

TOC

| | |
|--|-----------|
| About This Guide | 19 |
| Chart Canvas Charts | 20 |
| Using the Chart Wizard | 21 |
| Chart Canvas Charts Basics | 28 |
| Chart Geography | 28 |
| Manual Chart Quick Start | 30 |
| The Chart Canvas Element | 32 |
| Getting Data | 37 |
| The Chart Canvas Child Elements | 39 |
| Default Colors | 46 |
| Refreshing Chart Data | 47 |
| X-Axis and Y-Axis Elements | 48 |
| About the Axis Elements | 48 |
| Axis Element Attributes | 50 |

| | |
|---|-----------|
| Plotting Negative Values | 53 |
| Styling the Axis Caption | 54 |
| Styling the Axis Labels | 55 |
| Auto Rotation | 57 |
| Adding a Marker Line | 58 |
| Adding a Marker Band | 59 |
| Configuring Major Ticks and Grid Lines | 60 |
| Configuring Minor Ticks and Grid Lines | 63 |
| Showing Totals for Stacked Series | 65 |
| Adding Multiple Axes | 67 |
| Legend | 70 |
| About the Legend | 70 |
| Legend Filtering | 73 |
| Legend Item Order | 74 |
| Legend Attributes | 75 |

| | |
|---|------------|
| Styling the Legend Caption | 79 |
| Styling the Legend Items | 80 |
| Styling the Legend Navigation Arrows | 82 |
| The Resizer | 83 |
| Attributes | 84 |
| Gradient Style | 85 |
| Linear Gradient | 86 |
| Linear Gradient Attributes | 90 |
| Radial Gradient | 90 |
| Radial Gradient Attributes | 95 |
| Input Selection | 97 |
| About Input Selection | 97 |
| Using Input Selection Point | 100 |
| Attributes | 101 |
| Input Selection Point Event | 103 |

| | |
|---|------------|
| Input Selection Style | 103 |
| Using Input Selection Range | 106 |
| Input Selection Range | 108 |
| Input Selection Range Event | 110 |
| Input Selection Style | 111 |
| Caption Style and SubCaption Style | 113 |
| Attributes | 114 |
| No Data Caption Style | 117 |
| Attributes | 117 |
| Chart Annotation | 119 |
| AnnotationLabel.Mock | 120 |
| Chart Color Axis | 124 |
| Chart Custom Symbol | 128 |
| Chart Description Style | 131 |
| Attributes | 131 |

| | |
|--|------------|
| Trend Line | 134 |
| About the Trend Line | 134 |
| Trend Line Attributes | 136 |
| Including Legend Items | 137 |
| Zoom Control | 138 |
| About the Zoom Control | 138 |
| Attributes | 139 |
| Group Drillthrough | 140 |
| About Group Drillthrough | 140 |
| Group Drillthrough Attributes | 143 |
| Customizing Detail Report Columns | 145 |
| Series.Area | 147 |
| About Series.Area | 147 |
| Series.Area - Using Multiple Series | 151 |
| Forecasting | 153 |

| | |
|--|------------|
| Series.Area - Attributes | 154 |
| Series.Area - Using the Data Labels Elements | 158 |
| Series.Area - Using the Marker Points Element | 159 |
| Series.Area - Using the Quicktips Element | 160 |
| Series.Area - Using the Trend Line Element | 162 |
| Series.Area - Using Action Elements | 163 |
| Series.Area - Using the Series Annotation Element | 164 |
| Series.Area - Using Input Selection | 166 |
| Series.Area - Using Refresh Series Timer | 167 |
| Refresh Mode | 167 |
| Series.Area Range | 169 |
| About Series.Area Range | 169 |
| Series.Area Range - Using Multiple Series | 173 |
| Series.Area Range - Attributes | 176 |
| Series.Area Range - Using the Data Labels Element | 180 |

| | |
|--|------------|
| Series.Area Range - Using the Quicktips Element | 181 |
| Series.Area Range - Using Action Elements | 183 |
| Series.Area Range - Using the Series Annotation Element | 184 |
| Series.Area Range - Using Input Selection | 187 |
| Series.Area Range - Using the Refresh Series Timer | 188 |
| Refresh Mode | 188 |
| Series.Area Spline | 190 |
| About Series.Area Spline | 190 |
| Series.Area Spline - Using the Refresh Series Timer | 194 |
| Refresh Mode | 194 |
| Series.Area Spline - Using Multiple Series | 196 |
| Series.Area Spline - Attributes | 199 |
| Series.Area Spline - Using the Data Label Element | 203 |
| Series.Area Spline - Using the Marker Points Element | 204 |
| Series.Area Spline - Using the Quicktips Element | 205 |

| | |
|---|------------|
| Series.Area Spline - Using the Trend Line Element | 207 |
| Series.Area Spline - Using Action Elements | 208 |
| Series.Area Spline - Using the Series Annotation Element | 209 |
| Series.Area Spline - Using Input Selection | 211 |
| Series.Area Spline Range | 212 |
| About Series.Area Spline Range | 212 |
| Series.Area Spline Range - Using Multiple Series | 216 |
| Series.Area Spline Range - Attributes | 219 |
| Series.Area Spline Range - Using the Data Labels Element | 223 |
| Series.Area Spline Range - Using the Quicktips Element | 224 |
| Series.Area Spline Range - Using Action Elements | 226 |
| Series.Area Spline Range - Using the Series Annotation Element | 227 |
| Series.Area Spline Range - Using Input Selection | 230 |
| Series.Area Spline Range - Using the Refresh Series Timer | 231 |
| Refresh Mode | 231 |

| | |
|---|------------|
| Series.Bar | 233 |
| About Series.Bar | 233 |
| Series.Bar - Using Multiple Series | 238 |
| Forecasting | 240 |
| Series.Bar - Attributes | 242 |
| Series.Bar - Using the Data Labels Element | 246 |
| Series.Bar - Using the Quicktips Element | 248 |
| Series.Bar - Using the Chart Drill To Element | 251 |
| Series.Bar - Using the Trend Line Element | 256 |
| Series.Bar - Using Action Elements | 257 |
| Series.Bar - Using the Series Annotation Element | 258 |
| Series.Bar - Using Input Selection | 260 |
| Series.Bar - Using the Refresh Series Timer | 261 |
| Refresh Mode | 261 |
| Series.Bar Range | 263 |

| | |
|---|------------|
| About Series.Bar Range | 263 |
| Series.Bar Range - Using Multiple Series | 267 |
| Series.Bar Range - Attributes | 270 |
| Series.Bar Range - Using the Data Labels Element | 274 |
| Series.Bar Range - Using the Quicktips Element | 275 |
| Series.Bar Range - Using Action Elements | 277 |
| Series.Bar Range - Using the Series Annotation Element | 278 |
| Series.Bar Range - Using Input Selection | 281 |
| Series.Bar Range - Using the Refresh Series Timer | 282 |
| Refresh Mode | 282 |
| Series.Funnel | 284 |
| About Series.Funnel | 284 |
| Series.Funnel - Using Multiple Series | 288 |
| Series.Funnel - Attributes | 290 |
| Series.Funnel - Adding Data Labels | 293 |

| | |
|---|------------|
| Series.Funnel - Using the Quicktips Element | 296 |
| Series.Funnel - Using the Action Elements | 298 |
| Series.Funnel - Using the Series Annotation Element | 299 |
| Series.Funnel - Using Input Selection | 301 |
| Series.Funnel - Using the Refresh Series Timer | 302 |
| Refresh Mode | 302 |
| Series.Heatmap | 304 |
| About Series.Heatmap | 304 |
| Series.Heatmap - Attributes | 308 |
| Series.Heatmap - Grouping Heatmap Cells | 310 |
| Series.Heatmap - Using the Heatmap Label Style Element | 312 |
| Series.Heatmap - Using the Quicktips Element | 313 |
| Series.Heatmap - Using the Chart Drill to Element | 315 |
| Series.Heatmap - Using Action Elements | 319 |
| Series.Heatmap - Using the Series Annotation Element | 320 |

| | |
|--|------------|
| Series.Heatmap - Using the Refresh Series Timer | 323 |
| Refresh Mode | 323 |
| Series.Line | 325 |
| About Series.Line | 325 |
| Series.Line - Using Multiple Series | 329 |
| Forecasting | 332 |
| Series.Line - Attributes | 334 |
| Series.Line - Using the Data Labels Element | 337 |
| Series.Line - Using the Marker Points Element | 338 |
| Series.Line - Using the Quicktips Element | 339 |
| Series.Line - Using the Trend Line Element | 341 |
| Series.Line - Using Action Elements | 342 |
| Series.Line - Using the Series Annotation Element | 343 |
| Series.Line - Using Input Selection | 345 |
| Series.Line - Using the Refresh Series Timer | 346 |

| | |
|---|------------|
| Refresh Mode | 346 |
| Series.Pie | 348 |
| About Series.Pie | 348 |
| Series.Pie - Shaping the Pie | 352 |
| Series.Pie - Using Multiple Series | 355 |
| Series.Pie - Attributes | 357 |
| Series.Pie - Adding Data Labels | 362 |
| Series.Pie - Using the Quicktips Element | 366 |
| Series.Pie - Using the Chart Drill to Element | 368 |
| Series.Pie - Using Action Elements | 373 |
| Series.Pie - Using the Series Annotation Element | 374 |
| Series.Pie - Using Input Selection | 377 |
| Series.Pie - Using the Refresh Series Timer | 378 |
| Refresh Mode | 378 |
| Series.Pyramid | 380 |

| | |
|---|------------|
| About Series.Pyramid | 380 |
| Series.Pyramid - Using Multiple Series | 383 |
| Series.Pyramid - Attributes | 385 |
| Series.Pyramid - Adding Data Labels | 388 |
| Series.Pyramid - Using the Quicktips Element | 391 |
| Series.Pyramid - Using the Action Elements | 393 |
| Series.Pyramid - Using the Series Annotation Element | 394 |
| Series.Pyramid - Using the Input Selection | 396 |
| Series.Pyramid - Using the Refresh Series Timer | 397 |
| Refresh Mode | 397 |
| Series.Scatter | 399 |
| About Series.Scatter | 399 |
| Series.Scatter - Using Multiple Series | 403 |
| Adding a Secondary Axis | 405 |
| Series.Scatter - Attributes | 407 |

| | |
|---|------------|
| Series.Scatter - Using the Data Labels Element | 410 |
| Series.Scatter - Using the Marker Points Element | 411 |
| Series.Scatter - Using the Quicktips Element | 413 |
| Series.Scatter - Using the Trend Line Element | 415 |
| Series.Scatter - Using Action Elements | 416 |
| Series.Scatter - Using the Series Annotation Element | 417 |
| Series.Scatter - Using Input Selection | 419 |
| Series.Scatter - Using the Refresh Series Timer | 420 |
| Refresh Mode | 420 |
| Series.Spline | 422 |
| About Series.Spline | 422 |
| Series.Spline - Using Multiple Series | 425 |
| Series.Spline - Attributes | 428 |
| Series.Spline - Using the Data Labels Element | 431 |
| Series.Spline - Using the Marker Points Element | 432 |

| | |
|---|------------|
| Series.Spline - Using the Quicktips Element | 433 |
| Series.Spline - Using the Trend Line Element | 435 |
| Series.Spline - Using Action Elements | 436 |
| Series.Spline - Using the Series Annotation Element | 437 |
| Series.Spline - Using Input Selection | 439 |
| Series.Spline - Using the Refresh Series Timer | 440 |
| Refresh Mode | 440 |
| Series.Waterfall | 442 |
| About Series.Waterfall | 442 |
| Series.Waterfall - Using Multiple Series | 445 |
| Series.Waterfall - Attributes | 447 |
| Series.Waterfall - Using the Data Labels Element | 451 |
| Series.Waterfall - Using the Quicktips Element | 453 |
| Series.Waterfall - Using Action Elements | 455 |
| Series.Waterfall - Using the Series Annotation Element | 456 |

| | |
|--|------------|
| Series.Waterfall - Using Input Selection | 458 |
| Series.Waterfall - Using the Refresh Series Timer | 459 |
| Refresh Mode | 459 |
| Series.Whiskers | 461 |
| About Series.Whiskers | 461 |
| Series.Whiskers - Attributes | 464 |
| Series.Whiskers - Using the Quicktips Element | 467 |
| Series.Whiskers - Using Action Elements | 471 |
| Series.Whiskers - Using the Series Annotation Element | 472 |
| Series.Whiskers - Using the Refresh Series Timer | 475 |
| Refresh Mode | 475 |
| Series.Bubble | 477 |
| About Series.Bubble | 477 |
| Series.Bubble - Using Multiple Series | 480 |
| Series.Bubble - Attributes | 483 |

| | |
|--|------------|
| Series.Bubble - Using the Data Labels Element | 486 |
| Series.Bubble - Using the Marker Points Element | 487 |
| Series.Bubble - Using the Quicktips Element | 489 |
| Series.Bubble - Using Action Elements | 491 |
| Series.Bubble - Using the Series Annotation Element | 492 |
| Series.Bubble - Using Input Selection | 495 |
| Series.Bubble - Using the Refresh Series Timer | 496 |
| Refresh Mode | 496 |
| Glossary | 498 |

About This Guide

This is an archived copy of the v23 documentation provided for Logi Info v23.3 and its service packs.

Notice: Archived Documentation

This documentation is provided as a courtesy reference for a version of our software that is no longer under active development or support. The information contained herein is offered without warranties of any kind, either expressed or implied, including but not limited to warranties of accuracy, completeness, or fitness for a particular purpose.

While this archived material may assist with understanding historical functionality, please be aware that the software described is no longer maintained at this version level and may contain outdated or inaccurate information. Images may not reflect currently supported modules, support sites, or third party products. This software may not be compatible with current versions of previously compatible third party products.

To access and upgrade to current software solutions and receive ongoing support, please contact our customer support team. They can assist you in migrating to the latest appropriate software version that meets your needs. Our support team is available to help ensure a smooth transition to actively maintained alternatives that provide the functionality and reliability you require.

Chart Canvas Charts

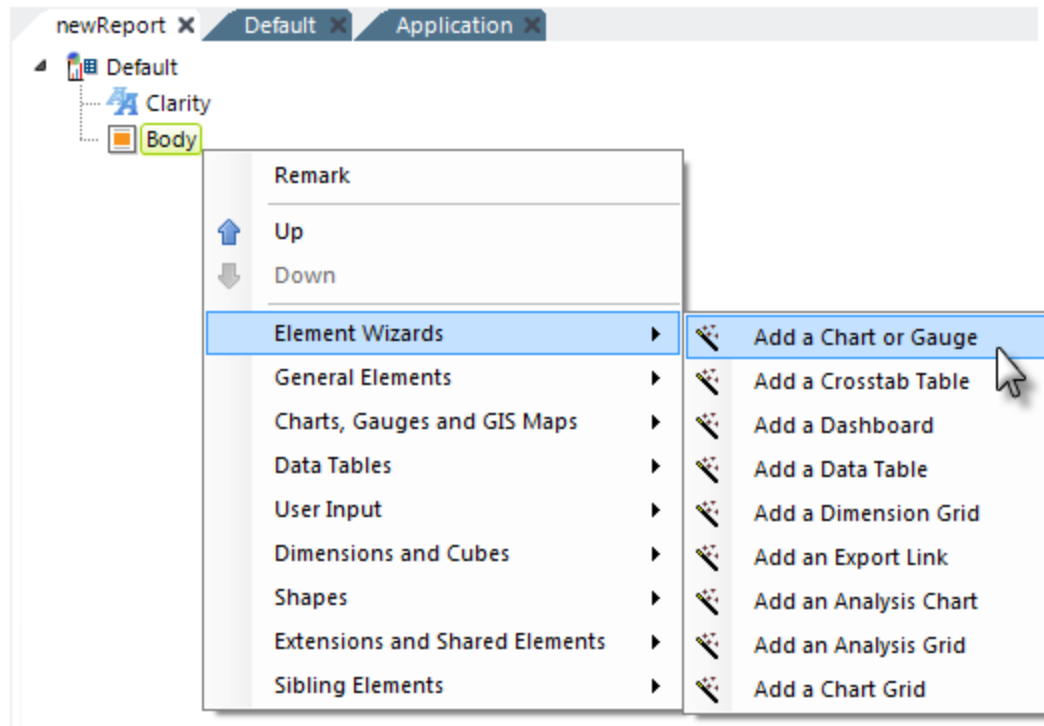
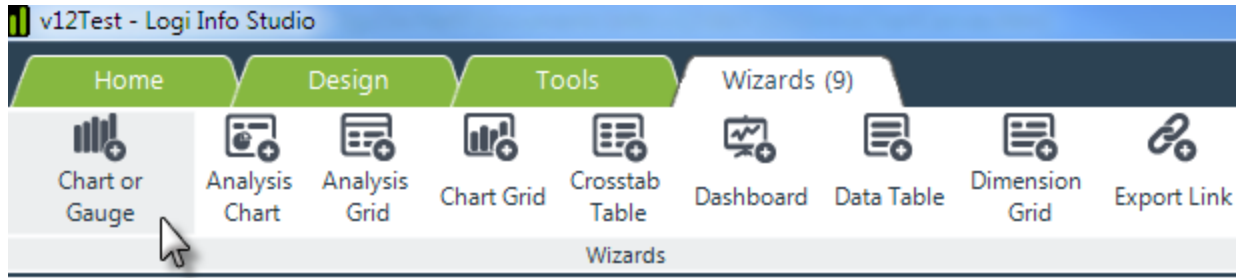
Logi Info includes a fast, JavaScript-based family of charting elements that offers a number of improvements over the legacy Static and Animated chart elements.

The following topics discuss the Chart Canvas charts elements:

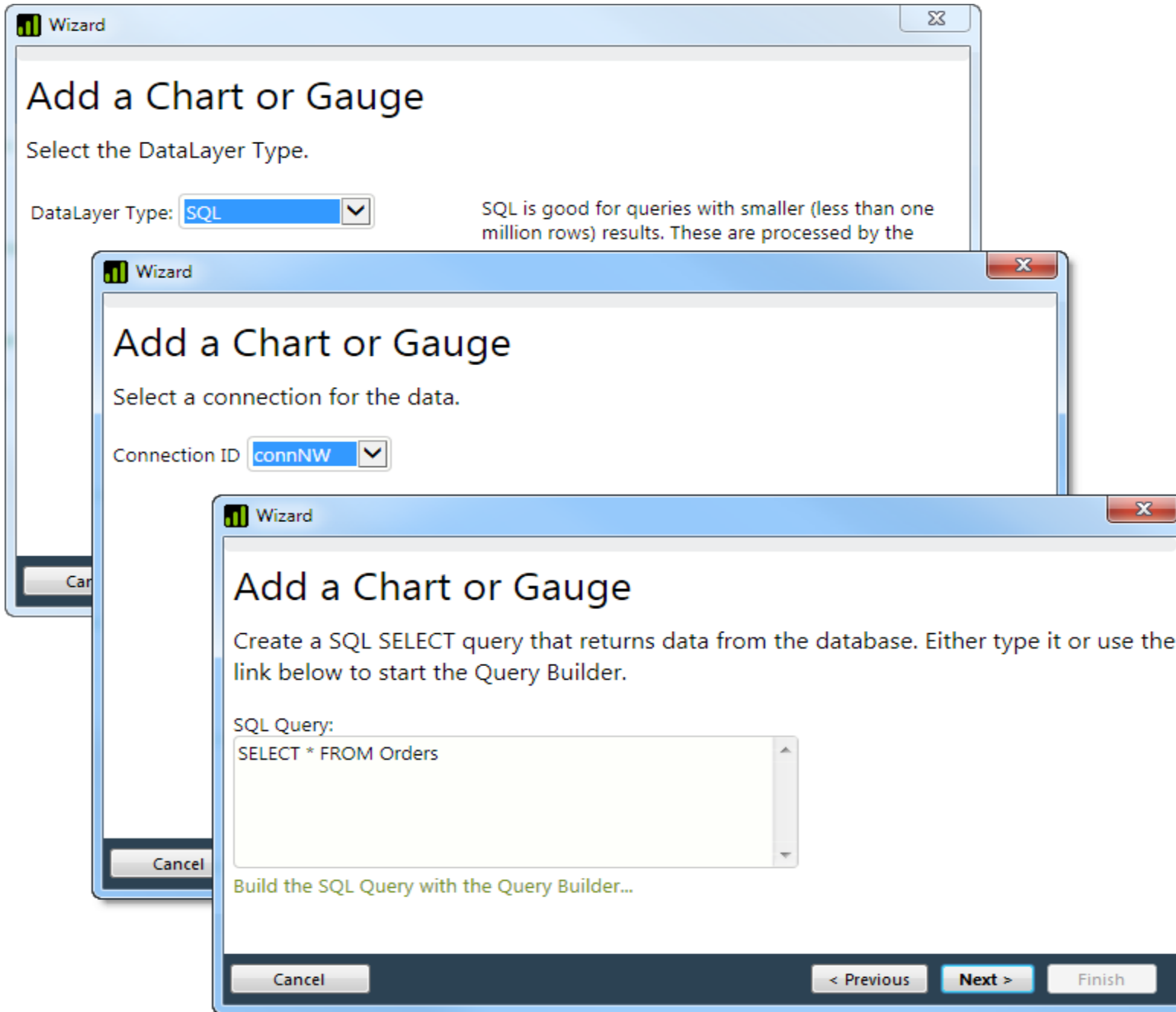
- [Using the Chart Wizard](#)
- [Chart Canvas Chart Basics](#)
- [Manual Chart Quick Start](#)
- [The Chart Canvas Elements](#)
- [The Chart Canvas Children Elements](#)
- [Default Colors](#)
- [Refreshing Chart Data](#)

Using the Chart Wizard

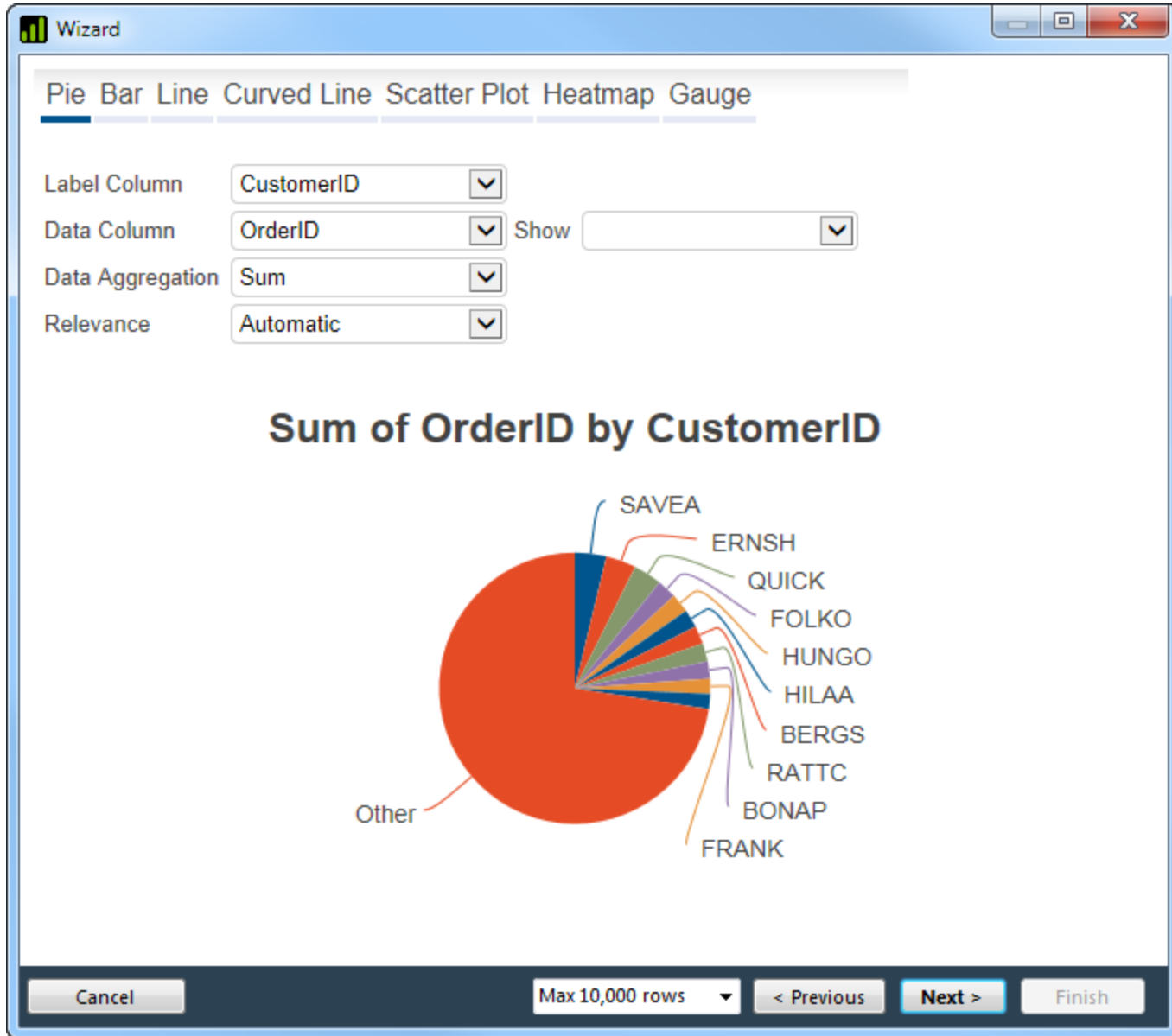
The fastest and easiest way to create a Chart Canvas chart is to use the **Chart Wizard** in Logi Studio. Here's how:



1. As shown above, in a definition in Logi Studio, select an element like Body or a Division, and either click the **Chart or Gauge** item in the main menu's Wizards tab, or right-click the element and select *Element Wizards*, and then select *Add a Chart or Gauge* from the secondary menu.



2. A series of dialog boxes, shown above, will be displayed. These all relate to the retrieval of the data. Make appropriate selections for your application and click **Next** to move to the next dialog box. As you do, you'll see the wizard adding elements to your definition.

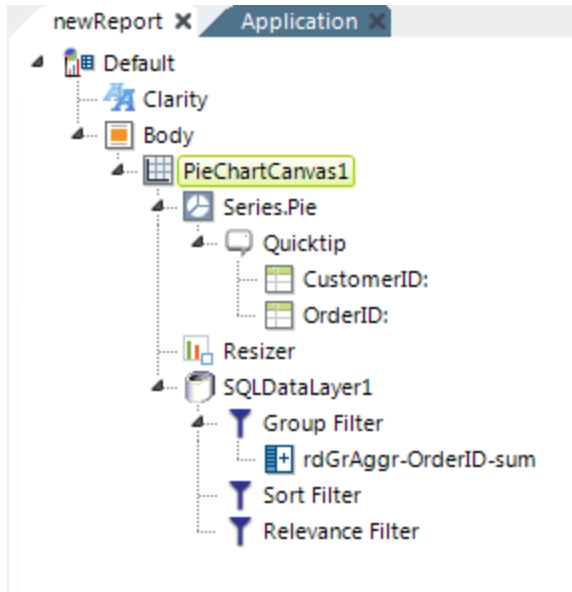


- Based on the data, the wizard will make an educated guess about appropriate chart types and will display a preview chart in the next dialog box. You can select other chart types from the options across the top, and select other data columns and analysis options. When the chart is just the way you want it, click **Next**.

The *Max Rows* option at the bottom of the dialog box controls the maximum number of rows that will be retained in the chart's datalayer.



- Finally, you'll be asked if you want to add "resizer handles" to the chart. These allow the user to resize the chart at runtime by dragging the handles. Click **Next** to exit the wizard.




When you exit the wizard, it'll finish inserting all the configured elements for the chart into your report definition, as shown above. Preview your application and you should see your new chart.

Chart Canvas Charts Basics

The Chart Canvas chart elements are based on a library written entirely in **HTML5** and **JavaScript**, and work very well in all modern desktop and mobile device browsers. No client-side technology, such as Flash or Java, is required. These elements provide a wide range of features *and* the flexibility to customize them to your requirements. Many desirable features, including animations and quicktips, are enabled by default, making it very easy to create great-looking charts quickly. In addition, these charts are exportable in reports.

By default, Chart Canvas charts are rendered by the browser using SVG technology. For browsers that do not support SVG, primarily IE 7 & 8, charts are rendered with VML.

The PDF export engine was changed to Gecko-based technology and Chart Canvas charts are now exported as SVG objects rather than as images. This results in Chart Canvas charts exported to PDF having extremely high resolution - they can be zoomed or printed with high-quality at any resolution.

 When exporting Chart Canvas charts to PDF, the **Chart Canvas** element must have Height and Width attribute values set. The legacy Static and Animated charts available in earlier versions of Logi Info have been deprecated. Developers should use Chart Canvas Charts for all new projects.


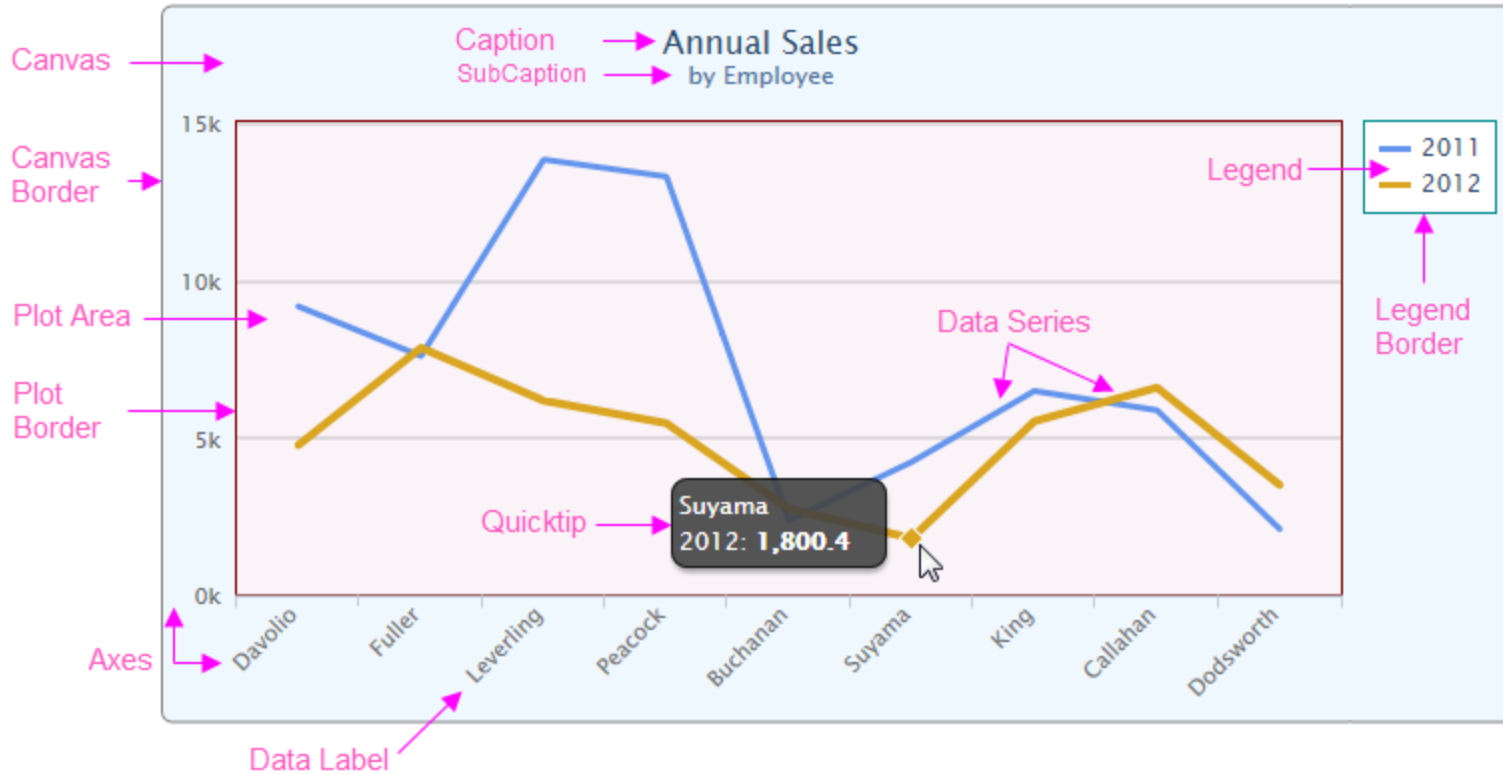
 Chart Canvas charts are used by default for Super Elements, such as the Analysis Grid, in all *new applications*. Logi applications that are *upgraded* to v12 will use the classic static charts for their Super Elements. To force upgraded apps to use Chart Canvas charts, add the constant `rdFavorChartCanvas = True` to your `_Settings` definition.

Chart Geography

The following image explains many of the terms used to describe a typical Chart Canvas chart:

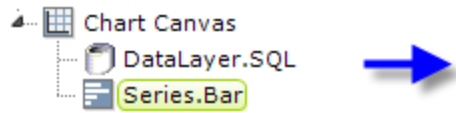


The background color, line properties, and transparency of the areas and lines shown can all be controlled by setting attributes. Additional major and minor tick marks, and grid lines can be added. The spacing, style, and format of all the items shown can also be configured.

💡 Depending on the number of bars in your chart, a scroll bar can be enabled for optimal display. The scroll bar is available for both vertical and horizontal orientations and provides a fixed width, while displaying the chart's axis. To enable this feature, set the MinWidth/MinHeight attributes greater than the current chart Width/Height.

Manual Chart Quick Start

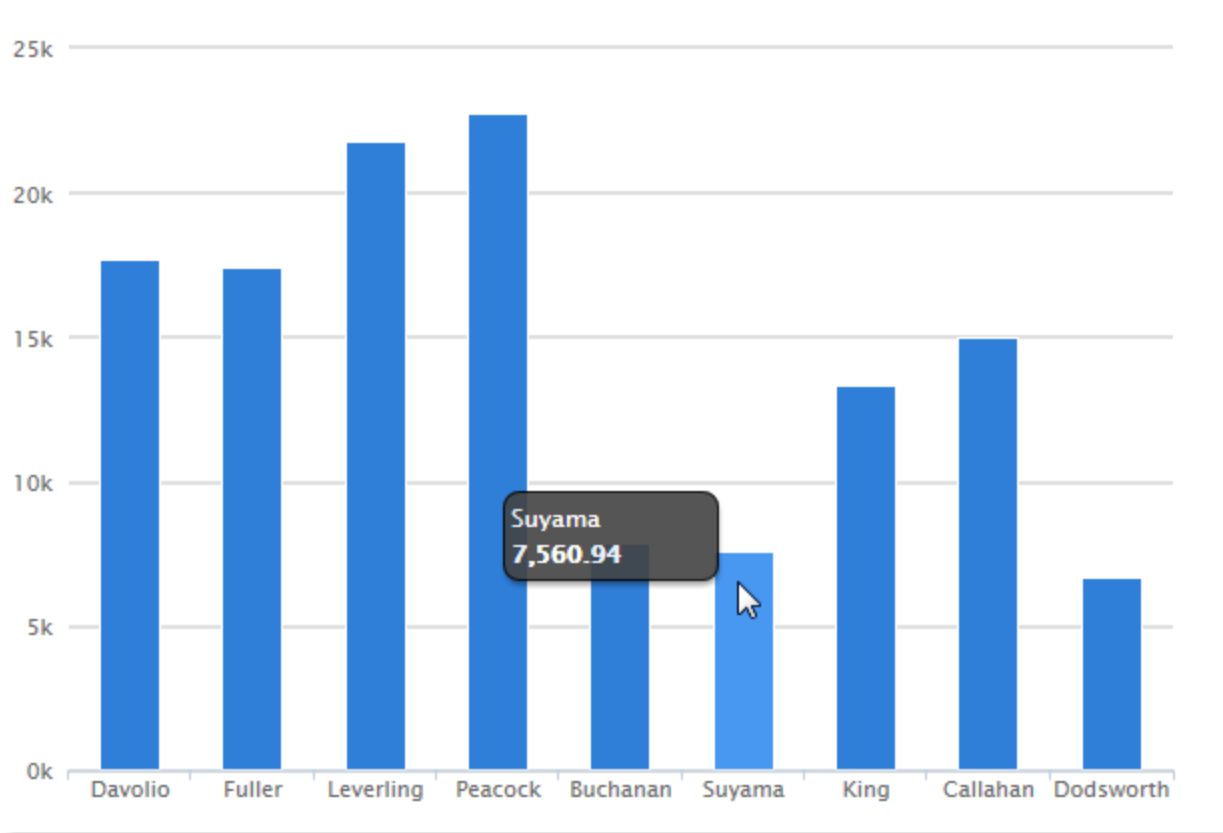
Chart Canvas charts are incredibly easy to use! All you need to do to create a chart manually is to add two chart elements, set two attributes, and include a datalayer.



1. Add a **Chart Canvas** element
2. Add a child **datalayer** and set appropriate attributes
3. Add a child **Series** and set X- and Y-axis data column names

| Element - Series.Bar | |
|-------------------------------|------------|
| *Required Attributes | |
| Y-axis Data Column | sumFreight |
| Optional Attributes | |
| Bar Border Color | |
| Bar Border Color Transparency | |
| Bar Border Radius | |
| ... | |
| Transparency | |
| X-axis Data Column | LastName |
| X-axis Data Column Type | |

No other chart element attributes need be set - everything else will be done automatically for you.



The resulting chart will be automatically-sized and styled, and will include hover-highlighting and quicktips, as shown above. Now let's dig into the elements we used, and others in the Chart Canvas family.

The Chart Canvas Element

The **Chart Canvas** element is the parent, or "container", for all of the other elements used in these charts. As such, its attributes specify the size and appearance of the overall chart, the canvas, and the plot area.



Here are its attributes:

| Attribute | Description |
|-------------------------------|--|
| Alternate Text | Specifies text to be displayed when the browser's options are set to <i>not</i> display images. The text is also used by browsers that convert text to speech or Braille output. |
| Animation Duration | Sets the duration of chart animations, in milliseconds. To disable animation, enter a <i>0</i> value. |
| Auto Quicktip | Enables the display of Quicktips, with automatically determined information, when the mouse is hovered over data points. To create customized Quicktips, add a Quicktip element under the Series element. |
| Background Color | Sets the canvas background color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. |
| Background Color Transparency | Specifies the transparency of the canvas background color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| Border Color | Sets the canvas border line color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. |

| Attribute | Description |
|--|--|
| Border Color Transparency | Specifies the transparency of the canvas border line color. The lowest value of <i>0</i> specifies that the border line is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent line. Use medium-level transparency to allow different chart layers to show through each other. |
| Border Radius | Sets the amount of rounding for canvas border line corners, in pixels. |
| Border Thickness | Sets the canvas border line thickness, in pixels. The default value is <i>0</i> , for no border. |
| Caption | Specifies the text for the chart main title. Line feeds may be used in this attribute. In Logi Studio, open the Attribute Zoom window for this attribute and use the Enter key while typing text to introduce line feeds where desired. The text will appear on different lines when the chart is drawn. |
| Chart Description | Specifies the text for the chart's description. The default value for this attribute is <i>empty</i> . Enter a description to enable this feature. |
| Class | Specifies the Cascading Style Sheet class used by the element. When set, this class will also be used by all child elements that don't have their own class. The class should be defined in the report's style sheet file or theme. |
| v23.1 Disable Legend Filtering | Specifies whether the legend can be filtered. The default value for this attribute is <i>empty</i> , or <i>False</i> , the legend can be filtered. To disable legend filtering, set this attribute to <i>True</i> . |
| Height | Specifies the height of the canvas, in pixels. If left blank, a height will be determined automatically. Must be |

| Attribute | Description |
|------------------------------------|--|
| | set to a value in order to export chart to PDF. |
| MinHeight | Specifies the minimum height of the canvas. Set this value to greater than the chart's current height to enable scroll bars. |
| MinWidth | Specifies the minimum width of the canvas. Set this value to greater than the chart's current width to enable scroll bars. |
| No Data Caption | Specifies text to be shown in the plot area when no data is received in the chart's datalayers. |
| No Debugger Link | Specifies whether or not individual chart debug icons will appear when debugging is turned on. You may want to suppress them to reduce "visual clutter" and ensure a realistic layout when debugging a page. The default value is <i>False</i> . |
| Orientation | Allows swapping of the X- and Y-axes. The default value is <i>Vertical</i> , meaning data points are plotted on the Y-axis and labels are listed across the X-axis. |
| Plot Background Color | Sets the plot area background color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. |
| Plot Background Color Transparency | Specifies the transparency of the plot background area color. The lowest value of <i>0</i> specifies that the border line is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent line. Use medium-level transparency to allow different chart layers to show through each other. |

| Attribute | Description |
|--------------------------------|---|
| Plot Background Image | Specifies the complete URL of an image to be displayed in the chart plot area (a chart "background" image). Use CSS, rather than this attribute, to specify a background image for the entire canvas. |
| Plot Border Color | Sets the plot area border line color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. |
| Plot Border Color Transparency | Specifies the transparency of the plot area border line color. The lowest value of <i>0</i> specifies that the border line is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent line. Use medium-level transparency to allow different chart layers to show through each other. |
| Plot Border Thickness | Sets the plot area border line thickness, in pixels. The default value is <i>0</i> , for no border. |
| Polar | Set to <i>True</i> to transform Cartesian charts (e.g. Area, Bar, Line, and Spline) into the Polar/Radial coordinate system, producing a "Polar Chart". The default value is <i>False</i> . |
| Show DrillTo Breadcrumb | Specifies whether the breadcrumb trail displays when using the Drill To feature. The default value is <i>False</i> , no breadcrumb trail displays. |
| Spacing Bottom | Sets the space between the bottom edge of the chart content (including the plot area and labels) and the bottom border of the chart, in pixels. The default value is <i>15</i> pixels. |
| Spacing Left | Sets the space between the left edge of the chart content (including the plot area and labels) and the left border of the chart, in pixels. The default value is <i>10</i> pixels. |

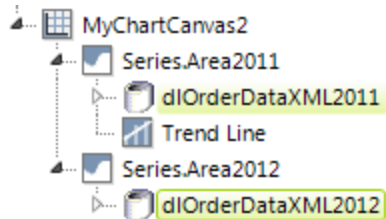
| Attribute | Description |
|---|--|
| Spacing Right | Sets the space between the right edge of the chart content (including the plot area and any labels) and the right border of the chart, in pixels. The default value is <i>10</i> pixels. |
| Spacing Top | Sets the space between the top edge of the chart content (including the plot area and any captions) and the top border of the chart, in pixels. The default value is <i>10</i> pixels. |
| SubCaption | Specifies a subcaption for the chart, which normally appears below the chart caption. Line feeds may be used in this attribute. In Logi Studio, open the Attribute Zoom window for this attribute and use the Enter key while typing text to introduce line feeds where desired. The text will appear on different lines when the chart is drawn. |
| <div style="background-color: #e0ffe0; border-radius: 5px; padding: 2px; display: inline-block; margin-bottom: 5px;">v23.1</div> Tooltip Outside | Controls whether the tooltip renders outside the chart's SVG element box. The default value is <i>empty</i> . To enable this feature, set the attribute to <i>True</i> .  Gauge.Bar and Gauge.Angular do not possess this capability. |
| Tooltip Split | Specifies whether the tooltip displays the value per segment, or per series. The default value is <i>False</i> . To enable this feature, set the attribute to <i>True</i> . The following charts do not possess this capability: Gauge, Pie, Funnel, Pyramid, and Heatmap. Furthermore, when the Chart Orientation attribute is set to <i>Swap Axes</i> , and the Polar attribute is set to <i>True</i> , this feature will not work.  This option takes precedence over <code>tooltip.shared</code> . |
| Width | Specifies the width of the canvas. By default, the unit of measure is pixels, but you may type in a value and the percent sign to indicate a percentage of its container. If left blank, a width will be determined auto- |

| Attribute | Description |
|-----------|--|
| | matically. Must be set to a value in order to export chart to PDF. |

It's recommended that you set a unique element ID for charts if there will be more than one chart in a definition.

Getting Data

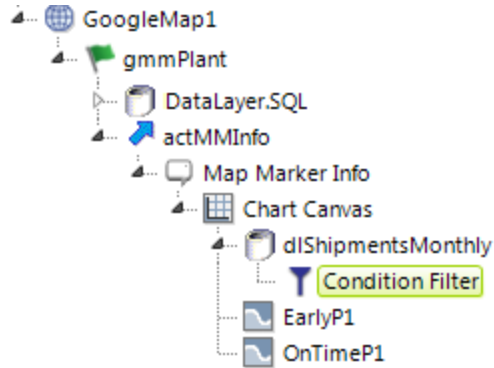
The example shown earlier used a datalayer beneath the Chart Canvas element, which makes the data retrieved available to *all* of its other child elements.



However, as shown above, instead of using a common datalayer, you can use *individual* datalayers beneath the child elements, where appropriate, to give them their own data.




Do not mix these two arrangements. Either use a single datalayer as a child of the Chart Canvas element OR a datalayer as a child of each Series element. *Do not* use a datalayer as a child of the Chart Canvas element AND a datalayer as a child of one or more Series elements.




| Element - ConditionFilter | |
|-----------------------------|-----------------------------------|
| *Required Attributes | |
| Condition | "@Chart.Locn~" = "@Request.Locn~" |
| Optional Attributes | |
| Error Result | |
| ID | |
| Include Condition | |

Use @Chart token when chart is the child of an element with its own datalayer.


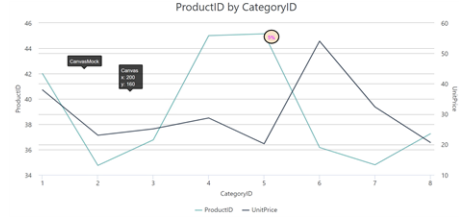
As shown in the example above, when using Chart Canvas charts as the child of an element that has its own datalayer, such as a DataTable or Google Map, you must use @Chart tokens, not @Data tokens, in any elements used to condition the chart's datalayer.

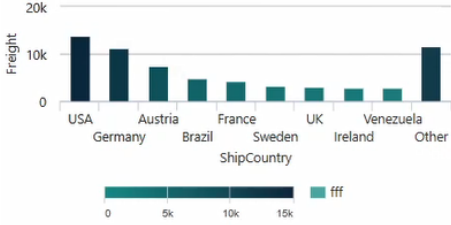



 Due to order-of-operations differences, datalayers that have the **Handle Quotes Inside Tokens** attribute set to *True* will not process quotes as expected when used underneath Chart Canvas elements. In this scenario, we recommend that you use the datalayer under Local Data and link it to a datalayer for the chart.

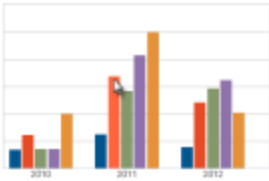
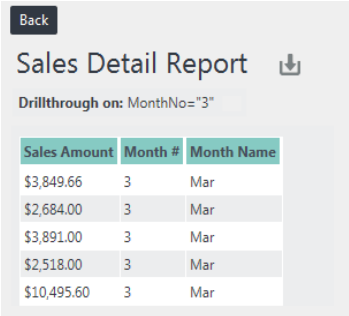
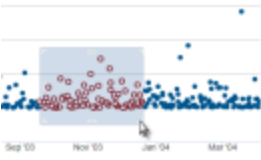
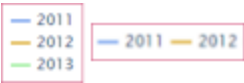
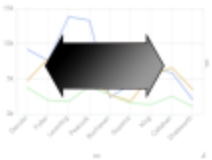
 When **DataLayer.MDX** is used as a child of Chart Canvas, only *one* Series element may be used. The datalayer cannot be used as the child of a Series element.


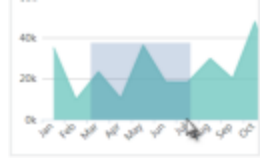



The Chart Canvas Child Elements


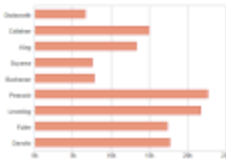



The Chart Canvas element also uses the following child elements to generate, control, and format the canvas and charts. Select the element link for more information.

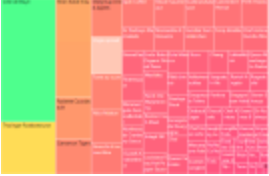




| Element | Description |
|---|--|
| <p>"X-Axis and Y-Axis Elements" on page 48</p> |  <p>Specifies properties of axis lines, grid lines, tick marks, and label fonts.</p> |
| <p>"Caption Style and SubCaption Style" on page 113</p> | <p>Investment Performance for 2013</p> <p>Specifies Caption and SubCaption font-related and positioning attributes.</p> |
| <p>v23.1 "Chart Annotation" on page 119</p> |  <p>ProductID by CategoryID</p> <p>ProductID</p> <p>UnitPrice</p> <p>CategoryID</p> <p>ProductID</p> <p>UnitPrice</p> <p>Competitor</p> <p>Growth Rate</p> <p>Specifies chart annotations such as labels and shapes placed at various points of interest.</p> |



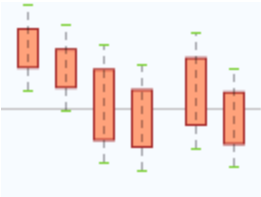
| Element | Description |
|---|--|
| <p>v23.1 "Chart Color Axis" on page 124</p> | <p>Adds a color axis for series charts.</p>  <p>The chart displays Freight values for various countries. The y-axis ranges from 0 to 20k. The x-axis lists ShipCountry: USA, Germany, Austria, Brazil, France, Sweden, UK, Ireland, Venezuela, and Other. A color axis legend at the bottom shows a gradient from light green to dark blue, with a color code 'fff'.</p> |
| <p>v23.1 "Chart Custom Symbol" on page 128</p> | <p>Adds a predefined shape or symbol for the marker point.</p>  <p>The chart shows Price vs Units in Stock. It features two data series: 'Economy' (blue line with downward arrow markers) and 'Deluxe' (red line with downward arrow markers). A legend in the top right corner allows for filtering between 'Economy' and 'Deluxe'.</p> |
| <p>"Chart Description Style" on page 131</p> | <p>Specifies Chart Description font-related and positioning attributes.</p> |
| <p>"No Data Caption Style" on page 117</p> | <p>Specifies font-related properties for the No Data Caption.</p>  <p>The image shows a placeholder caption that reads "No data was found" in red text on a white background.</p> |
| <p>"Gradient Style" on page 85</p> | <p>Allows the user to add gradient fill to represent data.</p>  <p>The chart shows a data series with a gradient fill transitioning from yellow to green to blue.</p> |


| Element | Description |
|--|--|
| <p><i>Crosstab Filter</i></p> |  <p>"Pivots" and groups the data for easier comparison.</p> |
| <p><i>"Group Drillthrough" on page 140</i></p> |  <p>Adds grouped data point links which, when clicked, display a basic drill-through report showing the detail data used to create the group.</p> |
| <p><i>"Input Selection" on page 97</i></p> |  <p>Provides the ability to select chart areas or points with the mouse.</p> |
| <p><i>"Legend" on page 70</i></p> |  <p>Specifies Legend border, background, location, and other properties.</p> |
| <p><i>"The Resizer" on page 83</i></p> |  <p>Adds "resizing handles" to the bottom and right sides of the canvas.</p> |

| Element | Description |
|----------------------------------|---|
| "Trend Line" on page 134 |  <p>Generates a line on the chart indicating the "trend" of the data.</p> |
| "Zoom Control" on page 138 |  <p>Allows the user to "zoom" into the canvas, magnifying a portion of it.</p> |
| "Series.Area " on page 147 |  <p>Adds this data visualization to the canvas.</p> |
| "Series.Area Range" on page 169 |  <p>Adds this data visualization to the canvas.</p> |
| "Series.Area Spline" on page 190 |  <p>Adds this data visualization to the canvas.</p> |

| Element | Description | |
|---|---|--|
| <p>"Series.Area Spline Range" on page 212</p> |  | <p>Adds this data visualization to the canvas.</p> |
| <p>"Series.Bar" on page 233</p> |  | <p>Adds this data visualization to the canvas.</p> |
| <p>"Series.Bar Range" on page 263</p> |  | <p>Adds this data visualization to the canvas.</p> |
| <p>"Series.Bubble" on page 477</p> |  | <p>Adds this data visualization to the canvas.</p> |
| <p>"Series.Funnel" on page 284</p> |  | <p>Adds this visualization to the canvas.</p> |

| Element | Description |
|------------------------------|--|
| "Series.Heatmap" on page 304 |  <p>Adds this visualization to the canvas.</p> |
| "Series.Line" on page 325 |  <p>Adds this data visualization to the canvas.</p> |
| "Series.Pie" on page 348 |  <p>Adds this data visualization to the canvas.</p> |
| "Series.Pyramid" on page 380 |  <p>Adds this data visualization to the canvas.</p> |
| "Series.Scatter" on page 399 |  <p>Adds this data visualization to the canvas.</p> |

| Element | Description |
|--------------------------------|--|
| "Series.Spline" on page 422 |  <p>Adds this data visualization to the canvas.</p> |
| "Series.Waterfall" on page 442 |  <p>Adds this data visualization to the canvas.</p> |
| "Series.Whiskers" on page 461 |  <p>Adds this data visualization to the canvas.</p> |

 There is no "Series.Polar" element. To produce a Polar chart, set the Chart Canvas element's **Polar** attribute to *True*, which converts Cartesian series (Area, Line, Bar, etc.) to the Polar/Radial coordinate system.

Default Colors

If no colors are specified for a Series and no theme has been applied, the following colors will be used, in left-to-right order:



| | | | | | | | | | |
|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| #2F7E- D8 | #0D23- 3A | #8BBC- 21 | #9100- 00 | #1AAD- CE | #4929- 70 | #F28F- 43 | #77A1- E5 | #C425- 25 | #A6C9- 6A |
|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|

If more than ten colors are needed, colors will be re-used, starting from the left again.

Or, enter the token @Gradient to use a gradient fill to represent the data. For more information, see "Gradient Style" on page 85.

Refreshing Chart Data

Chart visualizations can be updated by refreshing their data using two methods:

- The **Refresh Series Timer** element is discussed in the documentation for individual Series elements.
- The [Logi JavaScript API for Chart Canvas Charts](#) is a JavaScript library that developers can use to load data into Chart Canvas Charts, updating the visualizations whenever desired.

X-Axis and Y-Axis Elements

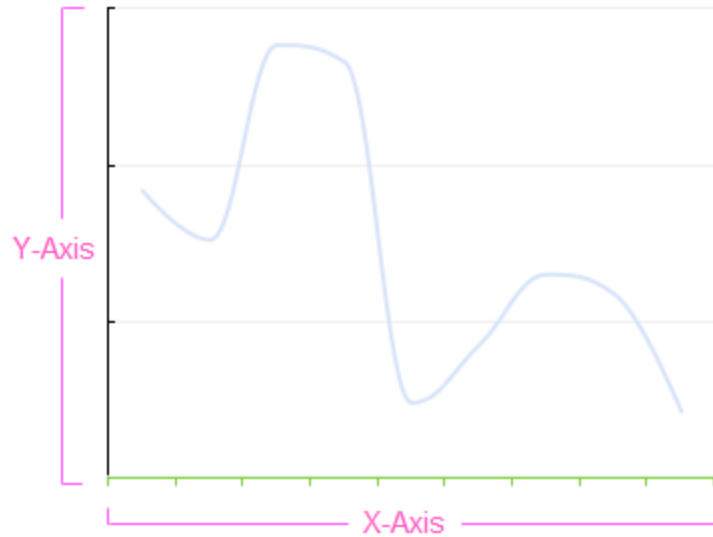
When the Chart Canvas element renders charts that use Cartesian coordinates, an X-axis and a Y-axis are drawn on the chart.

The following topics discuss how these axes can be configured:


- [Axis Element Attributes](#)
- [Plotting Negative Values](#)
- [Styling the Axis Caption](#)
- [Styling the Axis Labels](#)
- [Adding a Marker Line](#)
- [Adding a Marker Band](#)
- [Configuring Major Ticks and Grid Lines](#)
- [Configuring Minor Ticks and Grid Lines](#)
- [Showing Totals for Stacked Series](#)
- [Adding Multiple Axes](#)

About the Axis Elements

When a chart uses Cartesian coordinates, the X- and Y-axes provide the two references used by the eye to compare data values. Though they can be reversed, we generally think of the X-axis as the "label" and the Y-axis as the "data value".



You can think of them as an "overlay" on the canvas, providing the reference points over which the data values will be drawn. The **X-Axis** and **Y-Axis** elements are children of the Chart Canvas and can be used to control a variety of visual elements in the chart associated with the axes.


 A Chart Canvas element will draw an X- and Y-axis by default; the X-Axis and Y-Axis elements are only needed if you want to customize the axis.

Axis Element Attributes

The X-Axis and Y-Axis elements have the same attributes, except as noted:

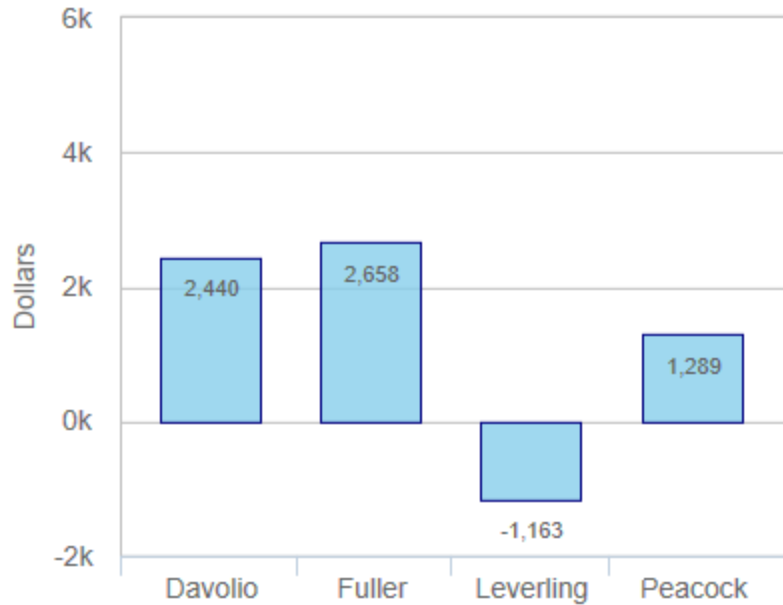
| Attribute | Description |
|-----------------------|--|
| Auto Rotation Degrees | (X-Axis only) Automatically adjusts the chart's horizontal axis when the data points become crowded. By default, the value for this attribute is "undefined" and labels will word-wrap, when possible. When undefined, the top and bottom axes default to a -45 degree rotation. To utilize this feature, enter a numeric value between 0-90. |
| Axis Padding Left | (X-Axis only) When the X-axis presents Numeric or DateTime data, sets the padding between the edge of the plot area and the first data value, as a percentage of the axis length. This is useful for preventing the first data value from appearing at the very edge of the plot area. Set to 0 to remove all padding. The default value is 1. |
| Axis Padding Right | (X-Axis only) When the X-axis presents Numeric or DateTime data, sets the padding between the edge of the plot area and the last data value, as a percentage of the axis length. This is useful for preventing the last data value from appearing at the very edge of the plot area. Set to 0 to remove all padding. The default value is 1. |
| Axis Padding Top | (Y-Axis only) Sets the padding between the edge of the plot area and the highest data value, in pixels. This is useful for preventing the highest data value from appearing right at the edge of the plot area. The default value is 5 pixels. |
| Axis Type | Sets the scaling type for the axis. This can be a "smooth" scale, consistent with a stream of numbers or dates/times or, when working with the X-Axis element, it can also be a category (e.g. Apples, Oranges) |

| Attribute | Description |
|-------------------------|---|
| | <p>which are often visualized using Bar and Pie charts. The default value varies based on the axis column data type: Text = <i>Category</i>, DateTime = <i>DateTimeLinear</i>, Numeric = <i>NumericLinear</i> or <i>NumericLogarithmic</i>.</p> |
| Caption | <p>Specifies the text of the axis caption.</p> |
| Fixed Scale Lower Bound | <p>Sets the lower scaling boundary of the axis. Use this attribute, for example, to cause negative Y-axis values to be plotted below the zero line (see "Plotting Negative Values" on page 53). Set this value to a number, or leave it blank for automatic scaling. If you specify a lower bound, then you must also specify an upper bound.</p> |
| Fixed Scale Upper Bound | <p>Sets the upper scaling boundary of the axis. Set this value to a number, or leave it blank for automatic scaling. If you specify an upper bound, then you must also specify a lower bound.</p> |
| Hide Axis | <p>Hides the axis from view.</p> |
| Hide First Label | <p>Hides the first tick mark label. The default value is <i>False</i>.</p> |
| Hide Last Label | <p>Hides the last tick mark label. The default value is <i>False</i>.</p> |
| Line Color | <p>Sets the axis line color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233.</p> |
| Line Color Transparency | <p>Specifies the transparency of the axis line color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other.</p> |

| Attribute | Description |
|-----------------------------|--|
| Line Thickness | Specifies the thickness of the axis line, in pixels. The default value is <i>1</i> pixel. |
| Opposite Side | Specifies that the axis be displayed on the opposite side from its normal position, or <i>Left</i> for vertical axes and <i>Bottom</i> for horizontal axes. The default value is <i>False</i> . |
| Reverse Axis | Reverses the axis so low and high values are swapped. Chart is then rendered to match. For example, if Y-axis is reversed, low values are at the <i>top</i> of the canvas and the chart is drawn "downward" from it. |
| Reversed Stack Series Order | (Y-Axis only) When there are multiple Series in use, or a <code>DataLayer.Crosstab</code> which generates multiple Series, and the Series are stacked, this attribute reverses how the values are stacked. |
| Spacing | Sets the space, in pixels, between the legend and chart plot area. The default value is <i>10</i> pixels. |
| Tick Count | <p>Sets the number of ticks for your X and Y-axis. The default value is <i>Undefined</i>. Set this value to a number greater than 2.</p> <p> Tick Count can only be used with Linear axes; DateTime, Logarithmic, and Category axes are not affected.</p> |

Plotting Negative Values

Charts are usually scaled automatically but, to plot negative values, you need to adjust the scaling of the axes "boundaries":

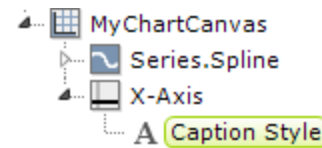


| Element - ChartYAxis | |
|-------------------------|---------|
| Optional Attributes | |
| Axis Padding Top | |
| Axis Type | |
| Caption | Dollars |
| Fixed Scale Lower Bound | 500 |
| Fixed Scale Upper Bound | 5000 |
| Hide Axis | |
| Hide First Label | |

In the example shown above, the Y-Axis element's **Fixed Scale Lower Bound** and **Fixed Scale Upper Bound** attributes have been set to show negative values below zero by setting custom chart scaling.

Styling the Axis Caption

The axis caption, set in the X-Axis and Y-Axis element's Caption attribute, can be styled by adding a child **Caption Style** element.



The Caption Style element has an **Alignment Horizontal** attribute for positioning the caption along the axis. For an X-axis, the options are *Left*, *Center* (the default), and *Right*. For a Y-axis, the options are the same but apply vertically, where *Left* = Bottom, and *Right* = Top.

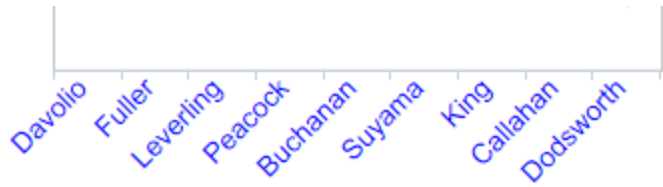
The **Caption Spacing** attribute allows you to adjust the space between the caption and its axis, in pixels.

Caption Style also has a set of font-related attributes that allow you to specify font family, color, size, etc.

A **Format** attribute allows you to easily format the caption text.

Styling the Axis Labels

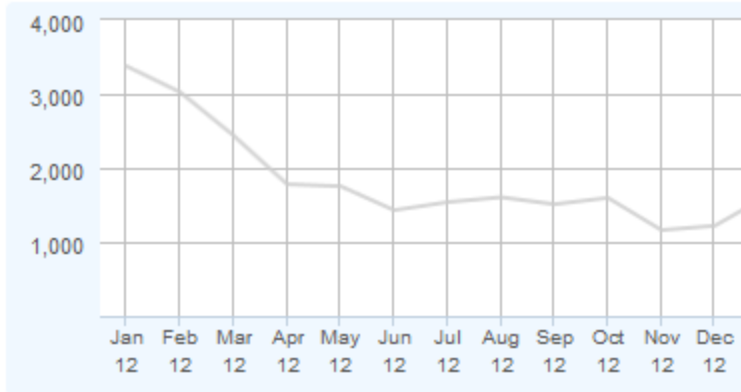
The axis labels can be styled using the **Label Style** child element. For example, Label Style has a set of font-related attributes that allow you to specify font family, color, size, angle, etc.



In the example above, the labels have been styled to have a 45-degree font angle and a Blue font color.



Labels can also be arranged into multiple rows, using the element's **Stagger Labels** attribute, as shown above. This element also has **Offset X** and **Offset Y** attributes that allow you to shift the entire set of labels in any direction and a **Format** attribute that allows you to format the label data, using standard Logi formatting options. For more information on Logi formatting, see *Format Data*.



If you want to "stack" the labels into two lines, as shown above, it's easy to do:

First, concatenate the text "
" into your label data at the point where you want to insert the line break. Here's an example SQL query:

```
SELECT LEFT(DATENAME(m, start_date), 3) + '<br>' + RIGHT(DATEPART(yy, start_date), 2) AS StartDate...
```

Then, set the Label Style element's **Format** attribute to *HTML*. Finally, use (for this example) *StartDate* as the series' **X-Axis Data Column**.

The Label Style element's **Maximum Label Length** attribute lets you specify the maximum number of characters that will be displayed for a label before the text is truncated and ellipsis (...) is appended.

When used beneath a Y-Axis element, the Label Style element's **Format** attribute can include #####; (###) to display negative numbers within parentheses.

The default formatting of automatic tooltips is inherited from the **Format** attribute of the Label Style element.

Auto Rotation

Enable auto rotation to prevent overlapping labels on your horizontal axis using the Chart Canvas attribute, Auto Rotation Degrees. By default, the value for this attribute is "undefined" and labels will word-wrap, when possible. When undefined, the top and bottom axes default to a -45 degree rotation. To utilize this feature, enter a numeric value between 0-90 in the Auto Rotation Degrees attribute.

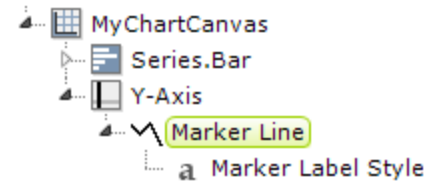
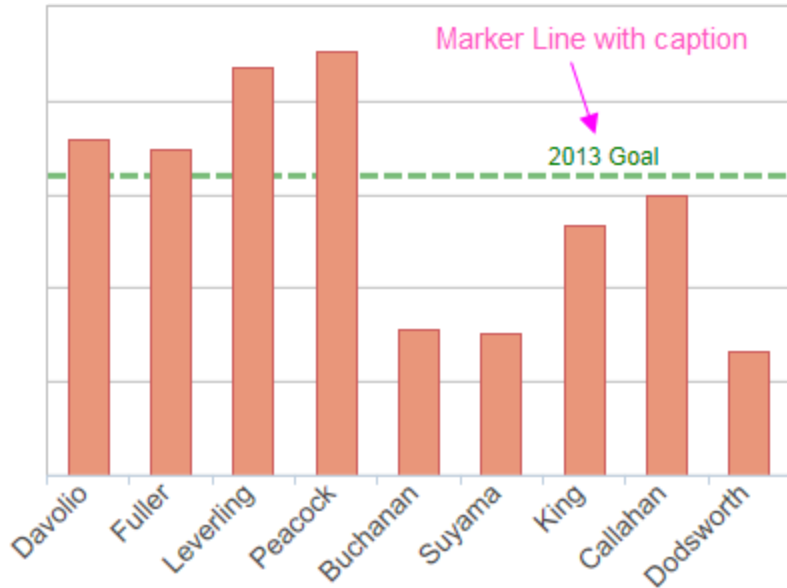
The screenshot shows the Logi Info interface. On the left, a tree view displays the chart structure. The 'Label Style' property of the 'X-Axis' is selected. On the right, the 'Element - AxisLabelStyle' properties panel is open, showing a list of 'Optional Attributes'. The 'Auto Rotation Degrees' attribute is highlighted, with a value of '-10, -45, -90'.

| Optional Attributes | |
|-----------------------|---------------|
| Auto Rotation Degrees | -10, -45, -90 |
| Font Angle | |
| Font Color | |
| Font Family | |
| Font Italic | |
| Font Size | |
| Font Weight | |
| Format | |
| Maximum Label Length | |
| Offset X | |
| Offset Y | |
| Stagger Labels | |

For more information about configuring your chart axes, see "Axis Element Attributes" on page 50.

Adding a Marker Line

The **Marker Line** child element allows you to add a value-based line to the chart, independent of the series:

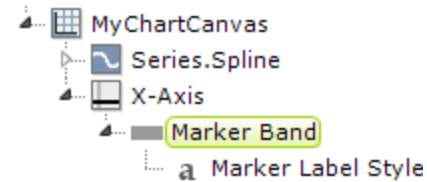
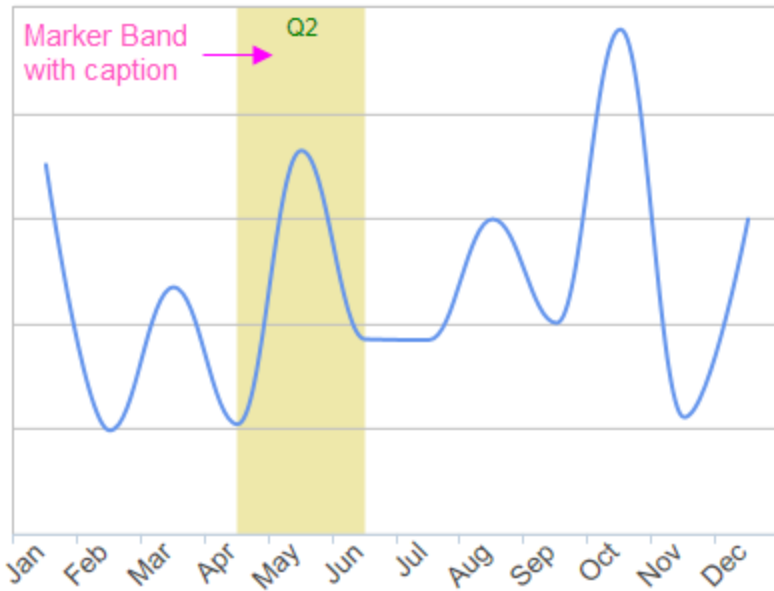


A marker line is shown in the example above. The Marker Line element has attributes that allow you to set the properties of the line, such as color, thickness, and line style, and to set its caption and value. Its **Value** attribute accepts tokens and can be set from data, however, it's a *straight* line rather than a line based on a number of data points, so using a data token that represents a range of values will produce unpredictable results.

The Marker Line element has a child element, the **Marker Label Style** element, which allows you to style the optional caption, by specifying its font properties, format, positioning, and alignment.

Adding a Marker Band

The **Marker Band** child element allows you to add a value-based band to the chart, independent of the series:

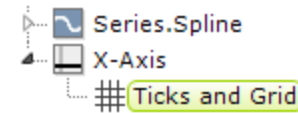
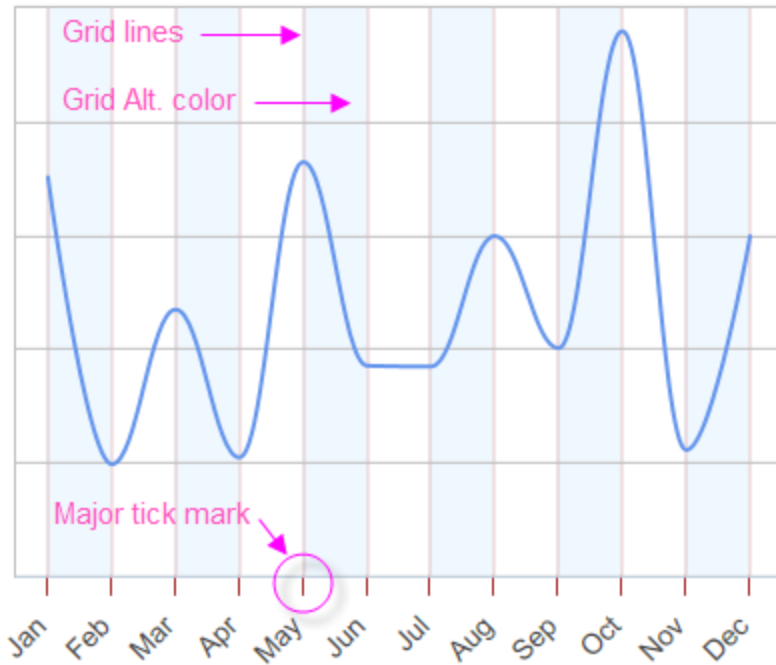


A marker band is shown in the example above. The Marker Band element has attributes that allow you to set the properties of the band, such as color and transparency, and to set its caption and values. Its **From Value** and **To Value** attributes accept tokens and can be set from data, however, it's a *straight* band rather than a band based on a number of data points, so using a data token that represents a range of values will produce unpredictable results.

The Marker Band element has a child element, the **Marker Label Style** element, which allows you to style the optional caption, by specifying its font properties, format, positioning, and alignment.

Configuring Major Ticks and Grid Lines

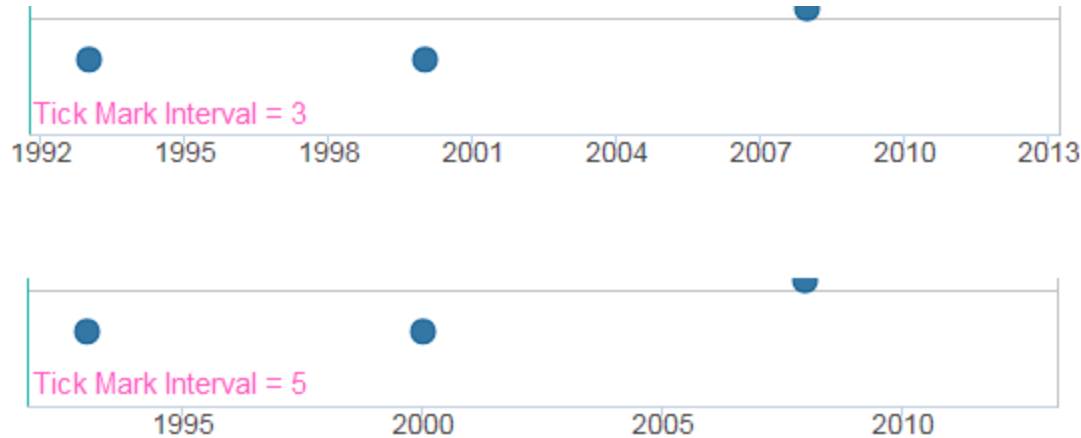
Tick marks and grid lines help associate the labels with the axis and provide visual reference points inside the plot area. "Major" tick marks and grid lines generally reference the labels and data, "minor" marks and lines are between them. Major tick marks and grid lines can be configured using the **Ticks and Grid** child element.



Grid lines and the space in between them can be customized, as shown above, in a number of ways, including color, width, and transparency. Tick marks can be customized similarly and their length, position and interval can also be set.


By default, charts are drawn *without* any X-axis (vertical) grid lines; you must add an X-Axis element with a Ticks and Grid child element beneath it to draw them.


By default, charts are drawn *with* Y-axis (horizontal) grid lines; you must add a Y-Axis element with a Ticks and Grid child element beneath it, then set the Ticks and Grid element's **Grid Line Thickness** attribute to 0 to suppress this.



As shown above, setting the Ticks and Grid element's **Tick Interval** attribute allows you to determine how many labels, tick marks, and grid lines will be shown for the axis.

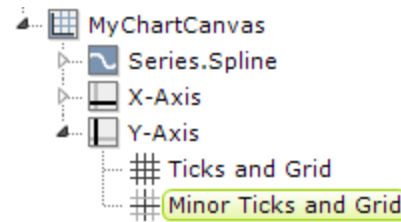
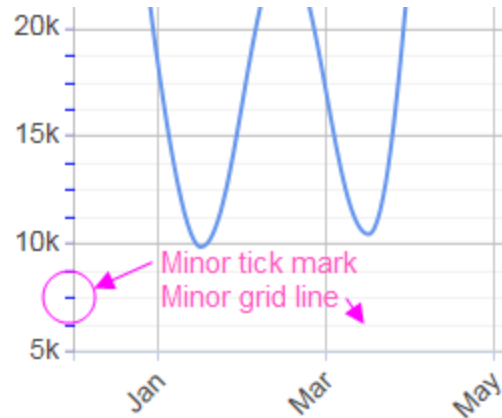
The Tick Interval is sensitive to the axis data type. For DateTime data, you can select standard values including *Years*, *Weeks*, *Hours* and more. For Numeric data, you should enter a numeric value such as *100* or *1000*. The default value, *Auto*, automatically calculates the Tick Interval based on the data values.

 Setting the Tick Interval for numeric data to an unrealistically small value will cause an error when the chart is rendered. For example, using a value of 2 for a chart whose data values range from 0 to 50,000 will cause the Logi engine to time-out after attempting to render 25,000 tick marks and labels.

 The X and Y-axis attribute 'Tick Count' can also be used to set the number of ticks for your chart. For more information, see "Axis Element Attributes" on page 50.

Configuring Minor Ticks and Grid Lines


"Minor" tick marks and grid lines are drawn in between major tick marks and grid lines. Adding them provides an additional visual reference without adding additional axis labels. Minor tick marks and grid lines can be configured using the **Minor Ticks and Grid** child element.



In the example shown above, the Minor Ticks and Grid element's **Tick Interval** attribute has been set to *1,250*. Minor grid lines can be customized in a number of ways, including color, width, and transparency. Tick marks can be customized similarly and their length, position and interval can also be set.

The Tick Interval is sensitive to the axis data type. For DateTime data, you can select standard values including *Years*, *Weeks*, *Hours* and more. For Numeric data, you should enter a numeric value, such as *100* or *1000*. The default value, *Auto*, automatically calculates the Tick Interval based on the data values.

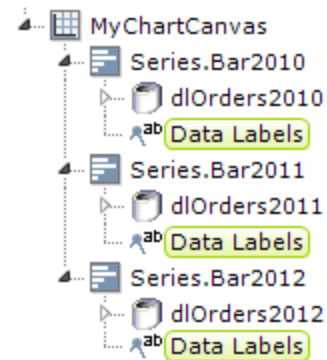
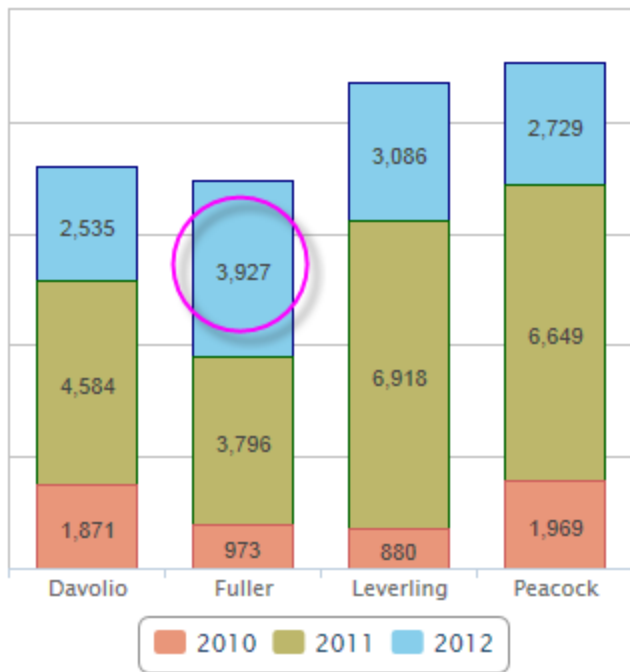
If used with a linear axis and the standard value, *Auto*, is selected, the minor Tick Interval will be calculated as one fifth of the Major Tick Interval value.

 Setting the Tick Interval for numeric data to an unrealistically small value will cause an error when the chart is rendered. For example, using a value of 2 for the chart above will cause the Logi engine to time-out after attempting to render 25,000 tick marks and labels.

Showing Totals for Stacked Series

When multiple Series are used and stacked, the Data Labels element can be used to display the individual data values. However, what if you want to display a *total* of the stacked values?

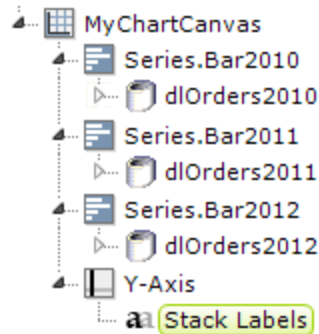
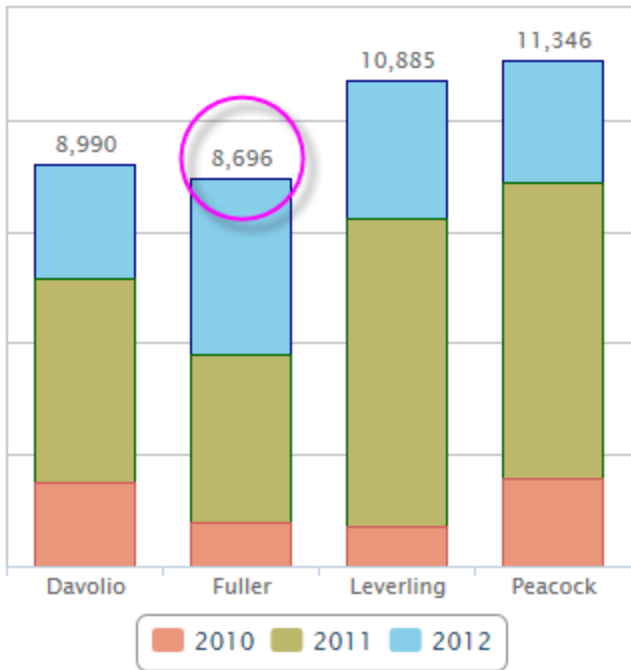
The examples below illustrates the first scenario: Data Labels elements used beneath its Series.Bar elements, resulting in individual values being displayed within the bars.



Stacked Series.Bar using Data Labels to display individual values

💡 One Data Labels element attribute, Series Name as Caption, allows you to specify the y-axis label, x-axis label, or legend label as the data point value inside the chart.

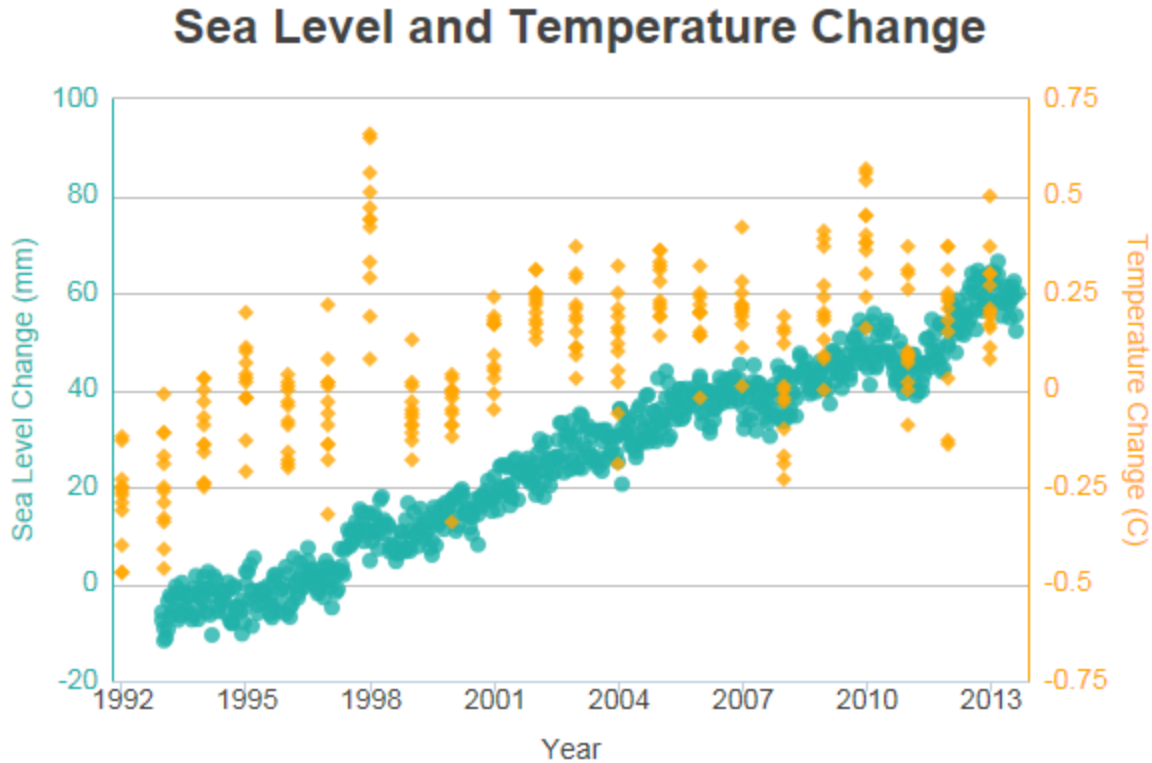
However, if you don't use Data Labels but instead use a Y-Axis element and its **Stack Labels** child element, the stacked values will be totaled and displayed above the bars, as shown below. The Stack Labels element allows you to configure the font-related, format, and positioning attributes of the total text.



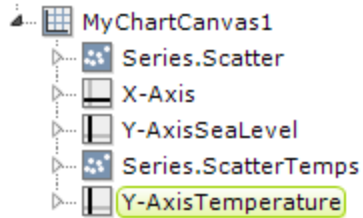
Stacked Series.Bar using Y-Axis and Stack Labels to display totals

Adding Multiple Axes

It's not unusual to need to produce a chart that has multiple series, requiring a secondary axis. Here's an example:



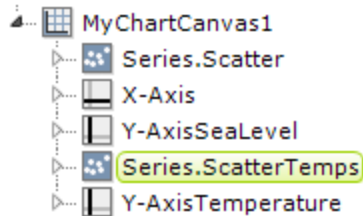
The example above shows two Series but, because they have different data value scales, a secondary Y-axis has been added on the right side. Many charts that compare multiple data series over time can use the same data scale for all series, but that's not the case here, so a secondary Y-axis is necessary. It's easy to add a secondary axis to a chart.



Two **Y-Axis** elements are used, one for each Series, each with a unique ID

| Element - ChartYAxis | |
|----------------------------|------------------------|
| Optional Attributes | |
| Axis Padding Top | |
| Axis Type | |
| Caption | Temperature Change (C) |
| ... | |
| ID | Y-AxisTemperature |
| Line Color | Orange |
| Line Color Transparency | |
| Line Thickness | |
| Opposite Side | True |
| Spacing | |

To add a second Y-axis to the chart, add a second **Y-Axis** element to the definition, as shown above. Set its **Opposite Side** attribute to *True* and give both Y-Axis elements a unique element ID. This creates the secondary Y-axis, on the *right* side of the chart.



Associate a Series with an axis using the **Linked to Y-Axis ID** attribute

| Element - Series.Scatter | |
|-----------------------------|---------------------|
| *Required Attributes | |
| Y-axis Data Column | Change |
| Optional Attributes | |
| Color | Orange |
| Combine With Series ID | |
| Disable Mouse Tracking | |
| ID | Series.ScatterTemps |
| Legend Label | |
| Linked to X-Axis ID | |
| Linked to Y-Axis ID | Y-AxisTemperature |
| Transparency | |
| X-axis Data Column | Year |
| X-axis Data Column Type | |

But which Series will use which Y-axis? You associate a Series with a Y-Axis element by setting the Series' **Linked to Y-Axis ID** attribute, as shown above, to the element ID of the desired Y-Axis element. Do this for each Series element.

Legend

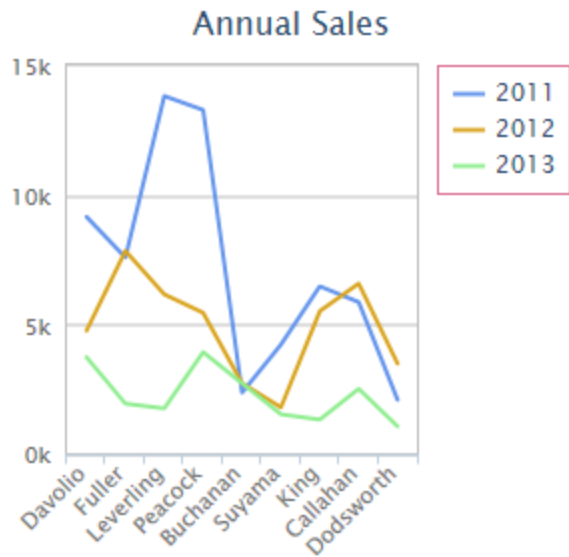
The Chart Canvas element's **Series** child elements cause a data visualization (the chart) to be rendered in the canvas. Series elements include a **Legend Label** attribute that let you specify the text that appears within the chart legend for this data. If one or more Series elements has a value in this attribute, a horizontal legend will be displayed; otherwise no legend is displayed.

