Overview  563
Using Multidimensional Plots  564
About Multidimensional Plots  564
Creating a Multidimensional Plot  566
Specifying Data for a Multidimensional Plot  566
Specifying the Type of Multidimensional Plot  567
Specifying Variables and a Viewing Mode for Multidimensional Plots  567
Controlling Markers in Scatter Plots  568
Using Color in a Multidimensional Plot  569
   Color Ranges in Multidimensional Scatter Plots  569
Attributes  570
Dictionary  570
Methods  598
Events  630

Overview

The scatter control lets you create two types of plots: a scatter plot and a surface plot. In a scatter plot, each observation is displayed as a geometric shape plotted on an axis in a two- or three-dimensional view. In a surface plot, the observations are connected by lines and displayed as a three-dimensional surface. Once created, the scatter object is fully interactive: you can zoom the scatter plot in and out; rotate or tilt it; display, hide, move, or size its legend; manipulate color; and more.

Note: To use the Scatter control, you must have SAS/GRAPH software licensed at your site. △

Ancestors:

   sashelp.fsp.object.class
   sashelp.fsp.widget.class

Class:

   sashelp.classes.Scatter_c.class

Attributes “Attributes” on page 570
Methods “Methods” on page 598
Events “Events” on page 630
Using Multidimensional Plots (the Scatter Control) “Using Multidimensional Plots” on page 564
Using Multidimensional Plots

About Multidimensional Plots

The Scatter object supports visualization of two to six variables at a time using a two- or three-dimensional view. The object can display either of two types of plots: a scatter plot or a surface plot. In a scatter plot, each observation is displayed as a geometric shape plotted on an axis in a two- or three-dimensional view. In a surface plot, the observations are connected by lines and displayed as a three-dimensional surface. The simplest scatter plot displays one independent X or Z variable and one dependent Y variable in a two-dimensional view.
You can add a second independent variable and display the plot in three full dimensions.

To display a surface plot, you must specify X, Y, and Z variables and display the plot in three full dimensions.
Creating a Multidimensional Plot

To create a multidimensional plot, open a frame and select a Scatter Control from the Components window. This instantiates the object for you. You can name the object by opening the Property sheet and assigning a name on the name attribute.

You can then perform the following tasks in any order:
- Specify the data that contains the variables you want to analyze. “Specifying Data for a Multidimensional Plot” on page 566
- Specify the type of scatter plot. “Specifying the Type of Multidimensional Plot” on page 567
- Specify the variables and viewing mode for the analysis. “Specifying Variables and a Viewing Mode for Multidimensional Plots” on page 567

Specifying Data for a Multidimensional Plot

To specify data for a multidimensional plot, use the dataSet or dataSetID attributes, or reference the SCL list that contains the data. Use the scatter object’s dataSource attribute to indicate the source of the data.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataSet</td>
<td>Specify the data set in the form libref.filename. You can use this attribute in the property sheet, or in SCL code.</td>
</tr>
<tr>
<td>dataSetID</td>
<td>Specify the data set ID, which is the identifier returned by the open() method. The dataSetID attribute is useful when you want to create a subset of the data “Using Subsets of Data for Multidimensional Graphs” on page 232 before using it. This attribute is only available in SCL code.</td>
</tr>
</tbody>
</table>

For example, the following code sets the dataSet attribute to Sashelp.Revhub:
Scatter Chart Control △ Specifying Variables and a Viewing Mode for Multidimensional Plots

scatter1.dataSource='dataset';
scatter1.dataset = 'sashelp.revhub';

In the following example, the data set is first subset for those revenues that are over $500,000:

dsid=open('sashelp.revhub');
rc=where(dsid,'revenue gt 500000');
scatter1.dataSource='datasetid';
scatter1.dataSetID=dsid;

Specifying the Type of Multidimensional Plot

The scatter object can generate a scatter plot and a surface plot. To specify the plot type, set the chartType attribute to either Scatter or Surface. For example, for a scatter object named scatter1, the following code sets the chartType attribute to Scatter:

scatter1.chartType = 'scatter';

Specifying Variables and a Viewing Mode for Multidimensional Plots

Multidimensional plots let you specify two categories of variables:

- axes variables. At least two axes variables are required: one independent (x or z) and one dependent (y).
- shape, size, and color variables, which affect the plot markers (see “Controlling Markers in Scatter Plots” on page 568).

The object supports long variable names, and mixed-case variable names.

To specify the axes variables, use the attributes XVariable and/or ZVariable to specify independent variable(s), and YVariable to specify the dependent variable. The axes-variable specification must be coordinated with the visual dimensions, which you set with the object’s viewMode attribute. The viewing mode affects which actions are available for interacting with the object.

