0.items.0.label	About Us
props.items.0.items.0.target	/screen_2
props.items.0.items.0.enabled	true
props.items.0.items.1.target	https:// (link to about)
props.items.0.items.1.icon.path	material/local_see
props.items.0.items.1.icon.color	#FF0000
props.items.0.items.1.label	Map
props.items.0.items.1.target	https:// (link to map)
props.items.0.items.1.enabled	true

Example 2

This example shows a Horizontal Menu with four items that are linked to views within the Perspective project. The third item in the list, Field Offices, has four subitems. Each item links to a view for a different field office. Icons are taken from the Material Design icons that can be found here: <https://material.io/tools/icons/>.



WEST SITE



EAST SITE



FIELD OFFICES



OVERSEAS



Reservoir



Dock



Warehouse



Surveillance

Top Level Properties

Property	Value
props.items.0.enabled	true
props.items.0.target	/west
props.items.0.icon.path	material/explore
props.items.0.icon.color	#D97700
props.items.0.label	West Site
props.items.1.enabled	true
props.items.1.target	/View_East
props.items.1.icon	material/store
props.items.1.color	#D97700
props.items.1.label	East Site
props.items.2.enabled	true
props.items.2.icon.path	material/landscape
props.items.2.icon.color	#800080
props.items.2.label	Field Offices
props.items.3.enabled	true
props.items.3.target	/satellite
props.items.3.icon.path	material/satellite
props.items.3.icon.color	#00AC00
props.items.3.label	Overseas
props.style.borderStyle	solid
props.style.fontStyle	normal
props.style.fontVariant	small-caps
props.style.borderRadius	16
props.style.borderColor	#555555

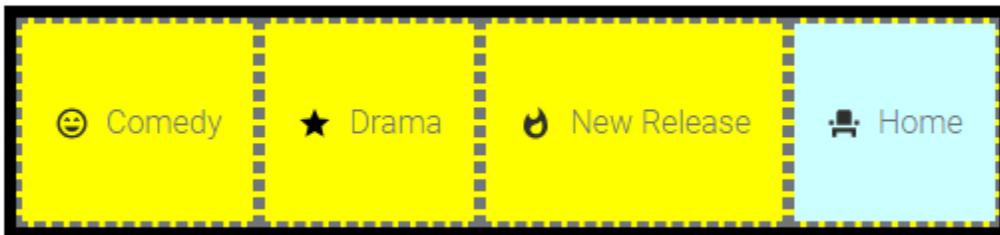
Sub Level Properties of Item 2

Property	Value
props.items.2.items.0.enabled	true

props.items.2.items.0.target	/reservoir
props.items.2.items.0.icon.path	material/rowing
props.items.2.items.0.icon.color	#00ACAC
props.items.2.items.0.label	Reservoir
props.items.2.items.1.enabled	true
props.items.2.items.1.target	/dock
props.items.2.items.1.icon.path	material/directions_boat
props.items.2.items.1.icon.color	#9E6635
props.items.2.items.1.label	Dock
props.items.2.items.2.enabled	true
props.items.2.items.2.target	/warehouse
props.items.2.items.2.icon.path	material/local_shipping
props.items.2.items.2.icon.color	#0000AC
props.items.2.items.2.label	Warehouse
props.items.2.items.2.enabled	true
props.items.2.items.2.target	/surveillance
props.items.2.items.2.icon.path	material/videocam
props.items.2.items.2.icon.color	#AAAAAA
props.items.2.items.2.label	Warehouse

Example 3

This example shows a Horizontal Menu with four items.



Top Level Properties

Property	Value
props.items.0.enabled	true
props.items.0.icon.path	material /sentiment_very_satisfied
props.items.0.icon.color	#D97700
props.items.0.label	Comedy
props.items.1.enabled	true
props.items.1.icon	material/star
props.items.1.label	Drama
props.items.2.enabled	true
props.items.2.icon.path	material/whatshot
props.items.2.label	New Release

props.items.3.enabled	true
props.items.3.icon.path	material/event_seat
props.items.3.label	Home
props.items.3.style.backgroundColor	#CCFFFF
props.itemStyle.backgroundColor	#FFFF00
props.itemStyle.borderColor	dashed
props.style.borderColor	solid
props.style.borderWidth	6px
props.style.borderColor	#000000

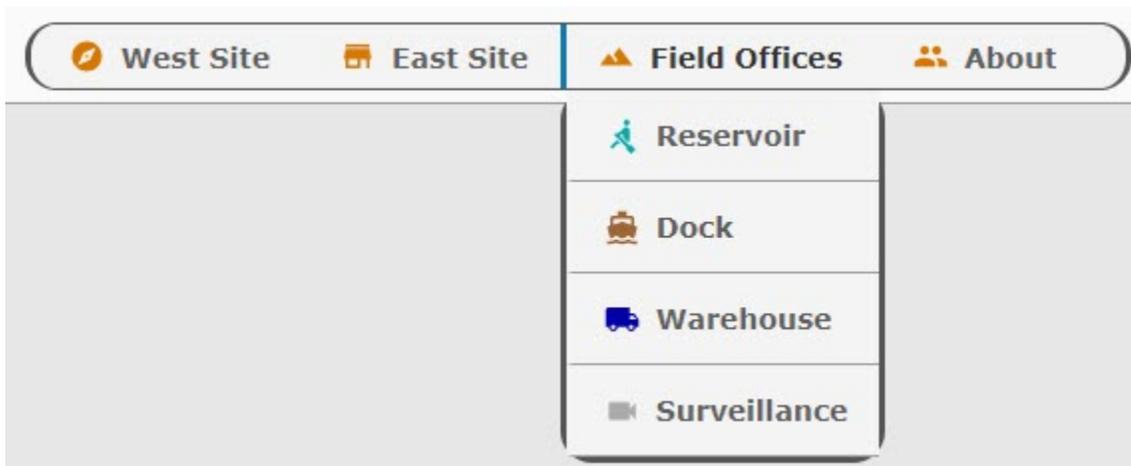
Navigating with the Horizontal Menu Component

The Horizontal Menu component enables you to build a menu structure by setting up multiple links to different page URLs from the component. Our example has a menu with links to three internal pages and one external page on the Internet.

This example shows a Horizontal Menu with four items. Three items are linked to views within the Perspective project and the fourth is linked to a website. The third item in the list, Field Offices, has four subitems.

On this page ...

- [Initial Project Setup](#)
- [Set Up a Header View](#)
- [Configure the Tabs in the Horizontal Menu](#)
 - [Configure the Field Offices Tab in the Horizontal Menu](#)



Initial Project Setup

To begin with we have created three views: **WestView**, **EastView**, and **Welcome**. They are each coordinate type views. As we create each new view, we checked the Page URL property and added a page name.

View Name	Page URL
WestView	/west_page
EastView	/east_page
Welcome	/

New View

Name: WestView ✓

Root Container Type: Coordinate

Page URL

Page URL: /west_page ✓

Buttons: Cancel, Create View

On Page Configuration, you'll note that each page is already set up with a Primary view.

Perspective [Examples_2019_11_and_Later] 1.0.7-rc1 (b2019121216) Create New View

Page Configuration

- Shared settings
- / → Test/Welcome
- /east_page → EastView
- /west_page → WestView

Configuration for /:

- Page URL: /
- Primary View: Test/Welcome
- Corner Priority: left-right, top-bottom, inherited

Now we're ready to start building the navigation.

Set Up a Header View

The first thing we'll set up is a view that will hold the Horizontal Menu component. We'll use this view as a header for our pages within this project.

1. In the Project Browser, right click on Views and select the **NewFolder** option. Name the folder "Header".
2. Right click on the Header folder and select the **NewView** option.

Name: **Horizontal-Menu-Nav**
 Layout: **Coordinate**
 Page URL: unchecked

3. Click **Create View**.
4. In the Property Editor for the view, set the **defaultSize** property as follows:

width: **800**
height: **50**

5. Drag a Horizontal Menu component onto the view.
6. In the Property Editor, set the Position Properties as follows:

Property	Value
position.x	15
position.y	10
position.width	550
position.height	30

7. In the Property Editor, scroll down to style and click the **ModifyStyle**  icon.

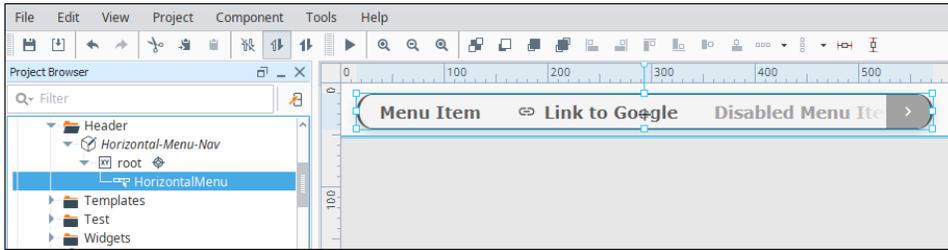
- a. Expand the Text section and set the style options as follows:

Property	Value
props.style.fontWeight	bolder
props.style.fontFamily	Verdana
props.style.fontSize	14px

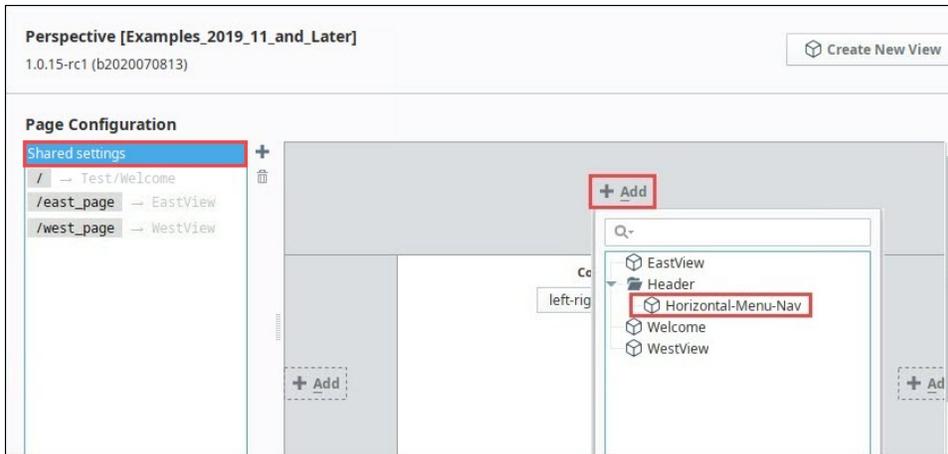
- b. Expand the Border section and set the style options as follows:

Property	Value
props.style.borderStyle	solid
props.style.borderColor	#555555
props.style.borderRadius	16px
All Corners	(selected)

Your Designer will look like this at this point:



8. Open Page Configuration by clicking on the **Settings**  icon at the bottom left of the Designer window.
9. Under Page Configuration, click on **Shared Settings**.
10. In the header part of the page mockup, click on the **Add**  icon.
11. Select the Horizontal-Menu-Nav view from the dropdown.