The following topics discuss the CCC Legend child element:

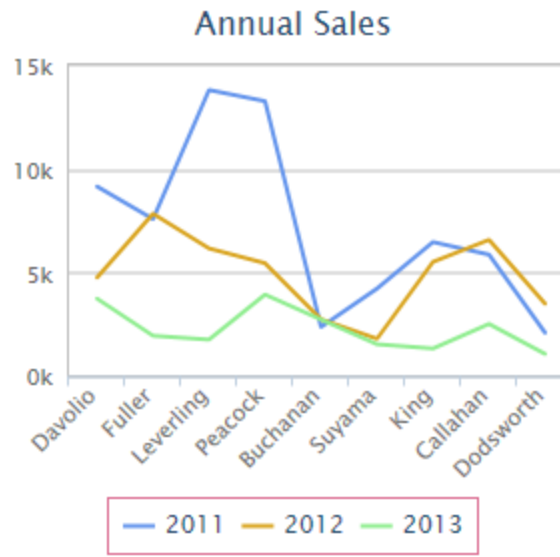
- [Legend Filtering](#)
- [Legend Attributes](#)
- [Styling the Legend Caption](#)
- [Styling the Legend Items](#)
- [Styling the Legend Navigation Arrows](#)

About the Legend

Chart legends provide visual clarity when multiple series are used. The **Legend** element can be used to control many aspects of the legend display.



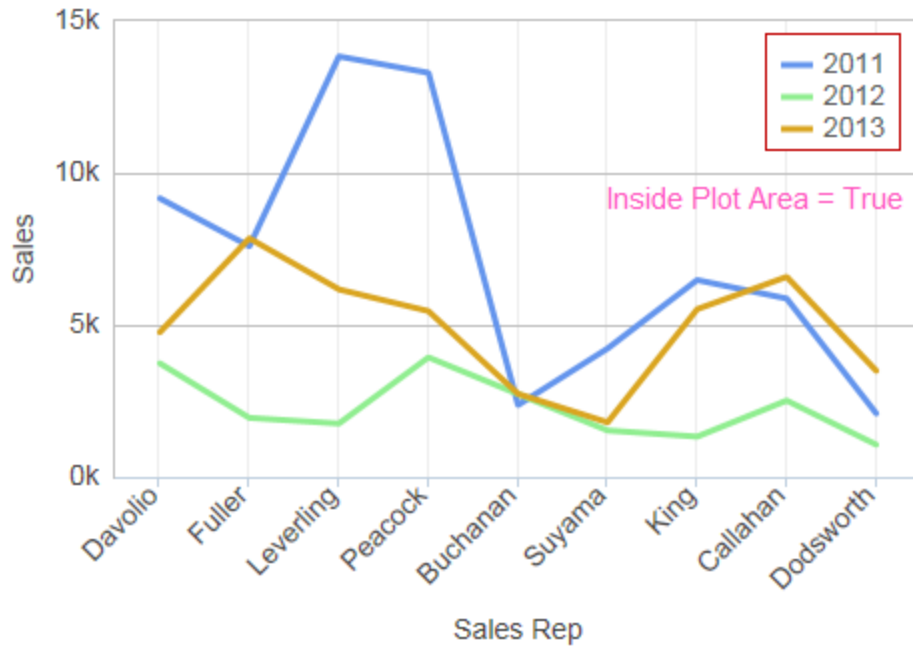
Legend Orientation = Vertical



Legend Orientation = Horizontal

In the example above, legends are shown in *Vertical* and *Horizontal* orientations, which is set using Legend element's **Orientation** attribute.

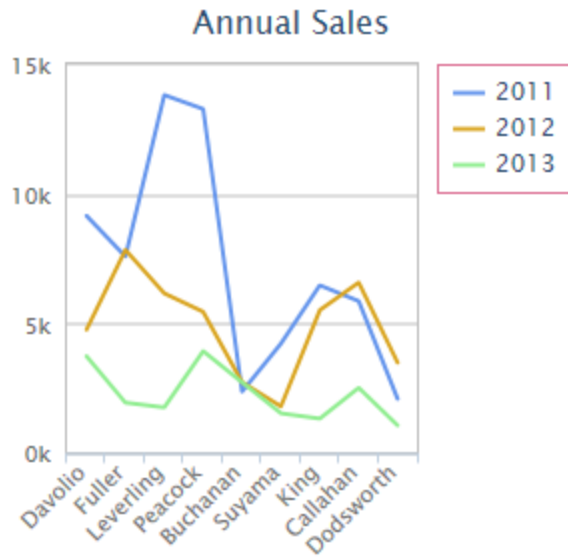
💡 The plot area is automatically scaled to accommodate the inclusion of the legend within the canvas area.



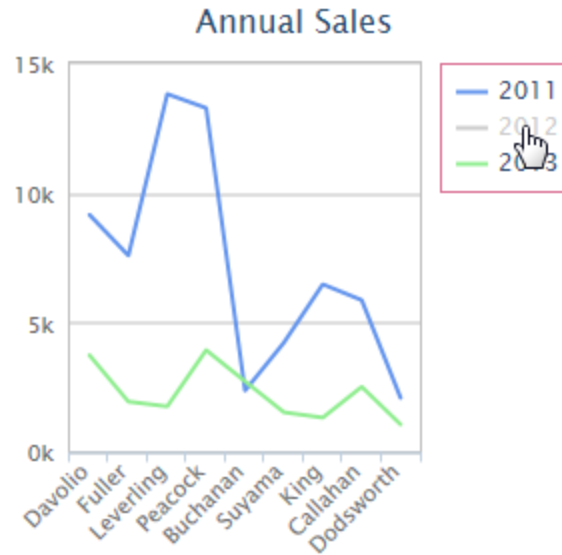
In the example shown above, the Legend element's **Inside Plot Area** attribute has been set to *True*, moving the legend inside the plot area. Additional fine positioning has been done using the element's **Offset X** and **Offset Y** attributes. The Legend element has a variety of attributes that can be used to customize its border, background, positioning, symbols, and more. These are described in detail in "Legend Attributes" on page 75. Even if the Legend element is included, the legend will *not* be displayed if there is no data to be charted.

Legend Filtering

Legend "filtering" allows users to toggle the display of series at runtime, by clicking on legend items:



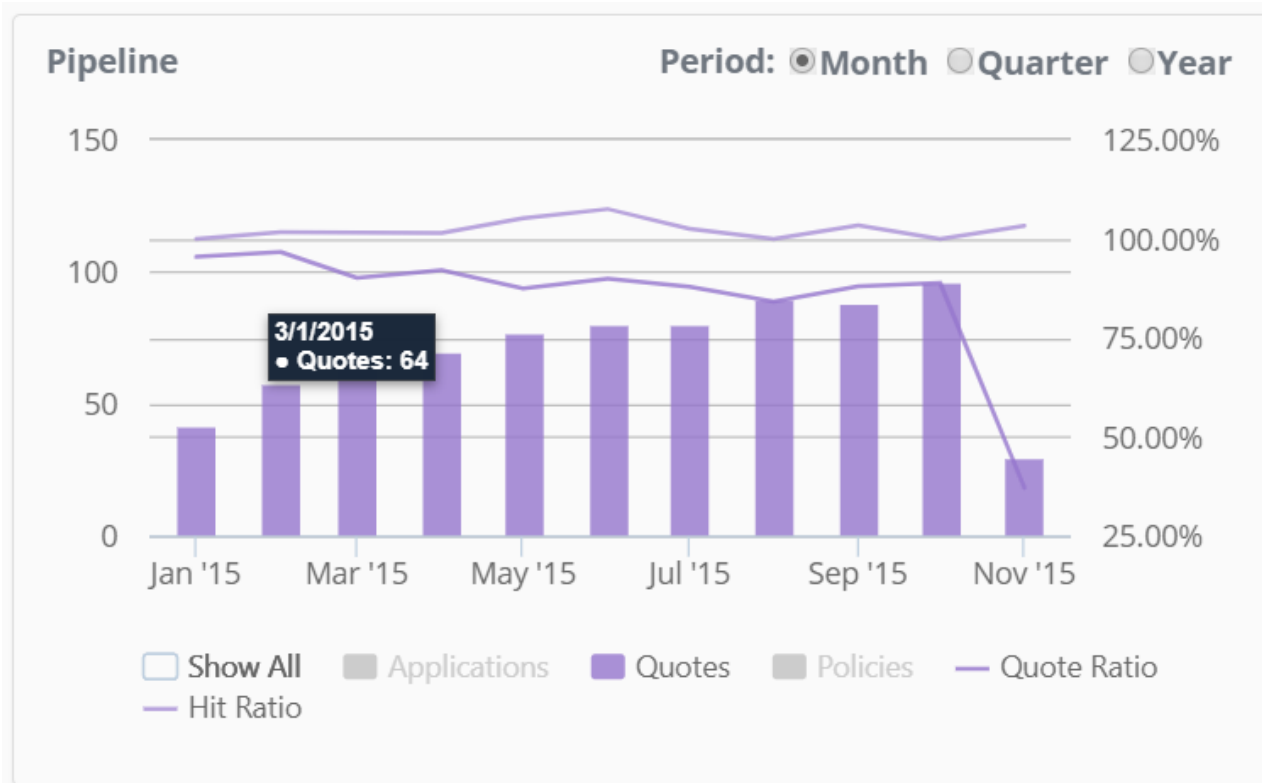
No Legend filtering



Clicking a Legend item filters out the series in the chart

When legend items are selected, the Chart Canvas element refreshes and changes the other appropriate legend objects' color attributes from HEX codes to "CCC", creating a greyed-out appearance. In the example above, the 2012 series is hidden by clicking its legend item. Clicking again will redisplay the series.

By default, the legend filtering is set to "Show All," and in turn, the Show All feature is greyed out. As you select legend items to display in the chart, the Show All legend object will redisplay and become actionable:



Legend filtering is a standard feature that's always enabled. v23.1 To disable this feature, set the `Disable Legend Filtering` attribute in the `Chart Canvas` element to `True`.

Legend Item Order




The top-to-bottom order of the items in the legend, by default, is the same as the order of the `Series` elements in the report definition. This order can be reversed by setting the `Legend` element's **Reversed Item Order** attribute to `True`.

Legend Attributes

The Legend element has the following attributes:

| Attribute | Description |
|-------------------------------|--|
| Alignment Horizontal | Sets the horizontal alignment of the legend at the <i>Left</i> , <i>Center</i> , or <i>Right</i> , of the plot area. The default value is <i>Right</i> . |
| Alignment Vertical | Sets the vertical alignment of the legend at the <i>Top</i> , <i>Middle</i> , or <i>Bottom</i> , of the plot area. The default value is <i>Top</i> . |
| Background Color | Sets the legend background color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| Background Color Transparency | Specifies the transparency of the legend background color. The lowest value of 0 specifies that the background is opaque, with no transparency. At the other end of the scale, 15 specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| Border Color | Sets the legend border line color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| Border Color Transparency | Specifies the transparency of the legend border line color. The lowest value of 0 specifies that the border line is opaque, with no transparency. At the other end of the scale, 15 specifies a completely transparent |

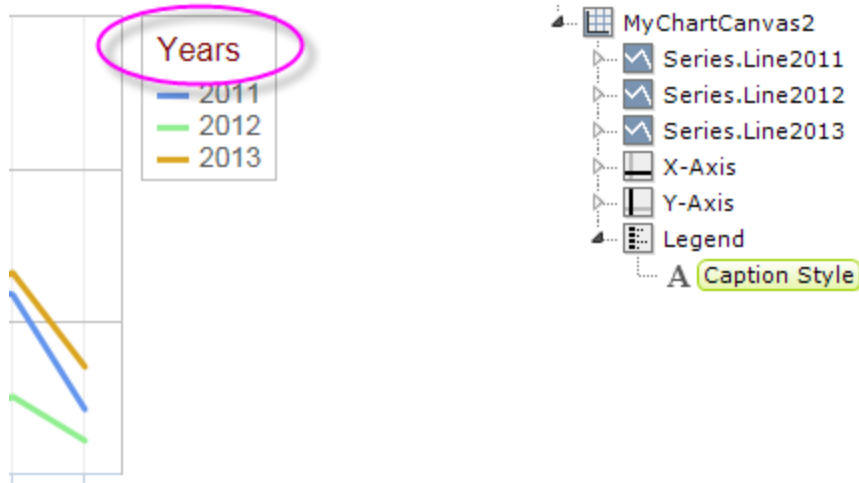
| Attribute | Description |
|--------------------|--|
| | line. Use medium-level transparency to allow different chart layers to show through each other. |
| Border Radius | Sets the amount of rounding for legend border line corners, in pixels. The default value is 4 pixels. |
| Border Thickness | Sets the canvas border line thickness, in pixels. The default value is 0, for no border. |
| Caption | Specifies the text of a caption that will appear at the top of the legend. |
| Format | Specifies a format for the legend items, i.e. the data or text from the Series element's Legend Label attribute. |
| Inside Plot Area | Specifies whether space in the canvas will be reserved for the legend (<i>False</i>) or whether it will overlap other content (<i>True</i>). The default value is <i>False</i> . |
| Legend Orientation | Specifies whether the legend will be displayed in a <i>Horizontal</i> format at the center-bottom of the canvas (the default) or a <i>Vertical</i> format at the top-right of the canvas. |
| Maximum Height | Specifies the maximum height, in pixels, of the legend. If the number of legend items causes the height of the legend to exceed this value, scroll arrows will be displayed to allow scrolling of the items. |
| Offset X | Sets the horizontal position offset of the legend relative to the chosen horizontal alignment, in pixels. |
| Offset Y | Sets the vertical position offset of the legend relative to the chosen vertical alignment, in pixels. |
| Reversed Item | Set to <i>True</i> to reverse the order of legend items. The default value is <i>False</i> . |

| Attribute | Description |
|--|--|
| Order | |
| Show All Caption | <p>Specifies the caption for the clickable "Show All" link, which shows or hides all legend items. The link appears when the number of items is equal to or greater than the Show All Threshold attribute's value. To hide the link, set Show All Threshold to a large number, such as 9999. The default value is <i>Show All</i>.</p> |
| v23.1 Show All Inverted Caption | <p>Sets fill color for default (disabled) Show All button state; the state when all series are visible. The default value is <i>Show All</i>.</p> <p> The Show All Caption button must be enabled to set up this attribute.</p> |
| v23.1 Show All Inverted Marker Color | <p>Sets text for inverted (enabled) Show All button state; the state when one or more series is hidden. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. <i>#112233</i>. Or, choose a color via the color picker by selecting the "..." option next to this field.</p> <p> The Show All Caption button must be enabled to set up this attribute.</p> |
| v23.1 Show All Marker Color | <p>Sets fill color for inverted (enabled) Show All button state; the state when one or more series is hidden. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. <i>#112233</i>. Or, choose a color via the color picker by selecting the "..." option next to this field.</p> <p> The Show All Caption button must be enabled to set up this attribute.</p> |
| Show All Threshold | <p>Specifies the legend item threshold number that determines when the "Show All" link will appear. When the number of items is equal to or greater than this value, the link appears. To hide the link, set this attribute to</p> |

| Attribute | Description |
|--------------|--|
| | a large number, such as 9999. The default value is 4. |
| Spacing | Sets the space, in pixels, between the legend and chart plot area. The default value is 10 pixels. |
| Symbol Width | Sets the width, in pixels, of the symbol preceding legend items. The default value is 16 pixels. |
| Width | Sets the width, in pixels, of the legend. If left blank, a width will be automatically determined. |

Styling the Legend Caption

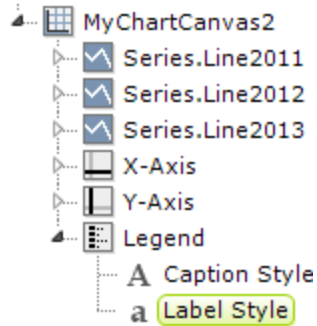
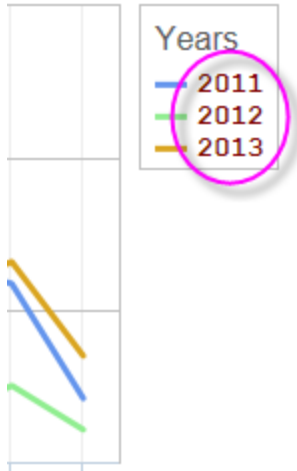
The Legend's **Caption** attribute can be set to place a caption inside the Legend:



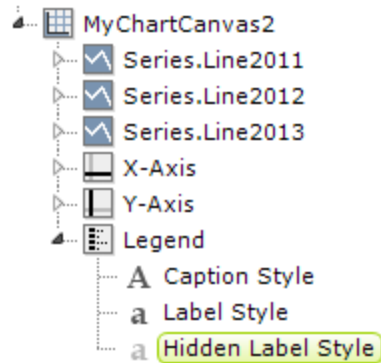
As shown above, the **Caption Style** element can be added to configure the caption's font family, color, size, and weight.

Styling the Legend Items

The text of the legend items can be styled using the **Label Style** element:



As shown in the example above, this element can be added to configure the legend items' font family, color, size, and weight.



A related element, the **Hidden Label Style** element, is used to configure of the font family, color, size, and weight that's used when a legend item is clicked to engage Legend Filtering, as shown above. The color affects both the item symbol and the item text and, without the element, defaults to light gray, as shown above.

Styling the Legend Navigation Arrows

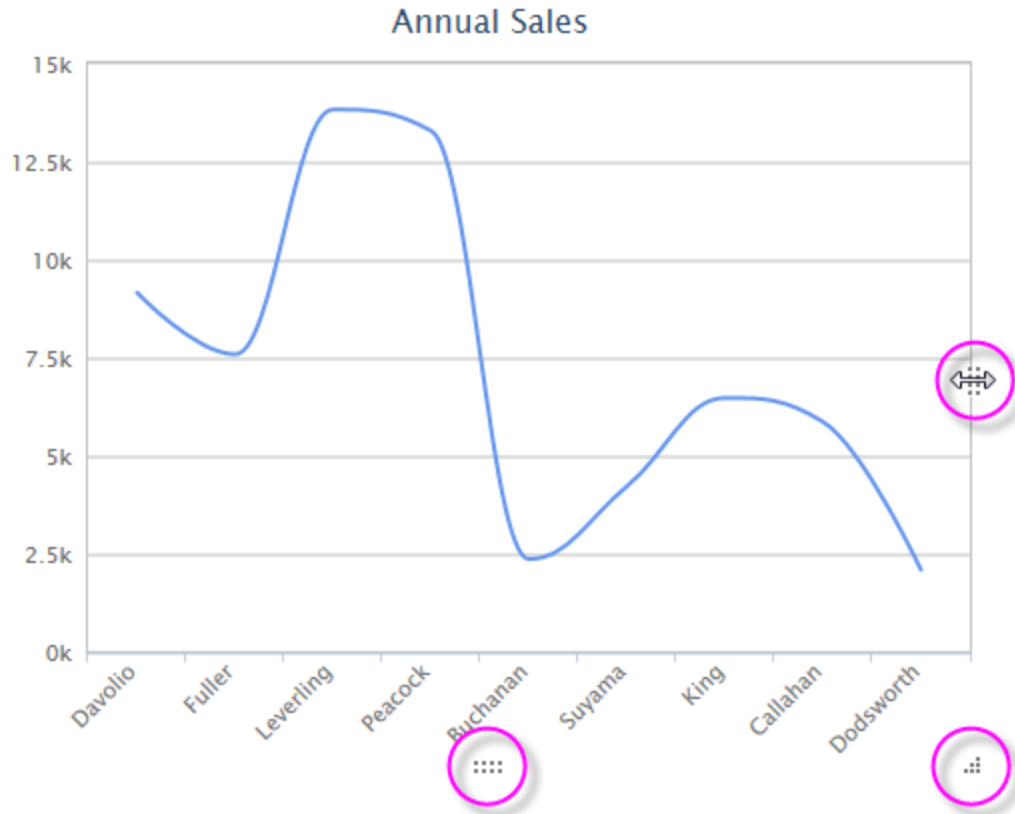
It's possible for your legend to include so many items that it exceeds the height of the legend (as set by either the height of the canvas or the legend's Maximum Height attribute).



In this case, as shown above, the legend items will automatically be placed in a scrolling list within the legend, and up-down navigation arrows and a "page counter" will appear. The **Legend Navigation Style** element can be added beneath the Legend element to control the color and size of the navigation arrows.

The Resizer

The Chart Canvas element's **Resizer** child element enables functionality that allows the user to resize the chart at runtime.



When the Resizer element is included, the "resizer handles" shown in the example above, will appear when the cursor passes near or over them. These handles are just inside the canvas border, and the user can click and drag them to resize the chart. The dimensions of the resized chart are stored for the duration of the current session only. When the Resizer is used with a chart in a

Dashboard that has a columnar layout, any resizing of the chart is saved in the Dashboard's Save File, which makes it persist beyond the user's current session.

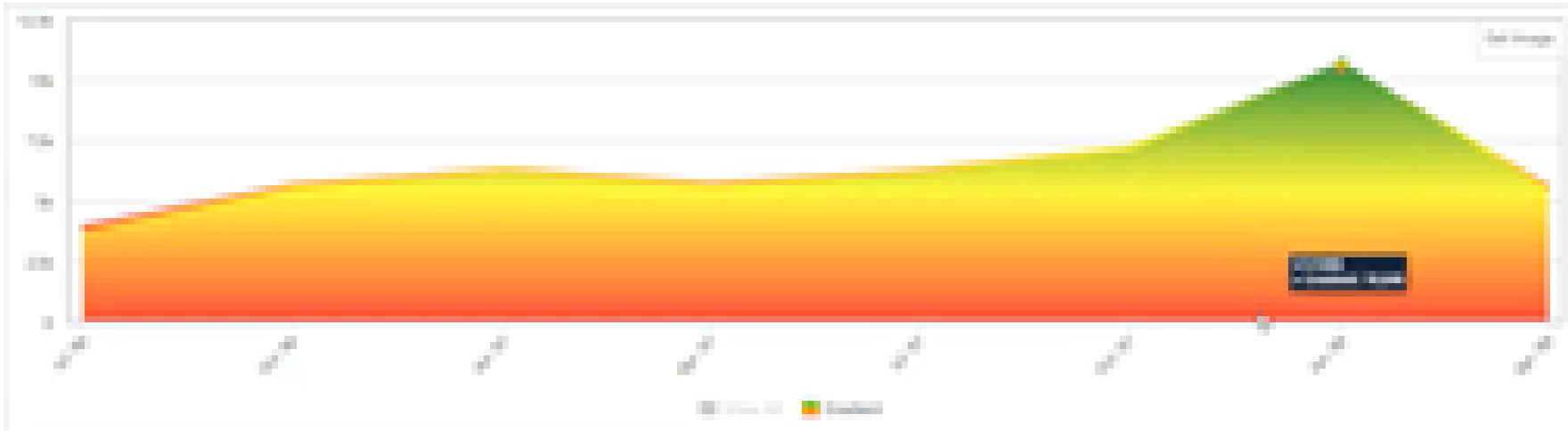
Attributes

The Resizer element has the following attributes:

| Attribute | Description |
|----------------|---|
| Maximum Height | Sets the maximum height, in pixels, that the chart may be expanded to by dragging the handles. |
| Maximum Width | Sets the maximum width, in pixels, that the chart may be expanded to by dragging the handles. |
| Minimum Height | Sets the minimum height, in pixels, that the chart may be collapsed to by dragging the handles. If left blank, the minimum is 100 pixels. |
| Minimum Width | Sets the minimum width, in pixels, that the chart may be collapsed to by dragging the handles. If left blank, the minimum is 100 pixels. |

Gradient Style

The Chart Canvas element's **Gradient Style** child element enables you to customize your visualization with gradient fills.



When adding the Gradient Style element to your chart, you have the option between Linear Gradient and Radial Gradient style. Some gradients may work better than others, depending on the chart type.

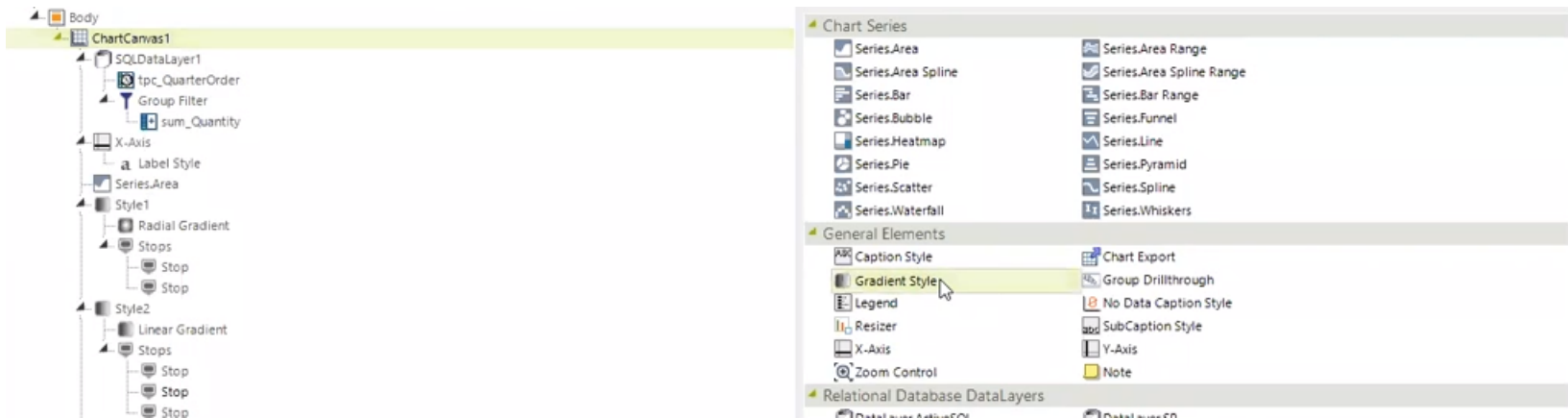
This topic discusses the following types of gradients:

- [Linear Gradient](#)
- [Radial Gradient](#)

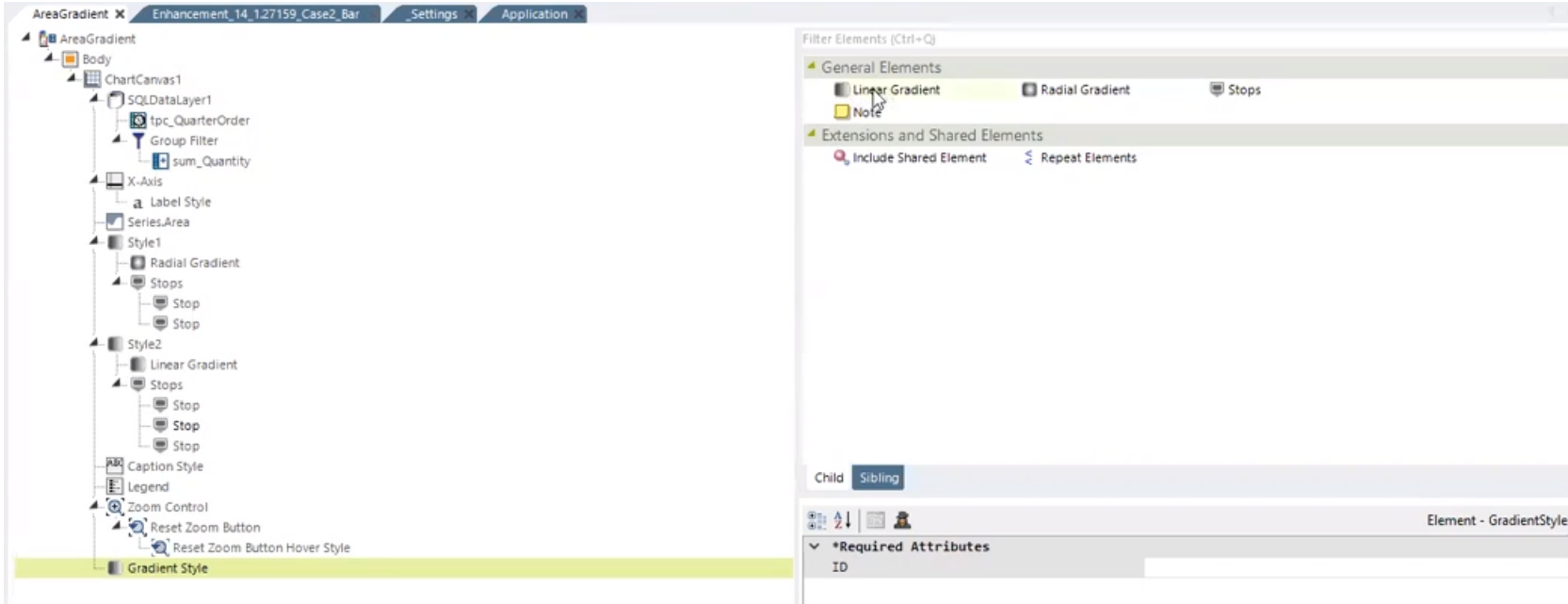
Linear Gradient

The Linear Gradient style can be used to represent two or more colors along a straight line. This style is best used with Line, Bar, Bubble, and Area charts.

1. To utilize this feature add the **Gradient Style** element to your chart:



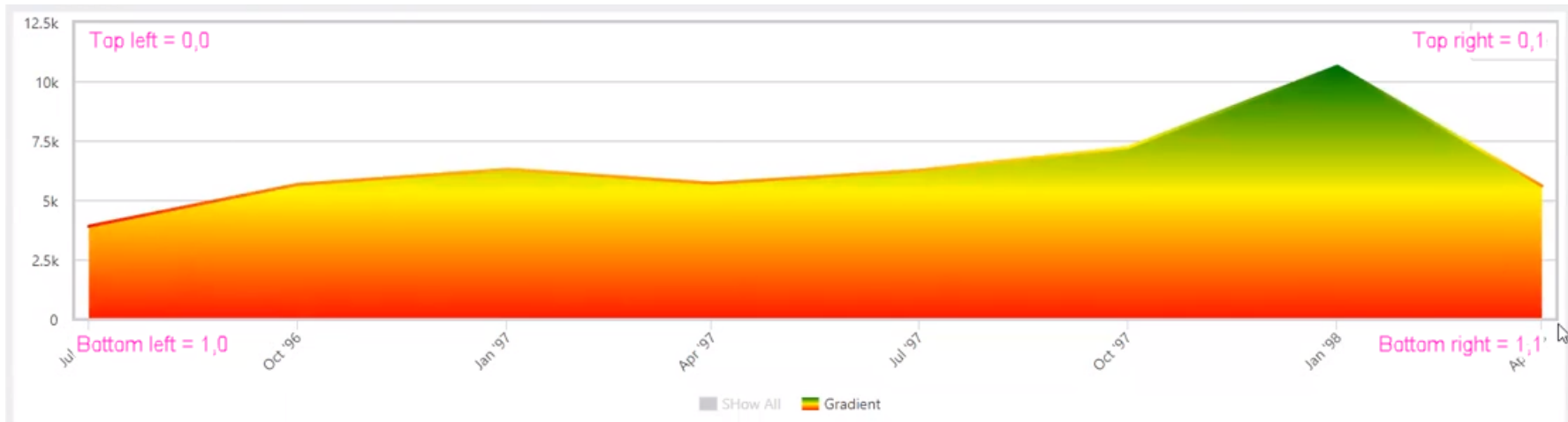
2. Choose the **Linear Gradient** element:



- Next, enter values between 0-1 (1=100, .5=50%) for the required attributes x1, x2, y1, and y2. These values act as intercepts to calculate the distance between the coordinates in your chart. Note that the 'x1' and 'y1' attributes are both reflective of the starting point, while 'x2' and 'y2' are both reflective of the gradient's end point.



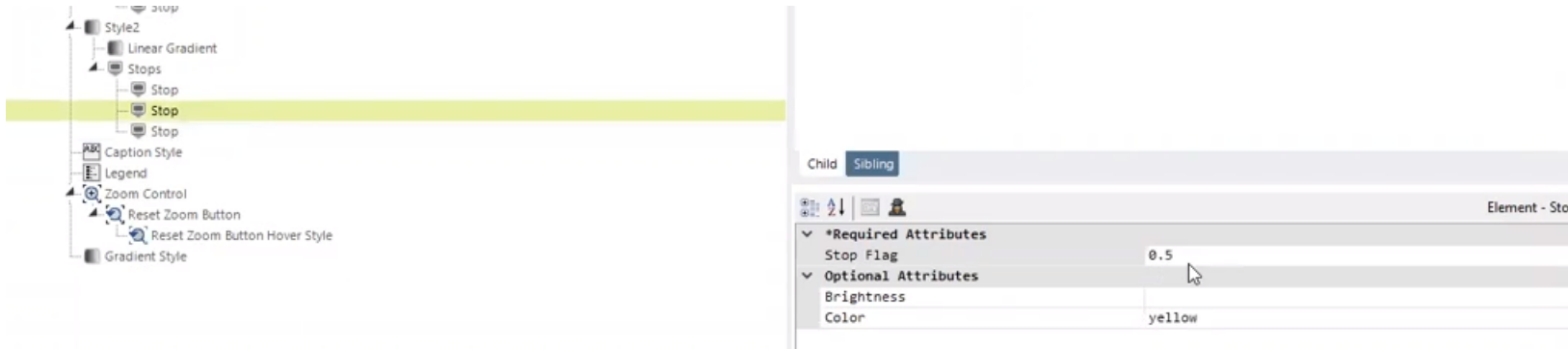
For reference, the chart below represents the coordinates that are allocated using x1, y1, x2, and y2:



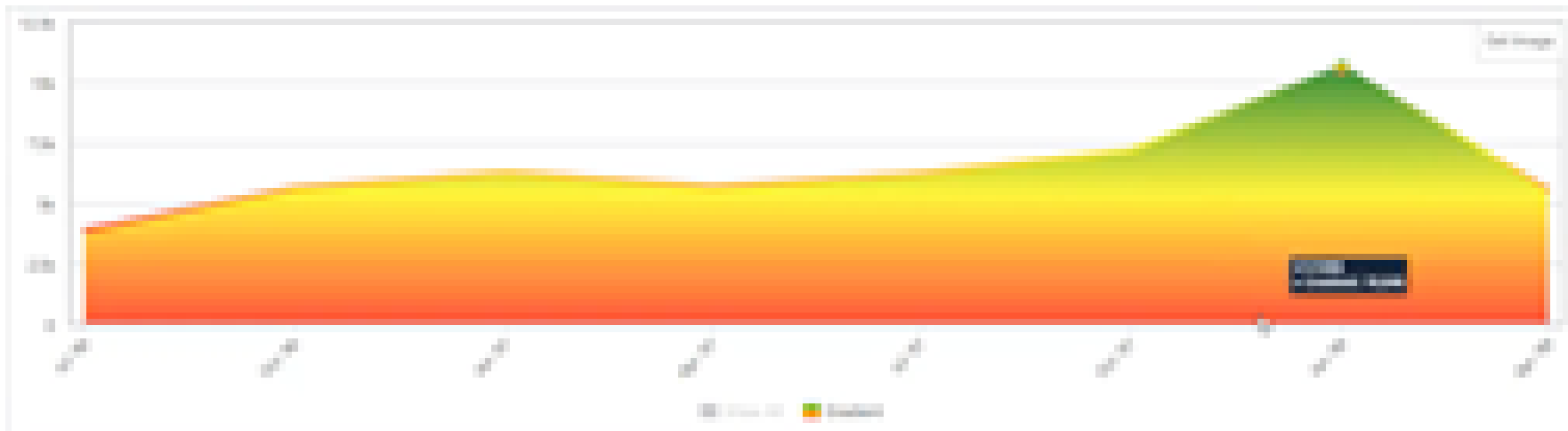
- Once you've entered the required attributes, add the @Chart token to the chart's Color attribute and choose the **Style** you want to reference.

The screenshot shows a software configuration interface. On the left is a tree view under 'Series.Area' with sub-items: 'Style1' (Radial Gradient, Stops), 'Style2' (Linear Gradient, Stops), 'Caption Style', 'Legend', 'Zoom Control' (Reset Zoom Button, Reset Zoom Button Hover Style), and 'Gradient Style'. On the right is a configuration panel with a list of actions (Action.Exit, Action.Link, Action.Popup Menu, Action.Refresh Element, Action.Show Element, Action.Javascript, Action.Map Location, Action.Process, Action.Report) and an 'Exports' section containing 'Style1-' and 'Style2-'. At the bottom, the 'Optional Attributes' section shows 'Color' set to '@Gradient.' and 'Combine With Series ID'.

5. Lastly, specify how many stops you want to appear on the chart and use the Stop Flag attribute to set the gradient fill number from 0-1 (1=100, .5=50%) to separate the stops.



In the example below, the Linear Gradient's stops are denoted in green, yellow, and red.



💡 The chart's legend is indicative of the style, color, and fill of the gradient.

You can add as many styles as you want under the Chart Canvas. Using these styles, you can add the gradient fill and color to Legend, Zoom Control, and other elements in your Chart Canvas.

Linear Gradient Attributes

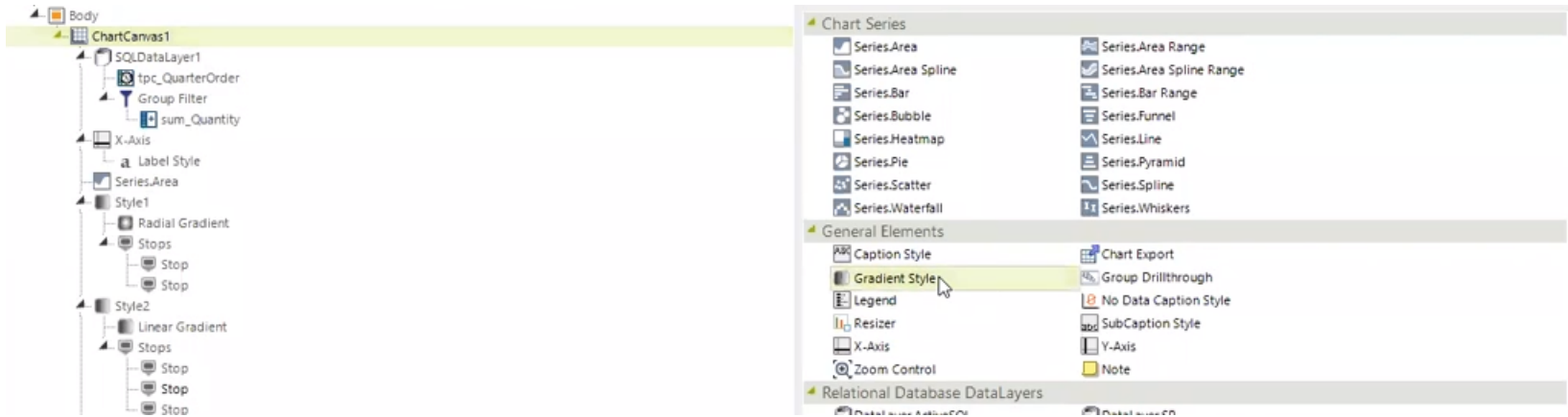
The Linear Gradient element has the following attributes:

| Attribute | Description |
|-----------|---|
| x1 | Sets the x coordinate of the first point. Enter a value between 0-100 (1=100%, .5=50%). |
| x2 | Sets the x coordinate of the end point. Enter a value between 0-100 (1=100%, .5=50%). |
| y1 | Sets the y coordinate of the first point. Enter a value between 0-100 (1=100%, .5=50%). |
| y2 | Sets the y coordinate of the end point. Enter a value between 0-100 (1=100%, .5=50%). |

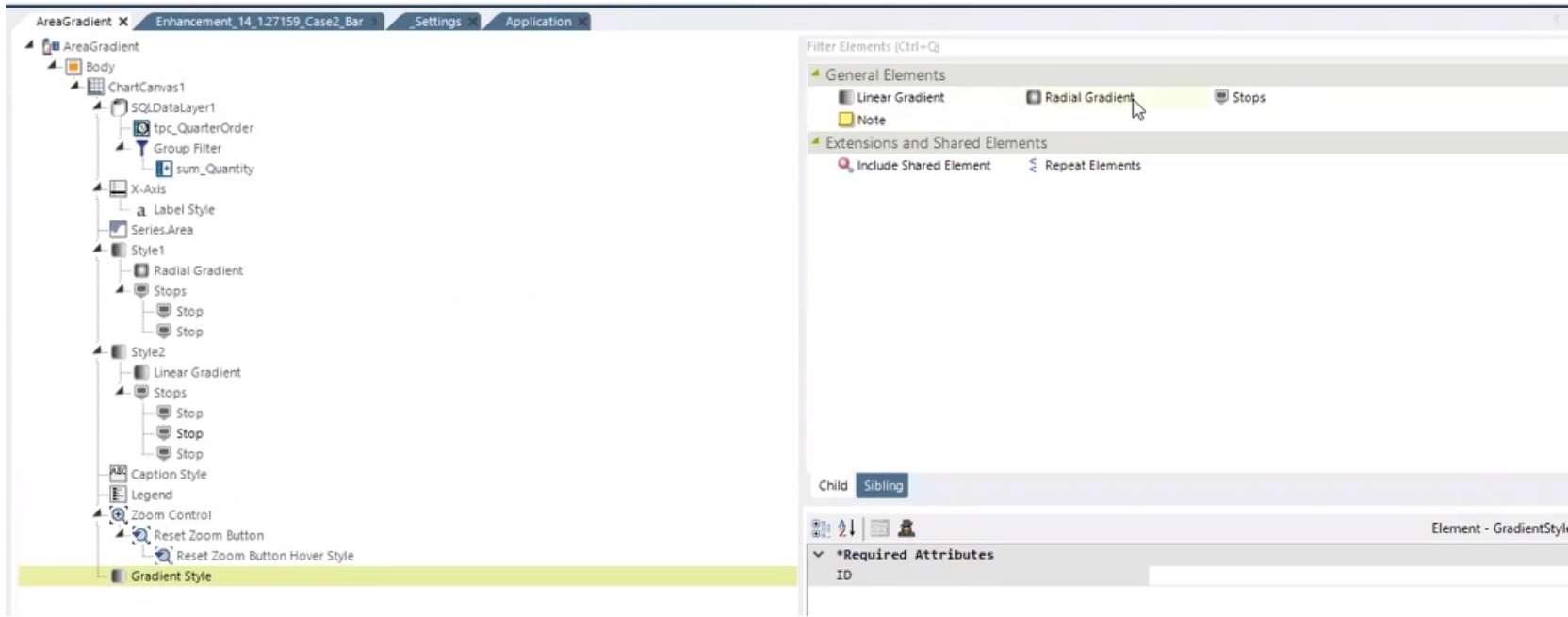
Radial Gradient

The Radial Gradient style can be used to represent two or more colors that radiate from an origin. This style is best used with Pie or Polar chart.

1. To utilize this feature add the **Gradient Style** element to your chart:



2. Choose the **Radial Gradient** element:



- Next, enter values between 0-1 (1=100, .5=50%) for the required attributes *cx*, *cy*, and *r*. These values act as intercepts to calculate the distance between the coordinates in your chart. Note that when the '*r*' attribute is set to 1, the gradient covers the entire circle.

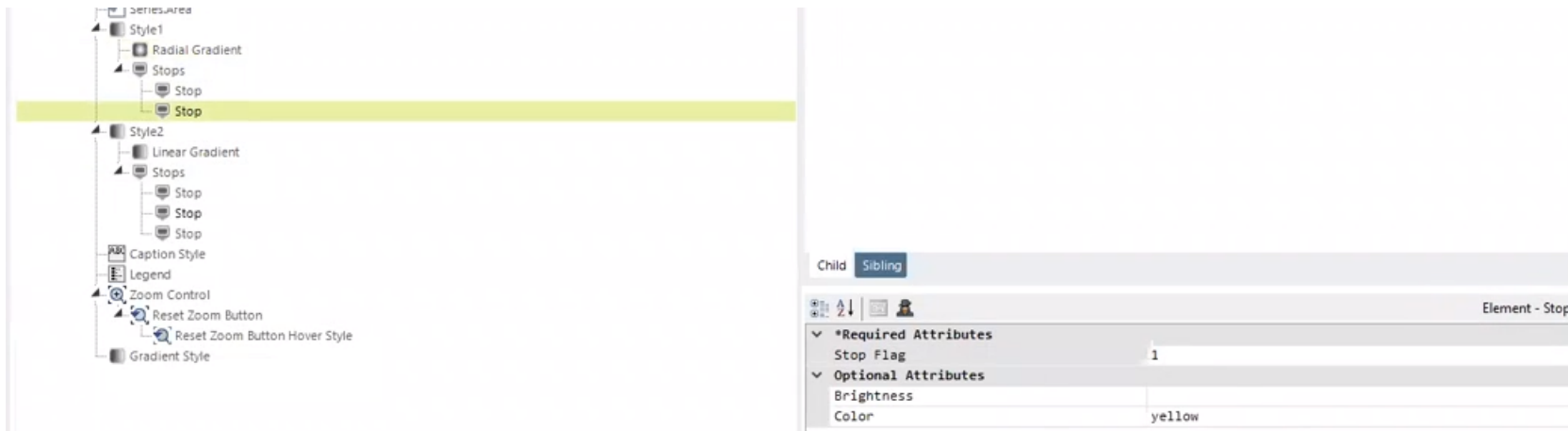
The screenshot shows the Logi Info interface. On the left is a style tree with 'Radial Gradient' selected. On the right is the 'Element - RadialGradient' property editor. The 'Required Attributes' section is expanded, showing a table with the following data:

| Attribute | Value |
|-----------|-------|
| cx | 0.5 |
| cy | 0.5 |
| r | 0.5 |

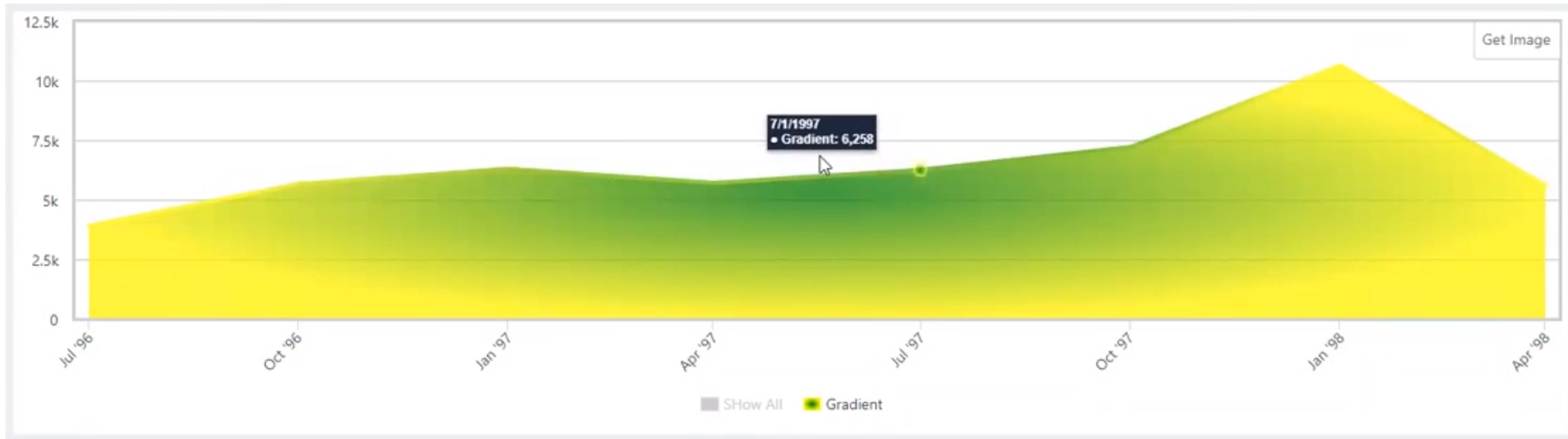
4. Once you've entered the required attributes, add the @Chart token to the chart's Color attribute and choose the **Style** you want to reference.


The screenshot shows the Logi Info interface. On the left is a style tree with 'Series.Area' selected. On the right is the 'Color' property editor. The 'Exports' section is expanded, showing a list of available styles: 'Style1' and 'Style2'. The 'Color' attribute is set to '@Gradient.', and the 'Combine With Series ID' checkbox is checked.

5. Lastly, specify how many stops you want to appear on the chart and use the Stop Flag attribute to set the gradient fill number from 0-1 (1=100, .5=50%) to separate the stops.



In the example below, the Radial Gradient's stops are denoted in green and yellow.



 The chart's legend is indicative of the style, color, and fill of the gradient.

You can add as many styles as you want under the Chart Canvas. Using these styles, you can add the gradient fill and color to Legend, Zoom Control, and other elements in your Chart Canvas.

Radial Gradient Attributes

The Radial Gradient element has the following attributes:

| Attribute | Description |
|-----------|---|
| cx | Sets the x coordinate for the circle. Enter a value between 0-100 (1=100%, .5=50%). |

| Attribute | Description |
|-----------|---|
| cy | Sets the y coordinate for the circle. Enter a value between 0-100 (1=100%, .5=50%). |
| r | Sets the radius of the circle. Enter a value between 0-100 (1=100%, the gradient will cover the entire circle). |

Input Selection

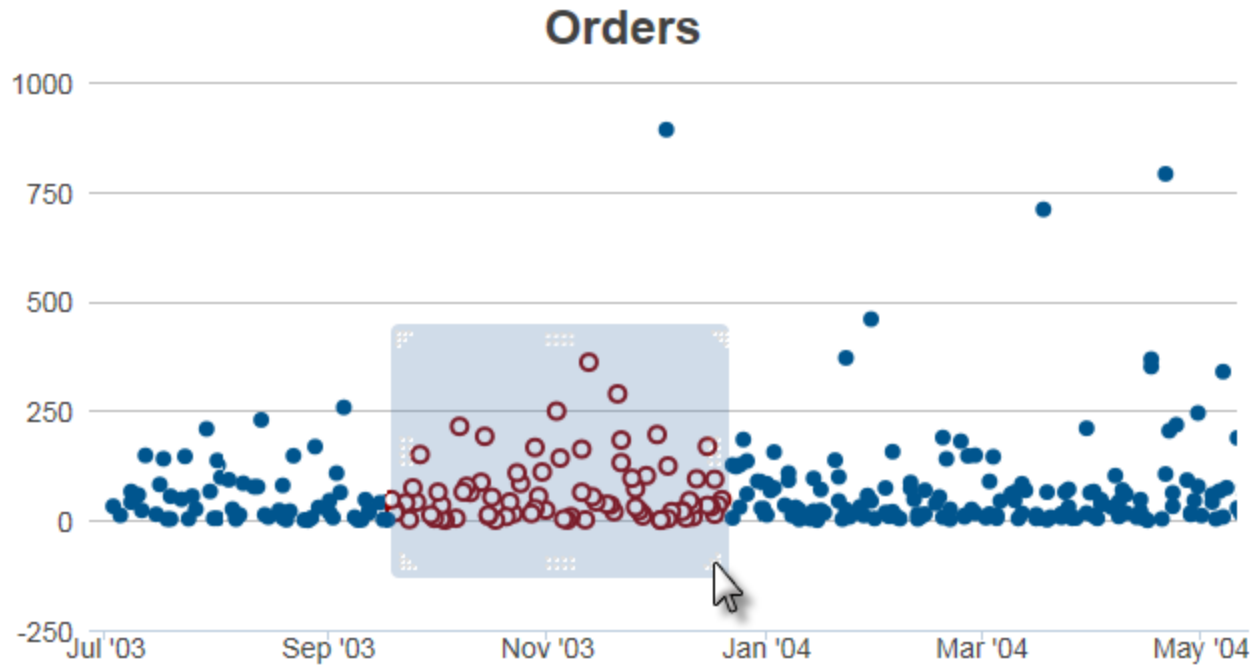
The Chart Canvas element's Series child elements cause a data visualization (the chart) to be rendered in the canvas. The **Input Selection** family of elements is available beneath some Series elements and provides the ability to interactively select portions of charts using the mouse and initiate actions using the selected data.

The following topics discuss the Input Selection child element:

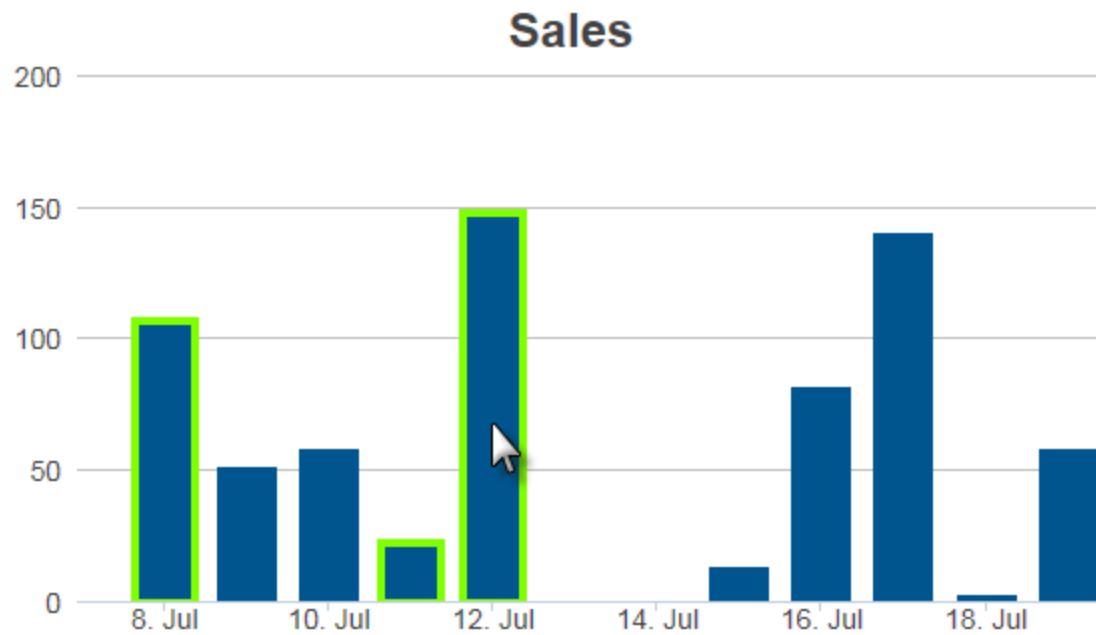
- [Using Input Selection Point](#)
- [Using Input Selection Range](#)

About Input Selection

Input Selection elements turn a chart into an input control, allowing the user to make selections on the chart at runtime. The data represented within the selected chart area can then be used with other actions. Drilling into detail data is one example of how this can be used.



In the example above, at runtime, a "selection area" has been "drawn" (by dragging the mouse) onto the chart to select values. The values within the selection area are automatically submitted to the next definition as Request tokens. Page submission may or may not be triggered when drawing the area ends. The appearance of selected areas or points can be independently styled.



The example shown above uses Input Selection Point configured to allow selection of multiple data points. In this case, the page submission is *not* triggered by the selection process.

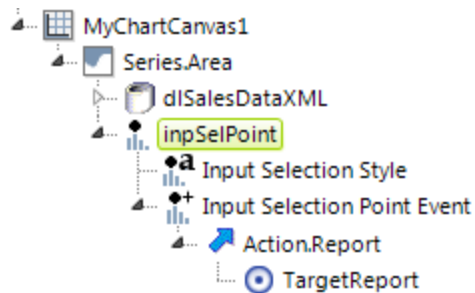
Let's examine the elements used for chart input selection.

Using Input Selection Point

The **Input Selection Point** element enables the selection, at runtime, of one or more chart data points. This element is available for use with:

- Series.Area
- Series.Area Spline
- Series.Bar
- Series.Bar Range
- Series.Bubble
- Series.Funnel
- Series.Line
- Series.Pie
- Series.Pyramid
- Series.Scatter
- Series.Spline
- Series.Waterfall

Selection can occur by clicking individual data points or by "drawing" a selection rectangle around them.



Input Selection Point works with two other elements, **Input Selection Style** and **Input Selection Point Event**, along with Action and Target elements, to produce the desired effect, as shown in the typical element arrangement above.

You can link Input Select Point to User Input elements (**Input Text** and **Input Hidden**) in the same report definition and the values selected in the chart will automatically be reflected in the input elements. Selecting multiple points in the chart will create a comma-separated list of values in the User Input elements.

The following additional User Input elements now work automatically with Input Select Point: **Input check box List**, **Input Radio Buttons**, and **Input Select List**.

Attributes

Input Selection Point attributes:

| Attribute | Description |
|------------------------|---|
| ID | (Required) This is the unique element ID and is also the name of the Request variable passed when the page input selection action occurs. If this ID matches the ID of one of the User Input elements mentioned in the previous section, that element will be automatically updated with the value(s) selected in the chart. |
| Change Flag Element ID | Specifies the ID of an element to be set to indicate that an input value has been changed by the user. This is often an Input Hidden element. |
| Excluded Point Values | Specifies a comma-separated list of one or more values that cannot be selected. The values correspond to those from the element's Point Value Column attribute. For example, a bar chart with a Relevance Filter may include a bar indicating "Other", which should not be selectable. When a chart's datalayer has a Relevance Filter child ele- |

| Attribute | Description |
|--------------------|--|
| | ment, the area represented by the filter's Irrelevance Label (the "Other" area) will automatically be the default for the Excluded Point Values attribute. |
| Point Value Column | When a point is clicked, the default value returned is that of the Series' X-axis Data Column. This attribute specifies the name of an alternate data column to be used. |
| Selection Type | Specifies how users interact with the chart. Options include: <i>ClickSinglePoint</i> (the default) lets users click on individual data points, bars or pie wedges. A previously selected point becomes deselected. Re-clicking the point deselects it. <i>ClickMultiplePoints</i> lets users click on any number of points, bars or pie wedges. Re-clicking a point deselects it. <i>ClickRangePoints</i> (for Bar, Area, Area Spline, Line, and Spline series only) lets the user select a range of values by clicking the range's starting and ending points. <i>Area</i> lets the user select points by drawing a rectangular region within X- and Y-axis charts. The chart example at the top of the page uses this selection type. This and the following "area" types are <i>not</i> supported for Funnel, Pyramid, and Pie series. <i>AreaXAxis</i> is the same as <i>Area</i> , except the Y-axis region fills the chart. Only the X-axis is selected by the user. <i>AreaYAxis</i> is the same as <i>Area</i> , except the X-axis region fills the chart. Only the Y-axis is selected by the user. |

In the target report or process task, all of the selected values can be referenced using the token `@Request.XX~`, where "XX" is the Input Selection Point element ID. The token for the example shown above would be `@Request.inpSelPoint~`.

If multiple selections are made, the `@Request` token will contain a comma-separated list of selected values. For use with SQL statements, it may be desirable to surround each individual value in the list with single quotes. This can be done using the `SingleQuote` token modifier. Thus `@Request.inpSelPoint~` which has a value of `1,2,3` becomes `@SingleQuote.Request.inpSelPoint~` which has a value of `'1','2','3'`

In SQL query "IN" clauses, this is used as "...WHERE LastName IN ('@SingleQuote.Request.inpSelPoint~')..."

The **Validation.Required** element is also available as a child element so a warning message can be displayed if the page is submitted without data points being selected.

 The Validation element's *Class* attribute will not be applied.

Input Selection Point Event

This element specifies, through its child elements, the action to be taken when point selection is completed.

| Attribute | Description |
|-----------------|--|
| ID | The unique element ID. |
| Selection Event | Specifies the type of event that will trigger the action. Options include: <i>Change</i> (the default) triggers the action when the selected status of any data point changes. <i>PointsSelected</i> triggers the action only when points are selected. <i>PointsCleared</i> triggers the action when points are deselected. |

Input Selection Style

This element allows you to style the selection point or area.

| Attribute | Description |
|--------------|---|
| Border Color | Specifies the color of the border line displayed around selected points. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. <i>#112233</i> . |

| Attribute | Description |
|-----------------------------------|---|
| Border Color Transparency | Specifies the transparency of the selected point border line color. The lowest value of <i>0</i> specifies that the line is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent line. Use medium-level transparency to allow different chart layers to show through each other. |
| Border Thickness | Sets the thickness of the selected point border line, in pixels. The default value is <i>1</i> pixel. |
| Color | Specifies the color of selected points. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. <i>#112233</i> . |
| Selection Area Color | When using an "Area" Selection Type, specifies the color of the selection rectangle background. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. <i>#112233</i> . |
| Selection Area Color Transparency | When using an "Area" Selection Type, specifies the transparency of the selection rectangle. The lowest value of <i>0</i> specifies that the line is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent line. Use medium-level transparency to allow different chart layers to show through each other. |
| Transparency | Specifies the transparency of selected points. The lowest value of <i>0</i> specifies that the line is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent line. Use medium-level transparency to allow different chart layers to show through each other. |

When using Input Selection Point, you may want to turn off the automatic Quicktips generated by the chart. This is done by setting the Chart Canvas element's **Auto Quicktip** attribute to *False*.

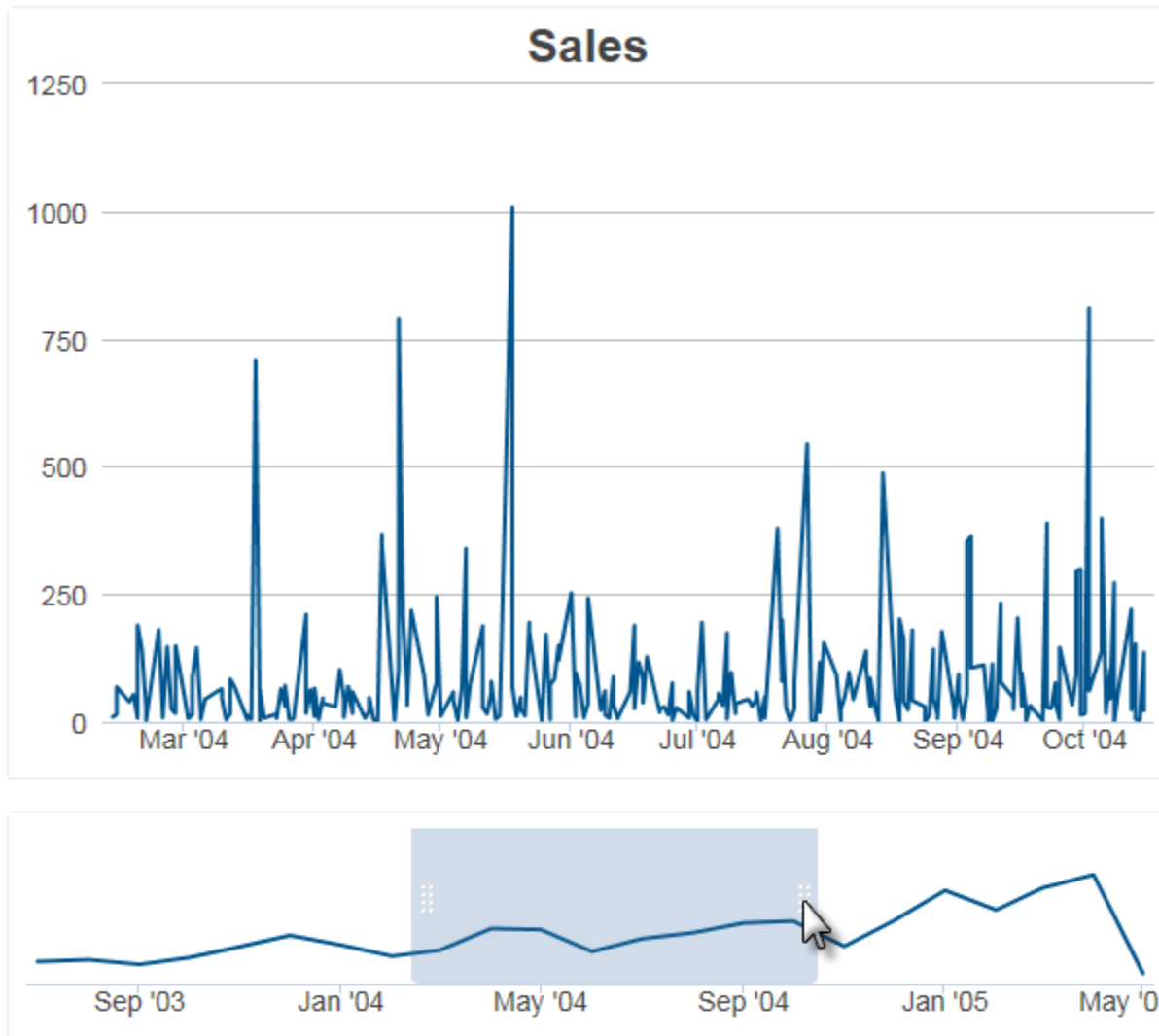
Using Input Selection Range

The **Input Selection Range** element enables the selection, at runtime, of a range of data values by drawing a selection area.

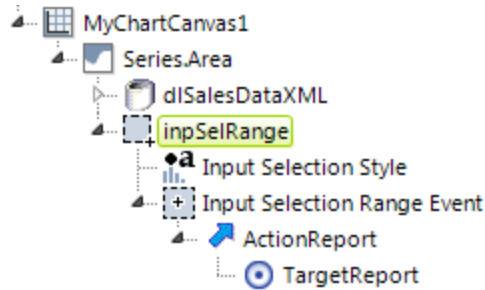
This element is available for use with:

- Series.Area
- Series.Area Range
- Series.Area Spline
- Series.Area Spline Range
- Series.Bar
- Series.Bar Range
- Series.Bubble
- Series.Line
- Series.Scatter
- Series.Spline

Here's an example:



In the example above, the upper chart displays the details of the time window selected in the lower chart.



Input Selection Range works with two other elements, **Input Selection Style** and **Input Selection Range Event**, along with Action and Target elements, to produce the desired effect, as shown in the typical element arrangement above.

Input Selection Range

This element enables the user interaction and has the following attributes:

| Attribute | Description |
|-------------------------|---|
| Change Flag Element ID | Specifies the ID of an element to be set to indicate that an input value has been changed by the user. This is often an Input Hidden element. |
| Disable Clear Selection | Specifies whether the user can clear the selection area after it's been drawn (this is done by click outside the area). Specifying <i>True</i> prevents the user from doing this. Default: <i>False</i> |
| ID | This is the unique element ID. |

| Attribute | Description |
|----------------|--|
| Max X-axis ID | Specifies the name of a Request variable that will hold the value within the right-most edge of the selection area. Not needed if Selection Type is <i>AreaYAxis</i> . |
| Max Y-axis ID | Specifies the name of a Request variable that will hold the value within the top-most edge of the selection area. Not needed if Selection Type is <i>AreaXAxis</i> . |
| Min X-axis ID | Specifies the name of a Request variable that will hold the value within the left-most edge of the selection area. Not needed if Selection Type is <i>AreaYAxis</i> . |
| Min Y-axis ID | Specifies the name of a Request variable that will hold the value within the bottom-most edge of the selection area. Not needed if Selection Type is <i>AreaXAxis</i> . |
| Selection Type | Specifies how users interact with the chart: Options include: <i>Area</i> lets the user select points by drawing a rectangular region within X- and Y-axis charts. This and the following types are <i>not</i> supported for Funnel, Pyramid, and Pie series. <i>AreaXAxis</i> is the same as <i>Area</i> , except the Y-axis region fills the chart. Only the X-axis is selected by the user. The previous chart example uses this selection type to produce a selection area that can only be dragged left and right. <i>AreaYAxis</i> is the same as <i>Area</i> , except the X-axis region fills the chart. Only the Y-axis is selected by the user. Produces a selection area that can only be dragged up and down. |

In the target report or process task, all of the selected values can be referenced using the tokens named in the Min-Max X-Y Axis attributes. For example, if the attributes are specified as:

Max X-axis ID = *myMaxX*

Max Y-axis ID = *myMaxY*

Min X-axis ID = *myMinX*

Min Y-axis ID = *myMinY*

Selection Type = *Area*

and a selection is made, the resulting values can be found in the next report or process task using these tokens:

@Request.myMaxX~

@Request.myMaxY~

@Request.myMinX~

@Request.myMinY~

The **Validation.Required** element is also available as a child element so a warning message can be displayed if the page is submitted without a chart area being selected.

 The Validation element's *Class* attribute will not be applied.

Input Selection Range Event

This element specifies, through its child elements, the action to be taken when an area selection is completed.

| Attribute | Description |
|-----------------|--|
| ID | The unique element ID. |
| Selection Event | Specifies the type of event that will trigger the action. Options include: <i>Change</i> (the default) triggers the action when the status of selection area changes. <i>AreaDrawn</i> triggers the action when the drawing of the selection area ends. <i>AreaCleared</i> triggers the action when the selection area is cleared (by clicking outside of it). |

Input Selection Style

This element allows you to style the selection point or area.

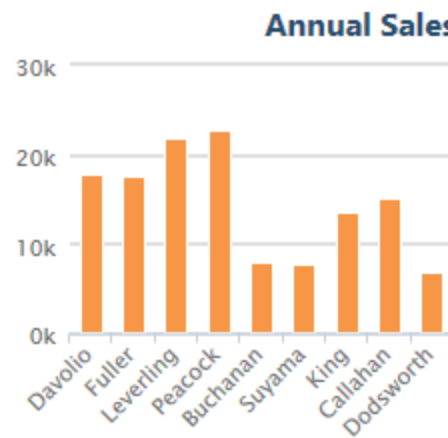
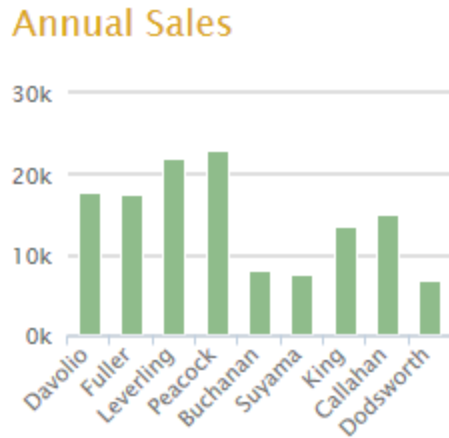
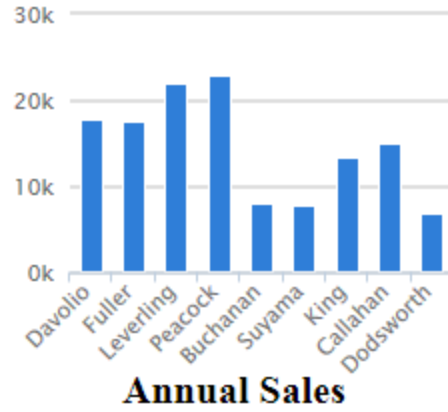
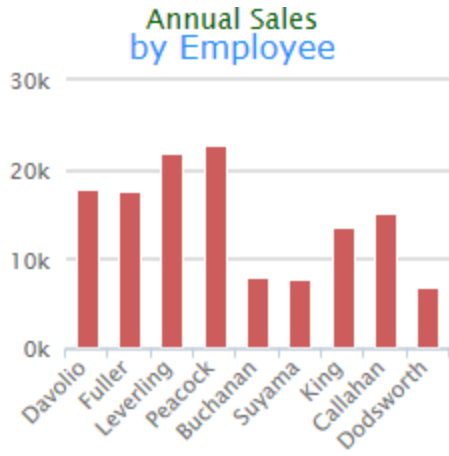
| Attribute | Description |
|-----------------------------------|---|
| Border Color | Not used with Input Selection Range. |
| Border Color Transparency | Not used with Input Selection Range. |
| Border Thickness | Not used with Input Selection Range. |
| Color | Not used with Input Selection Range. |
| Selection Area Color | When using an "Area" Selection Type, specifies the color of the selection rectangle background. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. |
| Selection Area Color Transparency | When using an "Area" Selection Type, specifies the transparency of the selection rectangle. The lowest value of 0 specifies that the line is opaque, with no transparency. At the other end of the scale, 15 specifies a completely transparent line. Use medium-level transparency to allow different chart layers to show through each other. |
| Transparency | Not used with Input Selection Range. |

When using Input Selection Range, you may want to turn off the automatic Quicktips generated by the chart. This is done by setting the Chart Canvas element's **Auto Quicktip** attribute to *False*.

Caption Style and SubCaption Style

The Chart Canvas element's `Caption` and `SubCaption` attributes let you specify text that appears at the top (or elsewhere) on the canvas. Its **`Caption Style`** and **`SubCaption Style`** child elements can be used to format and position this text.

The examples below show some variations in the format and positioning of `Caption` and `SubCaption` text:



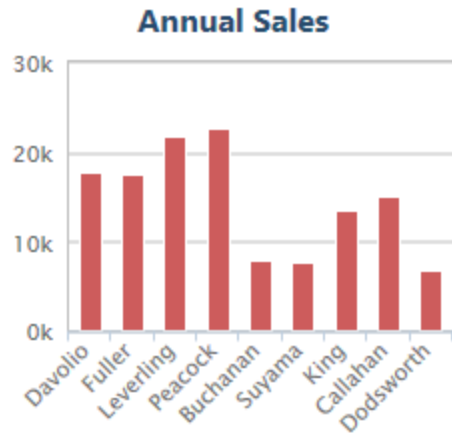
Attributes

Both elements have the following attributes, except as noted:

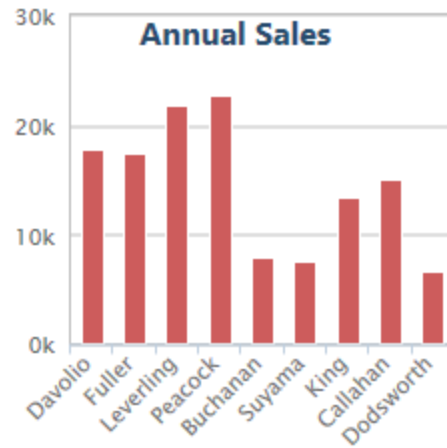
| Attribute | Description |
|----------------------|---|
| Alignment Horizontal | Sets the horizontal alignment of the text as <i>Left</i> , <i>Center</i> , or <i>Right</i> , within the canvas. The default value is <i>Center</i> . |
| Alignment Vertical | Sets the vertical alignment of the text as <i>Top</i> , <i>Middle</i> , or <i>Bottom</i> , within the canvas. The default value is <i>Top</i> . |
| Caption Spacing | Sets the space between the caption and its subject, in pixels. |
| Font Color | Sets the caption font color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. |
| Font Family | Specifies the names of one or more fonts. When multiple fonts are specified, with commas between each, the browser uses the first recognized font. |
| Font Italic | Specifies whether the font is <i>Italic</i> . |
| Font Size | Sets the caption font size, in pixels. |
| Font Weight | Specifies the weight (thickness) of characters in the caption text. Options include <i>Lighter</i> , <i>Normal</i> , <i>Bold</i> , and <i>Bolder</i> . The default value is <i>Normal</i> . |
| Format | Specifies a format for the text displayed. For more information, see <i>Format Data</i> . |
| Inside Plot Area | Specifies whether space in the canvas will be reserved for the caption (<i>False</i>) or whether it will overlap other content (<i>True</i>). The default value is <i>False</i> . |

| Attribute | Description |
|-----------|---|
| Offset X | Sets the horizontal position offset of the caption relative to the horizontal alignment, in pixels. |
| Offset Y | Sets the vertical position offset of the caption relative to the vertical alignment, in pixels. |

The **Inside Plot Area** attribute causes the Caption to be displayed *inside* the plot area, instead of between it and the edge of the canvas:



Inside Plot Area = False

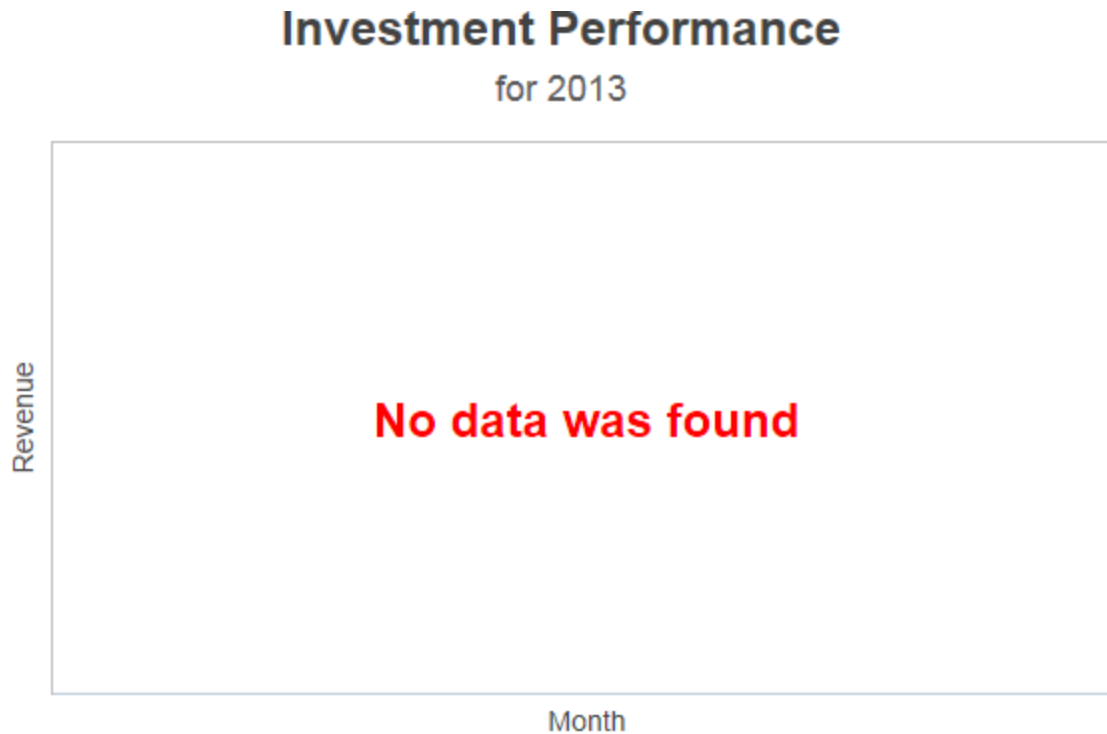


Inside Plot Area = True

As shown in the example above, when Inside Plot Area is set to *True*, the plot area is scaled to create space for the Caption text.

No Data Caption Style

The Chart Canvas element's No Data Caption attribute lets you specify text that appears in the middle of the canvas if no data is retrieved for any series. Its **No Data Caption Style** child element can be used to format this text. For example:



The example above shows one way the text of the No Data Caption can be formatted.

Attributes

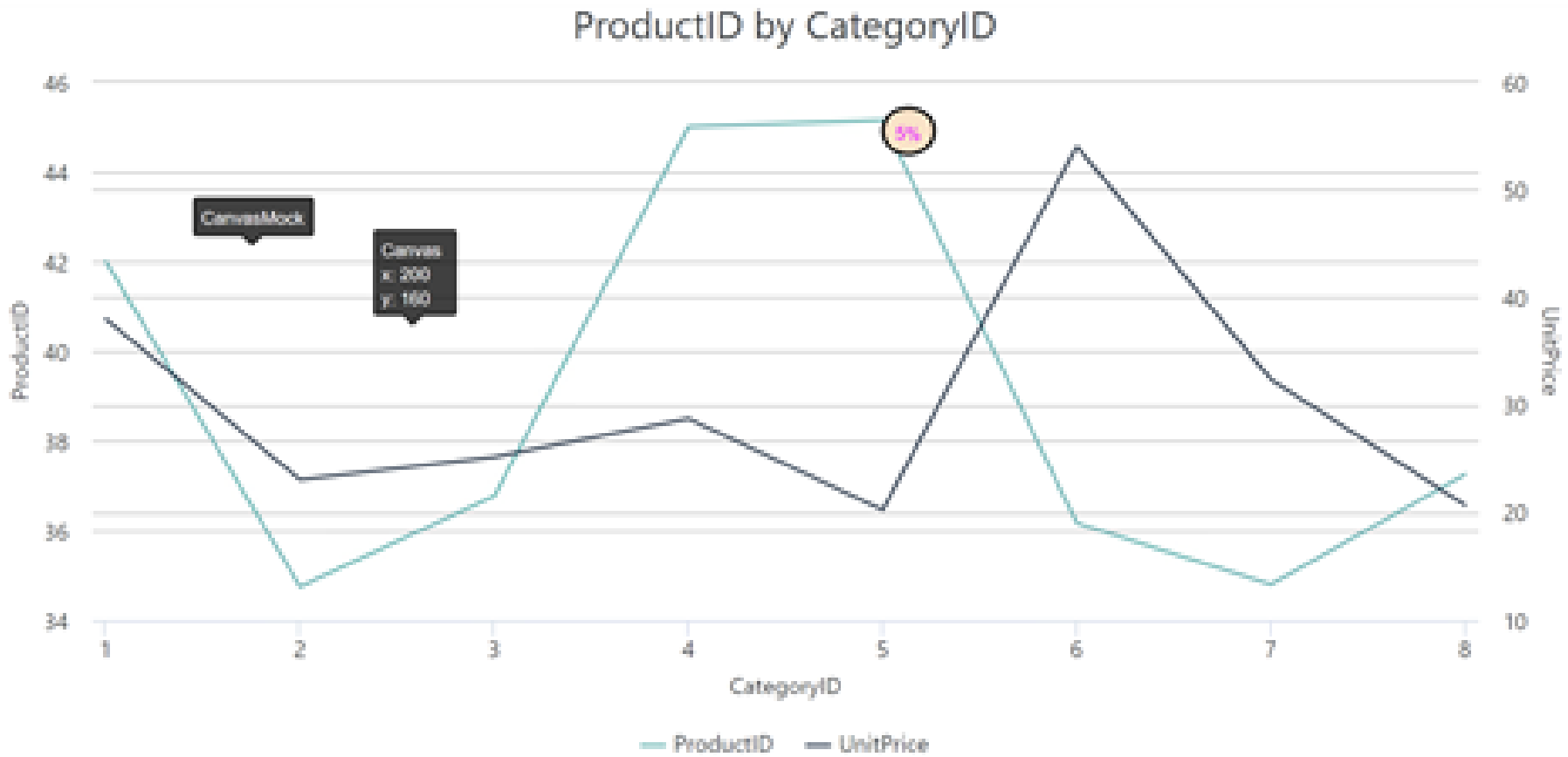
The No Data Caption Style element has the following attributes:

| Attribute | Description |
|-------------|---|
| Font Color | Sets the caption font color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. |
| Font Family | Specifies the names of one or more fonts. When multiple fonts are specified, with commas between each, the browser uses the first recognized font. |
| Font Size | Sets the caption font size, in pixels. |
| Font Weight | Specifies the weight (thickness) of characters in the caption text. Options include <i>Lighter</i> , <i>Normal</i> , <i>Bold</i> , and <i>Bolder</i> . The default value is <i>Normal</i> . |

v23.1

Chart Annotation

The Chart Canvas element's **Chart Annotation** child element allows you to Annotates the chart with labels and shapes placed at various points of interest.



The Chart Annotation element has the following attributes:

| Attribute | Description |
|---------------|---|
| ID | Unique value within the definition. |
| Popup Movable | Customize along which axis the annotation can be draggable. Valid values include: X, Y, XY, and Disable. The default value is XY. |
| Visible | Controls visibility. Values include True/False. The default value is <i>empty (False)</i> . |

AnnotationLabel.Mock

The Chart Annotation element works in conjunction with the AnnotationLabel.Mock child element. This element creates an annotation label and position on the chart by linking it to a created mock point.

The AnnotationLabel.Mock element has the following attributes:

| Attribute | Description |
|----------------------|--|
| Alignment Horizontal | Sets the horizontal alignment of the text as <i>Left, Center, or Right</i> , within the canvas. The default value is <i>Center</i> . |
| Alignment Vertical | Sets the vertical alignment of the text as <i>Top, Middle, or Bottom</i> , within the canvas. The default value is <i>Top</i> . |
| Allow Overlap | Controls whether the labels can overlap. The Default value is <i>False</i> . |
| Background | Sets the background color of the element. Enter a color by name, decimal RGB value, or hex RGB value. Pre- |

| Attribute | Description |
|---------------|--|
| Color | fix hex values with the pound sign, e.g. #112233. |
| Border Color | Sets the color of the border lines. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. |
| Border Radius | Sets the amount of rounding for canvas, legend, and data label corners, in pixels. |
| Border Width | For highlighting, controls width of border around data points in Line, Area, and Scatter charts. |
| Caption | Identifies the text to be displayed. |
| Condition | An expression that evaluates to a value of True or False. Expressions should be in JavaScript (recommended) or VBScript syntax. For .NET versions, the scripting language can be set in the <code>_settings.lgx</code> General element. For Java versions, the scripting language is always JavaScript. Typically, you would compare values using a token, such as <code>"@Data.value~ < 0"</code> or <code>"@Session.ShowCol1~ == 'true'"</code> . Elements are removed when the Condition evaluates to False. Use quotes when working with strings: <code>"@Data.myColumn~" == "SomeValue"</code> If your data token may have a double quote inside of it, you should use single quotes: <code>'@Data.myColumn~' == 'SomeValue'</code> String valued data tokens with single and double quotes inside are not supported by the condition attribute. |
| Crop | Specifies whether to crop/hide the label that is outside the plot area. The default value is <i>False</i> . |
| Font Color | Sets the caption font color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. |

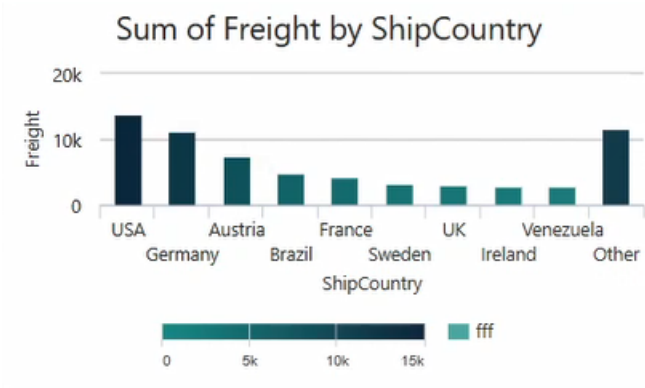
| Attribute | Description |
|-------------|--|
| Font Family | Specifies the names of one or more fonts. When multiple fonts are specified, with commas between each, the browser uses the first recognized font. |
| Font Italic | Specifies whether the font is <i>Italic</i> . |
| Font Size | Sets the caption font size, in pixels. |
| Font Weight | Specifies the weight (thickness) of characters in the caption text. Options include: <i>Lighter</i> , <i>Normal</i> , <i>Bold</i> , and <i>Bolder</i> . The default value is <i>Normal</i> . |
| Format | Specifies a format for the text displayed. For more information, see <i>Format Data</i> . |
| Offset X | Sets the horizontal position offset of the caption relative to the horizontal alignment, in pixels. |
| Offset Y | Sets the vertical position offset of the caption relative to the vertical alignment, in pixels. |
| Overflow | Handles how the label renders when it goes outside the plot area. The default value is <i>Justify</i> , which aligns the label inside the plot area. This attribute can also be set to <i>None</i> . |
| Padding | Sets the label padding. The default value is 5. |
| Shape | Specifies the shape of the label. Options include: Circle, Rect, Diamond, and Triangle. The default value is <i>callout</i> . |
| X | Specifies the X position of the point. Units can be in either axis or chart pixel coordinates. |

| Attribute | Description |
|-----------|---|
| X Axis | <p>Defines which X Axis the point is connected to. It refers to either the axis ID, or the index of the axis in the X Axis array. If the option is not configured, or the axis is not found, the x coordinate refers to the chart pixels. If a numeric value is used, it is considered to be the axis ID first.</p> |
| Y | <p>Specifies the Y position of the point. Units can be in either axis or chart pixel coordinates.</p> |
| Y Axis | <p>Defines which Y Axis the point is connected to. It refers to either the axis ID, or the index of the axis in the Y Axis array. If the option is not configured, or the axis is not found, the y coordinate refers to the chart pixels. If a numeric value is used, it is considered to be the axis ID first.</p> |

v23.1

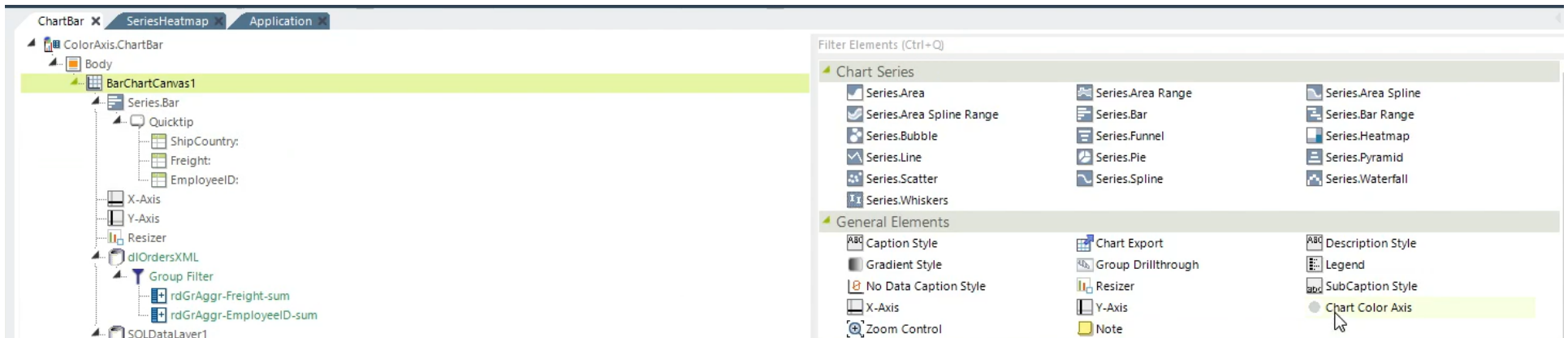
Chart Color Axis

The Chart Canvas element's **Chart Color Axis** child element allows you to add a color axis for series charts.



You have the option to display the axis as scalar or based on data classes. Scalar axes are represented by a gradient, ranging between Min Color and Max Color, as shown above. On the other hand, Chart Color axes based on data classes are divided into separate classes based on their values.


To utilize this feature, add the **Chart Color Axis** child element to your ChartCanvas.




 Series.Whiskers is the only series chart that does not support this feature.

The Chart Color Axis element has the following attributes:

| Attribute | Description |
|-----------|---|
| ID | Sets a unique element ID for the chart axis. This attribute is optional, but beneficial when using numerous color axes for a series (see below). |
| Max Color | <p>Sets the max color for the color axis. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, choose a color via the color picker by selecting the "... " option next to this field.</p> <p>If not set, the default color will be obtained according to the current theme.</p> |
| Max Value | Sets the maximum number the color axis will display. If this value is not set, the extreme value of the color axis will be calculated automatically. |
| Min Color | <p>Sets the min color for the color axis. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, choose a color via the color picker by selecting the "... " option next to this field.</p> <p>If not set, the default color will be obtained according to the current theme.</p> |
| Min Value | Sets the minimum number the color axis will display. If this value is not set, the extreme value of the color axis will be calculated automatically. |
| Reverse | Reverses the axis so low and high values are swapped. Chart is then rendered to match. For example, if Y-axis is |

| Attribute | Description |
|----------------|--|
| Axis | reversed, low values are at the <i>top</i> of the canvas and the chart is drawn "downward" from it. |
| Show In Legend | Specifies whether to display the Color Axis in the legend. The default value is <i>True</i> . |
| Tick Count | <p>Sets the number of ticks for your X and Y-axis. The default value is <i>Undefined</i>. Set this value to a number greater than 2.</p> <p> Tick Count can only be used with Linear axes; DateTime, Logarithmic, and Category axes are not affected.</p> |

 For Heatmaps, the attributes "Max Color" and "Min Color" will be ignored and replaced with "High Value Color" and "Low Value Color".

To support this feature, the Chart Canvas element also includes two new attributes, Linked to Color Axis ID and Color-Axis Data Column. The Linked to Color Axis ID attribute specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes. If you are using the Chart Color Axis element in a mult-series chart, by default, the series will link to the first color axis. To link the series to a special color axis, set the ID attribute of the Chart Color Axis element to the corresponding Series' Linked to Color Axis ID attribute, as demonstrated below:



The Color-Axis Data Column specifies the name of a datalayer column whose values will be plotted along the x-axis. See example configuration below:

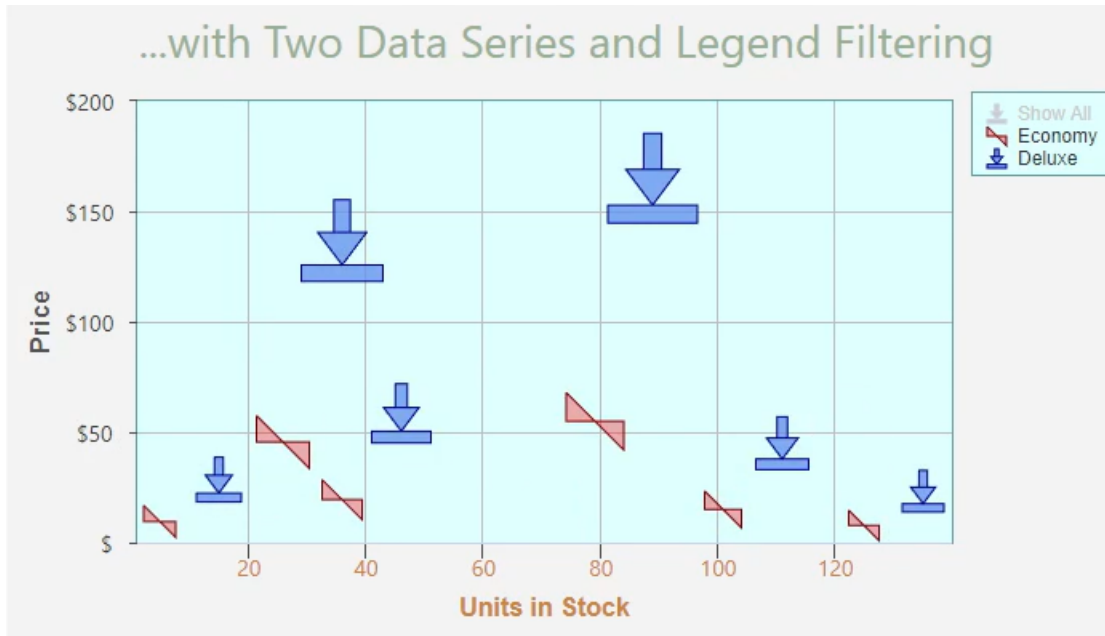
Multimedia content not available in this format



- For Heatmaps, the "Color-Axis Data Column" of the series is ignored and replaced with "Heat Map Color Data Column" of the Heatmap series.
- For other chart types, the series' Chart Color-Axis Data Column attribute must be the output column of the datalayer. If this value is not set, the column of "y" or "low" will be used.

Chart Custom Symbol

The Chart Canvas element's **Chart Custom Symbol** child element allows you to add a predefined shape or symbol for the chart marker.



A "marker point" is a symbol that appears on the chart at each data point. The Marker Points element enables customization of Marker Point appearance, with its Symbol attribute specifying the symbol that identifies each data point on the chart. With Chart Custom Symbol, you can create your own symbol and utilize it in the Symbol attribute of the Marker Points element.

The Chart Custom Symbol element has the following attributes:

| Attribute | Description |
|------------|--|
| SymbolKey | The key or ID of the custom symbol. This key links the custom symbol to the Marker Points element's Symbol attribute list. |
| SymbolPath | <p>A string representing the Scalable Vector Graphics (SVG) path, written as follows:</p> <ul style="list-style-type: none"> • Separate commands and parameters with commas. • Parameters can be numerical values or expression with variables x(x coordinator of plot area), y(y coordinator of plot area), w(graphic's width), h(graphics's height). <p>For example: M, x, y, L, x+w, y+h, V, y+h*0.5, H ,x, L ,x, y</p> |

To configure the Chart Custom Symbol element and link it to the Marker Points' Symbol attribute, complete the following:

1. Begin by adding a Chart Custom Symbol element to your chart.
2. Then, define a SymbolKey.
3. Next, enter a SymbolPath to pass the definition.
4. Access the Marker Points element and enter or select the SymbolKey (added in step 2) in the "Symbol" attribute.

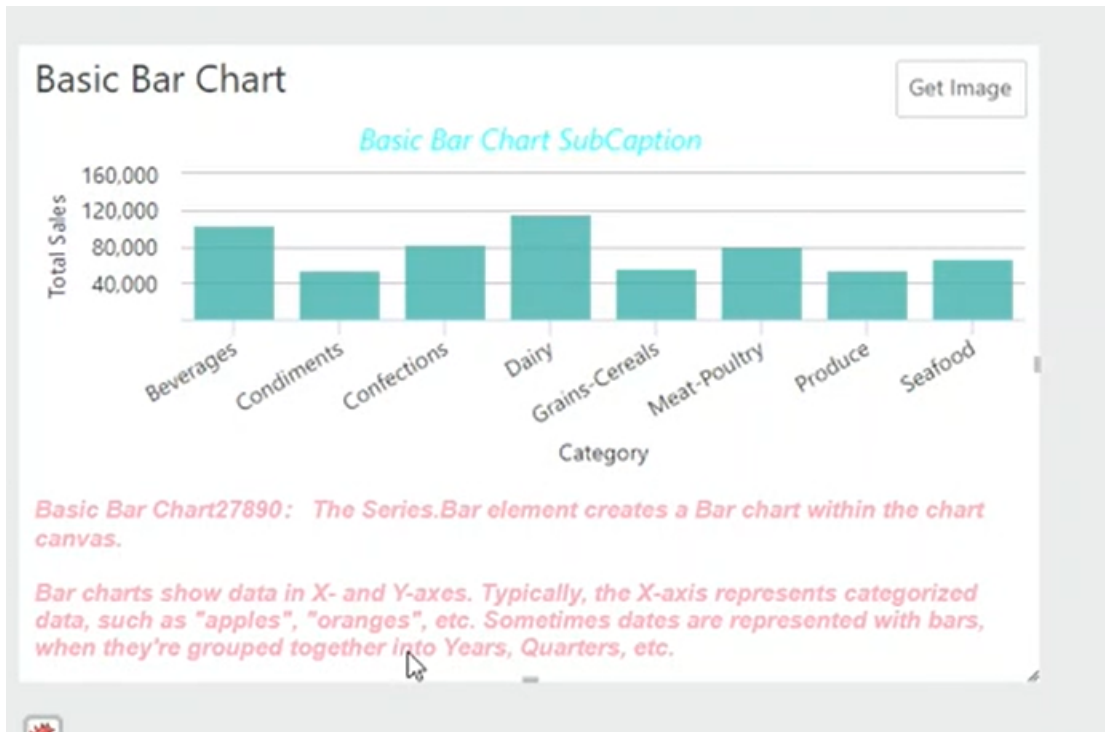
The video below demonstrates the steps for defining a symbol key and path to pass a definition, linking the custom symbol to the Marker Point element, as well as the finished product:

**Multimedia content not
available in this format**

The following series charts support this feature: Area, Bubble, Line, Scatter, Spline, and Area Spline.

Chart Description Style

The Chart Canvas element's Chart Description attribute lets you specify text that appears in the middle of the canvas if no data is retrieved for any series. Its **Chart Description Style** child element can be used to format this text. For example:



The example above shows one way the Chart Description text can be formatted.

Attributes

The Chart Description Style element has the following attributes:

| Attribute | Description |
|----------------------|---|
| Alignment Horizontal | Sets the horizontal alignment of the text as <i>Left</i> , <i>Center</i> , or <i>Right</i> , within the canvas. The default value is <i>Center</i> . |
| Alignment Vertical | Sets the vertical alignment of the text as <i>Top</i> , <i>Middle</i> , or <i>Bottom</i> , within the canvas. The default value is <i>Top</i> . |
| Font Color | Sets the caption font color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. |
| Font Family | Specifies the names of one or more fonts. When multiple fonts are specified, with commas between each, the browser uses the first recognized font. |
| Font Size | Sets the caption font size, in pixels. |
| Font Weight | Specifies the weight (thickness) of characters in the caption text. Options include <i>Lighter</i> , <i>Normal</i> , <i>Bold</i> , and <i>Bolder</i> . The default value is <i>Normal</i> . |
| Format | Specifies a format for the text displayed. For more information, see <i>Format Data</i> . |
| Inside Plot Area | Specifies whether space in the canvas will be reserved for the caption (<i>False</i>) or whether it will overlap other content (<i>True</i>). The default value is <i>False</i> . |
| Offset X | Sets the horizontal position offset of the caption relative to the horizontal alignment, in pixels. |
| Offset Y | Sets the vertical position offset of the caption relative to the vertical alignment, in pixels. |

Trend Line

The Chart Canvas element's Series child elements cause a data visualization (the chart) to be rendered in the canvas. The **Trend Line** element is available as a child of Series elements and generates a line on the chart that indicates the "trend" of the data. The line connects a number of data points generated using a regression algorithm.

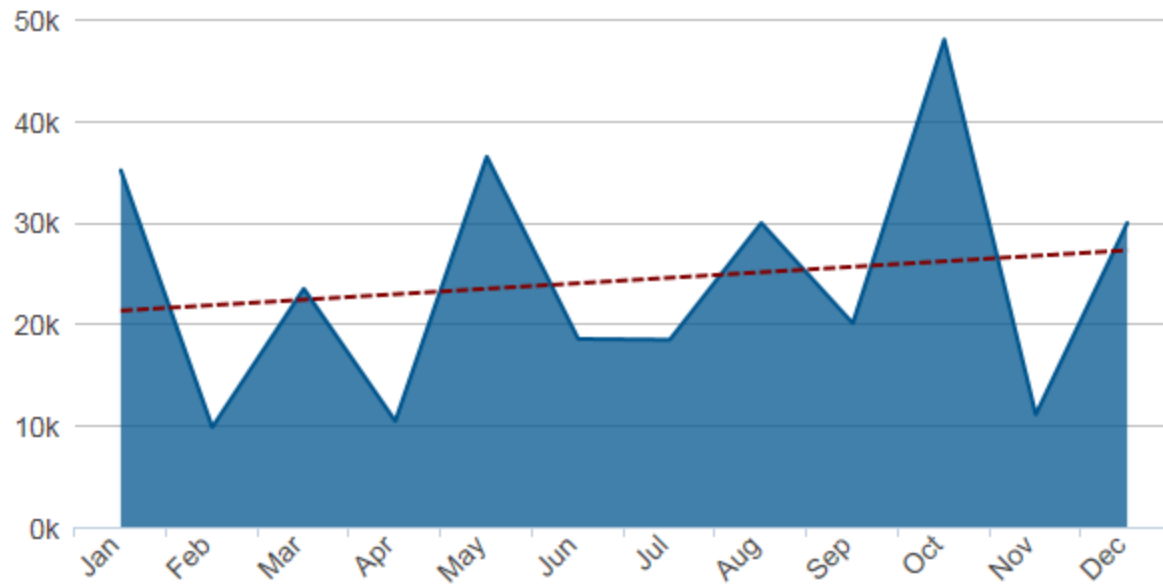
The following topics discuss the Trend Line child element:

- [Trend Line Attributes](#)
- [Including in Legend Items](#)

About the Trend Line

Trend lines provide a visual clue to data trends and are very useful when the type of Series being used doesn't make that readily apparent. The **Trend Line** element is available for use with Area, Area Spline, Bar, Line, Scatter, and Spline series.

Monthly Sales 2013



In the example above, the red dotted line is the Trend Line, indicating that the data trend is upward. You can see how that might be difficult to discern from the Area chart itself. The Trend Line can be any color, can be solid, dotted, or dashed, and you have a choice of regression algorithms.

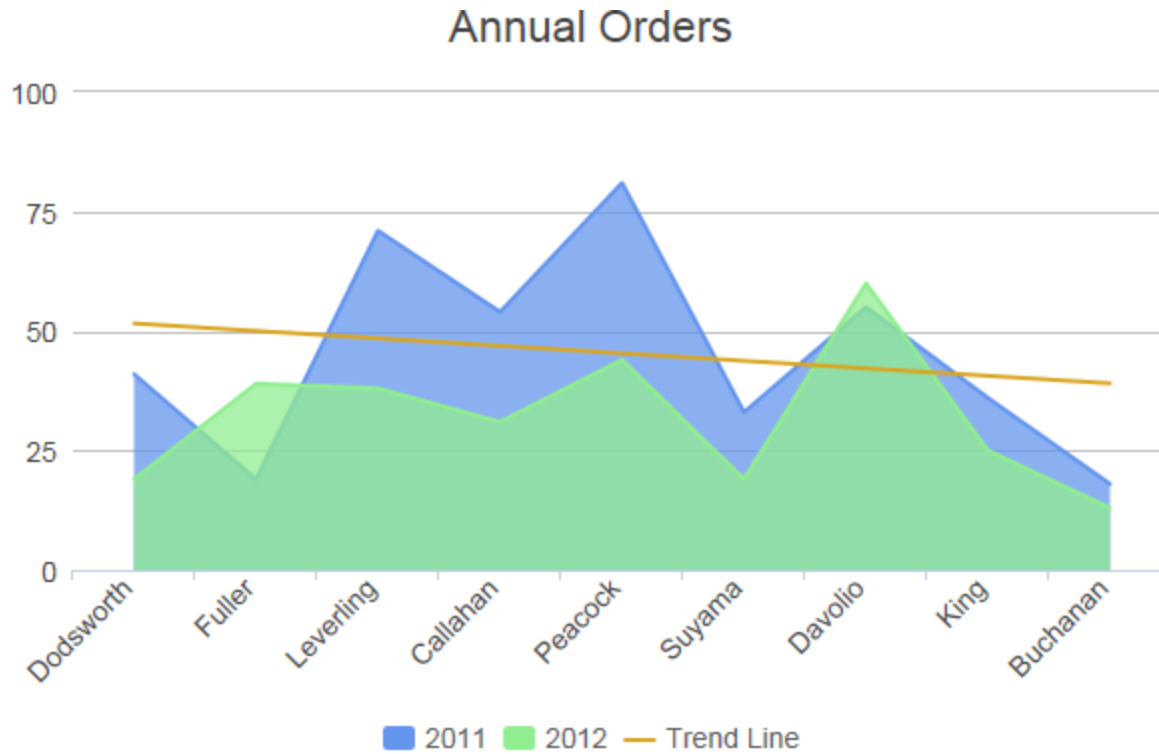
Trend Line Attributes

The Trend Line element has the following attributes:

| Attribute | Description |
|----------------|---|
| Color | Sets the line color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| ID | The unique element ID. |
| Legend Label | Specifies the text that will be shown inside the chart legend for the trend line. A value <i>must</i> be provided here for an entry to appear in the legend. |
| Line Algorithm | Specifies the regression algorithm that will be used to generate the data points plotted by the trend line. Options include <i>LinearRegression</i> (the default) and the curve-fitting <i>LOWESS</i> algorithm. |
| Line Style | Specifies the pattern of the trend line as either <i>Solid</i> (the default) or a combination of dashes and/or dots. |
| Line Thickness | Specifies the thickness of the line, in pixels. The default value is 1 pixel. |
| Transparency | Specifies the transparency of the line color. The lowest value of 0 specifies that the line is opaque, with no transparency. At the other end of the scale, 15 specifies a completely transparent line. |

Including Legend Items

If a legend is used with the chart, Trend Lines can be included in it:



As shown above, the Trend Line appears as an item in the legend. The Trend Line element's **Legend Label** attribute specifies the text that will appear in the legend and must have a value in order for the Trend Line to be included in the legend.

Zoom Control

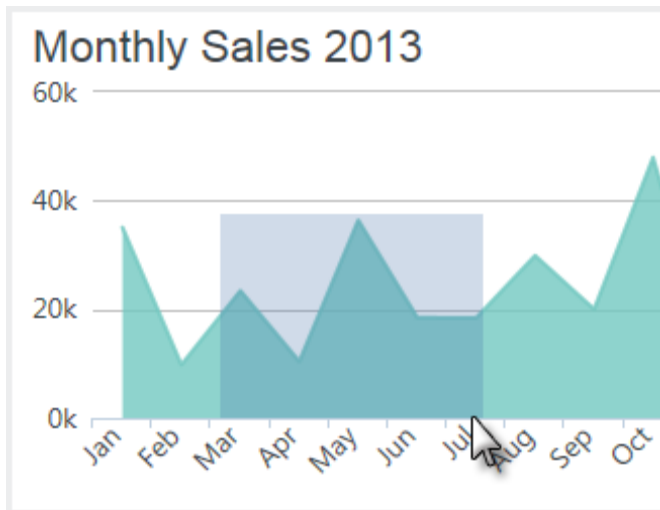
The Chart Canvas element's Series child elements cause a data visualization (the chart) to be rendered in the canvas. The **Zoom Control** element is available as a child of Chart Canvas and allows the user to "zoom" into the canvas, magnifying a portion of it.

This topic contains the following sections:

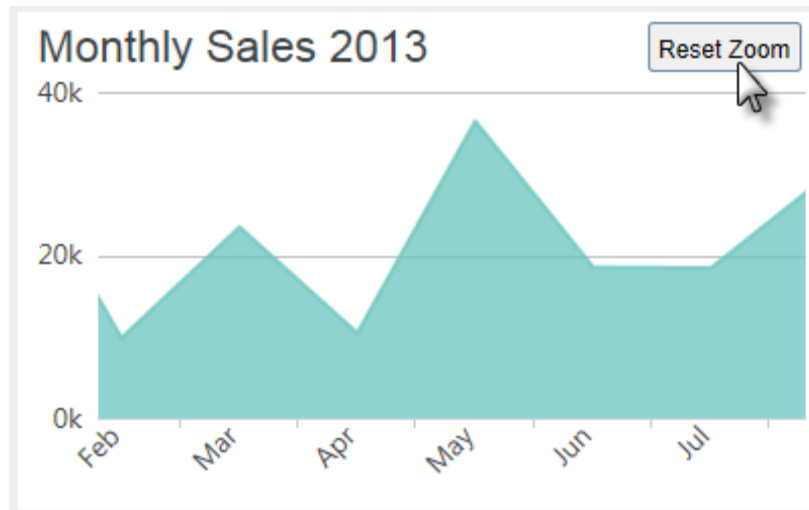
- About the Zoom Control
- [Attributes](#)

About the Zoom Control

The **Zoom Control** element imparts a magnification feature to Chart Canvas.



Dragging selection area...



...zooms canvas in on selected values

As shown above, at runtime users can use the mouse to draw a selection area on the canvas and the canvas will be "zoomed in" on the selected data. A "Reset Zoom" button is also displayed, which will return the canvas to its original magnification.

The Reset Zoom button only appears when the mouse cursor is hovered over the upper right-hand corner of the chart.

The appearance of the Reset Zoom button can be styled using the **Reset Zoom Button** and **Reset Zoom Button Hover Style** child elements.

 The Zoom Control is *not* available for Series.Funnel, Series.Pie, or Series.Pyramid.

Attributes

The Zoom Control element has the following attributes:

| Attribute | Description |
|-----------------------------------|--|
| Selection Area Color | <p>Specifies the color of the selection area on the chart.</p> <p>Use the token @Chart to use a gradient fill to represent the selected data.</p> |
| Selection Area Color Transparency | <p>Specifies the transparency of the Selection Area Color. The lowest value of 0 specifies that the point fill is opaque, with no transparency. At the other end of the scale, a value of 15 specifies a completely transparent fill.</p> |
| Zoom Control Type | <p>Specifies the nature of the selection area that can be drawn. Options include: <i>Area</i> (the default) - allows dragging the mouse in any direction to draw the selection area. <i>AreaXAxis</i> - dragging can only be done along the X-axis. <i>AreaYAxis</i> - dragging can only be done along the Y-axis.</p> |

Group Drillthrough

The Chart Canvas element's **Group Drillthrough** child element adds grouped data point links which, when clicked, display a basic drill-through report showing the detail data used to create the group.

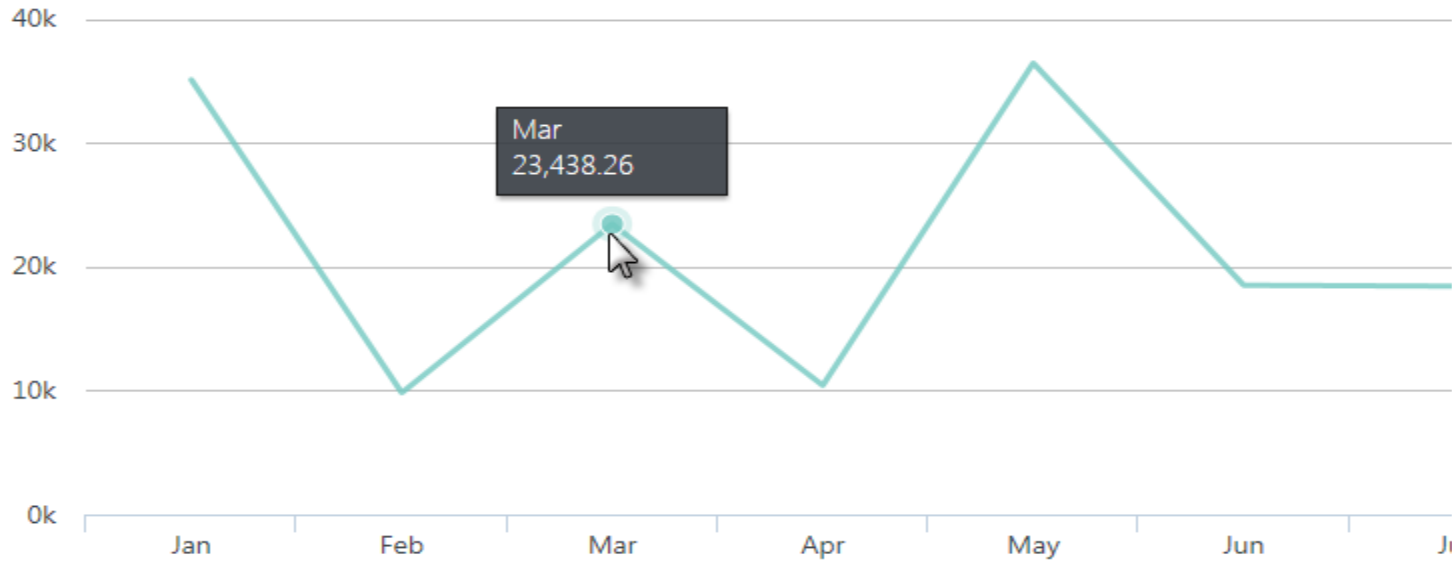
The following topics discuss the Group Drillthrough child element:

- [Group Drillthrough Attributes](#)
- [Customizing Detail Report Columns](#)

About Group Drillthrough

Data is often grouped, especially when it's related to time, to show an aggregated view of it. "Sales by Month" is an example where the data for each month is aggregated by month. This feature is often useful to be able to quickly examine the data in the aggregation. The **Group Drillthrough** element can be used as a child of the **Chart Canvas** element, or as the child of individual **Series** elements.

In the example below, a line chart visualizes data that has been grouped by month. When the March data point is clicked, the detail report displays, containing all of the grouped data in a Data Table:



[Back](#)

Sales Detail Report

[Click to export](#)

Drillthrough on: MonthNo="3"

| EmployeeID | OrderID | CustomerID | OrderDate | RequiredDate | ShippedDate | ShipVia | Freight | ShipName |
|------------|---------|------------|-----------------------|------------------------|-----------------------|---------|---------|---------------------------|
| 5 | 10248 | VINET | 7/11/2010 12:00:00 AM | 8/1/2010 12:00:00 AM | 7/16/2010 12:00:00 AM | 3 | 32.38 | Vins et alcools Chevalier |
| 5 | 10254 | CHOPS | 7/11/2010 12:00:00 AM | 8/8/2010 12:00:00 AM | 7/23/2010 12:00:00 AM | 2 | 22.98 | Chop-suey Chinese |
| 5 | 10269 | WHITC | 7/31/2010 12:00:00 AM | 8/14/2010 12:00:00 AM | 8/9/2010 12:00:00 AM | 1 | 4.56 | White Clover Markets |
| 5 | 10297 | BLONP | 9/4/2010 12:00:00 AM | 10/16/2010 12:00:00 AM | 9/10/2010 12:00:00 AM | 2 | 5.74 | Blondel père et fils |

Group Drillthrough can also be used in conjunction with the Drill To feature. To utilize this feature, ensure that your Chart Canvas or Series element includes a **Chart Drill To** child element. Once the Drill To element is enabled, selecting a data point displays a list of columns. Selecting a column re-draws the chart so that only the data representing that specific filter is shown. From there, you can use Group Drillthrough to drill further into the data, like below:

**Multimedia content not
available in this format**

Group Drillthrough Attributes

The Group Drillthrough element has the following attributes:

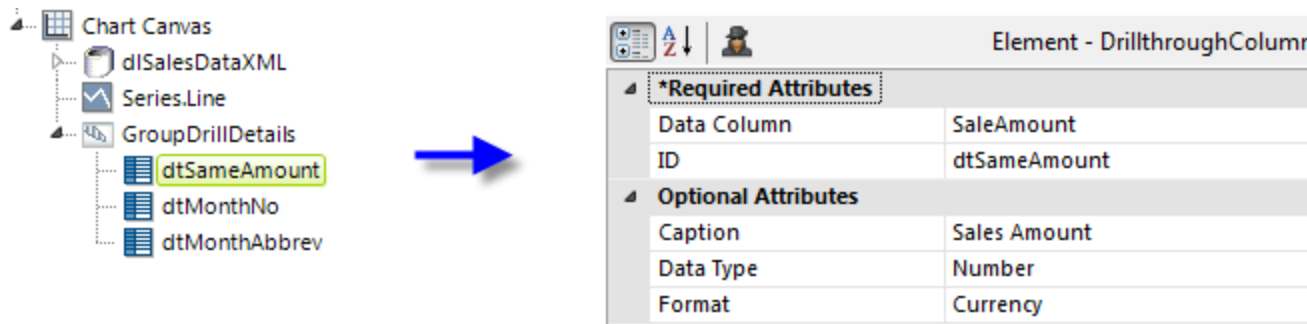
| Attribute | Description |
|-------------------|---|
| ID | (Required) Specifies a unique element ID. |
| Caption | Specifies the text of a caption that will appear at the top of the detail report. |
| Export | Specifies which exports will be available in the detail report, in a comma-separated list. Options include <i>CSV</i> , <i>Excel</i> , and <i>PDF</i> . The default value is <i>no exports</i> , which hides the Export icon. |
| Frame ID | Specifies the target window for the detail report. Leave blank for the current browser window, or enter <i>NewWindow</i> to open a new browser window. You can also specify an existing <i>FrameID</i> to re-use the same window for each request. |
| Group Filter ID | Specifies the unique element ID of the Group Filter element used to group the data that will be shown in the detail report. Provides differentiation when multiple Group Drillthrough elements are being used beneath multiple Series elements. May be left blank if there's just one Group Filter in the report. |
| Image | This attribute is ignored when the element is used with Chart Canvas Charts. |
| Security Right ID | If entered, controls access to this element via Logi security. Supply the ID of a Right defined in the application's settings/security section. Only users that have a Role referenced in the Right will be able to see the element. (Be careful - when the Right is not defined in the settings, the element is visible.) Multiple Right IDs, separated by commas may be entered. In this case, the user will see the element if he has access to any one of the Rights. |

| Attribute | Description |
|------------------------|--|
| Show Modes | <p>Specifies a text string that controls whether elements will be displayed or hidden. Leave this blank for the element to always be displayed or set it to None to hide it (it can be displayed again later with an Action.Show Element element). Set it to your own string value to have it appear only when the report's (root element's) Show Modes includes that value. Show Modes can contain multiple, comma-delimited values.</p> |
| Template Modifier File | <p>Specifies the name of an optional Template Modifier File, used to programmatically modify the Report Author super-element UI or behavior at runtime. See "Customizing Detail Report Columns" on the next page for more information about template modifiers. You can provide a fully-qualified file path and name for the file, within the application root folder or, if a fully-qualified file path is not provided, the application expects the file to be in its <code>_SupportFiles</code> folder.</p> |

Customizing Detail Report Columns

When used by itself, the Group Drillthrough element will display all of the detail data in its table. However, you may want to limit the number of columns displayed. This can be done using **Drillthrough Column** elements.

Customize the table by adding one **Drillthrough Column** element for each column to be displayed in the report. You can also customize the column header text (Caption) and format the data.



The example below shows the detail report from "Group Drillthrough" on page 140, with the custom columns from the previous image applied:

Back

Sales Detail Report

Drillthrough on: MonthNo="3"

| Sales Amount | Month # | Month Name |
|--------------|---------|------------|
| \$3,849.66 | 3 | Mar |
| \$2,684.00 | 3 | Mar |
| \$3,891.00 | 3 | Mar |
| \$2,518.00 | 3 | Mar |
| \$10,495.60 | 3 | Mar |

Series.Area

The Chart Canvas element's Series child elements cause a data visualization (the chart) to be rendered in the canvas.

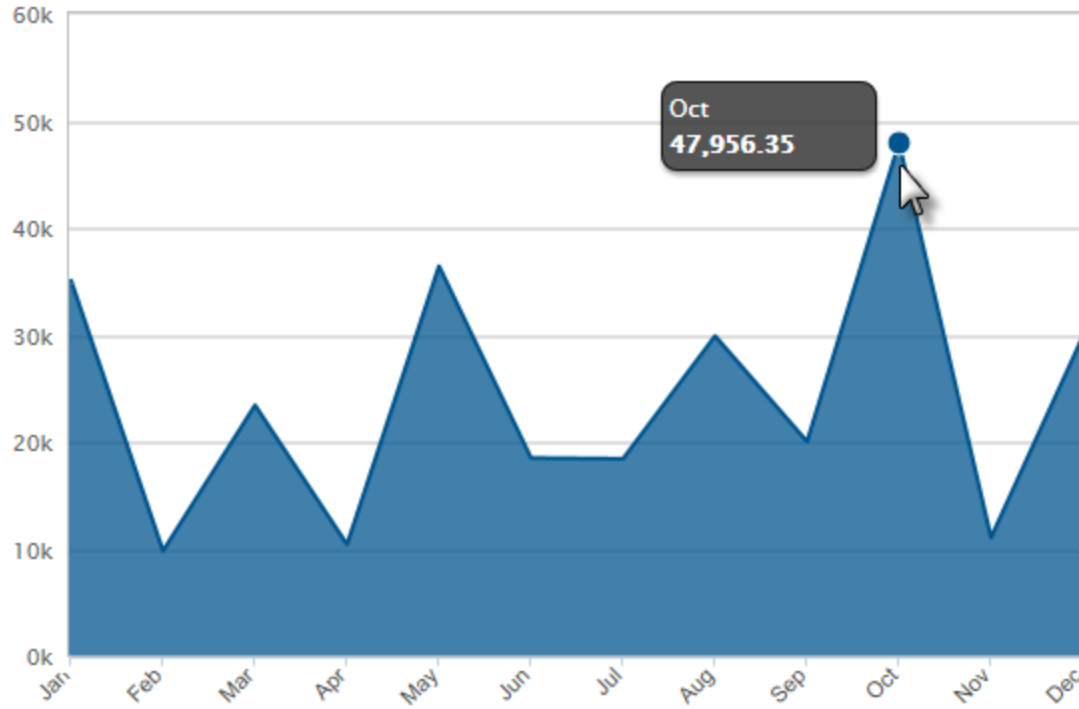
The following topics discuss the Series.Area child element:

- [Using Multiple Series](#)
- [Series.Area Attributes](#)
- [Using the Data Labels Element](#)
- [Using the Marker Points Element](#)
- [Using the Quicktips Element](#)
- [Using the Trend Line Element](#)
- [Using Action Elements](#)
- [Using the Series Annotation Element](#) v23.1
- [Using Input Selection](#)
- [Using the Refresh Series Timer](#)

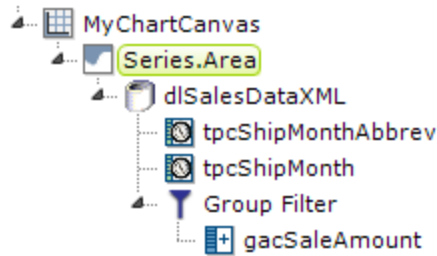
About Series.Area

The **Series.Area** element generates an Area chart, which is commonly used to represent aggregated totals, as numbers or percentages, over time.

Monthly Sales 2013



The example above shows a simple Area chart, representing sales per month for a year.

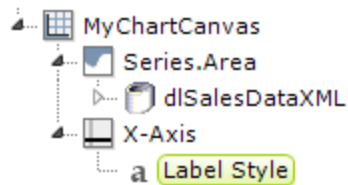


| Element - Series.Area | |
|-----------------------------|--------------------|
| *Required Attributes | |
| Y-axis Data Column | gacSaleAmount |
| Optional Attributes | |
| Color | |
| Combine With Series ID | |
| ... | |
| X-axis Data Column | tpcShipMonthAbbrev |
| X-axis Data Column Type | |

As shown above, the chart is created by adding Series.Area to the canvas, along with a datalayer and, typically, some child elements that may include **Time Period Column** elements, a **Group Filter**, and a **Group Aggregate Column** element. Very few attributes need to be set for the Series element in order to produce a basic chart.



A datalayer element can be used either beneath Series.Area, as shown above, or beneath Chart Canvas. If used as a child of Chart Canvas, its data is available to *all* child Series elements. This can improve performance if you have several series, all using the *same* data, beneath the same Chart Canvas element.

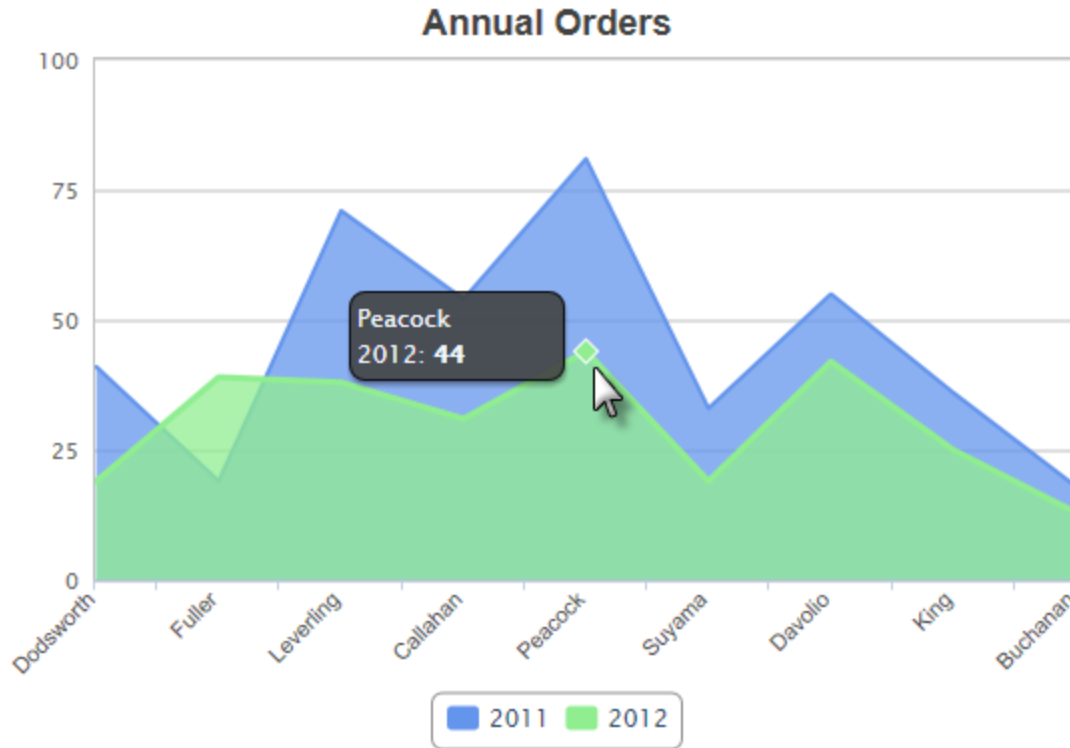


| Element - AxisLabelStyle | |
|----------------------------|----|
| Optional Attributes | |
| Font Angle | 45 |
| Font Color | |
| ... | |
| Stagger Labels | |

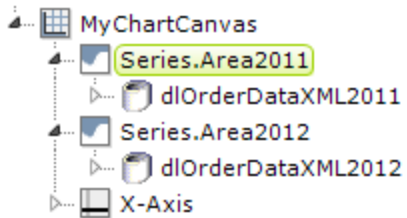
The properties of the X- and Y-axis, including captions, are set using the **X-Axis** and **Y-Axis** elements. For example, in order to angle the X-axis labels, add an X-Axis element beneath Chart Canvas (none of its attributes need to be set) and add its child **Label Style** element. Set the Label Style element's attribute as shown above.

Series.Area - Using Multiple Series

You can add additional series to the chart by adding additional Series elements:



The example above shows two Series, one for each year. A legend can also be added very easily.



| Element - Series.Area | |
|-----------------------------|-----------------|
| *Required Attributes | |
| Y-axis Data Column | gacOrderCount |
| Optional Attributes | |
| Color | CornflowerBlue |
| Combine With Series ID | |
| ... | |
| ID | Series.Area2011 |
| Legend Label | 2011 |
| Line Color | |
| ... | |
| X-axis Data Column | LastName |
| X-axis Data Column Type | |

The example above shows the two Series elements, their datalayers, and an X-Axis element used to produce the previous chart. You can adjust which series appears "in front" of the other in the chart by changing the order of the Series elements in the definition. Setting the Series elements' **Legend Label** attribute will automatically cause the legend to be displayed.

Charts with multiple series or aggregations are encouraged to use the ChartCanvas element's attribute, `Tooltip Split`. This attribute allows you to split a chart's tooltip into one label per series, rather than per segment. The default value for this attribute is *False*. Once enabled, when hovering over a data point, it displays all the values for the series. This method is recommended over shared tooltips for charts with multiple line series, and as such, takes precedence over `tooltip.shared`.



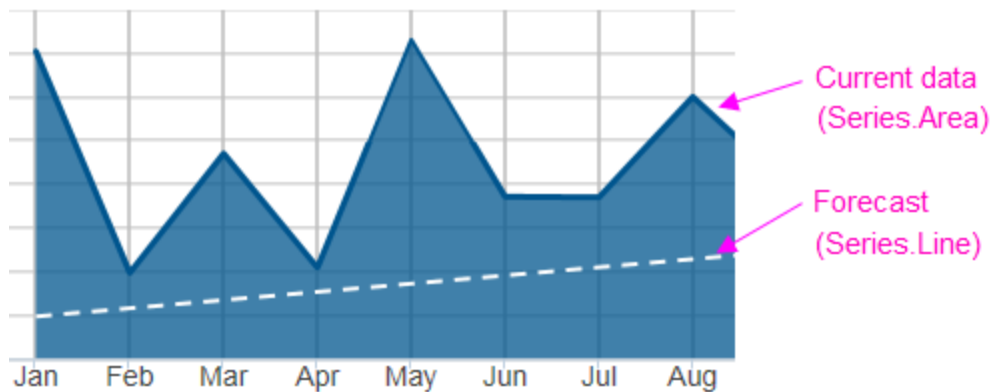
When using multiple series, you may be able to reduce the number of data queries and improve performance by using local data to read all of the data once, see *Datalayer Introduction*. Then, link its datalayer to share it to the series, see *Link Datalayers*. At each Series element, link its datalayer to the shared Local Data datalayer and filter the data to meet the needs of each individual series.

You can combine different types of Series elements, for example, Series.Area and Series.Line, to produce combinations of visualizations. A good example of this is when you wish to use forecasting.

v23.1 If you are using the Chart Color Axis element in a multi-series chart, by default, the series will link to the first color axis. To link the series to a special color axis, set the ID attribute of the Chart Color Axis element to the corresponding Series' Linked to Color Axis ID attribute. For more information, see "Chart Color Axis" on page 124.

Forecasting

Forecasting elements use a variety of techniques to produce projected values by analyzing existing values. The future values they "predict" are, in most cases, added as rows or columns to a datalayer so the data can be displayed along with the existing data. When using Chart Canvas charts, the forecast data is typically displayed using a Series.Line element, in conjunction with other series elements.



Forecasting elements add a "forecast value" column to the datalayer, and this column is used as the series' Y-axis data column. For more information about using forecasting elements, see *The Forecasting Elements*.

Series.Area - Attributes

The Series.Area element has the following attributes:

| Attribute | Description |
|--|---|
| Y-Axis Data Column | (Required) Specifies the name of a datalayer column whose values will be plotted along the Y-axis. |
| v23.1 Class Name | Specifies the Class Name referenced in the .css style sheet. The default value is <i>undefined</i> . |
| Color | Sets the data region fill color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| v23.1 Color Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the x-axis. |
| Combine With Series ID | Set this attribute to the element ID of another series to combine it with this series in the legend. When two series are combined, by default only the first one will appear in the legend but clicking the item in the legend toggles both series to appear and disappear. Or, the value <i>Previous</i> can be entered to combine this series with the previous series. |
| Connect Nulls | Specifies if data rows with null or blank values are to be ignored, allowing adjacent values to be connected in the chart. The default value is <i>False</i> . |

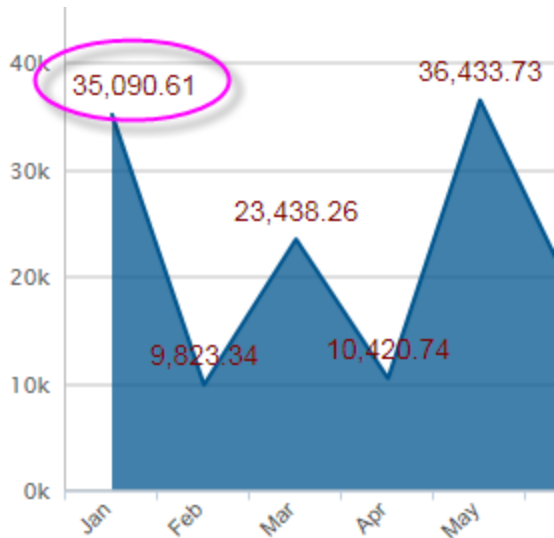
| Attribute | Description |
|---|--|
| Disable Mouse Tracking | Disables mouse tracking for the series, when set to <i>True</i> . This affects tooltips and click events on graphs and points. For large data sets, this may improve performance. The default value is <i>False</i> . |
| Hover Line Thickness | Sets the thickness of the line, in pixels, when the mouse pointer is hovered over it. The default value is <i>1</i> pixel. |
| Legend Label | Indicates text that will be shown for this series inside the chart legend. When a value is provided, automatically causes the legend to be displayed. |
| Line Color | Sets the data region's border line color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| Line Color Transparency | Specifies the transparency of the data region border line color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| Line Style | Specifies the pattern of the data region's border line as either <i>Solid</i> or a combination of dashes and dots. |
| Line Thickness | Specifies the thickness of the data region's border line, in pixels. The default value is <i>1</i> pixel. |
| v23.1 Linked to Color- | Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes |

| Attribute | Description |
|-----------------------------|---|
| Axis ID | |
| Linked to X-Axis ID | Specifies the ID of an X-Axis element that this series should be linked to when using multiple X-axes. |
| Linked to Y-Axis ID | Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes. |
| Negative Color | Sets the color for negative values. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #D5F484. The default value is <i>Red</i> . |
| Negative Color Transparency | Specifies the transparency of the Negative Color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| Negative Threshold | Sets the positive-negative value threshold, also called the "zero level" or "base level". The default value is <i>0</i> . |
| Stack Group Name | Enables the grouping of multiple series in a stacked chart. Enter the same arbitrary string value in this attribute for each series to be stacked. |
| Stacking | Sets the stacking mode. <i>Normal</i> stacks series on top of each other, while <i>Percent</i> stacks series, then raises the levels to 100 percent, showing the percentage for each series' values. The default value is <i>None</i> for no stacking. |

| Attribute | Description |
|-------------------------|---|
| Transparency | <p>Specifies the transparency of the data region fill color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other.</p> |
| X-Axis Data Column | <p>Specifies the name of a datalayer column whose values will be plotted along the X-axis.</p> |
| X-Axis Data Column Type | <p>Specifies the data type of the datalayer column named in the X-axis Data Column attribute. Options include <i>Auto</i> (the default), <i>Text</i>, <i>Number</i>, and <i>DateTime</i>.</p> <p>By default, X-axis data values that are <i>DateTime</i> type will be automatically distributed evenly across the time series. If you want to disable this behavior, set this attribute value to <i>Text</i>.</p> |

Series.Area - Using the Data Labels Elements

A "data label" is text shown next to each data point that shows its value. When the **Data Labels** element is used as a child of **Series.Area**, text representing the data values will appear on the chart:

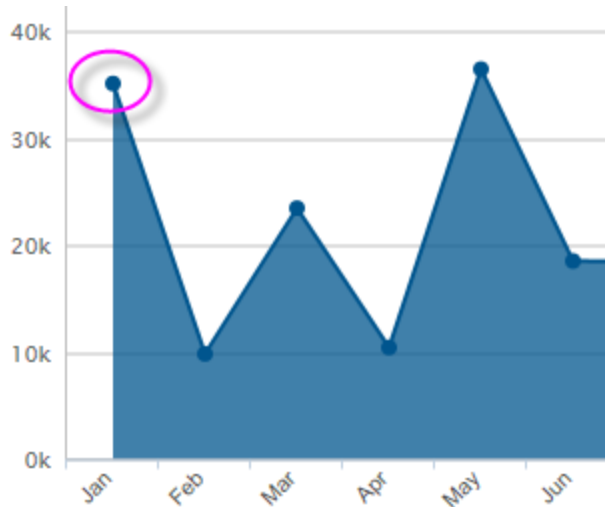


The Data Labels element has attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text. One attribute, **Series Name as Caption**, allows you to specify the y-axis label, x-axis label, or legend label as the data point value inside the chart.

The Data Labels element's color-related attribute values can be set using @Chart tokens. **v23.1** Or, utilize the "Class Name" attribute to apply unique styling to your data labels by referencing a class name from a stylesheet in the report.

Series.Area - Using the Marker Points Element

A "marker point" is a symbol that appears on the chart at each data point. When the **Marker Points** element is used as a child of Series.Area, a small dot matching the color of the area will be displayed at each data point:

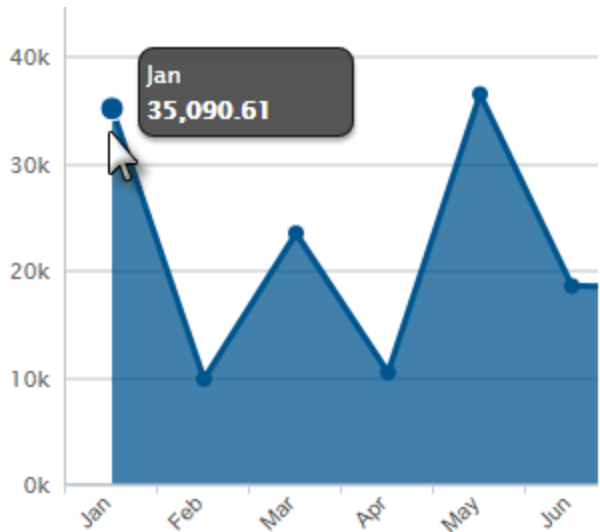


The default marker points are shown above. The **Marker Points** element allows you to select a different symbol for the marker point, and to control its size, color, border color, and transparency. When the cursor hovers over a marker point, it's automatically enlarged slightly.

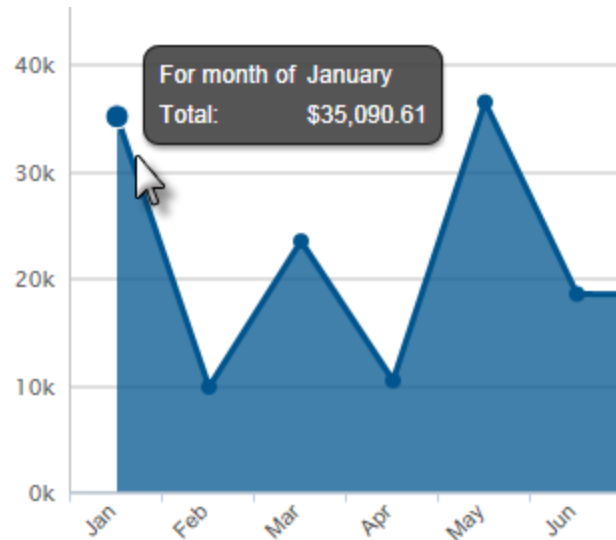
v23.1 You can also create your own marker point symbol using the Chart Canvas child element, Chart Custom Symbol. Once defined, link your custom symbol using the Marker Points element's Symbol attribute. For more information, see "Chart Custom Symbol" on page 128.

Series.Area - Using the Quicktips Element

By default, a "quicktip" is displayed when the mouse hovers near or over a data point:




Default quicktip



With Quicktip child element

The automatically-generated quicktip displays information for the X- and Y-axis, as shown above, left. However, you may want to display other information or format it differently, perhaps as shown above, right, which can be done by adding a **Quicktip** child element beneath Series.Area and setting its attributes and child elements. Use @Chart tokens to include chart data in the quicktip.

You can also configure the tooltip to display values for any available column in the datalayer. To do so, enter a token representing the data column in the Value attribute of the Quicktip Row element.

 To use this feature with DataLayer.ActiveSQL, please make sure the keep Grouped Rows attribute of the SqlGroup element is set to *False*.

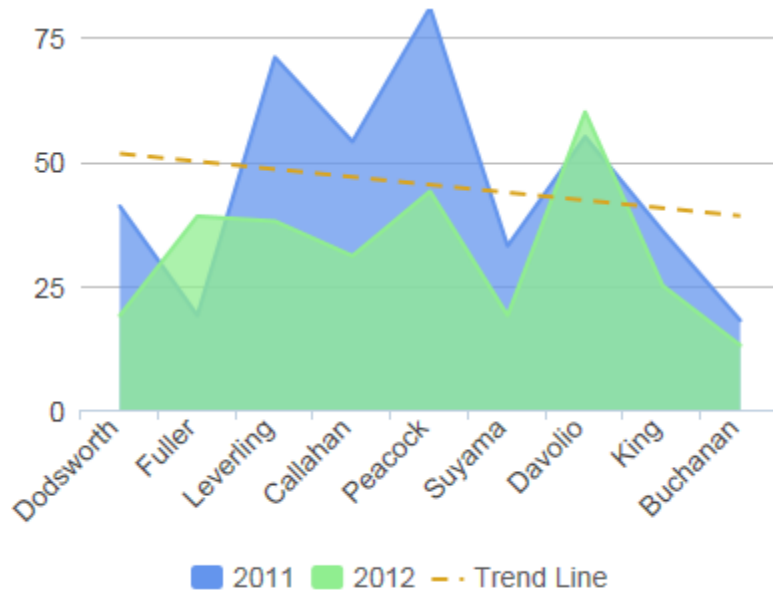
Intrinsic functions are supported in the Quicktip attributes.

You can deliberately add additional lines by adding child **Quicktip Row** elements beneath the Quicktip, one for each additional line of text you want to include.

The Quicktip Row element has been made context-sensitive with the addition of a **Condition** attribute.

Series.Area - Using the Trend Line Element

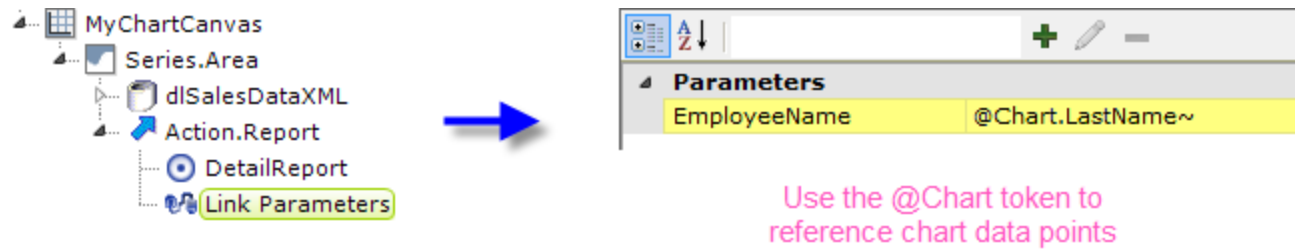
The **Trend Line** element creates a line on the chart that indicates the "trend" of the data. The line connects a number of data points generated using a regression algorithm.



The Trend Line element is a child of the Series.Area element and can be styled for color, line width, etc. When configured with a legend label, it will be represented by an item in the legend, as shown above.

Series.Area - Using Action Elements

Action elements initiate processing of a report or process task definition, redirection to a link, or other processing when a data point in the series is clicked.

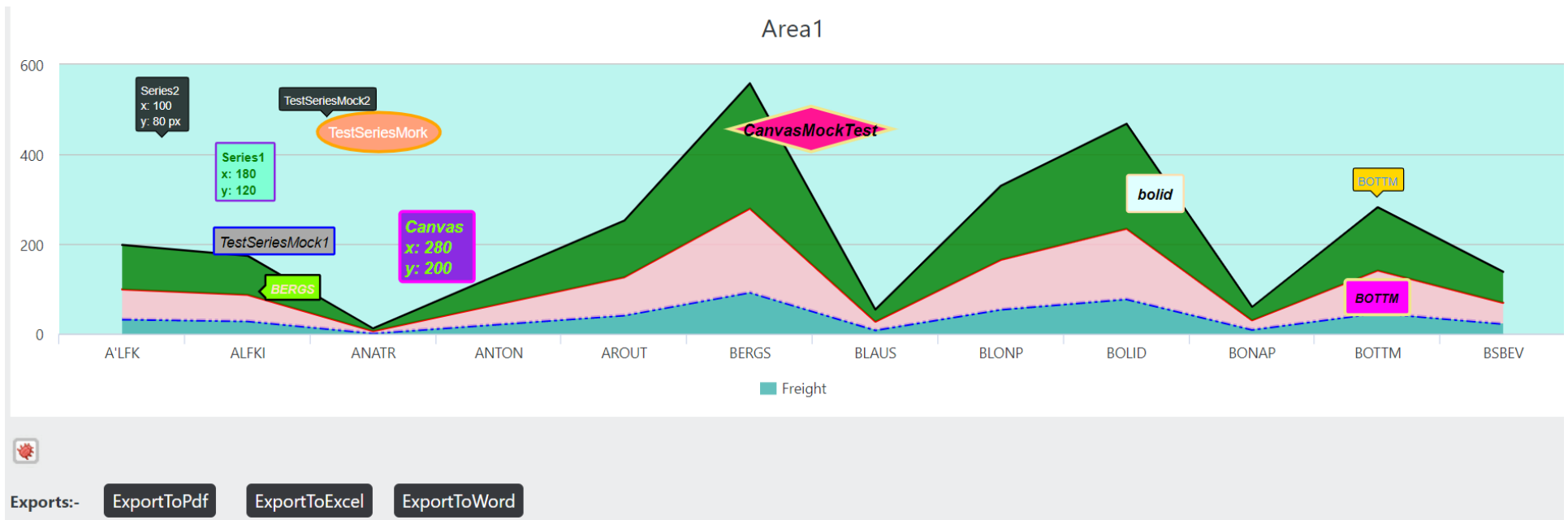


In the example above, an **Action.Report** element has been added as a child of the Series, along with its **Target.Report** and **Link Parameters** child elements. To reference chart data in parameters, use the @Chart token, as shown above.

A variety of Action elements are available for use with Series, including Action.Link, Action.Process, and Action.Refresh Element. Additional Action elements will be added in future releases.

Series.Area - Using the Series Annotation Element

Annotations allow you to annotate a chart freely using custom labels and shapes. When the **Series Annotation** element is used as a child of Series.Area, you can place these annotations at various points of interest:



The Series Annotation element offers two child elements: **AnnotationLabel.Point** and **AnnotationLabel.Mock**.

The AnnotationLabel.Point child element is used to create an annotation label and position on the chart by linking it to an existing point. With this element, you cannot specify the value, it always refers to the value of the point. Additionally, you can use the Caption and Condition attributes to refer to different datalayers.

On the other hand, the AnnotationLabel.Mock is used to create an annotation label and position it on the chart by linking it to a created mock point. With this element, you can specify the coordinate, relative position, and axis.

Both the AnnotationLabel.Point and AnnotationLabel.Mock have attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text.

Series.Area - Using Input Selection

This series can also be used with the **Input Selection** family of elements, which turn the chart into an input control. This allows users, at runtime, to select points or values on the chart with their mouse and your report can then take action using the selected data. This is very useful, for example, for drilling into the data.

For more information about this functionality, see "Input Selection" on page 97.

Series.Area - Using Refresh Series Timer

The **Refresh Series Timer**, added as a Series child element, updates charts automatically, based on a time interval. When the interval is reached, a request is sent back to the web server for updated data. Series are updated with a smooth animation.



The **Refresh Interval** attribute specifies the number of seconds to wait before refreshing the series data. A value is required and must be an integer greater than 0.

Refresh Mode

Series may be refreshed in two ways: either all of the data is refreshed with each request, or the series data automatically slides to the left one data interval per refresh. The refresh mode is selected in the Chart X Axis element's **Axis Type** attribute.

When the Axis Type is *not* set to *DateTimeLinear*, all of the data will be refreshed.

When the Axis Type value is *DateTimeLinear*, newer values are added to the right side of the chart, previous values slide left, and older values disappear as they reach the left edge of the chart. In this case, the Refresh Series Timer element's **Time Span** attribute is used to specify the age of the data included. This value must be in *hh:mm:ss* format (for example "00:02:00" for two minutes) and must be larger than the Refresh Interval attribute value.

When a Time Span attribute is set, the series' datalayer can make use of a special timestamp token, `@Chart.rdTimeSpanStart~`, in

a query to retrieve just the rows necessary to update the chart. On the initial request, this token returns the current time minus the Time Span value. For subsequent refreshes, when *Axis Type = DateTimeLinear* and the X-axis will be sliding, the token returns the last time the chart was updated, so only the newest rows are retrieved. Here's an MS SQL Server query example:

```
SELECT * FROM MyTable WHERE UpdateTime BETWEEN '@Chart.rdTimeSpanStart~' AND '@Function.DateTime~'
```

For best performance, avoid setting a very short refresh interval if a very large number of users will be displaying the report.

Series.Area Range

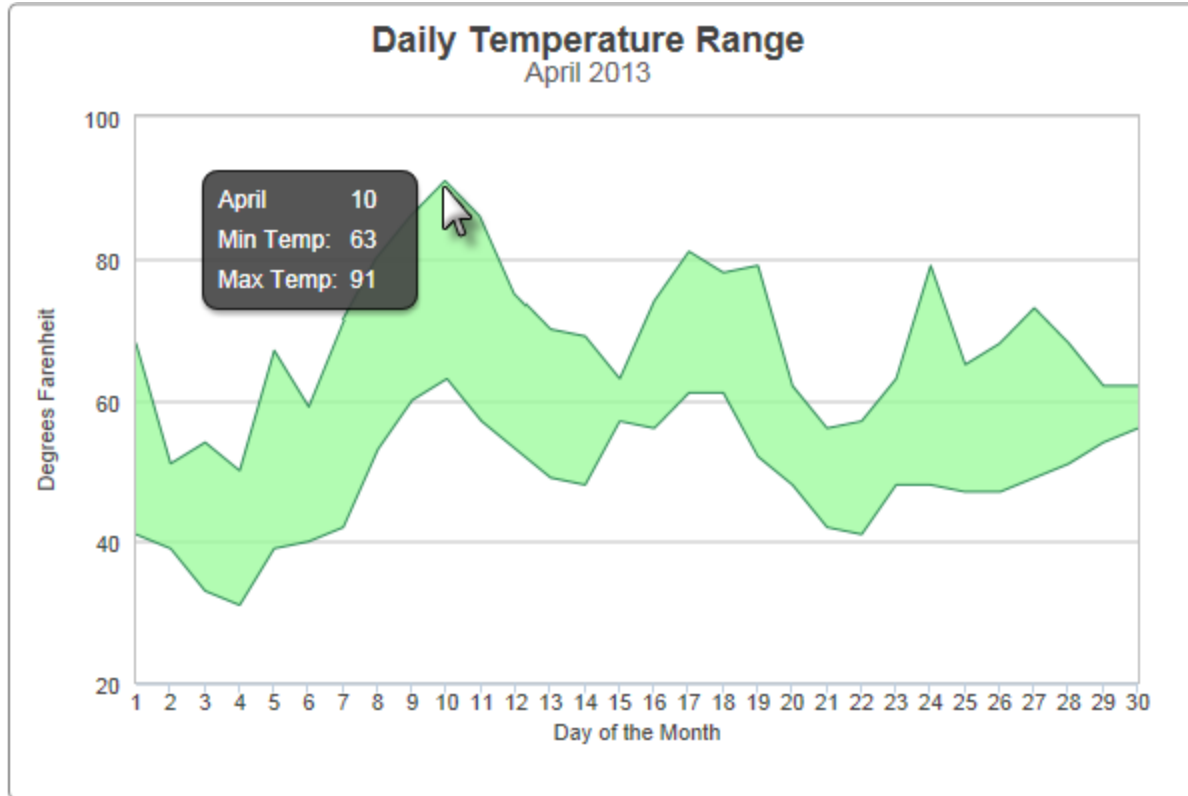
The Chart Canvas element's Series child elements cause a data visualization (the chart) to be rendered in the canvas.

The following topics discuss the Series.Area Range child element:

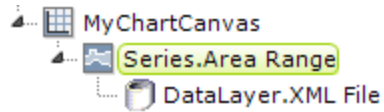
- [Using Multiple Series](#)
- [Series.Area Range Attributes](#)
- [Using the Data Labels Element](#)
- [Using the Quicktips Element](#)
- [Using Action Elements](#)
- [Using the Series Annotation Element](#) v23.1
- [Using Input Selection](#)
- [Using the Refresh Series Timer](#)

About Series.Area Range

The **Series.Area Range** element generates an Area Range chart, which is commonly used to represent sets of low and high values, as numbers or percentages, over time.



The example above shows a simple Area Range chart, presenting the low and high temperatures for a month.

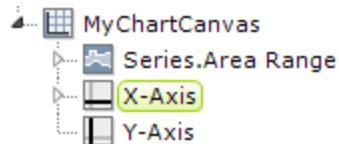


| Element - Series.AreaRange | |
|-----------------------------|------------|
| *Required Attributes | |
| High Value Data Column | MaxTemp |
| Low Value Data Column | MinTemp |
| Optional Attributes | |
| Color | PaleGreen |
| ... | |
| Line Color | SeaGreen |
| Line Color Transparency | |
| ... | |
| X-axis Data Column | DayOfMonth |
| X-axis Data Column Type | |

As shown above, the chart is created by adding Series.Area Range to the canvas, along with a datalayer. Very few attributes need to be set for this Series element in order to produce a basic chart.



A datalayer element can be used either beneath Series.Area Range, as shown above, or beneath Chart Canvas. If used as a child of Chart Canvas, its data is available to *all* child Series elements. This can improve performance if you have several series, all using the *same* data, beneath the same Chart Canvas element.

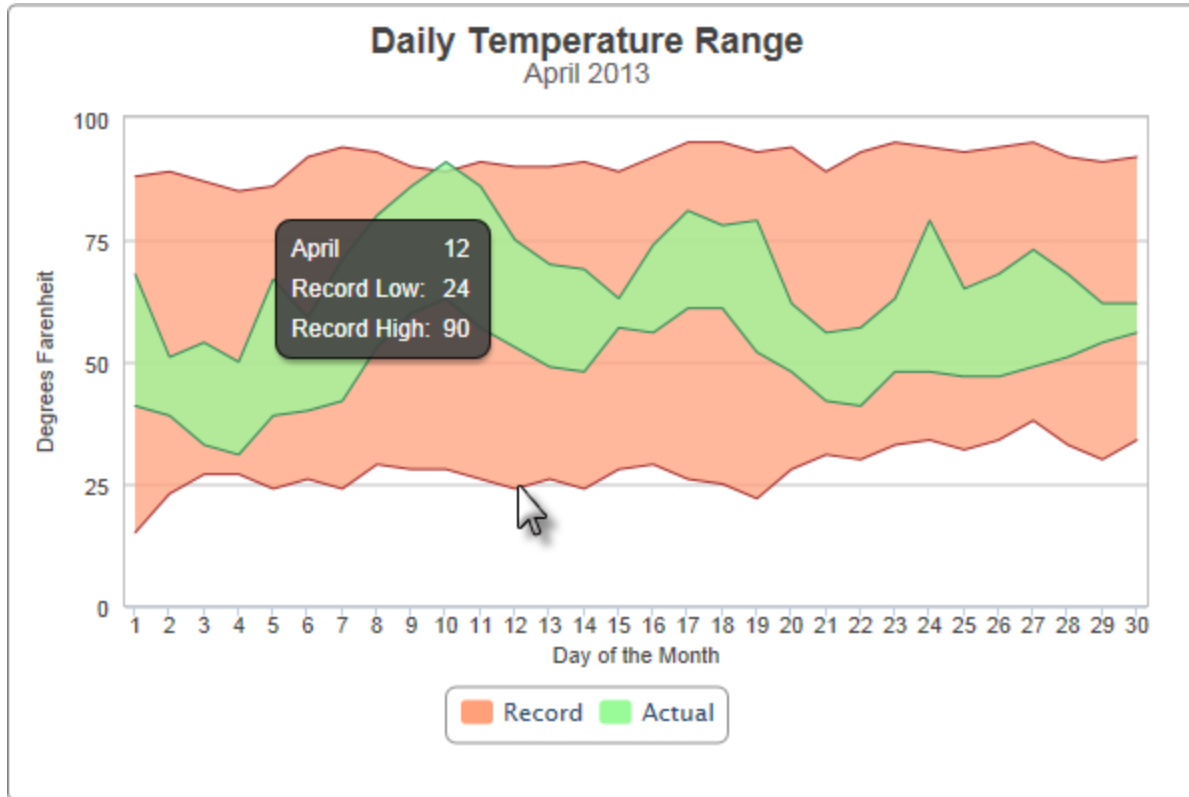


| Element - ChartXAxis | |
|----------------------------|------------------|
| Optional Attributes | |
| Axis Padding Left | |
| Axis Padding Right | |
| ... | |
| Caption | Day of the Month |
| ... | |
| Spacing | |

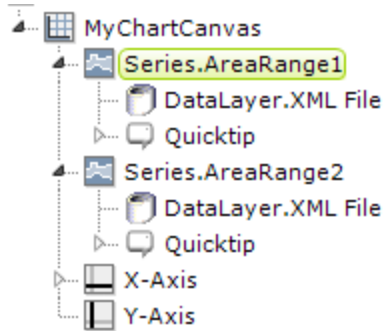
The properties of the X- and Y-axis, including captions, are set using the **X-Axis** and **Y-Axis** elements. For example, in order to provide a caption for the X-axis, add an **X-Axis** element beneath Chart Canvas and set its **Caption** attribute, as shown above. Repeat for the Y-axis.

Series.Area Range - Using Multiple Series

You can add additional series to the chart by adding additional Series elements:



The example above shows two series, one for each set of temperature ranges. A legend can also be added very easily.



| Element - Series.AreaRange | |
|-----------------------------|-------------------|
| *Required Attributes | |
| High Value Data Column | RecHigh |
| Low Value Data Column | RecLow |
| Optional Attributes | |
| Color | LightSalmon |
| Combine With Series ID | |
| ... | |
| Disable Mouse Tracking | |
| ID | Series.AreaRange1 |
| Legend Label | Record |
| Line Color | Brown |
| ... | |
| X-axis Data Column | DayOfMonth |
| X-axis Data Column Type | |

The example above shows the two Series elements, their datalayers and optional Quicktips, and the X-Axis and Y-Axis elements used to produce the previous chart. You can adjust which data range appears "in front" of the other in the chart by changing the order of the Series elements in the definition. Setting the Series elements' **Legend Label** attribute will automatically cause the legend to be displayed.

Charts with multiple series or aggregations are encouraged to use the ChartCanvas element's attribute, `Tooltip Split`. This attribute allows you to split a chart's tooltip into one label per series, rather than per segment. The default value for this attribute is *False*. Once enabled, when hovering over a data point, it displays all the values for the series. This method is recommended over shared tooltips for charts with multiple line series, and as such, takes precedence over `tooltip.shared`.



When using multiple series, you may be able to reduce the number of data queries and improve performance by using local data to read all of the data once, see *Datalayer Introduction*. Then, link its datalayer to share it to the series, see *Link Datalayers*.

At each Series element, link its datalayer to the shared Local Data datalayer and filter the data to meet the needs of each individual series.

You can combine different types of Series elements, for example, Series.Area and Series.Bar, to produce combinations of visualizations.

v23.1 If you are using the Chart Color Axis element in a mult-series chart, by default, the series will link to the first color axis. To link the series to a special color axis, set the ID attribute of the Chart Color Axis element to the corresponding Series' Linked to Color Axis ID attribute. For more information, see "Chart Color Axis" on page 124.

Series.Area Range - Attributes

The Series.Area Range element has the following attributes:

| Attribute | Description |
|--|---|
| High Value Data Column | (Required) Specifies the name of the datalayer column containing the high data value for each row. |
| Low Value Data Column | (Required) Specifies the name of the datalayer column containing the lowdata value for each row. |
| v23.1 Class Name | Specifies the Class Name referenced in the .css style sheet. The default value is <i>undefined</i> . |
| Color | Sets the data region fill color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| v23.1 Color Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the x-axis. |
| Combine With Series ID | Set this attribute to the element ID of another series to combine it with this series in the legend. When two series are combined, by default only the first one will appear in the legend but clicking the item in the legend toggles both series to appear and disappear. Or, the value <i>Previous</i> can be entered to combine this series with the previous series. |

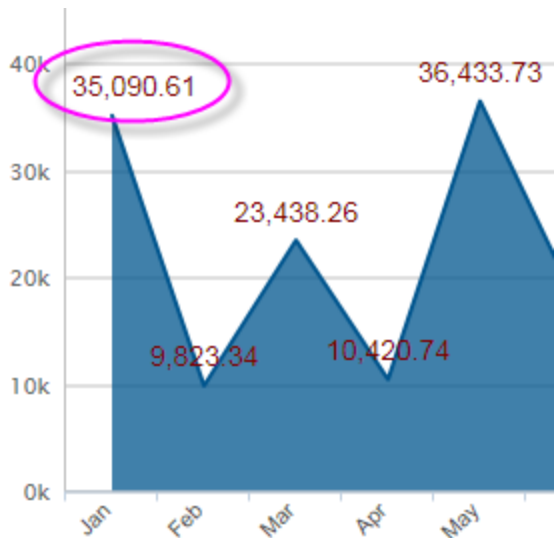
| Attribute | Description |
|-------------------------|--|
| Connect Nulls | Specifies if data rows with null or blank values are to be ignored, allowing adjacent values to be connected in the chart. The default value is <i>False</i> . |
| Disable Mouse Tracking | Disables mouse tracking for the series, when set to <i>True</i> . This affects tooltips and click events on graphs and points. For large datasets, this may improve performance. The default value is <i>False</i> . |
| Hover Line Thickness | Sets the thickness of the line, in pixels, when the mouse pointer is hovered over it. The default value is <i>1</i> pixel. |
| Legend Label | Indicates text that will be shown for this series inside the chart legend. When a value is provided, automatically causes the legend to be displayed. |
| Line Color | Sets the data region's border line color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| Line Color Transparency | Specifies the transparency of the data region border line color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| Line Style | Specifies the pattern of the data region's border line as either <i>Solid</i> or a combination of dashes and dots. |
| Line Thickness | Specifies the thickness of the data region's border line, in pixels. The default value is <i>1</i> pixel. |

| Attribute | Description |
|--|---|
| <p>v23.1</p> <p>Linked to Color-Axis ID</p> | <p>Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes</p> |
| <p>Linked to X-Axis ID</p> | <p>Specifies the ID of an X-Axis element that this series should be linked to when using multiple X-axes.</p> |
| <p>Linked to Y-Axis ID</p> | <p>Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes.</p> |
| <p>Negative Color</p> | <p>Sets the color for negative values. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #D5F484. The default value is <i>Red</i>.</p> |
| <p>Negative Color Transparency</p> | <p>Specifies the transparency of the Negative Color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other.</p> |
| <p>Negative Fill Color</p> | <p>Sets the data region fill color for negative values. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #D5F484. The default value is <i>Red</i>.</p> |
| <p>Negative Fill Color Transparency</p> | <p>Specifies the transparency of the Negative Fill Color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other.</p> |

| Attribute | Description |
|-------------------------|---|
| Negative Threshold | Sets the positive-negative value threshold, also called the "zero level" or "base level". The default value is <i>0</i> . |
| Transparency | Specifies the transparency of the data region fill color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| X-Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the X-axis. |
| X-Axis Data Column Type | <p>Specifies the data type of the datalayer column named in the X-axis Data Column attribute. Options include <i>Auto</i> (the default), <i>Text</i>, <i>Number</i>, and <i>DateTime</i>.</p> <p>By default, X-axis data values that are <i>DateTime</i> type will be automatically distributed evenly across the time series. If you want to disable this behavior, set this attribute value to <i>Text</i>.</p> |

Series.Area Range - Using the Data Labels Element

A "data label" is text shown next to each data point that shows its value. When the **Data Labels** element is used as a child of **Series.Area Range**, text representing the data values will appear on the chart:

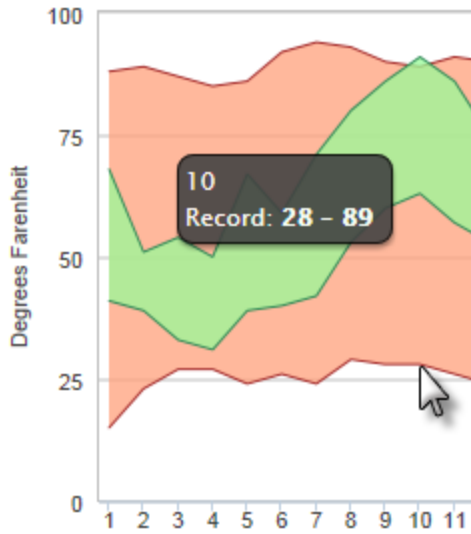


The Data Labels element has attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text. One attribute, **Series Name as Caption**, allows you to specify the y-axis label, x-axis label, or legend label as the data point value inside the chart.

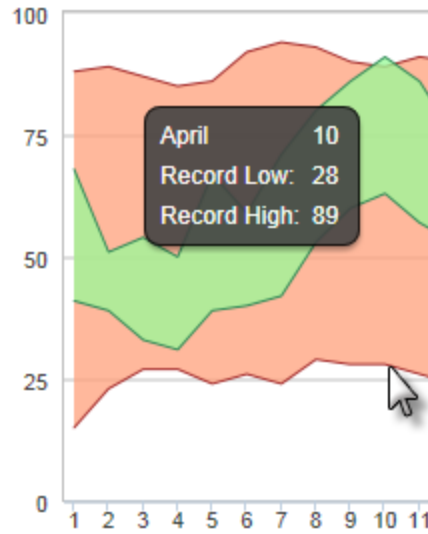
The Data Labels element's color-related attribute values can be set using @Chart tokens. **v23.1** Or, utilize the "Class Name" attribute to apply unique styling to your data labels by referencing a class name from a stylesheet in the report.

Series.Area Range - Using the Quicktips Element

By default, a "blacktop" is displayed when the mouse hovers near or over a data point:

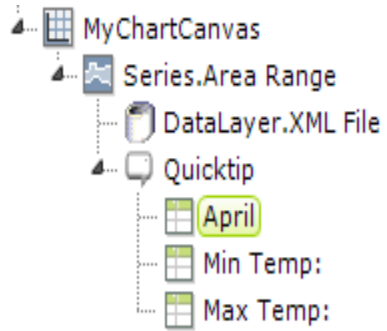


Default quicktip



With Quicktip child element


The automatically-generated quicktip displays information for the X- and Y-axis, as shown above, left. However, you may want to display other information or format it differently, as shown above, right, which can be done by adding a **Quicktip** child element beneath Series.Area Range.



| Optional Attributes | |
|---------------------|--------------------|
| Caption | April |
| Format | |
| ID | |
| Value | @Chart.DayOfMonth~ |

The example above shows a Quicktip element (no attributes need to be set) and three **Quicktip Row** child elements, used to create the quicktip shown in the previous image. Use @Chart tokens to include chart data in the quicktip.

You can also configure the tooltip to display values for any available column in the datalayer. To do so, enter a token representing the data column in the Value attribute of the Quicktip Row element.

 To use this feature with DataLayer.ActiveSQL, please make sure the keep Grouped Rows attribute of the SqlGroup element is set to *False*.

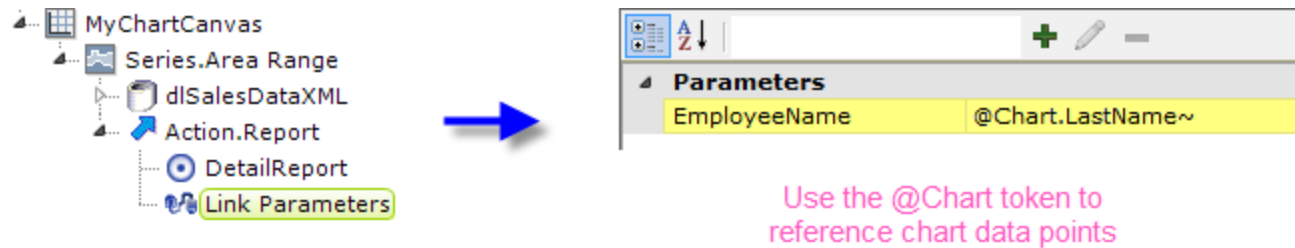
To utilize this feature, add a Quicktip Row element below the Quicktip. Then, enter a token representing the data column in the Value attribute.

Intrinsic functions are supported in the Quicktip attributes.

The Quicktip Row element has been made context-sensitive with the addition of a **Condition** attribute.

Series.Area Range - Using Action Elements

Action elements initiate processing of a report or process task definition, redirection to a link, or other processing when a data point in the series is clicked.



In the example above, an **Action.Report** element has been added as a child of the Series, along with its **Target.Report** and **Link Parameters** child elements. To reference chart data in parameters, use the @Chart token, as shown above.

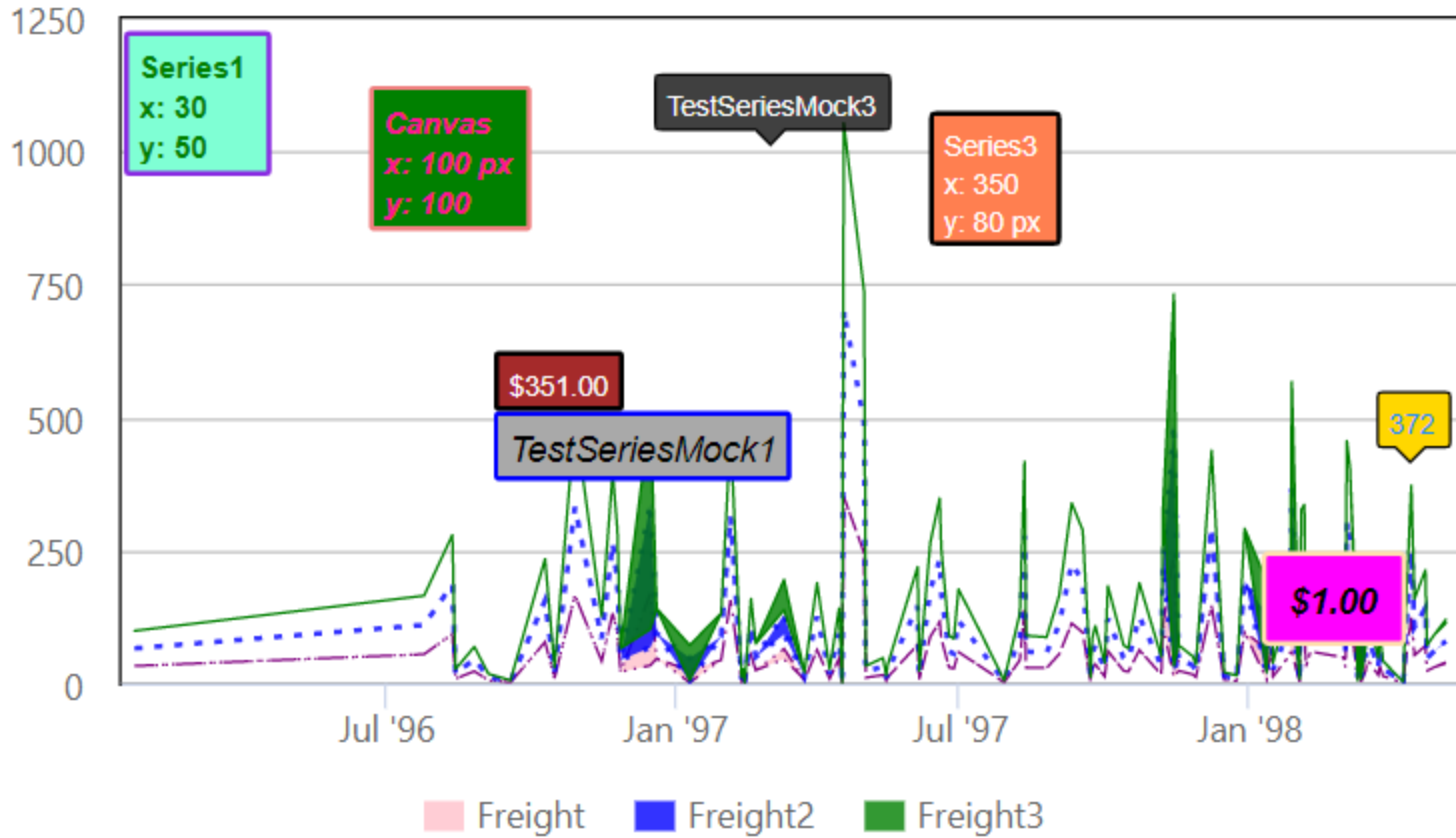
A variety of Action elements are available for use with Series, including Action.Link, Action.Process, and Action.Refresh Element. Additional Action elements will be added in future releases.

v23.1

Series.Area Range - Using the Series Annotation Element

Annotations allow you to annotate a chart freely using custom labels and shapes. When the **Series Annotation** element is used as a child of Series.Area Range, you can place these annotations at various points of interest:

Area Range



Exports:-

ExportToPdf

ExportToExcel

ExportToWord

The Series Annotation element offers two child elements: **AnnotationLabel.Point** and **AnnotationLabel.Mock**.

The AnnotationLabel.Point child element is used to create an annotation label and position on the chart by linking it to an existing point. With this element, you cannot specify the value, it always refers to the value of the point. Additionally, you can use the Caption and Condition attributes to refer to different datalayers.

On the other hand, the AnnotationLabel.Mock is used to create an annotation label and position it on the chart by linking it to a created mock point. With this element, you can specify the coordinate, relative position, and axis.

Both the AnnotationLabel.Point and AnnotationLabel.Mock have attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text.

Series.Area Range - Using Input Selection

This series can also be used with the **Input Selection** family of elements, which turn the chart into an input control. This allows users, at runtime, to select points or values on the chart with their mouse and your report can then take action using the selected data. This is very useful, for example, for drilling into the data.

For more information about this functionality, see "Input Selection" on page 97.

Series.Area Range - Using the Refresh Series Timer

The **Refresh Series Timer**, added as a Series child element, updates charts automatically, based on a time interval. When the interval is reached, a request is sent back to the web server for updated data. Series are updated with a smooth animation.



The **Refresh Interval** attribute specifies the number of seconds to wait before refreshing the series data. A value is required and must be an integer greater than 0.

Refresh Mode

Series may be refreshed in two ways: either all of the data is refreshed with each request, or the series data automatically slides to the left one data interval per refresh. The refresh mode is selected in the Chart X Axis element's **Axis Type** attribute.

When the Axis Type is *not* set to *DateTimeLinear*, all of the data will be refreshed.

When the Axis Type value is *DateTimeLinear*, newer values are added to the right side of the chart, previous values slide left, and older values disappear as they reach the left edge of the chart. In this case, the Refresh Series Timer element's **Time Span** attribute is used to specify the age of the data included. This value must be in *hh:mm:ss* format (for example "00:02:00" for two minutes) and must be larger than the Refresh Interval attribute value.

When a Time Span attribute is set, the series' datalayer can make use of a special timestamp token, `@Chart.rdTimeSpanStart~`, in

a query to retrieve just the rows necessary to update the chart. On the initial request, this token returns the current time minus the Time Span value. For subsequent refreshes, when *Axis Type = DateTimeLinear* and the X-axis will be sliding, the token returns the last time the chart was updated, so only the newest rows are retrieved. Here's an MS SQL Server query example:

```
SELECT * FROM MyTable WHERE UpdateTime BETWEEN '@Chart.rdTimeSpanStart~' AND '@Function.DateTime~'
```

For best performance, avoid setting a very short refresh interval if a very large number of users will be displaying the report.

Series.Area Spline

The Chart Canvas element's Series child elements cause a data visualization (the chart) to be rendered in the canvas.