12. Click **OK**. The Horizontal-Menu-Nav view will now appear at the top of all pages in the project.

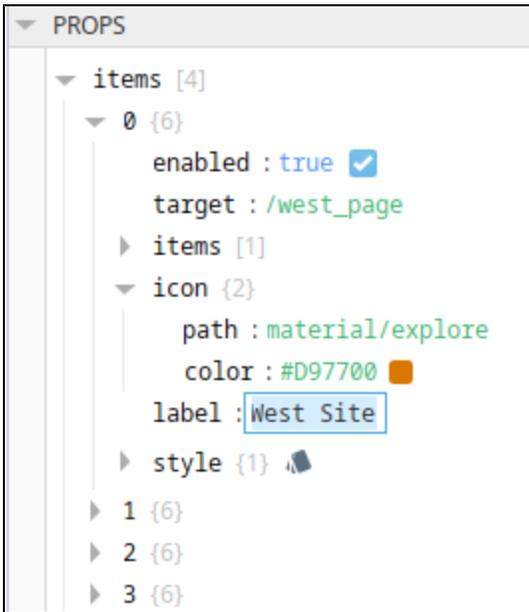
Configure the Tabs in the Horizontal Menu

Now let's set the properties for each of the four tabs in the Horizontal Menu. Each tab will have a display a name and an icon and will have a target page or website to open when clicked.

1. In the Property Editor, expand the items property and set the following for item 0:

Property	Value
props.items.0.enabled	true
props.items.0.target	/west_page
props.items.0.icon.path	material/explore
props.items.0.icon.color	#D97700
props.items.0.label	West Site

The Property Editor will look like this:



2. In the Property Editor, set the following for item 1:

Property	Value
props.items.1.enabled	true
props.items.1.target	/east_page
props.items.1.icon	material/store
props.items.1.color	#D97700
props.items.1.label	East Site

3. In the Property Editor, set the following for item 2:

Note: Do not set a props.items.2.target property value for this tab because we will set up dropdown tabs in the next section.

Property	Value
props.items.2.enabled	true
props.items.2.icon.path	material/landscape
props.items.2.icon.color	#D97700
props.items.2.label	Field Offices

4. In the Property Editor, set the following for item 3:

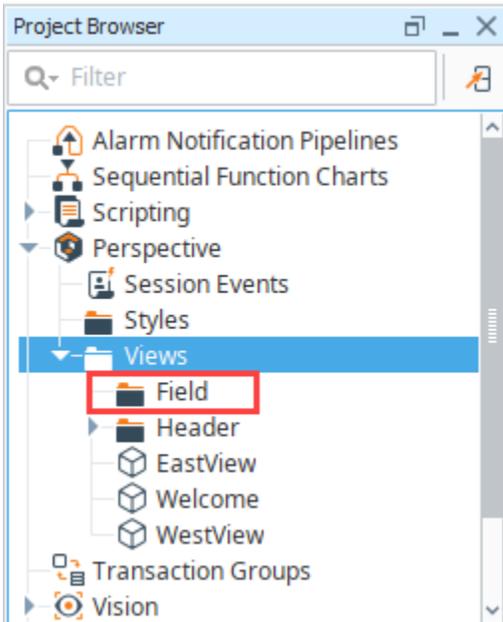
Note: This tab uses a website as its target, therefore it does not need to target a page within Perspective.

Property	Value
props.items.3.enabled	true
props.items.3.target	http://inductiveautomation.com/about/
props.items.3.icon.path	material/people
props.items.3.icon.color	#D97700
props.items.3.label	About

Configure the Field Offices Tab in the Horizontal Menu

The third tab in the Horizontal Menu is titled "Field Offices." Instead of navigating to one page, this tab has a dropdown menu with four options on it: Reservoir, Dock, Warehouse, and Surveillance.

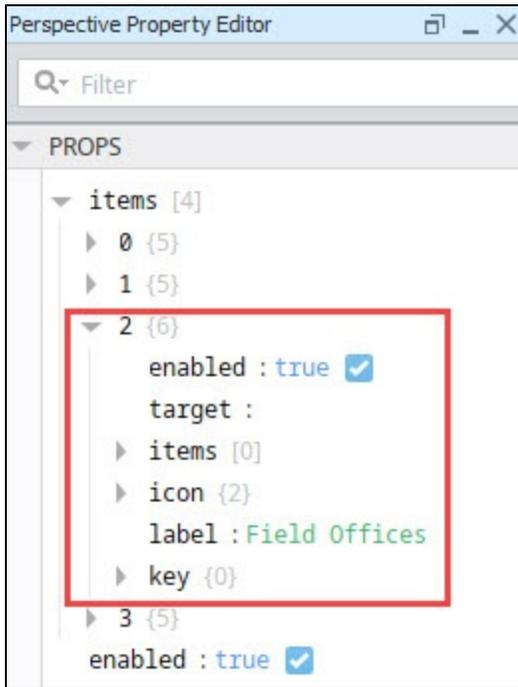
1. To start, we created a new folder in Views called **Field**.



2. Within the Field folder, we create four views: **Reservoir**, **Dock**, **Warehouse**, and **Surveillance**. Make them Coordinate types and set up a page URL for each.

View Name	Page URL
Reservoir	/reservoir_page
Dock	/dock_page
Warehouse	/warehouse_page
Surveillance	/surveillance_page

3. Open the Horizontal-Menu-Nav view and select the Horizontal Menu component.
4. In the Property Editor, expand the properties for Item 2.



5. Select the props.items.2.items property.
6. Click the Add  Add Array Element... icon to add four items.
7. Set the properties for Item 0 as follows:

Property	Value
props.items.2.items.0.enabled	true
props.items.2.items.0.target	/reservoir_page
props.items.2.items.0.icon.path	material/rowing
props.items.2.items.0.icon.color	#00ACAC
props.items.2.items.0.label	Reservoir

The Property Editor will look like this:



8. Now we'll do the same for the other three items. Set the properties for Item 1 as follows:

Property	Value
props.items.2.items.1.enabled	true
props.items.2.items.1.target	/dock_page
props.items.2.items.1.icon.path	material/directions_boat
props.items.2.items.1.icon.color	#9E6635
props.items.2.items.1.label	Dock

9. Set the properties for Item 2 as follows:

Property	Value
props.items.2.items.2.enabled	true
props.items.2.items.2.target	/warehouse_page
props.items.2.items.2.icon.path	material/local_shipping
props.items.2.items.2.icon.color	#0000AC

props.items.2.items.2.label	Warehouse
-----------------------------	-----------

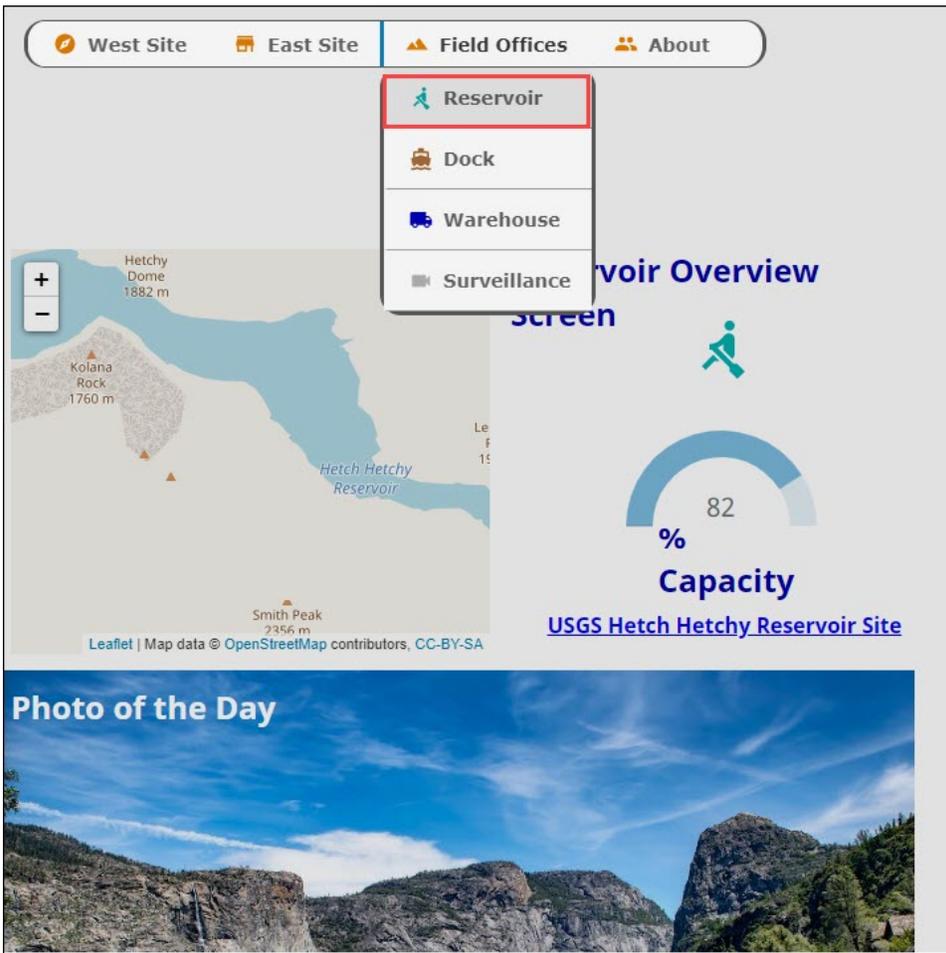
10. Set the properties for item 3 as follows:

Property	Value
props.items.2.items.3.enabled	true
props.items.2.items.3.target	/surveillance_page
props.items.2.items.3.icon.path	material/videocam
props.items.2.items.3.icon.color	#AAAAAA
props.items.2.items.3.label	Surveillance

11. Save your project.

12. Click **Tools > Launch Perspective > Launch Session**.

13. Click on the tabs in the header to view different pages. For our example, we have put a few components on each view. Here is an example of what the Field Offices > Reservoir page might look like:



Perspective - Horizontal Menu Scripting

This page details the various scripting, component, and extension functions available for [Perspective's Horizontal Menu](#) component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

On this page ...

- [Component Events](#)
 - [onItemClicked](#)
- [Component Functions](#)
- [Extension Functions](#)

onItemClicked

Fired when an item is selected.

event.enabled

- Object Path

event.enabled

- Type

[Boolean](#)

- Description

Whether the item interacted with is enabled.

event.label

- Object Path

event.label

- Type

[String](#)

- Description

Text to display for this option.

event.path

- Object Path

event.label

- Type

[List](#)

- Description

A list containing the item indexes leading to the item that was clicked.

event.target

- Object Path

event.target

- Type

[String](#)

- Description

A URL (external) or a mounted path to a page.

Component Functions

This component does not have component functions associated with it.

Extension Functions

This component does not have extension functions associated with it.

Perspective - Link



Component Palette Icon:



On this page ...