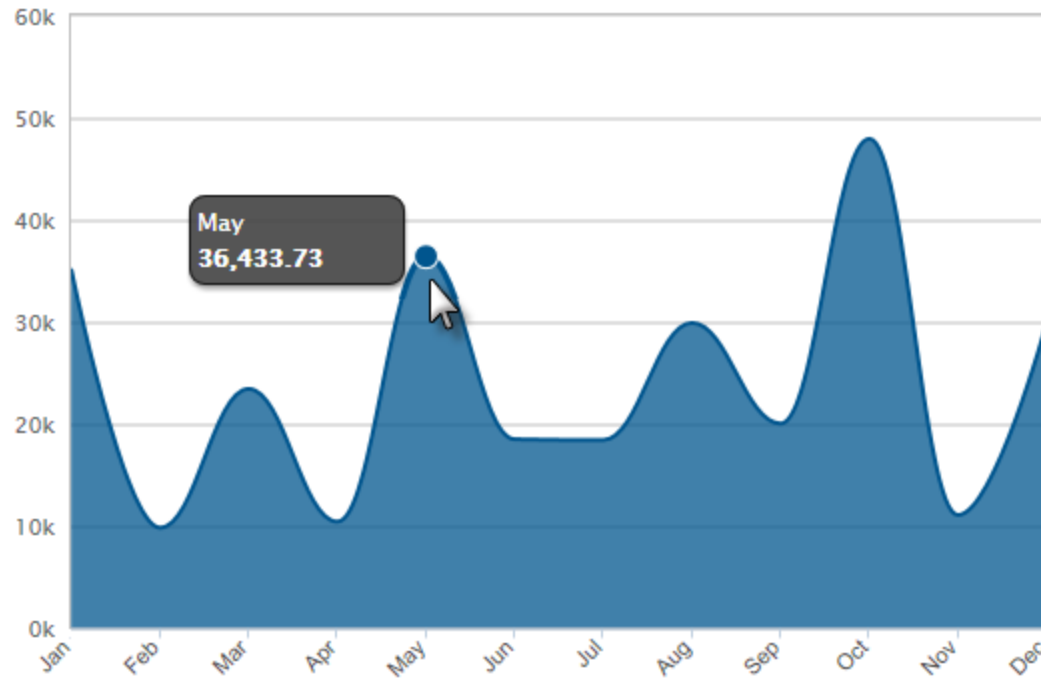
The following topics discuss the Series.Area Spline child element:

- [Using Multiple Series](#)
- [Series.Area Spline Attributes](#)
- [Using the Data Labels Element](#)
- [Using the Marker Points Element](#)
- [Using the Quicktips Element](#)
- [Using the Trend Line Element](#)
- [Using Action Elements](#)
- [Using the Series Annotation Element](#) v23.1
- [Using Input Selection](#)
- [Using the Refresh Series Timer](#)

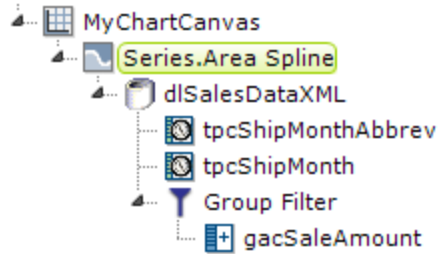
About Series.Area Spline

The **Series.Area Spline** element generates an Area chart with the characteristic curves of a Spline chart, which is commonly used to represent aggregated totals, as numbers or percentages, over time.

Monthly Sales 2013




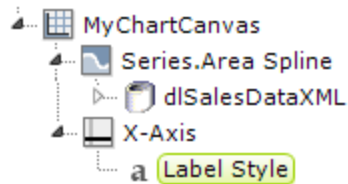
The example above shows a simple Area Spline chart, representing sales per month for a year. Notice that, unlike a regular Area chart, the data region border is smoothly curved.



| Element - Series.AreaSpline | |
|-----------------------------|--------------------|
| *Required Attributes | |
| Y-axis Data Column | gacSaleAmount |
| Optional Attributes | |
| Color | |
| Combine With Series ID | |
| ... | |
| X-axis Data Column | tpcShipMonthAbbrev |
| X-axis Data Column Type | |

As shown above, the chart is created by adding Series.Area Spline to the canvas, along with a datalayer and, typically, some child elements that may include **Time Period Column** elements, a **Group Filter**, and a **Group Aggregate Column** element. Very few attributes need to be set for the Series element in order to produce a basic chart.

 A datalayer element can be used either beneath Series.Area Spline, as shown above, or beneath Chart Canvas. If used as a child of Chart Canvas, its data is available to *all* child Series elements. This can improve performance if you have several series, all using the *same* data, beneath the same Chart Canvas element.



| Element - AxisLabelStyle | |
|----------------------------|----|
| Optional Attributes | |
| Font Angle | 45 |
| Font Color | |
| ... | |
| Stagger Labels | |

The properties of the X- and Y-axis, including captions, are set using the **X-Axis** and **Y-Axis** elements. For example, in order to angle the X-axis labels, add an X-Axis element beneath Chart Canvas (none of its attributes need to be set) and add its child **Label Style** element. Set the Label Style element's attribute as shown above.

Series.Area Spline - Using the Refresh Series Timer

The **Refresh Series Timer**, added as a Series child element, updates charts automatically, based on a time interval. When the interval is reached, a request is sent back to the web server for updated data. Series are updated with a smooth animation.



The **Refresh Interval** attribute specifies the number of seconds to wait before refreshing the series data. A value is required and must be an integer greater than 0.

Refresh Mode

Series may be refreshed in two ways: either all of the data is refreshed with each request, or the series data automatically slides to the left one data interval per refresh. The refresh mode is selected in the Chart X Axis element's **Axis Type** attribute. When the Axis Type is *not* set to *DateTimeLinear*, all of the data will be refreshed.

When the Axis Type value is *DateTimeLinear*, newer values are added to the right side of the chart, previous values slide left, and older values disappear as they reach the left edge of the chart. In this case, the Refresh Series Timer element's **Time Span** attribute is used to specify the age of the data included. This value must be in *hh:mm:ss* format (for example "00:02:00" for two minutes) and must be larger than the Refresh Interval attribute value.

When a Time Span attribute is set, the series' datalayer can make use of a special timestamp token, `@Chart.rdTimeSpanStart~`, in

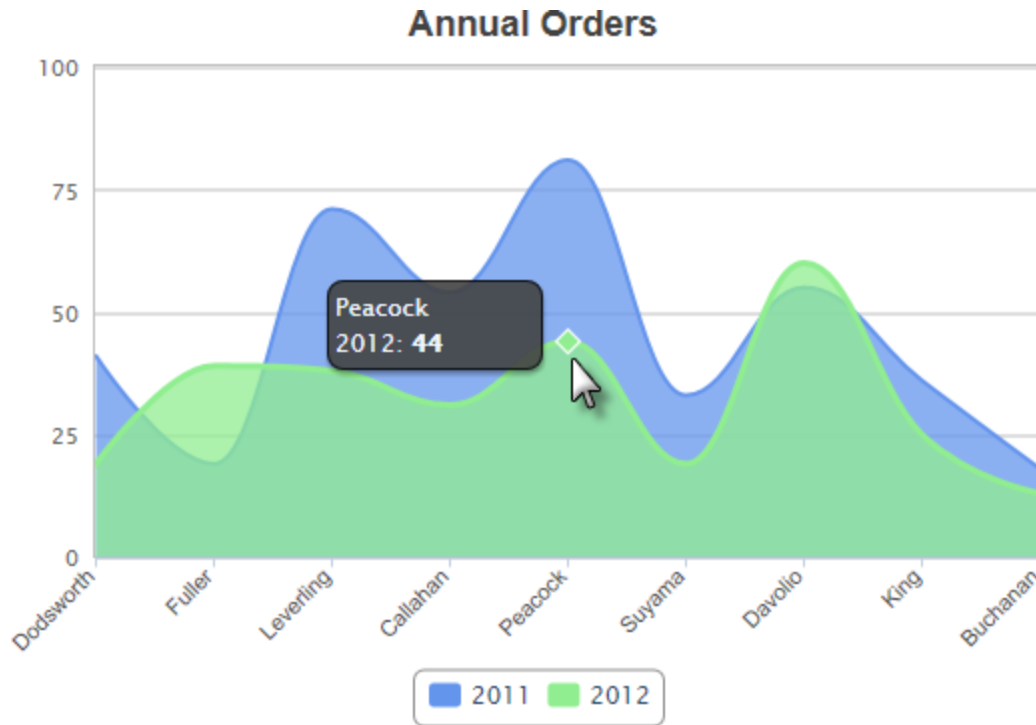
a query to retrieve just the rows necessary to update the chart. On the initial request, this token returns the current time minus the Time Span value. For subsequent refreshes, when *Axis Type = DateTimeLinear* and the X-axis will be sliding, the token returns the last time the chart was updated, so only the newest rows are retrieved. Here's an MS SQL Server query example:

```
SELECT * FROM MyTable WHERE UpdateTime BETWEEN '@Chart.rdTimeSpanStart~' AND '@Function.DateTime~'
```

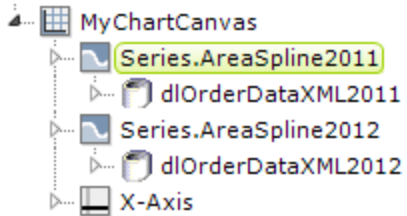
For best performance, avoid setting a very short refresh interval if a very large number of users will be displaying the report.

Series.Area Spline - Using Multiple Series

You can add additional series to the chart by adding additional Series elements:




The example above shows two Series, one for each year. A legend can also be added very easily.



| Element - Series.AreaSpline | |
|-----------------------------|-----------------------|
| *Required Attributes | |
| Y-axis Data Column | gacOrderCount |
| Optional Attributes | |
| Color | CornflowerBlue |
| Combine With Series ID | |
| ... | |
| ID | Series.AreaSpline2011 |
| Legend Label | 2011 |
| Line Color | |
| ... | |
| X-axis Data Column | LastName |
| X-axis Data Column Type | |

The example above shows the two Series elements, their datalayers, and an X-Axis element used to produce the previous chart. You can adjust which series appears "in front" of the other in the chart by changing the order of the Series elements in the definition. Setting the Series elements' **Legend Label** attribute will automatically cause the legend to be displayed.

Charts with multiple series or aggregations are encouraged to use the ChartCanvas element's attribute, `Tooltip Split`. This attribute allows you to split a chart's tooltip into one label per series, rather than per segment. The default value for this attribute is *False*. Once enabled, when hovering over a data point, it displays all the values for the series. This method is recommended over shared tooltips for charts with multiple line series, and as such, takes precedence over `tooltip.shared`.

 When using multiple series, you may be able to reduce the number of data queries and improve performance by using local data to read all of the data once, see *Datalayer Introduction*. Then, link its datalayer to share it to the series, see *Link Datalayers*. At each Series element, link its datalayer to the shared Local Data datalayer and filter the data to meet the needs of each individual series.

You can combine different types of Series elements, for example, Series.Area Spline and Series.Bar, to produce combinations of visualizations.

v23.1 If you are using the Chart Color Axis element in a mult-series chart, by default, the series will link to the first color axis. To link the series to a special color axis, set the ID attribute of the Chart Color Axis element to the corresponding Series' Linked to Color Axis ID attribute. For more information, see "Chart Color Axis" on page 124.

Series.Area Spline - Attributes

The Series.Area Spline element has the following attributes:

| Attribute | Description |
|--|---|
| Y-Axis Data Column | (Required) Specifies the name of a datalayer column whose values will be plotted along the Y-axis. |
| v23.1 Class Name | Specifies the Class Name referenced in the .css style sheet. The default value is <i>undefined</i> . |
| Color | Sets the data region fill color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| v23.1 Color Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the x-axis. |
| Combine With Series ID | Set this attribute to the element ID of another series to combine it with this series in the legend. When two series are combined, by default only the first one will appear in the legend but clicking the item in the legend toggles both series to appear and disappear. Or, the value <i>Previous</i> can be entered to combine this series with the previous series. |
| Connect Nulls | Specifies if data rows with null or blank values are to be ignored, allowing adjacent values to be connected in the chart. The default value is <i>False</i> . |

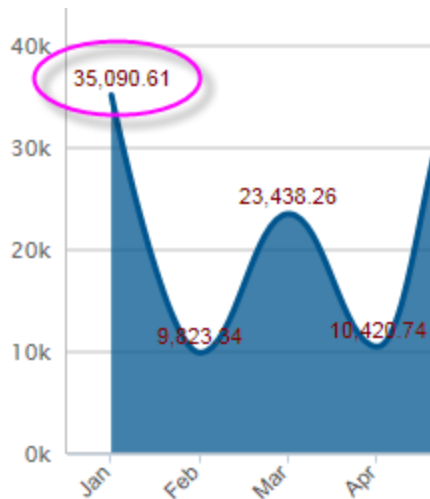
| Attribute | Description |
|---|--|
| Disable Mouse Tracking | Disables mouse tracking for the series, when set to <i>True</i> . This affects tooltips and click events on graphs and points. For large datasets, this may improve performance. The default value is <i>False</i> . |
| Hover Line Thickness | Sets the thickness of the line, in pixels, when the mouse pointer is hovered over it. The default value is <i>1</i> pixel. |
| Legend Label | Indicates text that will be shown for this series inside the chart legend. When a value is provided, automatically causes the legend to be displayed. |
| Line Color | Sets the data region's border line color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| Line Color Transparency | Specifies the transparency of the data region border line color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| Line Style | Specifies the pattern of the data region's border line as either <i>Solid</i> or a combination of dashes and dots. |
| Line Thickness | Specifies the thickness of the data region's border line, in pixels. The default value is <i>1</i> pixel. |
| v23.1 Linked to Color- | Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes |

| Attribute | Description |
|-----------------------------|---|
| Axis ID | |
| Linked to X-Axis ID | Specifies the ID of an X-Axis element that this series should be linked to when using multiple X-axes. |
| Linked to Y-Axis ID | Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes. |
| Negative Color | Sets the color for negative values. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #D5F484. The default value is <i>Red</i> . |
| Negative Color Transparency | Specifies the transparency of the Negative Color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| Negative Threshold | Sets the positive-negative value threshold, also called the "zero level" or "base level". The default value is <i>0</i> . |
| Stack Group Name | Enables the grouping of multiple series in a stacked chart. Enter the same arbitrary string value in this attribute for each series to be stacked. |
| Stacking | Sets the stacking mode. <i>Normal</i> stacks series on top of each other, while <i>Percent</i> stacks series, then raises the levels to 100 percent, showing the percentage for each series' values. The default value is <i>None</i> for no stacking. |

| Attribute | Description |
|-------------------------|---|
| Transparency | Specifies the transparency of the data region fill color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| X-Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the X-axis. |
| X-Axis Data Column Type | Specifies the data type of the datalayer column named in the X-axis Data Column attribute. Options include <i>Auto</i> (the default), <i>Text</i> , <i>Number</i> , and <i>DateTime</i> . By default, X-axis data values that are <i>DateTime</i> type will be automatically distributed evenly across the time series. If you want to disable this behavior, set this attribute value to <i>Text</i> . |

Series.Area Spline - Using the Data Label Element

A "data label" is text shown next to each data point that shows its value. When the **Data Labels** element is used as a child of Series.Area Spline, text representing the data values will appear on the chart:

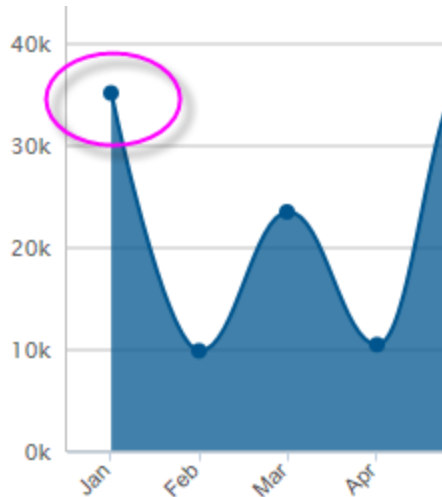


The Data Labels element has attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text. One attribute, Series Name as Caption, allows you to specify the y-axis label, x-axis label, or legend label as the data point value inside the chart.

The Data Labels element's color-related attribute values can be set using @Chart tokens. **v23.1** Or, utilize the "Class Name" attribute to apply unique styling to your data labels by referencing a class name from a stylesheet in the report.

Series.Area Spline - Using the Marker Points Element

A "marker point" is a symbol that appears on the chart at each data point. When the **Marker Points** element is used as a child of Series.Area Spline, a small dot matching the color of the area will be displayed at each data point:



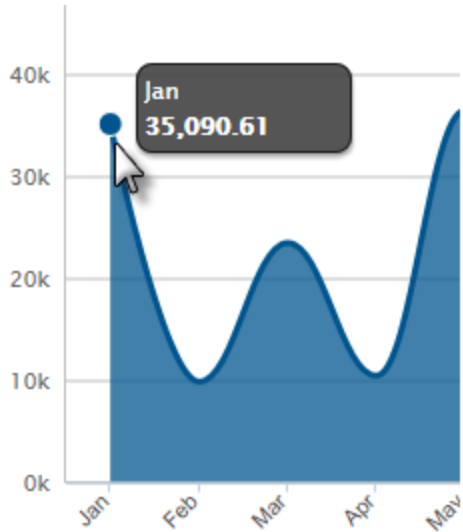
The default marker points are shown above. The **Marker Points** element allows you to select a different symbol for the marker point, and to control its size, color, border color, and transparency. When the cursor hovers over a marker point, it's automatically enlarged slightly.

v23.1

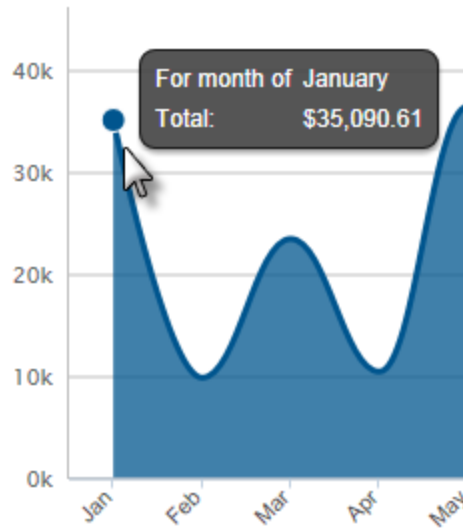
You can also create your own marker point symbol using the Chart Canvas child element, Chart Custom Symbol. Once defined, link your custom symbol using the Marker Points element's Symbol attribute. For more information, see "Chart Custom Symbol" on page 128.

Series.Area Spline - Using the Quicktips Element

By default, a "quicktip" is displayed when the mouse hovers near or over a data point:




Default quicktip



With Quicktip child element

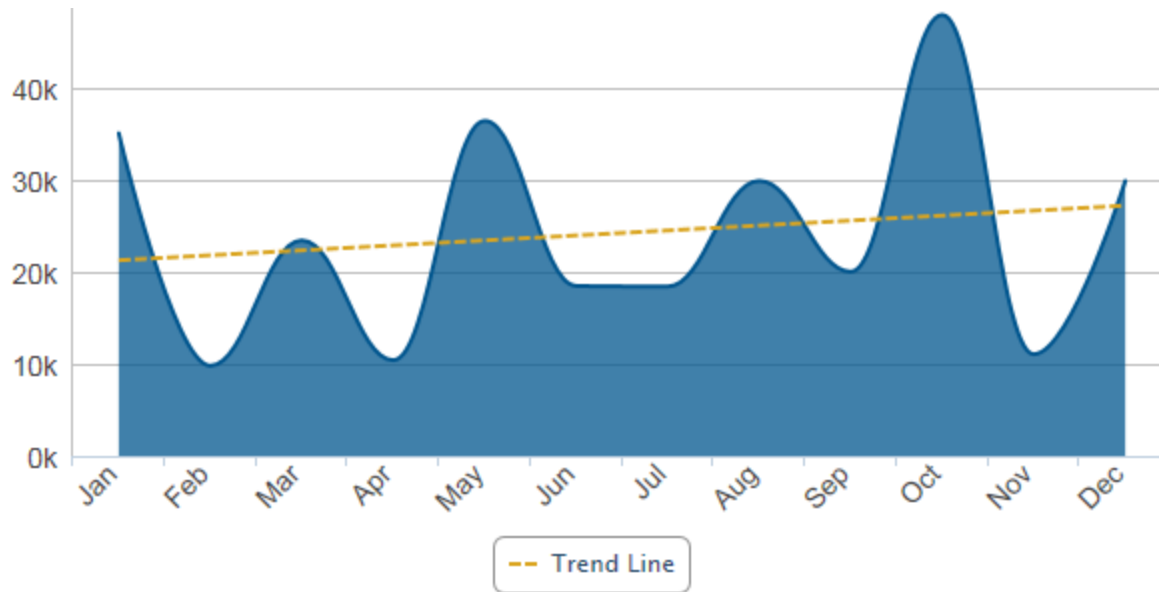
The automatically-generated quicktip displays information for the X- and Y-axis, as shown above, left. However, you may want to display other information or format it differently, perhaps as shown above, right, which can be done by adding a **Quicktip** child element beneath Series.Area Spline and setting its attributes and child elements. Use @Chart tokens to include chart data in the quicktip. Intrinsic functions are supported in the Quicktip attributes.

You can also configure the tooltip to display values for any available column in the datalayer. To do so, enter a token representing the data column in the Value attribute of the Quicktip Row element.

 To use this feature with `DataLayer.ActiveSQL`, please make sure the `keep Grouped Rows` attribute of the `SqlGroup` element is set to *False*.

Series.Area Spline - Using the Trend Line Element

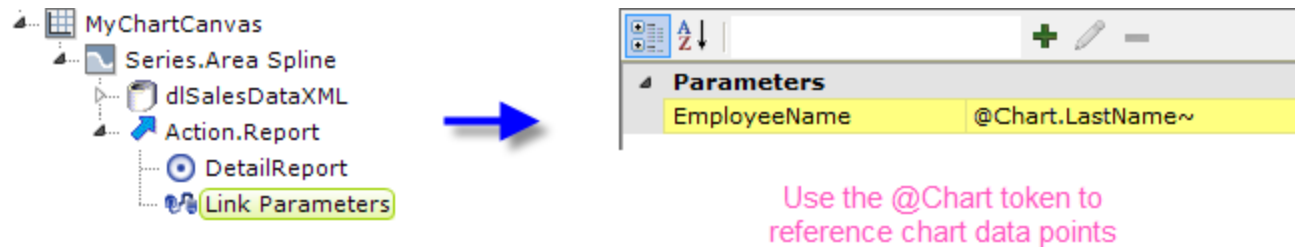
The **Trend Line** element creates a line on the chart that indicates the "trend" of the data. The line connects a number of data points generated using a regression algorithm.



The Trend Line element is a child of the Series.Area Spline element and can be styled for color, line width, etc. When configured with a legend label, it will be represented by an item in the legend, as shown above.

Series.Area Spline - Using Action Elements

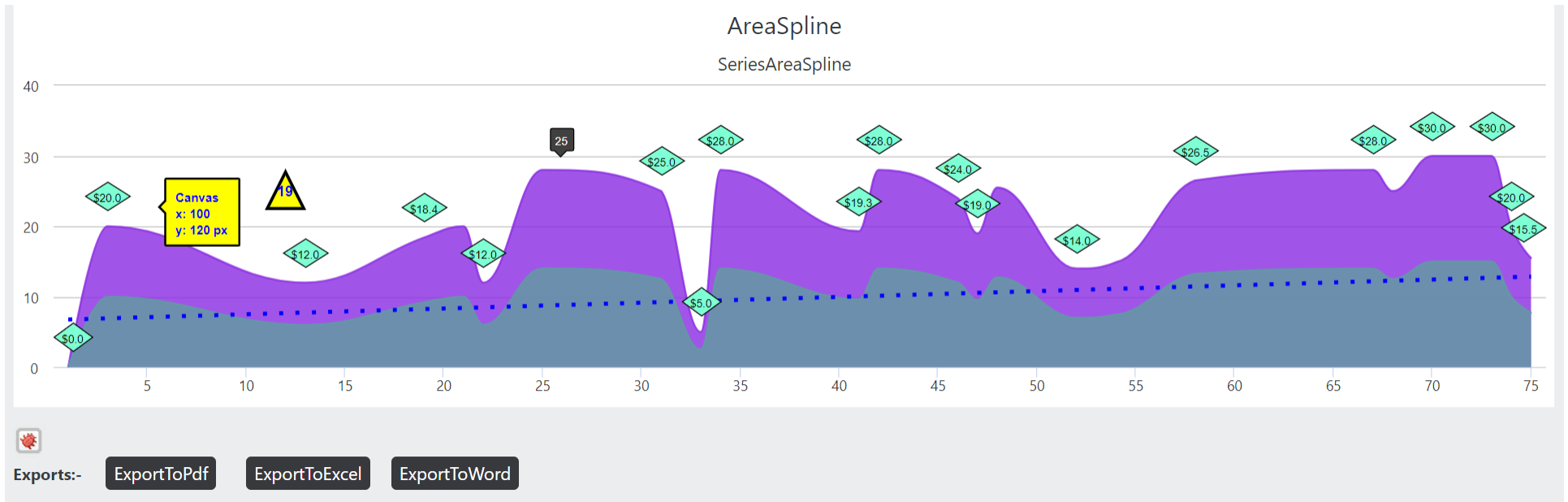
Action elements initiate processing of a report or process task definition, redirection to a link, or other processing when a data point in the series is clicked.



In the example above, an **Action.Report** element has been added as a child of the Series, along with its **Target.Report** and **Link Parameters** child elements. To reference chart data in parameters, use the @Chart token, as shown above. A variety of Action elements are available for use with Series, including Action.Link, Action.Process, and Action.Refresh Element. Additional Action elements will be added in future releases.

v23.1 **Series.Area Spline - Using the Series Annotation Element**

Annotations allow you to annotate a chart freely using custom labels and shapes. When the **Series Annotation** element is used as a child of Series.Area Spline, you can place these annotations at various points of interest:



The Series Annotation element offers two child elements: **AnnotationLabel.Point** and **AnnotationLabel.Mock**.

The AnnotationLabel.Point child element is used to create an annotation label and position on the chart by linking it to an existing point. With this element, you cannot specify the value, it always refers to the value of the point. Additionally, you can use the Caption and Condition attributes to refer to different datalayers.

On the other hand, the AnnotationLabel.Mock is used to create an annotation label and position it on the chart by linking it to a created mock point. With this element, you can specify the coordinate, relative position, and axis.

Both the AnnotationLabel.Point and AnnotationLabel.Mock have attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text.

Series.Area Spline - Using Input Selection

This series can also be used with the **Input Selection** family of elements, which turn the chart into an input control. This allows users, at runtime, to select points or values on the chart with their mouse and your report can then take action using the selected data. This is very useful, for example, for drilling into the data.

For more information about this functionality, see "Input Selection" on page 97.

Series.Area Spline Range

The Chart Canvas element's Series child elements cause a data visualization (the chart) to be rendered in the canvas.

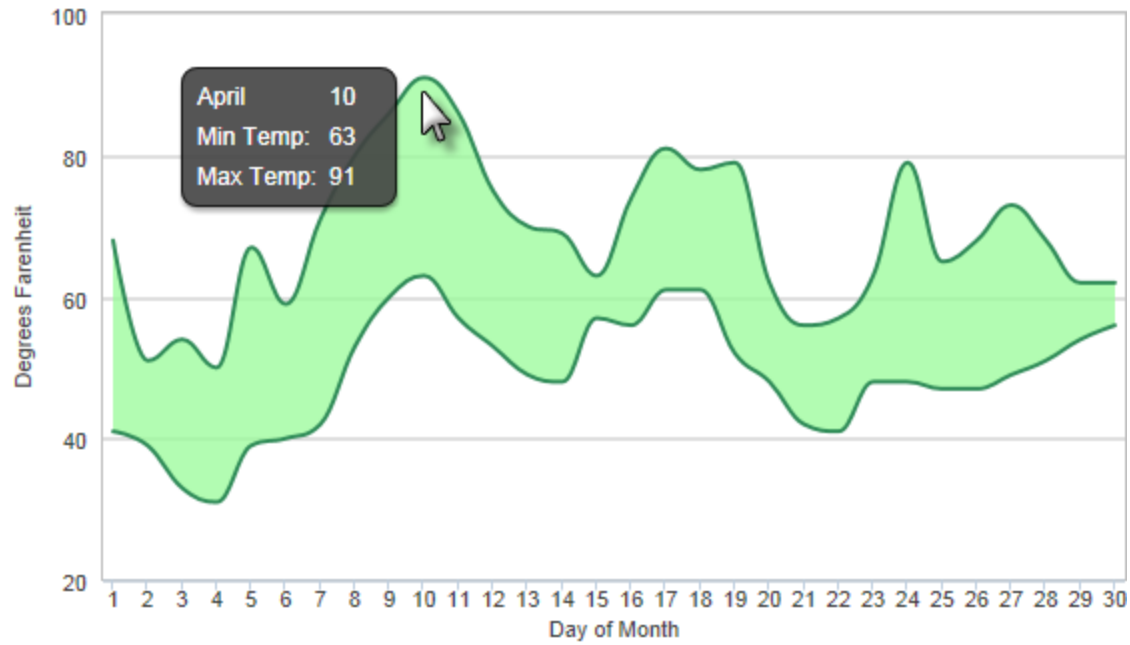
The following topics discuss the Series.Area Spline Range child element:

- [Using Multiple Series](#)
- [Series.Area Spline Range Attributes](#)
- [Using the Data Labels Element](#)
- [Using the Quicktips Element](#)
- [Using Action Elements](#)
- [Using the Series Annotation Element](#) v23.1
- [Using Input Selection](#)
- [Using the Refresh Series Timer](#)

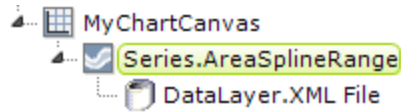
About Series.Area Spline Range

The **Series.Area Spline Range** element generates an Area Spline Range chart, which is commonly used to represent sets of low and high values, as numbers or percentages, over time.

Daily Temperature Range April 2013



The example above shows a simple Area Spline Range chart, presenting the low and high temperatures for a month.

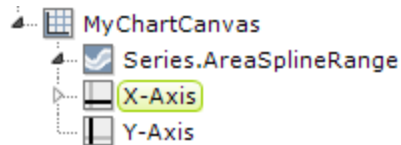


| Element - Series.AreaSplineRange | |
|----------------------------------|------------|
| *Required Attributes | |
| High Value Data Column | MaxTemp |
| Low Value Data Column | MinTemp |
| Optional Attributes | |
| Color | PaleGreen |
| ... | |
| Line Color | SeaGreen |
| Line Color Transparency | |
| ... | |
| X-axis Data Column | DayOfMonth |
| X-axis Data Column Type | |

As shown above, the chart is created by adding Series.Area Spline Range to the canvas, along with a datalayer. Very few attributes need to be set for this Series element in order to produce a basic chart.



A datalayer element can be used either beneath Series.Area Spline Range, as shown above, or beneath Chart Canvas. If used as a child of Chart Canvas, its data is available to *all* child Series elements. This can improve performance if you have several series, all using the *same* data, beneath the same Chart Canvas element.

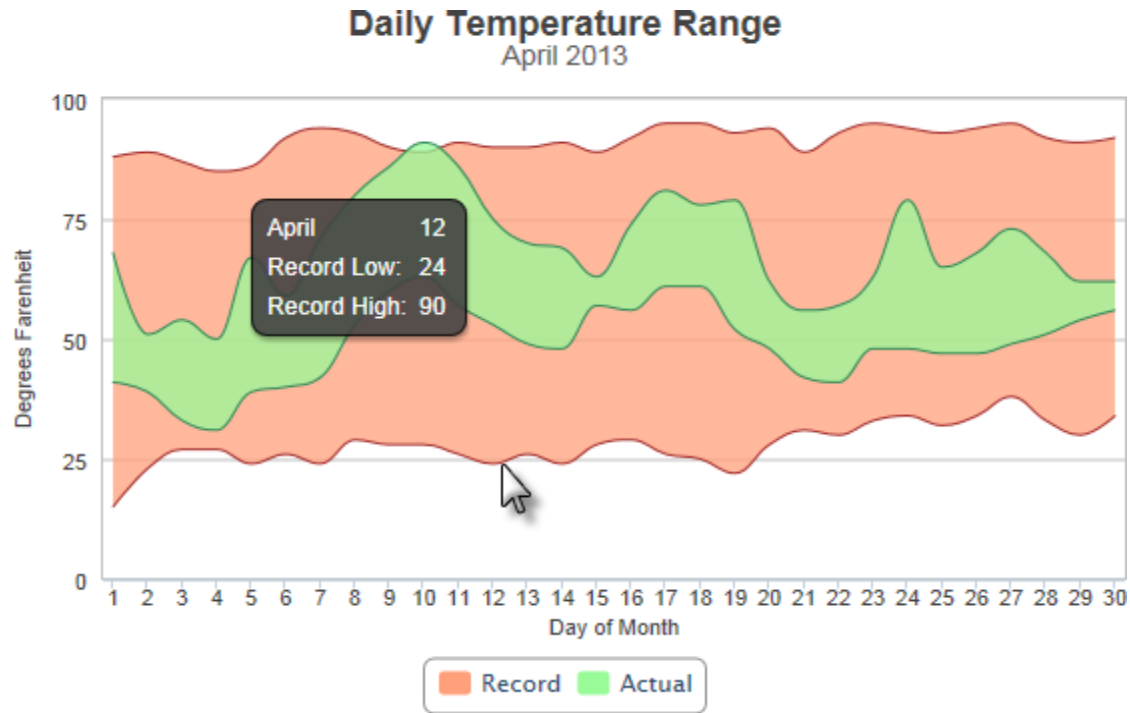


| Element - ChartXAxis | |
|----------------------------|------------------|
| Optional Attributes | |
| Axis Padding Left | |
| Axis Padding Right | |
| ... | |
| Caption | Day of the Month |
| ... | |
| Spacing | |

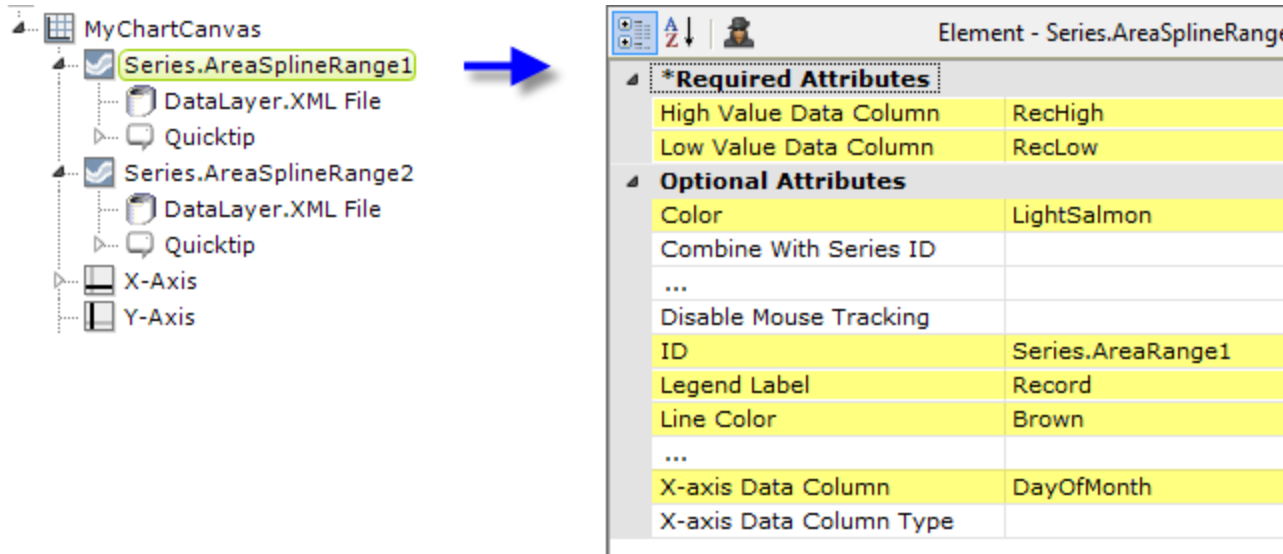
The properties of the X- and Y-axis, including captions, are set using the **X-Axis** and **Y-Axis** elements. For example, in order to provide a caption for the X-axis, add an **X-Axis** element beneath Chart Canvas and set its **Caption** attribute, as shown above. Repeat for the Y-axis.

Series.Area Spline Range - Using Multiple Series

You can add additional series to the chart by adding additional Series elements:



The example above shows two Series, one for each set of temperature ranges. A legend can also be added very easily.



The image shows a tree view of a chart canvas on the left and a detailed view of a series element on the right. A blue arrow points from the selected element in the tree to the detailed view.

MyChartCanvas


- Series.AreaSplineRange1
 - DataLayer.XML File
 - Quicktip
- Series.AreaSplineRange2
 - DataLayer.XML File
 - Quicktip
- X-Axis
- Y-Axis

Element - Series.AreaSplineRange

| *Required Attributes | |
|-------------------------|-------------------|
| High Value Data Column | RecHigh |
| Low Value Data Column | RecLow |
| Optional Attributes | |
| Color | LightSalmon |
| Combine With Series ID | |
| ... | |
| Disable Mouse Tracking | |
| ID | Series.AreaRange1 |
| Legend Label | Record |
| Line Color | Brown |
| ... | |
| X-axis Data Column | DayOfMonth |
| X-axis Data Column Type | |

The example above shows the two Series elements, their datalayers and optional Quicktips, and the X-Axis and Y-Axis elements used to produce the previous chart. You can adjust which data range appears "in front" of the other in the chart by changing the order of the Series elements in the definition. Setting the Series elements' **Legend Label** attribute will automatically cause the legend to be displayed.

Charts with multiple series or aggregations are encouraged to use the ChartCanvas element's attribute, `Tooltip Split`. This attribute allows you to split a chart's tooltip into one label per series, rather than per segment. The default value for this attribute is *False*. Once enabled, when hovering over a data point, it displays all the values for the series. This method is recommended over shared tooltips for charts with multiple line series, and as such, takes precedence over `tooltip.shared`.

 When using multiple series, you may be able to reduce the number of data queries and improve performance by using local data to read all of the data once, see *Datalayer Introduction*. Then, link its datalayer to share it to the series, see *Link Datalayers*.

At each Series element, link its datalayer to the shared Local Data datalayer and filter the data to meet the needs of each individual series.

You can combine different types of Series elements, for example, Series.Area and Series.Bar, to produce combinations of visualizations.

v23.1 If you are using the Chart Color Axis element in a mult-series chart, by default, the series will link to the first color axis. To link the series to a special color axis, set the ID attribute of the Chart Color Axis element to the corresponding Series' Linked to Color Axis ID attribute. For more information, see "Chart Color Axis" on page 124.

Series.Area Spline Range - Attributes

The Series.Area Spline Range element has the following attributes:

| Attribute | Description |
|--|---|
| High Value Data Column | (Required) Specifies the name of the datalayer column containing the high data value for each row. |
| Low Value Data Column | (Required) Specifies the name of the datalayer column containing the low data value for each row. |
| v23.1 Class Name | Specifies the Class Name referenced in the .css style sheet. The default value is <i>undefined</i> . |
| Color | Sets the data region fill color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| v23.1 Color Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the x-axis. |
| Combine With Series ID | Set this attribute to the element ID of another series to combine it with this series in the legend. When two series are combined, by default only the first one will appear in the legend but clicking the item in the legend toggles both series to appear and disappear. Or, the value <i>Previous</i> can be entered to combine this series with the previous series. |

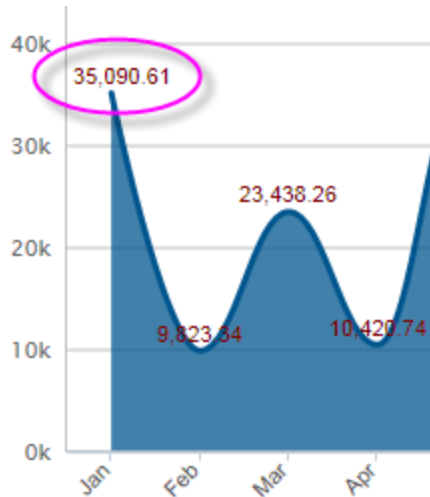
| Attribute | Description |
|-------------------------|--|
| Connect Nulls | Specifies if data rows with null or blank values are to be ignored, allowing adjacent values to be connected in the chart. The default value is <i>False</i> . |
| Disable Mouse Tracking | Disables mouse tracking for the series, when set to <i>True</i> . This affects tooltips and click events on graphs and points. For large datasets, this may improve performance. The default value is <i>False</i> . |
| Hover Line Thickness | Sets the thickness of the line, in pixels, when the mouse pointer is hovered over it. The default value is <i>1</i> pixel. |
| Legend Label | Indicates text that will be shown for this series inside the chart legend. When a value is provided, automatically causes the legend to be displayed. |
| Line Color | Sets the data region's border line color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| Line Color Transparency | Specifies the transparency of the data region border line color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| Line Style | Specifies the pattern of the data region's border line as either <i>Solid</i> or a combination of dashes and dots. |
| Line Thickness | Specifies the thickness of the data region's border line, in pixels. The default value is <i>1</i> pixel. |

| Attribute | Description |
|--|---|
| <p>v23.1</p> <p>Linked to Color-Axis ID</p> | <p>Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes</p> |
| <p>Linked to X-Axis ID</p> | <p>Specifies the ID of an X-Axis element that this series should be linked to when using multiple X-axes.</p> |
| <p>Linked to Y-Axis ID</p> | <p>Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes.</p> |
| <p>Negative Color</p> | <p>Sets the color for negative values. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #D5F484. The default value is <i>Red</i>.</p> |
| <p>Negative Color Transparency</p> | <p>Specifies the transparency of the Negative Color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other.</p> |
| <p>Negative Fill Color</p> | <p>Sets the data region fill color for negative values. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #D5F484. The default value is <i>Red</i>.</p> |
| <p>Negative Fill Color Transparency</p> | <p>Specifies the transparency of the Negative Fill Color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other.</p> |

| Attribute | Description |
|-------------------------|---|
| Negative Threshold | Sets the positive-negative value threshold, also called the "zero level" or "base level". The default value is <i>0</i> . |
| Transparency | Specifies the transparency of the data region fill color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| X-Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the X-axis. |
| X-Axis Data Column Type | <p>Specifies the data type of the datalayer column named in the X-axis Data Column attribute. Options include <i>Auto</i> (the default), <i>Text</i>, <i>Number</i>, and <i>DateTime</i>.</p> <p>By default, X-axis data values that are <i>DateTime</i> type will be automatically distributed evenly across the time series. If you want to disable this behavior, set this attribute value to <i>Text</i>.</p> |

Series.Area Spline Range - Using the Data Labels Element

A "data label" is text shown next to each data point that shows its value. When the **Data Labels** element is used as a child of Series.Area Spline Range, text representing the data values will appear on the chart:

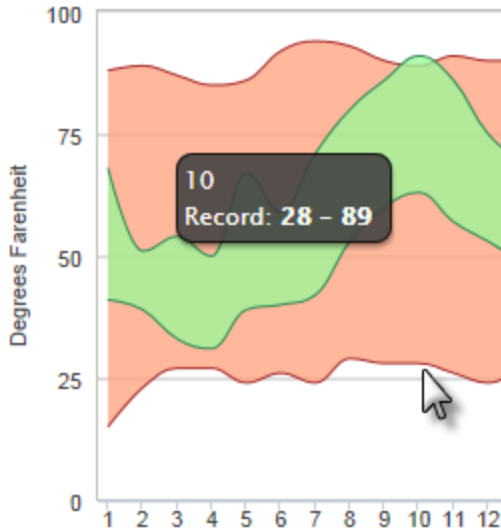


The Data Labels element has attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text. One attribute, Series Name as Caption, allows you to specify the y-axis label, x-axis label, or legend label as the data point value inside the chart.

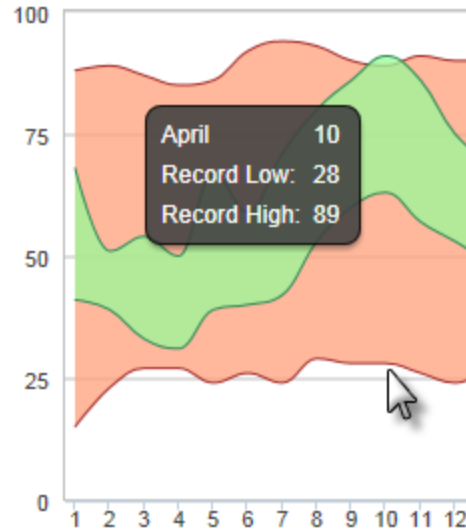
The Data Labels element's color-related attribute values can be set using @Chart tokens. **v23.1** Or, utilize the "Class Name" attribute to apply unique styling to your data labels by referencing a class name from a stylesheet in the report.

Series.Area Spline Range - Using the Quicktips Element

By default, a "quicktip" is displayed when the mouse hovers near or over a data point:

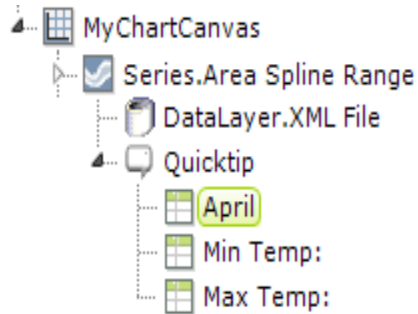


Default quicktip



With Quicktip child element


The automatically-generated quicktip displays information for the X- and Y-axis, as shown above, left. However, you may want to display other information or format it differently, as shown above, right, which can be done by adding a **Quicktip** child element beneath Series.Area Spline Range.



| Optional Attributes | |
|---------------------|--------------------|
| Caption | April |
| Format | |
| ID | |
| Value | @Chart.DayOfMonth~ |

The example above shows a Quicktip element (no attributes need to be set) and three **Quicktip Row** child elements, used to create the quicktip shown in the previous image. Use @Chart tokens to include chart data in the quicktip.

You can also configure the tooltip to display values for any available column in the datalayer. To do so, enter a token representing the data column in the Value attribute of the Quicktip Row element.

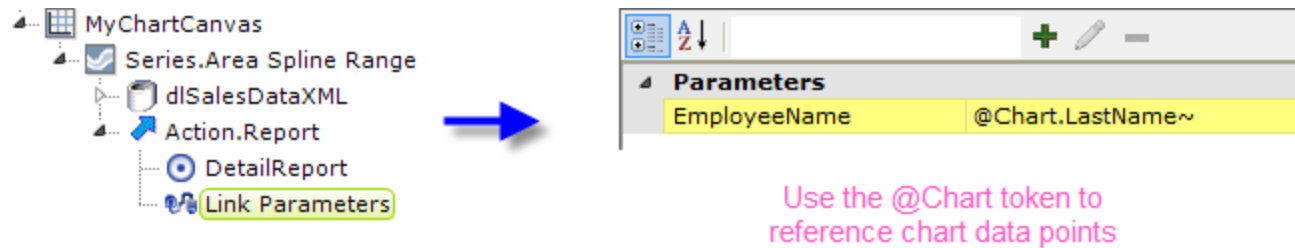
 To use this feature with DataLayer.ActiveSQL, please make sure the keep Grouped Rows attribute of the SqlGroup element is set to *False*.

Intrinsic functions are supported in the Quicktip attributes.

The Quicktip Row element has been made context-sensitive with the addition of a **Condition** attribute.

Series.Area Spline Range - Using Action Elements

Action elements initiate processing of a report or process task definition, redirection to a link, or other processing when a data point in the series is clicked.

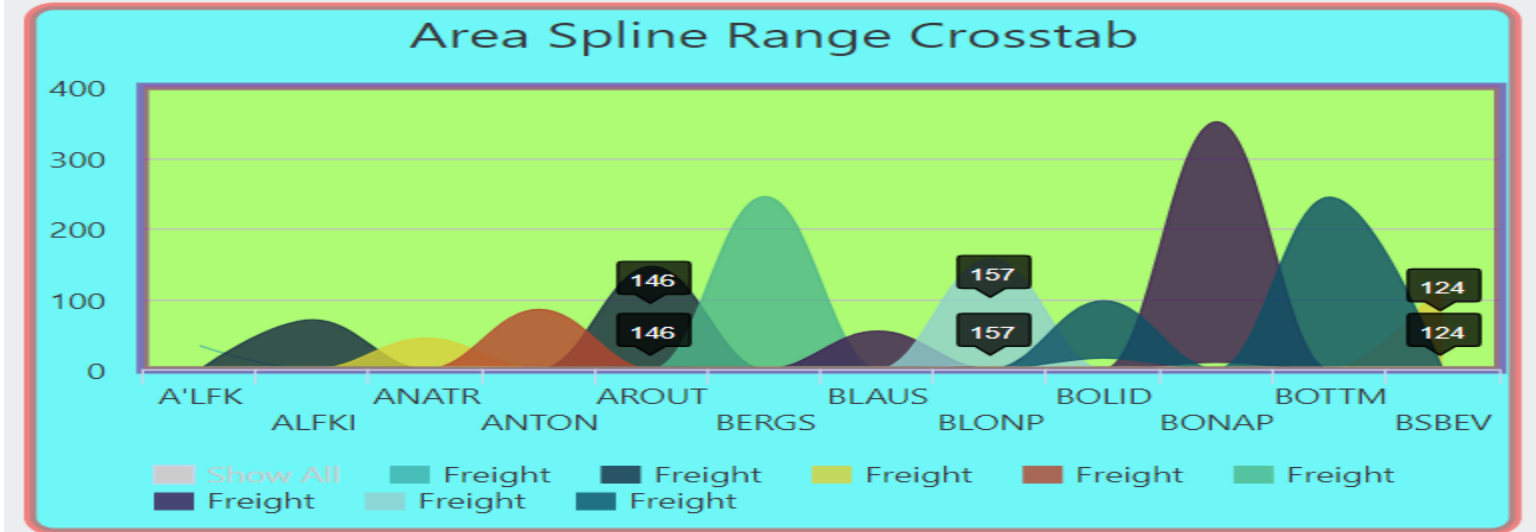
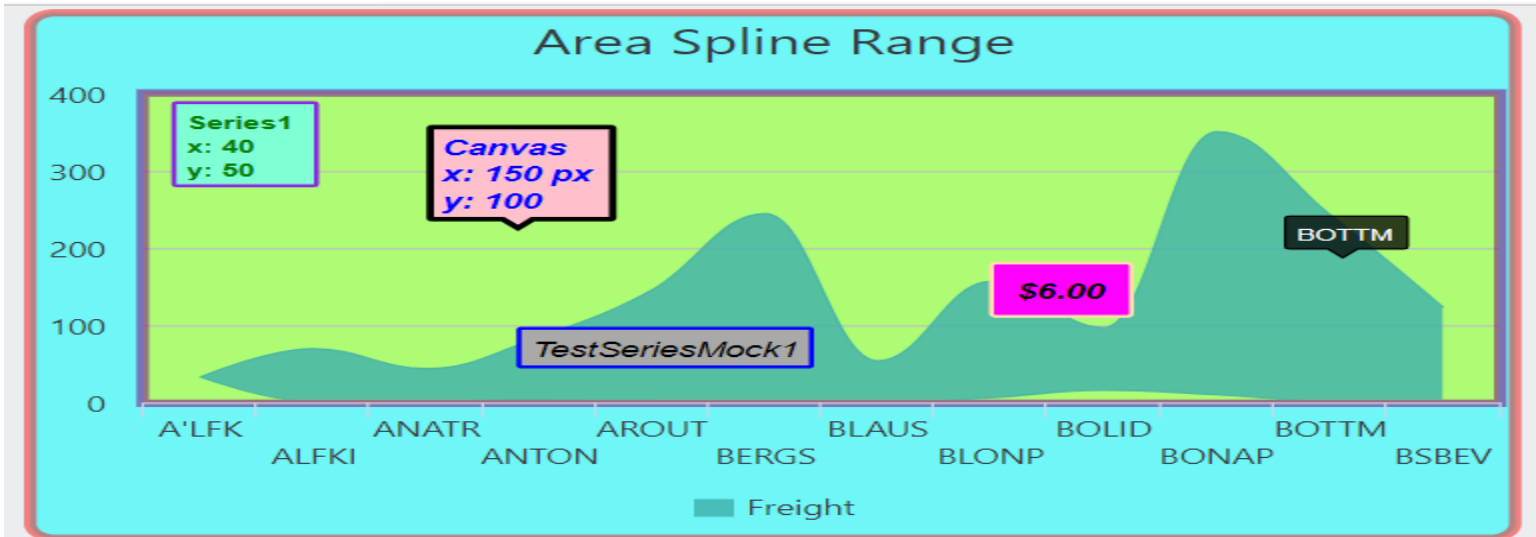


In the example above, an **Action.Report** element has been added as a child of the Series, along with its **Target.Report** and **Link Parameters** child elements. To reference chart data in parameters, use the @Chart token, as shown above.

A variety of Action elements are available for use with Series, including Action.Link, Action.Process, and Action.Refresh Element. Additional Action elements will be added in future releases.

v23.1 Series.Area Spline Range - Using the Series Annotation Element

Annotations allow you to annotate a chart freely using custom labels and shapes. When the **Series Annotation** element is used as a child of Series.Area Spline Range, you can place these annotations at various points of interest:



The Series Annotation element offers two child elements: **AnnotationLabel.Point** and **AnnotationLabel.Mock**.

The `AnnotationLabel.Point` child element is used to create an annotation label and position on the chart by linking it to an existing point. With this element, you cannot specify the value, it always refers to the value of the point. Additionally, you can use the `Caption` and `Condition` attributes to refer to different datalayers.

On the other hand, the `AnnotationLabel.Mock` is used to create an annotation label and position it on the chart by linking it to a created mock point. With this element, you can specify the coordinate, relative position, and axis.

Both the `AnnotationLabel.Point` and `AnnotationLabel.Mock` have attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text.

Series.Area Spline Range - Using Input Selection

This series can also be used with the **Input Selection** family of elements, which turn the chart into an input control. This allows users, at runtime, to select points or values on the chart with their mouse and your report can then take action using the selected data. This is very useful, for example, for drilling into the data.

For more information about this functionality, see "Input Selection" on page 97.

Series.Area Spline Range - Using the Refresh Series Timer

The **Refresh Series Timer**, added as a Series child element, updates charts automatically, based on a time interval. When the interval is reached, a request is sent back to the web server for updated data. Series are updated with a smooth animation.



The **Refresh Interval** attribute specifies the number of seconds to wait before refreshing the series data. A value is required and must be an integer greater than 0.

Refresh Mode

Series may be refreshed in two ways: either all of the data is refreshed with each request, or the series data automatically slides to the left one data interval per refresh. The refresh mode is selected in the Chart X Axis element's **Axis Type** attribute.

When the Axis Type is *not* set to *DateTimeLinear*, all of the data will be refreshed.

When the Axis Type value is *DateTimeLinear*, newer values are added to the right side of the chart, previous values slide left, and older values disappear as they reach the left edge of the chart. In this case, the Refresh Series Timer element's **Time Span** attribute is used to specify the age of the data included. This value must be in *hh:mm:ss* format (for example "00:02:00" for two minutes) and must be larger than the Refresh Interval attribute value.

When a Time Span attribute is set, the series' datalayer can make use of a special timestamp token, `@Chart.rdTimeSpanStart~`, in

a query to retrieve just the rows necessary to update the chart. On the initial request, this token returns the current time minus the Time Span value. For subsequent refreshes, when *Axis Type = DateTimeLinear* and the X-axis will be sliding, the token returns the last time the chart was updated, so only the newest rows are retrieved. Here's an MS SQL Server query example:

```
SELECT * FROM MyTable WHERE UpdateTime BETWEEN '@Chart.rdTimeSpanStart~' AND '@Function.DateTime~'
```

For best performance, avoid setting a very short refresh interval if a very large number of users will be displaying the report.

Series.Bar

The Chart Canvas element's Series child elements cause a data visualization (the chart) to be rendered in the canvas.

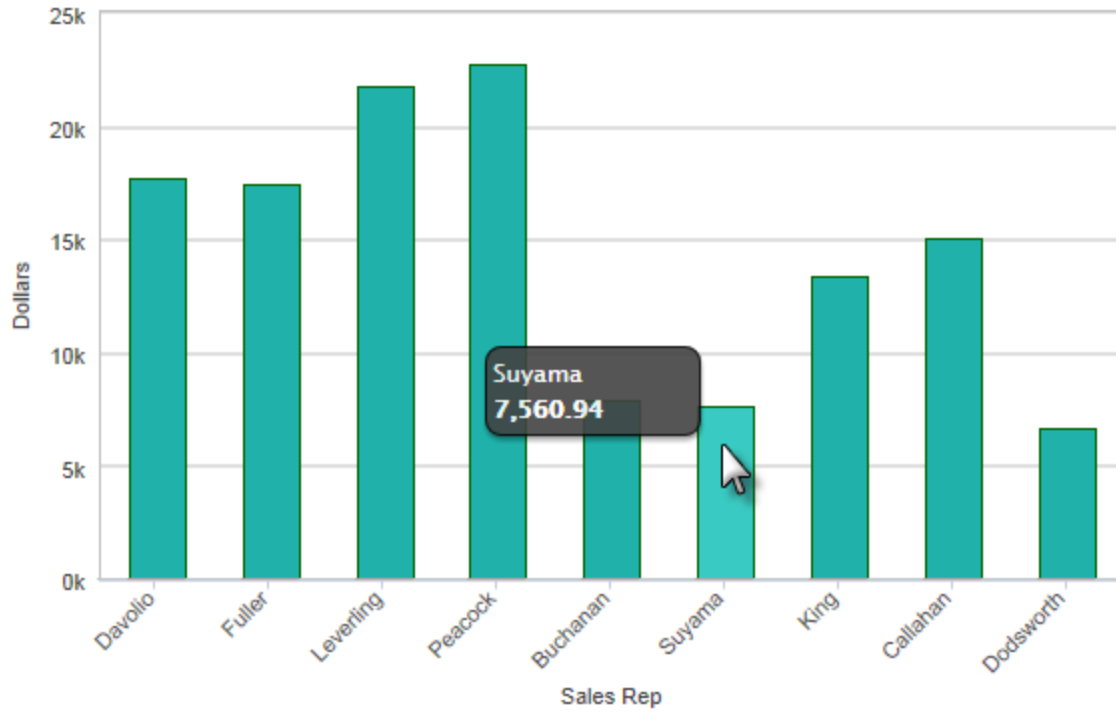
The following topics discuss the Series.Bar child element:

- [Using Multiple Series](#)
- [Series.Bar Attributes](#)
- [Using the Data Labels Element](#)
- [Using the Quicktips Element](#)
- [Using the Chart Drill To Element](#)
- [Using the Trend Line Element](#)
- [Using Action Elements](#)
- [Using the Series Annotation Element](#) v23.1
- [Using Input Selection](#)
- [Using the Refresh Series Timer](#)

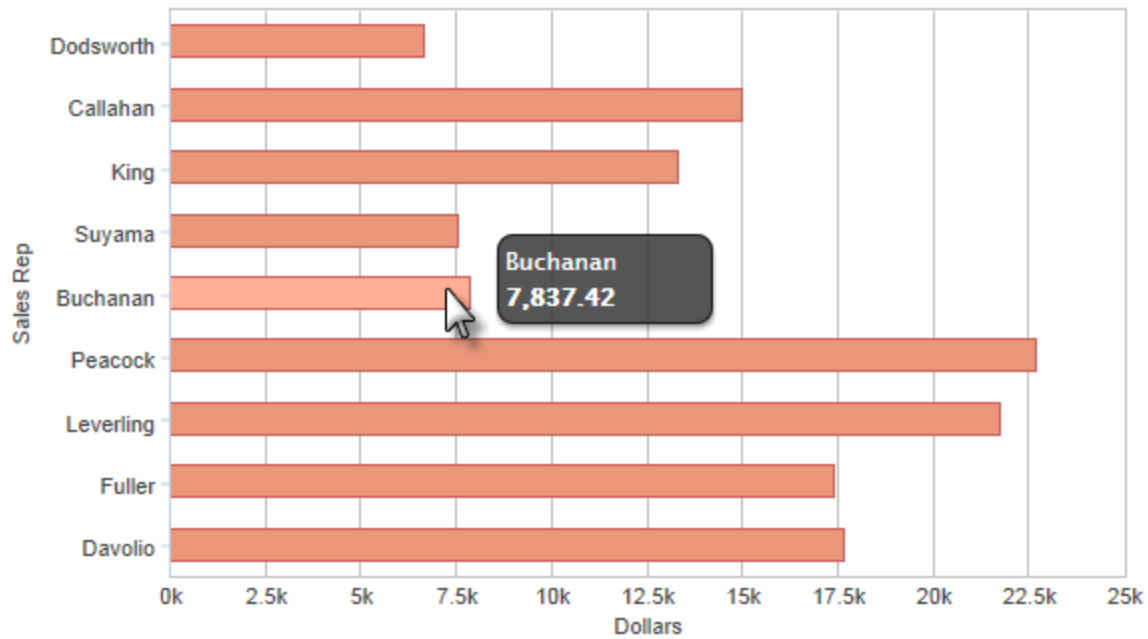
About Series.Bar

The **Series.Bar** element generates a Bar chart, which uses either vertical or horizontal bars to show comparisons among categories.

Monthly Sales April 2013



Monthly Sales April 2013




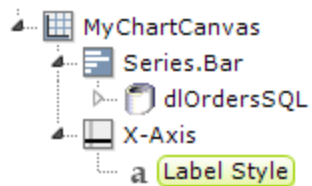
The example above shows a simple Bar chart, representing sales per month by sales rep, in both vertical and horizontal orientations.



| Element - Series.Bar | |
|-----------------------------|------------|
| *Required Attributes | |
| Y-axis Data Column | sumSales |
| Optional Attributes | |
| Bar Border Color | IndianRed |
| ... | |
| Color | DarkSalmon |
| Combine With Series ID | |
| ... | |
| X-axis Data Column | LastName |
| X-axis Data Column Type | |

As shown above, the chart is created by adding Series.Bar to the canvas, along with a datalayer and, typically, some child elements that may include a **Group Filter**, and a **Group Aggregate Column** element. Very few attributes need to be set for the Series element in order to produce a basic chart.

 A datalayer element can be used either beneath Series.Bar, as shown above, or beneath Chart Canvas. If used as a child of Chart Canvas, its data is available to *all* child Series elements. This can improve performance if you have several series, all using the *same* data, beneath the same Chart Canvas element.

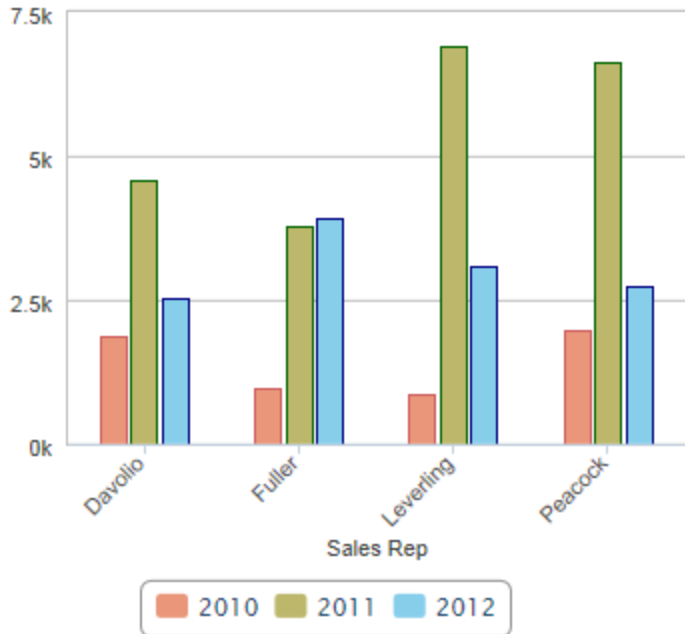


| Element - AxisLabelStyle | |
|----------------------------|----|
| Optional Attributes | |
| Font Angle | 45 |
| Font Color | |
| ... | |
| Stagger Labels | |

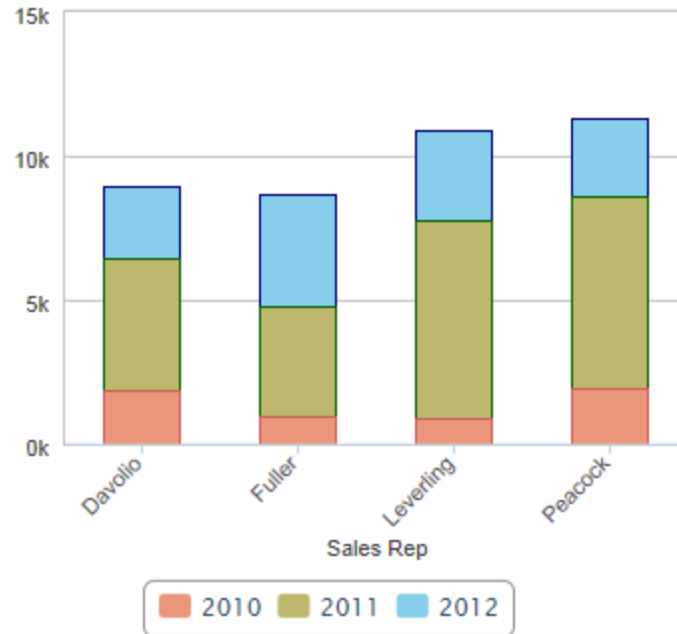
The properties of the X- and Y-axis, including captions, are set using the **X-Axis** and **Y-Axis** elements. For example, in order to angle the X-axis labels, add an X-Axis element beneath Chart Canvas (none of its attributes need to be set) and add its child **Label Style** element. Set the Label Style element's attribute as shown above.

Series.Bar - Using Multiple Series

You can add additional series to the chart by adding additional Series elements:



Stacking = SidebySide

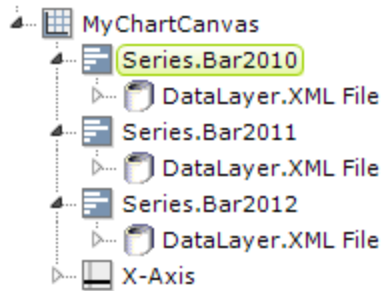


Stacking = Stacked

+

The example above uses three Series, one for each year, and two different "stacking modes" are shown, along with a legend. *Overlay* and *StackedPercentage* stacking modes are also available. Combinations of stacking modes are also possible, for example, two years stacked, and one year beside them.

💡 The scaling of the data axis may change depending on the stacking mode.



| Element - Series.Bar | |
|-----------------------------|----------------|
| *Required Attributes | |
| Y-axis Data Column | sumSales |
| Optional Attributes | |
| Bar Border Color | IndianRed |
| ... | |
| Color | DarkSalmon |
| Combine With Series ID | |
| ... | |
| ID | Series.Bar2010 |
| Legend Label | 2010 |
| ... | |
| Stacking | SideBySide |
| Transparency | |
| X-axis Data Column | LastName |
| X-axis Data Column Type | |

The example above shows the two Series elements, their datalayers, and an X-Axis element used to produce the previous chart. You can adjust the order of the series in the chart and the legend by changing the order of the Series elements in the definition. Setting the Series elements' **Legend Label** attribute will automatically cause the legend to be displayed.

Charts with multiple series or aggregations are encouraged to use the ChartCanvas element's attribute, `Tooltip Split`. This attribute allows you to split a chart's tooltip into one label per series, rather than per segment. The default value for this attribute is *False*. Once enabled, when hovering over a data point, it displays all the values for the series. This method is recommended over shared tooltips for charts with multiple line series, and as such, takes precedence over `tooltip.shared`.



When using multiple series, you may be able to reduce the number of data queries and improve performance by using local data to read all of the data once, see *Datalayer Introduction*. Then, link its datalayer to share it to the series, see *Link Datalayers*. At each Series element, link its datalayer to the shared Local Data datalayer and filter the data to meet the needs of each individual series.

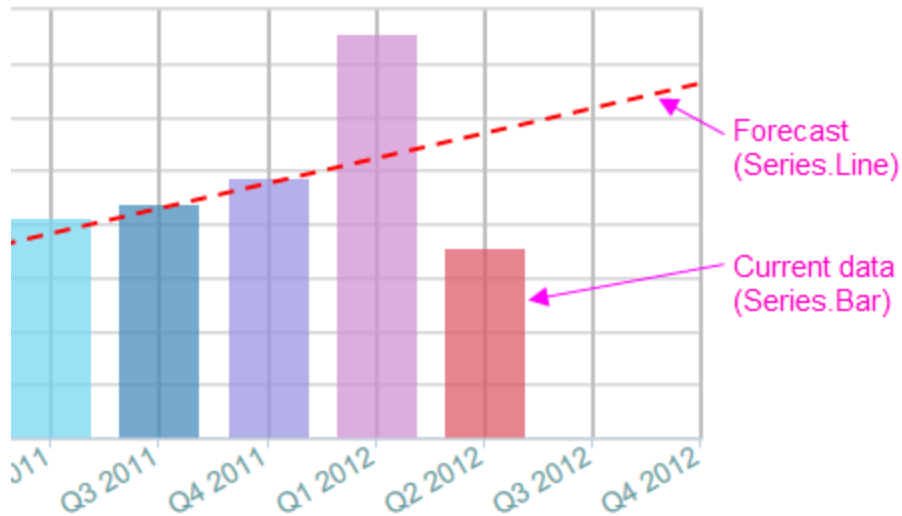
You can combine different types of Series elements, for example, Series.Bar and Series.Line, to produce combinations of visualizations. A good example of this is when you wish to use forecasting.

v23.1 If you are using the Chart Color Axis element in a multi-series chart, by default, the series will link to the first color axis.

To link the series to a special color axis, set the ID attribute of the Chart Color Axis element to the corresponding Series' Linked to Color Axis ID attribute. For more information, see "Chart Color Axis" on page 124.

Forecasting

Forecasting elements use a variety of techniques to produce projected values by analyzing existing values. The future values they "predict" are, in most cases, added as rows or columns to a datalayer so the data can be displayed along with the existing data. When using Chart Canvas charts, the forecast data is typically displayed using a Series.Line element, in conjunction with other series elements.



Forecasting elements add a "forecast value" column to the datalayer, and this column is used as the series' Y-axis data column. For more information about using forecasting elements, see *The Forecasting Elements*.


Series.Bar - Attributes

The Series.Bar element has the following attributes:

| Attribute | Description |
|---|---|
| Y-Axis Data Column | (Required) Specifies the name of a datalayer column whose values will be plotted along the Y-axis. |
| Bar Border Color | Sets the color of the thin border line around each bar. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| Bar Border Color Transparency | Specifies the transparency of the thin border line around each bar. The lowest value of 0 specifies that the background is opaque, with no transparency. At the other end of the scale, 15 specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| Bar Border Radius | Sets the amount of rounding for bar corners, in pixels. The default value is 0 pixels, which produces square corners. |
| Bar Border Thickness | Sets the thickness of the bar border lines, in pixels, when the related Bar Border Color attribute has a value. The default value is 1 pixel. |
| Bar Thickness | Sets the width of the bar in pixels. If left blank, the width will be determined automatically. |
| v23.1 Class | Specifies the Class Name referenced in the .css style sheet. The default value is <i>undefined</i> . |

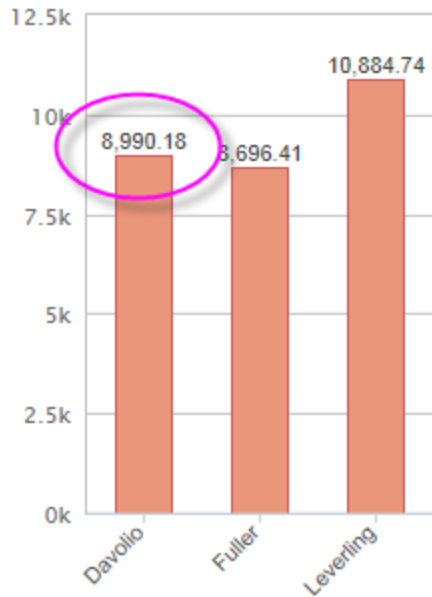
| Attribute | Description |
|--|---|
| Name | |
| Color | Sets the bar fill color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| v23.1 Color Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the x-axis. |
| Combine With Series ID | Set this attribute to the element ID of another series to combine it with this series in the legend. When two series are combined, by default only the first one will appear in the legend but clicking the item in the legend toggles both series to appear and disappear. Or, the value <i>Previous</i> can be entered to combine this series with the previous series. |
| Disable Mouse Tracking | Disables mouse tracking for the series, when set to <i>True</i> . This affects tooltips and click events on graphs and points. For large datasets, this may improve performance. The default value is <i>False</i> . |
| Hover Bright- ness | Specifies the amount to change a bar's color when the mouse pointer is hovered over it. Values can be between 0 (no change) and 15(lighter). The default value is 2. |
| Legend Label | Indicates text that will be shown for this series inside the chart legend. When a value is provided, it automatically causes the legend to be displayed. |
| v23.1 | Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes |

| Attribute | Description |
|-----------------------------|--|
| Linked to Color-Axis ID | |
| Linked to X-Axis ID | Specifies the ID of an X-Axis element that this series should be linked to when using multiple X-axes. |
| Linked to Y-Axis ID | Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes. |
| Negative Color | Sets the color for negative values. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #D5F484. The default value is <i>Red</i> . |
| Negative Color Transparency | Specifies the transparency of the Negative Color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| Negative Threshold | Sets the positive-negative value threshold, also called the "zero level" or "base level". The default value is <i>0</i> . |
| Stack Group Name | Enables the grouping of multiple series in a stacked chart. Enter the same arbitrary string value in this attribute for each series to be stacked. |
| Stacking | Sets the bar stacking mode. <i>SideBySide</i> arranges the bars beside each other, <i>Stacked</i> stacks the series one above the other in a single bar, while <i>StackedPercentage</i> stacks the series, then raises the levels to 100 percent, showing the percentage for each series' value. <i>Overlay</i> draws the bars on top of each other (use |

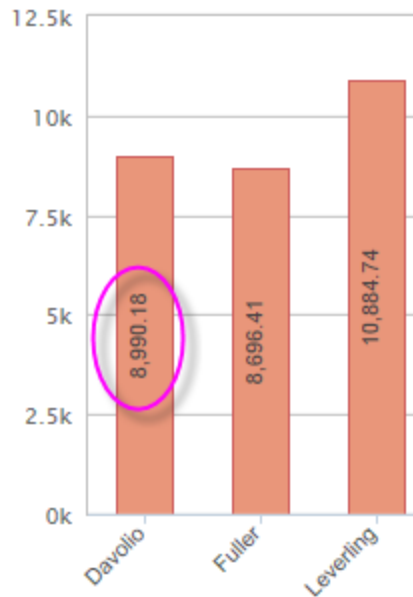
| Attribute | Description |
|-------------------------|---|
| | <p>Transparency to let them show through). All series don't have to use the same stacking mode - they can be mixed to provide hybrid arrangements, such as two stacked and one beside them. Leave blank for no stacking.</p> |
| Transparency | <p>Specifies the transparency of the data region fill color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other.</p> |
| X-Axis Data Column | <p>Specifies the name of a datalayer column whose values will be plotted along the X-axis.</p> |
| X-Axis Data Column Type | <p>Specifies the data type of the datalayer column named in the X-axis Data Column attribute. Options include <i>Auto</i> (the default), <i>Text</i>, <i>Number</i>, and <i>DateTime</i>. By default, X-axis data values that are <i>DateTime</i> type will be automatically distributed evenly across the time series. If you want to disable this behavior, set this attribute value to <i>Text</i>.  In general, when working with a Bar chart, you'll typically want to reference your X-axis column as <i>Text</i> data. Bar Charts do not depict any correlation from one group of bars to the next. However, when applying a data type of <i>Number</i> or <i>DateTime</i>, the chart attempts to establish a correlation between the points on the X-axis. Since there is only a single point per bar, it improperly displays the bars. Since a Bar Chart does not have a correlation between points, using <i>DateTime</i> data as text will not impact your results and you can still format the X-axis to meet your desired layout.</p> |

Series.Bar - Using the Data Labels Element

A "data label" is text shown near each data point that shows its value. When the **Data Labels** element is used as a child of Series.Bar, text representing the data values will appear on the chart:



Default data labels



Font Angle = 90
Label Placement = Inside

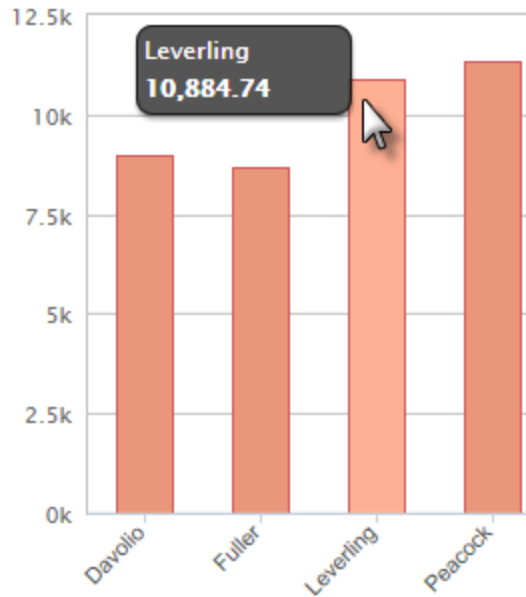
By default, with a single data series, the data labels will appear outside the bar, as shown above, left. The example above, right, shows data labels placed inside the bars and rotated 90-degrees. When series stacking is being used, the default is for data labels to appear *inside* the bars. The Data Labels element has attributes that allow you to control the font family, color, size, and weight,

the data format, border color, and positioning of the text. One attribute, Series Name as Caption, allows you to specify the y-axis label, x-axis label, or legend label as the data point value inside the chart.

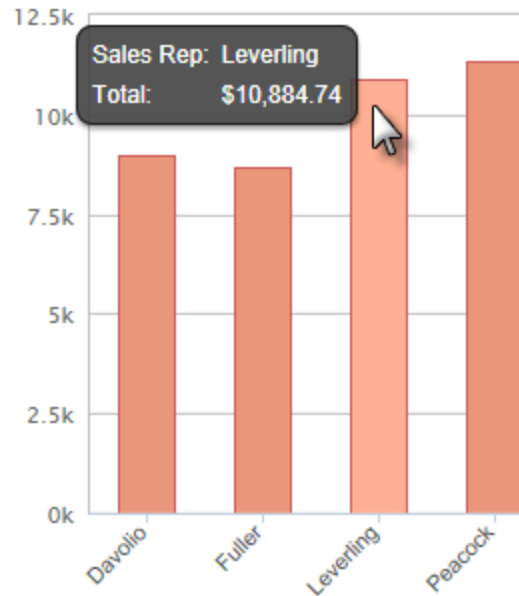
The Data Labels element's color-related attribute values can be set using @Chart tokens. v23.1 Or, utilize the "Class Name" attribute to apply unique styling to your data labels by referencing a class name from a stylesheet in the report.

Series.Bar - Using the Quicktips Element

By default, a "quicktip" is displayed when the mouse hovers near or over a data point:



Default quicktip

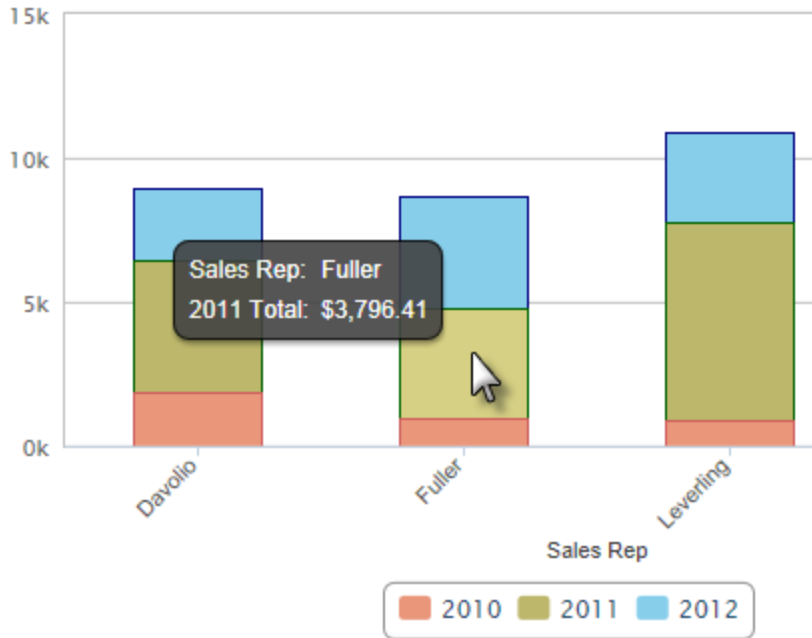


With Quicktip child element

The automatically-generated quicktip displays information for the X- and Y-axis, as shown above, left. However, you may want to display other information or format it differently, perhaps as shown above, right, which can be done by adding a **Quicktip** child element beneath Series.Bar and setting its attributes and child elements. Use @Chart tokens to include chart data in the quicktip.

You can also configure the tooltip to display values for any available column in the datalayer. To do so, enter a token representing the data column in the Value attribute of the Quicktip Row element.

💡 To use this feature with `DataLayer.ActiveSQL`, please make sure the `keep Grouped Rows` attribute of the `SqlGroup` element is set to `False`.




When using stacked series, if each series has its own Quicktips child element, the data shown in the quicktip will vary depending on which stacked segment the cursor hovers over, as shown above. Intrinsic functions are supported in the Quicktip attributes.

To display a percentage for each bar in a Stacked Bar Chart, use the special token `@Chart.rdStackedChartPercentage~`, like below:

| Optional Attributes | |
|---------------------|----------------------------------|
| Description | @Chart.rdStackedChartPercentage- |
| ID | |
| Security Right ID | |
| Title | ValueB |

This feature enables the user to hover over the stacked bars and show the percentage for each value. The Title attribute can be used to distinguish between values in a stacked chart.

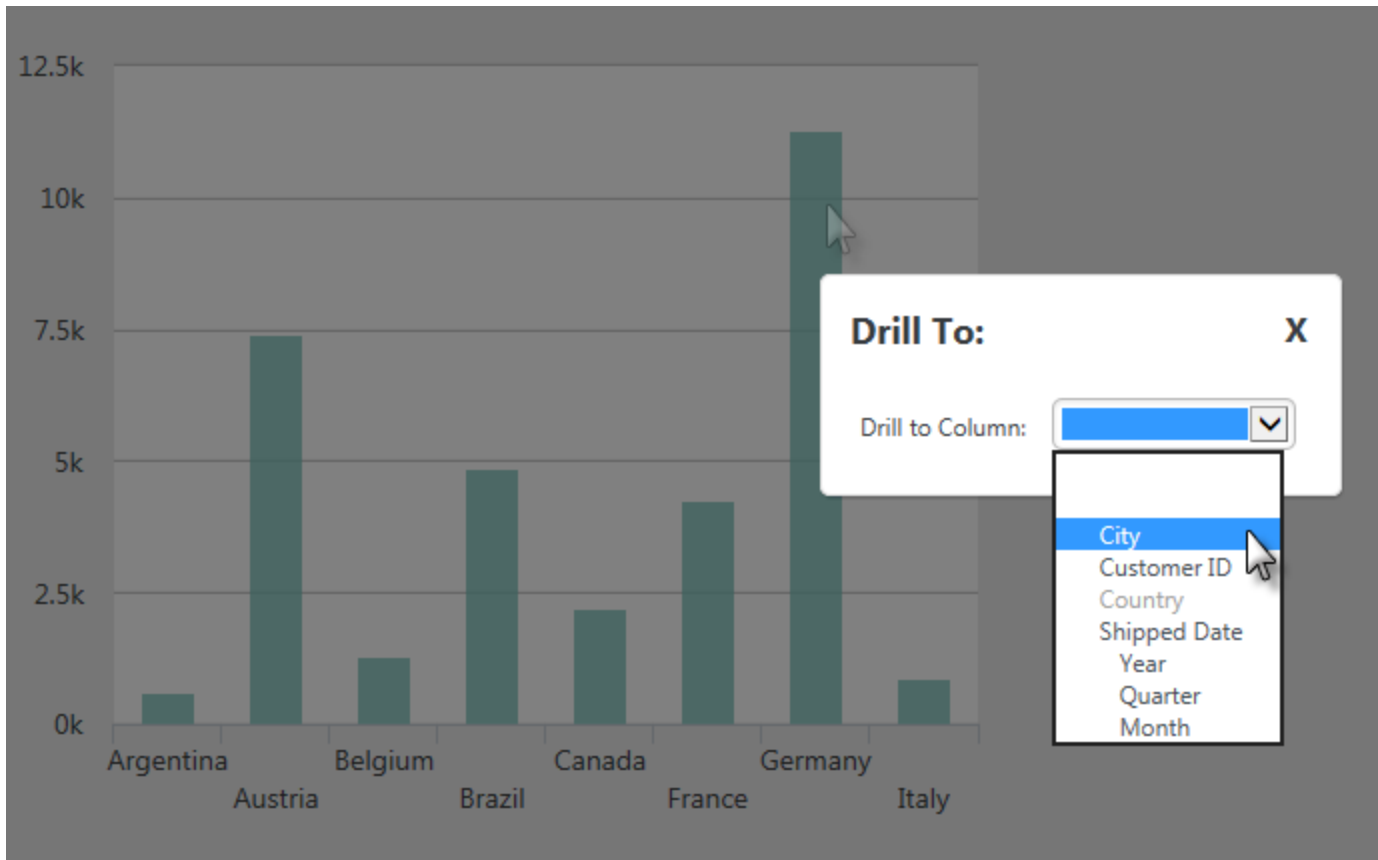
 The format of the percentage will display according to the Format property in QuickTipRow. If this format is not set, it will display the default format "Percent" (0.000%).

Series.Bar - Using the Chart Drill To Element

The **Chart Drill To** element, a child of the Series element, enhances charts by allowing you to examine the data "behind" the chart.

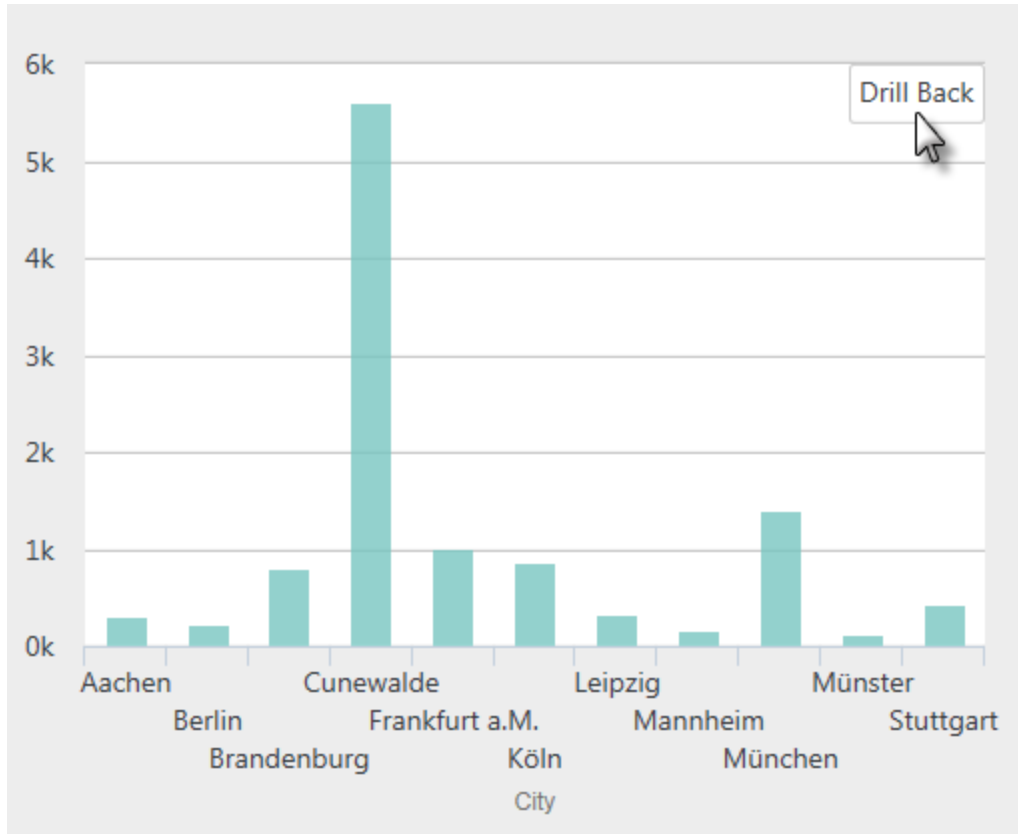
 The parent Chart Canvas element must have an ID attribute value in order to use the Chart Drill To element with it.


When the Drill To element is enabled, selecting a chart bar displays a list of columns, as shown below:



Selecting a column re-draws the chart so that only the data representing that specific bar is shown, and the X-axis changes to the selected Drill To column.

Select **Drill Back** to return the chart to it's previous state:



 Chart Canvas Charts that use the Chart Drill To element can only have a single Series. The Drill Back button only appears when the mouse cursor is hovered over the upper right-hand corner of the chart.

Or, continue drilling further down into the data (assuming the elements are configured for it). For example, you could select the **Munchen** chart bar next and then select **Customer ID** to drill down to the customers in that city.

If your chart has been configured for it, as you drill further into the data, a breadcrumb trail becomes available. The breadcrumb trail can be used to access previously drilled levels, or clear the Drill To data altogether. To enable this feature, set the **Show DrillTo Breadcrumb** attribute to *True*.



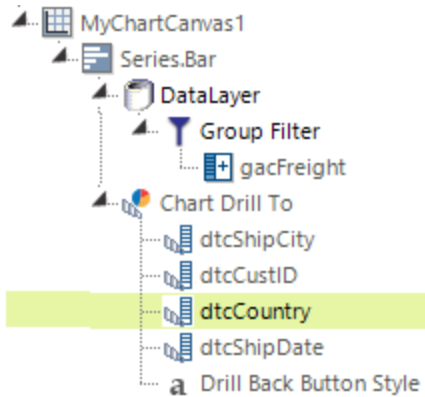
As shown below, the Chart Drill To element drills into *grouped* data, so the datalayer used beneath the Chart Canvas Chart or Series element *must* be grouped using a **Group Filter** or **Sql Group** element. The example datalayer has been grouped on the *ShipCountry* column and the Keep Grouped Rows attribute was set to *False*. The Series itself has been configured so that its X-axis Data Column = *ShipCountry* and its Y-axis Data Column = *gacFreight* (the Group Aggregate Column that sums the Freight values).