- [Properties](#)
- [Component Events](#)

The Link component allows users to create a hyperlink that points to a destination such as a page, view, resource, or mount path that they can quickly navigate to. Links are easily identifiable because they typically have a different color font than the rest of the content, and an underline when you hover over them. To configure a link, enter the URL for the destination, and the name of the link in the 'text' property.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

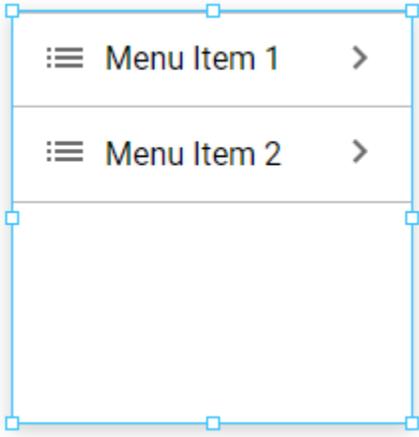
Name	Description	Property Type
url	A URL, URL fragment, or a Page URL given to a Page. If the target is a Page, its Page URL can be found in the Page Configuration settings of the project. See Page URLs .	value: string
text	Text to display in the link.	value: string
target	Specifies where to display the linked URL. Options are: <ul style="list-style-type: none">• self for the current tab/context• tab or 'blank' for a new tab• parent for the parent frame tab/context• top for the full body of the window. Otherwise, supports standard w3c values for anchor link target attributes.	value: string
style	Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Perspective - Menu Tree



On this page ...

- [Properties](#)
- [Scripting](#)
- [Examples](#)
 - [Example 1 - Basic Menu Tree](#)
 - [Example 2 - Tree Navigation](#)

Component Palette Icon:



The Menu Tree component can be configured to allow users to navigate pages in a Perspective Session. The Menu Tree defines a hierarchical view of information that can be configured to expand submenu branches and menu items. The subitems can be further expanded to expose more subitems if any exist, and collapsed to hide subitems.

Each menu item has its own path, for example, "Western Region/CA/San Jose" that determines its location in the Menu Tree. The Separation Character property (by default its forward-slash), dictates how the paths are broken up.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Prop Type																											
items	Configure items representing nodes in a subtree from this option. If defined, a submenu will branch from here with these options.	array																											
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>target</td> <td>A url (external), or mounted path to a page. If "items" is empty (no subtree to this item), this will navigate to that location.</td> <td>value: string</td> </tr> <tr> <td>items</td> <td>Configure items representing nodes in a subtree from this option. If defined, a submenu will branch from here with these options.</td> <td>array</td> </tr> <tr> <td>navIcon</td> <td>Icon image appended to the right of the menu item.</td> <td>object</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Shorthand path to icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons.</td> <td>value: string</td> </tr> <tr> <td>color</td> <td>Color of the icon. Can also assign the "fill" property in styles. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>color</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>label</td> <td>Menu item label.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	target	A url (external), or mounted path to a page. If "items" is empty (no subtree to this item), this will navigate to that location.	value: string	items	Configure items representing nodes in a subtree from this option. If defined, a submenu will branch from here with these options.	array	navIcon	Icon image appended to the right of the menu item.	object		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Shorthand path to icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons.</td> <td>value: string</td> </tr> <tr> <td>color</td> <td>Color of the icon. Can also assign the "fill" property in styles. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>color</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Shorthand path to icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string	color	Color of the icon. Can also assign the "fill" property in styles. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color		label	Menu item label.	object	
Name	Description	Property Type																											
target	A url (external), or mounted path to a page. If "items" is empty (no subtree to this item), this will navigate to that location.	value: string																											
items	Configure items representing nodes in a subtree from this option. If defined, a submenu will branch from here with these options.	array																											
navIcon	Icon image appended to the right of the menu item.	object																											
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Shorthand path to icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons.</td> <td>value: string</td> </tr> <tr> <td>color</td> <td>Color of the icon. Can also assign the "fill" property in styles. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector.</td> <td>color</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Shorthand path to icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string	color	Color of the icon. Can also assign the "fill" property in styles. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color																			
Name	Description	Property Type																											
path	Shorthand path to icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string																											
color	Color of the icon. Can also assign the "fill" property in styles. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color																											
label	Menu item label.	object																											

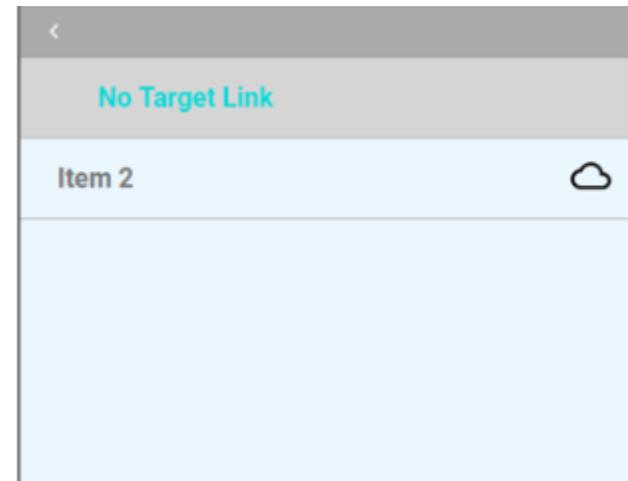
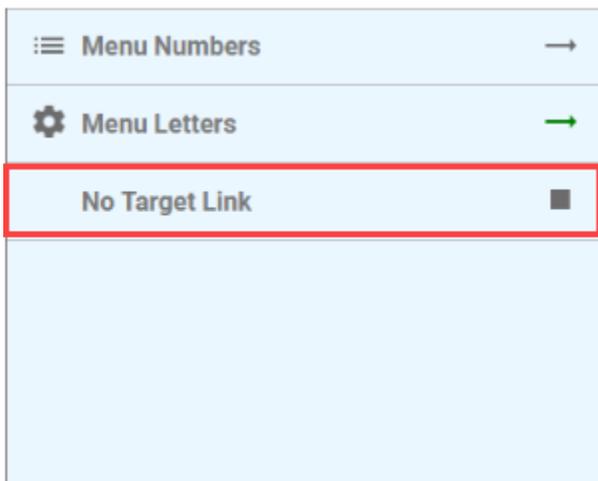
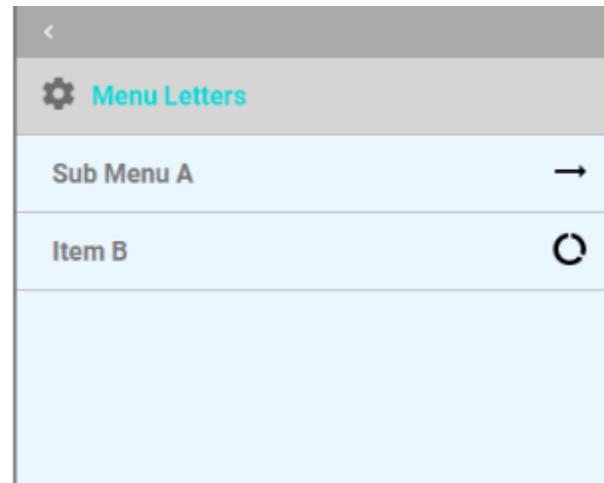
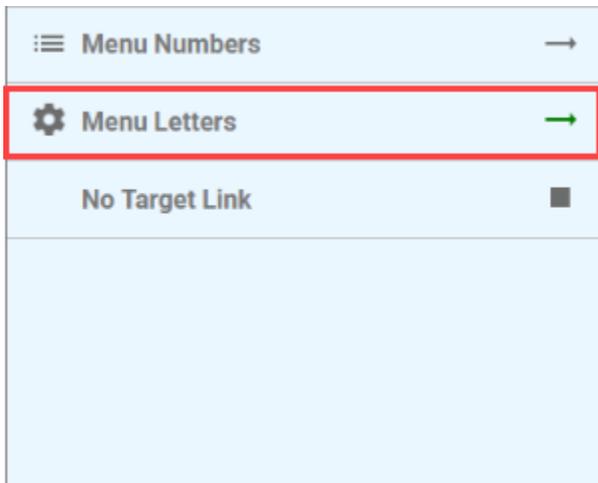
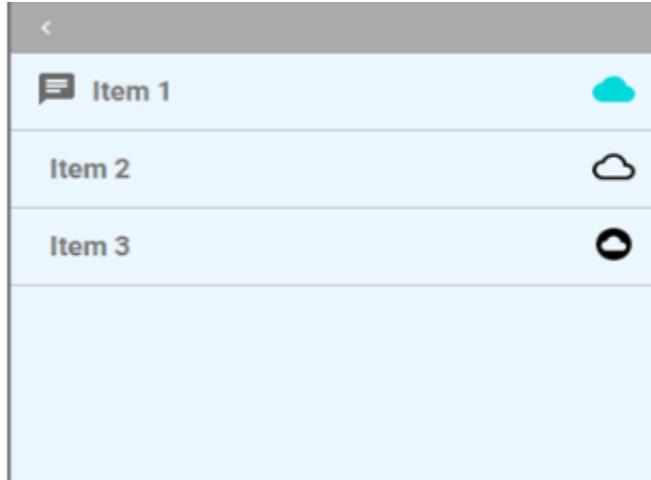
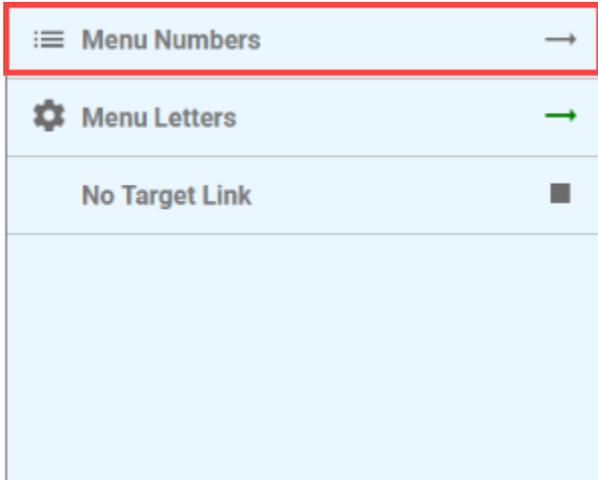
	Name	Description	Property Type						
	text	Text to display for this option.	value: string						
	icon	Image icon added to the right of the label text.							
		<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>path</td> <td>Shorthand path to icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	path	Shorthand path to icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string	
Name	Description	Property Type							
path	Shorthand path to icon source, in format: library/IconName. The materials icon library is a the primary source for icons, see https://fonts.google.com/icons?selected=Material+Icons .	value: string							
visible		Whether this option should be displayed in the menu tree.	value: boolean						
enabled		If true, this option is currently enabled to perform its action or render its submenu.	value: boolean						
showHeader		Whether to display this option's text as a header/title for its submenu.	value: boolean						
style		Sets a style for this item. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object						
resetOnClick		<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.8 Click here to check out the other new features</p> </div> <p>While true, clicking on the item will cause the Menu Tree to reset back to its root level. Useful in cases where you want to provide your users with a quick way to return to the start of the menus.</p>	value: boolean						
backActionText		<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.24 Click here to check out the other new features</p> </div> <p>Text to display in prompt to go back to the previous menu. This configuration property will override the text set to display in the root backActionText property. If left blank, menu items default to the root backActionText.</p>	value: string						
layoutAlignment		Specifies which side of the root menu is aligned to. Submenu slides in from the opposite side.	value: boolean						
enabled		If true, this component is currently enabled to perform its actions.	value: boolean						
itemStyle		Sets style options for the menu tree component. You can also specify a style class .	object						
headerStyle		Sets style options for the menu header. You can also specify a style class .	object						
backActionStyle		StySets the style options to display in the root menu. You can also specify a style class .	object						
backActionText		Text to display in prompt to go back to the root menu. You can also specify a style class .	value: string						
style		Sets a style for this component. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object						

Scripting

See the [Perspective - Menu Tree Scripting](#) page for the full list of scripting functions available for this component.