The image shows a screenshot of the Logi Analytics interface. On the left, a tree view displays the hierarchy: MyChartCanvas1 (containing Series.Bar, which contains DataLayer, which contains Group Filter, which contains gacFreight). A blue arrow points from the Group Filter element to a configuration table on the right.

The configuration table is titled "Element - GroupFilter" and contains the following attributes:

| *Required Attributes | |
|----------------------|-------------|
| Group Column | ShipCountry |
| Optional Attributes | |
| Data Type | |
| Hierarchical | |
| ID | |
| Include Condition | |
| Keep Grouped Rows | False |
| Sort Sequence | |

Next, a **Chart Drill To** element was added beneath the Series element, shown below. Required child **Drill To Column** elements have also been added; they define the columns the user can select to drill into. They should be added and configured for columns that can be reasonably grouped, such as text-type columns with a limited number of unique values and date-type columns.



| *Required Attributes | |
|----------------------|-------------|
| Column Header | Country |
| Data Column | ShipCountry |
| Data Type | Text |
| ID | dtcCountry |
| *Optional Attributes | |
| Format | |
| Security Right ID | |



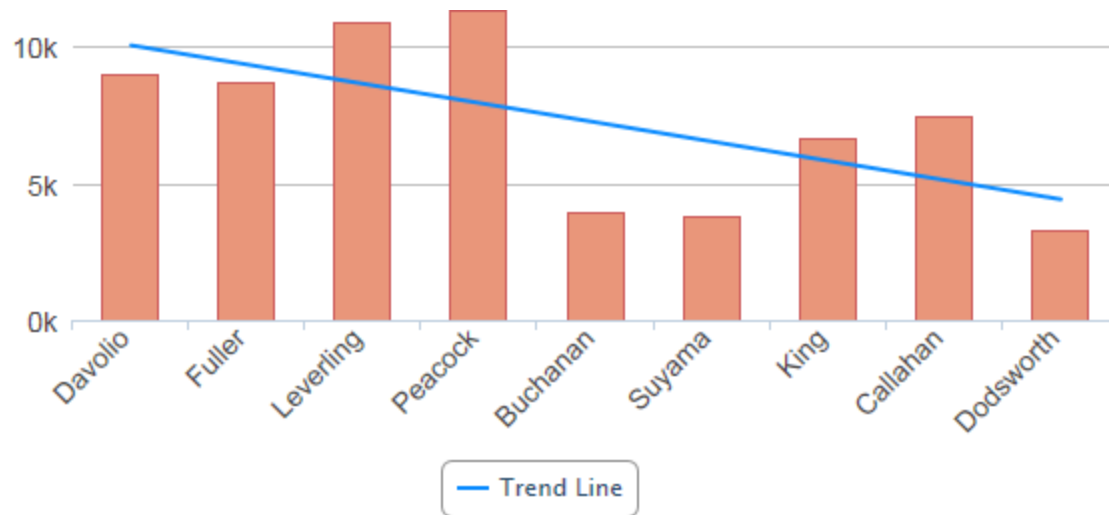
You *must* also configure a Drill To Column element for the column that's specified as the chart's X-axis Data Column, in this example the *ShipCountry* column.

The **Column Header** attribute specifies the text that will appear in the Drill to Column drop-down list options and will become the drilled chart's X-axis caption. The vertical order of the drop-down list options will match the vertical order of the Drill To Column elements in the definition. Year, Quarter, and Month options will be added automatically for Drill To Columns with Data Type = *Date*. You do not need to add any filter or additional grouping elements to achieve the results shown in the example above.

The Drill Back button displayed on the chart after drilling has occurred can be styled using the **Drill Back Button Style** element.

Series.Bar - Using the Trend Line Element

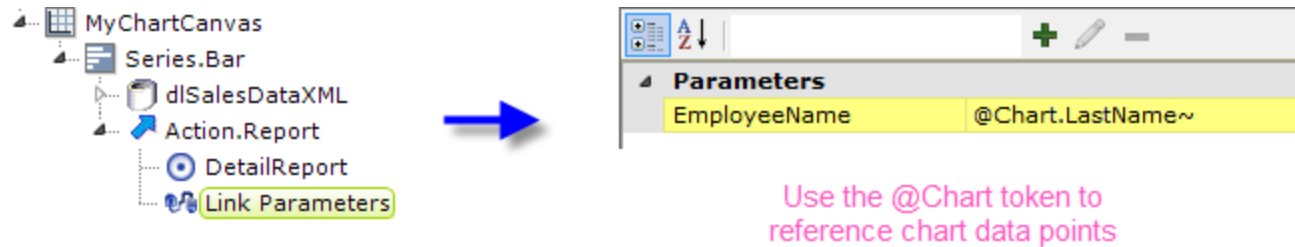
The **Trend Line** element creates a line on the chart that indicates the "trend" of the data. The line connects a number of data points generated using a regression algorithm.



The Trend Line element is a child of the Series.Bar element and can be styled for color, line width, etc. When configured with a legend label, it will be represented by an item in the legend, as shown above.

Series.Bar - Using Action Elements

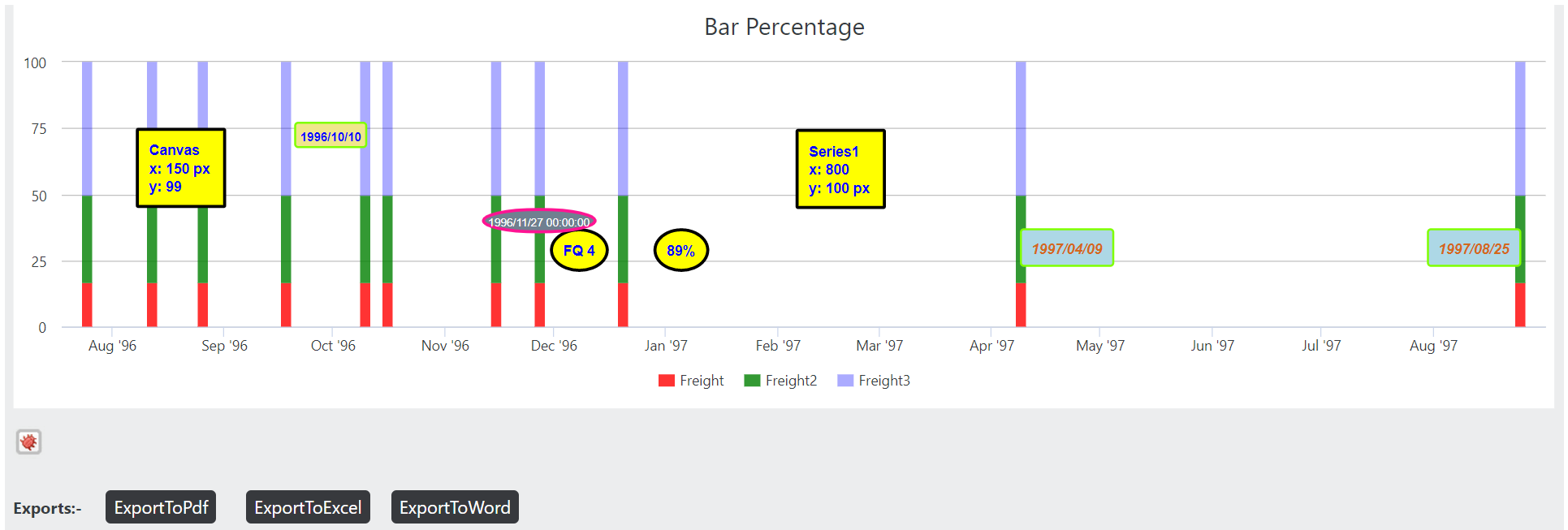
Action elements initiate processing of a report or process task definition, redirection to a link, or other processing when a data point in the series is clicked.



In the example above, an **Action.Report** element has been added as a child of the Series, along with its **Target.Report** and **Link Parameters** child elements. To reference chart data in parameters, use the @Chart token, as shown above. A variety of Action elements are available for use with Series, including Action.Link, Action.Process, and Action.Refresh Element. Additional Action elements will be added in future releases.

Series.Bar - Using the Series Annotation Element

Annotations allow you to annotate a chart freely using custom labels and shapes. When the **Series Annotation** element is used as a child of Series.Bar, you can place these annotations at various points of interest:



The Series Annotation element offers two child elements: **AnnotationLabel.Point** and **AnnotationLabel.Mock**.

The AnnotationLabel.Point child element is used to create an annotation label and position on the chart by linking it to an existing point. With this element, you cannot specify the value, it always refers to the value of the point. Additionally, you can use the Caption and Condition attributes to refer to different datalayers.

On the other hand, the `AnnotationLabel.Mock` is used to create an annotation label and position it on the chart by linking it to a created mock point. With this element, you can specify the coordinate, relative position, and axis.

Both the `AnnotationLabel.Point` and `AnnotationLabel.Mock` have attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text.

Series.Bar - Using Input Selection

This series can also be used with the **Input Selection** family of elements, which turn the chart into an input control. This allows users, at runtime, to select points or values on the chart with their mouse and your report can then take action using the selected data. This is very useful, for example, for drilling into the data.

For more information about this functionality, see [y](#).

Series.Bar - Using the Refresh Series Timer

The **Refresh Series Timer**, added as a Series child element, updates charts automatically, based on a time interval. When the interval is reached, a request is sent back to the web server for updated data. Series are updated with a smooth animation.



The **Refresh Interval** attribute specifies the number of seconds to wait before refreshing the series data. A value is required and must be an integer greater than 0.

Refresh Mode

Series may be refreshed in two ways: either all of the data is refreshed with each request, or the series data automatically slides to the left one data interval per refresh. The refresh mode is selected in the Chart X Axis element's **Axis Type** attribute. When the *Axis Type* is *not* set to *DateTimeLinear*, all of the data will be refreshed.

When the *Axis Type* value is *DateTimeLinear*, newer values are added to the right side of the chart, previous values slide left, and older values disappear as they reach the left edge of the chart. In this case, the Refresh Series Timer element's **Time Span** attribute is used to specify the age of the data included. This value must be in *hh:mm:ss* format (for example "00:02:00" for two minutes) and must be larger than the Refresh Interval attribute value.

When a Time Span attribute is set, the series' datalayer can make use of a special timestamp token, `@Chart.rdTimeSpanStart~`, in a query to retrieve just the rows necessary to update the chart. On the initial request, this token returns the current time minus the Time Span value. For subsequent refreshes, when *Axis Type = DateTimeLinear* and the X-axis will be sliding, the token returns the last time the chart was updated, so only the newest rows are retrieved. Here's an MS SQL Server query example:

```
SELECT * FROM MyTable WHERE UpdateTime BETWEEN '@Chart.rdTimeSpanStart~' AND '@Function.DateTime~'
```

For best performance, avoid setting a very short refresh interval if a very large number of users will be displaying the report.

Series.Bar Range

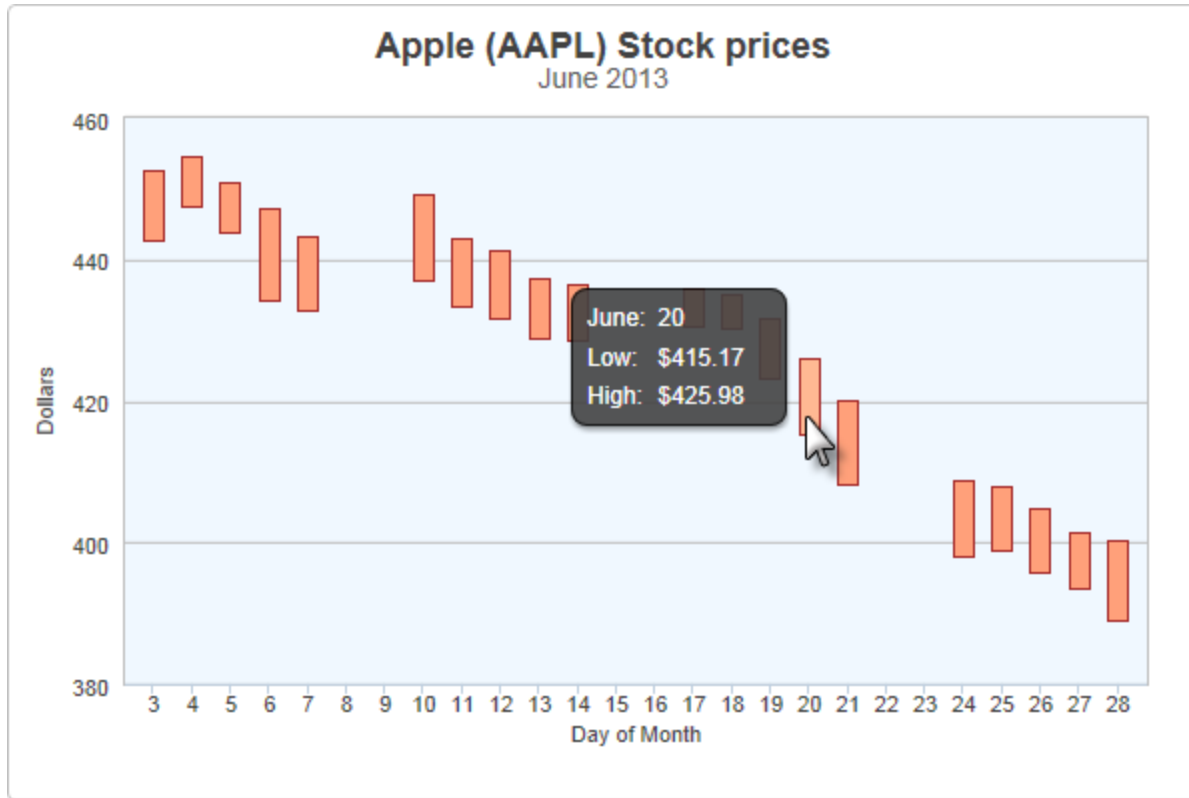
The Chart Canvas element's Series child elements cause a data visualization (the chart) to be rendered in the canvas.

The following topics discuss the Series.Bar Range child element:

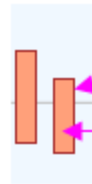
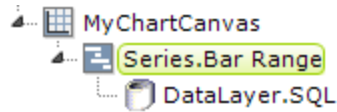
- [Using Multiple Series](#)
- [Series.Bar Range Attributes](#)
- [Using the Data Labels Element](#)
- [Using the Quicktips Element](#)
- [Using Action Elements](#)
- [Using the Series Annotation Element](#) v23.1
- [Using Input Selection](#)
- [Using the Refresh Series Timer](#)

About Series.Bar Range

The **Series.Bar Range** element generates a Bar Range chart, which is commonly used to represent separate events that have starting and ending values, over time.



The example above shows a simple Bar Range chart, presenting daily low and high stock prices for a month.



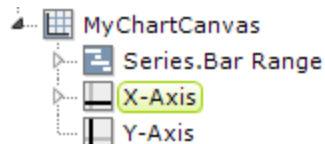
Element - Series.BarRange

| *Required Attributes | |
|-------------------------|-------------|
| High Value Data Column | High |
| Low Value Data Column | Low |
| *Optional Attributes | |
| Bar Border Color | Brown |
| ... | |
| Color | LightSalmon |
| ... | |
| X-axis Data Column | DayOfMonth |
| X-axis Data Column Type | |

As shown above, the chart is created by adding Series.Bar Range to the canvas, along with a datalayer. Very few attributes need to be set for this Series element in order to produce a basic chart.



A datalayer element can be used either beneath Series.Bar Range, as shown above, or beneath Chart Canvas. If used as a child of Chart Canvas, its data is available to *all* child Series elements. This can improve performance if you have several series, all using the *same* data, beneath the same Chart Canvas element.



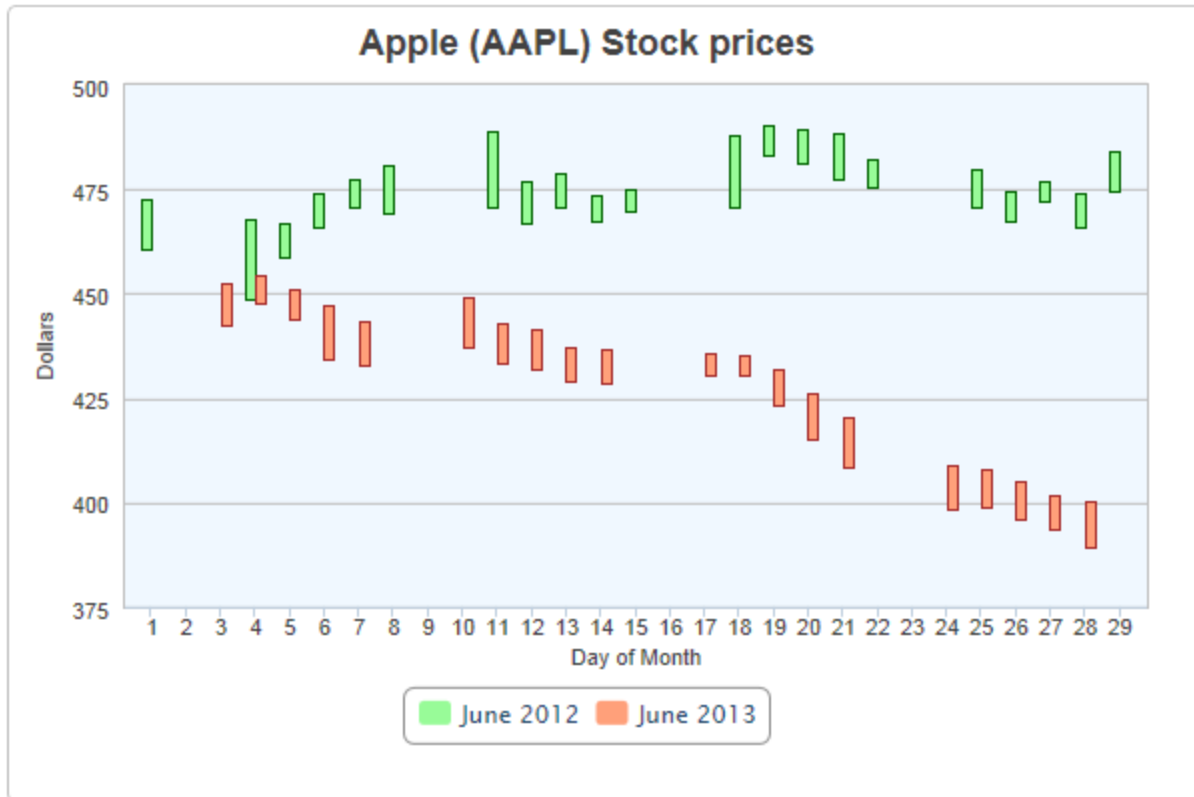
Element - ChartXAxis

| *Optional Attributes | |
|----------------------|------------------|
| Axis Padding Left | |
| Axis Padding Right | |
| ... | |
| Caption | Day of the Month |
| ... | |
| Spacing | |

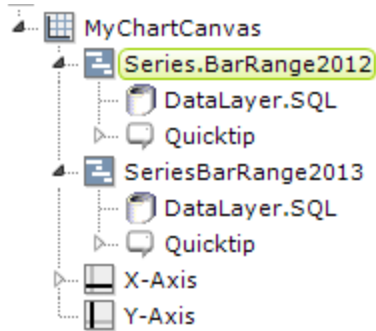
The properties of the X- and Y-axis, including captions, are set using the **X-Axis** and **Y-Axis** elements. For example, in order to provide a caption for the X-axis, add an **X-Axis** element beneath Chart Canvas and set its **Caption** attribute, as shown above. Repeat for the Y-axis.

Series.Bar Range - Using Multiple Series

You can add additional series to the chart by adding additional Series elements:




The example above shows two Series, for a set of stock prices from different years, along with a legend.



| Element - Series.BarRange | |
|-----------------------------|---------------------|
| *Required Attributes | |
| High Value Data Column | High |
| Low Value Data Column | Low |
| Optional Attributes | |
| Bar Border Color | DarkGreen |
| ... | |
| Color | PaleGreen |
| ... | |
| ID | Series.BarRange2012 |
| Legend Label | June 2012 |
| ... | |
| X-axis Data Column | DayOfMonth |
| X-axis Data Column Type | |

The example above shows the two Series elements, their datalayers and optional Quicktips, and the X-Axis and Y-Axis elements used to produce the previous chart. You can adjust which data range appears "in front" of the other in the chart and in the legend by changing the order of the Series elements in the definition. Setting the Series elements' **Legend Label** attribute will automatically cause the legend to be displayed.

Charts with multiple series or aggregations are encouraged to use the ChartCanvas element's attribute, Tooltip Split. This attribute allows you to split a chart's tooltip into one label per series, rather than per segment. The default value for this attribute is *False*. Once enabled, when hovering over a data point, it displays all the values for the series. This method is recommended over shared tooltips for charts with multiple line series, and as such, takes precedence over `tooltip.shared`.

 When using multiple series, you may be able to reduce the number of data queries and improve performance by using local data to read all of the data once, see *Datalayer Introduction*. Then, link its datalayer to share it to the series, see *Link Datalayers*.

At each Series element, link its datalayer to the shared Local Data datalayer and filter the data to meet the needs of each individual series.

You can combine different types of Series elements, for example, Series.Area and Series.Bar, to produce combinations of visualizations.

v23.1 If you are using the Chart Color Axis element in a mult-series chart, by default, the series will link to the first color axis. To link the series to a special color axis, set the ID attribute of the Chart Color Axis element to the corresponding Series' Linked to Color Axis ID attribute. For more information, see "Chart Color Axis" on page 124.

Series.Bar Range - Attributes

The Series.Bar Range element has the following attributes:

| Attribute | Description |
|-------------------------------|---|
| High Value Data Column | (Required) Specifies the name of the datalayer column containing the high data value for each row. |
| Low Value Data Column | (Required) Specifies the name of the datalayer column containing the low data value for each row. |
| Bar Border Color | Sets the color of the thin border line around each bar. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| Bar Border Color Transparency | Specifies the transparency of the thin border line around each bar. The lowest value of 0 specifies that the background is opaque, with no transparency. At the other end of the scale, 15 specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| Bar Border Radius | Sets the amount of rounding for bar corners, in pixels. The default value is 0 pixels, which produces square corners. |
| Bar Border Thickness | Sets the thickness of the bar border lines, in pixels, when the related Bar Border Color attribute has a value. The default value is 1 pixel. |

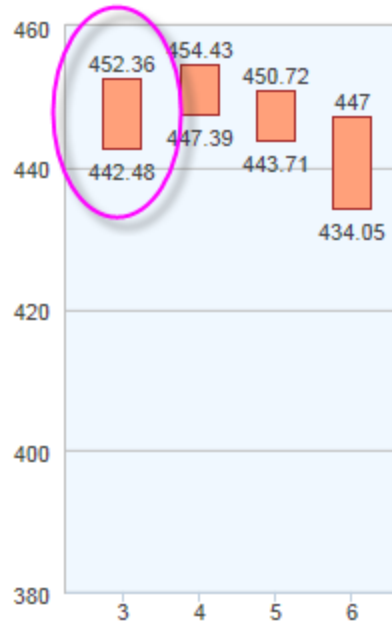
| Attribute | Description |
|--|---|
| Bar Thickness | Sets the width of the bar in pixels. If left blank, the width will be determined automatically. |
| v23.1 Class Name | Specifies the Class Name referenced in the .css style sheet. The default value is <i>undefined</i> . |
| Color | Sets the data region fill color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| v23.1 Color Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the x-axis. |
| Combine With Series ID | Set this attribute to the element ID of another series to combine it with this series in the legend. When two series are combined, by default only the first one will appear in the legend but clicking the item in the legend toggles both series to appear and disappear. Or, the value <i>Previous</i> can be entered to combine this series with the previous series. |
| Disable Bar Grouping | Disables grouping of side-by-side bars, when set to <i>True</i> . Bars that are not grouped are drawn individually and overlap each other. The default value is <i>False</i> . |
| Disable Mouse Tracking | Disables mouse tracking for the series, when set to <i>True</i> . This affects tooltips and click events on graphs and points. For large datasets, this may improve performance. The default value is <i>False</i> . |

| Attribute | Description |
|--|--|
| Hover Brightness | Specifies the amount to change a bar's color when the mouse pointer is hovered over it. Values can be between 0 (no change) and 15 (lighter). The default value is 2. |
| Legend Label | Indicates text that will be shown for this series inside the chart legend. When a value is provided, automatically causes the legend to be displayed. |
| Line Color | Sets the data region's border line color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| Line Color Transparency | Specifies the transparency of the data region border line color. The lowest value of 0 specifies that the background is opaque, with no transparency. At the other end of the scale, 15 specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| Line Style | Specifies the pattern of the data region's border line as either <i>Solid</i> or a combination of dashes and dots. |
| Line Thickness | Specifies the thickness of the data region's border line, in pixels. The default value is 1 pixel. |
| v23.1 Linked to Color-Axis ID | Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes |
| Linked to X-Axis ID | Specifies the ID of an X-Axis element that this series should be linked to when using multiple X-axes. |

| Attribute | Description |
|-----------------------------|---|
| Linked to Y-Axis ID | Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes. |
| Negative Color | Sets the color for negative values. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #D5F484. The default value is <i>Red</i> . |
| Negative Color Transparency | Specifies the transparency of the Negative Color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| Negative Threshold | Sets the positive-negative value threshold, also called the "zero level" or "base level". The default value is <i>0</i> . |
| Transparency | Specifies the transparency of the data region fill color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| X-Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the X-axis. |
| X-Axis Data Column Type | <p>Specifies the data type of the datalayer column named in the X-axis Data Column attribute. Options include <i>Auto</i> (the default), <i>Text</i>, <i>Number</i>, and <i>DateTime</i>.</p> <p>By default, X-axis data values that are <i>DateTime</i> type will be automatically distributed evenly across the time series. If you want to disable this behavior, set this attribute value to <i>Text</i>.</p> |

Series.Bar Range - Using the Data Labels Element

A "data label" is text shown next to each data point that shows its value. When the **Data Labels** element is used as a child of Series.Bar Range, text representing the data values will appear on the chart:

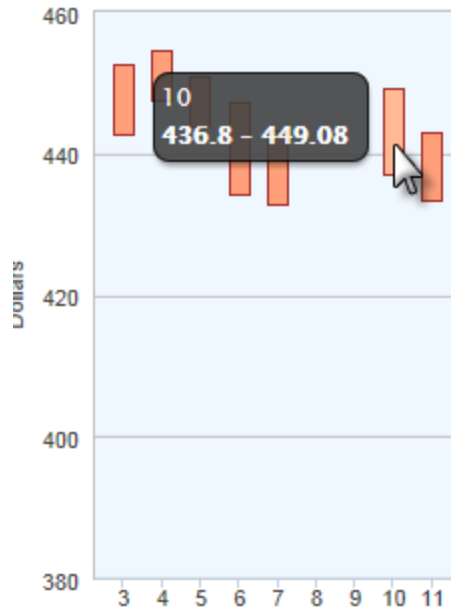


The Data Labels element has attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text. One attribute, Series Name as Caption, allows you to specify the y-axis label, x-axis label, or legend label as the data point value inside the chart.

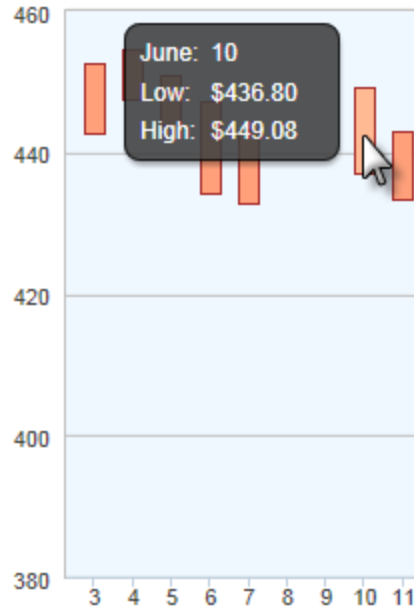
The Data Labels element's color-related attribute values can be set using @Chart tokens. v23.1 Or, utilize the "Class Name" attribute to apply unique styling to your data labels by referencing a class name from a stylesheet in the report.

Series.Bar Range - Using the Quicktips Element

By default, a "quicktip" is displayed when the mouse hovers near or over a data point:

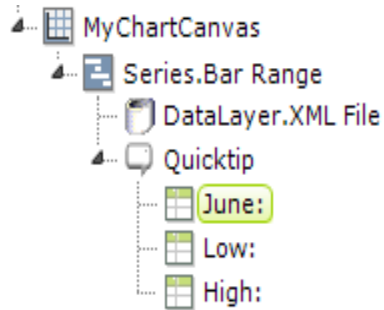


Default quicktip



With Quicktip child element

The automatically-generated quicktip displays information for the X- and Y-axis, as shown above, left. However, you may want to display other information or format it differently, as shown above, right, which can be done by adding a **Quicktip** child element beneath Series.Bar Range.



| Optional Attributes | |
|---------------------|--------------------|
| Caption | June: |
| Format | |
| ID | |
| Value | @Chart.DayOfMonth~ |

The example above shows a Quicktip element (no attributes need to be set) and three **Quicktip Row** child elements, used to create the quicktip shown in the previous image. Use @Chart tokens to include chart data in the quicktip.

You can also configure the tooltip to display values for any available column in the datalayer. To do so, enter a token representing the data column in the Value attribute of the Quicktip Row element.



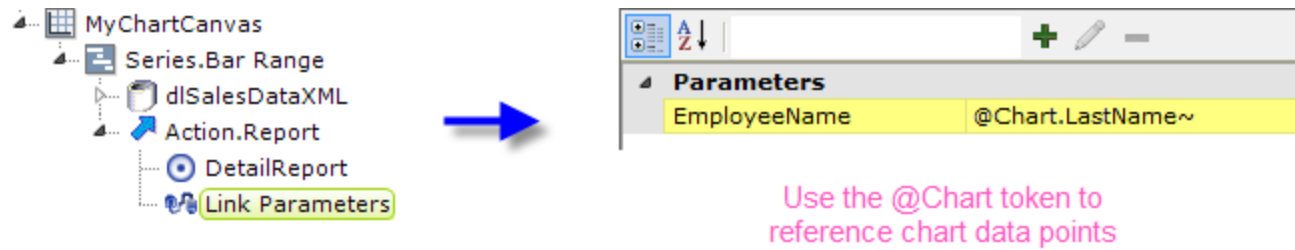
To use this feature with DataLayer.ActiveSQL, please make sure the keep Grouped Rows attribute of the SqlGroup element is set to *False*.

Intrinsic functions are supported in the Quicktip attributes.

The Quicktip Row element has been made context-sensitive with the addition of a **Condition** attribute.

Series.Bar Range - Using Action Elements

Action elements initiate processing of a report or process task definition, redirection to a link, or other processing when a data point in the series is clicked.



In the example above, an **Action.Report** element has been added as a child of the Series, along with its **Target.Report** and **Link Parameters** child elements. To reference chart data in parameters, use the @Chart token, as shown above.

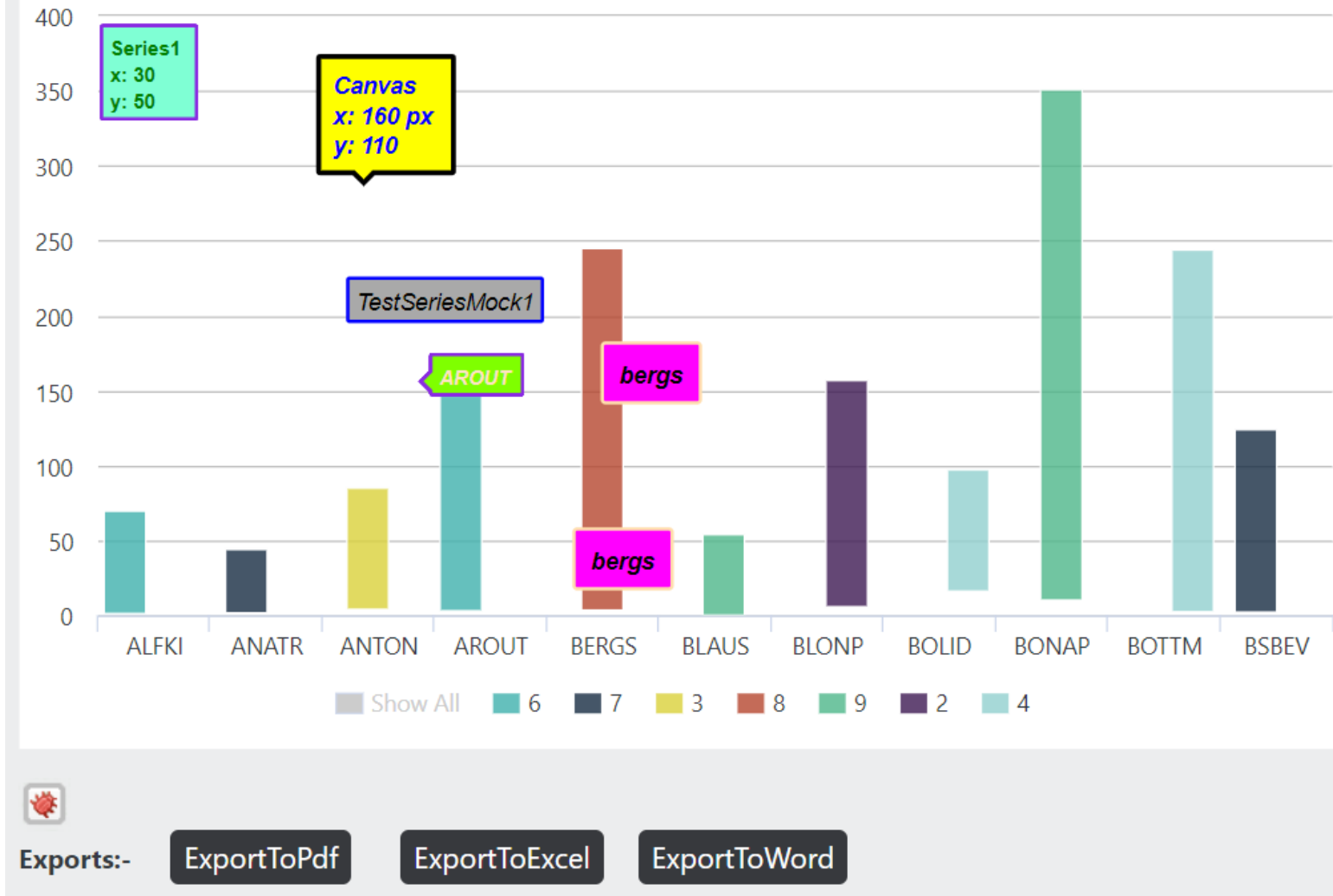
A variety of Action elements are available for use with Series, including Action.Link, Action.Process, and Action.Refresh Element. Additional Action elements will be added in future releases.

v23.1

Series.Bar Range - Using the Series Annotation Element

Annotations allow you to annotate a chart freely using custom labels and shapes. When the **Series Annotation** element is used as a child of Series.Bar Range, you can place these annotations at various points of interest:

Bar Range



The Series Annotation element offers two child elements: **AnnotationLabel.Point** and **AnnotationLabel.Mock**.

The `AnnotationLabel.Point` child element is used to create an annotation label and position on the chart by linking it to an existing point. With this element, you cannot specify the value, it always refers to the value of the point. Additionally, you can use the `Caption` and `Condition` attributes to refer to different datalayers.

On the other hand, the `AnnotationLabel.Mock` is used to create an annotation label and position it on the chart by linking it to a created mock point. With this element, you can specify the coordinate, relative position, and axis.

Both the `AnnotationLabel.Point` and `AnnotationLabel.Mock` have attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text.

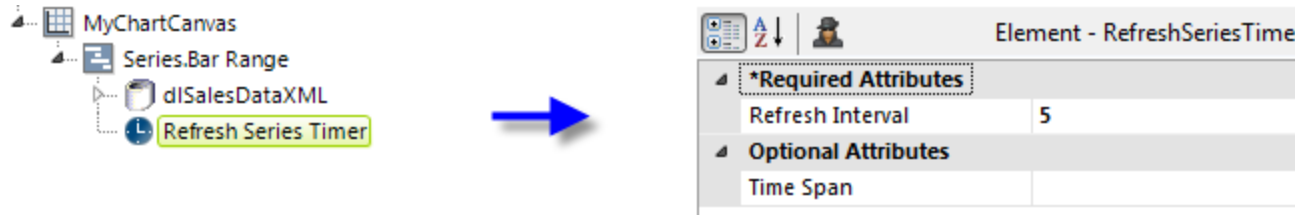
Series.Bar Range - Using Input Selection

This series can also be used with the **Input Selection** family of elements, which turn the chart into an input control. This allows users, at runtime, to select points or values on the chart with their mouse and your report can then take action using the selected data. This is very useful, for example, for drilling into the data.

For more information about this functionality, see "Input Selection" on page 97.

Series.Bar Range - Using the Refresh Series Timer

The **Refresh Series Timer**, added as a Series child element, updates charts automatically, based on a time interval. When the interval is reached, a request is sent back to the web server for updated data. Series are updated with a smooth animation.



The **Refresh Interval** attribute specifies the number of seconds to wait before refreshing the series data. A value is required and must be an integer greater than 0.

Refresh Mode

Series may be refreshed in two ways: either all of the data is refreshed with each request, or the series data automatically slides to the left one data interval per refresh. The refresh mode is selected in the Chart X Axis element's **Axis Type** attribute.

When the Axis Type is *not* set to *DateTimeLinear*, all of the data will be refreshed.

When the Axis Type value is *DateTimeLinear*, newer values are added to the right side of the chart, previous values slide left, and older values disappear as they reach the left edge of the chart. In this case, the Refresh Series Timer element's **Time Span** attribute is used to specify the age of the data included. This value must be in *hh:mm:ss* format (for example "00:02:00" for two minutes) and must be larger than the Refresh Interval attribute value.

When a Time Span attribute is set, the series' datalayer can make use of a special timestamp token, `@Chart.rdTimeSpanStart~`, in

a query to retrieve just the rows necessary to update the chart. On the initial request, this token returns the current time minus the Time Span value. For subsequent refreshes, when *Axis Type = DateTimeLinear* and the X-axis will be sliding, the token returns the last time the chart was updated, so only the newest rows are retrieved. Here's an MS SQL Server query example:

```
SELECT * FROM MyTable WHERE UpdateTime BETWEEN '@Chart.rdTimeSpanStart~' AND '@Function.DateTime~'
```

For best performance, avoid setting a very short refresh interval if a very large number of users will be displaying the report.

Series.Funnel

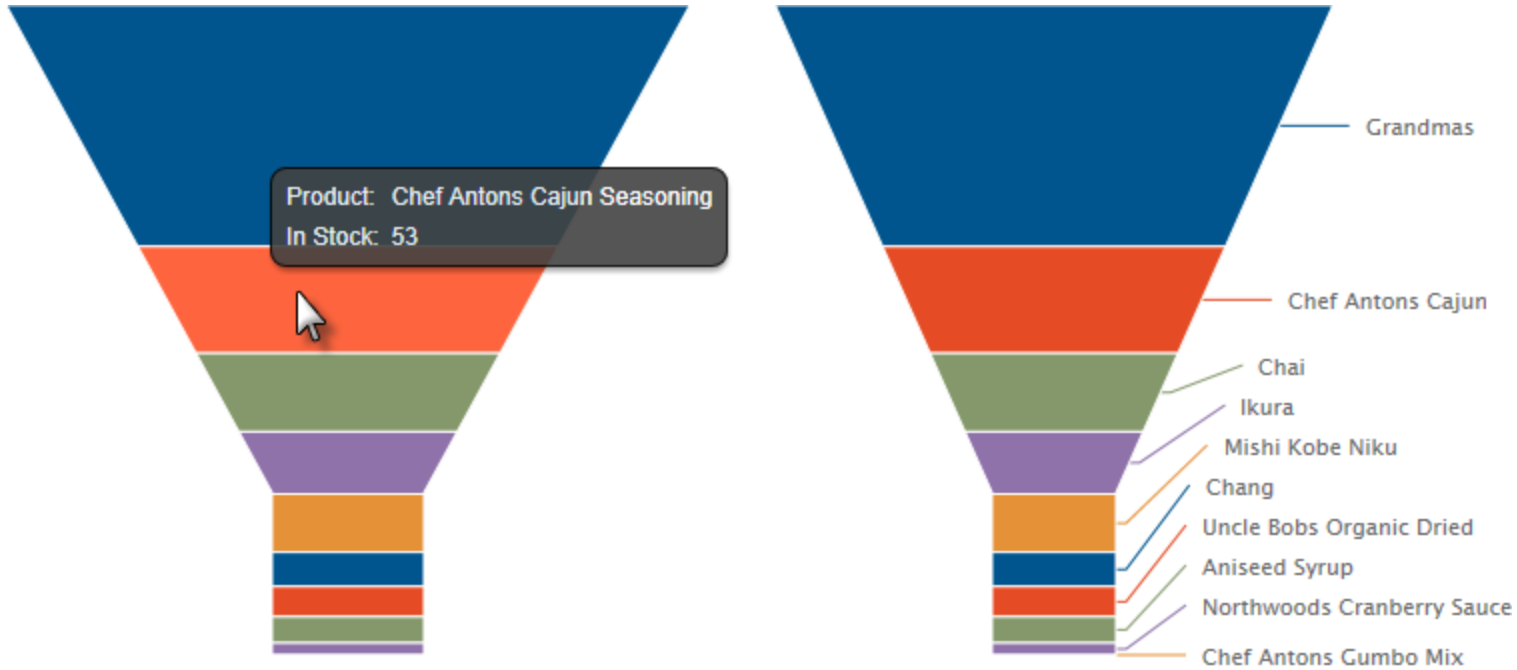
The Chart Canvas element's Series child elements cause a data visualization (the chart) to be rendered in the canvas.

The following topics discuss the Series.Funnel child element:

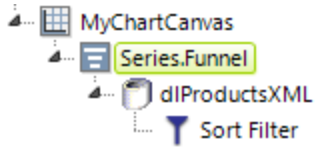
- [Using Multiple Series](#)
- [Series.Funnel Attributes](#)
- [Adding Data Labels](#)
- [Using the Quicktips Element](#)
- [Using Action Elements](#)
- [Using the Series Annotation Element](#) v23.1
- [Using Input Selection](#)
- [Using the Refresh Series Timer](#)

About Series.Funnel

The **Series.Funnel** element generates a Funnel chart, which displays values as progressively decreasing proportions. The size of each funnel segment is determined as a percentage of the total of all values.




The example above shows Funnel charts, without and with side labels.

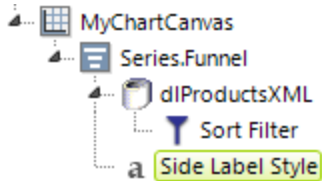


Element - Series.Funnel

| *Required Attributes | |
|--------------------------|--------------|
| Y-axis Data Column | UnitsInStock |
| Optional Attributes | |
| Colors | |
| ... | |
| Label Data Column X-axis | ProductName |
| Label Location | SideLayout |
| Neck Height | |
| Neck Width | 20% |
| Show Data Values | |

As shown above, the chart is created by adding Series.Funnel to the canvas, along with a datalayer. Very few attributes need to be set for the Series element in order to produce a basic chart.

 A datalayer element can be used either beneath Series.Funnel, as shown above, or beneath Chart Canvas. If used as a child of Chart Canvas, its data is available to *all* child Series elements. This can improve performance if you have several series, all using the *same* data, beneath the same Chart Canvas element.



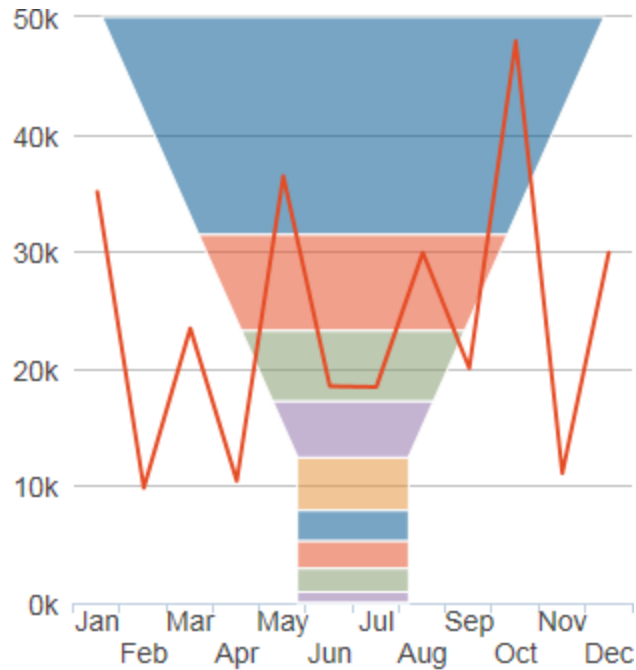
Element - SideLabelStyle

| Optional Attributes | |
|-------------------------------|-------|
| Background Color | |
| Background Color Transparency | |
| ... | |
| Connector Distance | |
| Connector Style | Lines |
| Connector Thickness | |
| Font Color | |
| Connector Color Transparency | |

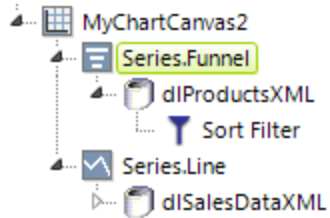
When the **Label Location** attribute has been set to *SideLayout*, as in the example, the labels can be styled using the **Side Label Style** element, as shown above.

Series.Funnel - Using Multiple Series

Funnel charts do not generally lend themselves to use with other series types, but it is possible. You can add additional series to the chart by adding additional Series elements:




The example above shows Series.Funnel with Series.Line. The funnel's transparency has been set to allow the grid and data lines to show through it.



| Element - Series.Funnel | |
|-----------------------------|--------------|
| *Required Attributes | |
| Y-axis Data Column | UnitsInStock |
| Optional Attributes | |
| Colors | |
| ... | |
| Label Data Column X-axis | ProductName |
| Label Location | SideLayout |
| Neck Height | |
| Neck Width | 20% |
| Show Data Values | |
| Transparency | 7 |

The example above shows the two Series elements and their datalayers, used to produce the previous chart. You can adjust which series appears "in front" of the other in the chart by changing the order of the Series elements in the definition. The funnel's **Transparency** attribute has been set to allow the grid and data lines to show through it.

 When using multiple series, you may be able to reduce the number of data queries and improve performance by using local data to read all of the data once, see *Datalayer Introduction*. Then, link its datalayer to share it to the series, see *Link Datalayers*. At each Series element, link its datalayer to the shared Local Data datalayer and filter the data to meet the needs of each individual series.

v23.1 If you are using the Chart Color Axis element in a mult-series chart, by default, the series will link to the first color axis. To link the series to a special color axis, set the ID attribute of the Chart Color Axis element to the corresponding Series' Linked to Color Axis ID attribute. For more information, see "Chart Color Axis" on page 124.

Series.Funnel - Attributes

The Series.Funnel element has the following attributes:

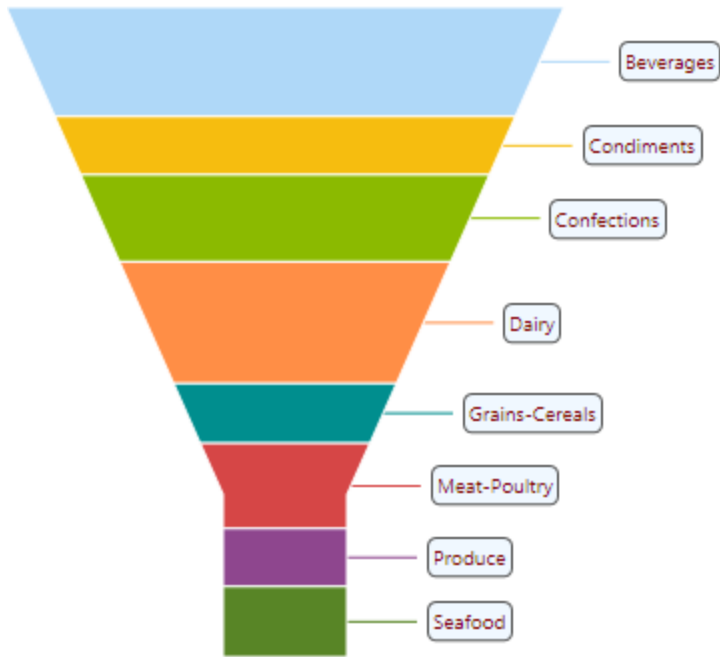
| Attribute | Description |
|--|---|
| Y-Axis Data Column | (Required) Specifies the name of a datalayer column whose values determine the height of each funnel segment. |
| v23.1 Class Name | Specifies the Class Name referenced in the .css style sheet. The default value is <i>undefined</i> . |
| Colors | Sets the colors of the funnel segments, which should be entered as a comma-separated list. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. When there are more segments than colors specified, the listed colors are used again from the beginning. A default set of colors is used if this is left blank and no theme has been applied, see "Chart Canvas Charts" on page 20. |
| v23.1 Color Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the x-axis. |
| Disable Mouse Tracking | Disables mouse tracking for the series, when set to <i>True</i> . This affects tooltips and click events. For large datasets, this may improve performance. The default value is <i>False</i> . |
| Hover Brightness | Specifies the change in a funnel segment's brightness when the mouse pointer hovers over it. Values can be 0 (no change) through 15(lighter). The default value is 2. |

| Attribute | Description |
|--|---|
| Label Data Column X-axis | Specifies the name of a datalayer column whose values be represented by funnel segments. |
| Label Location | <p>Specifies where funnel segment labels will appear. Options include: <i>SideLayout</i> - (the default) labels will appear to the right, with lines connecting them to segments.</p> <p><i>Legend</i> - segments to be identified in a legend. The legend items can be clicked to toggle the visibility of individual segments.</p> <p><i>NoLabels</i> - labels will not be shown.</p> |
| v23.1 Linked to Color-Axis ID | Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes |
| Neck Height | Specifies the height of the "neck", or lower part, of the funnel. May be specified as a number alone, indicating pixels, or with a percentage sign, indicating a percentage of the plot area height. The default is value is <i>25%</i> . |
| Neck Width | Specifies the width of the "neck", or lower part, of the funnel. May be specified as a number alone, indicating pixels, or with a percentage sign, indicating a percentage of the plot area width. The default is value is <i>30%</i> . |
| Show Data Values | Specifies if the value of each data point should be shown on the chart. Depending on the value of the Label Location attribute, they may be shown alone or with the label values. The default value is <i>False</i> . |

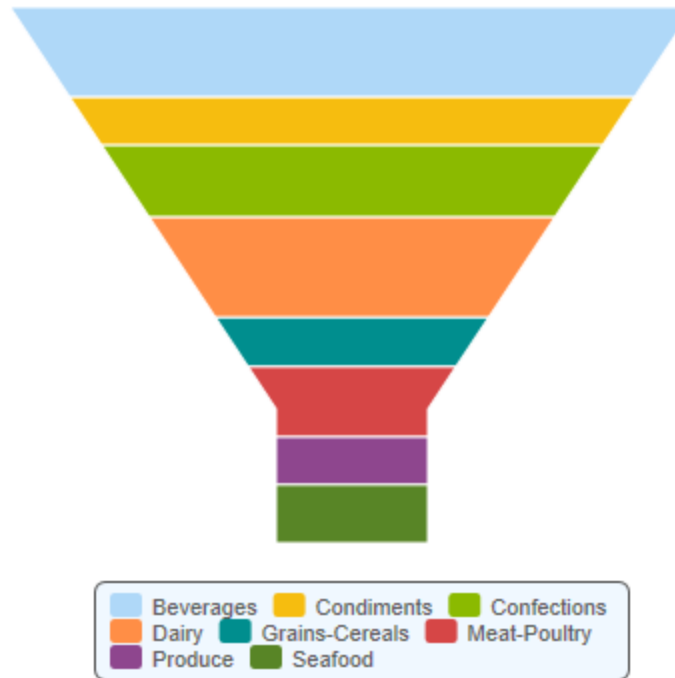
| Attribute | Description |
|-------------------------|---|
| Show Data Values Format | <p>Specifies formatting characteristics for data values. For dates, the non-specific formats, such as <i>General Date</i>, <i>Short Time</i>, etc., are converted according to the browser's international settings. For very large reports, the non-specific formats perform better. Special formats include: < and > - change strings to lower and upper case.</p> <p><i>Expanded Spaces</i> - preserves space characters that would otherwise be collapsed by the web browser.</p> <p><i>mp</i> - formats numbers with the "metric prefix". For example, to format 1,234,567 as "\$1.23M", enter <i>\$.00mp</i>. Supported metric prefixes are from 1000^6 to 1000^{-6}. For more information see this page.</p> <p><i>qq</i> - returns the number of the quarter when the value being formatted represents a date. To return the year and quarter together like "2010 Q1", set the format to <i>yyyy Qqq</i>.</p> |
| Transparency | <p>Specifies the transparency of the funnel segments. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different series to show through each other.</p> |

Series.Funnel - Adding Data Labels

A "data label" is text that can be shown adjacent to each funnel segment, that shows its X-axis data value. This is controlled using the Label Location attribute.



Label Location = *SideLayout*

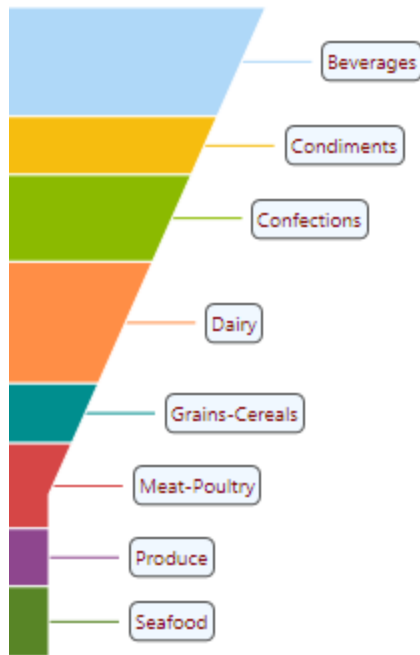


Label Location = *Legend*

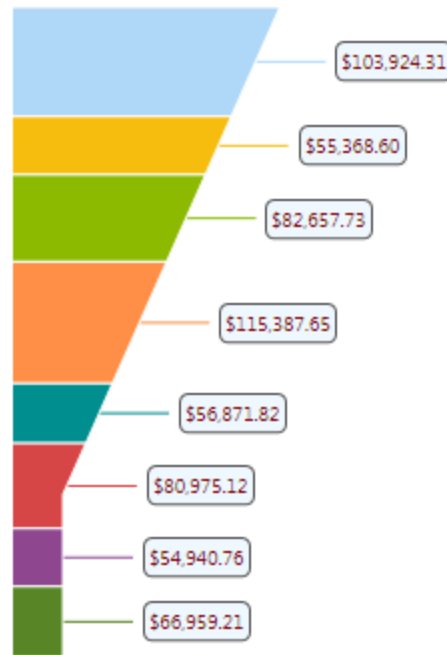
As shown above, labels can be placed beside the segments, or in a legend, or not shown at all.

When the legend option is selected, "legend filtering" is active: clicking a segment's entry in the legend will toggle its appearance in the chart. For more information about legends, see "Legend" on page 70.

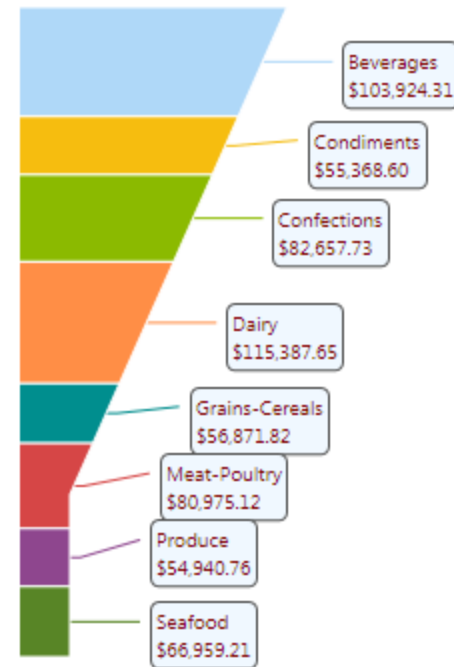
💡 When working with labels beside the funnel, you may need to adjust the Chart Canvas element's **Spacing Left** and **Spacing Right** attributes to "fit" all of the label text inside the canvas.



Label Location = *SideLayout*



Label Location = *NoLabels*
Show Data Values = *True*



Label Location = *SideLayout*
Show Data Values = *True*

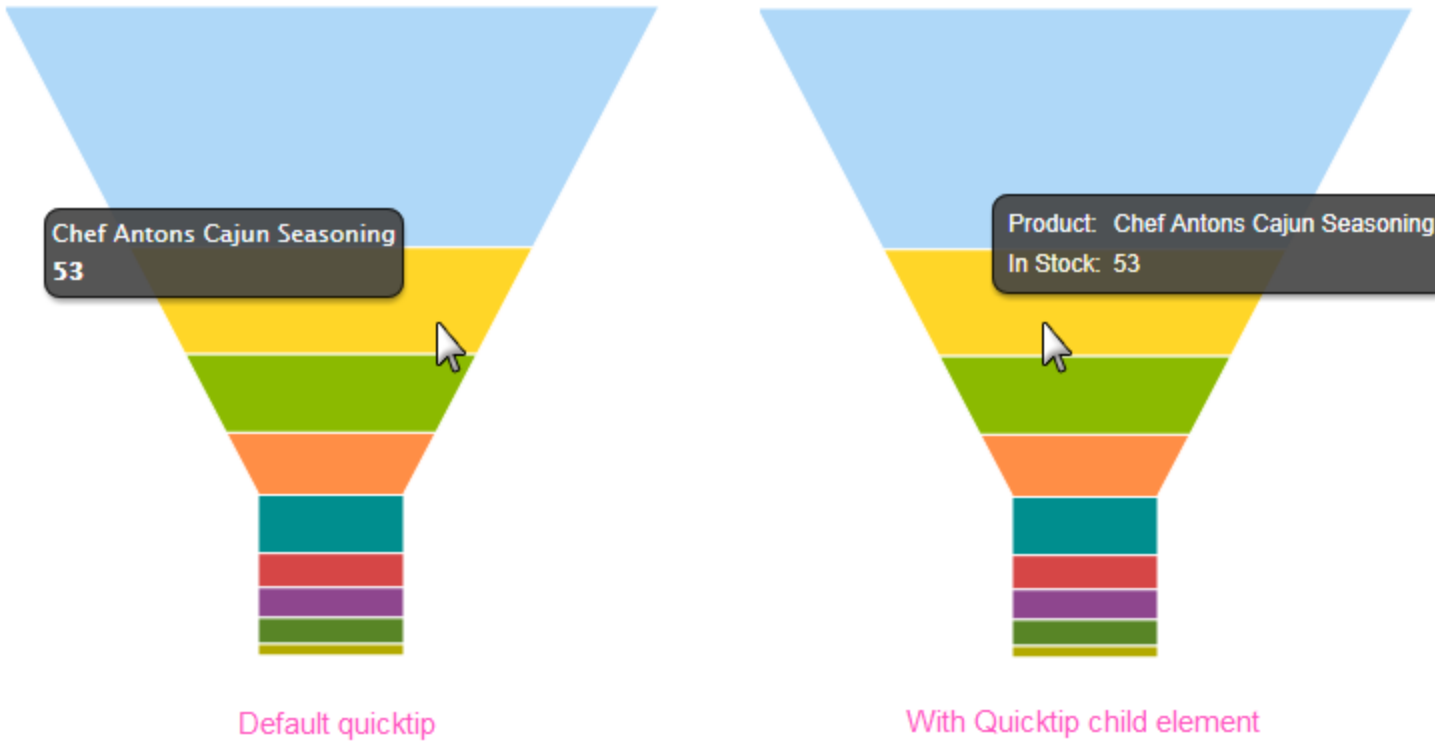
If you want to show the data *values* instead of, or with, the data *labels* near each funnel segment, as shown above, use the Series.Funnel element's **Label Location**, **Show Data Values**, and **Show Data Values Format** attributes.

The values displayed can also be styled using the **Side Label Style** element. Its attributes allow you to control the font family, color, size, and weight, the data format, background color, border color, connecting lines and more. The side labels in the image above have been styled.

The Side Label Style element's **Maximum Label Length** attribute lets you specify the maximum number of characters that will be displayed for a label before the text is truncated and ellipsis (...) is appended.


Series.Funnel - Using the Quicktips Element

By default, a "quicktip" is displayed when the mouse hovers over a funnel segment:



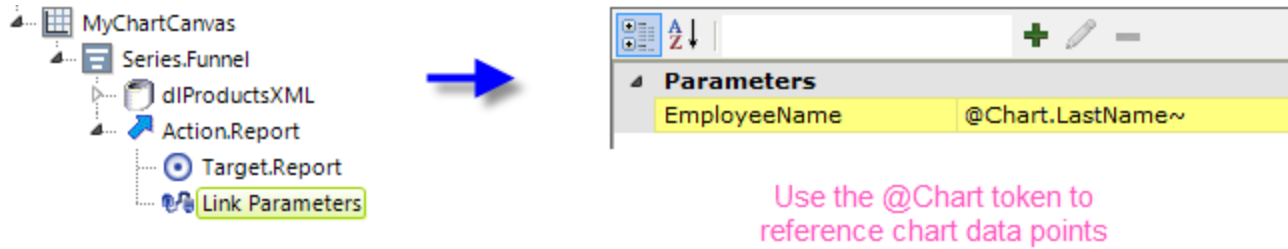
The automatically-generated quicktip displays information for the X- and Y-axis, as shown above, left. However, you may want to display other information or format it differently, perhaps as shown above, right, which can be done by adding a **Quicktip** child element beneath Series.Funnel and setting its attributes and child elements. Use @Chart tokens to include chart data in the quicktip. Intrinsic functions are supported in the Quicktip attributes.

You can also configure the tooltip to display values for any available column in the datalayer. To do so, enter a token representing the data column in the Value attribute of the Quicktip Row element.

 To use this feature with DataLayer.ActiveSQL, please make sure the keep Grouped Rows attribute of the SqlGroup element is set to *False*.

Series.Funnel - Using the Action Elements

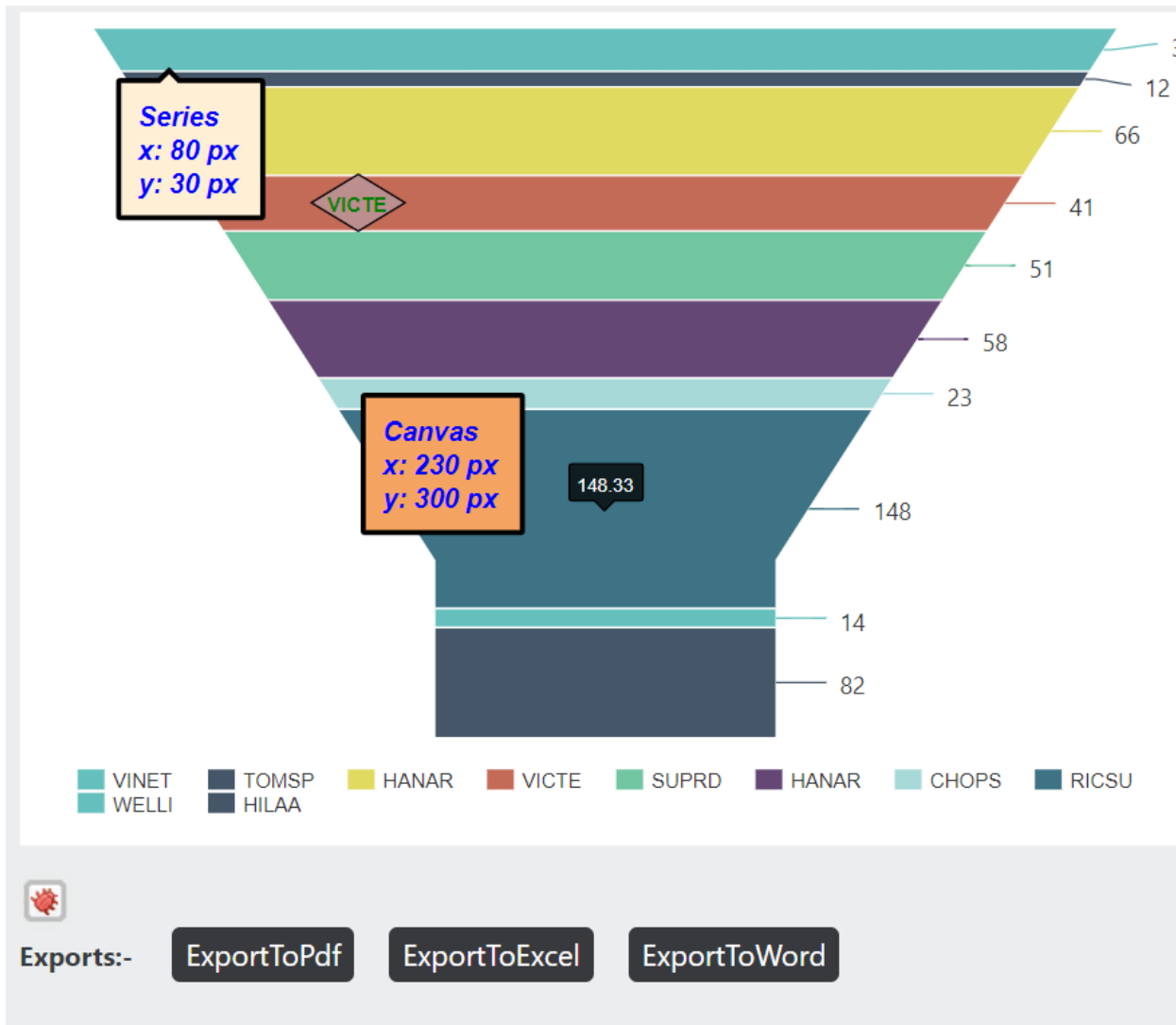
Action elements initiate processing of a report or process task definition, redirection to a link, or other processing when a segment in the funnel is clicked.



In the example above, an **Action.Report** element has been added as a child of the Series, along with its **Target.Report** and **Link Parameters** child elements. To reference chart data in parameters, use the @Chart token, as shown above. A variety of Action elements are available for use with Series, including Action.Link, Action.Process, and Action.Refresh Element. Additional Action elements will be added in future releases.

Series.Funnel - Using the Series Annotation Element

Annotations allow you to annotate a chart freely using custom labels and shapes. When the **Series Annotation** element is used as a child of Series.Funnel, you can place these annotations at various points of interest:



The Series Annotation element offers two child elements: **AnnotationLabel.Point** and **AnnotationLabel.Mock**.

The AnnotationLabel.Point child element is used to create an annotation label and position on the chart by linking it to an existing point. With this element, you cannot specify the value, it always refers to the value of the point. Additionally, you can use the Caption and Condition attributes to refer to different datalayers.

On the other hand, the AnnotationLabel.Mock is used to create an annotation label and position it on the chart by linking it to a created mock point. With this element, you can specify the coordinate, relative position, and axis.

Both the AnnotationLabel.Point and AnnotationLabel.Mock have attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text.

Series.Funnel - Using Input Selection

This series can also be used with the **Input Selection** family of elements, which turn the chart into an input control. This allows users, at runtime, to select points or values on the chart with their mouse and your report can then take action using the selected data. This is very useful, for example, for drilling into the data. For more information about this functionality, see "Input Selection" on page 97.

Series.Funnel - Using the Refresh Series Timer

The **Refresh Series Timer**, added as a Series child element, updates charts automatically, based on a time interval. When the interval is reached, a request is sent back to the web server for updated data. Series are updated with a smooth animation.



The **Refresh Interval** attribute specifies the number of seconds to wait before refreshing the series data. A value is required and must be an integer greater than 0.

Refresh Mode

Series may be refreshed in two ways: either all of the data is refreshed with each request, or the series data automatically slides to the left one data interval per refresh. The refresh mode is selected in the Chart X Axis element's **Axis Type** attribute. When the Axis Type is *not* set to *DateTimeLinear*, all of the data will be refreshed.

When the Axis Type value is *DateTimeLinear*, newer values are added to the right side of the chart, previous values slide left, and older values disappear as they reach the left edge of the chart. In this case, the Refresh Series Timer element's **Time Span** attribute is used to specify the age of the data included. This value must be in *hh:mm:ss* format (for example "00:02:00" for two minutes) and must be larger than the Refresh Interval attribute value.

When a Time Span attribute is set, the series' datalayer can make use of a special timestamp token, `@Chart.rdTimeSpanStart~`, in

a query to retrieve just the rows necessary to update the chart. On the initial request, this token returns the current time minus the Time Span value. For subsequent refreshes, when *Axis Type = DateTimeLinear* and the X-axis will be sliding, the token returns the last time the chart was updated, so only the newest rows are retrieved. Here's an MS SQL Server query example:

```
SELECT * FROM MyTable WHERE UpdateTime BETWEEN '@Chart.rdTimeSpanStart~' AND '@Function.DateTime~'
```

For best performance, avoid setting a very short refresh interval if a very large number of users will be displaying the report.

Series.Heatmap

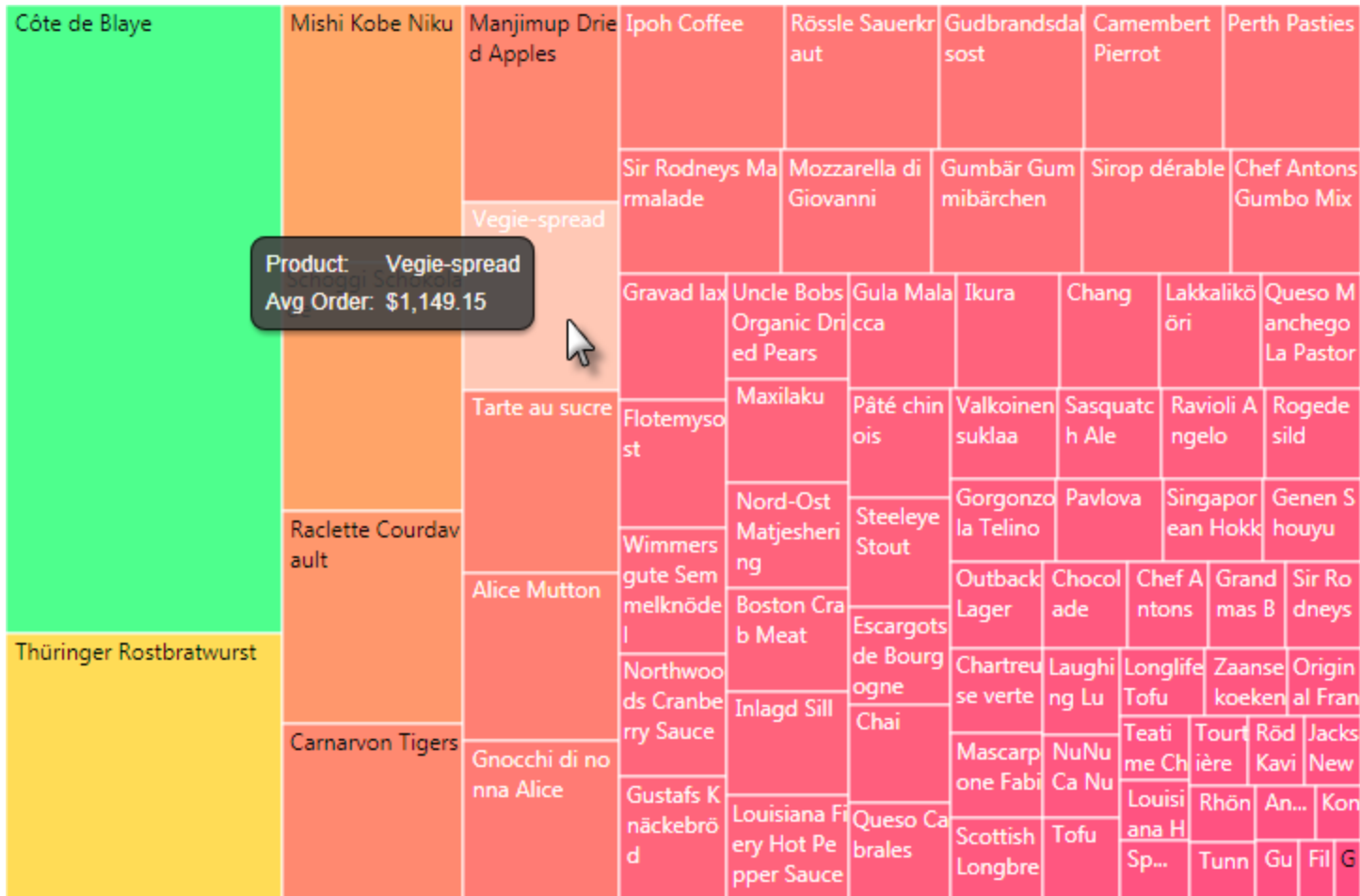
The Chart Canvas element's Series child elements cause a data visualization (the chart) to be rendered in the canvas.

The following topics discuss the Series.Heatmap child element:

- [Series.Heatmap Attributes](#)
- [Grouping Heatmap Cells](#)
- [Using the Heatmap Label Style Element](#)
- [Using the Quicktips Element](#)
- [Using the Chart Drill To Element](#)
- [Using Action Elements](#)
- [Using the Series Annotation Element](#) v23.1
- [Using the Refresh Series Timer](#)

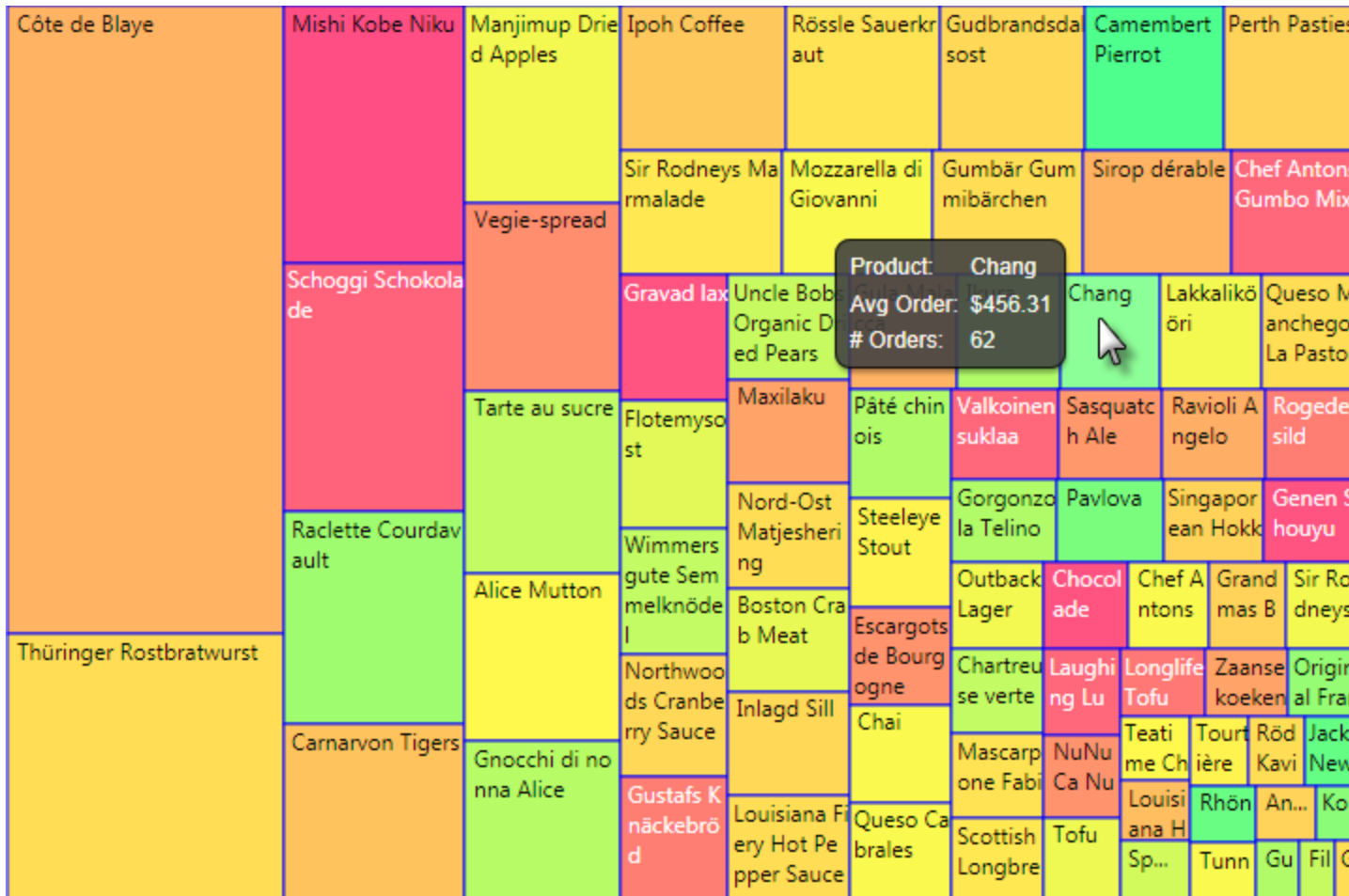
About Series.Heatmap

The **Series.Heatmap** element generates a Heatmap chart, sometimes called a "tree map", which uses a unique arrangement of rectangles to represent data and relationships, using color and size.



Both the color intensity (shifting within a range) and the size of the rectangles indicate relative performance, helping you to easily spot trends with a glance. This type of chart is very useful for showing hidden trends and relationships.

In the example above, the rectangle's (or cell's) size represents the average order value for each food item. The cell color also represents the average order value, ranked from red (low) to green (high). The relationship isn't linear but we can still see that items with small, red cells produced fewer orders and less revenue than items with large, green cells.




The example above is the same data but now the cell color reflects the number of orders. "Greener" cells have a higher number of orders. This allows us to compare an item's popularity (size) to its average order value (color).

The diagram illustrates the configuration of a heatmap series. On the left, a tree view shows the hierarchy: MyChartCanvas contains Series.Heatmap, which contains diHeatMapData. diHeatMapData contains joinInner, which contains calcOrderValue. calcOrderValue contains grpCatID, which contains grpProdID, which contains avgOrderValue. A blue arrow points from this hierarchy to a configuration table on the right.

The configuration table, titled "Element - Series.Heatmap", lists the following attributes:

| *Required Attributes | |
|----------------------------|---------------|
| Heat Map Color Data Column | avgOrderValue |
| Heat Map Label Data Column | ProductName |
| Heat Map Size Data Column | avgOrderValue |
| Optional Attributes | |
| ... | |
| High Value Color | #4EFF8D |
| Hover Brightness | |
| ID | |
| Low Value Color | #FF4E83 |
| Medium Value Color | #FFF94E |

As shown above, the chart is created by adding Series.Heatmap to the canvas, along with a datalayer and, typically, some child elements that may include a **Group Filter**, and a **Group Aggregate Column** element. Very few attributes need to be set for the Series element in order to produce a basic chart.

 A datalayer element can be used either beneath Series.Heatmap, as shown above, or beneath Chart Canvas. If used as a child of Chart Canvas, its data is available to *all* child Series elements. This can improve performance if you have several series, all using the *same* data, beneath the same Chart Canvas element.

Series.Heatmap - Attributes

The Series.Heatmap element has the following attributes:

| Attribute | Description |
|--------------------------------|--|
| Heat Map Color Data Column | (Required) Specifies the name of a datalayer column whose values will determine the color of the cells, using a high-to-low gradient based on the three Value Color attributes. |
| Heat Map Label Data Column | (Required) Specifies the name of a datalayer column whose values will be used as labels inside the cells. |
| Heat Map Size Data Column | (Required) Specifies the name of a datalayer column whose values will determine the size of the cells. |
| Cell Border Color | Sets the color of the cell borders. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| Cell Border Color Transparency | Specifies the transparency of the Cell Border Color. The lowest value of 0 specifies that the background is opaque, with no transparency. At the other end of the scale, 15 specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| Cell Border Thickness | Sets the thickness of the cell border lines, in pixels, when the related Cell Border Color attribute has a value. The default value is 1 pixel. |
| Disable Mouse | Disables mouse tracking for the series, when set to <i>True</i> . This affects tooltips and click events on cells. For |

| Attribute | Description |
|--------------------|---|
| Tracking | <p>large datasets, this may improve performance.</p> <p>The default value is <i>False</i>.</p> |
| High Value Color | <p>Specifies the color for the highest data value. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #D5F484. Or, enter the token @Gradient to use a gradient fill to represent the data.</p> |
| Hover Brightness | <p>Specifies the amount to change a cell's color when the mouse pointer is hovered over it. Values can be between 0 (no change) and 15(lighter). The default value is 2.</p> |
| ID | <p>Specifies the unique element ID.</p> |
| Low Value Color | <p>Specifies the color for the lowest data value. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #D5F484. Or, enter the token @Gradient to use a gradient fill to represent the data.</p> |
| Medium Value Color | <p>Specifies the color for the data values in the middle of the dataset. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #D5F484. Or, enter the token @Gradient to use a gradient fill to represent the data.</p> |

Series.Heatmap - Grouping Heatmap Cells

You can also group cells in a Heatmap using the **Heatmap Group** element:



In the example above, the products have grouped by category. In order to use grouping, you must ensure that your datalayer includes the appropriate data. The Heatmap Group element allows you to control the appearance of the group title font, title alignment, borders, and background color.

Series.Heatmap - Using the Heatmap Label Style Element

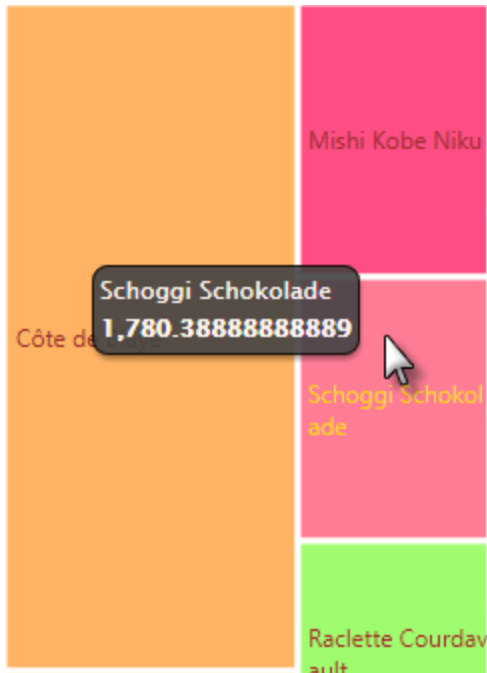
The **Heatmap Label Style** element can be used to control the appearance of the label text that displayed inside the cells:



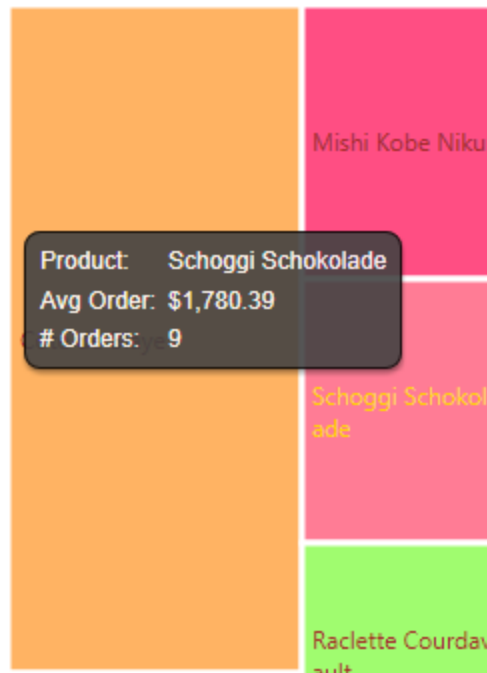
By default, labels will appear at the top of the cell, as shown above, top. The Heatmap Label Style element has attributes that allow you to control the font family, color, size, and weight, and positioning of the text. You can also specify different font colors for cells with light and dark backgrounds.

Series.Heatmap - Using the Quicktips Element

By default, a "quicktip" is displayed when the mouse hovers near or over a data point:




Default quicktip



With Quicktip child element

The automatically-generated quicktip displays information from the label and size data columns, as shown above, left. However, you may want to display other information or format it differently, perhaps as shown above, right, which can be done by adding a **Quicktip** child element beneath Series.Heatmap and setting its attributes and child elements. Use @Chart tokens to include chart data in the quicktip. Intrinsic functions are supported in the Quicktip attributes. A default quicktip is also displayed when a Heatmap Group element is used. To customize it, the Quicktip element can also be a child of Heatmap Group.

You can also configure the tooltip to display values for any available column in the datalayer. To do so, enter a token representing the data column in the Value attribute of the Quicktip Row element.

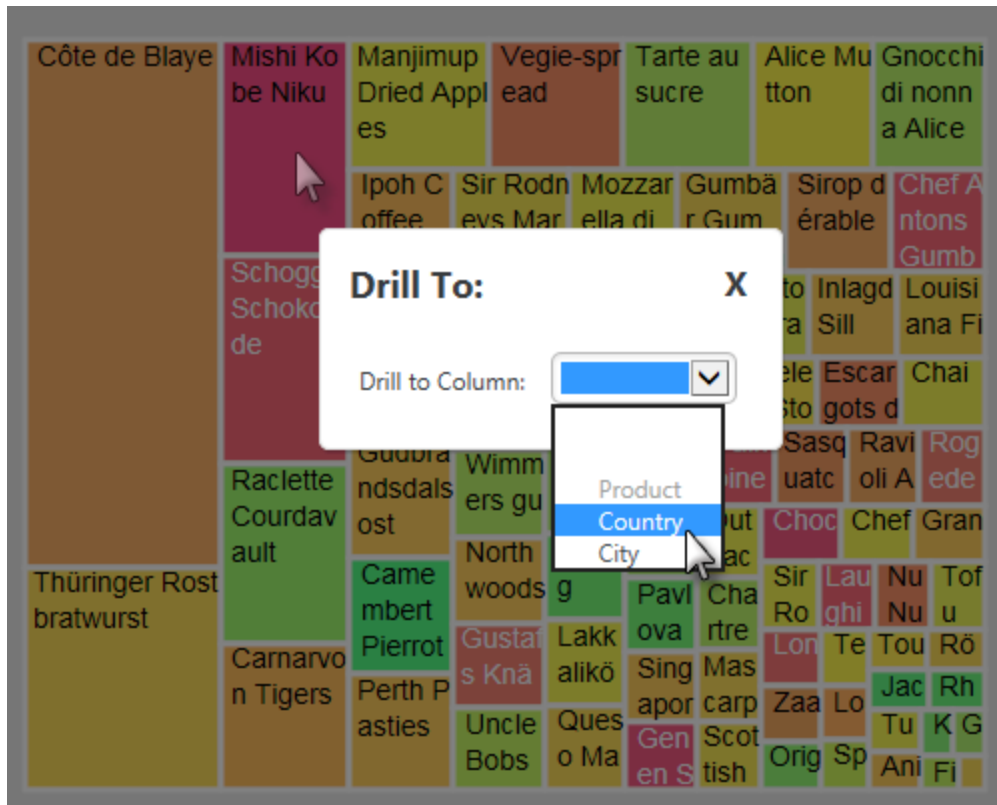
 To use this feature with `DataLayer.ActiveSQL`, please make sure the `keep Grouped Rows` attribute of the `SqlGroup` element is set to *False*.

Series.Heatmap - Using the Chart Drill to Element

The **Chart Drill To** element, a child of the Series element, enhances charts by allowing you to examine the data "behind" the chart.

 The parent Chart Canvas element must have an ID attribute value in order to use the Chart Drill To element with it.

When the Drill To element is enabled, selecting a rectangle displays a list of columns, as shown below:



Selecting a column re-draws the chart so that only the data representing that specific rectangle is shown.