Examples

Example 1 - Basic Menu Tree



You can use links to go to URL links outside your project or internal to your project.

For this example assume you have some screens defined in the project page configuration (home gear). Assume the URL's for those are as follows:

- /screen_1
- /screen_2
- /screen_3
- /screen_b
- /screen_a1
- /screen_a2

Note: Images on the left side of the Menu Tree are associated with the label.icon.path
 Images on the right side of the Menu Tree are associated with the navIcon.path
 target property is the link to the screen to navigate to.

Top Level Properties

Property	Value
props.items.0.navIcon.path	material/trending_flat
props.items.0.navIcon.color	#6C6C6C
props.items.0.label.text	Menu Numbers
props.items.0.label.icon.path	material/list
props.items.0.label.showHeader	false
props.items.1.navIcon.path	material/trending_flat
props.items.1.navIcon.color	#008000
props.items.1.label.text	Menu Letters
props.items.1.label.icon.path	material/settings
props.items.1.label.showHeader	true
props.items.2.navIcon.path	material/stop
props.items.2.navIcon.color	#FF8A8A
props.items.2.label.text	No Target Link
props.items.2.label.icon.path	material/stop_screen_sharing
props.items.2.label.showHeader	true

Sub Level Properties of Item 0

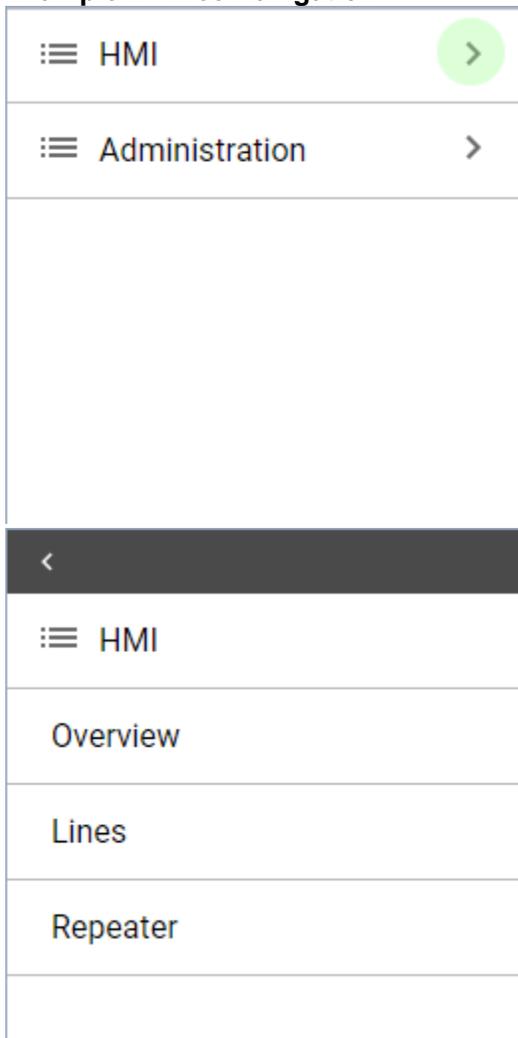
Property	Value
props.items.0.items.0.target	/screen_1
props.items.0.items.0.navIcon.path	material/cloud
props.items.0.items.0.navIcon.color	#00D9D9
props.items.0.items.0.label.text	Item 1
props.items.0.items.0.label.icon.path	material/chat
props.items.0.items.0.label.showHeader	true
props.items.0.items.1.target	/screen_2
props.items.0.items.1.navIcon.path	material/cloud_queue
props.items.0.items.1.label.text	Item 2
props.items.0.items.1.label.showHeader	true
props.items.0.items.2.target	/screen_3
props.items.0.items.2.navIcon.path	material/cloud_circle
props.items.0.items.2.label.text	Item 3
props.items.0.items.2.label.showHeader	true

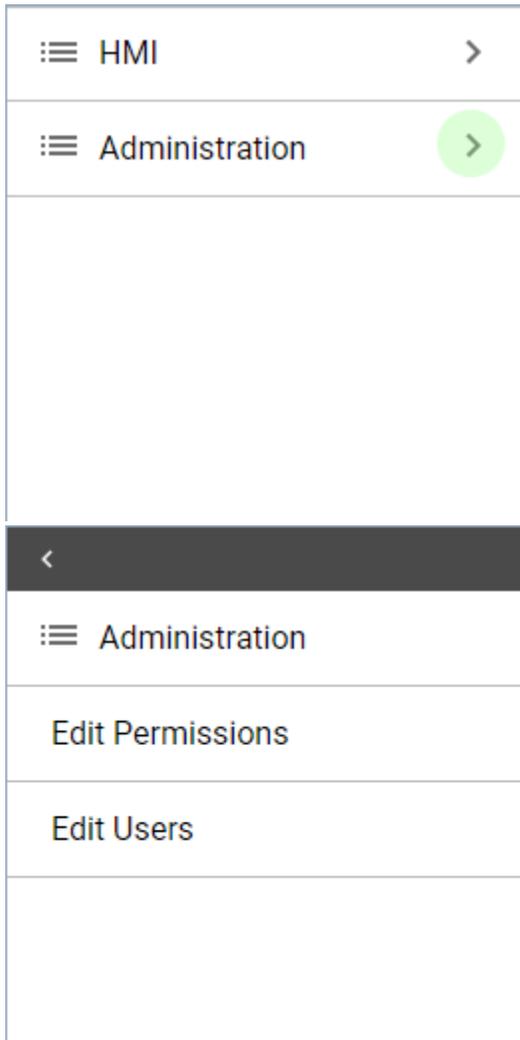
Sublevel Properties of Item 1

Property	Value

props.items.1.items.0.navIcon.path	material/trending_flat
props.items.1.items.0.label.text	Sub Menu A
props.items.0.items.0.label.showHeader	true
props.items.1.items.1.target	/screen_b
props.items.1.items.1.navIcon.path	material/data_usage
props.items.1.items.1.label.text	Item B
props.items.1.items.0.items.0.target	/screen_a1
props.items.1.items.0.items.0.navIcon.path	material/group
props.items.1.items.0.items.0.navIcon.color	#6C6C6C
props.items.1.items.0.items.0.label.text	Item A 1
props.items.1.items.0.items.0.label.showHeader	true
props.items.1.items.0.items.1.target	/screen_a2
props.items.1.items.0.items.1.navIcon.path	material/group_add
props.items.1.items.0.items.1.navIcon.color	#6C6C6C
props.items.1.items.0.items.1.label.text	Item A 2
props.items.1.items.0.items.1.label.showHeader	true

Example 2 - Tree Navigation





The **items** property for this component is complex. It is an "object" type that has as many levels as you want. Here is an example of the JSON used to create the two level image above. You can copy the text below and paste it into the items property of your Menu Tree component. No code is needed to make this component navigate.

This example assumes several things in order to work:

- Target fields that are blank (HMI and Administration) do not navigate when clicked.
- There are several pages created in the project:
 - /overview
 - /lines
 - /packaging
 - /userPermissions
 - /userEdit
- There are icons stored in ignition.
 - A hamburger menu icon with a path of "menu_list"
 - A greater than symbol with a path of "chevron_right"

PROPS

- items [2]
 - 0 {5}
 - target :
 - items [3]
 - 0 {5}
 - target : /overview
 - items [0]
 - + Add Array Element...
 - navIcon {1}
 - path :
 - label {2}
 - text : Overview
 - icon {1}
 - path :
 - showHeader : true
 - 1 {5}
 - 2 {5}
 - + Add Array Element...
 - navIcon {2}
 - path : chevron_right
 - color : #6C6C6C
 - label {2}
 - text : HMI
 - icon {1}
 - showHeader : true
 - 1 {5}
 - + Add Array Element...

items Property

```
[
  {
    "target": "",
    "items": [
      {
        "target": "/overview",
        "items": [],
        "navIcon": {
          "path": ""
        },
        "label": {
          "text": "Overview",
          "icon": {
            "path": ""
          }
        }
      },
      {
        "showHeader": true
      }
    ],
    {
      "target": "/lines",
      "items": [],
      "navIcon": {
```

```

        "path": ""
    },
    "label": {
        "text": "Lines",
        "icon": {
            "path": ""
        }
    },
    "showHeader": true
},
{
    "target": "/packaging",
    "items": [],
    "navIcon": {
        "path": "Packaging"
    },
    "label": {
        "text": "Repeater",
        "icon": {
            "path": ""
        }
    },
    "showHeader": true
}
],
"navIcon": {
    "path": "chevron_right",
    "color": "#6C6C6C"
},
"label": {
    "text": "HMI",
    "icon": {
        "path": "menu_list"
    }
},
"showHeader": true
},
{
    "target": "",
    "items": [
        {
            "target": "/userPermissions",
            "items": [],
            "navIcon": {
                "path": ""
            },
            "label": {
                "text": "Edit Permissions",
                "icon": {
                    "path": ""
                }
            }
        },
        "showHeader": true
    ],
    {
        "target": "/userEdit",
        "items": [],
        "navIcon": {
            "path": ""
        },
        "label": {
            "text": "Edit Users",
            "icon": {
                "path": ""
            }
        },
        "showHeader": true
    }
],
"navIcon": {
    "path": "chevron_right",

```

```
    "color": "#6C6C6C"
  },
  "label": {
    "text": "Administration",
    "icon": {
      "path": "menu_list"
    }
  },
  "showHeader": true
}
]
```

Perspective - Menu Tree Scripting

This page details the various scripting, component, and extension functions available for [Perspective's Menu Tree](#) component.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

On this page ...

- [Component Events](#)
 - [onItemClicked](#)
- [Component Functions](#)
- [Extension Functions](#)

onItemClicked

Fired when an item is selected.

event.enabled

- Object Path

event.enabled

- Type

[Boolean](#)

- Description

Whether the item interacted with is enabled.

event.label

- Object Path

event.label

- Type

[String](#)

- Description

Text to display for this option.

event.path

- Object Path

event.label

- Type

[List](#)

- Description

A list containing the item indexes leading to the item that was clicked.

event.target

- Object Path

event.target

- Type

[String](#)

- Description

A URL (external) or a mounted path to a page.