Select **Drill Back** to return the chart to it's previous state:

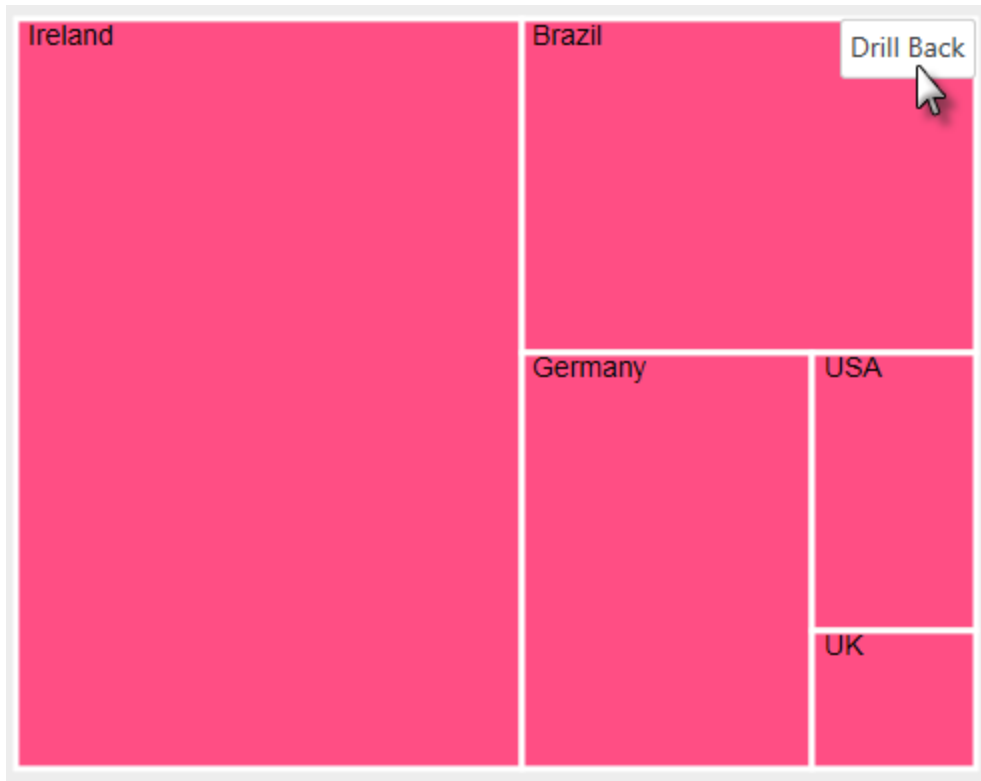
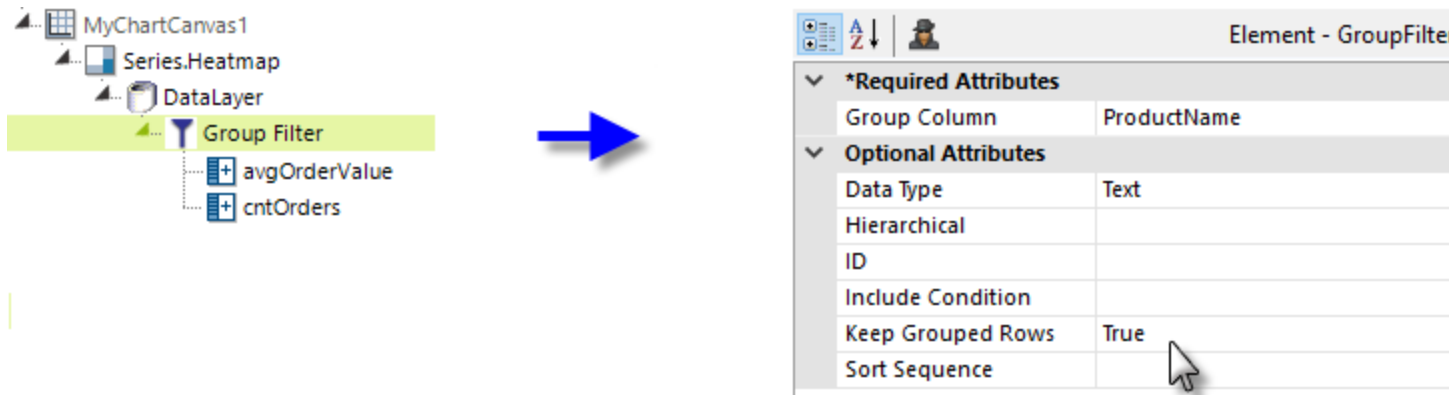


Chart Canvas Charts that use the Chart Drill To element can only have a single Series. The Drill Back button only appears when the mouse cursor is hovered over the upper right-hand corner of the chart.

Or, continue drilling further down into the data (assuming the elements are configured for it). For example, you could select the **Germany** rectangle next and then select **City** to drill down to the cities with customers who bought the selected product in that country.

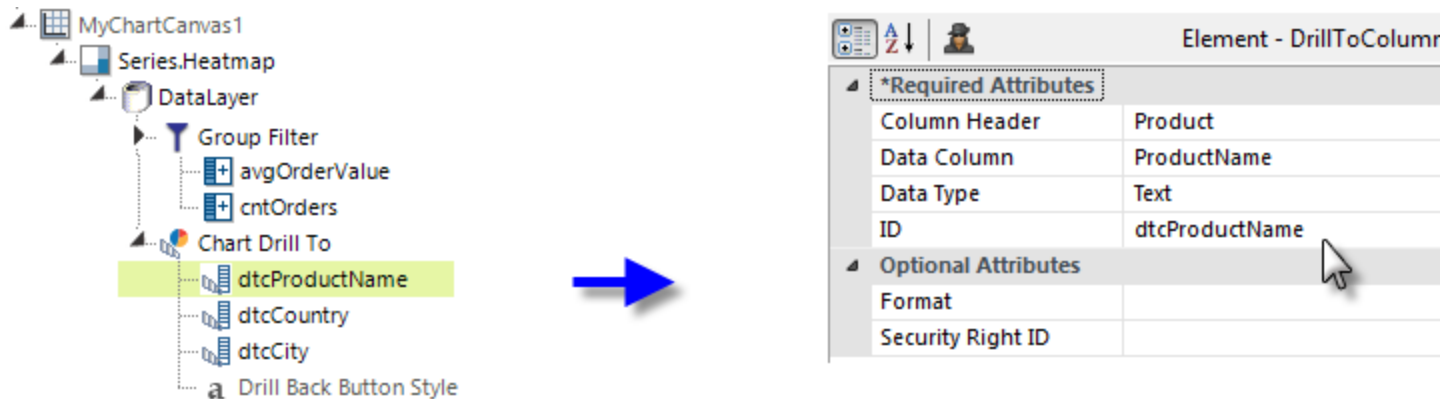
If your chart has been configured for it, as you drill further into the data, a breadcrumb trail becomes available. The breadcrumb trail can be used to access previously drilled levels, or clear the Drill To data altogether. To enable this feature, set the **Show DrillTo Breadcrumb** attribute to *True*.


As shown below, the Chart Drill To element drills into *grouped* data, so the datalayer used beneath the Chart Canvas Chart or Series element *must* be grouped using a **Group Filter** or **Sql Group** element. The example datalayer has been grouped on the *CategoryName* column and the Keep Grouped Rows attribute was set to *True*. The Series itself has been configured so that its Label Data Column = *CategoryName* and its Size Data Column = *avgOrderValue* (the Group Aggregate Column that averages the order values). Its Color Data Column = *cntOrders* (the Group Aggregate Column that counts the orders).



Next, a **Chart Drill To** element was added beneath the Series element, shown below. Required child **Drill To Column** elements have also been added; they define the columns the user can select to drill into. They should be added and configured for columns

that can be reasonably grouped, such as text-type columns with a limited number of unique values and date-type columns.



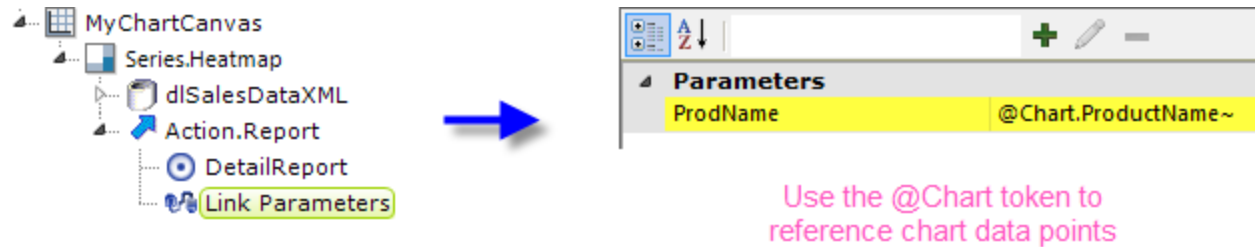
 You *must* also configure a Drill To Column element for the column that's specified as the chart's Label Data Column, in this example the *CategoryName* column.

The **Column Header** attribute specifies the text that will appear in the Drill to Column drop-down list options. The vertical order of the drop-down list options will match the vertical order of the Drill To Column elements in the definition. Year, Quarter, and Month options will be added automatically for Drill To Columns with Data Type = *Date*. You do not need to add any filter or additional grouping elements to achieve the results shown in the example above.

The Drill Back button displayed on the chart after drilling has occurred can be styled using the **Drill Back Button Style** element.

Series.Heatmap - Using Action Elements

Action elements initiate processing of a report or process task definition, redirection to a link, or other processing when a data point in the series is clicked.



In the example above, an **Action.Report** element has been added as a child of the Series, along with its **Target.Report** and **Link Parameters** child elements. To reference chart data in parameters, use the @Chart token, as shown above.

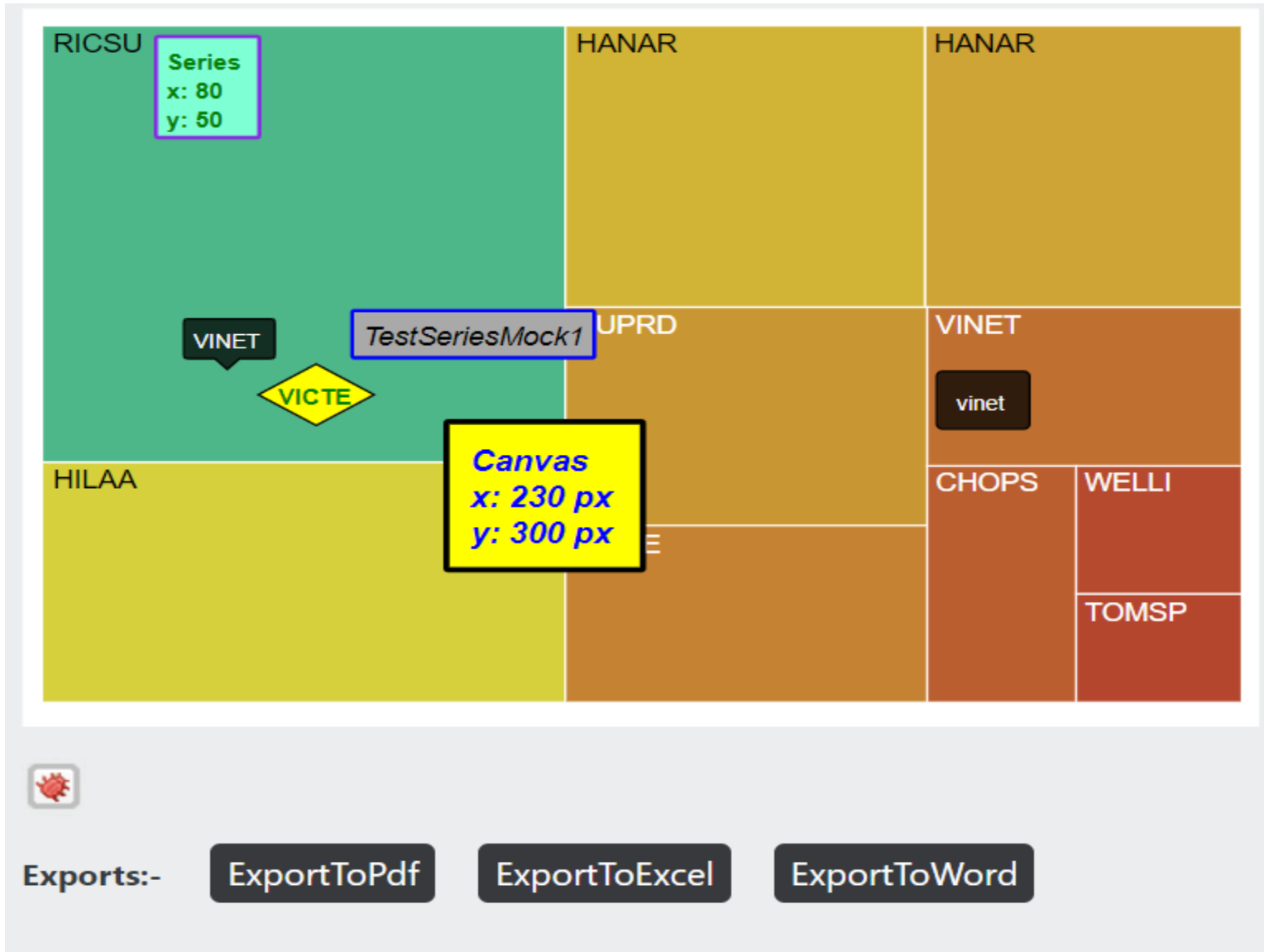
A variety of Action elements are available for use with the Series, including Action.Link, Action.Process, and Action.Refresh Element. Additional Action elements will be added in future releases.

Action elements can be also used beneath the Heatmap Group element to provide drill-down and drill-through functionality when group titles are clicked.

v23.1

Series.Heatmap - Using the Series Annotation Element

Annotations allow you to annotate a chart freely using custom labels and shapes. When the **Series Annotation** element is used as a child of Series.Heatmap, you can place these annotations at various points of interest:



The Series Annotation element offers two child elements: **AnnotationLabel.Point** and **AnnotationLabel.Mock**.

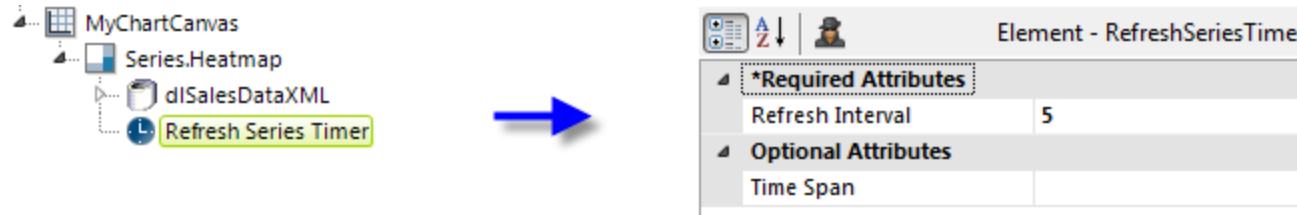
The `AnnotationLabel.Point` child element is used to create an annotation label and position on the chart by linking it to an existing point. With this element, you cannot specify the value, it always refers to the value of the point. Additionally, you can use the `Caption` and `Condition` attributes to refer to different datalayers.

On the other hand, the `AnnotationLabel.Mock` is used to create an annotation label and position it on the chart by linking it to a created mock point. With this element, you can specify the coordinate, relative position, and axis.

Both the `AnnotationLabel.Point` and `AnnotationLabel.Mock` have attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text.

Series.Heatmap - Using the Refresh Series Timer

The **Refresh Series Timer**, added as a Series child element, updates charts automatically, based on a time interval. When the interval is reached, a request is sent back to the web server for updated data. Series are updated with a smooth animation.



The **Refresh Interval** attribute specifies the number of seconds to wait before refreshing the series data. A value is required and must be an integer greater than 0.

Refresh Mode

Series may be refreshed in two ways: either all of the data is refreshed with each request, or the series data automatically slides to the left one data interval per refresh. The refresh mode is selected in the Chart X Axis element's **Axis Type** attribute. When the Axis Type is *not* set to *DateTimeLinear*, all of the data will be refreshed.

When the Axis Type value is *DateTimeLinear*, newer values are added to the right side of the chart, previous values slide left, and older values disappear as they reach the left edge of the chart. In this case, the Refresh Series Timer element's **Time Span** attribute is used to specify the age of the data included. This value must be in *hh:mm:ss* format (for example "00:02:00" for two minutes) and must be larger than the Refresh Interval attribute value.

When a Time Span attribute is set, the series' datalayer can make use of a special timestamp token, `@Chart.rdTimeSpanStart~`, in

a query to retrieve just the rows necessary to update the chart. On the initial request, this token returns the current time minus the Time Span value. For subsequent refreshes, when *Axis Type = DateTimeLinear* and the X-axis will be sliding, the token returns the last time the chart was updated, so only the newest rows are retrieved. Here's an MS SQL Server query example:

```
SELECT * FROM MyTable WHERE UpdateTime BETWEEN '@Chart.rdTimeSpanStart~' AND '@Function.DateTime~'
```

For best performance, avoid setting a very short refresh interval if a very large number of users will be displaying the report.

Series.Line

The Chart Canvas element's Series child elements cause a data visualization (the chart) to be rendered in the canvas.

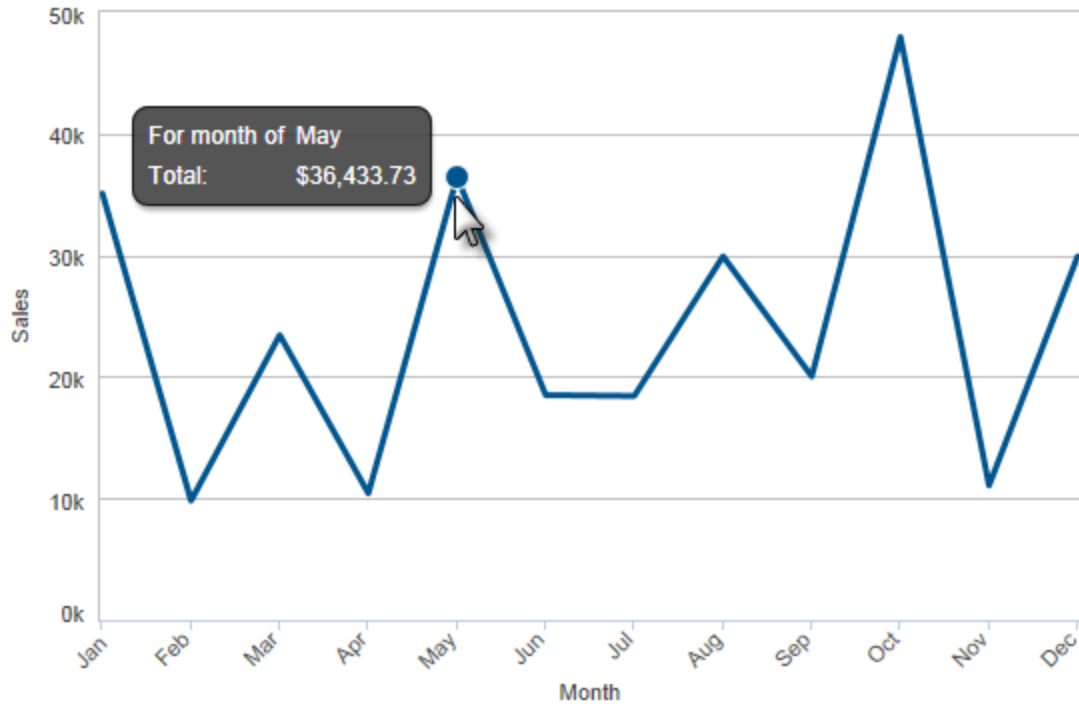
The following topics discuss the Series.Line child element:

- [Using Multiple Series](#)
- [Series.Line Attributes](#)
- [Using the Data Labels Element](#)
- [Using the Marker Points Element](#)
- [Using the Quicktips Element](#)
- [Using the Trend Line Element](#)
- [Using Action Elements](#)
- [Using the Series Annotation Element](#) v23.1
- [Using Input Selection](#)
- [Using the Refresh Series Timer](#)

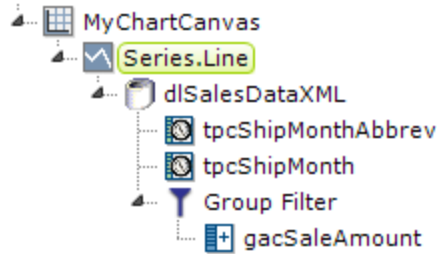
About Series.Line

The **Series.Line** element generates a Line chart, which is commonly used to represent aggregated totals, as numbers or percentages, over time.

Monthly Sales 2013



The example above shows a simple Line chart, representing sales per month for a year.

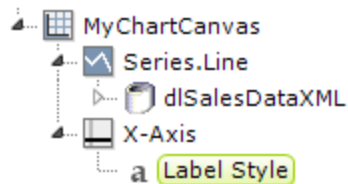


| Element - Series.Line | |
|-----------------------------|--------------------|
| *Required Attributes | |
| Y-axis Data Column | gacSaleAmount |
| Optional Attributes | |
| Color | |
| Combine With Series ID | |
| ... | |
| X-axis Data Column | tpcShipMonthAbbrev |
| X-axis Data Column Type | |

As shown above, the chart is created by adding Series.Line to the canvas, along with a datalayer and, typically, some child elements that may include **Time Period Column** elements, a **Group Filter**, and a **Group Aggregate Column** element. Very few attributes need to be set for the Series element in order to produce a basic chart.



A datalayer element can be used either beneath Series.Line, as shown above, or beneath Chart Canvas. If used as a child of Chart Canvas, its data is available to *all* child Series elements. This can improve performance if you have several series, all using the *same* data, beneath the same Chart Canvas element.

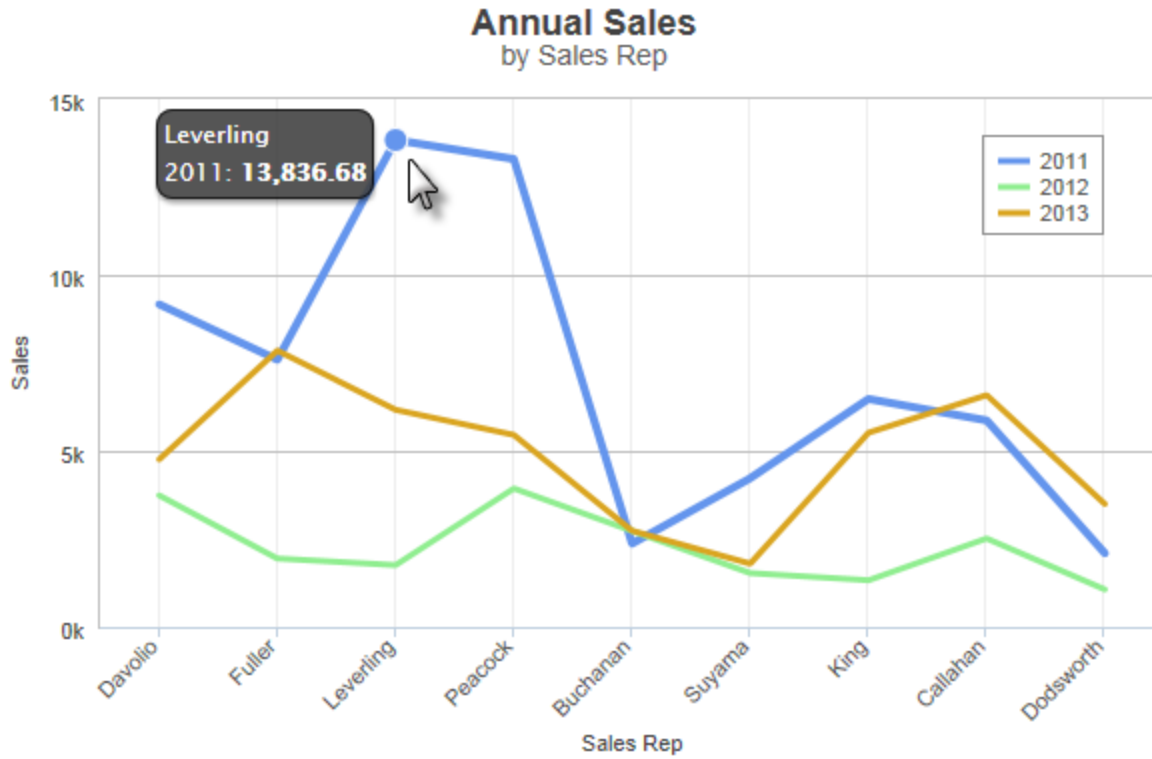


| Element - AxisLabelStyle | |
|----------------------------|----|
| Optional Attributes | |
| Font Angle | 45 |
| Font Color | |
| ... | |
| Stagger Labels | |

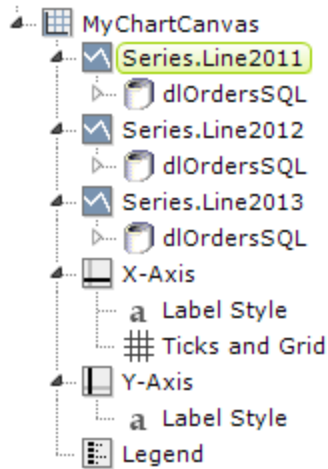
The properties of the X- and Y-axis, including captions, are set using the **X-Axis** and **Y-Axis** elements. For example, in order to angle the X-axis labels, add an X-Axis element beneath Chart Canvas (none of its attributes need to be set) and add its child **Label Style** element. Set the Label Style element's attribute as shown above.

Series.Line - Using Multiple Series

You can add additional series to the chart by adding additional Series elements:

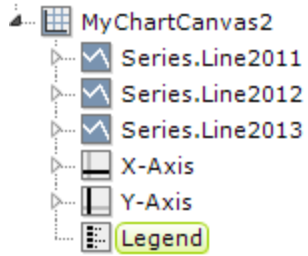


The example above shows three Series, one for each year, with a legend.



| Element - Series.Line | |
|-----------------------------|-----------------|
| *Required Attributes | |
| Y-axis Data Column | sumSales |
| Optional Attributes | |
| Color | CornflowerBlue |
| Combine With Series ID | |
| ... | |
| ID | Series.Line2011 |
| Legend Label | 2011 |
| Line Style | |
| Line Thickness | 3 |
| ... | |
| X-axis Data Column | LastName |
| X-axis Data Column Type | |


The example above shows the three Series elements, their datalayers, the X- and Y-Axis elements, and the **Legend** element used to produce the previous chart. You can adjust which series appears "in front" of the others in the chart by changing the order of the Series elements in the definition. Setting the Series elements' **Legend Label** attribute will automatically cause the legend to be displayed.



| Optional Attributes | |
|----------------------|----------|
| Alignment Horizontal | |
| ... | |
| Border Radius | 0 |
| Border Thickness | |
| ... | |
| Inside Plot Area | True |
| Legend Orientation | Vertical |
| Maximum Height | |
| Offset X | -30 |
| Offset Y | 65 |
| ... | |
| Width | |

In order to produce the legend shown in the previous example, a **Legend** element must be added and its attributes configured as shown above.

Charts with multiple series or aggregations are encouraged to use the ChartCanvas element's attribute, `Tooltip Split`. This attribute allows you to split a chart's tooltip into one label per series, rather than per segment. The default value for this attribute is *False*. Once enabled, when hovering over a data point, it displays all the values for the series. This method is recommended over shared tooltips for charts with multiple line series, and as such, takes precedence over `tooltip.shared`.

 When using multiple series, you may be able to reduce the number of data queries and improve performance by using local data to read all of the data once, see *Datalayer Introduction*. Then, link its datalayer to share it to the series, see *Link Datalayers*. At each Series element, link its datalayer to the shared Local Data datalayer and filter the data to meet the needs of each individual series.

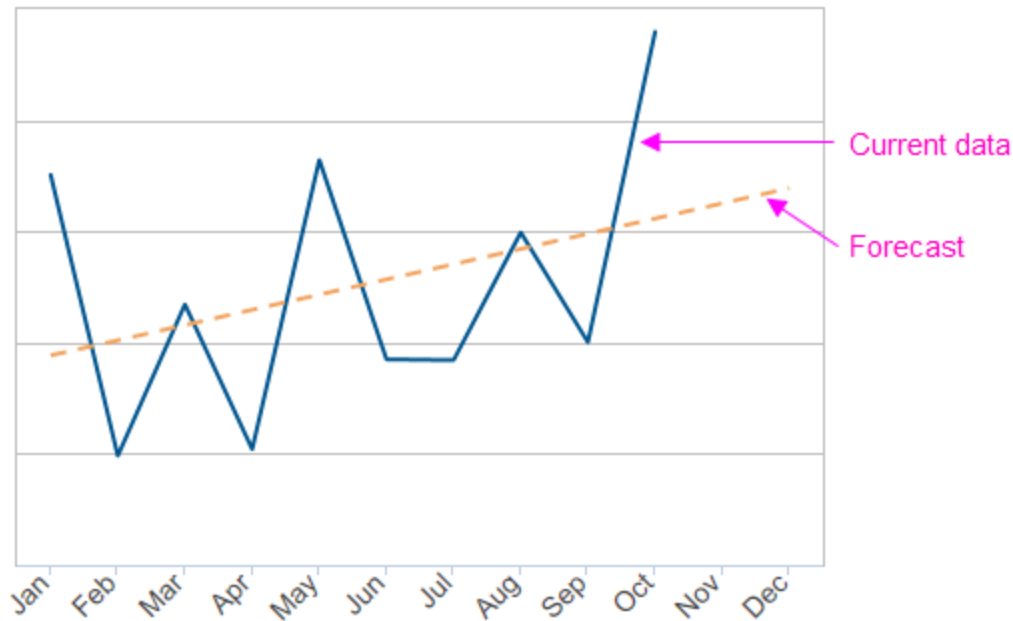
You can combine different types of Series elements, for example, Series.Line and Series.Bar, to produce combinations of visualizations.

v23.1 If you are using the Chart Color Axis element in a mult-series chart, by default, the series will link to the first color axis.

To link the series to a special color axis, set the ID attribute of the Chart Color Axis element to the corresponding Series' Linked to Color Axis ID attribute. For more information, see "Chart Color Axis" on page 124.

Forecasting

Forecasting elements use a variety of techniques to produce projected values by analyzing existing values. The future values they "predict" are, in most cases, added as rows or columns to a datalayer so the data can be displayed along with the existing data. When using Chart Canvas charts, the forecast data is typically displayed using a Series.Line element, in conjunction with other series elements.



Forecasting elements add a "forecast value" column to the datalayer, and this column is used as the series' Y-axis data column. For more information about using forecasting elements, see *The Forecasting Elements*.

Series.Line - Attributes

The Series.Line element has the following attributes:

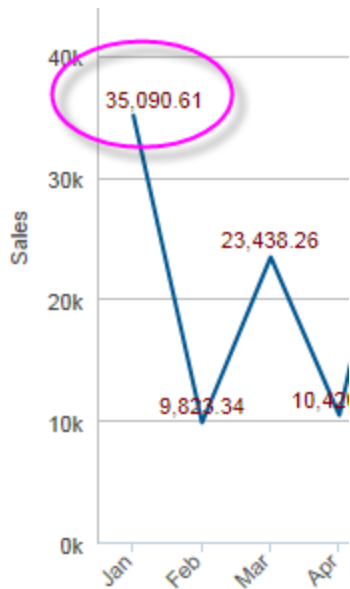
| Attribute | Description |
|--|---|
| Y-Axis Data Column | (Required) Specifies the name of a datalayer column whose values will be plotted along the Y-axis. |
| v23.1 Class Name | Specifies the Class Name referenced in the .css style sheet. The default value is <i>undefined</i> . |
| Color | Sets the line color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. @Chart tokens may be used here to provide dynamic, data-driven line segment colors. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| v23.1 Color Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the x-axis. |
| Combine With Series ID | Set this attribute to the element ID of another series to combine it with this series in the legend. When two series are combined, by default only the first one will appear in the legend but clicking the item in the legend toggles both series to appear and disappear. Or, the value <i>Previous</i> can be entered to combine this series with the previous series. |
| Connect Nulls | Specifies if data rows with null or blank values are to be ignored, allowing adjacent values to be connected in the chart. The default value is <i>False</i> . |

| Attribute | Description |
|--|--|
| Disable Mouse Tracking | Disables mouse tracking for the series, when set to <i>True</i> . This affects tooltips and click events on graphs and points. For large datasets, this may improve performance. The default value is <i>False</i> . |
| Hover Line Thickness | Sets the thickness of the line, in pixels, when the mouse pointer is hovered over it. The default value is <i>1</i> pixel. |
| Legend Label | Indicates text that will be shown for this series inside the chart legend. When a value is provided, automatically causes the legend to be displayed. |
| Line Style | Specifies the pattern of the line as either <i>Solid</i> or a combination of dashes and dots. @Chart tokens may be used here to provide dynamic, data-driven line segment styles. |
| Line Thickness | Specifies the thickness of the line, in pixels. The default value is <i>1</i> pixel. |
| v23.1 Linked to Color-Axis ID | Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes |
| Linked to X-Axis ID | Specifies the ID of an X-Axis element that this series should be linked to when using multiple X-axes. |
| Linked to Y-Axis ID | Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes. |

| Attribute | Description |
|-----------------------------|---|
| Negative Color | Sets the color for negative values. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #D5F484. The default value is <i>Red</i> . |
| Negative Color Transparency | Specifies the transparency of the Negative Color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| Negative Threshold | Sets the positive-negative value threshold, also called the "zero level" or "base level". The default value is <i>0</i> . |
| Transparency | Specifies the transparency of the line color. The lowest value of <i>0</i> specifies that the line is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent line. Use medium-level transparency to allow different chart layers to show through each other. |
| X-Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the X-axis. |
| X-Axis Data Column Type | Specifies the data type of the datalayer column named in the X-axis Data Column attribute. Options include <i>Auto</i> (the default), <i>Text</i> , <i>Number</i> , and <i>DateTime</i> . By default, X-axis data values that are <i>DateTime</i> type will be automatically distributed evenly across the time series. If you want to disable this behavior, set this attribute value to <i>Text</i> . |

Series.Line - Using the Data Labels Element

A "data label" is text shown next to each data point that shows its value. When the **Data Labels** element is used as a child of Series.Line, text representing the data values will appear on the chart:

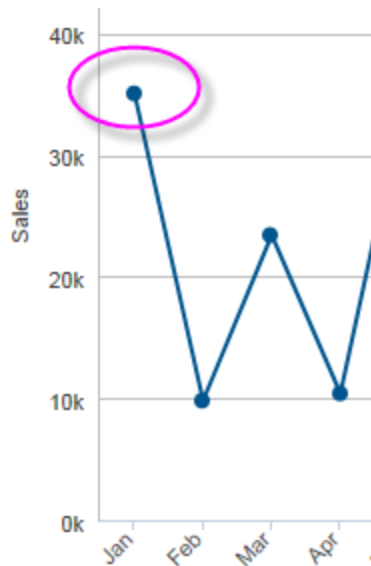


The Data Labels element has attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text. One attribute, Series Name as Caption, allows you to specify the y-axis label, x-axis label, or legend label as the data point value inside the chart.

The Data Labels element's color-related attribute values can be set using @Chart tokens. v23.1 Or, utilize the "Class Name" attribute to apply unique styling to your data labels by referencing a class name from a stylesheet in the report.

Series.Line - Using the Marker Points Element

A "marker point" is a symbol that appears on the chart at each data point. When the **Marker Points** element is used as a child of Series.Line, a small dot matching the color of the line will be displayed at each data point:



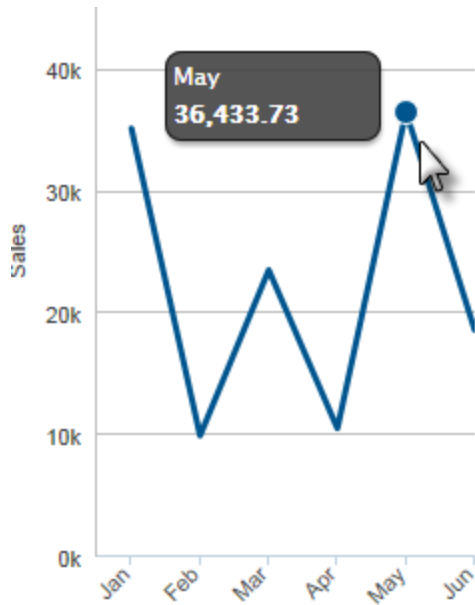
The default marker points are shown above. The **Marker Points** element allows you to select a different symbol for the marker point, and to control its size, color, border color, and transparency. When the cursor hovers over a marker point, it's automatically enlarged slightly.

v23.1

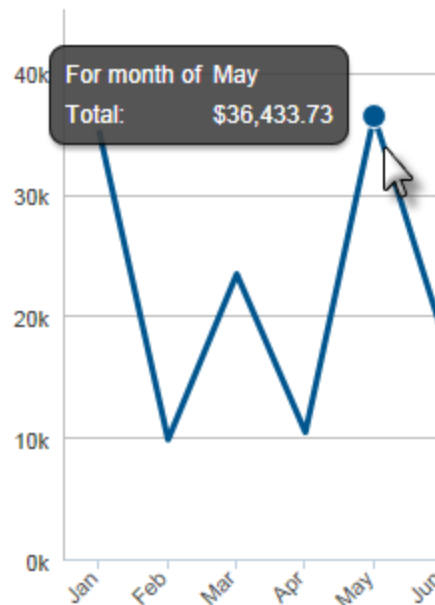
You can also create your own marker point symbol using the Chart Canvas child element, Chart Custom Symbol. Once defined, link your custom symbol using the Marker Points element's Symbol attribute. For more information, see "Chart Custom Symbol" on page 128.

Series.Line - Using the Quicktips Element

By default, a "quicktip" is displayed when the mouse hovers near or over a data point:




Default quicktip



With Quicktip child element

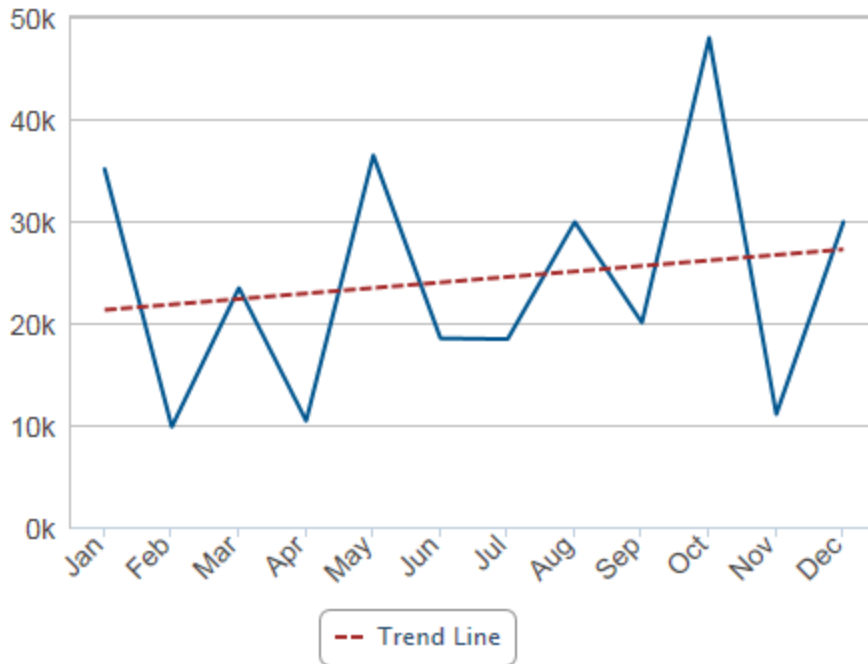
The automatically-generated quicktip displays information for the X- and Y-axis, as shown above, left. However, you may want to display other information or format it differently, perhaps as shown above, right, which can be done by adding a **Quicktip** child element beneath Series.Line and setting its attributes and child elements. Use @Chart tokens to include chart data in the quicktip. Intrinsic functions are supported in the Quicktip attributes.

You can also configure the tooltip to display values for any available column in the datalayer. To do so, enter a token representing the data column in the Value attribute of the Quicktip Row element.

 To use this feature with DataLayer.ActiveSQL, please make sure the keep Grouped Rows attribute of the SqlGroup element is set to *False*.

Series.Line - Using the Trend Line Element

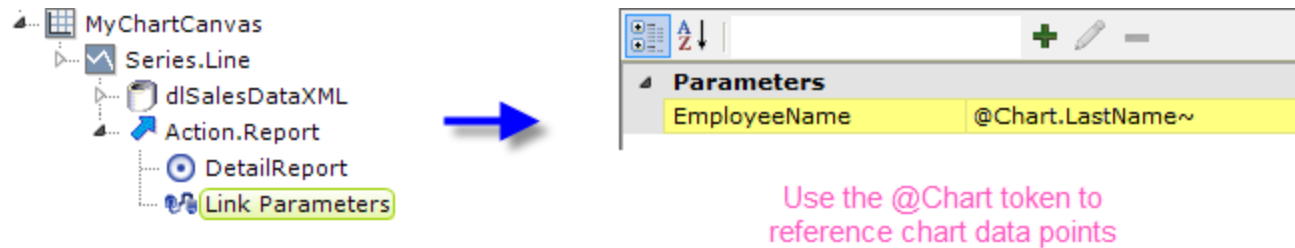
The **Trend Line** element creates a line on the chart that indicates the "trend" of the data. The line connects a number of data points generated using a regression algorithm.



The Trend Line element is a child of the Series.Line element and can be styled for color, line width, etc. When configured with a legend label, it will be represented by an item in the legend, as shown above.

Series.Line - Using Action Elements

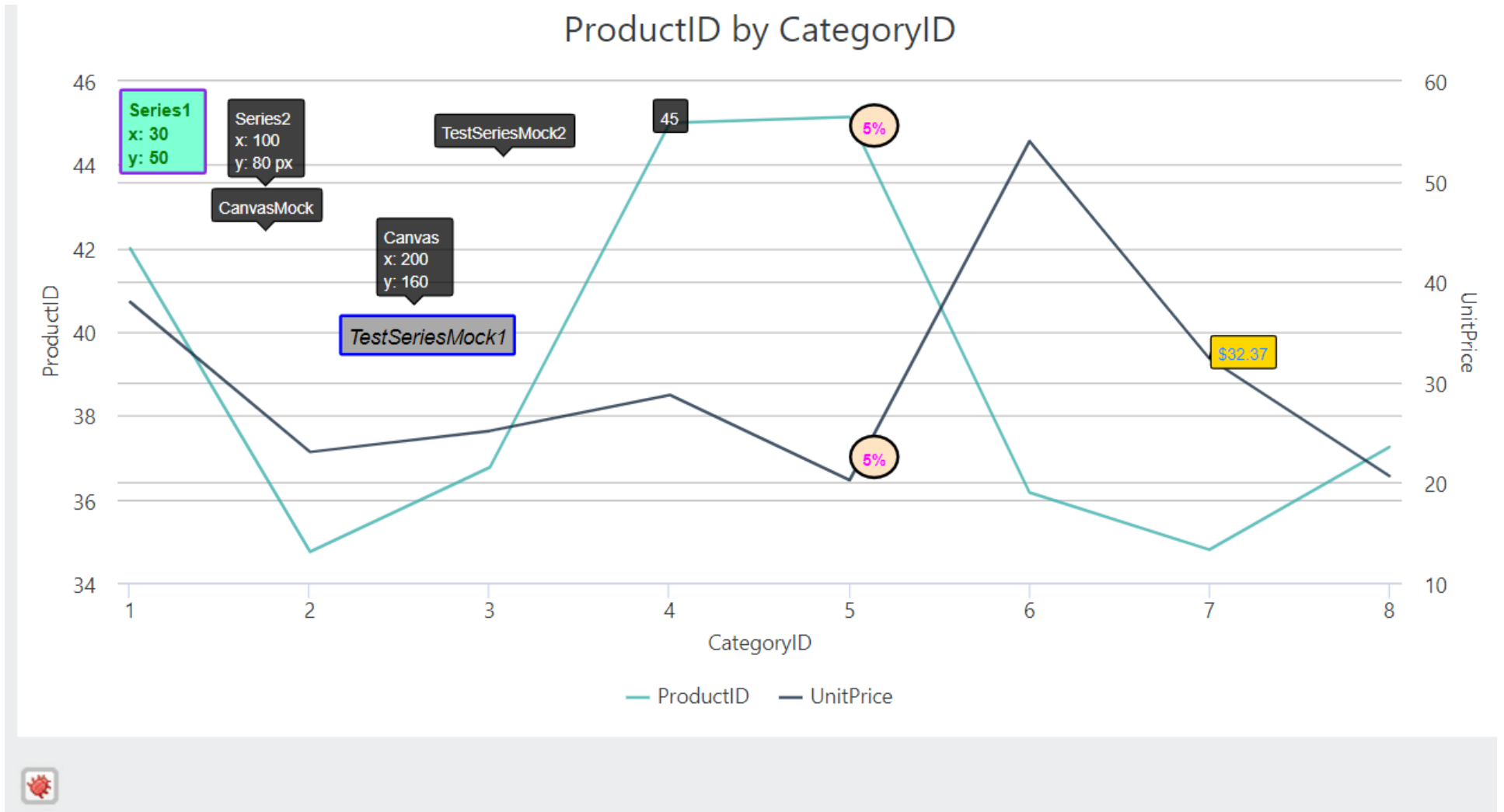
Action elements initiate processing of a report or process task definition, redirection to a link, or other processing when a data point in the series is clicked.



In the example above, an **Action.Report** element has been added as a child of the Series, along with its **Target.Report** and **Link Parameters** child elements. To reference chart data in parameters, use the @Chart token, as shown above. A variety of Action elements are available for use with Series, including Action.Link, Action.Process, and Action.Refresh Element. Additional Action elements will be added in future releases.

Series.Line - Using the Series Annotation Element

Annotations allow you to annotate a chart freely using custom labels and shapes. When the **Series Annotation** element is used as a child of Series.Line, you can place these annotations at various points of interest:



The Series Annotation element offers two child elements: **AnnotationLabel.Point** and **AnnotationLabel.Mock**.

The AnnotationLabel.Point child element is used to create an annotation label and position on the chart by linking it to an existing point. With this element, you cannot specify the value, it always refers to the value of the point. Additionally, you can use the Caption and Condition attributes to refer to different datalayers.

On the other hand, the AnnotationLabel.Mock is used to create an annotation label and position it on the chart by linking it to a created mock point. With this element, you can specify the coordinate, relative position, and axis.

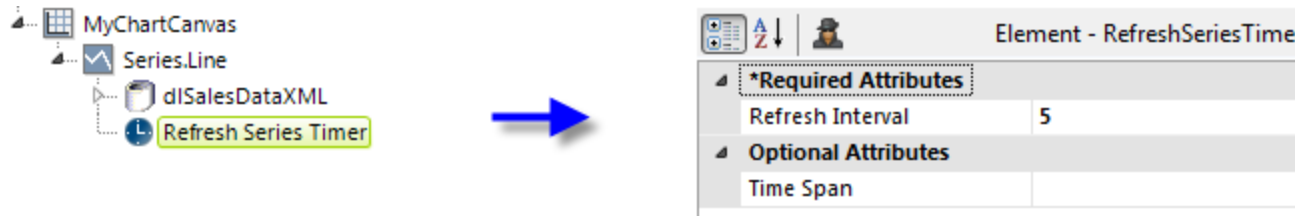
Both the AnnotationLabel.Point and AnnotationLabel.Mock have attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text.

Series.Line - Using Input Selection

This series can also be used with the **Input Selection** family of elements, which turn the chart into an input control. This allows users, at runtime, to select points or values on the chart with their mouse and your report can then take action using the selected data. This is very useful, for example, for drilling into the data. For more information about this functionality, see "Input Selection" on page 97.

Series.Line - Using the Refresh Series Timer

The **Refresh Series Timer**, added as a Series child element, updates charts automatically, based on a time interval. When the interval is reached, a request is sent back to the web server for updated data. Series are updated with a smooth animation.



The **Refresh Interval** attribute specifies the number of seconds to wait before refreshing the series data. A value is required and must be an integer greater than 0.

Refresh Mode

Series may be refreshed in two ways: either all of the data is refreshed with each request, or the series data automatically slides to the left one data interval per refresh. The refresh mode is selected in the Chart X Axis element's **Axis Type** attribute. When the Axis Type is *not* set to *DateTimeLinear*, all of the data will be refreshed.

When the Axis Type value is *DateTimeLinear*, newer values are added to the right side of the chart, previous values slide left, and older values disappear as they reach the left edge of the chart. In this case, the Refresh Series Timer element's **Time Span** attribute is used to specify the age of the data included. This value must be in *hh:mm:ss* format (for example "00:02:00" for two minutes) and must be larger than the Refresh Interval attribute value.

When a Time Span attribute is set, the series' datalayer can make use of a special timestamp token, `@Chart.rdTimeSpanStart~`, in a query to retrieve just the rows necessary to update the chart. On the initial request, this token returns the current time minus the Time Span value. For subsequent refreshes, when *Axis Type = DateTimeLinear* and the X-axis will be sliding, the token returns the last time the chart was updated, so only the newest rows are retrieved. Here's an MS SQL Server query example:

```
SELECT * FROM MyTable WHERE UpdateTime BETWEEN '@Chart.rdTimeSpanStart~' AND '@Function.DateTime~'
```

For best performance, avoid setting a very short refresh interval if a very large number of users will be displaying the report.

Series.Pie

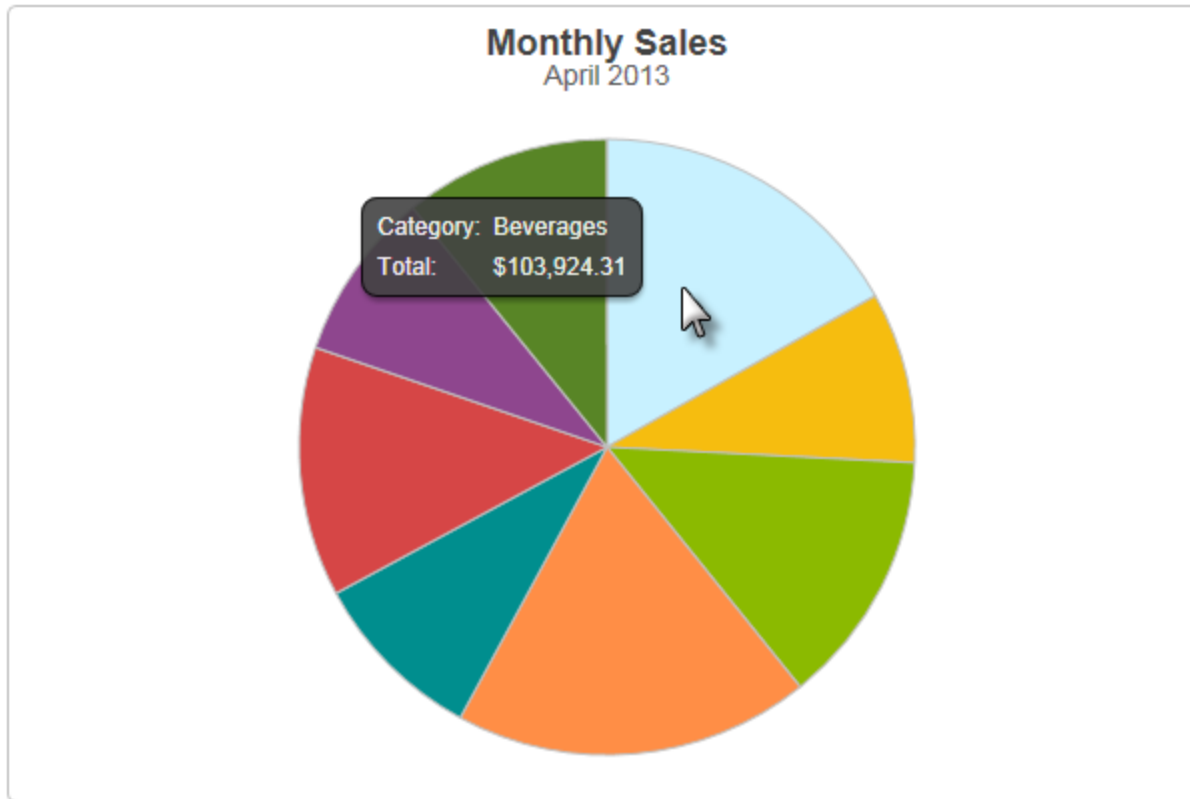
The Chart Canvas element's Series child elements cause a data visualization (the chart) to be rendered in the canvas.

The following topics discuss the Series.Pie child element:

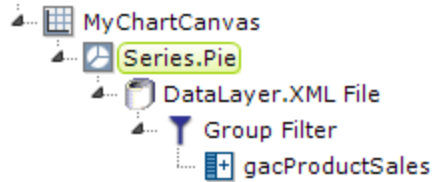
- [Shaping the Pie](#)
- [Using Multiple Series](#)
- [Series.Pie Attributes](#)
- [Adding Data Labels](#)
- [Using the Quicktips Element](#)
- [Using the Chart Drill To Element](#)
- [Using Action Elements](#)
- [Using the Series Annotation Element](#) v23.1
- [Using Input Selection](#)
- [Using the Refresh Series Timer](#)

About Series.Pie

The **Series.Pie** element generates a Pie chart, which is a circular chart divided into sectors, illustrating numerical proportion.



The example above shows a simple Pie chart, presenting one month's sales of different product categories.



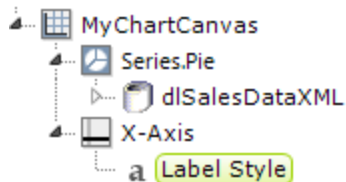
Wedge colors can be specified as a comma-separated list, using names, decimal RGB values, or hex RGB values

| *Required Attributes | |
|--------------------------|----------------------|
| Y-axis Data Column | gacProductSales |
| Optional Attributes | |
| Angle End | |
| Angle Start | |
| Border Color | Silver |
| ... | |
| Colors | #AFD8F8,#F6BD0F,#8BB |
| ... | |
| Label Data Column X-axis | CategoryName |
| ... | |
| Transparency | |

As shown above, the chart is created by adding Series.Pie to the canvas, along with a datalayer and, typically, some child elements that may include **Time Period Column** elements, a **Group Filter**, and a **Group Aggregate Column** element. Very few attributes need to be set for the Series element in order to produce a basic chart. In the example, a list of custom wedge colors was provided.



A datalayer element can be used either beneath Series.Pie, as shown above, or beneath Chart Canvas. If used as a child of Chart Canvas, its data is available to *all* child Series elements. This can improve performance if you have several series, all using the *same* data, beneath the same Chart Canvas element.

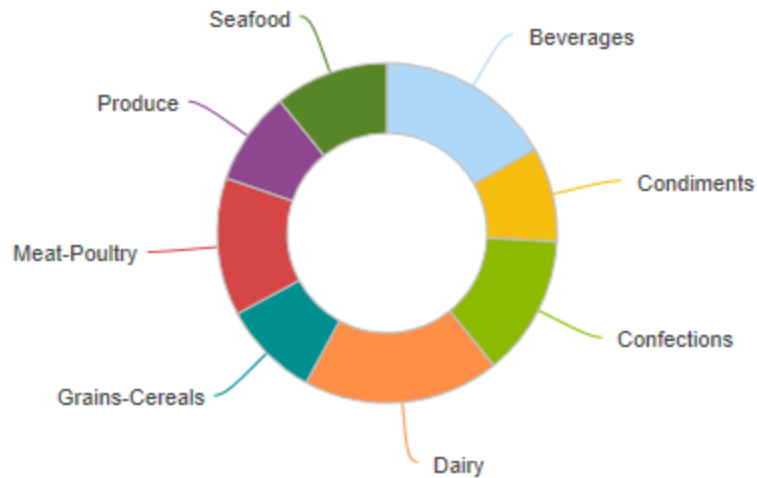


| Optional Attributes | |
|---------------------|----|
| Font Angle | 45 |
| Font Color | |
| ... | |
| Stagger Labels | |

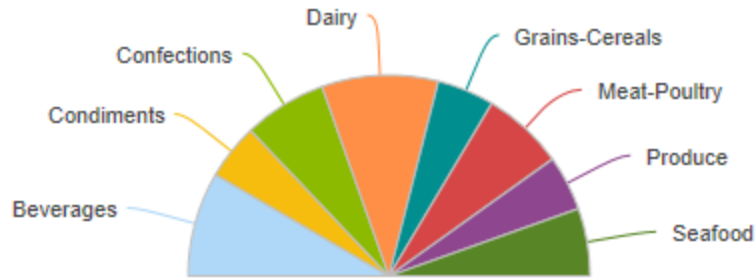
The properties of the X- and Y-axis, including captions, are set using the **X-Axis** and **Y-Axis** elements. For example, in order to angle the X-axis labels, add an X-Axis element beneath Chart Canvas (none of its attributes need to be set) and add its child **Label Style** element. Set the Label Style element's attribute as shown above.

Series.Pie - Shaping the Pie

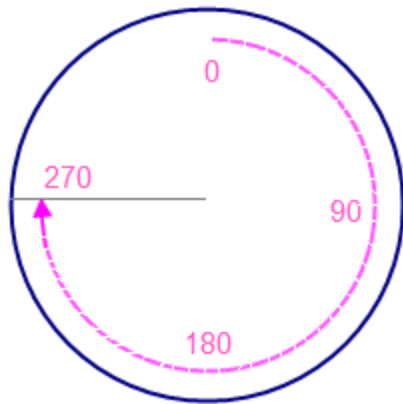
You can control the shape of the Pie chart in several ways.



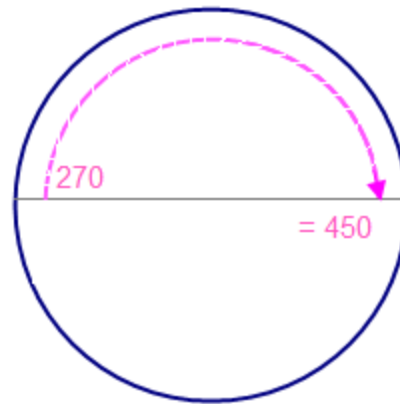
The example above shows a Pie chart with a "donut hole" in its middle. This is controlled by setting the Series.Pie element's **Donut Hole Size** attribute to a value, in pixels, equal to the diameter of the hole.



The example above shows a Pie chart that only fills part of the usual 360-degree pie space. This is created by using the Series.Pie element's **Angle Start** and **Angle End** attributes. The example uses Angle Start = 270 degrees and Angle End = 450 degrees. How do we come up with those numbers?



Angle Start = 270 degrees

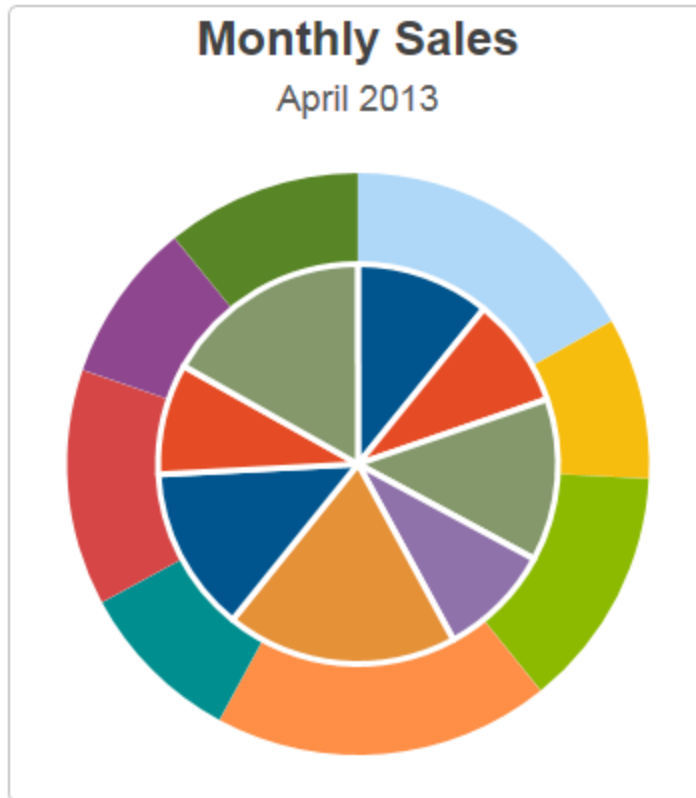


Half-circle = 180 degrees,
Angle End = 270 + 180 = 450

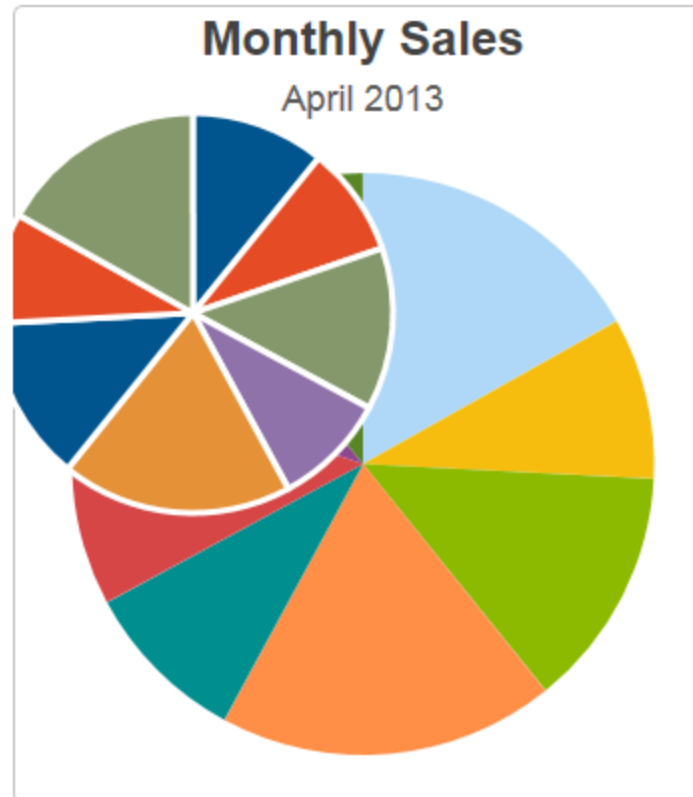
Assume we want a Pie chart that's a half-circle, above the equator, like our previous example. To find the starting angle, we navigate clockwise around a circle to the starting point, as shown above left, to 270. Then to find the ending angle, we add the number of degrees in a half-circle (180) to the starting angle, or $270 + 180$, which equals 450, as shown above, right. Using this method, you can create a Pie chart that's any portion of a complete circle.

Series.Pie - Using Multiple Series

Just as it is with other Series, it's possible to add multiple Pie charts to one Chart Canvas.




Two Series, one over the other



Second series offset using
Center X and Center Y attributes

Overlaying Pie charts can produce "visual clutter" that's hard to decipher (and that's without any data labels displayed), as you can see in the example above, left. In the example above, right, however, the second Series has been configured so that it's off center, using its **Center X** and **Center Y** attributes, which provides a little more clarity.

 When using multiple series, you may be able to reduce the number of data queries and improve performance by using local data to read all of the data once, see *Datalayer Introduction*. Then, link its datalayer to share it to the series, see *Link Datalayers*. At each Series element, link its datalayer to the shared Local Data datalayer and filter the data to meet the needs of each individual series.

You can combine different types of Series elements, for example, Series.Pie and Series.Line, to produce combinations of visualizations.

v23.1 If you are using the Chart Color Axis element in a mult-series chart, by default, the series will link to the first color axis. To link the series to a special color axis, set the ID attribute of the Chart Color Axis element to the corresponding Series' Linked to Color Axis ID attribute. For more information, see "Chart Color Axis" on page 124.

Series.Pie - Attributes

The Series.Pie element has the following attributes:

| Attribute | Description |
|---------------------------|--|
| Y-Axis Data Column | (Required) Specifies the name of a datalayer column whose values will be plotted along the Y-axis, dictating how large each pie slice is. |
| Angle End | Sets the ending angle of the Pie chart's <i>last</i> slice, in degrees. For example, to create a Pie chart which has 180 degrees total, facing upwards, set these attributes: Angle Start = 270, Angle End = 450 |
| Angle Start | Sets the starting angle of the Pie chart's <i>first</i> slice, in degrees. The default value of 0 starts the first slice at the 12 o'clock position and a value of 90 is the 3 o'clock position. |
| Border Color | Sets the color of the border line around the pie and around each slice. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, like #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| Border Color Transparency | Specifies the transparency of the border line. The lowest value of 0 specifies that the background is opaque, with no transparency. At the other end of the scale, 15 specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| Border Thickness | The thickness of border line around the pie and around each slice, in pixels. The default is value is 1 pixel. |
| Center X | Sets the horizontal center point of the Pie chart, in pixels, within the canvas. |

| Attribute | Description |
|--|--|
| Center Y | Sets the vertical center point of the Pie chart, in pixels, within the canvas. |
| Colors | Sets the colors of the pie slices, which should be entered as a comma-separated list. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. When there are more slices than colors specified, the listed colors are used again from the beginning. A default set of colors is used if this is left blank and no theme has been applied, see "Chart Canvas Charts" on page 20. |
| v23.1 Color Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the x-axis. |
| Disable Mouse Tracking | Disables mouse tracking for the series, when set to <i>True</i> . This affects tooltips and click events on graphs and points. For large datasets, this may improve performance. The default value is <i>False</i> . |
| Donut Hole Size | Sets the size of the inner diameter of the Pie chart, in pixels. Any size greater than 0 renders a "doughnut" style chart. The default value is 0pixels. |
| Hover Bright- ness | Specifies the amount to change a pie slice's color when the mouse pointer is hovered over it. Values can be between 0 (no change) and 15(lighter). The default value is 2. |
| Label Data Column X-Axis | Set this to a column returned from the DataLayer. It represents the "name", or x-axis value, of each pie slice. |

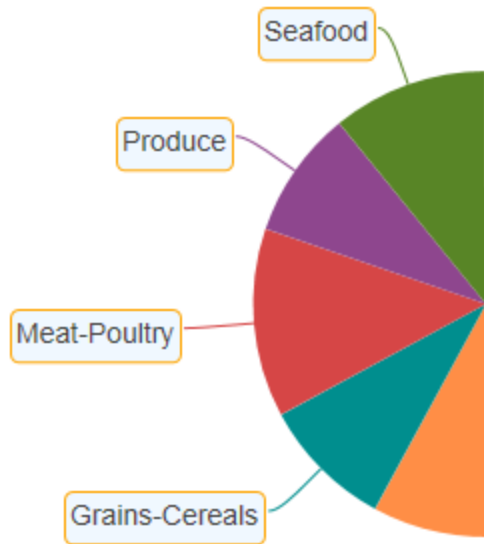
| Attribute | Description |
|---|---|
| Label Location | <p>Specifies how Pie chart slice labels will appear. Select:</p> <ul style="list-style-type: none"> - <i>SideLayout</i> to have labels appear outside the pie, with arrows connecting them to slices. - <i>Legend</i> to have slices identified in a legend. The legend can be clicked to toggle the visibility of individual slices in the pie. - <i>NoLabels</i> to hide the labels altogether. <p>The default value is <i>SideLayout</i>.</p> |
| v23.1 Linked to Color-Axis ID | <p>Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes</p> |
| Min Radius | <p>Sets the minimum acceptable size of the Pie chart, in pixels. During rendering, the pie radius will try to shrink automatically, if necessary, to accommodate any exterior slice labels and still fit the canvas, but it can be no smaller than this size.</p> |
| Pie Label Style | <p>Specifies how Pie chart slice labels will appear. Select:</p> <ul style="list-style-type: none"> - <i>SideLayout</i> to have labels appear outside the pie, with arrows connecting them to slices. - <i>Legend</i> to have slices identified in a legend. The legend can be clicked to toggle the visibility of individual slices in the pie. - <i>NoLabels</i> to hide the labels altogether. |

| Attribute | Description |
|-------------------------|--|
| | The default value is <i>SideLayout</i> . |
| Radius | Determines the size, by setting the radius, of a Pie chart, in pixels. If left blank, the chart size is determined automatically. |
| Show Data Values | Specifies if the value of each data point should be shown on the Pie chart. Depending on the value of the Label Location attribute, they may be shown alone or with the label values. The default value is <i>False</i> . |
| Show Data Values Format | <p>Specifies formatting characteristics for data values. For dates, the non-specific formats, such as <i>General Date</i>, <i>Short Time</i>, etc., are converted according to the browser's international settings. For very large reports, the non-specific formats perform better. Special formats include:</p> <p>< and > - change strings to lower and upper case.</p> <p><i>Expanded Spaces</i> - preserves space characters that would otherwise be collapsed by the web browser.</p> <p><i>mp</i> - formats numbers with the "metric prefix". For example, to format 1,234,567 as "\$1.23M", enter <i>\$.00mp</i>. Supported metric prefixes are from 1000^{^6} to 1000^{^-6}. For more information see this page.</p> <p><i>qq</i> - returns the number of the quarter when the value being formatted represents a date. To return the year and quarter together like "2010 Q1", set the format to <i>yyyy Qqq</i>.</p> |
| Show Second Data Values | Specifies if the value of the second data point should be shown on the Pie chart. Depending on the value of the Label Location attribute, they may be shown alone or with the label values. The default value is <i>False</i> . |
| Show Second Data Values | Specifies formatting characteristics for the second data values. For dates, the non-specific formats, such as <i>General Date</i> , <i>Short Time</i> , etc., are converted according to the browser's international settings. For very |

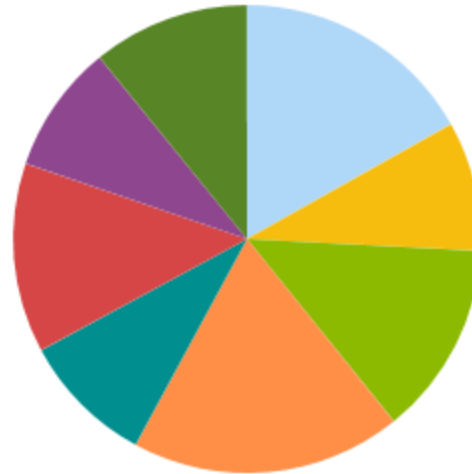
| Attribute | Description |
|--------------|---|
| Format | <p>large reports, the non-specific formats perform better. Special formats include:</p> <p>< and > - change strings to lower and upper case.</p> <p><i>Expanded Spaces</i> - preserves space characters that would otherwise be collapsed by the web browser.</p> <p><i>mp</i> - formats numbers with the "metric prefix". For example, to format 1,234,567 as "\$1.23M", enter <i>\$.00mp</i>. Supported metric prefixes are from 1000^6 to 1000^{-6}. For more information see this page.</p> <p><i>qq</i> - returns the number of the quarter when the value being formatted represents a date. To return the year and quarter together like "2010 Q1", set the format to <i>yyyy Qqq</i>.</p> |
| Transparency | <p>Specifies the transparency of the pie slice color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other.</p> |

Series.Pie - Adding Data Labels

A "data label" is text that can be shown adjacent to each pie slice that shows its X-axis data value. This is controlled using the **Label Location** attribute.



Label Location = *SideLayout*

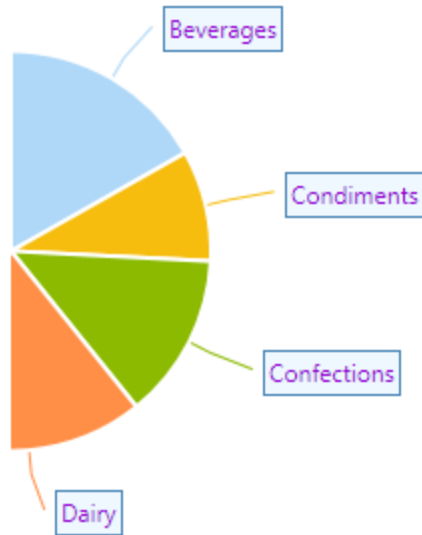


Label Location = *Legend*

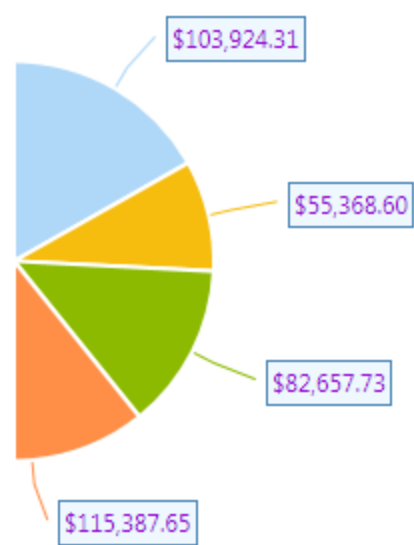
As shown above, labels can be placed beside the slices, or in a legend, or not shown at all.

When side labels are shown, the **Side Label Style** element can be used to style them and their connector lines.

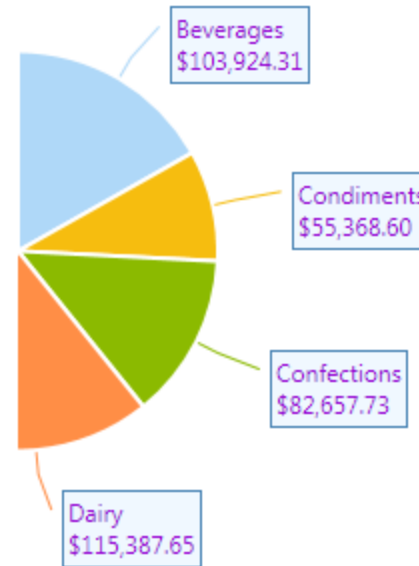
When the legend option is selected, "legend filtering" is active: clicking a slice's entry in the legend will toggle its appearance in the chart. For more information about legends, see "Legend" on page 70.



Label Location = *SideLayout*

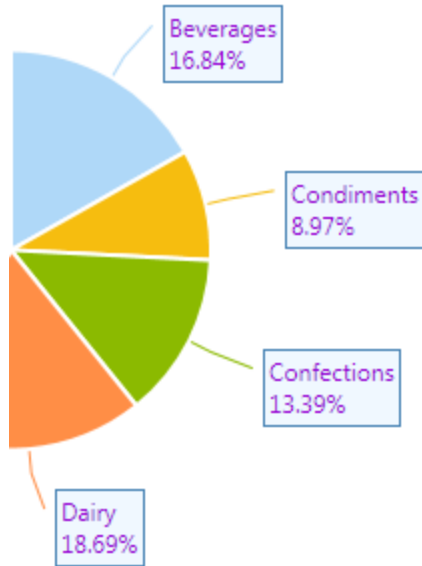


Label Location = *NoLabels*
Show Data Values = *True*



Label Location = *SideLayout*
Show Data Values = *True*

If you want to show the data *values* instead of, or with, the data *labels* near each pie slice, as shown above, use the Series.Pie element's **Label Location**, **Show Data Values**, and **Show Data Values Format** attributes.

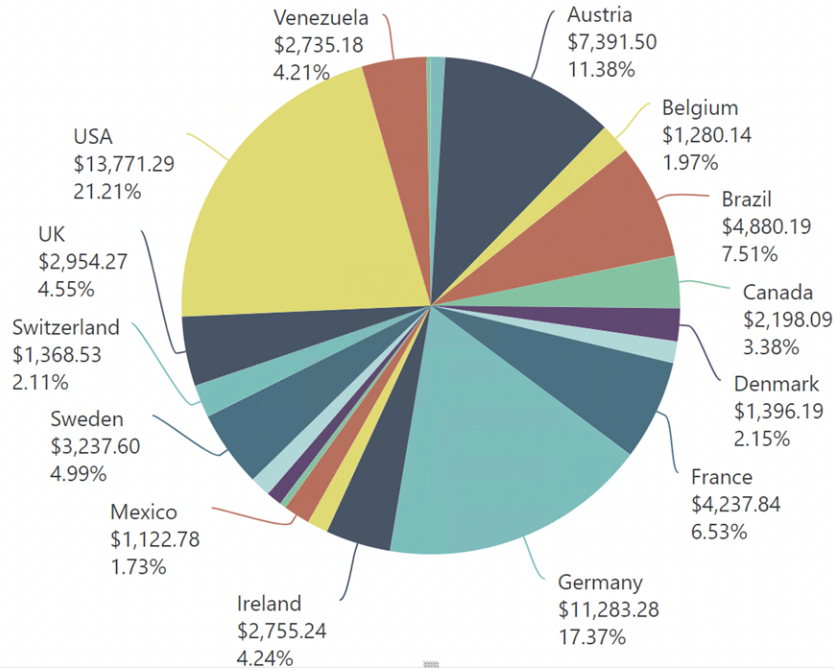


Element - Series.Pie

| *Required Attributes | |
|-------------------------|-----------------|
| Y-axis Data Column | pctProductSales |
| *Optional Attributes | |
| Angle End | |
| Angle Start | |
| ... | ... |
| Label Location | SideLayout |
| Min Radius | |
| Radius | |
| Show Data Values | True |
| Show Data Values Format | Percent |

In the example shown above, the data values are displayed as *percentages*. This is done by adding a **Percent of Total Column** element to your datalayer and using it as the chart's Y-axis Data Column. The Show Data Values and Show Data Values Format attributes are then set as shown. You also have the option to show additional data values by setting the attribute **Show Second Data Values** to "True". Then, select a format from the **Show Second Data Values Format** attribute's drop-down menu. As a result, your Pie chart will display two data values, simultaneously, like below:

Sum of Freight by ShipCountry

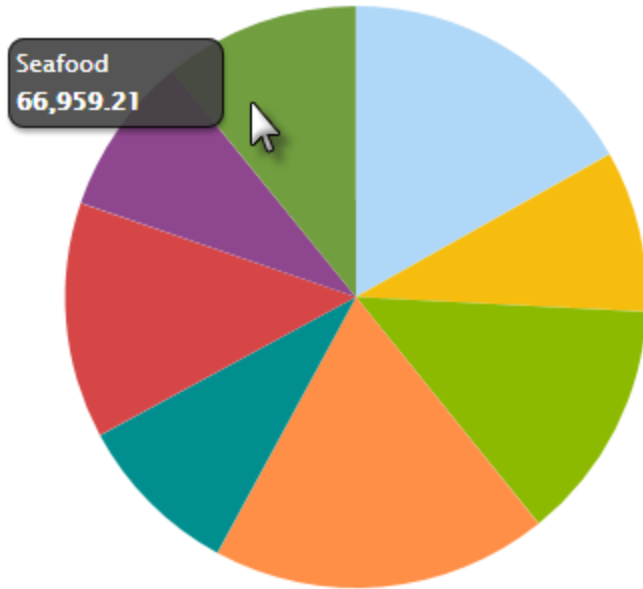


The values displayed can also be styled using the **Side Label Style** element. Its attributes allow you to control the font family, color, size, and weight, the data format, background color, border color, connecting lines and more. The side labels in the image above have been styled.

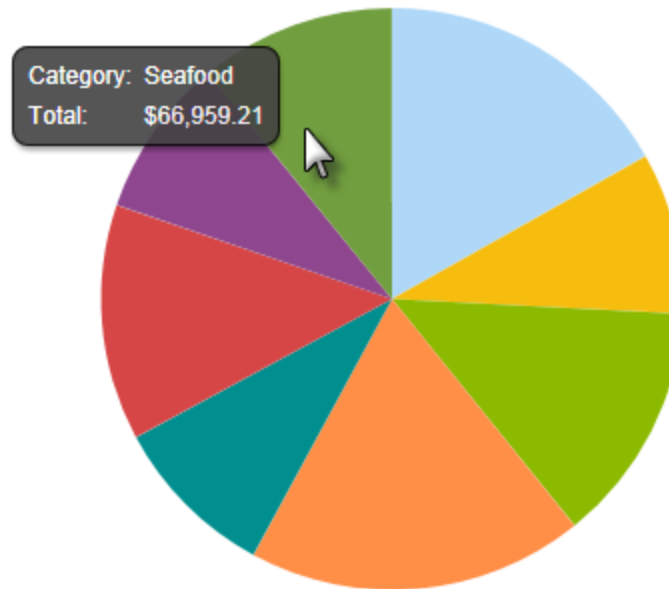
The Side Label Style element's **Maximum Label Length** attribute lets you specify the maximum number of characters that will be displayed for a label before the text is truncated and ellipsis (...) is appended.

Series.Pie - Using the Quicktips Element

By default, a "quicktip" is displayed when the mouse hovers near or over a pie slice:

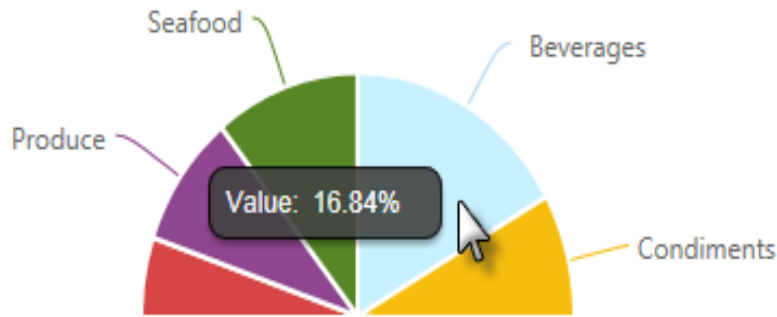


Default quicktip



With Quicktip child element

The automatically-generated quicktip displays information for the X- and Y-axis, as shown above, left. However, you may want to display other information or format it differently, perhaps as shown above, right. This "custom" Quicktip is created by adding a **Quicktip** child element beneath Series.Pie and setting its attributes and, optionally, using its **Quicktip Row** child element. Use @Chart tokens to include chart data in the Quicktip.



Element - QuicktipRow

| Optional Attributes | |
|---------------------|-------------------------|
| Caption | Value: |
| Format | Percent |
| ID | |
| Value | @Chart.pctProductSales~ |

The example above shows another custom Quicktip in use and this time the chart data values are displayed in the Quicktip as *percentages*. This is done by adding a **Percent of Total Column** element to your datalayer and using its @Chart token in a **Quicktip Row** element's **Value** attribute. The Quicktip Row element's **Format** attribute is then set to *Percent*, as shown.

You can also configure the tooltip to display values for any available column in the datalayer. To do so, enter a token representing the data column in the Value attribute of the Quicktip Row element.

💡 To use this feature with DataLayer.ActiveSQL, please make sure the keep Grouped Rows attribute of the SqlGroup element is set to *False*.

Intrinsic functions are supported in the Quicktip attributes.

The Quicktip Row element has been made context-sensitive with the addition of a **Condition** attribute.

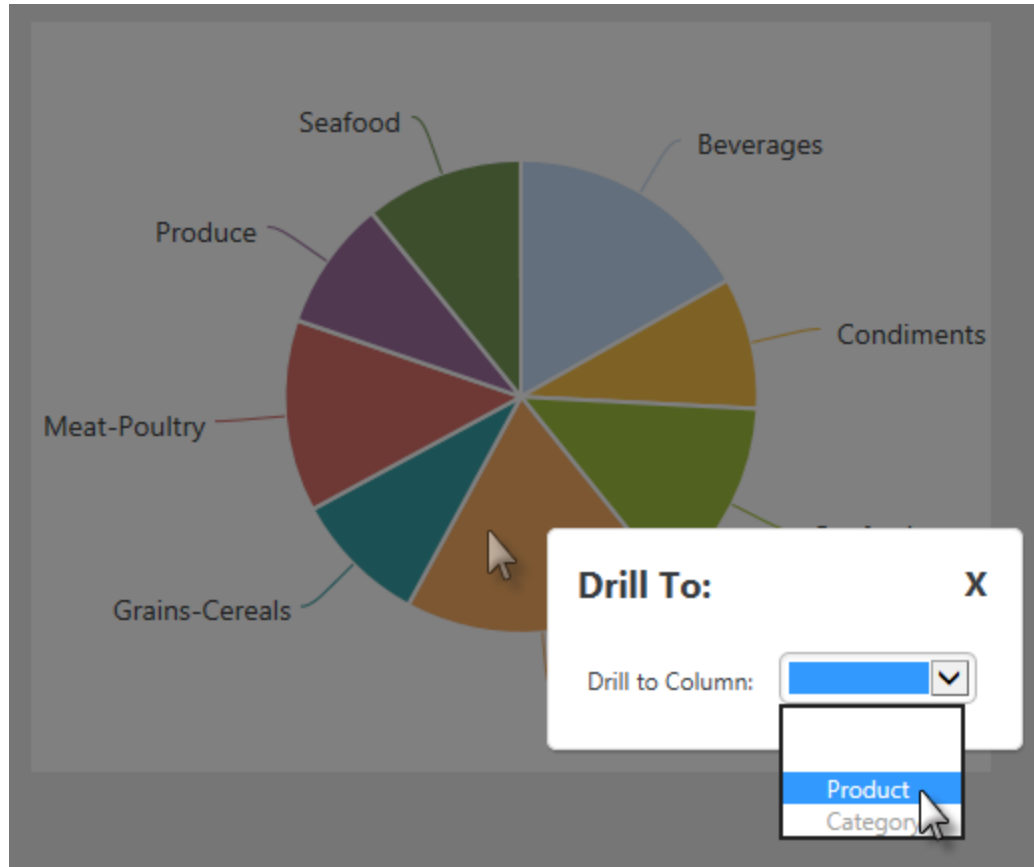
Series.Pie - Using the Chart Drill to Element

The **Chart Drill To** element, a child of the Series element, enhances charts by allowing users to examine the data "behind" the chart.



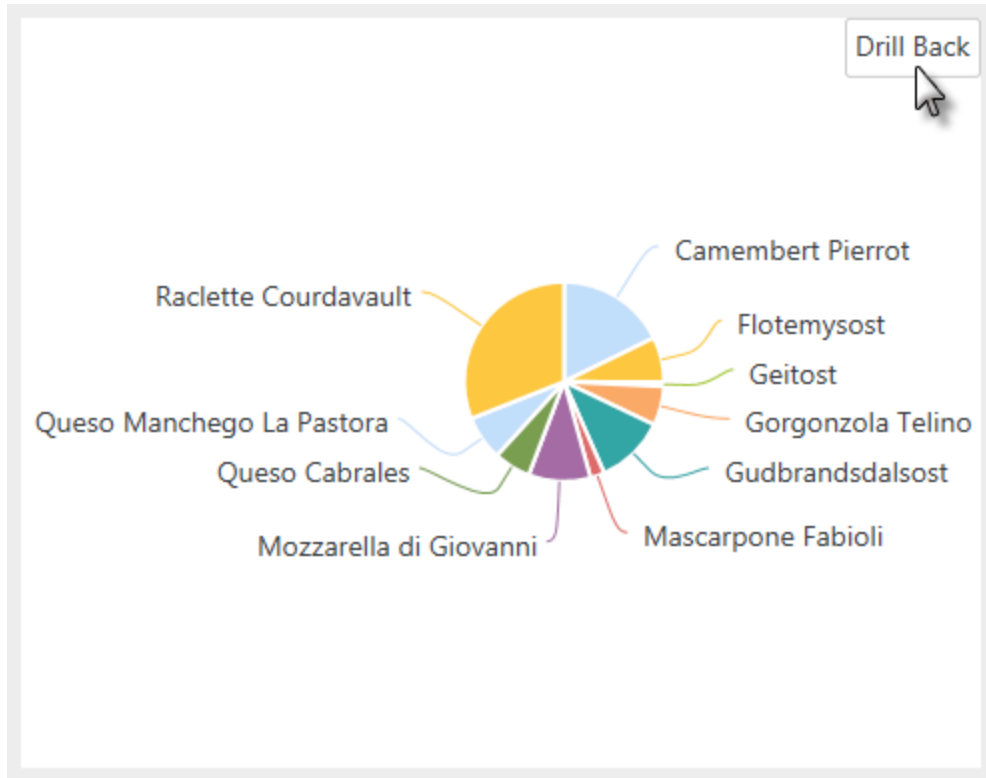
The parent Chart Canvas element must have an ID attribute value in order to use the Chart Drill To element with it.


When the Drill To element is enabled, selecting a pie wedge displays a list of columns, as shown below:



Selecting a column re-draws the chart so that only the data representing the clicked wedge is shown.

Select **Drill Back** to return the chart to it's previous state:



 Chart Canvas Charts that use the Chart Drill To element can only have a single Series. The Drill Back button only appears when the mouse cursor is hovered over the upper right-hand corner of the chart.

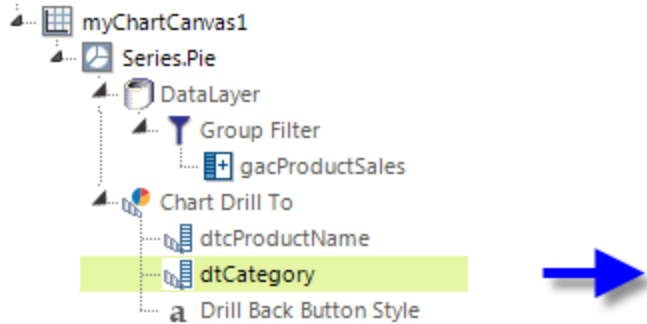
Or, continue drilling further down into the data (assuming the elements are configured for it) by selecting a wedge again.

If your chart has been configured for it, as you drill further into the data, a breadcrumb trail becomes available. The breadcrumb trail can be used to access previously drilled levels, or clear the Drill To data altogether. To enable this feature, set the **Show DrillTo Breadcrumb** attribute to *True*.

As shown below, the Chart Drill To element drills into *grouped* data, so the datalayer used beneath the Chart Canvas Chart or Series element *must* be grouped using a **Group Filter** or **Sql Group** element. The example datalayer has been grouped on the *CategoryName* column and the Keep Grouped Rows attribute was set to *False*. The Series itself has been configured so that its Label Data Column = *CategoryName* and its Y-axis Data Column = *gacProductSales* (the Group Aggregate Column that sums the ProductSales values).


| Element - GroupFilter | |
|-----------------------|--------------|
| *Required Attributes | |
| Group Column | CategoryName |
| Optional Attributes | |
| Data Type | |
| Hierarchical | |
| ID | |
| Include Condition | |
| Keep Grouped Rows | False |
| Sort Sequence | |

Next, a **Chart Drill To** element was added beneath the Series element, shown below. Required child **Drill To Column** elements have also been added; they define the columns the user can select to drill into. They should be added and configured for columns that can be reasonably grouped, such as text-type columns with a limited number of unique values and date-type columns.