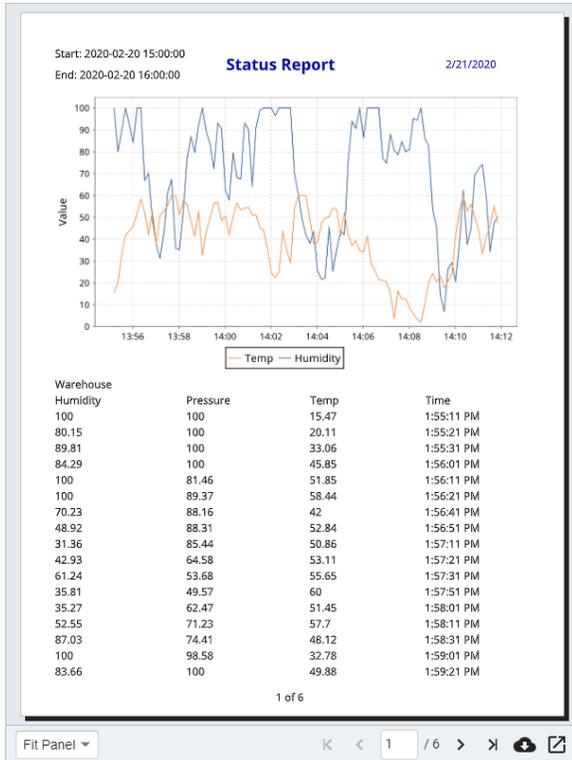
Component Functions

This component does not have component functions associated with it.

Extension Functions

This component does not have extension functions associated with it.

Perspective - Report Viewer



- On this page ...**
- [Interface Elements](#)
 - [Properties](#)
 - [Component Events](#)
 - [Example](#)
 - [Configuring the Report Viewer](#)

Component Palette Icon:



The Report Viewer component allows embedding reports from the Reporting Module into a Perspective views. The Reporting Module must be installed to use the Report Viewer component. To configure the Report Viewer, you must first create a report and provide the Path of that report in the 'source' property of the component.

You can specify any parameters that you are using in the report as values under the params property, but the parameter names must match. The values specified will be used instead of the default report parameters.

Interface Elements



Icon	Name	Description
	Zoom	Controls the zoom level of the PDF view.
	Pager	Displays the current page and allows the user to increment or decrement pages.
	Download	Downloads the report. This feature was changed in Ignition version 8.1.12: As of 8.1.12, after clicking or pressing "Download" the button is disabled until the user either report or cancels the download.

	Popout	Opens the report in a new window.
--	--------	-----------------------------------

Properties

Name	Description	Property Type						
source	Path of the report (case sensitive) from the Reporting Module that this component should display. For example, your Report source path might be: "Folder/ReportName". Right-click on the report in the Project Browser to Copy Path.	value: string						
params	An object that can be enhanced with report parameters to use in a report.	object						
page	The current page number displayed with the report. This property is updated as the pages are viewed.	value: numeric						
pageCount	A read-only property that provides the total number of pages in the report.	value: numeric						
zoomLevel	A number representing the desired zoom level as a percentage of the report width. The Fit Panel option is simply 1 as opposed to a percentage value.	value: numeric						
allowDownload	If set to 'true,' an icon will be added to the control bar at the bottom of the component that will download the report as a PDF file when clicked. The value of this property is always treated as "false" when the report viewer component is viewed from the iOS and Android Perspective Mobile apps.	value: boolean						
downloadFilename	<div data-bbox="279 991 1188 1075" style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.20 Click here to check out the other new features</p> </div> <p>A filename used for the report if downloaded. Filename will default to the name of the report if this property is left blank.</p>	value: string						
allowOpenInTab	If set to 'true,' an icon will be added to the control bar at the bottom of the component that will open a temporary PDF file in a new tab. The value of this property is always treated as "false" when the report viewer component is viewed from Perspective Workstation, as well as the iOS and Android Perspective Mobile apps.	value: boolean						
controlStyle	An object containing CSS style properties that are applied to the control bar and controls across the bottom.	object						
<table border="1"> <thead> <tr> <th data-bbox="305 1381 370 1402">Name</th> <th data-bbox="402 1381 516 1402">Description</th> <th data-bbox="954 1381 1084 1402">Property Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="305 1428 370 1449">classes</td> <td data-bbox="402 1428 922 1449">Styles defined in the project to be applied to this component.</td> <td data-bbox="954 1428 1003 1449">object</td> </tr> </tbody> </table>		Name	Description	Property Type	classes	Styles defined in the project to be applied to this component.	object	
Name	Description	Property Type						
classes	Styles defined in the project to be applied to this component.	object						
style	An object containing CSS style properties that are applied to the background display of the component.	object						

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

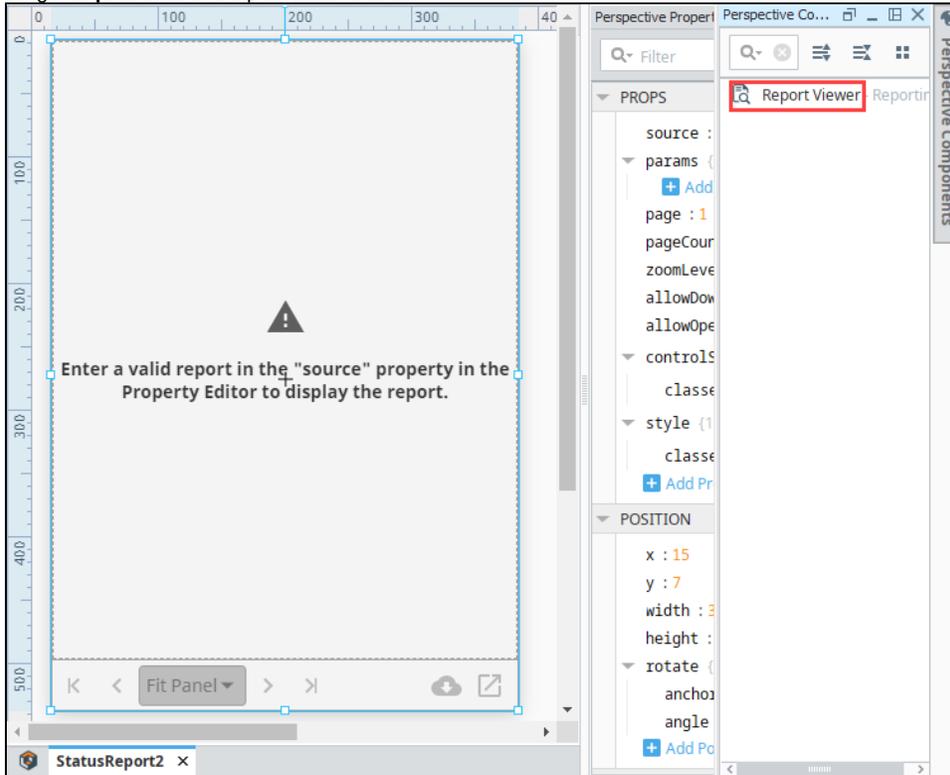
Example

Configuring the Report Viewer

The Report Viewer component provides an easy way to view and print reports in a Perspective View. Let's configure the Report Viewer for a report.

Note: This example requires that you already have a report created in your Gateway. You can learn more about creating reports [here](#).

1. Drag a **Report Viewer** component into a view.



2. Enter the Path of the report you want to view in the **'source'** property of the Property Editor. The data from your report will immediately load into the Report Viewer.

Tip: You can right-click on any report in the designer Project Browser to Copy the full Path of the report.

100 200 300 400 500 600
Perspective Property Editor

Start: 3:18:29 PM
Status Report
2/21/2020

End: 3:26:29 PM

— Temp — Humidity

Warehouse	Pressure	Temp	Time
100	100	15.47	1:55:11 PM
80.15	100	20.11	1:55:21 PM
89.81	100	33.06	1:55:31 PM
84.29	100	45.85	1:56:01 PM
100	81.46	51.85	1:56:11 PM
100	89.37	58.44	1:56:21 PM
70.23	88.16	42	1:56:41 PM
48.92	88.31	52.84	1:56:51 PM
31.36	85.44	50.86	1:57:11 PM
42.93	64.58	53.11	1:57:21 PM
61.24	53.68	55.65	1:57:31 PM
35.81	49.57	60	1:57:51 PM
35.27	62.47	51.45	1:58:01 PM
52.55	71.23	57.7	1:58:11 PM
87.03	74.41	48.12	1:58:31 PM
100	98.58	32.78	1:59:01 PM
83.66	100	49.88	1:59:21 PM

1 of 6

Fit Panel ▾
⏪ < 1 / 6 > ⏩
📄 🗑️

PROPS

source : Status Report

params {0}

+ Add Object Member...

page : 1

pageCount : 6

zoomLevel : 1 ▾

allowDownload : true

allowOpenInTab : true

controlStyle (1) 🗑️

classes :

style (1) 🗑️

classes :

+ Add Property...

POSITION

CUSTOM

META

- Parameters added during report creation are provided as properties in the Report Viewer. Add them under the **'params'** object. (The example below uses the EndDate and StartDate parameters). The parameter names must match exactly to the parameters in your Report Resource, and will override any default values set in the Report Resource.

The screenshot displays a report viewer interface. The main content area shows a line chart titled "Status Report" for the date 2/21/2020. The chart plots "Value" (0 to 100) against time (13:56 to 14:12). Two lines are shown: "Temp" (blue) and "Humidity" (orange). Below the chart is a data table with columns: Office Humidity, Pressure, Temp, and Time. The table contains 16 rows of data. The report viewer includes a "Fit Panel" dropdown and navigation controls at the bottom.

The Perspective Property Editor on the right shows the configuration for the report. The "params" object is highlighted with a red box and contains the following properties:

```

source : Status Report
params (2)
  EndDate : 2020-02-21 16:00:00
  StartDate : 2021-02-21 15:50:00
page : 5
pageCount : 6
zoomLevel : 1
allowDownload : true
allowOpenInTab : true
controlStyle (1)
  classes :
style (1)
  classes :
Add Property...
POSITION
CUSTOM
META

```

Perspective - Symbols Palette

The following symbol components provide you with design options for visualizing HMI. Click on the Symbol component name for a link pointing to a page containing the component's description, properties, and an example of how to configure it.

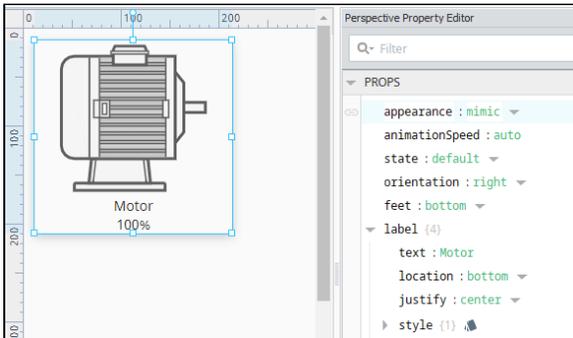
In This Section ...

Perspective - Motor

The following feature is new in Ignition version **8.1.0**
[Click here](#) to check out the other new features

On this page ...

- [Properties](#)
- [Component Events](#)
- [Example](#)



Component Palette Icon:



An animated component that looks like a motor. Full menu of [style options](#) is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a [style class](#).

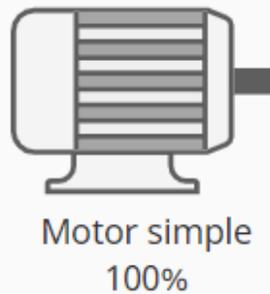
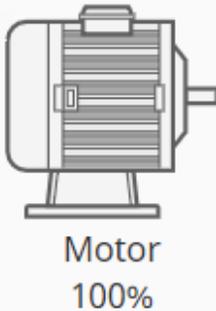
Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
appearance	Options for the appearance of the component. Options are auto, p&id, mimic, and simple. If set to auto, the value is taken from the Perspective Session Property symbols.autoAppearance.	value: string
animationSpeed	The speed of the animation as a percent. Set to "0" to turn off animation. If set to auto, the setting is taken from the Perspective Session Property symbols.autoAnimationSpeed.	value: numeric
state	State of the animation. Built-in options are default, running, stopped, or faulted. Default is default.	value: string

The following feature is new in Ignition version **8.1.26**
[Click here](#) to check out the other new features

Built-in states and new states can be configured and applied on the [Project Properties > Symbols](#) page.



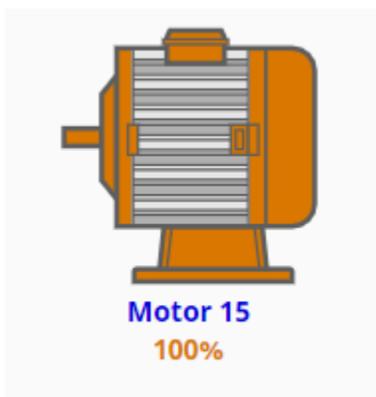
orientation	Orientation of the component. Options are top, bottom, left, or right. Default is right.	color															
label	Label settings for the component. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Text for the label.</td> <td>value: string</td> </tr> <tr> <td>location</td> <td>Label position relative to the motor: top, bottom, left, right, or hidden. Default is bottom.</td> <td>value: string</td> </tr> <tr> <td>justify</td> <td>Label text justification: left, center, right, or auto. Default is center.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Modify label text style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	text	Text for the label.	value: string	location	Label position relative to the motor: top, bottom, left, right, or hidden. Default is bottom.	value: string	justify	Label text justification: left, center, right, or auto. Default is center.	value: string	style	Modify label text style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	object
Name	Description	Property Type															
text	Text for the label.	value: string															
location	Label position relative to the motor: top, bottom, left, right, or hidden. Default is bottom.	value: string															
justify	Label text justification: left, center, right, or auto. Default is center.	value: string															
style	Modify label text style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															
value	Value settings for the component. <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Value to display as text.</td> <td>value: string</td> </tr> <tr> <td>location</td> <td>Value location relative to the motor: top, bottom, left, right, or hidden. Default is bottom.</td> <td>value: string</td> </tr> <tr> <td>justify</td> <td>Value text justification: center, left, or right. Default is center.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Sets a style for the value. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	text	Value to display as text.	value: string	location	Value location relative to the motor: top, bottom, left, right, or hidden. Default is bottom.	value: string	justify	Value text justification: center, left, or right. Default is center.	value: string	style	Sets a style for the value. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	object
Name	Description	Property Type															
text	Value to display as text.	value: string															
location	Value location relative to the motor: top, bottom, left, right, or hidden. Default is bottom.	value: string															
justify	Value text justification: center, left, or right. Default is center.	value: string															
style	Sets a style for the value. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															
style	Sets a style for this cylindrical tank. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Example



Property	Value

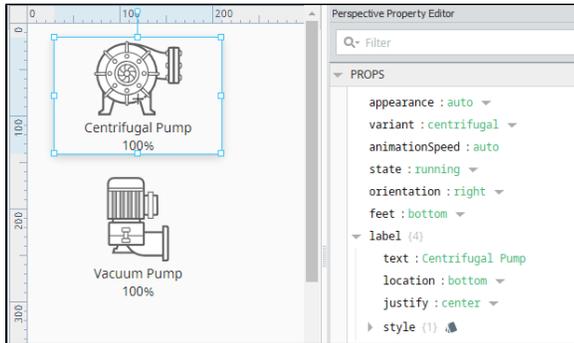
props.orientation	left
props.label.text	Motor 15
props.label.style.color	#0000D9
props.label.style.fontWeight	bold
props.label.style.fontFamily	Noto Sans
props.value.style.color	#D97700
props.value.style.fontWeight	bold
props.value.style.fontFamily	Noto Sans
props.style.fill	#D97700

Perspective - Pump

The following feature is new in Ignition version **8.1.0**
[Click here](#) to check out the other new features

On this page ...

- [Properties](#)
- [Component Events](#)
- [Example](#)



Component Palette Icon:



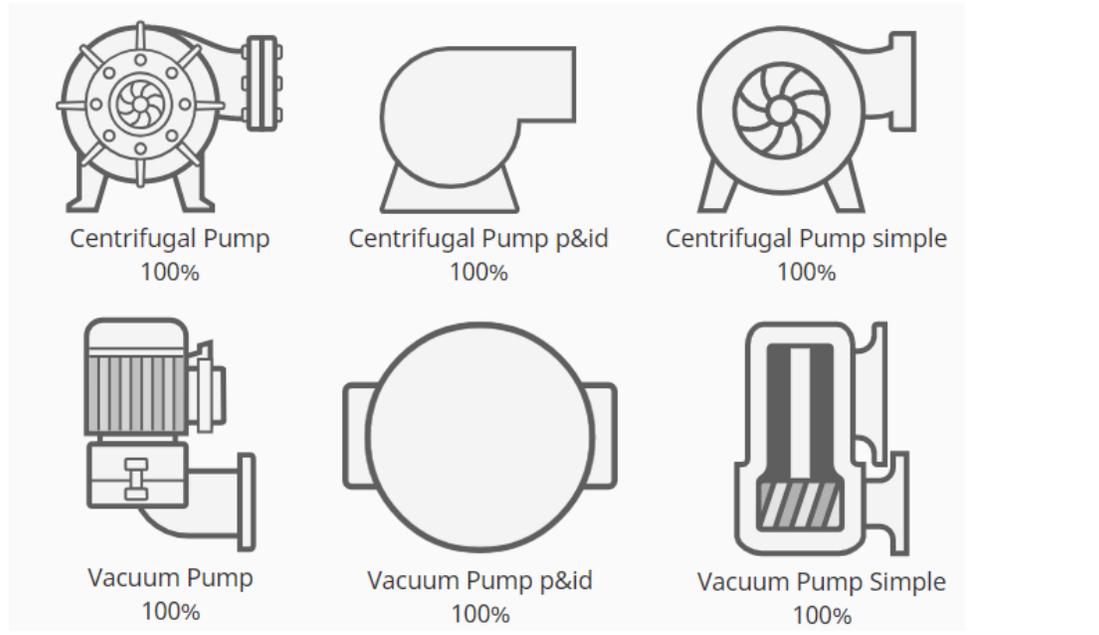
An animated component that looks like a pump. Full menu of [style options](#) is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a [style class](#). The Pump has two pre-configured [variants](#):

- Centrifugal - Component appearance is that of a centrifugal pump.
- Vacuum - Component appearance is that of a vacuum pump.

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
appearance	Options for the appearance of the component. Options are auto, p&id, mimic, and simple. If set to auto, the value is taken from the Perspective Session Property symbols.autoAppearance.	value: string



variant	Variant of pump to display. Options are centrifugal or vacuum. Default is centrifugal.	value: string															
animationSpeed	The speed of the animation as a percent. Set to "0" to turn off animation. If set to auto, then the animationSpeed setting is taken from the Perspective Session Properties .	value: numeric															
state	State of the animation. Built-in options are default, running, stopped, or faulted. Default is default. <div style="border: 1px solid orange; padding: 5px; margin: 10px 0;">The following feature is new in Ignition version 8.1.26 Click here to check out the other new features</div> Built-in states and new states can be configured and applied on the Project Properties > Symbols page.	value: string															
orientation	Orientation of the pump. Options are top, bottom, left, or right. Default is right.	color															
feet	Feet location for the pump. Options are top, bottom, left, or right. Default is bottom.																
label	Label settings for the component. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Name</th> <th style="width: 60%;">Description</th> <th style="width: 25%;">Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Text for the label.</td> <td>value: string</td> </tr> <tr> <td>location</td> <td>Label position relative to the pump: top, bottom, left, right, or hidden. Default is bottom.</td> <td>value: string</td> </tr> <tr> <td>justify</td> <td>Label text justification: center, left or right. Default is center.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Modify label text style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	text	Text for the label.	value: string	location	Label position relative to the pump: top, bottom, left, right, or hidden. Default is bottom.	value: string	justify	Label text justification: center, left or right. Default is center.	value: string	style	Modify label text style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	object
Name	Description	Property Type															
text	Text for the label.	value: string															
location	Label position relative to the pump: top, bottom, left, right, or hidden. Default is bottom.	value: string															
justify	Label text justification: center, left or right. Default is center.	value: string															
style	Modify label text style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															
value	Value settings for the component. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Name</th> <th style="width: 60%;">Description</th> <th style="width: 25%;">Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Value to display as text.</td> <td>value: numeric</td> </tr> <tr> <td>location</td> <td>Value location relative to the pump: top, bottom, left, right, or hidden. Default is bottom.</td> <td>value: string</td> </tr> <tr> <td>justify</td> <td>Value text justification: center, left, or right. Default is center.</td> <td>value: string</td> </tr> </tbody> </table>	Name	Description	Property Type	text	Value to display as text.	value: numeric	location	Value location relative to the pump: top, bottom, left, right, or hidden. Default is bottom.	value: string	justify	Value text justification: center, left, or right. Default is center.	value: string	object			
Name	Description	Property Type															
text	Value to display as text.	value: numeric															
location	Value location relative to the pump: top, bottom, left, right, or hidden. Default is bottom.	value: string															
justify	Value text justification: center, left, or right. Default is center.	value: string															

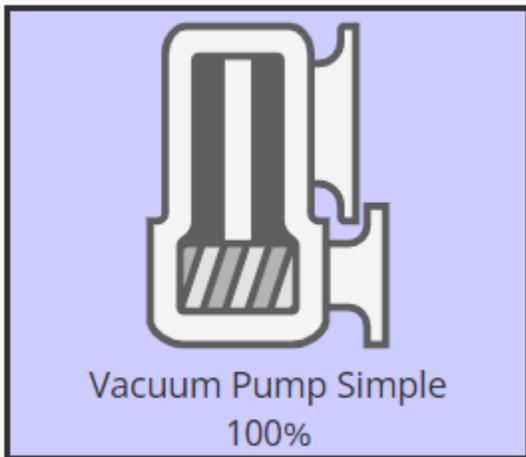
	style	Sets a style for the value. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object
style		Sets a style for this cylindrical tank. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Example



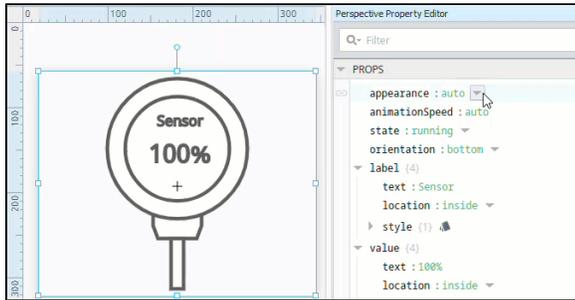
Property	Value
props.appearance	simple
props.variant	vacuum
props.label.text	Vacuum Pump Simple
props.style.backgroundColor	#CCCCFF
props.style.borderStyle	solid

Perspective - Sensor

The following feature is new in Ignition version **8.1.0**
[Click here](#) to check out the other new features

On this page ...