Element - DrillToColumn

| *Required Attributes | |
|----------------------|--------------|
| Column Header | Category |
| Data Column | CategoryName |
| Data Type | Text |
| ID | dtCategory |
| *Optional Attributes | |
| Format | |
| Security Right ID | |

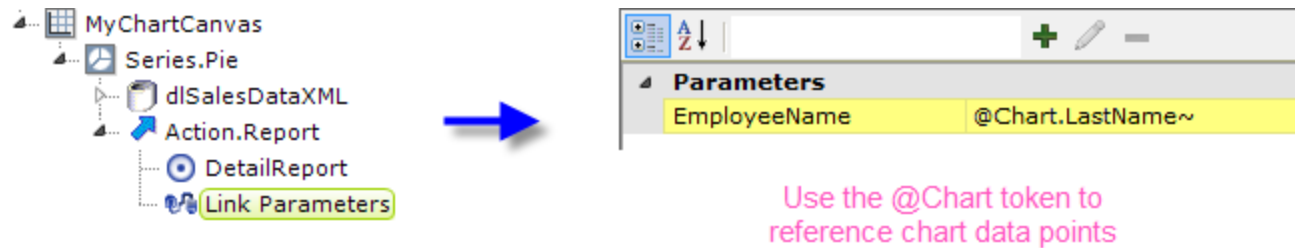
 You *must* also configure a Drill To Column element for the column that's specified as the chart's Label Data Column, in this example the *CategoryName* column.

The **Column Header** attribute specifies the text that will appear in the Drill to Column drop-down list options. The vertical order of the drop-down list options will match the vertical order of the Drill To Column elements in the definition. Year, Quarter, and Month options will be added automatically for Drill To Columns with Data Type = *Date*. You do not need to add any filter or additional grouping elements to achieve the results shown in the example above.

The Drill Back button displayed on the chart after drilling has occurred can be styled using the **Drill Back Button Style** element.

Series.Pie - Using Action Elements

Action elements initiate processing of a report or process task definition, redirection to a link, or other processing when a pie slice is clicked.



In the example above, an **Action.Report** element has been added as a child of the Series, along with its **Target.Report** and **Link Parameters** child elements. To reference chart data in parameters, use the @Chart token, as shown above.

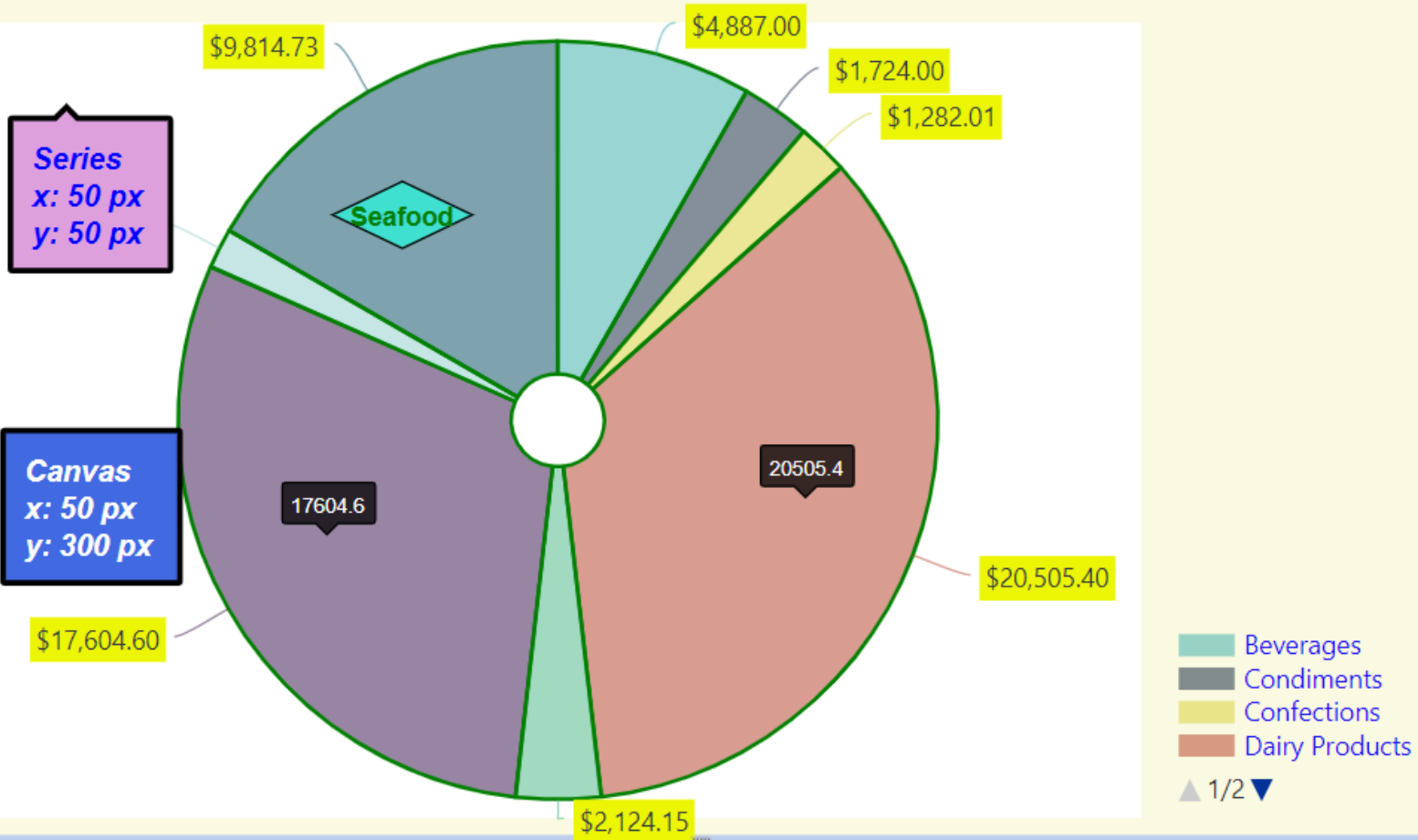
A variety of Action elements are available for use with Series, including Action.Link, Action.Process, and Action.Refresh Element. Additional Action elements will be added in future releases.

v23.1

Series.Pie - Using the Series Annotation Element

Annotations allow you to annotate a chart freely using custom labels and shapes. When the **Series Annotation** element is used as a child of Series.Pie, you can place these annotations at various points of interest:

Pie



Exports:-

ExportToPdf

ExportToExcel

ExportToWord

The Series Annotation element offers two child elements: **AnnotationLabel.Point** and **AnnotationLabel.Mock**.

The AnnotationLabel.Point child element is used to create an annotation label and position on the chart by linking it to an existing point. With this element, you cannot specify the value, it always refers to the value of the point. Additionally, you can use the Caption and Condition attributes to refer to different datalayers.

On the other hand, the AnnotationLabel.Mock is used to create an annotation label and position it on the chart by linking it to a created mock point. With this element, you can specify the coordinate, relative position, and axis.

Both the AnnotationLabel.Point and AnnotationLabel.Mock have attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text.

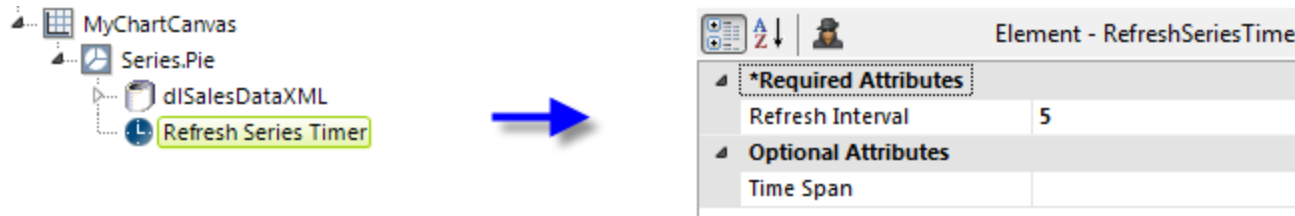
Series.Pie - Using Input Selection

This series can also be used with the **Input Selection** family of elements, which turn the chart into an input control. This allows users, at runtime, to select points or values on the chart with their mouse and your report can then take action using the selected data. This is very useful, for example, for drilling into the data.

For more information about this functionality, see "Input Selection" on page 97.

Series.Pie - Using the Refresh Series Timer

The **Refresh Series Timer**, added as a Series child element, updates charts automatically, based on a time interval. When the interval is reached, a request is sent back to the web server for updated data. Series are updated with a smooth animation.



The **Refresh Interval** attribute specifies the number of seconds to wait before refreshing the series data. A value is required and must be an integer greater than 0.

Refresh Mode

Series may be refreshed in two ways: either all of the data is refreshed with each request, or the series data automatically slides to the left one data interval per refresh. The refresh mode is selected in the Chart X Axis element's **Axis Type** attribute.

When the Axis Type is *not* set to *DateTimeLinear*, all of the data will be refreshed.

When the Axis Type value is *DateTimeLinear*, newer values are added to the right side of the chart, previous values slide left, and older values disappear as they reach the left edge of the chart. In this case, the Refresh Series Timer element's **Time Span** attribute is used to specify the age of the data included. This value must be in *hh:mm:ss* format (for example "00:02:00" for two minutes) and must be larger than the Refresh Interval attribute value.

When a Time Span attribute is set, the series' datalayer can make use of a special timestamp token, `@Chart.rdTimeSpanStart~`, in

a query to retrieve just the rows necessary to update the chart. On the initial request, this token returns the current time minus the Time Span value. For subsequent refreshes, when *Axis Type = DateTimeLinear* and the X-axis will be sliding, the token returns the last time the chart was updated, so only the newest rows are retrieved. Here's an MS SQL Server query example:

```
SELECT * FROM MyTable WHERE UpdateTime BETWEEN '@Chart.rdTimeSpanStart~' AND '@Function.DateTime~'
```

For best performance, avoid setting a very short refresh interval if a very large number of users will be displaying the report.

Series.Pyramid

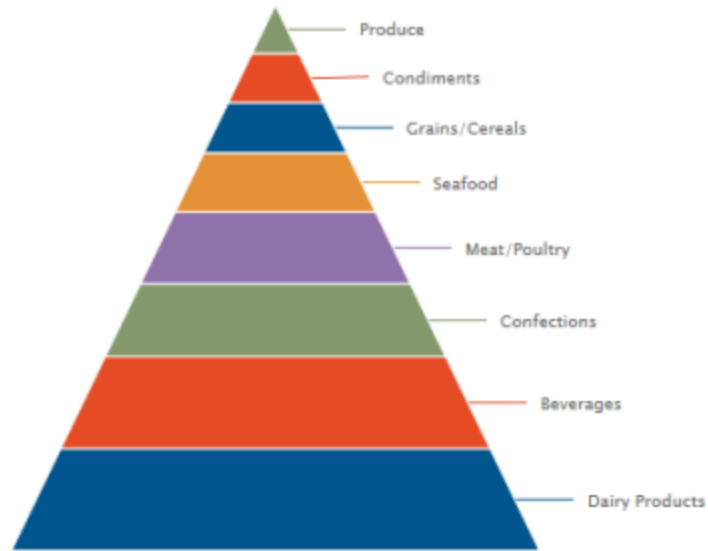
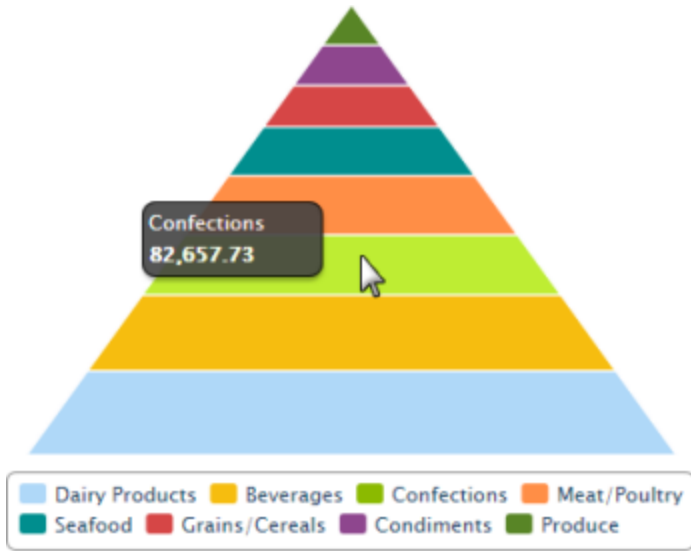
The Chart Canvas element's Series child elements cause a data visualization (the chart) to be rendered in the canvas.

The following topics discuss the Series.Pyramid child element:

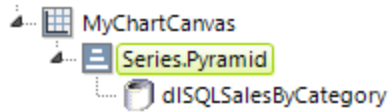
- [Using Multiple Series](#)
- [Series.Pyramid Attributes](#)
- [Adding Data Labels](#)
- [Using the Quicktips Element](#)
- [Using Action Elements](#)
- [Using the Series Annotation Element](#) v23.1
- [Using Input Selection](#)
- [Using the Refresh Series Timer](#)

About Series.Pyramid

The **Series.Pyramid** element generates a Pyramid chart, which displays values as progressively increasing proportions. The size of each pyramid segment is determined as a percentage of the total of all values.




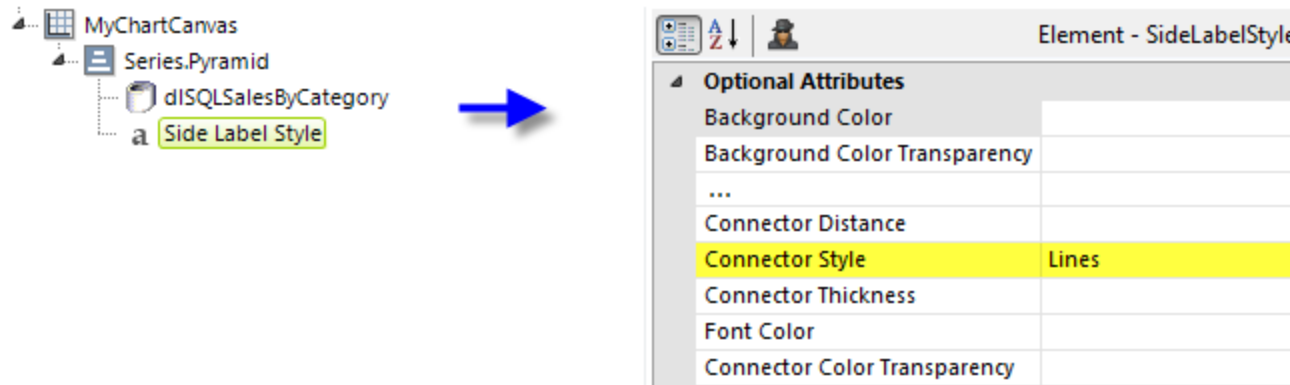
The example above shows Pyramid charts, with a legend and with side labels.



| Element - Series.Pyramid | |
|-----------------------------|----------|
| *Required Attributes | |
| Y-axis Data Column | Sales |
| Optional Attributes | |
| Colors | |
| ... | |
| Label Data Column X-axis | Category |
| Label Location | |
| Show Data Values | |
| Show Data Values Format | |
| Transparency | |

As shown above, the chart is created by adding Series.Pyramid to the canvas, along with a datalayer. Very few attributes need to be set for the Series element in order to produce a basic chart.

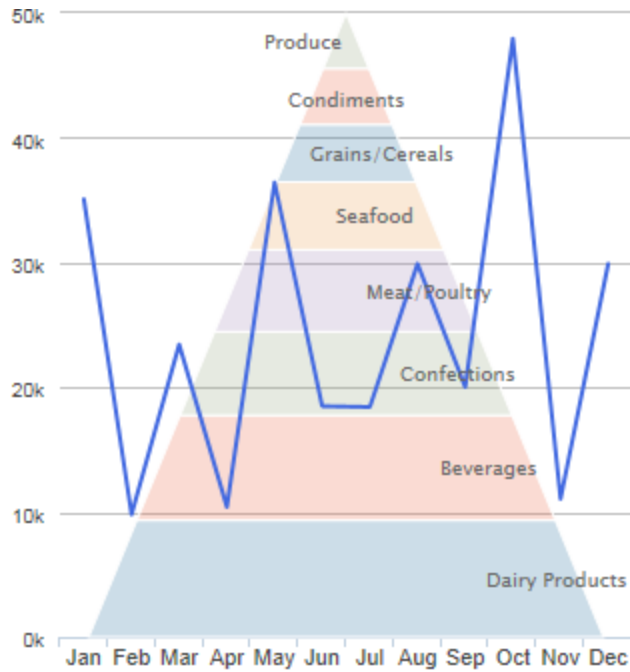
 A datalayer element can be used either beneath Series.Pyramid, as shown above, or beneath Chart Canvas. If used as a child of Chart Canvas, its data is available to *all* child Series elements. This can improve performance if you have several series, all using the *same* data, beneath the same Chart Canvas element.



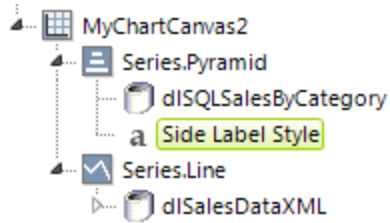
When the **Label Location** attribute has been set to *SideLayout*, as in the example, the labels can be styled using the **Side Label Style** element, as shown above.

Series.Pyramid - Using Multiple Series

Pyramid charts do not generally lend themselves to use with other series types, but it is possible. You can add additional series to the chart by adding additional Series elements:



The example above shows Series.Pyramid with Series.Line. The Pyramid's transparency has been set to allow the grid and data lines to show through it. In addition, the Pyramid's connector lines have been hidden and its labels offset to overlay the pyramid segments.




Hide connector lines

Position labels over segments

| Element - SideLabelStyle | |
|-------------------------------|---------|
| Optional Attributes | |
| Background Color | |
| Background Color Transparency | |
| ... | |
| Connector Thickness | 0 |
| Font Color | |
| Font Family | |
| Font Size | 10 |
| Font Weight | Lighter |
| Format | |
| Maximum Label Length | |
| Offset X | -90 |
| Offset Y | |

The example above shows the two Series elements and their datalayers, used to produce the previous chart. You can adjust which series appears "in front" of the other in the chart by changing the order of the Series elements in the definition. The Pyramid's **Transparency** attribute has been set to allow the grid and data lines to show through it.

 When using multiple series, you may be able to reduce the number of data queries and improve performance by using local data to read all of the data once, see *Datalayer Introduction*. Then, link its datalayer to share it to the series, see *Link Datalayers*. At each Series element, link its datalayer to the shared Local Data datalayer and filter the data to meet the needs of each individual series.

v23.1 If you are using the Chart Color Axis element in a mult-series chart, by default, the series will link to the first color axis. To link the series to a special color axis, set the ID attribute of the Chart Color Axis element to the corresponding Series' Linked to Color Axis ID attribute. For more information, see "Chart Color Axis" on page 124.

Series.Pyramid - Attributes

The Series.Pyramid element has the following attributes:

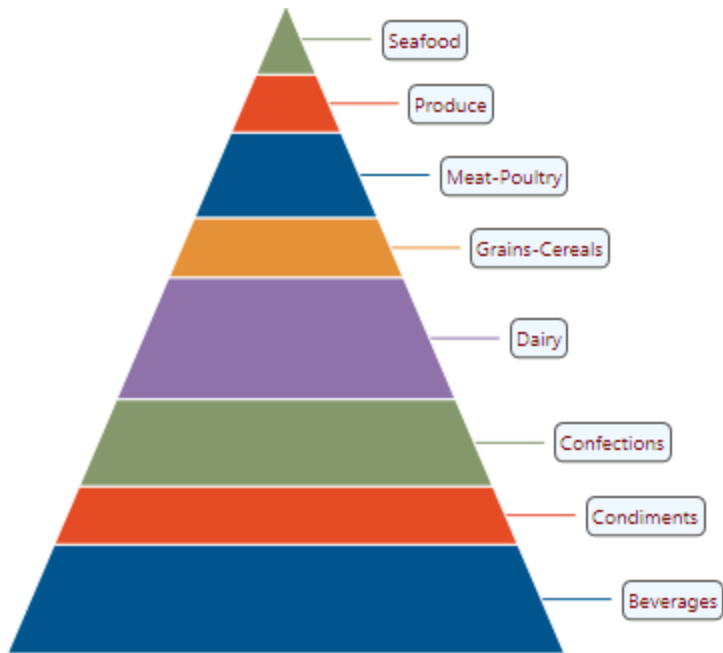
| Attribute | Description |
|--|--|
| Y-Axis Data Column | (Required) Specifies the name of a datalayer column whose values determine the height of each pyramid segment. |
| Colors | Sets the colors of the pyramid segments, which should be entered as a comma-separated list. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. When there are more segments than colors specified, the listed colors are used again from the beginning. A default set of colors is used if this is left blank and no theme has been applied, see "Chart Canvas Charts" on page 20. |
| v23.1 Color Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the x-axis. |
| Disable Mouse Tracking | Disables mouse tracking for the series, when set to <i>True</i> . This affects tooltips and click events. For large datasets, this may improve performance. The default value is <i>False</i> . |
| Hover Brightness | Specifies the change in a pyramid segment's brightness when the mouse pointer hovers over it. Values can be 0 (no change) through 15(lighter). The default value is 2. |
| Label Data Column X-axis | Specifies the name of a datalayer column whose values be represented by pyramid segments. |

| Attribute | Description |
|--|---|
| Label Location | <p>Specifies where pyramid segment labels will appear. Options include: <i>SideLayout</i> - (the default) labels will appear to the right, with lines connecting them to segments.</p> <p><i>Legend</i> - segments to be identified in a legend. The legend items can be clicked to toggle the visibility of individual segments.</p> <p><i>NoLabels</i> - labels will not be shown.</p> |
| v23.1 Linked to Color-Axis ID | <p>Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes</p> |
| Show Data Values | <p>Specifies if the value of each data point, rather than the data label, should be shown on the chart. Depending on the value of the Label Location attribute, they may be shown alone or with the label values. The default value is <i>False</i>.</p> |
| Show Data Values Format | <p>Specifies formatting characteristics for data values. For dates, the non-specific formats, such as <i>General Date</i>, <i>Short Time</i>, etc., are converted according to the browser's international settings. For very large reports, the non-specific formats perform better. Special formats include: < and > - change strings to lower and upper case.</p> <p><i>Expanded Spaces</i> - preserves space characters that would otherwise be collapsed by the web browser.</p> <p><i>mp</i> - formats numbers with the "metric prefix". For example, to format 1,234,567 as "\$1.23M", enter <i>\$.00mp</i>. Supported metric prefixes are from 1000⁶ to 1000⁻⁶. For more information see this page.</p> <p><i>qq</i> - returns the number of the quarter when the value being formatted represents a date. To return the year and quarter together like "2010 Q1", set the format to <i>yyyy Qqq</i>.</p> |

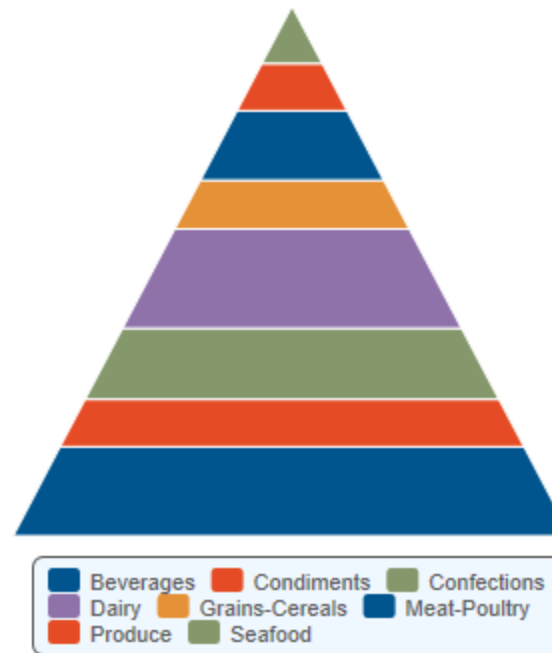
| Attribute | Description |
|--------------|--|
| Transparency | <p>Specifies the transparency of the pyramid segments. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different series to show through each other.</p> |

Series.Pyramid - Adding Data Labels

A "data label" is text that can be shown adjacent to each pyramid segment, that shows its X-axis data value. This is controlled using the **Label Location** attribute.



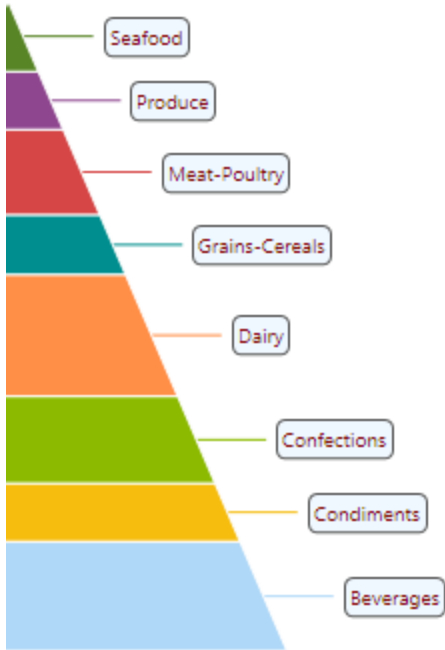
Label Location = *SideLayout*



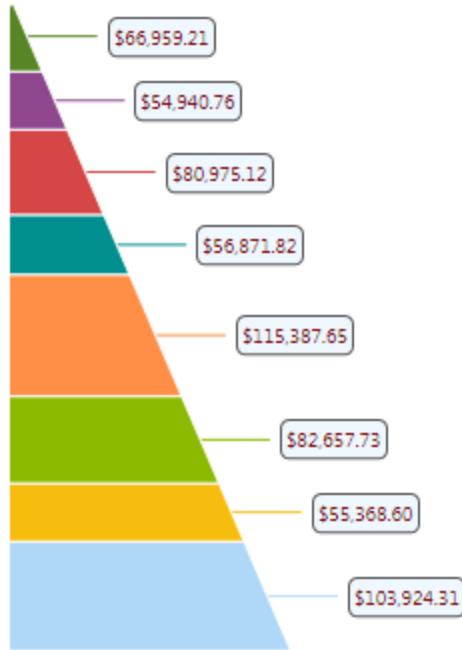
Label Location = *Legend*

As shown above, labels can be placed beside the segments, or in a legend, or not shown at all. When the legend option is selected, "legend filtering" is active: clicking a segment's entry in the legend will toggle its appearance in the chart. For more information about legends, see "Legend" on page 70.

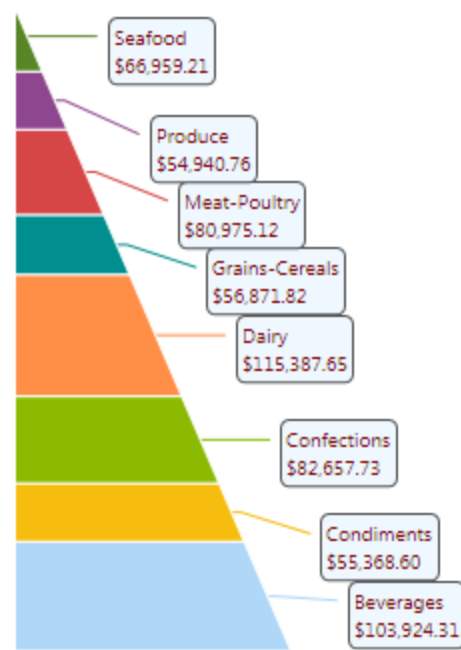
💡 When working with labels beside the pyramid, you may need to adjust the Chart Canvas element's **Spacing Left** and **Spacing Right** attributes to "fit" all of the label text inside the canvas.



Label Location = *SideLayout*



Label Location = *NoLabels*
Show Data Values = *True*



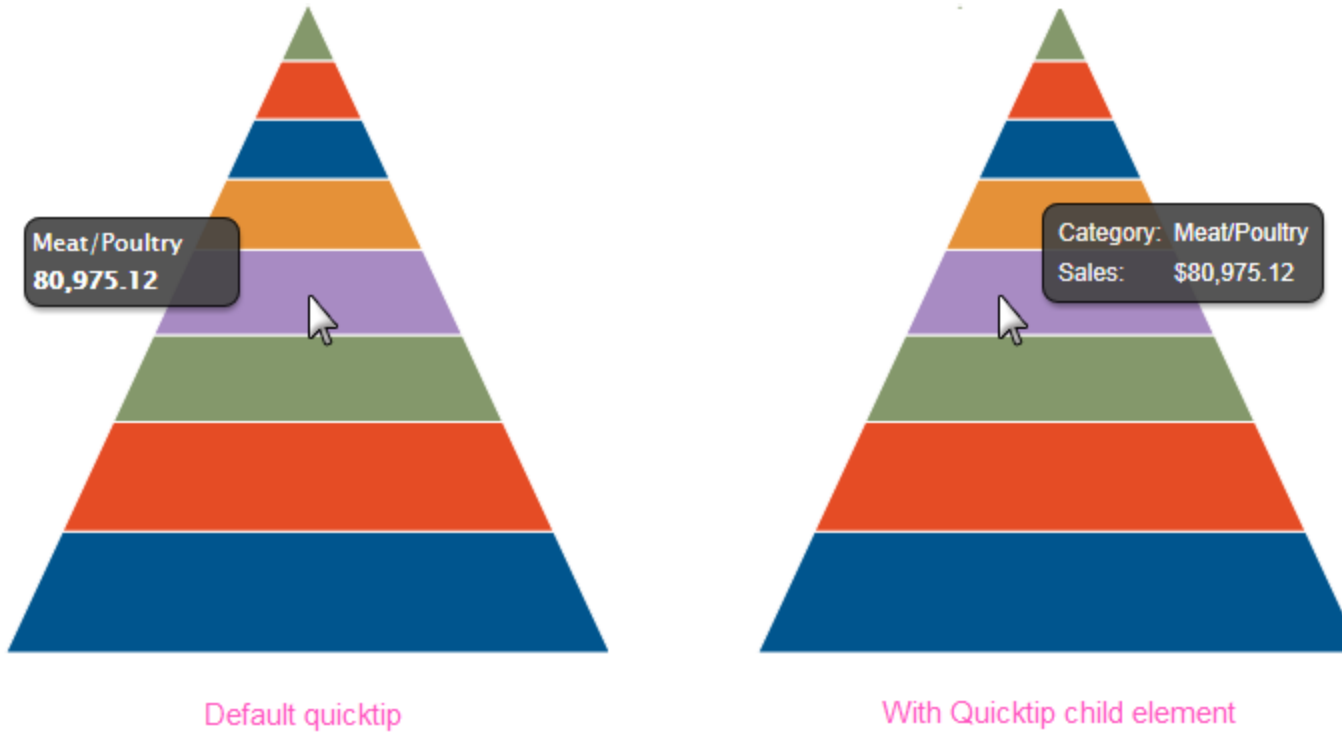
Label Location = *SideLayout*
Show Data Values = *True*

If you want to show the data *values* instead of, or with, the data *labels* near each pyramid segment, as shown above, use the Series.Pyramid element's **Label Location**, **Show Data Values**, and **Show Data Values Format** attributes. The values displayed can also be styled using the **Side Label Style** element. Its attributes allow you to control the font family, color, size, and weight, the data format, background color, border color, connecting lines and more. The side labels in the image above have been styled.

The Side Label Style element's **Maximum Label Length** attribute lets you specify the maximum number of characters that will be displayed for a label before the text is truncated and ellipsis (...) is appended.


Series.Pyramid - Using the Quicktips Element

By default, a "quicktip" is displayed when the mouse hovers over a Pyramid segment:



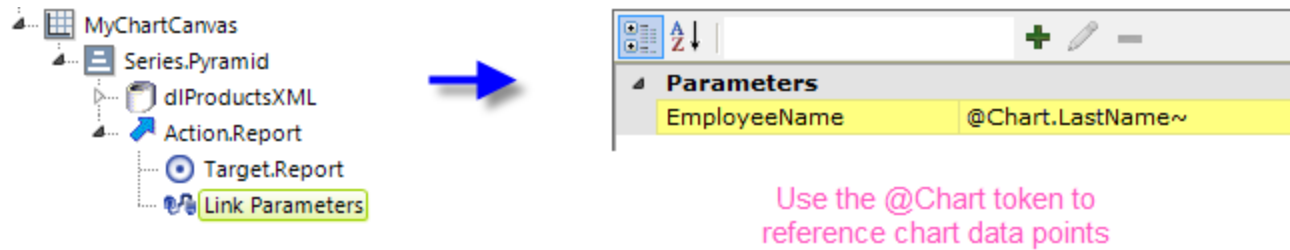
The automatically-generated quicktip displays information for the X- and Y-axis, as shown above, left. However, you may want to display other information or format it differently, perhaps as shown above, right, which can be done by adding a **Quicktip** child element beneath Series.Pyramid and setting its attributes and child elements. Use @Chart tokens to include chart data in the quicktip. Intrinsic functions are supported in the Quicktip attributes.

You can also configure the tooltip to display values for any available column in the datalayer. To do so, enter a token representing the data column in the Value attribute of the Quicktip Row element.

 To use this feature with `DataLayer.ActiveSQL`, please make sure the `keep Grouped Rows` attribute of the `SqlGroup` element is set to *False*.

Series.Pyramid - Using the Action Elements

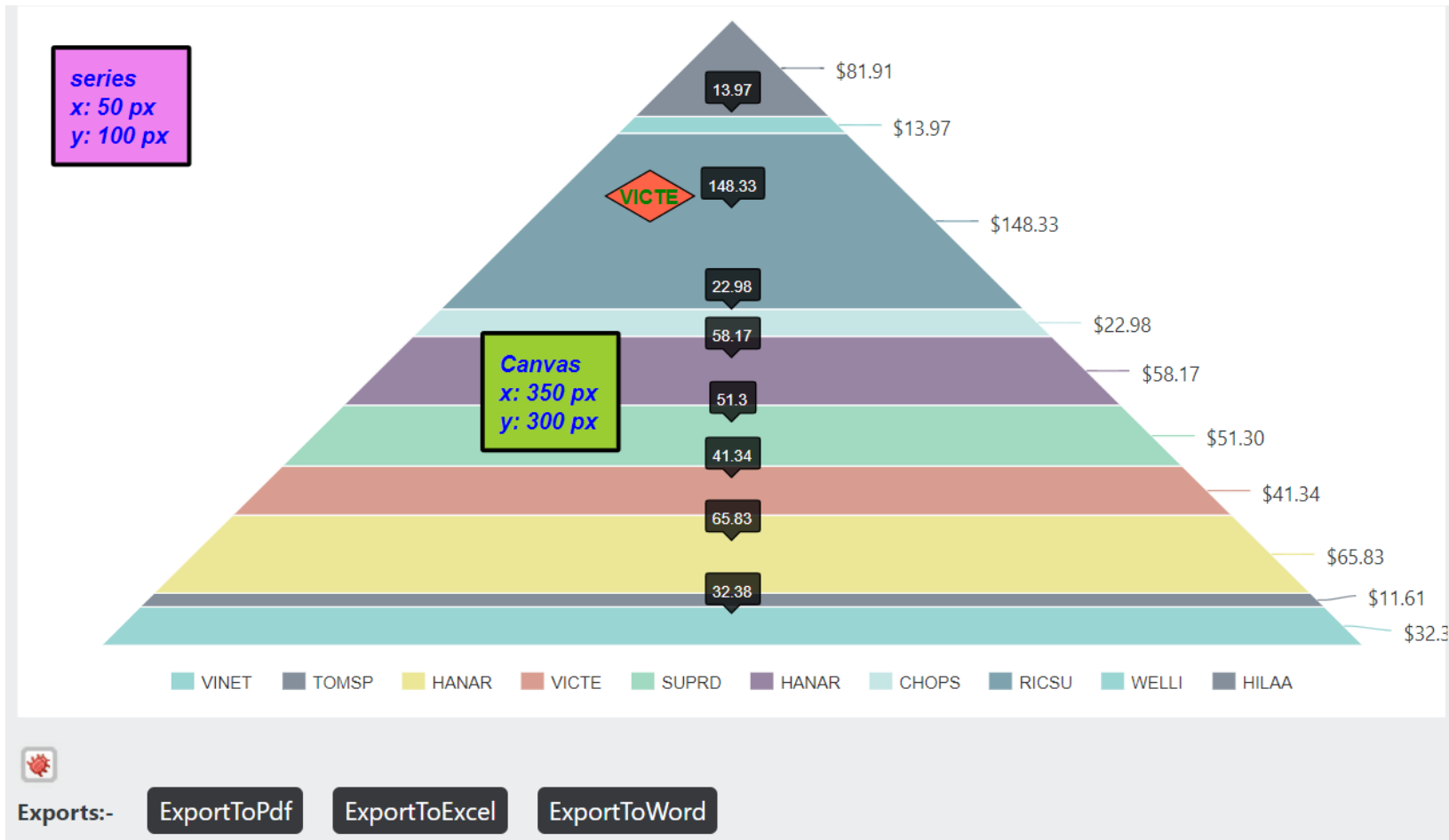
Action elements initiate processing of a report or process task definition, redirection to a link, or other processing when a segment in the pyramid is clicked.



In the example above, an **Action.Report** element has been added as a child of the Series, along with its **Target.Report** and **Link Parameters** child elements. To reference chart data in parameters, use the @Chart token, as shown above. A variety of Action elements are available for use with Series, including Action.Link, Action.Process, and Action.Refresh Element. Additional Action elements will be added in future releases.

Series.Pyramid - Using the Series Annotation Element

Annotations allow you to annotate a chart freely using custom labels and shapes. When the **Series Annotation** element is used as a child of Series.Pyramid, you can place these annotations at various points of interest:



The Series Annotation element offers two child elements: **AnnotationLabel.Point** and **AnnotationLabel.Mock**.

The `AnnotationLabel.Point` child element is used to create an annotation label and position on the chart by linking it to an existing point. With this element, you cannot specify the value, it always refers to the value of the point. Additionally, you can use the `Caption` and `Condition` attributes to refer to different datalayers.

On the other hand, the `AnnotationLabel.Mock` is used to create an annotation label and position it on the chart by linking it to a created mock point. With this element, you can specify the coordinate, relative position, and axis.

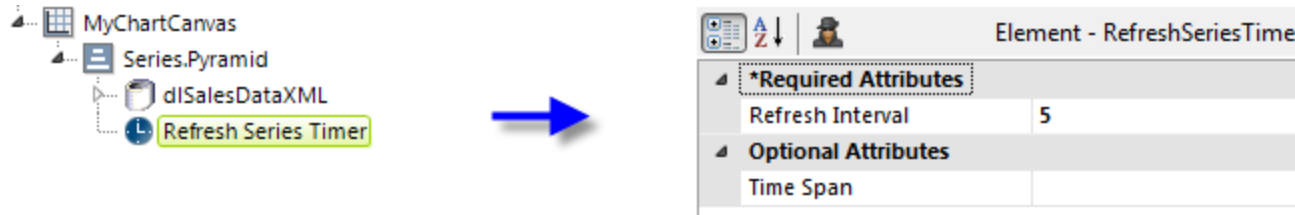
Both the `AnnotationLabel.Point` and `AnnotationLabel.Mock` have attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text.

Series.Pyramid - Using the Input Selection

This series can also be used with the **Input Selection** family of elements, which turn the chart into an input control. This allows users, at runtime, to select points or values on the chart with their mouse and your report can then take action using the selected data. This is very useful, for example, for drilling into the data. For more information about this functionality, see "Input Selection" on page 97.

Series.Pyramid - Using the Refresh Series Timer

The **Refresh Series Timer**, added as a Series child element, updates charts automatically, based on a time interval. When the interval is reached, a request is sent back to the web server for updated data. Series are updated with a smooth animation.



The **Refresh Interval** attribute specifies the number of seconds to wait before refreshing the series data. A value is required and must be an integer greater than 0.

Refresh Mode

Series may be refreshed in two ways: either all of the data is refreshed with each request, or the series data automatically slides to the left one data interval per refresh. The refresh mode is selected in the Chart X Axis element's **Axis Type** attribute. When the Axis Type is *not* set to *DateTimeLinear*, all of the data will be refreshed.

When the Axis Type value is *DateTimeLinear*, newer values are added to the right side of the chart, previous values slide left, and older values disappear as they reach the left edge of the chart. In this case, the Refresh Series Timer element's **Time Span** attribute is used to specify the age of the data included. This value must be in *hh:mm:ss* format (for example "00:02:00" for two minutes) and must be larger than the Refresh Interval attribute value.

When a Time Span attribute is set, the series' datalayer can make use of a special timestamp token, `@Chart.rdTimeSpanStart~`, in

a query to retrieve just the rows necessary to update the chart. On the initial request, this token returns the current time minus the Time Span value. For subsequent refreshes, when *Axis Type = DateTimeLinear* and the X-axis will be sliding, the token returns the last time the chart was updated, so only the newest rows are retrieved. Here's an MS SQL Server query example:

```
SELECT * FROM MyTable WHERE UpdateTime BETWEEN '@Chart.rdTimeSpanStart~' AND '@Function.DateTime~'
```

For best performance, avoid setting a very short refresh interval if a very large number of users will be displaying the report.

Series.Scatter

The Chart Canvas element's Series child elements cause a data visualization (the chart) to be rendered in the canvas.

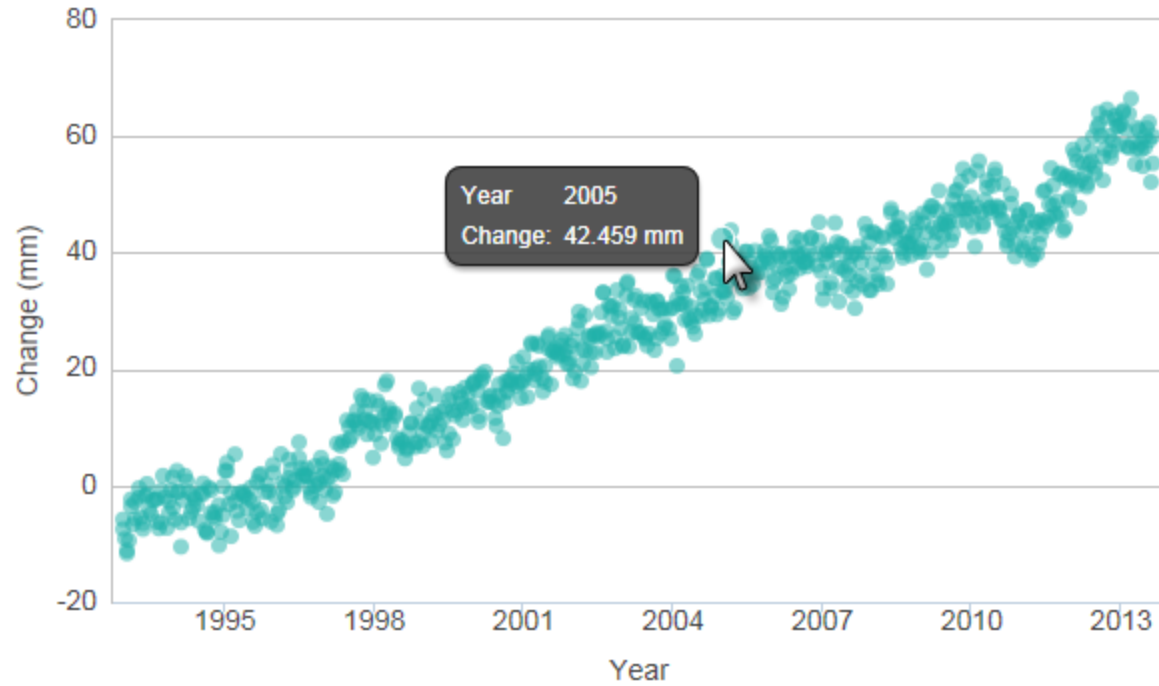
The following topics discuss the Series.Scatter child element:

- [Using Multiple Series](#)
- [Series.Scatter Attributes](#)
- [Using the Data Labels Element](#)
- [Using the Marker Points Element](#)
- [Using the Quicktips Element](#)
- [Using the Trend Line Element](#)
- [Using Action Elements](#)
- [Using the Series Annotation Element](#) v23.1
- [Using Input Selection](#)
- [Using the Refresh Series Timer](#)

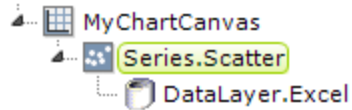
About Series.Scatter

The **Series.Scatter** element generates a Scatter chart, which displays a collection of points along Cartesian coordinates.

Sea Level Change



The example above shows a simple Scatter chart, presenting changes in sea level by year.



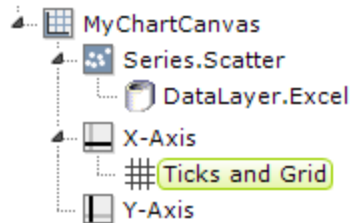
Element - Series.Scatter

| | |
|-----------------------------|---------------|
| *Required Attributes | |
| Y-axis Data Column | Change |
| Optional Attributes | |
| Color | LightSeaGreen |
| Combine With Series ID | |
| ... | |
| X-axis Data Column | Year |
| X-axis Data Column Type | |

As shown above, the chart is created by adding Series.Scatter to the canvas, along with a datalayer. Very few attributes need to be set for the Series element in order to produce a basic chart.



A datalayer element can be used either beneath Series.Scatter, as shown above, or beneath Chart Canvas. If used as a child of Chart Canvas, its data is available to *all* child Series elements. This can improve performance if you have several series, all using the *same* data, beneath the same Chart Canvas element.



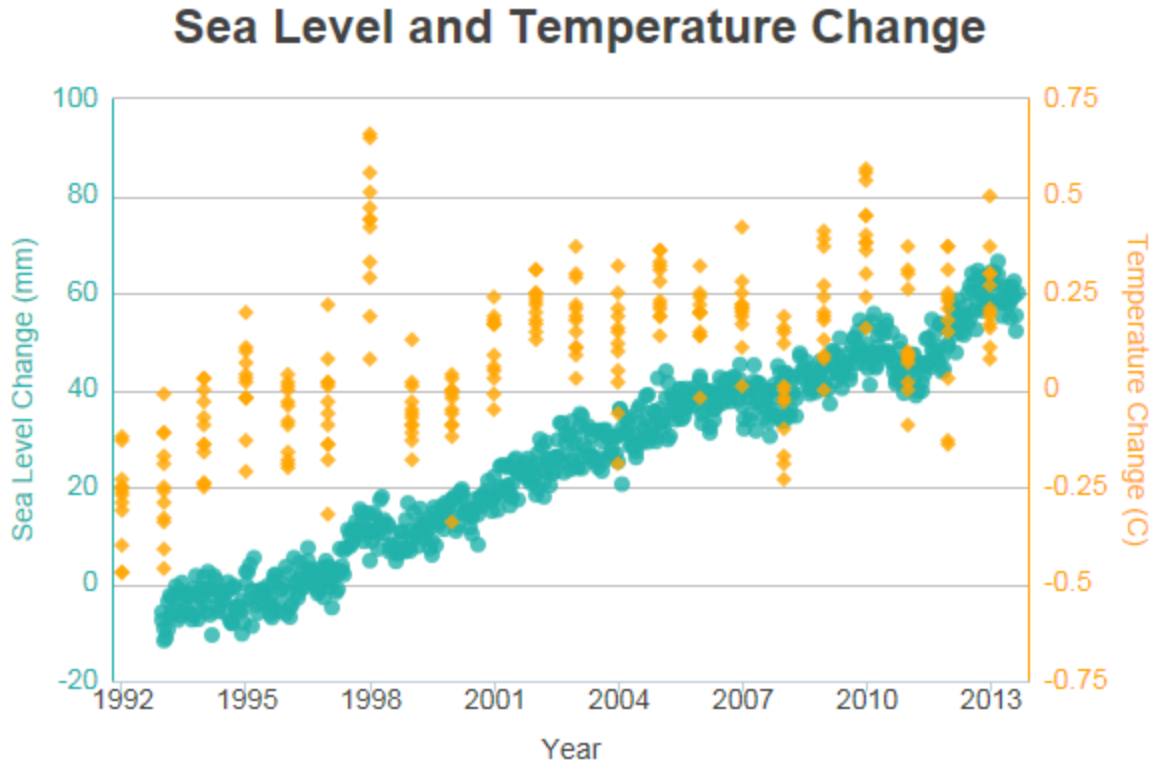
Element - ChartTicksAndGrid

| | |
|----------------------------|---|
| Optional Attributes | |
| Alternate Grid Color | |
| ... | |
| Grid Line Thickness | 0 |
| Start On Tick | |
| Tick Interval | 3 |
| ... | |
| Tickmark Placement | |

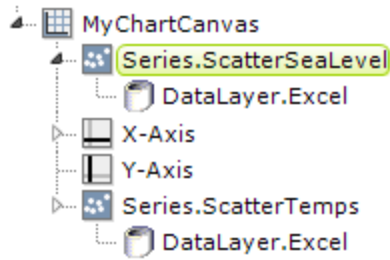
The properties of the X- and Y-axis, including captions, are set using the **X-Axis** and **Y-Axis** elements, as shown above. To limit the number of years shown on the X-axis, a **Ticks and Grid** child element is used to set the tick mark interval.

Series.Scatter - Using Multiple Series

You can add additional series to the chart by adding additional Series elements:



The example above shows two Series but, because they have different data value scales, a secondary Y-axis has been added on the right side. Many charts that compare multiple data series over time can use the same data scale for all series, but that's not the case here, so a secondary Y-axis is necessary. To see how this extra axis is implemented, see the section below.



| Element - Series.Scatter | |
|-----------------------------|------------------------|
| *Required Attributes | |
| Y-axis Data Column | Change |
| Optional Attributes | |
| Color | LightSeaGreen |
| Combine With Series ID | |
| Disable Mouse Tracking | |
| ID | Series.ScatterSeaLevel |
| ... | |
| X-axis Data Column | Year |
| X-axis Data Column Type | |

The example above shows the two Series elements, their datalayers, and **X-** and **Y-Axis** elements used to produce the previous chart. Should they overlap, you can adjust which symbols appear "in front" of the others in the chart by changing the order of the Series elements in the definition. Setting the Series elements' **Legend Label** attribute will automatically cause the legend to be displayed.

Charts with multiple series or aggregations are encouraged to use the ChartCanvas element's attribute, `Tooltip Split`. This attribute allows you to split a chart's tooltip into one label per series, rather than per segment. The default value for this attribute is *False*. Once enabled, when hovering over a data point, it displays all the values for the series. This method is recommended over shared tooltips for charts with multiple line series, and as such, takes precedence over `tooltip.shared`.

💡 When using multiple series, you may be able to reduce the number of data queries and improve performance by using local data to read all of the data once, see *Datalayer Introduction*. Then, link its datalayer to share it to the series, see *Link Datalayers*. At each Series element, link its datalayer to the shared Local Data datalayer and filter the data to meet the needs of each individual series.

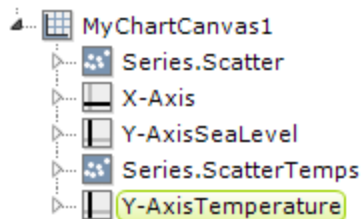
You can combine different types of Series elements, for example, Series.Scatter and Series.Line, to produce combinations of visualizations.

v23.1 If you are using the Chart Color Axis element in a multi-series chart, by default, the series will link to the first color axis.

To link the series to a special color axis, set the ID attribute of the Chart Color Axis element to the corresponding Series' Linked to Color Axis ID attribute. For more information, see "Chart Color Axis" on page 124.

Adding a Secondary Axis

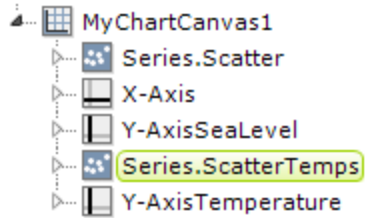
Using the example shown previously, it's easy to add a secondary axis to a chart.



| Optional Attributes | |
|-------------------------|--------------------------|
| Axis Padding Top | |
| Axis Type | |
| Caption | Temperature Change (C) |
| ... | |
| ID | Y-AxisTemperature |
| Line Color | Orange |
| Line Color Transparency | |
| Line Thickness | |
| Opposite Side | True |
| Spacing | |

Two **Y-Axis** elements are used, one for each Series, each with a unique ID

To add a second Y-axis to the chart, add a second **Y-Axis** element to the definition, as shown above. Set its **Opposite Side** attribute to *True* and give both Y-Axis elements a unique element ID. This creates the secondary Y-axis, on the right side of the chart.



Associate a Series with an axis using the **Linked to Y-Axis ID** attribute

| Element - Series.Scatter | |
|-----------------------------|--------------------------|
| *Required Attributes | |
| Y-axis Data Column | Change |
| Optional Attributes | |
| Color | Orange |
| Combine With Series ID | |
| Disable Mouse Tracking | |
| ID | Series.ScatterTemps |
| Legend Label | |
| Linked to X-Axis ID | |
| Linked to Y-Axis ID | Y-AxisTemperature |
| Transparency | |
| X-axis Data Column | Year |
| X-axis Data Column Type | |

But which Series will use which Y-axis? You associate a Series with a Y-Axis element by setting the Series' **Linked to Y-Axis ID** attribute, as shown above, to the element ID of the desired Y-Axis element. Do this for each Series element.

Series.Scatter - Attributes

The Series.Scatter element has the following attributes:

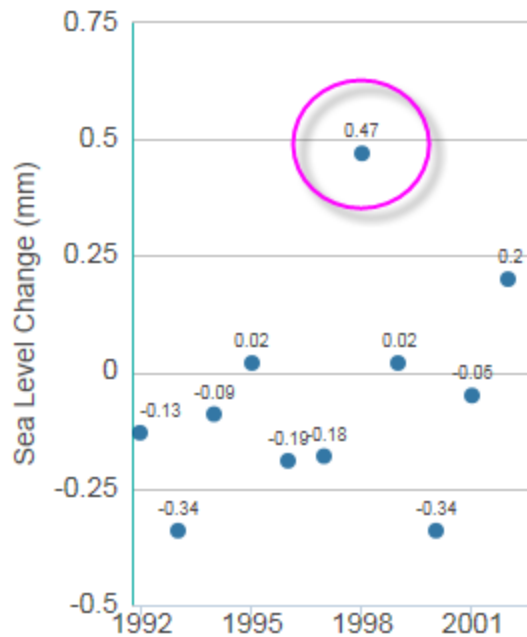
| Attribute | Description |
|---|---|
| Y-Axis Data Column | (Required) Specifies the name of a datalayer column whose values will be plotted along the Y-axis. |
| v23.1 Class Name | Specifies the Class Name referenced in the .css style sheet. The default value is <i>undefined</i> . |
| Color | Sets the chart symbol color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. @Chart tokens may be used here to provide dynamic, data-driven data point colors. For example, the token @Gradient can be used to represent the data with a gradient fill. |
| v23.1 Color Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the x-axis. |
| Combine With Series ID | Set this attribute to the element ID of another series to combine it with this series in the legend. When two series are combined, by default only the first one will appear in the legend but clicking the item in the legend toggles both series to appear and disappear. Or, the value <i>Previous</i> can be entered to combine this series with the previous series. |
| Disable Mouse Tracking | Disables mouse tracking for the series, when set to <i>True</i> . This affects tooltips and click events on graphs and points. For large datasets, this may improve performance. |

| Attribute | Description |
|--|--|
| | The default value is <i>False</i> . |
| Legend Label | Indicates text that will be shown for this series inside the chart legend. When a value is provided, automatically causes the legend to be displayed. |
| v23.1 Linked to Color-Axis ID | Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes |
| Linked to X-Axis ID | Specifies the ID of an X-Axis element that this series should be linked to when using multiple X-axes. |
| Linked to Y-Axis ID | Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes. |
| Transparency | Specifies the transparency of the symbol color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| X-Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the X-axis. |
| X-Axis Data Column Type | Specifies the data type of the datalayer column named in the X-axis Data Column attribute. Options include <i>Auto</i> (the default), <i>Text</i> , <i>Number</i> , and <i>DateTime</i> . By default, X-axis data values that are <i>DateTime</i> type will be automatically distributed evenly across the time series. If you want to disable this behavior, set this |

| Attribute | Description |
|-----------|----------------------------------|
| | attribute value to <i>Text</i> . |

Series.Scatter - Using the Data Labels Element

A "data label" is text shown next to each data point that shows its value. When the **Data Labels** element is used as a child of Series.Scatter, text representing the data values can be configured to appear on the chart, next to the symbols.

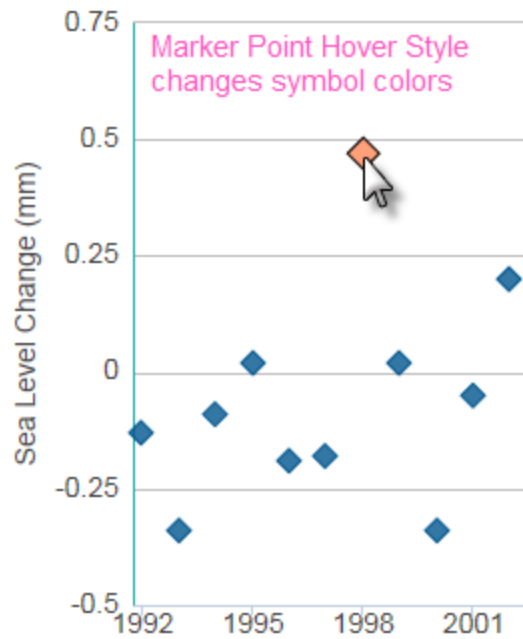


The Data Labels element has attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text. One attribute, Series Name as Caption, allows you to specify the y-axis label, x-axis label, or legend label as the data point value inside the chart.

The Data Labels element's color-related attribute values can be set using @Chart tokens. v23.1 Or, utilize the "Class Name" attribute to apply unique styling to your data labels by referencing a class name from a stylesheet in the report.

Series.Scatter - Using the Marker Points Element

A "marker point" is a symbol that appears on the chart at each data point. These behave differently in a Scatter chart than in other types of charts: the marker point *is* the chart symbol itself. When the **Marker Points** element is used as a child of Series.Scatter, several properties can be configured.

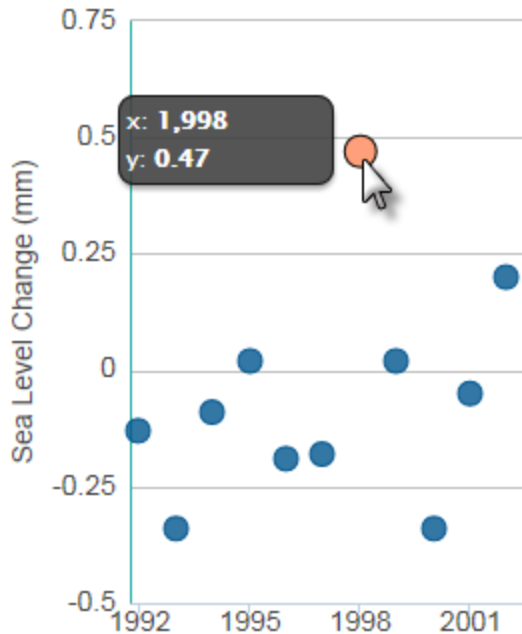


The default marker points are shown earlier, in the "Using Data Labels" topic. The marker points shown above are one of five symbol options that can be used. The Marker Points element allows you to control the symbol choice, its color and border color, and its transparency. When the cursor hovers over it, a symbol's color can change - the **Marker Points Hover Style** child element lets you configure that behavior and other properties.

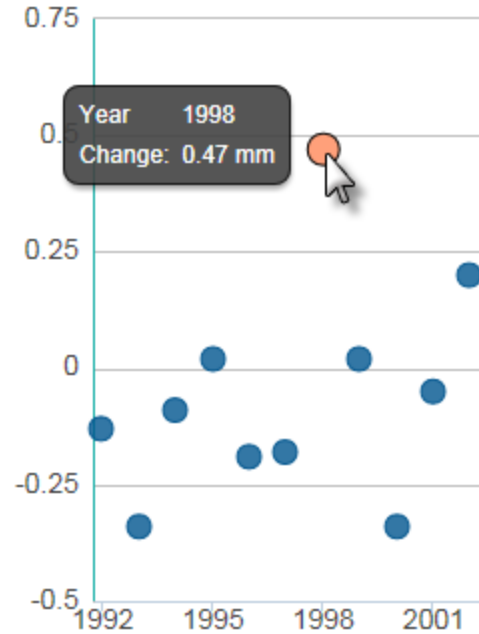
v23.1 You can also create your own marker point symbol using the Chart Canvas child element, Chart Custom Symbol. Once defined, link your custom symbol using the Marker Points element's Symbol attribute. For more information, see "Chart Custom Symbol" on page 128.

Series.Scatter - Using the Quicktips Element

By default, a "quicktip" is displayed when the mouse hovers over a symbol:



Default quicktip



With Quicktip child element

The automatically-generated quicktip displays information from the X- and Y-axes, as shown above, left. However, you may want to display other information or format it differently, perhaps as shown above, right, which can be done by adding a **Quicktip** child element beneath Series.Scatter and setting its attributes and child elements. Use @Chart tokens to include chart data in the quicktip. Intrinsic functions are supported in the Quicktip attributes.

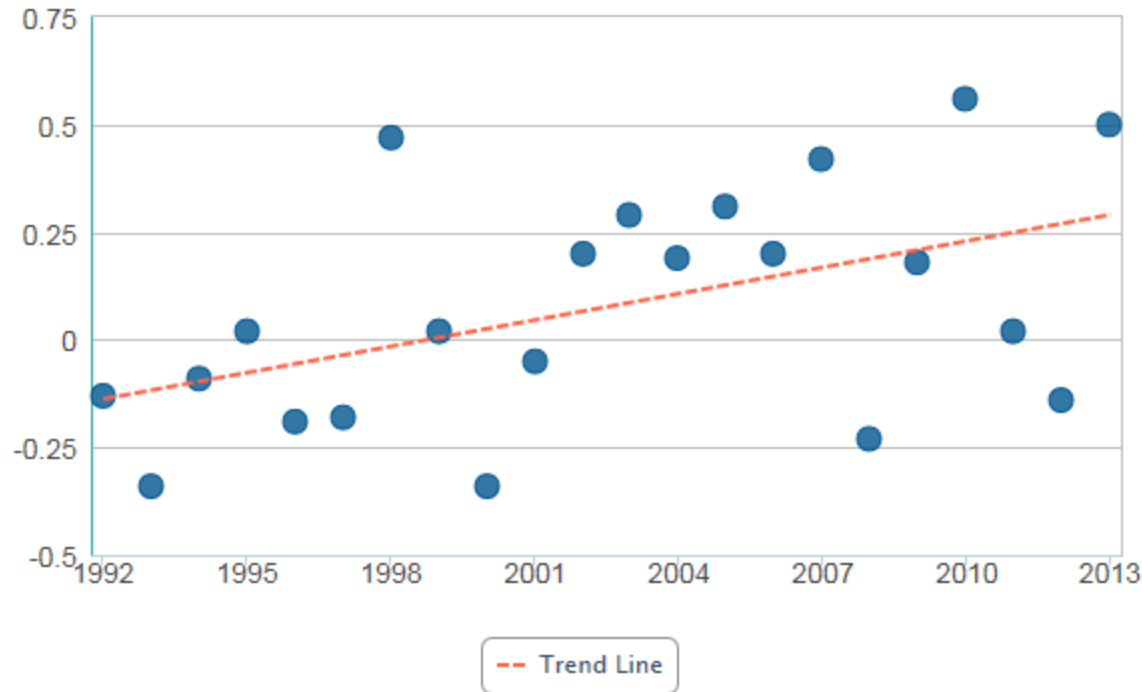
You can also configure the tooltip to display values for any available column in the datalayer. To do so, enter a token representing the data column in the Value attribute of the Quicktip Row element.



To use this feature with `DataLayer.ActiveSQL`, please make sure the `keep Grouped Rows` attribute of the `SqlGroup` element is set to *False*.

Series.Scatter - Using the Trend Line Element

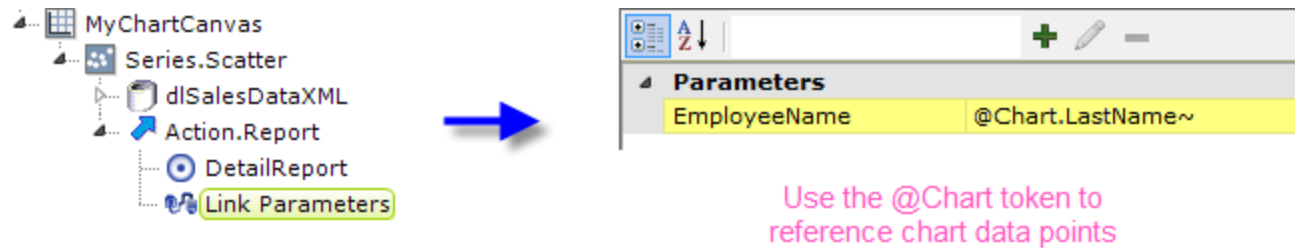
The **Trend Line** element creates a line on the chart that indicates the "trend" of the data. The line connects a number of data points generated using a regression algorithm.



The Trend Line element is a child of the Series.Scatter element and can be styled for color, line width, etc. When configured with a legend label, it will be represented by an item in the legend, as shown above.

Series.Scatter - Using Action Elements

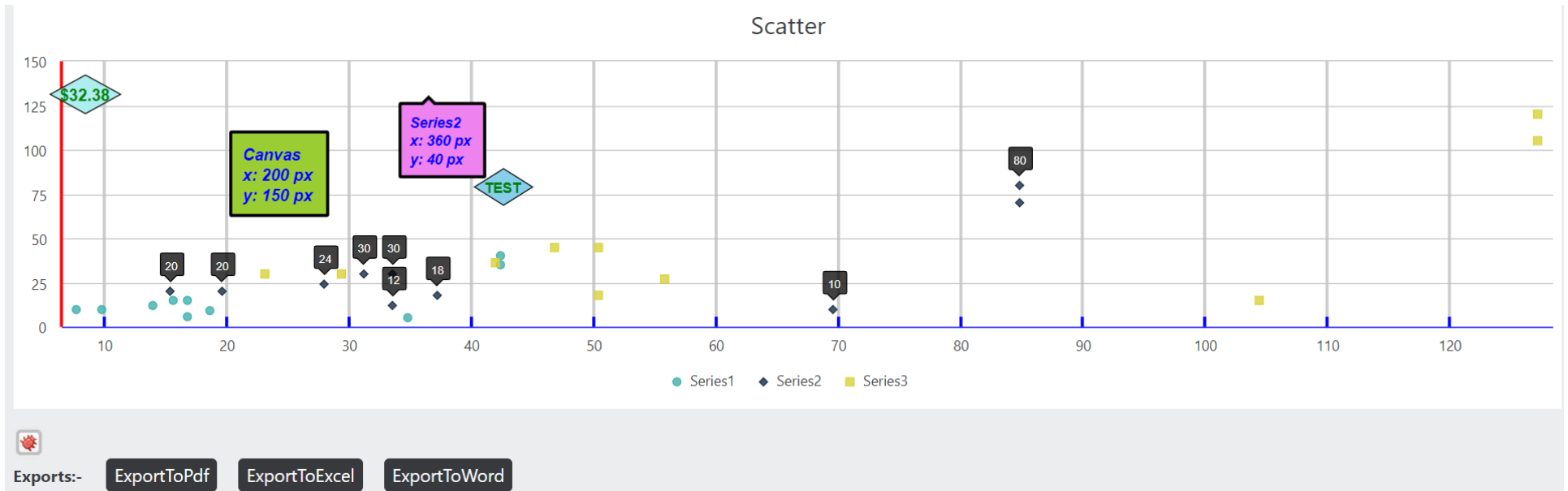
Action elements initiate processing of a report or process task definition, redirection to a link, or other processing when a data point in the series is clicked.



In the example above, an **Action.Report** element has been added as a child of the Series, along with its **Target.Report** and **Link Parameters** child elements. To reference chart data in parameters, use the @Chart token, as shown above. A variety of Action elements are available for use with Series, including Action.Link, Action.Process, and Action.Refresh Element. Additional Action elements will be added in future releases.

Series.Scatter - Using the Series Annotation Element

Annotations allow you to annotate a chart freely using custom labels and shapes. When the **Series Annotation** element is used as a child of Series.Scatter, you can place these annotations at various points of interest:



The Series Annotation element offers two child elements: **AnnotationLabel.Point** and **AnnotationLabel.Mock**.

The AnnotationLabel.Point child element is used to create an annotation label and position on the chart by linking it to an existing point. With this element, you cannot specify the value, it always refers to the value of the point. Additionally, you can use the Caption and Condition attributes to refer to different datalayers.

On the other hand, the AnnotationLabel.Mock is used to create an annotation label and position it on the chart by linking it to a created mock point. With this element, you can specify the coordinate, relative position, and axis.

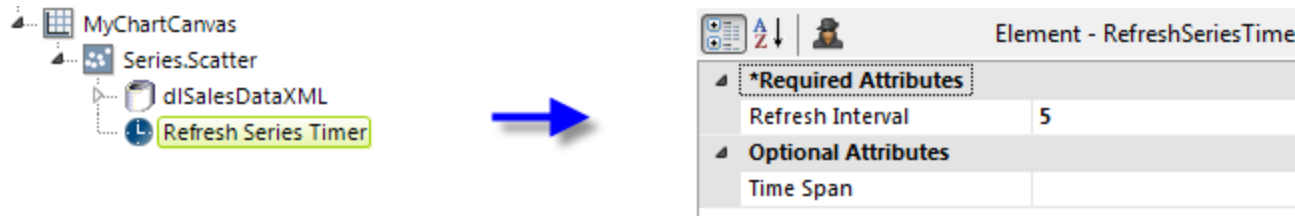
Both the AnnotationLabel.Point and AnnotationLabel.Mock have attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text.

Series.Scatter - Using Input Selection

This series can also be used with the **Input Selection** family of elements, which turn the chart into an input control. This allows users, at runtime, to select points or values on the chart with their mouse and your report can then take action using the selected data. This is very useful, for example, for drilling into the data. For more information about this functionality, see "Input Selection" on page 97.

Series.Scatter - Using the Refresh Series Timer

The **Refresh Series Timer**, added as a Series child element, updates charts automatically, based on a time interval. When the interval is reached, a request is sent back to the web server for updated data. Series are updated with a smooth animation.



The **Refresh Interval** attribute specifies the number of seconds to wait before refreshing the series data. A value is required and must be an integer greater than 0.

Refresh Mode

Series may be refreshed in two ways: either all of the data is refreshed with each request, or the series data automatically slides to the left one data interval per refresh. The refresh mode is selected in the Chart X Axis element's **Axis Type** attribute. When the Axis Type is *not* set to *DateTimeLinear*, all of the data will be refreshed.

When the Axis Type value is *DateTimeLinear*, newer values are added to the right side of the chart, previous values slide left, and older values disappear as they reach the left edge of the chart. In this case, the Refresh Series Timer element's **Time Span** attribute is used to specify the age of the data included. This value must be in *hh:mm:ss* format (for example "00:02:00" for two minutes) and must be larger than the Refresh Interval attribute value.

When a Time Span attribute is set, the series' datalayer can make use of a special timestamp token, `@Chart.rdTimeSpanStart~`, in a query to retrieve just the rows necessary to update the chart. On the initial request, this token returns the current time minus the Time Span value. For subsequent refreshes, when *Axis Type = DateTimeLinear* and the X-axis will be sliding, the token returns the last time the chart was updated, so only the newest rows are retrieved. Here's an MS SQL Server query example:

```
SELECT * FROM MyTable WHERE UpdateTime BETWEEN '@Chart.rdTimeSpanStart~' AND '@Function.DateTime~'
```

For best performance, avoid setting a very short refresh interval if a very large number of users will be displaying the report.

Series.Spline

The Chart Canvas element's Series child elements cause a data visualization (the chart) to be rendered in the canvas.

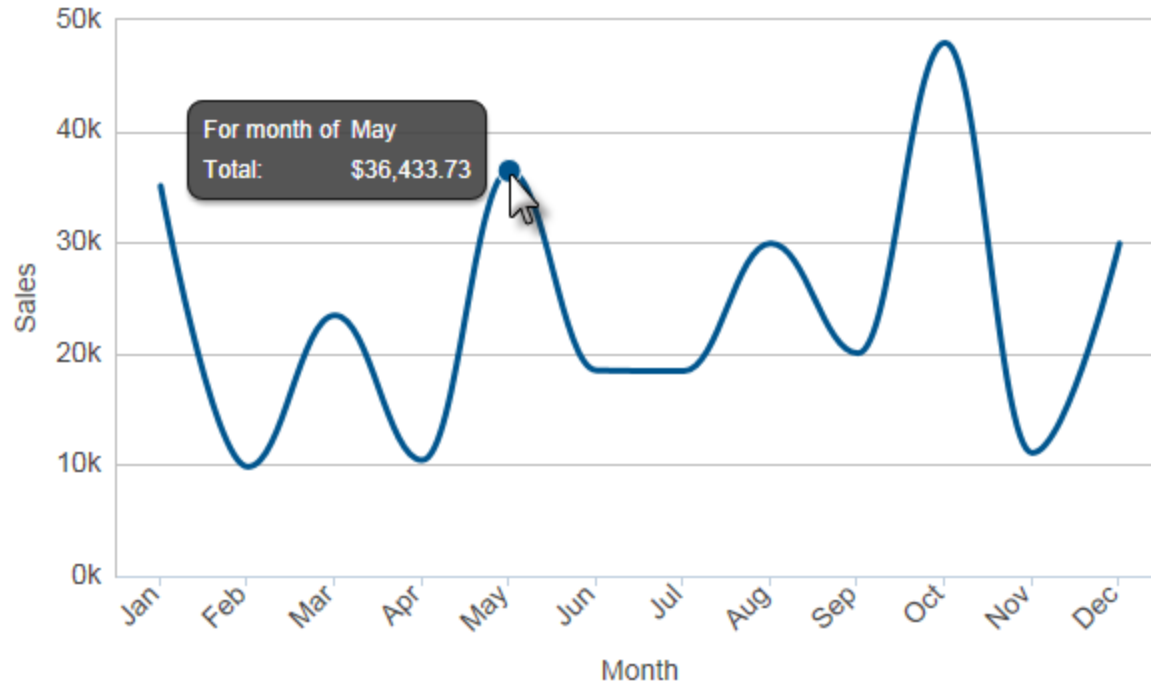
The following topics discuss the Series.Spline child element:

- [Using Multiple Series](#)
- [Series.Spline Attributes](#)
- [Using the Data Labels Element](#)
- [Using the Marker Points Element](#)
- [Using the Quicktips Element](#)
- [Using the Trend Line Element](#)
- [Using Action Elements](#)
- [Using the Series Annotation Element](#) v23.1
- [Using Input Selection](#)
- [Using the Refresh Series Timer](#)

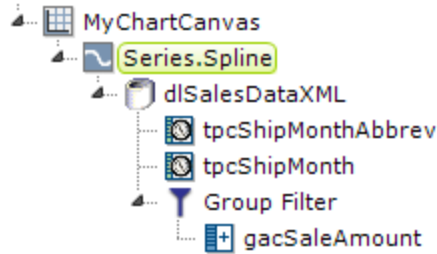
About Series.Spline

The **Series.Spline** element generates a Spline chart, which is commonly used to represent aggregated totals, as numbers or percentages, over time. It's similar to a Line chart but plots a fitted curve through the data points.

Monthly Sales 2013




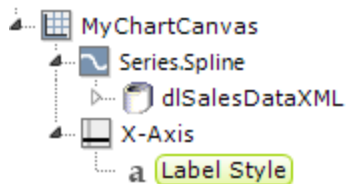
The example above shows a simple Spline chart, representing sales per month for a year.



| Element - Series.Spline | |
|-----------------------------|--------------------|
| *Required Attributes | |
| Y-axis Data Column | gacSaleAmount |
| Optional Attributes | |
| Color | |
| Combine With Series ID | |
| ... | |
| X-axis Data Column | tpcShipMonthAbbrev |
| X-axis Data Column Type | |

As shown above, the chart is created by adding Series.Spline to the canvas, along with a datalayer and, typically, some child elements that may include **Time Period Column** elements, a **Group Filter**, and a **Group Aggregate Column** element. Very few attributes need to be set for the Series element in order to produce a basic chart.

 A datalayer element can be used either beneath Series.Spline, as shown above, or beneath Chart Canvas. If used as a child of Chart Canvas, its data is available to *all* child Series elements. This can improve performance if you have several series, all using the *same* data, beneath the same Chart Canvas element.

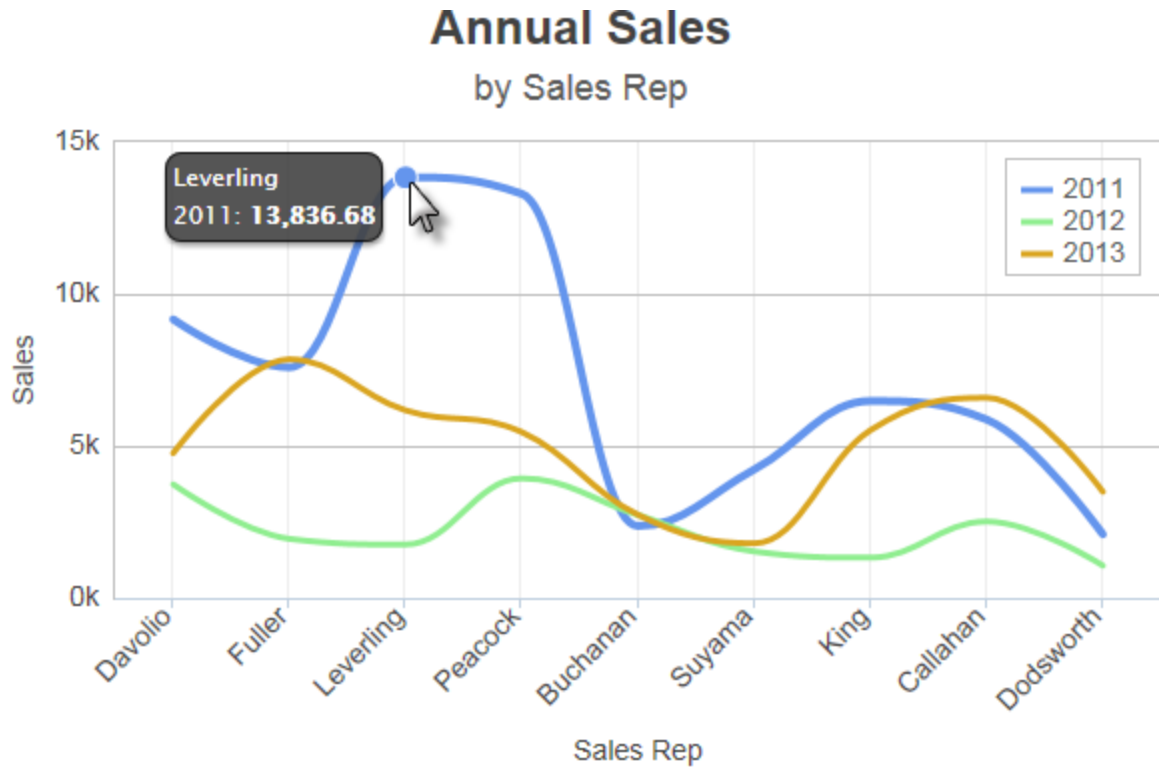


| Element - AxisLabelStyle | |
|----------------------------|----|
| Optional Attributes | |
| Font Angle | 45 |
| Font Color | |
| ... | |
| Stagger Labels | |

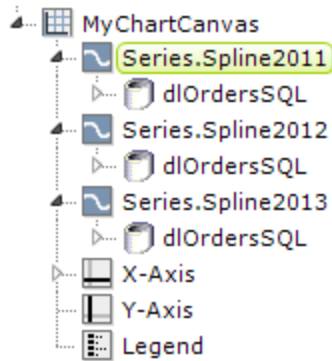
The properties of the X- and Y-axis, including captions, are set using the **X-Axis** and **Y-Axis** elements. For example, in order to angle the X-axis labels, add an X-Axis element beneath Chart Canvas (none of its attributes need to be set) and add its child **Label Style** element. Set the Label Style element's attribute as shown above.

Series.Spline - Using Multiple Series

You can add additional series to the chart by adding additional Series elements:



The example above shows three Series, one for each year, with a legend.



| Element - Series.Spline | |
|-----------------------------|-----------------|
| *Required Attributes | |
| Y-axis Data Column | sumSales |
| Optional Attributes | |
| Color | CornflowerBlue |
| Combine With Series ID | |
| ... | |
| ID | Series.Line2011 |
| Legend Label | 2011 |
| Line Style | |
| Line Thickness | 3 |
| ... | |
| X-axis Data Column | LastName |
| X-axis Data Column Type | |

The example above shows the three Series elements, their datalayers, the X- and Y-Axis elements, and the **Legend** element used to produce the previous chart. You can adjust which series appears "in front" of the others in the chart by changing the order of the Series elements in the definition. Setting the Series elements' **Legend Label** attribute will automatically cause the legend to be displayed.

Charts with multiple series or aggregations are encouraged to use the ChartCanvas element's attribute, `Tooltip Split`. This attribute allows you to split a chart's tooltip into one label per series, rather than per segment. The default value for this attribute is *False*. Once enabled, when hovering over a data point, it displays all the values for the series. This method is recommended over shared tooltips for charts with multiple line series, and as such, takes precedence over `tooltip.shared`.



When using multiple series, you may be able to reduce the number of data queries and improve performance by using local data to read all of the data once, see *Datalayer Introduction*. Then, link its datalayer to share it to the series, see *Link Datalayers*. At each Series element, link its datalayer to the shared Local Data datalayer and filter the data to meet the needs of each individual series.

You can combine different types of Series elements, for example, Series.Spline and Series.Bar, to produce combinations of visualizations.

The screenshot shows a tree view on the left with the following elements: MyChartCanvas2, Series.Spline2011, Series.Spline2012, Series.Spline2013, X-Axis, Y-Axis, and Legend. The Legend element is highlighted with a yellow box. A blue arrow points from the Legend element to the properties panel on the right. The properties panel is titled 'Element - ChartCanvasLegend' and shows the following 'Optional Attributes':

| Optional Attributes | |
|---------------------|------------|
| Alignment | Horizontal |
| ... | |
| Border Radius | 0 |
| Border Thickness | |
| ... | |
| Inside Plot Area | True |
| Legend Orientation | Vertical |
| Maximum Height | |
| Offset X | -10 |
| Offset Y | 72 |
| ... | |
| Width | |

In order to produce the legend shown in the previous example, a **Legend** element must be added and its attributes configured as shown above.

v23.1 If you are using the Chart Color Axis element in a multi-series chart, by default, the series will link to the first color axis. To link the series to a special color axis, set the ID attribute of the Chart Color Axis element to the corresponding Series' Linked to Color Axis ID attribute. For more information, see "Chart Color Axis" on page 124.

Series.Spline - Attributes

The Series.Spline element has the following attributes:

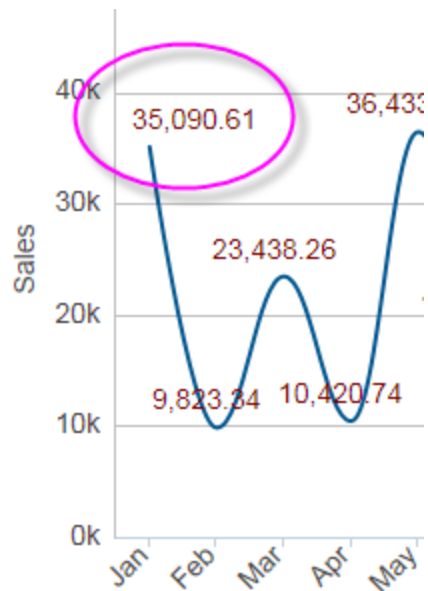
| Attribute | Description |
|--|---|
| Y-Axis Data Column | (Required) Specifies the name of a datalayer column whose values will be plotted along the Y-axis. |
| v23.1 Class Name | Specifies the Class Name referenced in the .css style sheet. The default value is <i>undefined</i> . |
| Color | Sets the line color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. @Chart tokens may be used here to provide dynamic, data-driven line segment colors. For example, the token @Gradient can be used to represent the data with a gradient fill. |
| v23.1 Color Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the x-axis. |
| Combine With Series ID | Set this attribute to the element ID of another series to combine it with this series in the legend. When two series are combined, by default only the first one will appear in the legend but clicking the item in the legend toggles both series to appear and disappear. Or, the value <i>Previous</i> can be entered to combine this series with the previous series. |
| Connect Nulls | Specifies if data rows with null or blank values are to be ignored, allowing adjacent values to be connected in the chart. The default value is <i>False</i> . |

| Attribute | Description |
|--|--|
| Disable Mouse Tracking | Disables mouse tracking for the series, when set to <i>True</i> . This affects tooltips and click events on graphs and points. For large datasets, this may improve performance. The default value is <i>False</i> . |
| Hover Line Thickness | Sets the thickness of the line, in pixels, when the mouse pointer is hovered over it. The default value is <i>1</i> pixel. |
| Legend Label | Indicates text that will be shown for this series inside the chart legend. When a value is provided, automatically causes the legend to be displayed. |
| Line Style | Specifies the pattern of the line as either <i>Solid</i> or a combination of dashes and dots. @Chart tokens may be used here to provide dynamic, data-driven line segment styles. |
| Line Thickness | Specifies the thickness of the line, in pixels. The default value is <i>1</i> pixel. |
| v23.1 Linked to Color-Axis ID | Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes |
| Linked to X-Axis ID | Specifies the ID of an X-Axis element that this series should be linked to when using multiple X-axes. |
| Linked to Y-Axis ID | Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes. |

| Attribute | Description |
|-----------------------------|---|
| Negative Color | Sets the color for negative values. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #D5F484. The default value is <i>Red</i> . |
| Negative Color Transparency | Specifies the transparency of the Negative Color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| Negative Threshold | Sets the positive-negative value threshold, also called the "zero level" or "base level". The default value is <i>0</i> . |
| Transparency | Specifies the transparency of the line color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| X-Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the X-axis. |
| X-Axis Data Column Type | Specifies the data type of the datalayer column named in the X-axis Data Column attribute. Options include <i>Auto</i> (the default), <i>Text</i> , <i>Number</i> , and <i>DateTime</i> . By default, X-axis data values that are <i>DateTime</i> type will be automatically distributed evenly across the time series. If you want to disable this behavior, set this attribute value to <i>Text</i> . |

Series.Spline - Using the Data Labels Element

A "data label" is text shown next to each data point that shows its value. When the **Data Labels** element is used as a child of **Series.Spline**, text representing the data values will appear on the chart:

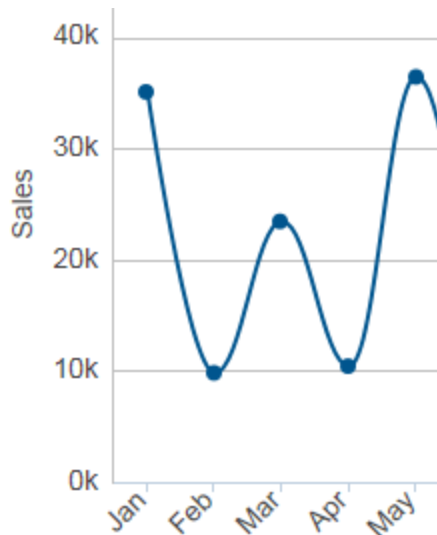


The Data Labels element has attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text. One attribute, Series Name as Caption, allows you to specify the y-axis label, x-axis label, or legend label as the data point value inside the chart.

The Data Labels element's color-related attribute values can be set using @Chart tokens. **v23.1** Or, utilize the "Class Name" attribute to apply unique styling to your data labels by referencing a class name from a stylesheet in the report.

Series.Spline - Using the Marker Points Element

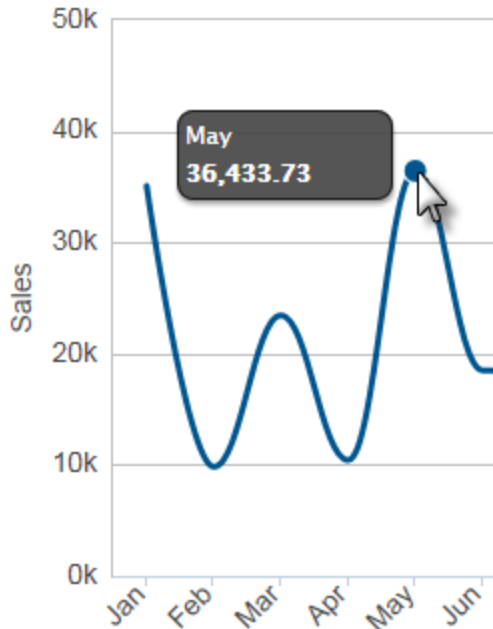
A "marker point" is a symbol that appears on the chart at each data point. When the **Marker Points** element is used as a child of Series.Spline, a small dot matching the color of the line will be displayed at each data point:



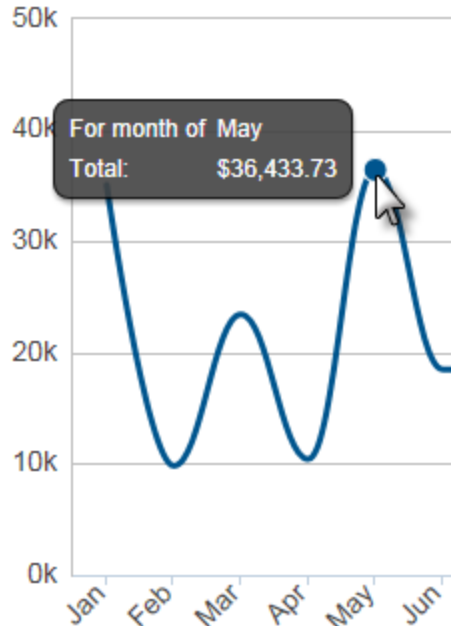
The default marker points are shown above. The **Marker Points** element allows you to select a different symbol for the marker point, and to control its size, color, border color, and transparency. When the cursor hovers over a marker point, it's automatically enlarged slightly.