- [Properties](#)
- [Component Events](#)



Component Palette Icon:



An animated component that looks like a sensor. Full menu of [style options](#) is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a [style class](#).

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
appearance	Options for the appearance of the component. Options are auto, p&i,d, mimic, and simple. If set to auto, the value is taken from the Perspective Session Property symbols.autoAppearance.	value: string
animationSpeed	The speed of the animation as a percent. Set to "0" to turn off animation. If set to auto, the setting is taken from the Perspective Session Property symbols.autoAnimationSpeed.	value: numeric
state	State of the animation. Built-in options are default, running, stopped, or faulted. <div style="border: 1px solid orange; padding: 5px; margin: 10px 0;">The following feature is new in Ignition version 8.1.26 Click here to check out the other new features</div> Built-in states and new states can be configured and applied on the Project Properties > Symbols page.	value: string

orientation	Orientation of the sensor. Options are top, bottom, left, or right. Default is right.	value: string															
reverse Flow	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p>This feature was changed in Ignition version 8.1.13:</p> </div> <p>This property was removed in 8.1.13.</p> <p>Whether or not to reverse the direction of the flow. Default is false.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Note: This property has no effect on the Sensor component.</p> </div>	value: boolean															
label	<p>Label settings for the component.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Name</th> <th style="width: 60%;">Description</th> <th style="width: 25%;">Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Text for the label.</td> <td>value: string</td> </tr> <tr> <td>location</td> <td>Label location relative to the sensor: top, bottom, left, right, or inside. Default is inside.</td> <td>value: string</td> </tr> <tr> <td>justify</td> <td>Label text justification: center, left, right, or auto. Default is center. This property is only valid if props.label.location is set to top, bottom, left, or right.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Modify label text style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	text	Text for the label.	value: string	location	Label location relative to the sensor: top, bottom, left, right, or inside. Default is inside.	value: string	justify	Label text justification: center, left, right, or auto. Default is center. This property is only valid if props.label.location is set to top, bottom, left, or right.	value: string	style	Modify label text style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	object
Name	Description	Property Type															
text	Text for the label.	value: string															
location	Label location relative to the sensor: top, bottom, left, right, or inside. Default is inside.	value: string															
justify	Label text justification: center, left, right, or auto. Default is center. This property is only valid if props.label.location is set to top, bottom, left, or right.	value: string															
style	Modify label text style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															
value	<p>Value settings for the component.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Name</th> <th style="width: 60%;">Description</th> <th style="width: 25%;">Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Value to display as text.</td> <td>value: numeric</td> </tr> <tr> <td>location</td> <td>Value location relative to the vessel: top, bottom, right, left, or inside. Default is inside.</td> <td>value: string</td> </tr> <tr> <td>justify</td> <td>Value text justification: center, left, or right. Default is center. This property is only valid if props.value.location is set to top, bottom, left, or right.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Sets a style for the value. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	text	Value to display as text.	value: numeric	location	Value location relative to the vessel: top, bottom, right, left, or inside. Default is inside.	value: string	justify	Value text justification: center, left, or right. Default is center. This property is only valid if props.value.location is set to top, bottom, left, or right.	value: string	style	Sets a style for the value. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	object
Name	Description	Property Type															
text	Value to display as text.	value: numeric															
location	Value location relative to the vessel: top, bottom, right, left, or inside. Default is inside.	value: string															
justify	Value text justification: center, left, or right. Default is center. This property is only valid if props.value.location is set to top, bottom, left, or right.	value: string															
style	Sets a style for the value. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															
style	Sets a style for this cylindrical tank. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															

Component Events

Perspective Component Events

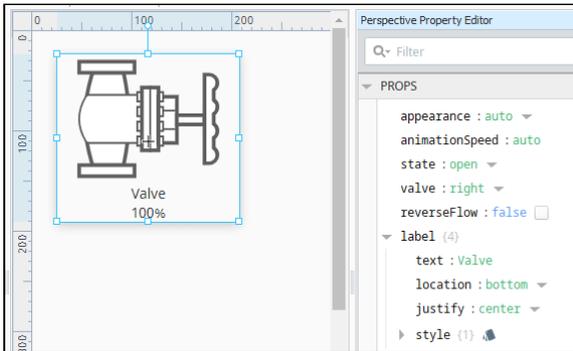
The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Perspective - Valve

The following feature is new in Ignition version **8.1.0**
[Click here](#) to check out the other new features

On this page ...

- [Properties](#)
- [Component Events](#)



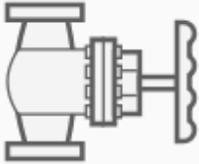
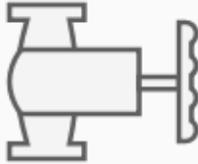
Component Palette Icon:



A component that looks like a valve. Full menu of [style options](#) is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a [style class](#).

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
appearance	<p>Options for the appearance of the component. Options are auto, p&id, mimic, and simple.</p> <p>If set to auto, the value is taken from the Perspective Session Property symbols.autoAppearance.</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">  <p>Valve 100%</p> </div> <div style="text-align: center;">  <p>Valve p&id 100%</p> </div> <div style="text-align: center;">  <p>Valve simple 100%</p> </div> </div>	value: string
animationSpeed	<p>The speed of the animation as a percent. Set to "0" to turn off animation.</p> <p>If set to auto, the setting is taken from the Perspective Session Property symbols.autoAnimationSpeed.</p>	value: numeric
state	<p>State of the valve. Built-in options are default, open, failedToOpen, partiallyClosed, closed, or failedToClose.</p> <div style="border: 1px solid orange; padding: 5px; margin: 10px 0;"> <p>The following feature is new in Ignition version 8.1.26 Click here to check out the other new features</p> </div> <p>Built-in states and new states can be configured and applied on the Project Properties > Symbols page.</p>	value: string

valve	Orientation of the valve. Options are top, bottom, left, or right. Default is right.	value: string															
reverse Flow	Whether or not to reverse the direction of the flow. Default is false.	value: boolean															
label	Label settings for the component.	value: numeric															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Text for the label.</td> <td>value: string</td> </tr> <tr> <td>location</td> <td>Label position relative to the valve: top, bottom, left, or right. Default is bottom.</td> <td>value: string</td> </tr> <tr> <td>justify</td> <td>Label text justification: center, left or right. Default is center.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Modify label text style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	text	Text for the label.	value: string	location	Label position relative to the valve: top, bottom, left, or right. Default is bottom.	value: string	justify	Label text justification: center, left or right. Default is center.	value: string	style	Modify label text style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	
Name	Description	Property Type															
text	Text for the label.	value: string															
location	Label position relative to the valve: top, bottom, left, or right. Default is bottom.	value: string															
justify	Label text justification: center, left or right. Default is center.	value: string															
style	Modify label text style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															
value	Label settings for the component.	color															
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Value to display as text.</td> <td>value: numeric</td> </tr> <tr> <td>location</td> <td>Value location relative to the valve: top, bottom, right, or left. Default is bottom.</td> <td>value: string</td> </tr> <tr> <td>justify</td> <td>Value text justification: center, left, or right. Default is center.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Sets a style for the value. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	text	Value to display as text.	value: numeric	location	Value location relative to the valve: top, bottom, right, or left. Default is bottom.	value: string	justify	Value text justification: center, left, or right. Default is center.	value: string	style	Sets a style for the value. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	
Name	Description	Property Type															
text	Value to display as text.	value: numeric															
location	Value location relative to the valve: top, bottom, right, or left. Default is bottom.	value: string															
justify	Value text justification: center, left, or right. Default is center.	value: string															
style	Sets a style for the value. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															
style	Sets a style for this cylindrical tank. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object															

Component Events

Perspective Component Events

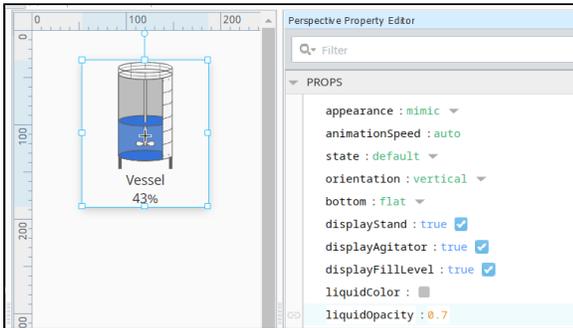
The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

Perspective - Vessel

The following feature is new in Ignition version **8.1.0**
[Click here](#) to check out the other new features

On this page ...

- [Properties](#)
- [Component Events](#)
- [Example](#)



Component Palette Icon:



An animated component that looks like a tank or vessel. Options include displaying fill level, the stand, and the agitator. Full menu of [style options](#) is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a [style class](#).

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type
appearance	<p>Options for the appearance of the component. Options are auto, p&id, mimic, and simple.</p> <p>If set to auto, the value is taken from the Perspective Session Property symbols.autoAppearance.</p>	value: string
animationSpeed	<p>The speed of the animation as a percent. Set to "0" to turn off animation.</p> <p>If set to auto, the setting is taken from the Perspective Session Property symbols.autoAnimationSpeed.</p>	value: numeric
state	<p>State of the animation. Built-in options are default, running, stopped, or faulted. Default is default.</p> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p>The following feature is new in Ignition version 8.1.26 Click here to check out the other new features</p> </div>	value: string