Series.Spline - Using the Quicktips Element

By default, a "quicktip" is displayed when the mouse hovers near or over a data point:




Default quicktip



With Quicktip child element

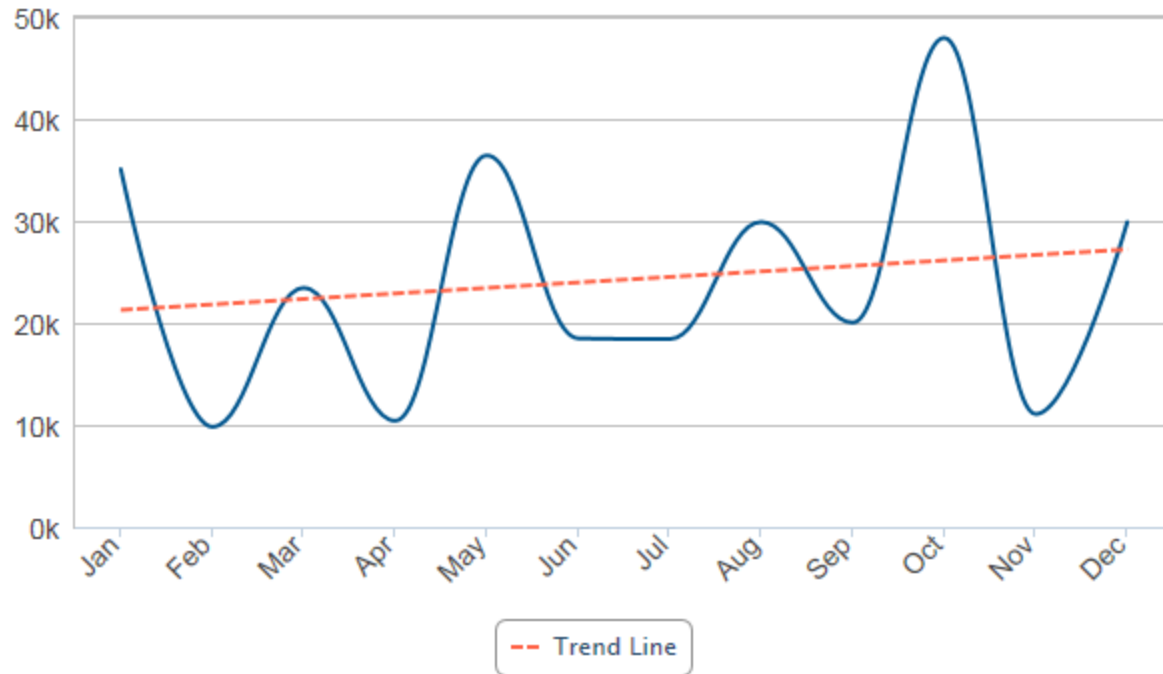
The automatically-generated quicktip displays information for the X- and Y-axis, as shown above, left. However, you may want to display other information or format it differently, perhaps as shown above, right, which can be done by adding a **Quicktip** child element beneath Series.Spline and setting its attributes and child elements. Use @Chart tokens to include chart data in the quicktip. Intrinsic functions are supported in the Quicktip attributes.

You can also configure the tooltip to display values for any available column in the datalayer. To do so, enter a token representing the data column in the Value attribute of the Quicktip Row element.

 To use this feature with `DataLayer.ActiveSQL`, please make sure the `keep Grouped Rows` attribute of the `SqlGroup` element is set to *False*.

Series.Spline - Using the Trend Line Element

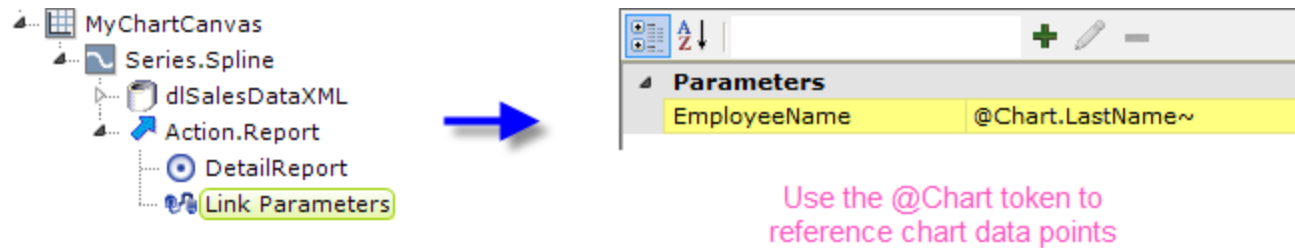
The **Trend Line** element creates a line on the chart that indicates the "trend" of the data. The line connects a number of data points generated using a regression algorithm.



The Trend Line element is a child of the Series.Spline element and can be styled for color, line width, etc. When configured with a legend label, it will be represented by an item in the legend, as shown above.

Series.Spline - Using Action Elements

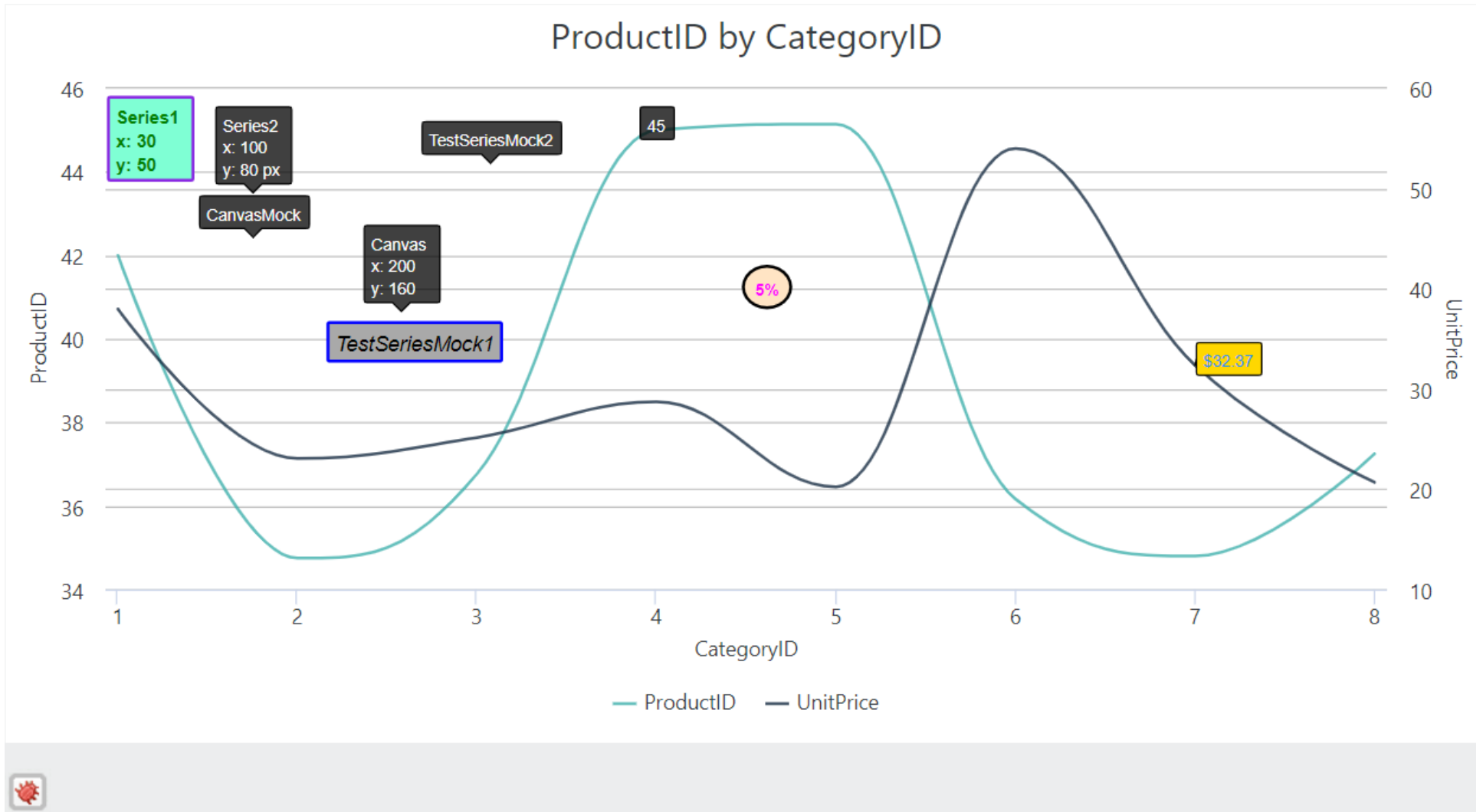
Action elements initiate processing of a report or process task definition, redirection to a link, or other processing when a data point in the series is clicked.



In the example above, an **Action.Report** element has been added as a child of the Series, along with its **Target.Report** and **Link Parameters** child elements. To reference chart data in parameters, use the @Chart token, as shown above. A variety of Action elements are available for use with Series, including Action.Link, Action.Process, and Action.Refresh Element. Additional Action elements will be added in future releases.

Series.Spline - Using the Series Annotation Element

Annotations allow you to annotate a chart freely using custom labels and shapes. When the **Series Annotation** element is used as a child of Series.Spline, you can place these annotations at various points of interest:



The Series Annotation element offers two child elements: **AnnotationLabel.Point** and **AnnotationLabel.Mock**.

The AnnotationLabel.Point child element is used to create an annotation label and position on the chart by linking it to an existing point. With this element, you cannot specify the value, it always refers to the value of the point. Additionally, you can use the Caption and Condition attributes to refer to different datalayers.

On the other hand, the AnnotationLabel.Mock is used to create an annotation label and position it on the chart by linking it to a created mock point. With this element, you can specify the coordinate, relative position, and axis.

Both the AnnotationLabel.Point and AnnotationLabel.Mock have attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text.

Series.Spline - Using Input Selection

This series can also be used with the **Input Selection** family of elements, which turn the chart into an input control. This allows users, at runtime, to select points or values on the chart with their mouse and your report can then take action using the selected data. This is very useful, for example, for drilling into the data. For more information about this functionality, see "Input Selection" on page 97.

Series.Spline - Using the Refresh Series Timer

The **Refresh Series Timer**, added as a Series child element, updates charts automatically, based on a time interval. When the interval is reached, a request is sent back to the web server for updated data. Series are updated with a smooth animation.



The **Refresh Interval** attribute specifies the number of seconds to wait before refreshing the series data. A value is required and must be an integer greater than 0.

Refresh Mode

Series may be refreshed in two ways: either all of the data is refreshed with each request, or the series data automatically slides to the left one data interval per refresh. The refresh mode is selected in the Chart X Axis element's **Axis Type** attribute. When the Axis Type is *not* set to *DateTimeLinear*, all of the data will be refreshed.

When the Axis Type value is *DateTimeLinear*, newer values are added to the right side of the chart, previous values slide left, and older values disappear as they reach the left edge of the chart. In this case, the Refresh Series Timer element's **Time Span** attribute is used to specify the age of the data included. This value must be in *hh:mm:ss* format (for example "00:02:00" for two minutes) and must be larger than the Refresh Interval attribute value.

When a Time Span attribute is set, the series' datalayer can make use of a special timestamp token, `@Chart.rdTimeSpanStart~`, in a query to retrieve just the rows necessary to update the chart. On the initial request, this token returns the current time minus the Time Span value. For subsequent refreshes, when *Axis Type = DateTimeLinear* and the X-axis will be sliding, the token returns the last time the chart was updated, so only the newest rows are retrieved. Here's an MS SQL Server query example:

```
SELECT * FROM MyTable WHERE UpdateTime BETWEEN '@Chart.rdTimeSpanStart~' AND '@Function.DateTime~'
```

For best performance, avoid setting a very short refresh interval if a very large number of users will be displaying the report.

Series.Waterfall

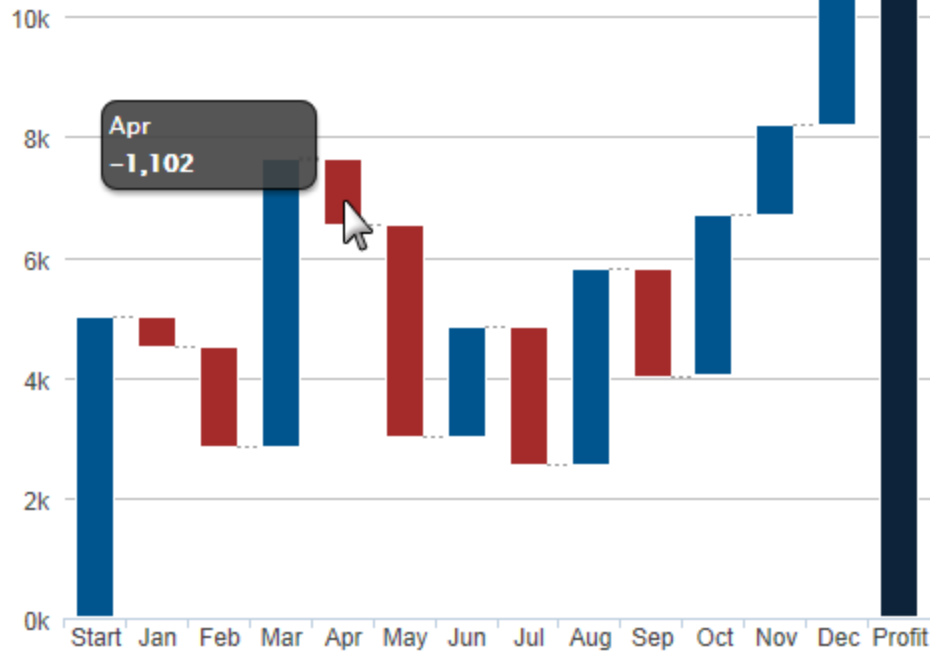
The Chart Canvas element's Series child elements cause a data visualization (the chart) to be rendered in the canvas.

The following topics discuss the Series.Waterfall child element:

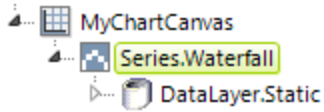
- [Using Multiple Series](#)
- [Series.Waterfall Attributes](#)
- [Using the Data Labels Element](#)
- [Using the Quicktips Element](#)
- [Using Action Elements](#)
- [Using the Series Annotation Element](#) v23.1
- [Using Input Selection](#)
- [Using the Refresh Series Timer](#)

About Series.Waterfall

The **Series.Waterfall** element generates a Waterfall chart, which helps in understanding the cumulative effect of sequentially introduced positive or negative values.




The example above shows a Waterfall chart, with negative values shown in red. A special "total bar", labeled *Profit* above, is automatically generated from the sum of all other values. It appears as the last bar on the right or, at the top, if the axes are swapped. Its generation can be controlled in the Series attributes.



Element - Series.Waterfall

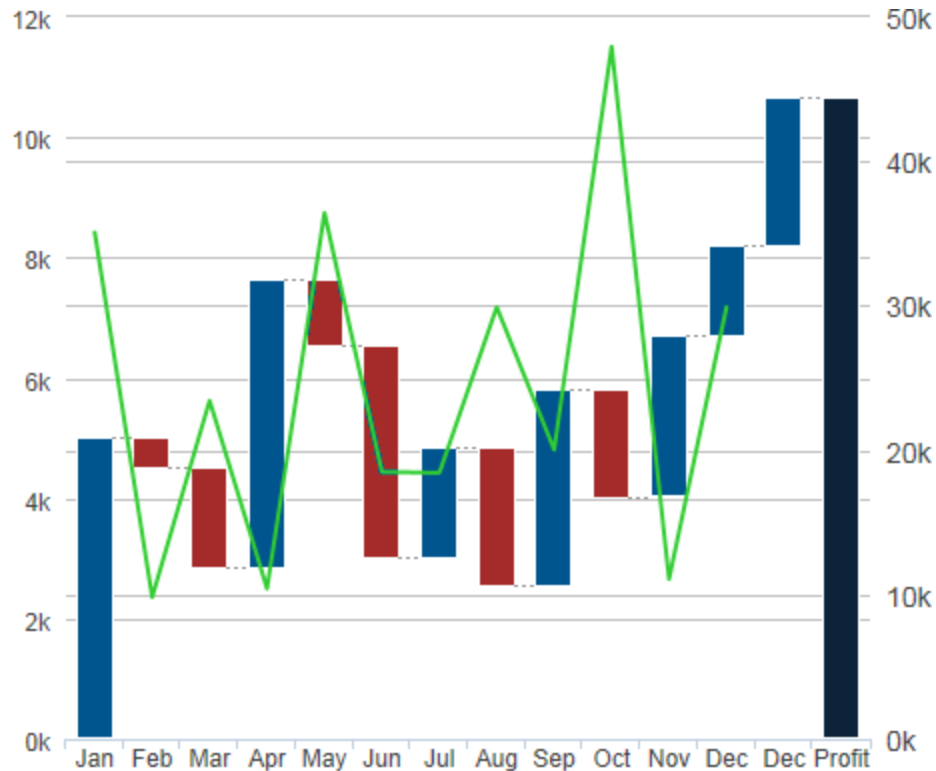
| *Required Attributes | |
|--------------------------|--------|
| Y-axis Data Column | Amount |
| Optional Attributes | |
| Bar Border Color | |
| ... | |
| Label Data Column X-axis | Period |
| ... | |
| Negative Color | Red |
| ... | |
| Total Bar Label | Profit |
| Transparency | |

As shown above, the chart is created by adding Series.Waterfall to the canvas, along with a datalayer. Very few attributes need to be set for the Series element in order to produce a basic chart.

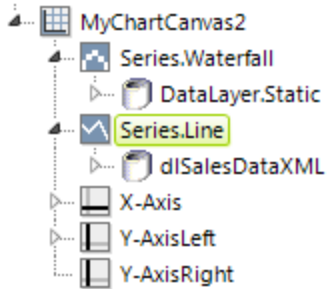
 A datalayer element can be used either beneath Series.Waterfall, as shown above, or beneath Chart Canvas. If used as a child of Chart Canvas, its data is available to *all* child Series elements. This can improve performance if you have several series, all using the *same* data, beneath the same Chart Canvas element.

Series.Waterfall - Using Multiple Series

Waterfall charts do not generally lend themselves to use with other series types, but it is possible. You can add additional series to the chart by adding additional Series elements:



The example above shows Series.Waterfall with Series.Line. The series have been linked to two separate Y-axis elements to provide two separate scales.




Element - Series.Line

| *Required Attributes | |
|-------------------------|--------------------|
| Y-axis Data Column | gacSaleAmount |
| Optional Attributes | |
| Color | LimeGreen |
| ... | |
| Linked to Y-Axis ID | Y-AxisRight |
| Negative Color | |
| ... | |
| X-axis Data Column | tpcShipMonthAbbrev |
| X-axis Data Column Type | |

The example above shows the two Series elements and their datalayers, used to produce the previous chart. You can adjust which series appears "in front" of the other in the chart by changing the order of the Series elements in the definition. The Line series has been linked to one of the two Y-Axis elements, creating a secondary scale on the right-hand side.

Charts with multiple series or aggregations are encouraged to use the ChartCanvas element's attribute, `Tooltip Split`. This attribute allows you to split a chart's tooltip into one label per series, rather than per segment. The default value for this attribute is `False`. Once enabled, when hovering over a data point, it displays all the values for the series. This method is recommended over shared tooltips for charts with multiple line series, and as such, takes precedence over `tooltip.shared`.

 When using multiple series, you may be able to reduce the number of data queries and improve performance by using local data to read all of the data once, see *Datalayer Introduction*. Then, link its datalayer to share it to the series, see *Link Datalayers*. At each Series element, link its datalayer to the shared Local Data datalayer and filter the data to meet the needs of each individual series.

Series.Waterfall - Attributes

The Series.Waterfall element has the following attributes:

| Attribute | Description |
|---|---|
| Y-Axis Data Column | (Required) Specifies the name of a datalayer column whose values determine the height of each bar. |
| Bar Border Color | Sets the color of the thin border line around each bar. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| Bar Border Color Transparency | Specifies the transparency of the thin border line around each bar. The lowest value of 0 specifies that the background is opaque, with no transparency. At the other end of the scale, 15 specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| Bar Border Radius | Sets the amount of rounding for bar corners, in pixels. The default value is 0 pixels, which produces square corners. |
| Bar Border Thickness | Sets the thickness of the bar border lines, in pixels, when the related Bar Border Color attribute has a value. The default value is 1 pixel. |
| Bar Thickness | Sets the width of the bar in pixels. If left blank, the width will be determined automatically. |
| v23.1 Class | Specifies the Class Name referenced in the .css style sheet. The default value is <i>undefined</i> . |

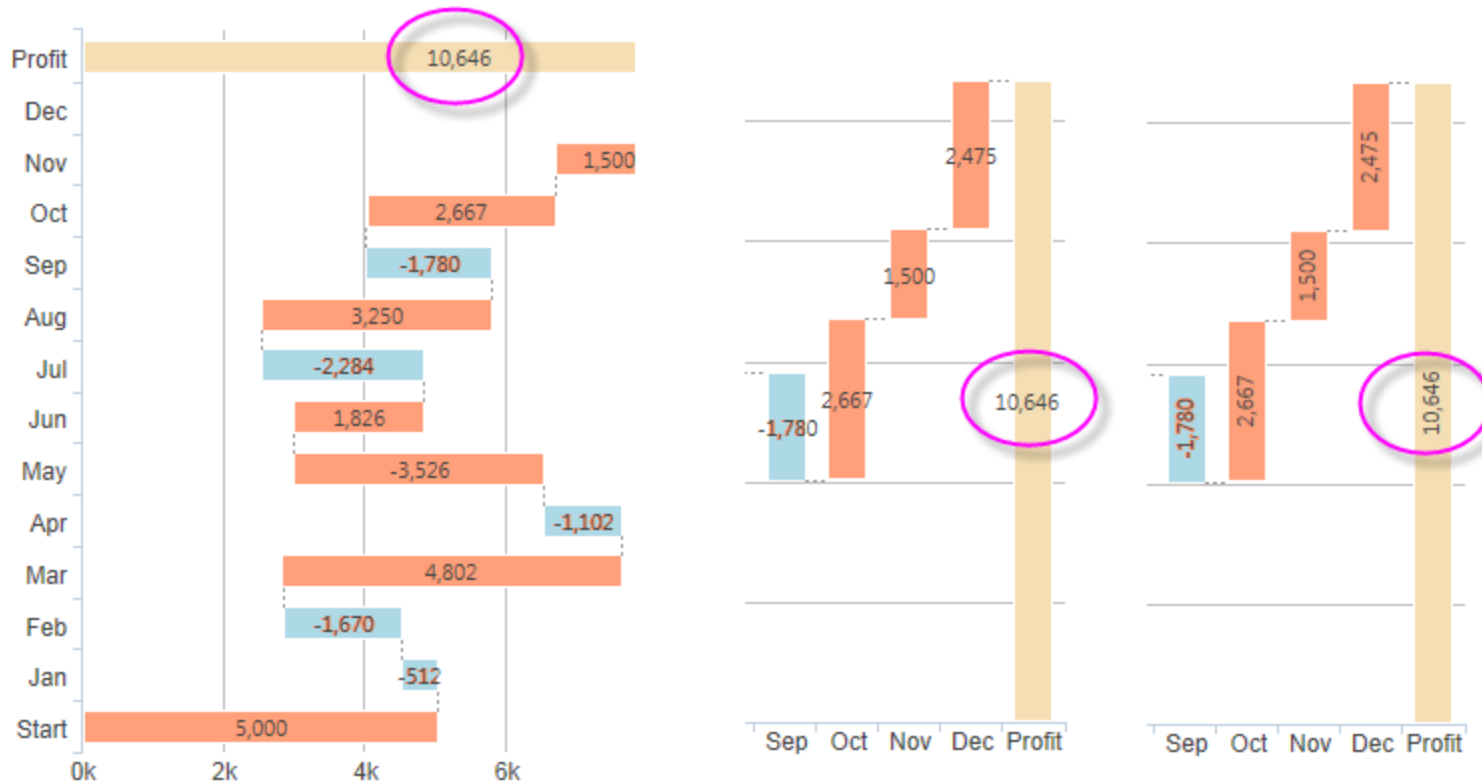
| Attribute | Description |
|---|---|
| Name | |
| Color | Sets the bar fill color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| Combine With Series ID | Set this attribute to the element ID of another series to combine it with this series in the legend. When two series are combined, by default only the first one will appear in the legend but clicking the item in the legend toggles both series to appear and disappear. Or, the value <i>Previous</i> can be entered to combine this series with the previous series. |
| Disable Mouse Tracking | Disables mouse tracking for the series, when set to <i>True</i> . This affects tooltips and click events. For large datasets, this may improve performance. The default value is <i>False</i> . |
| Hover Brightness | Specifies the change in a bar's brightness when the mouse pointer hovers over it. Values can be 0 (no change) through 15(lighter). The default value is 2. |
| Intermediate Sum Bar Color | Sets the fill color of the Intermediate Sum bar. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |
| Intermediate Sum Bar Color Transparency | Specifies the transparency of the thin border line around the Intermediate Sum bar. The lowest value of 0 specifies that the background is opaque, with no transparency. At the other end of the scale, 15 specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |

| Attribute | Description |
|--------------------------|---|
| Intermediate Sum Column | <p>Controls the display of an "Intermediate Sum" bar, which is an extra bar in the chart showing the sum of data values, beginning with the first value, up to a specific point on the X-axis. To show an intermediate sum, configure the datalayer so it includes a "special" row inserted at the point of the desired intermediate sum. In this row, include a column with a value of <i>True</i>. Specify the name of that column in this attribute. Any other data values in this special row are ignored. You can add multiple Intermediate Sum bars by adding additional special rows in the desired places in the datalayer; just duplicate the first special row where desired. However, when calculating each intermediate sum, the previous special row resets the summarization to zero. For example: A special row in the datalayer at record 10 will produce a sum of the values for rows 1-9. A second special row at record 15 will produce a sum of the values for rows 11-14. There is no cumulative or running total effect for all rows.</p> |
| Label Data Column X-axis | <p>Specifies the name of a datalayer column whose values be represented by the bars.</p> |
| Legend Label | <p>Indicates text that will be shown for this series inside the chart legend. When a value is provided, it automatically causes the legend to be displayed.</p> |
| Linked to X-Axis ID | <p>Specifies the ID of an X-Axis element that this series should be linked to when using multiple X-axes.</p> |
| Linked to Y-Axis ID | <p>Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes.</p> |
| Negative Color | <p>Sets the color for negative values. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #D5F484.</p> |

| Attribute | Description |
|------------------------------|--|
| Negative Color Transparency | Specifies the transparency of the Negative Color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| Total Bar Color | Sets the color for the automatically-generated "total bar" at the right side of the chart. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #D5F484. |
| Total Bar Color Transparency | Specifies the transparency of the Total Bar Color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| Total Bar Label | Specifies the label text for the total bar. When set to <i>None</i> , the total bar will not be generated. The default value is <i>Total</i> . |
| Transparency | Specifies the transparency of the general chart bars. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different series to show through each other. |

Series.Waterfall - Using the Data Labels Element

A "data label" is text shown near each data point that shows its value. When the **Data Labels** element is used as a child of Series.Waterfall, text representing the data values will appear on the bars:



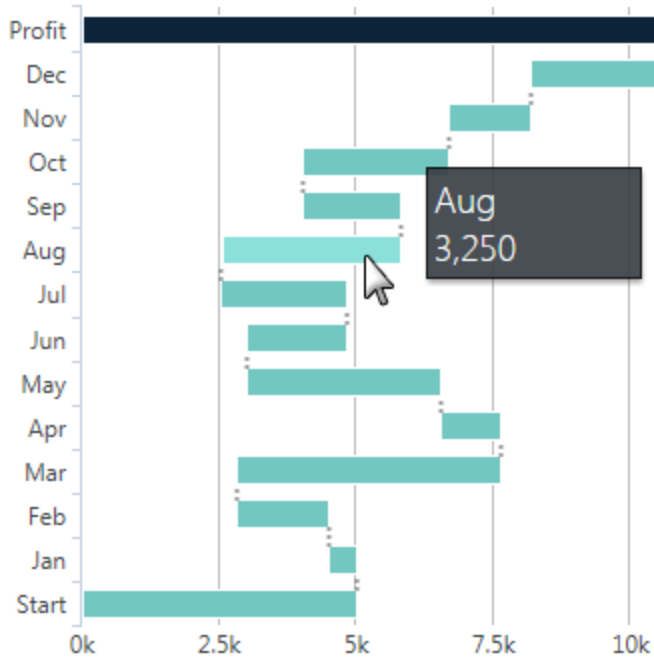
The example above, right, shows data labels placed inside the bars and rotated 90-degrees. The Data Labels element has attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text.

One attribute, Series Name as Caption, allows you to specify the y-axis label, x-axis label, or legend label as the data point value inside the chart.

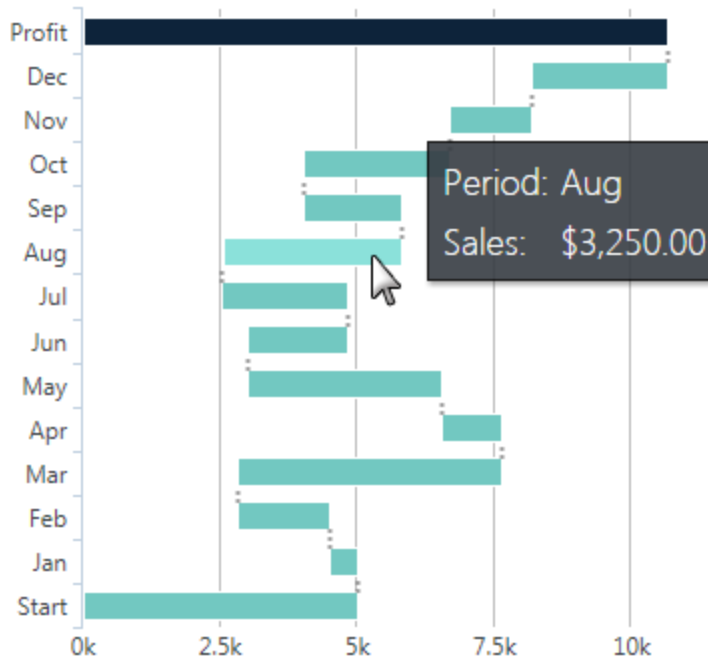
The Data Labels element's color-related attribute values can be set using @Chart tokens. **v23.1** Or, utilize the "Class Name" attribute to apply unique styling to your data labels by referencing a class name from a stylesheet in the report.

Series.Waterfall - Using the Quicktips Element

By default, a "quicktip" is displayed when the mouse hovers over a bar:




Default quicktip



With Quicktip child element

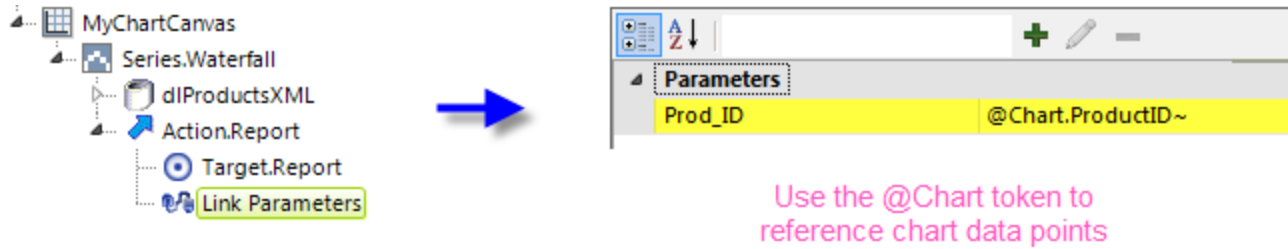
The automatically-generated quicktip displays information for the X- and Y-axis, as shown above, left. However, you may want to display other information or format it differently, perhaps as shown above, right. This can be done by adding a **Quicktip** child element beneath Series.Waterfall and setting its attributes and child elements. Use @Chart tokens to include chart data in the quicktip. Intrinsic functions are supported in the Quicktip attributes.

You can also configure the tooltip to display values for any available column in the datalayer. To do so, enter a token representing the data column in the Value attribute of the Quicktip Row element.

 To use this feature with `DataLayer.ActiveSQL`, please make sure the `keep Grouped Rows` attribute of the `SqlGroup` element is set to *False*.

Series.Waterfall - Using Action Elements

Action elements initiate processing of a report or process task definition, redirection to a link, or other processing when a bar is clicked.

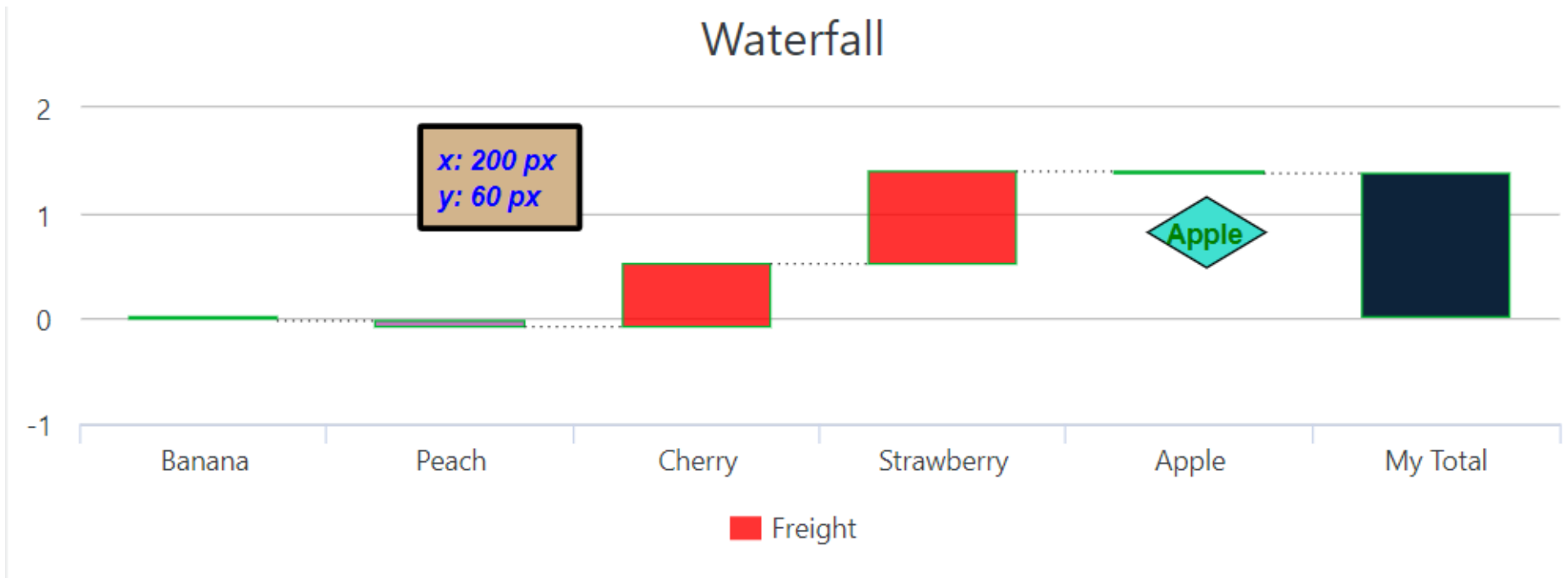


In the example above, an **Action.Report** element has been added as a child of the Series, along with its **Target.Report** and **Link Parameters** child elements. To reference chart data in parameters, use the @Chart token, as shown above.

A variety of Action elements are available for use with Series, including Action.Link, Action.Process, and Action.Refresh Element. Additional Action elements will be added in future releases.

Series.Waterfall - Using the Series Annotation Element

Annotations allow you to annotate a chart freely using custom labels and shapes. When the **Series Annotation** element is used as a child of Series.Waterfall, you can place these annotations at various points of interest:



The Series Annotation element offers two child elements: **AnnotationLabel.Point** and **AnnotationLabel.Mock**.

The AnnotationLabel.Point child element is used to create an annotation label and position on the chart by linking it to an existing point. With this element, you cannot specify the value, it always refers to the value of the point. Additionally, you can use the Caption and Condition attributes to refer to different datalayers.

On the other hand, the `AnnotationLabel.Mock` is used to create an annotation label and position it on the chart by linking it to a created mock point. With this element, you can specify the coordinate, relative position, and axis.

Both the `AnnotationLabel.Point` and `AnnotationLabel.Mock` have attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text.

Series.Waterfall - Using Input Selection

This series can also be used with the **Input Selection** family of elements, which turn the chart into an input control. This allows users, at runtime, to select points or values on the chart with their mouse and your report can then take action using the selected data. This is very useful, for example, for drilling into the data.

For more information about this functionality, see "Input Selection" on page 97.

Series.Waterfall - Using the Refresh Series Timer

The **Refresh Series Timer**, added as a Series child element, updates charts automatically, based on a time interval. When the interval is reached, a request is sent back to the web server for updated data. Series are updated with a smooth animation.



The **Refresh Interval** attribute specifies the number of seconds to wait before refreshing the series data. A value is required and must be an integer greater than 0.

Refresh Mode

Series may be refreshed in two ways: either all of the data is refreshed with each request, or the series data automatically slides to the left one data interval per refresh. The refresh mode is selected in the Chart X Axis element's **Axis Type** attribute. When the Axis Type is *not* set to *DateTimeLinear*, all of the data will be refreshed.

When the Axis Type value is *DateTimeLinear*, newer values are added to the right side of the chart, previous values slide left, and older values disappear as they reach the left edge of the chart. In this case, the Refresh Series Timer element's **Time Span** attribute is used to specify the age of the data included. This value must be in *hh:mm:ss* format (for example "00:02:00" for two minutes) and must be larger than the Refresh Interval attribute value.

When a Time Span attribute is set, the series' datalayer can make use of a special timestamp token, `@Chart.rdTimeSpanStart~`, in a query to retrieve just the rows necessary to update the chart. On the initial request, this token returns the current time minus the Time Span value. For subsequent refreshes, when *Axis Type = DateTimeLinear* and the X-axis will be sliding, the token returns the last time the chart was updated, so only the newest rows are retrieved. Here's an MS SQL Server query example:

```
SELECT * FROM MyTable WHERE UpdateTime BETWEEN '@Chart.rdTimeSpanStart~' AND '@Function.DateTime~'
```

For best performance, avoid setting a very short refresh interval if a very large number of users will be displaying the report.

Series.Whiskers

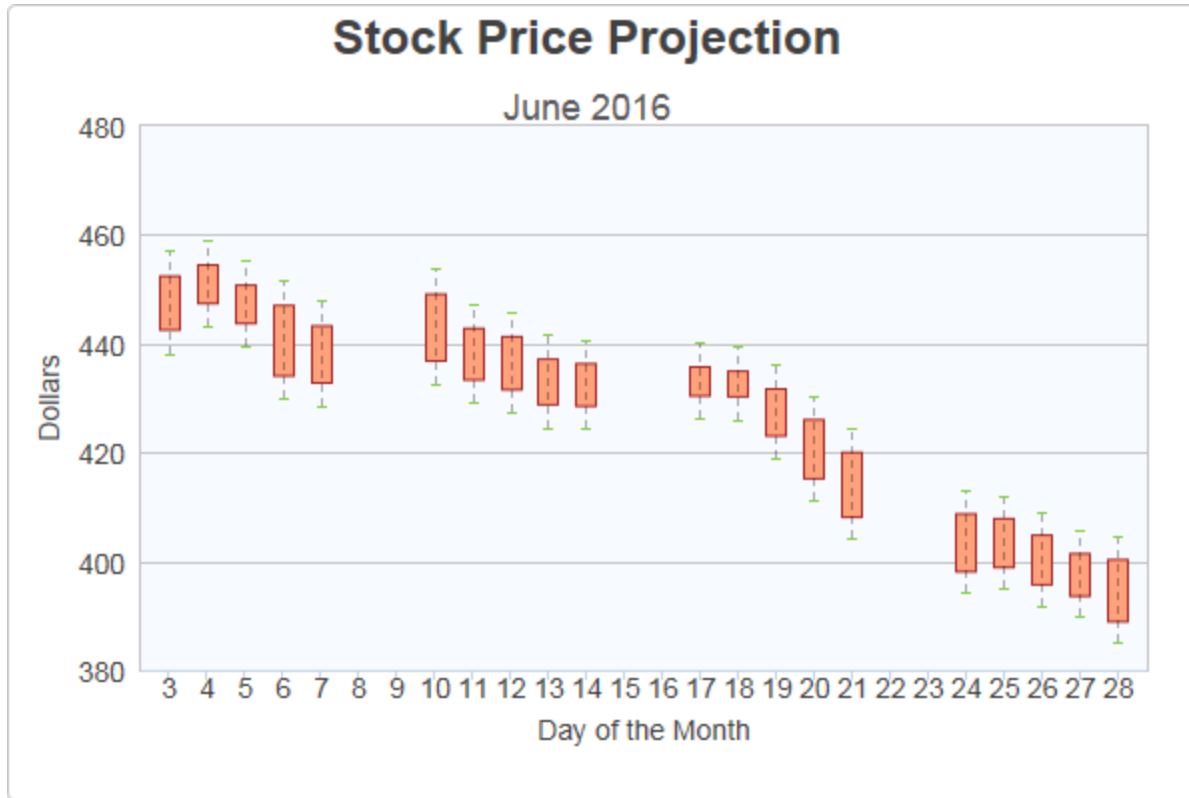
The Chart Canvas element's Series child elements cause a data visualization (the chart) to be rendered in the canvas.

The following topics discuss the Series.Whiskers child element:

- [Series.Whiskers Attributes](#)
- [Using the Quicktips Element](#)
- [Using Action Elements](#)
- [Using the Series Annotation Element](#) v23.1
- [Using the Refresh Series Timer](#)

About Series.Whiskers

The **Series.Whiskers** element generates graphics which, when used with other Series (usually Series.Bar Range), represents the *variability* of the data. It's most often used to indicate the uncertainty or the "margin of error" in the data.




The example above shows a Bar Range chart, representing projected stock prices for a month, with "whiskers" showing the margin of error in the projections.

The screenshot shows a tree view on the left with 'Body' containing 'MyChartCanvas1', which includes 'DataLayer.XML File', 'Series.Bar Range', and 'Series.Whiskers'. A blue arrow points from 'Series.Whiskers' to a configuration table titled 'Element - Series.Whiskers'. The table lists attributes for the whiskers series.

| *Required Attributes | |
|-------------------------|-------------|
| High Value Data Column | calcHighVal |
| Low Value Data Column | calcLowVal |
| Optional Attributes | |
| ... | |
| Stem Color Transparency | 10 |
| Stem Line Style | Dash |
| Stem Line Thickness | |
| Whisker Color | Green |
| ... | |
| X-axis Data Column | DayOfMonth |
| X-axis Data Column Type | |

Below the table, a small diagram shows a vertical whisker with a central stem and horizontal caps. Pink arrows point from the 'Stem Line Style' and 'Whisker Color' rows in the table to the corresponding parts of the diagram.

As shown above, the chart is created by adding Series.Bar Range and Series.Whiskers to the canvas, along with a datalayer and, typically, some datalayer child elements that may include a **Group Filter**, and a **Group Aggregate Column** element. Very few attributes need to be set for the Series element in order to produce a basic chart.

 A datalayer element *must* be used beneath the Chart Canvas element because, unlike other Series elements, Series.Whiskers cannot have its own child datalayer. Both series will then use this datalayer.

Series.Whiskers - Attributes

The Series.Whiskers element has the following attributes:

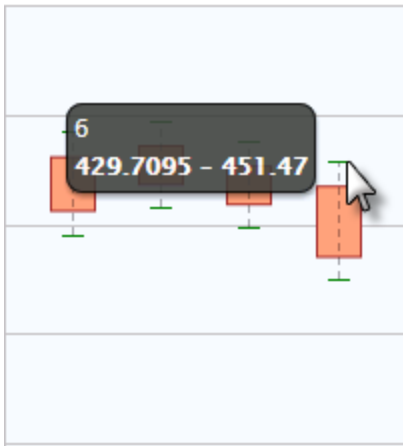
| Attribute | Description |
|--|---|
| High Value Data Column | (Required) Specifies the name of a datalayer column whose values will be plotted as the highest possible value. |
| Low Value Data Column | (Required) Specifies the name of a datalayer column whose values will be plotted as the lowest possible value. |
| v23.1 Color Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the x-axis. |
| Combine With Series ID | Set this attribute to the element ID of another series to combine it with this series in the legend. When two series are combined, by default only the first one will appear in the legend but clicking the item in the legend toggles both series to appear and disappear. Or, the value <i>Previous</i> can be entered to combine this series with the previous series. |
| Disable Mouse Tracking | Disables mouse tracking for the series, when set to <i>True</i> . This affects tooltips and click events on graphs and points. For large datasets, this may improve performance. The default value is <i>False</i> . |
| ID | Specifies a unique identifier for this element. |

| Attribute | Description |
|--|---|
| <p>v23.1</p> <p>Linked to Color-Axis ID</p> | <p>Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes</p> |
| <p>Linked to X-Axis ID</p> | <p>Specifies the ID of an X-Axis element that this series should be linked to when using multiple X-axes.</p> |
| <p>Linked to Y-Axis ID</p> | <p>Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes.</p> |
| <p>Stem Color</p> | <p>Specifies the color of the "stem", the vertical line extending from the bar graphic to the whiskers. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #D5F484. Or, enter the token @Gradient to use a gradient fill to represent the data. The default value is <i>Black</i>.</p> |
| <p>Stem Color Transparency</p> | <p>Specifies the transparency of the Stem Color. The lowest value of <i>0</i> specifies that the stem line is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent stem line. Use medium-level transparency to allow different chart layers to show through each other.</p> |
| <p>Stem Line Style</p> | <p>Specifies the pattern of the stem line as either <i>Solid</i> or as a combination of dashes and dots. The default value is <i>Solid</i></p> |
| <p>Stem Line Thickness</p> | <p>Sets the thickness of the stem line, in pixels. The default value is <i>1</i> pixel.</p> |
| <p>Whisker Color</p> | <p>Specifies the color of the "whiskers", the horizontal lines marking the high and low values. Enter a color by</p> |

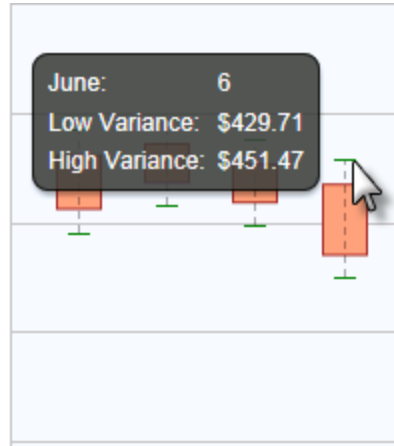
| Attribute | Description |
|----------------------------|---|
| | name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #D5F484. Or, enter the token @Gradient to use a gradient fill to represent the data. The default value is <i>Black</i> . |
| Whisker Color Transparency | Specifies the transparency of the Whisker Color. The lowest value of <i>0</i> specifies that the whisker lines are opaque, with no transparency. At the other end of the scale, <i>15</i> specifies completely transparent whisker lines. Use medium-level transparency to allow different chart layers to show through each other. |
| Whisker Line Thickness | Sets the thickness of the whisker lines, in pixels. The default value is <i>1</i> pixel. |
| Whisker Width | Sets the horizontal width of the whisker lines, in pixels. The default value is <i>33</i> pixels. |
| X-Axis Data Column | Specifies the name of a datalayer column whose values are being plotted along the X-axis. |
| X-Axis Data Column Type | Specifies the data type of the datalayer column named in the X-axis Data Column attribute. Options include <i>Auto</i> (the default), <i>Text</i> , <i>Number</i> , and <i>DateTime</i> . By default, X-axis data values that are <i>DateTime</i> type will be automatically distributed evenly across the time series. If you want to disable this behavior, set this attribute value to <i>Text</i> . |

Series.Whiskers - Using the Quicktips Element

By default, a "quicktip" is displayed when the mouse hovers near or over a stem or whisker line. This is separate from the quicktips which are displayed for the chart bars.




Default quicktip



With Quicktip child element

The automatically-generated quicktip displays information for the low and high values, as shown above, left. However, you may want to display other information or format it differently, perhaps as shown above, right, which can be done by adding a **Quicktip** child element beneath Series.Whiskers and setting its attributes and child elements. Use @Chart tokens to include chart data in the quicktip and intrinsic functions are supported in the Quicktip element attributes.

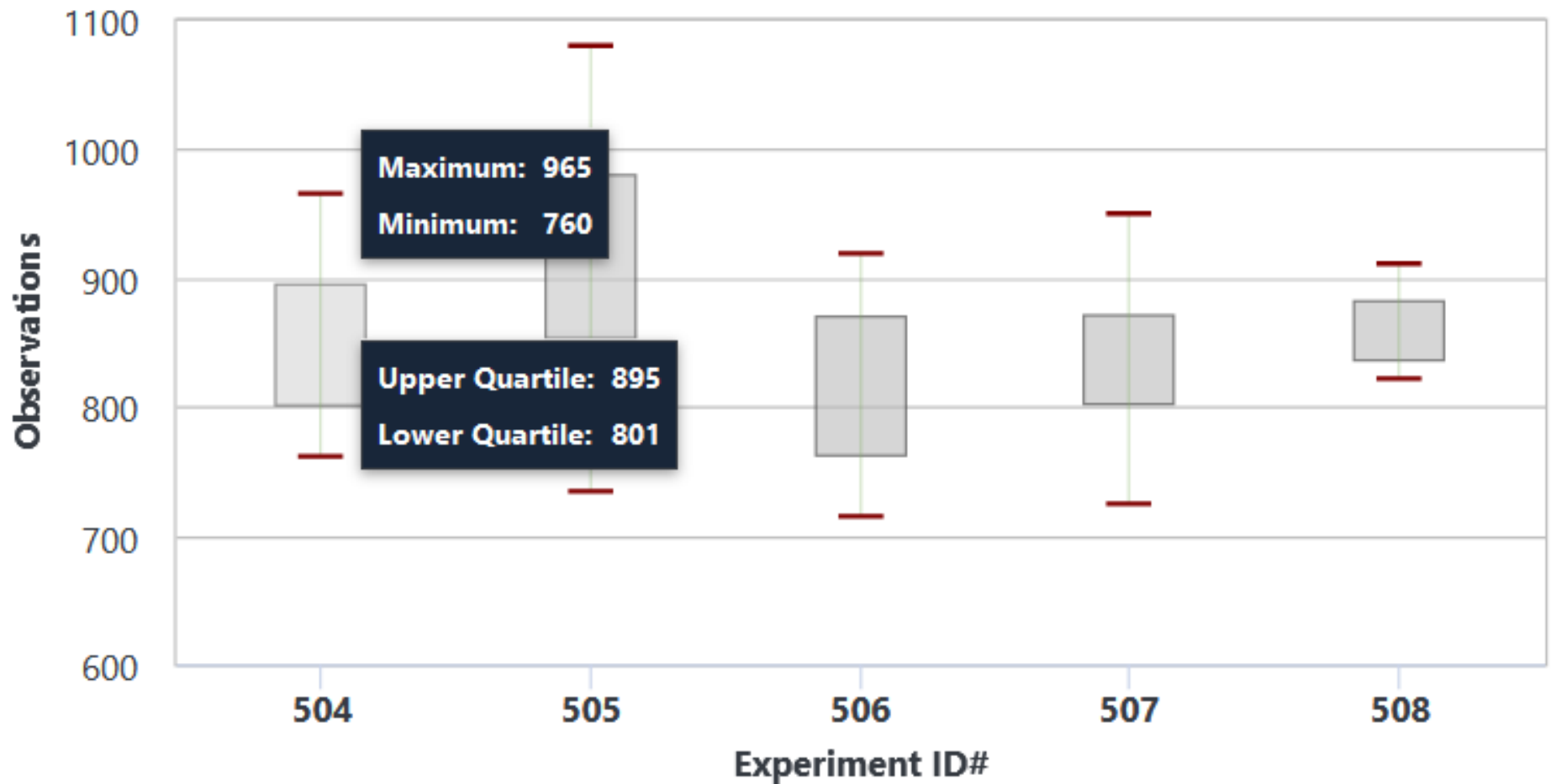
You can also configure the tooltip to display values for any available column in the datalayer. To do so, enter a token representing the data column in the Value attribute of the Quicktip Row element.

 To use this feature with `DataLayer.ActiveSQL`, please make sure the `keep Grouped Rows` attribute of the `SqlGroup` element is set to *False*.

Charts aggregations are encouraged to use the `ChartCanvas` element's attribute, `Tooltip Split`. This attribute allows you to split a chart's tooltip into one label per series, rather than per segment. The default value for this attribute is `False`. Once enabled, when hovering over a data point, it displays all the values for the series:

Clinical Analysis Observations

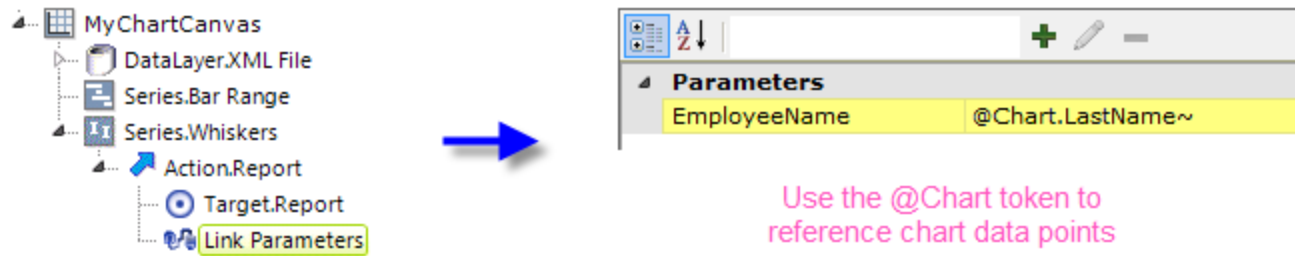
[Get Image](#)



This method is recommended over shared tooltips for charts with multiple line series, and as such, takes precedence over `tooltip.shared`.

Series.Whiskers - Using Action Elements

Action elements initiate processing of a report or process task definition, redirection to a link, or other processing when the stem or whisker line is clicked.



In the example above, an **Action.Report** element has been added as a child of Series.Whiskers, along with its **Target.Report** and **Link Parameters** child elements. To reference chart data in parameters, use the @Chart token, as shown above.

A variety of Action elements are available for use with Series, including Action.Link, Action.Process, Action.Refresh Element and more.

v23.1

Series.Whiskers - Using the Series Annotation Element

Annotations allow you to annotate a chart freely using custom labels and shapes. When the **Series Annotation** element is used as a child of Series.Whiskers, you can place these annotations at various points of interest:



The Series Annotation element offers two child elements: **AnnotationLabel.Point** and **AnnotationLabel.Mock**.

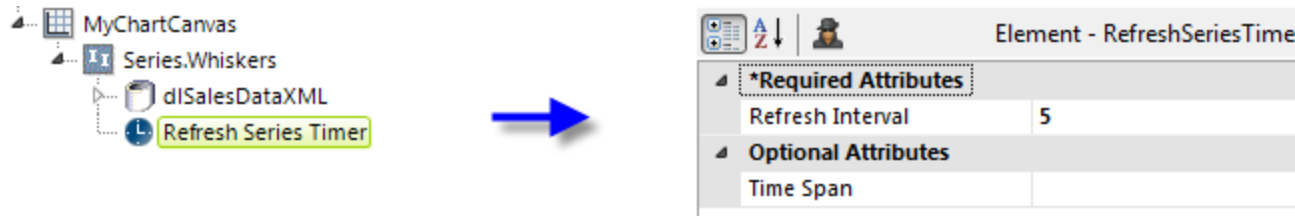
The `AnnotationLabel.Point` child element is used to create an annotation label and position on the chart by linking it to an existing point. With this element, you cannot specify the value, it always refers to the value of the point. Additionally, you can use the `Caption` and `Condition` attributes to refer to different datalayers.

On the other hand, the `AnnotationLabel.Mock` is used to create an annotation label and position it on the chart by linking it to a created mock point. With this element, you can specify the coordinate, relative position, and axis.

Both the `AnnotationLabel.Point` and `AnnotationLabel.Mock` have attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text.

Series.Whiskers - Using the Refresh Series Timer

The **Refresh Series Timer**, added as a Series child element, updates charts automatically, based on a time interval. When the interval is reached, a request is sent back to the web server for updated data. Series are updated with a smooth animation.



The **Refresh Interval** attribute specifies the number of seconds to wait before refreshing the series data. A value is required and must be an integer greater than 0.

Refresh Mode

Series may be refreshed in two ways: either all of the data is refreshed with each request, or the series data automatically slides to the left one data interval per refresh. The refresh mode is selected in the Chart X Axis element's **Axis Type** attribute. When the Axis Type is *not* set to *DateTimeLinear*, all of the data will be refreshed.

When the Axis Type value is *DateTimeLinear*, newer values are added to the right side of the chart, previous values slide left, and older values disappear as they reach the left edge of the chart. In this case, the Refresh Series Timer element's **Time Span** attribute is used to specify the age of the data included. This value must be in *hh:mm:ss* format (for example "00:02:00" for two minutes) and must be larger than the Refresh Interval attribute value.

When a Time Span attribute is set, the series' datalayer can make use of a special timestamp token, `@Chart.rdTimeSpanStart~`, in

a query to retrieve just the rows necessary to update the chart. On the initial request, this token returns the current time minus the Time Span value. For subsequent refreshes, when *Axis Type = DateTimeLinear* and the X-axis will be sliding, the token returns the last time the chart was updated, so only the newest rows are retrieved. Here's an MS SQL Server query example:

```
SELECT * FROM MyTable WHERE UpdateTime BETWEEN '@Chart.rdTimeSpanStart~' AND '@Function.DateTime~'
```

For best performance, avoid setting a very short refresh interval if a very large number of users will be displaying the report.

Series.Bubble

The Chart Canvas element's Series child elements cause a data visualization (the chart) to be rendered in the canvas.

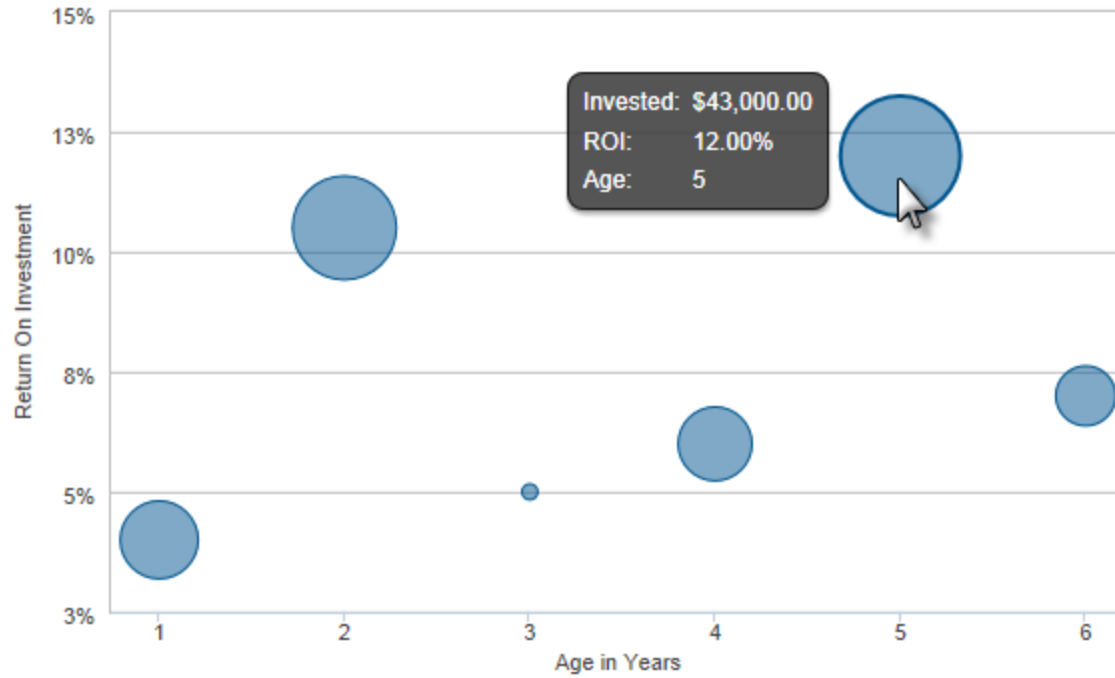
The following topics discuss the Series.Bubble child element:

- [Using Multiple Series](#)
- [Series.Bubble Attributes](#)
- [Using the Data Labels Element](#)
- [Using the Marker Points Element](#)
- [Using the Quicktips Element](#)
- [Using Action Elements](#)
- [Using the Series Annotation Element](#) v23.1
- [Using Input Selection](#)
- [Using the Refresh Series Timer](#)

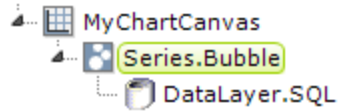
About Series.Bubble

The **Series.Bubble** element generates a Bubble chart, which is commonly used to present three values: two plotted along the X- and Y-axes and a third indicated by the *size* of the bubble symbol.

Investment Performance



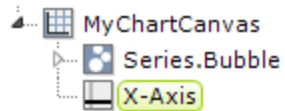
The example above shows a simple Bubble chart, presenting return on investment over time, and the size of each investment.



| Element - Series.Bubble | |
|-----------------------------|-------------|
| *Required Attributes | |
| Size Data Column | AmtInvested |
| Y-axis Data Column | ROI |
| Optional Attributes | |
| Bubble Max Size | |
| ... | |
| X-axis Data Column | Age |
| X-axis Data Column Type | |

As shown above, the chart is created by adding Series.Bubble to the canvas, along with a datalayer. Very few attributes need to be set for the Series element in order to produce a basic chart.

💡 A datalayer element can be used either beneath Series.Bubble, as shown above, or beneath Chart Canvas. If used as a child of Chart Canvas, its data is available to *all* child Series elements. This can improve performance if you have several series, all using the *same* data, beneath the same Chart Canvas element.

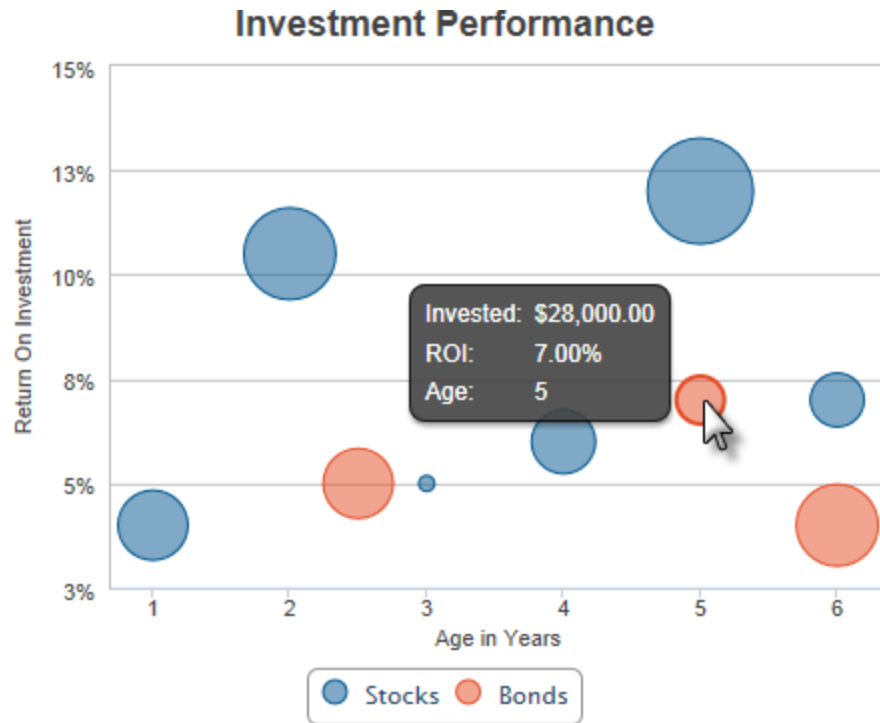


| Element - ChartXAxis | |
|----------------------------|--------------|
| Optional Attributes | |
| Axis Padding Left | |
| ... | |
| Caption | Age in Years |
| ... | |
| Spacing | |

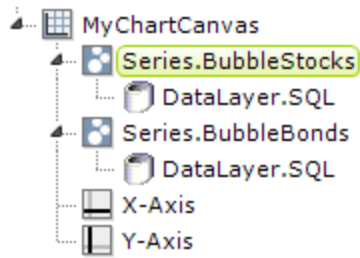
The properties of the X- and Y-axis, including captions, are set using the **X-Axis** and **Y-Axis** elements, as shown above.

Series.Bubble - Using Multiple Series

You can add additional series to the chart by adding additional Series elements:




The example above shows two Series, one for each year, with a legend.



| Element - Series.Bubble | |
|-----------------------------|---------------------|
| *Required Attributes | |
| Size Data Column | AmtInvested |
| Y-axis Data Column | ROI |
| Optional Attributes | |
| Bubble Max Size | |
| ... | |
| ID | Series.BubbleStocks |
| Legend Label | Stocks |
| ... | |
| X-axis Data Column | Age |
| X-axis Data Column Type | |

The example above shows the two Series elements, their datalayers, and **X-** and **Y-Axis** elements used to produce the previous chart. Should they overlap, you can adjust which bubbles appear "in front" of the others in the chart by changing the order of the Series elements in the definition. Setting the Series elements' **Legend Label** attribute will automatically cause the legend to be displayed.

Charts with multiple series or aggregations are encouraged to use the ChartCanvas element's attribute, `Tooltip Split`. This attribute allows you to split a chart's tooltip into one label per series, rather than per segment. The default value for this attribute is *False*. Once enabled, when hovering over a data point, it displays all the values for the series. This method is recommended over shared tooltips for charts with multiple line series, and as such, takes precedence over `tooltip.shared`.

 When using multiple series, you may be able to reduce the number of data queries and improve performance by using local data to read all of the data once, see *Datalayer Introduction*. Then, link its datalayer to share it to the series, see *Link Datalayers*. At each Series element, link its datalayer to the shared Local Data datalayer and filter the data to meet the needs of each individual series. You can combine different types of Series elements, for example, Series.Bubble and Series.Line, to produce combinations of visualizations.

v23.1 If you are using the Chart Color Axis element in a mult-series chart, by default, the series will link to the first color axis.

To link the series to a special color axis, set the ID attribute of the Chart Color Axis element to the corresponding Series' Linked to Color Axis ID attribute. For more information, see "Chart Color Axis" on page 124.

Series.Bubble - Attributes

The Series.Bubble element has the following attributes:

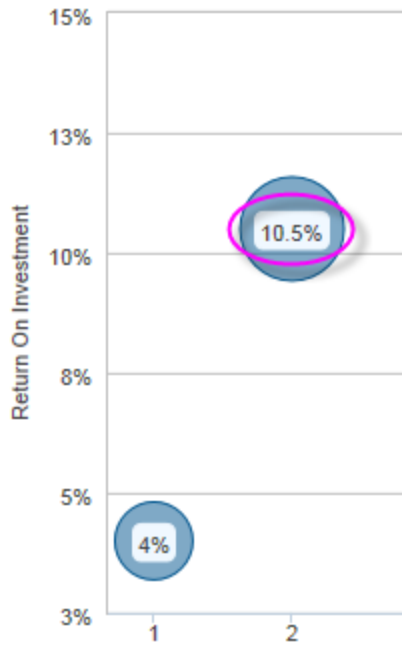
| Attribute | Description |
|--|---|
| Size Data Column | (Required) Specifies the name of a datalayer column that contains values for bubble sizes. |
| Y-Axis Data Column | (Required) Specifies the name of a datalayer column whose values will be plotted along the Y-axis. |
| Bubble Max Size | Sets the maximum bubble size, in pixels or a percentage. Bubbles will be automatically sized between the Bubble Min Size and Bubble Max Size to produce the size of each bubble. Indicate "pixels" by entering just a number, or a "percentage" of the lesser of the plot width and height by entering a number and the percent sign. The default value is 20%. |
| Bubble Min Size | Sets the minimum bubble size, in pixels or a percentage. Bubbles will be automatically sized between the Bubble Min Size and Bubble Max Size to produce the size of each bubble. Indicate "pixels" by entering just a number, or a "percentage" of the lesser of the plot width and height by entering a number and the percent sign. The default value is 20%. |
| v23.1 Class Name | Specifies the Class Name referenced in the .css style sheet. The default value is <i>undefined</i> . |
| Color | Sets the bubble color. Enter a color by name, decimal RGB value, or hex RGB value. Prefix hex values with the pound sign, e.g. #112233. Or, enter the token @Gradient to use a gradient fill to represent the data. |

| Attribute | Description |
|--|---|
| v23.1 Color Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the x-axis. |
| Combine With Series ID | Set this attribute to the element ID of another series to combine it with this series in the legend. When two series are combined, by default only the first one will appear in the legend but clicking the item in the legend toggles both series to appear and disappear. Or, the value <i>Previous</i> can be entered to combine this series with the previous series. |
| Disable Mouse Tracking | Disables mouse tracking for the series, when set to <i>True</i> . This affects tooltips and click events on graphs and points. For large datasets, this may improve performance. The default value is <i>False</i> . |
| Legend Label | Indicates text that will be shown for this series inside the chart legend. When a value is provided, automatically causes the legend to be displayed. |
| v23.1 Linked to Color- Axis ID | Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes |
| Linked to X-Axis ID | Specifies the ID of an X-Axis element that this series should be linked to when using multiple X-axes. |
| Linked to Y-Axis | Specifies the ID of a Y-Axis element that this series should be linked to when using multiple Y-axes. |

| Attribute | Description |
|-------------------------|---|
| ID | |
| Transparency | Specifies the transparency of the bubble color. The lowest value of <i>0</i> specifies that the background is opaque, with no transparency. At the other end of the scale, <i>15</i> specifies a completely transparent background. Use medium-level transparency to allow different chart layers to show through each other. |
| X-Axis Data Column | Specifies the name of a datalayer column whose values will be plotted along the X-axis. |
| X-Axis Data Column Type | Specifies the data type of the datalayer column named in the X-axis Data Column attribute. Options include <i>Auto</i> (the default), <i>Text</i> , <i>Number</i> , and <i>DateTime</i> . By default, X-axis data values that are <i>DateTime</i> type will be automatically distributed evenly across the time series. If you want to disable this behavior, set this attribute value to <i>Text</i> . |

Series.Bubble - Using the Data Labels Element

A "data label" is text shown next to each data point that shows its value. When the **Data Labels** element is used as a child of Series.Bubble, text representing the data values can be configured to appear on the chart, either inside or outside the bubbles:

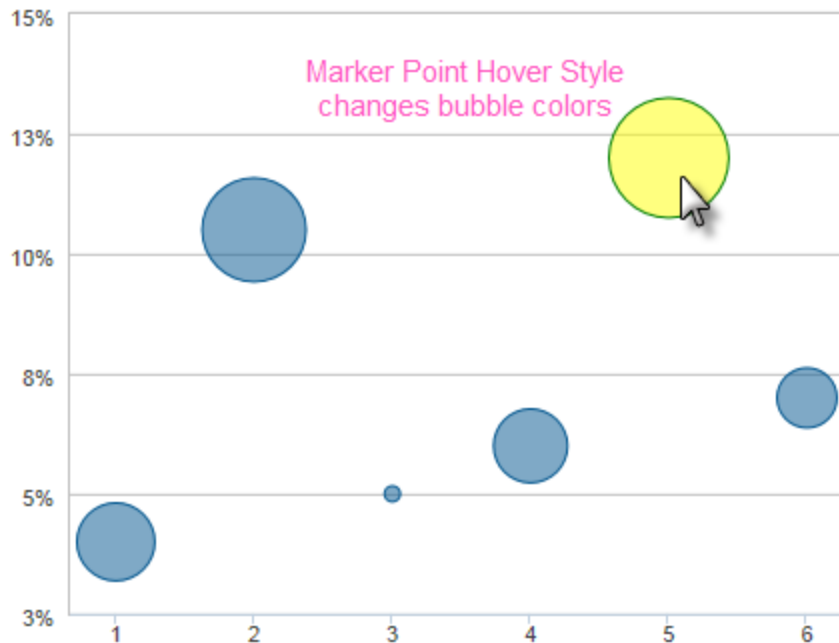


The Data Labels element has attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text. One attribute, Series Name as Caption, allows you to specify the y-axis label, x-axis label, or legend label as the data point value inside the chart.

The Data Labels element's color-related attribute values can be set using @Chart tokens. v23.1 Or, utilize the "Class Name" attribute to apply unique styling to your data labels by referencing a class name from a stylesheet in the report.

Series.Bubble - Using the Marker Points Element

A "marker point" is a symbol that appears on the chart at each data point. These behave differently in a Bubble chart than in other types of charts: the marker point *is* the bubble itself. When the **Marker Points** element is used as a child of Series.Bubble, several properties can be configured.

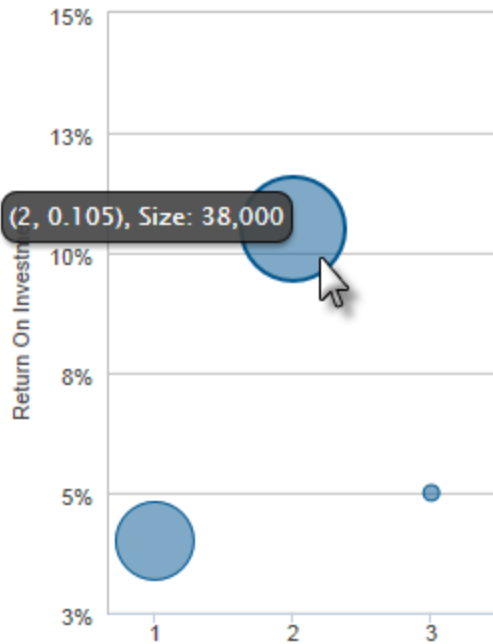


The default marker points are shown above. The Marker Points element allows you to control the bubble color, border color, and transparency. When the cursor hovers over it, a bubble's color can change - the **Marker Points Hover Style** child element lets you configure that behavior and other properties.

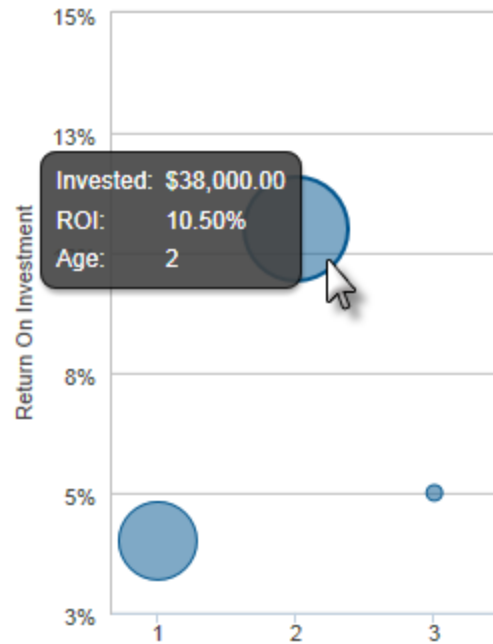
v23.1 You can also create your own marker point symbol using the Chart Canvas child element, Chart Custom Symbol. Once defined, link your custom symbol using the Marker Points element's Symbol attribute. For more information, see "Chart Custom Symbol" on page 128.

Series.Bubble - Using the Quicktips Element

By default, a "quicktip" is displayed when the mouse hovers over a bubble:




Default quicktip



With Quicktip child element

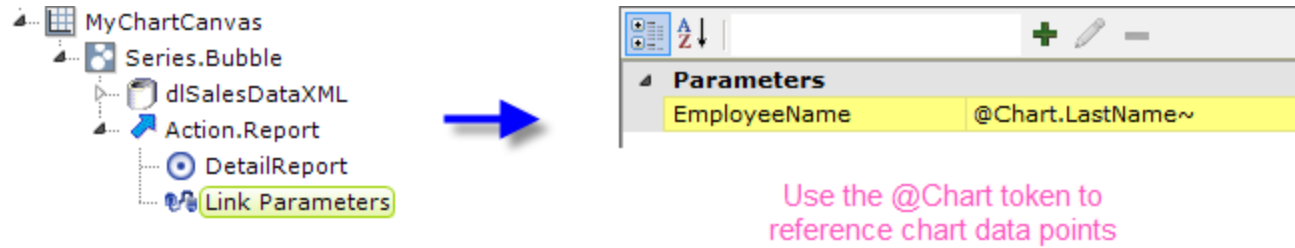
The automatically-generated quicktip displays information from the X- and Y-axes and the bubble size, as shown above, left. However, you may want to display other information or format it differently, perhaps as shown above, right, which can be done by adding a **Quicktip** child element beneath Series.Bubble and setting its attributes and child elements. Use @Chart tokens to include chart data in the quicktip. Intrinsic functions are supported in the Quicktip attributes.

You can also configure the tooltip to display values for any available column in the datalayer. To do so, enter a token representing the data column in the Value attribute of the Quicktip Row element.

 To use this feature with `DataLayer.ActiveSQL`, please make sure the `keep Grouped Rows` attribute of the `SqlGroup` element is set to *False*.

Series.Bubble - Using Action Elements

Action elements initiate processing of a report or process task definition, redirection to a link, or other processing when a data point in the series is clicked.



In the example above, an **Action.Report** element has been added as a child of the Series, along with its **Target.Report** and **Link Parameters** child elements. To reference chart data in parameters, use the @Chart token, as shown above. A variety of Action elements are available for use with Series, including Action.Link, Action.Process, and Action.Refresh Element. Additional Action elements will be added in future releases.

v23.1

Series.Bubble - Using the Series Annotation Element

Annotations allow you to annotate a chart freely using custom labels and shapes. When the **Series Annotation** element is used as a child of Series.Bubble, you can place these annotations at various points of interest:

Exports:-

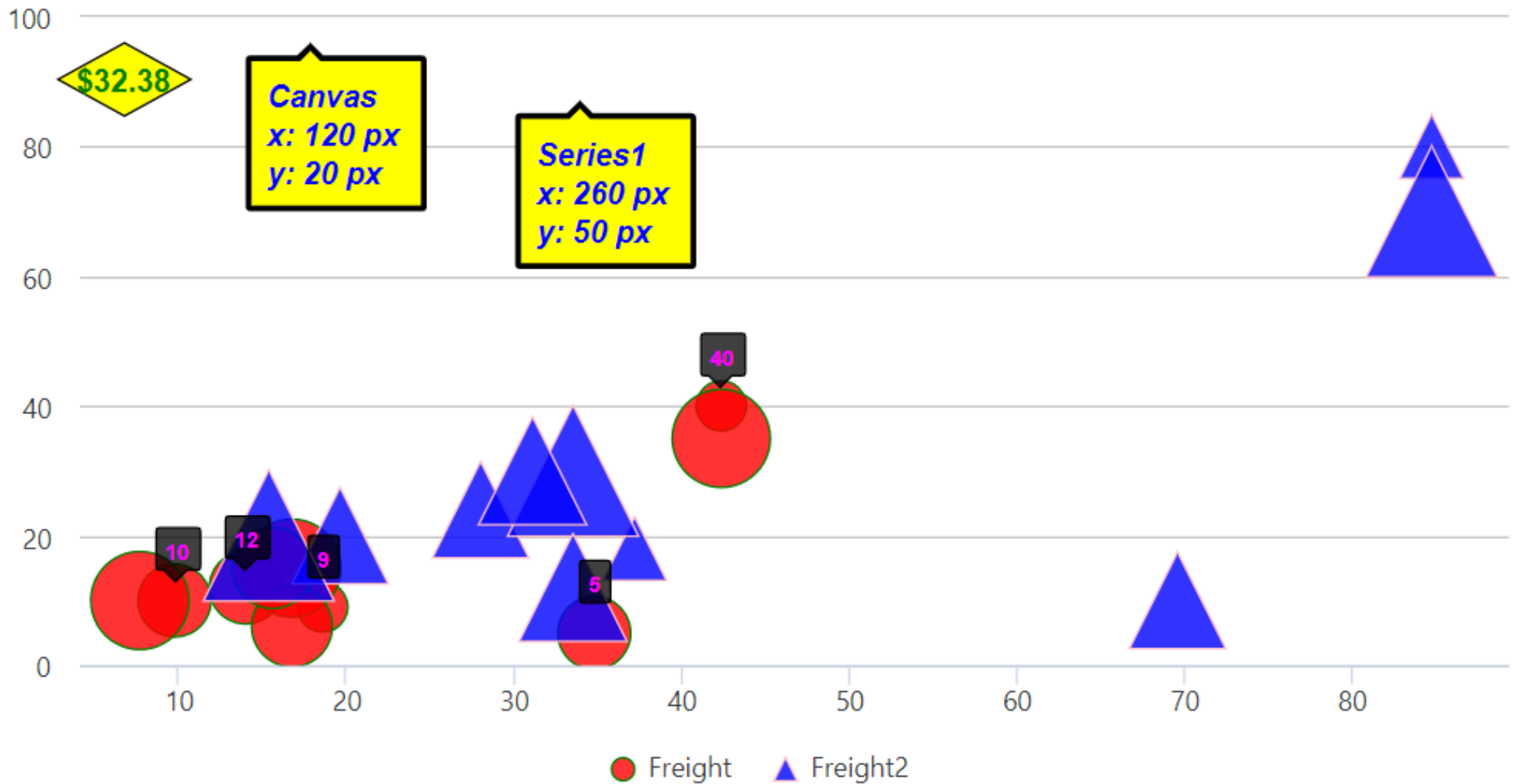
ExportToPdf

ExportToExcel

ExportToWord

Bubble Chart

Charts



The Series Annotation element offers two child elements: **AnnotationLabel.Point** and **AnnotationLabel.Mock**.

The `AnnotationLabel.Point` child element is used to create an annotation label and position on the chart by linking it to an existing point. With this element, you cannot specify the value, it always refers to the value of the point. Additionally, you can use the `Caption` and `Condition` attributes to refer to different datalayers.

On the other hand, the `AnnotationLabel.Mock` is used to create an annotation label and position it on the chart by linking it to a created mock point. With this element, you can specify the coordinate, relative position, and axis.

Both the `AnnotationLabel.Point` and `AnnotationLabel.Mock` have attributes that allow you to control the font family, color, size, and weight, the data format, border color, and positioning of the text.

Series.Bubble - Using Input Selection

This series can also be used with the **Input Selection** family of elements, which turn the chart into an input control. This allows users, at runtime, to select points or values on the chart with their mouse and your report can then take action using the selected data. This is very useful, for example, for drilling into the data. For more information about this functionality, see [Input Selection](#).

Series.Bubble - Using the Refresh Series Timer

The **Refresh Series Timer**, added as a Series child element, updates charts automatically, based on a time interval. When the interval is reached, a request is sent back to the web server for updated data. Series are updated with a smooth animation.



The **Refresh Interval** attribute specifies the number of seconds to wait before refreshing the series data. A value is required and must be an integer greater than 0.

Refresh Mode

Series may be refreshed in two ways: either all of the data is refreshed with each request, or the series data automatically slides to the left one data interval per refresh. The refresh mode is selected in the Chart X Axis element's **Axis Type** attribute. When the Axis Type is *not* set to *DateTimeLinear*, all of the data will be refreshed.

When the Axis Type value is *DateTimeLinear*, newer values are added to the right side of the chart, previous values slide left, and older values disappear as they reach the left edge of the chart. In this case, the Refresh Series Timer element's **Time Span** attribute is used to specify the age of the data included. This value must be in *hh:mm:ss* format (for example "00:02:00" for two minutes) and must be larger than the Refresh Interval attribute value.

When a Time Span attribute is set, the series' datalayer can make use of a special timestamp token, `@Chart.rdTimeSpanStart~`, in a query to retrieve just the rows necessary to update the chart. On the initial request, this token returns the current time minus

the Time Span value. For subsequent refreshes, when *Axis Type = DateTimeLinear* and the X-axis will be sliding, the token returns the last time the chart was updated, so only the newest rows are retrieved. Here's an MS SQL Server query example:

```
SELECT * FROM MyTable WHERE UpdateTime BETWEEN '@Chart.rdTimeSpanStart~' AND '@Function.DateTime~'
```

For best performance, avoid setting a very short refresh interval if a very large number of users will be displaying the report.

Glossary

A

API

API, short for Application Program Interface, is a set of routines, protocols, and tools for building software applications. In business intelligence, APIs may be used to enable end-users to directly update source systems.

Authentication

Authentication is the verification of a user's identity.

Authorization

After a user's identity has been authenticated, authorization grants or denies access to reports, columns, and records to selected users or user-groups.

B

Big Data

Refers to both the ever-growing volumes of data in use today and also to services that are specifically engineered to provide and manipulate very large data volumes.

Business Analytics

Business analytics, or business intelligence (BI), gives customers the ability to rapidly create scalable, interactive data analysis applications, and self-service capabilities users can access from anywhere and on any device.

C

Columnar Data Store

Columnar data store is a type of big data repository containing structured data in columns and rows. The main benefits are that the data can be highly compressed and is easily searchable.

CRM

A Customer Relationship Management (CRM) system is a database-based system that records a company's daily customer-related transactions. CRMs can help customer representatives to provide better service, close more deals, and increase revenue.

CSS

Cascading Style Sheets (CSS) is a technology that allows the presentation aspects of web pages to be separated from the page content. It can be used to add "styling" (e.g. apply fonts, colors, alignment, spacing, and more) to web pages.

D

Data Discovery

Data discovery is the capability to analyze data on-the-fly and uncover insights from it.

Data Enrichment

Data enrichment is a method of preparing data to make it ready for analysis and exploitation, and can include formatting, adding calculations, joining with other data, and more.

DevNet

The Logi Developer Network website.

Drill Down

Drill Down is a capability that allows the user to get a view of the underlying or supporting data used in an analysis.

Drill Through

Drill Through is similar to Drill Down but takes it one step further by applying analysis to the underlying or supporting data.

E

Elemental Development

A development approach used in Logi Info that lets developers build feature-rich applications by using reusable, pre-built elements, rather than by writing low-level code.

F

Forecasting

A technique involving data mining and analysis leading to predictions about what will happen in the future.

G

Geo Mapping

The combination of geographic and other data to produce map visualizations, such as Google or Leaflet maps.

H

Heatmap

A Heatmap chart, sometimes called a "tree map", which uses a unique arrangement of rectangles to represent data and relationships, using color and size.

I

Interpolation

The process of evaluating a literal value match containing one or more placeholders, yielding a result in which the placeholders are replaced with their corresponding values.

J

JavaScript

JavaScript is a programming language supported by the majority of modern web browsers and used by many websites.

JDBC

Java Database Connectivity (JDBC) is an API used to access relational databases. Open Database Connectivity (ODBC) is a similar API designed for use with Java.

JSON

JavaScript Object Notation (JSON) is a lightweight data-interchange format that's easy for humans to read and write, and easy for computers to parse and generate.

K

KPI

Key Performance Indicators (KPIs) are visual indicators, in the form of color-coded shapes, which are tied to a pre-defined, critical threshold.

L

LDAP

The Lightweight Directory Access Protocol (LDAP) is an Internet protocol applications use to look up information from a server and is frequently used for containing user login information.

M

My Term

My definition

N

NoSQL

"Not only SQL" (NoSQL) is an alternative to traditional relational databases, and doesn't rely on tables and a pre-determined schema. NoSQL databases are especially useful for working with large sets of distributed data.

O

ODBC

Open Database Connectivity (ODBC) is an API used to access relational databases. Java Database Connectivity (JDBC) is a similar API designed for use with Java.

OLAP

Online Analytical Processing (OLAP) is the process of analyzing data stored in multi-dimensional "cubes".

R

REST

Representational State Transfer (REST) is a type of API used to provide interoperability between computer systems on the Internet.

S

SSM

The Self-Service Module (SSM) is a package that includes Logi Info + SSRM + Discovery or Logi Platform Services.

SSRM

The Self-Service Reporting Module (SSRM) is a Logi Info add-on module that adds special elements to Info and includes the InfoGo application.

W

Write-Back

The ability to update data sources, typically by adding, editing, or deleting data.