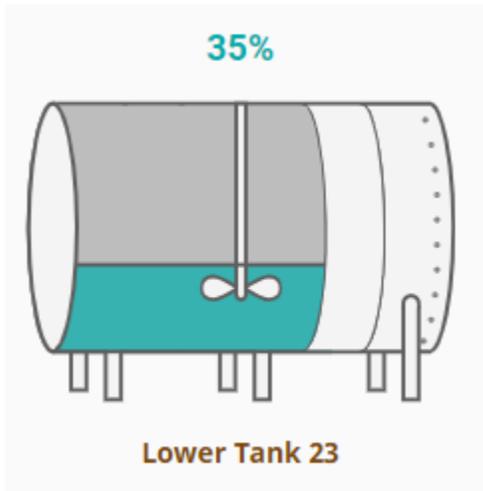
	Built-in states and new states can be configured and applied on the Project Properties > Symbols page.																						
orientation	Orientation of the vessel. Options are horizontal or vertical. Default is vertical.	value: string																					
displayStand	Whether or not to display the stand for the vessel. Default is true.	value: boolean																					
displayAgitator	Whether or not to display the agitator in the vessel. Default is true.	value: boolean																					
displayFillLevel	Whether or not to display the fill level of liquid in the vessel. Default is true.	value: boolean																					
liquidColor	Color used to render the filled part of the vessel. Can be chosen from color wheel, chosen from color palette, or entered as RGB or HSL value. See Color Selector .	color																					
liquidOpacity	The opacity of the liquid in the tank. 0 is fully transparent, 1 is fully opaque. Default is 0.7.	value: numeric																					
liquidWarningColor	The warning color of the liquid in the tank. See Color Selector .	color																					
label	Label settings for the component. <table border="1" data-bbox="256 737 1336 1045"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>text</td> <td>Text for the label.</td> <td>value: string</td> </tr> <tr> <td>location</td> <td>Label location relative to the vessel: top, bottom, left, or right. Default is bottom.</td> <td>value: string</td> </tr> <tr> <td>justify</td> <td>Label text justification: center, left or right. Default is center.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Modify label text style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	text	Text for the label.	value: string	location	Label location relative to the vessel: top, bottom, left, or right. Default is bottom.	value: string	justify	Label text justification: center, left or right. Default is center.	value: string	style	Modify label text style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	object						
Name	Description	Property Type																					
text	Text for the label.	value: string																					
location	Label location relative to the vessel: top, bottom, left, or right. Default is bottom.	value: string																					
justify	Label text justification: center, left or right. Default is center.	value: string																					
style	Modify label text style using the style properties. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																					
value	Value settings for the component. <table border="1" data-bbox="256 1108 1336 1619"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>displayValueAsPercentage</td> <td>Whether or not to display the value as a percentage.</td> <td>value: boolean</td> </tr> <tr> <td>value</td> <td>Text to display as value.</td> <td>value: string</td> </tr> <tr> <td>capacity</td> <td>Total capacity of the vessel. Default is 100.</td> <td>value: numeric</td> </tr> <tr> <td>location</td> <td>Value location relative to the vessel: top, bottom, right, or left. Default is bottom.</td> <td>value: string</td> </tr> <tr> <td>justify</td> <td>Value text justification: center, left, or right. Default is center.</td> <td>value: string</td> </tr> <tr> <td>style</td> <td>Sets a style for the value. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class.</td> <td>object</td> </tr> </tbody> </table>	Name	Description	Property Type	displayValueAsPercentage	Whether or not to display the value as a percentage.	value: boolean	value	Text to display as value.	value: string	capacity	Total capacity of the vessel. Default is 100.	value: numeric	location	Value location relative to the vessel: top, bottom, right, or left. Default is bottom.	value: string	justify	Value text justification: center, left, or right. Default is center.	value: string	style	Sets a style for the value. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object	object
Name	Description	Property Type																					
displayValueAsPercentage	Whether or not to display the value as a percentage.	value: boolean																					
value	Text to display as value.	value: string																					
capacity	Total capacity of the vessel. Default is 100.	value: numeric																					
location	Value location relative to the vessel: top, bottom, right, or left. Default is bottom.	value: string																					
justify	Value text justification: center, left, or right. Default is center.	value: string																					
style	Sets a style for the value. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																					
style	Sets a style for this cylindrical tank. Full menu of style options is available for text, background, margin and padding, border, shape and miscellaneous. You can also specify a style class .	object																					

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.

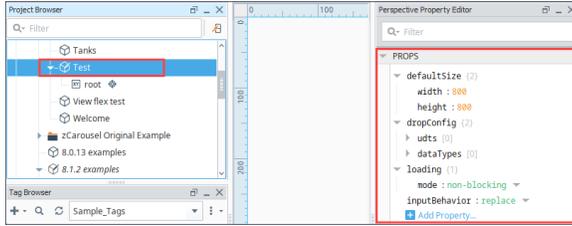
Example



Property	Value
props.orientation	horizontal
props.displayStand	true
props.liquidColor	#00ACAC
props.label.text	Lower Tank 23
props.label.location	bottom
props.label.style.fontWeight	bold
props.label.style.color	#804600
props.value.value	35
props.value.capacity	100
props.value.location	top
props.value.style.color	#00ACAC
props.value.style.fontFamily	Roboto
props.value.style.fontSize	18
props.value.style.fontWeight	bold

Perspective - View Object

This is the view itself. A view has properties that control the view size, defines how Tags are dropped into a view, and settings for loading client views. The view properties are the same for each container type. View properties are displayed in the Perspective Property Editor when you click on the view in the Project Browser.



On this page ...

- [Properties](#)
- [Component Events](#)

Properties

Most Properties have binding options. For more information on Bindings, see [Types of Bindings in Perspective](#). This section only documents the Props Category of properties. The other Categories are described on the [Perspective Component Properties](#) page.

Name	Description	Property Type																								
defaultSize	Default size of the container.	object																								
<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>width</td> <td>Width of the view. Default is 800px.</td> <td>value: numeric</td> </tr> <tr> <td>height</td> <td>Height of the view. Default is 800px.</td> <td>value: numeric</td> </tr> </tbody> </table>			Name	Description	Property Type	width	Width of the view. Default is 800px.	value: numeric	height	Height of the view. Default is 800px.	value: numeric															
Name	Description	Property Type																								
width	Width of the view. Default is 800px.	value: numeric																								
height	Height of the view. Default is 800px.	value: numeric																								
dropConfig	Provides an opportunity to automatically create an instance of this view via Tag drop. See the Drop Configuration page for more information.	object																								
<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>udts</td> <td>These settings allow for the automatic creation of this view when a UDT instance is dragged onto another view, using <i>this</i> view to represent the UDT. A parameter is passed into the view, allowing the view to reference the UDT.</td> <td>object</td> </tr> <tr> <td colspan="3"> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>type</td> <td>The name of the UDT to associate with this view.</td> <td>value: string</td> </tr> <tr> <td>param</td> <td>Name of the parameter on this view that the UDT member values will be passed to.</td> <td>value: string</td> </tr> <tr> <td>action</td> <td> <ul style="list-style-type: none"> • bind: Automatically configures a Tag binding on the parameter (specified by param) to the UDT. • path: Passes in a Tag path string on the parameter (specified by param) </td> <td>value: boolean</td> </tr> </tbody> </table> </td> </tr> <tr> <td>dataTypes</td> <td>These settings allow for the automatic creation of this view when a standard Tag is dragged onto another view. A parameter is passed into the view, allowing the view to reference the Tag.</td> <td>object</td> </tr> </tbody> </table>			Name	Description	Property Type	udts	These settings allow for the automatic creation of this view when a UDT instance is dragged onto another view, using <i>this</i> view to represent the UDT. A parameter is passed into the view, allowing the view to reference the UDT.	object	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>type</td> <td>The name of the UDT to associate with this view.</td> <td>value: string</td> </tr> <tr> <td>param</td> <td>Name of the parameter on this view that the UDT member values will be passed to.</td> <td>value: string</td> </tr> <tr> <td>action</td> <td> <ul style="list-style-type: none"> • bind: Automatically configures a Tag binding on the parameter (specified by param) to the UDT. • path: Passes in a Tag path string on the parameter (specified by param) </td> <td>value: boolean</td> </tr> </tbody> </table>			Name	Description	Property Type	type	The name of the UDT to associate with this view.	value: string	param	Name of the parameter on this view that the UDT member values will be passed to.	value: string	action	<ul style="list-style-type: none"> • bind: Automatically configures a Tag binding on the parameter (specified by param) to the UDT. • path: Passes in a Tag path string on the parameter (specified by param) 	value: boolean	dataTypes	These settings allow for the automatic creation of this view when a standard Tag is dragged onto another view. A parameter is passed into the view, allowing the view to reference the Tag.	object
Name	Description	Property Type																								
udts	These settings allow for the automatic creation of this view when a UDT instance is dragged onto another view, using <i>this</i> view to represent the UDT. A parameter is passed into the view, allowing the view to reference the UDT.	object																								
<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Property Type</th> </tr> </thead> <tbody> <tr> <td>type</td> <td>The name of the UDT to associate with this view.</td> <td>value: string</td> </tr> <tr> <td>param</td> <td>Name of the parameter on this view that the UDT member values will be passed to.</td> <td>value: string</td> </tr> <tr> <td>action</td> <td> <ul style="list-style-type: none"> • bind: Automatically configures a Tag binding on the parameter (specified by param) to the UDT. • path: Passes in a Tag path string on the parameter (specified by param) </td> <td>value: boolean</td> </tr> </tbody> </table>			Name	Description	Property Type	type	The name of the UDT to associate with this view.	value: string	param	Name of the parameter on this view that the UDT member values will be passed to.	value: string	action	<ul style="list-style-type: none"> • bind: Automatically configures a Tag binding on the parameter (specified by param) to the UDT. • path: Passes in a Tag path string on the parameter (specified by param) 	value: boolean												
Name	Description	Property Type																								
type	The name of the UDT to associate with this view.	value: string																								
param	Name of the parameter on this view that the UDT member values will be passed to.	value: string																								
action	<ul style="list-style-type: none"> • bind: Automatically configures a Tag binding on the parameter (specified by param) to the UDT. • path: Passes in a Tag path string on the parameter (specified by param) 	value: boolean																								
dataTypes	These settings allow for the automatic creation of this view when a standard Tag is dragged onto another view. A parameter is passed into the view, allowing the view to reference the Tag.	object																								

Name	Description	Property Type
type	The Tag data type to associate with this view.	value: string
param	Name of the parameter to pass into this view.	value: string
action	<ul style="list-style-type: none"> bind: Automatically configures a Tag binding on the parameter (specified by param) to the dropped Tag. path: Passes in a Tag path string on the parameter (specified by param) 	value: boolean

loading

Options for loading the view.

object

Name	Description	Property Type
mode	<p>Provides two options for loading views, particularly for views with a high number of components: blocking and non-blocking. Blocking loads faster for views with fewer components. Non-blocking loads views with a large number of components in chunks. Default is 'non-blocking'.</p> <p>Views set to non-blocking will only wait patiently for five seconds if they have not been called.</p>	value: string

inputBehavior

The following feature is new in Ignition version **8.1.4**
[Click here](#) to check out the other new features

value:
string

Controls whether the internal properties of object-typed input and bi-directional parameters are merged with existing default values or whether the object-typed parameter should be strictly replaced with the supplied values.

If "replace" is selected, and an object-typed parameter is supplied which does not contain a key for a default value, then that default value will not be used. Default is replace.

Defined "default" parameter:

```
myObject: {
  myParamOne: "One",
  myParamTwo: "Two"
}
```

Supplied parameter from Embedding Component:

```
myObject: {
  myParamOne: "1",
  myParamThree: "3"
}
```

If "merge" is selected:

```
myObject: {
  myParamOne: "1",
  myParamTwo: "Two",
  myParamThree: "3"
}
```

If "replace" is selected:

```
myObject: {  
  myParamOne: "1",  
  myParamThree: "3"  
}
```

Note that there is no value for `myParamTwo`, so any references to this property will fault.

Component Events

Perspective Component Events

The [Perspective Event Types Reference](#) page describes all the possible component event types for Perspective components. Not all component events support each Perspective component. The [Component Events and Actions](#) page shows how to configure events and actions on a Perspective component. Component scripting is handled separately and can be accessed from the Component menubar or by right clicking on the component.