The following table defines the viewing modes and shows how to coordinate their use with the analysis variables:

<table>
<thead>
<tr>
<th>viewMode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>creates a two-dimensional view that has height and width but no depth. Requires one independent variable (X or Z) and one dependent variable (Y). This mode is not available for surface plots.</td>
</tr>
<tr>
<td>3D</td>
<td>creates a view that has height and width and uses shading to create the illusion of depth. Requires only one independent variable (X or Z), but allows both. Also requires one dependent variable. This mode is not available for surface plots.</td>
</tr>
<tr>
<td>3DPerspective</td>
<td>creates a true 3D view that has depth. Requires two independent variables and one dependent variable. If you set the viewMode to 3DPerspective with only one independent variable, the object uses a viewMode of 3D.</td>
</tr>
</tbody>
</table>
For a scatter object named scatter1, the following code sets X and Y variables for a two-dimensional analysis:

```javascript
scatter1.viewMode='2d';
scatter1.xvariable = 'country';
scatter1.yvariable = 'sales';
```

## Controlling Markers in Scatter Plots

Scatter plots display data points with two- or three-dimensional markers. You can control the marker shape, size, and color with the markerShape, markerSize, and markerColor attributes. For example, for a scatter object named scatter1, the following code sets the marker shape to a triangle:

```javascript
scatter1.markerShape='triangle';
```

In addition to using the shape, size, and color attributes to control the markers, you can specify a categorical variable whose values control these marker characteristics. To do so, use the attributes shapeVariable, sizeVariable, and colorVariable. For example, the following code specifies variable Month from the data as the variable to control marker shape, so that each month value will be represented by a different marker shape:

```javascript
scatter1.shapeVariable='month';
```

The attributes that affect the plot markers are used or ignored, depending on whether variables are associated with the marker shape, size, or color. For example, the markerShape attribute determines the shape of plot markers if there is no variable associated with marker shapes. However, if the shapeVariable attribute is used to associate a variable with the marker shapes, then the markerShape attribute is ignored, and the attributes that affect marker shapes are shapeVariable, markerShapeOrder, shapeVariableType, and shapeVariableStatistic.

The following table shows the attributes that control marker shape, size, and color.

<table>
<thead>
<tr>
<th>Controlling attribute if no variable associated with marker</th>
<th>Controlling attributes if variable associated with marker</th>
</tr>
</thead>
<tbody>
<tr>
<td>markerShape</td>
<td>shapeVariable, markerShapeOrder, shapeVariableType, and shapeVariableStatistic</td>
</tr>
<tr>
<td>markerSize</td>
<td>sizeVariable, markerSizeMode, minimumMarkerSize, maximumMarkerSize, sizeVariableType, and sizeVariableStatistic</td>
</tr>
<tr>
<td>markerColor</td>
<td>colorVariable, colorRange, colorRangeObject, colorVariableType, and colorVariableStatistic</td>
</tr>
</tbody>
</table>

For more information on controlling marker colors, see “Color Ranges in Multidimensional Scatter Plots” on page 569.
Using Color in a Multidimensional Plot

The following table lists the attributes you can use to control color in a multidimensional plot. For information on controlling the color in titles and footnotes, see “Displaying Titles and Footnotes in Multidimensional Graphs” on page 233.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Specifies</th>
</tr>
</thead>
<tbody>
<tr>
<td>axisColor</td>
<td>Color for the axes, axes labels, tick marks, and legend</td>
</tr>
<tr>
<td>backgroundColor</td>
<td>Color outside of the plot axes but within the object’s border</td>
</tr>
<tr>
<td>borderColor</td>
<td>Color for the border around the plot object</td>
</tr>
<tr>
<td>borderTitleColor</td>
<td>Text color for the border’s title</td>
</tr>
<tr>
<td>chartBackgroundColor</td>
<td>Color inside the area defined by the plot axes</td>
</tr>
<tr>
<td>gridColor</td>
<td>Color of the plot’s grid lines</td>
</tr>
<tr>
<td>highlightColor</td>
<td>Color a marker turns when it is selected</td>
</tr>
<tr>
<td>labelVariableColor</td>
<td>Color for label variable text</td>
</tr>
<tr>
<td>markerColor</td>
<td>Color for the plot markers</td>
</tr>
<tr>
<td>missingColor</td>
<td>Color used to represent missing values in the plot</td>
</tr>
<tr>
<td>outlineColor</td>
<td>Color for the outline around plot markers</td>
</tr>
</tbody>
</table>

For all the above attributes except the colorVariable, you can specify a color name like red, green, or blue. Alternatively, you can set the color to match a color value that is set in the SAS environment. For example, you can set the color to match the color used for the SAS notes that print in the Log window.

To set colors in the SAS environment, use the SASCOLORS window, which you can open with the Tools menu:

```
Tools ▶ Options ▶ Colors
```

For a histogram object named histogram1, the following code sets the backgroundColor attribute to blue:

```
histogram1.backgroundColor = 'blue';
```

Color Ranges in Multidimensional Scatter Plots

By default, the marker colors in a multidimensional scatter plot range from red for the highest values to yellow for the lowest. The colors can represent the Sum, Mean, or Frequency statistic for the dependent variables. To specify the statistic represented by the color range, use the colorVariableStatistic attribute.

To change the color values, you can specify alternative colors on one of the attributes colorList or colorRangeObject. Use the colorSource attribute to indicate the color source.
You can also specify a numeric variable whose values are indexed into the defined color range to determine the specific color used for each marker. To assign the color variable, use the colorVariable attribute.

For a scatter object named scatter1, the following code sets the colorVariable to Sales and the colorVariableStatistic to Sum:

```javascript
scatter1.colorVariable = 'sales';
scatter1.colorVariableStatistic = 'sum';
```

### Attributes

Attributes specified for the Scatter Control class are described here. See sashelp.fsp.Widget.class"Attributes" on page 135 for inherited attributes.

**Note:** The following attributes are inherited from Widget class and appear in the Properties window in build mode, but they are not used by the Scatter control: attachedInterface, CBTFrameName, contentsUpdatedAttributes, help, helpText, model, and toolTipText.

**Public Attributes**

### Dictionary

**XAxisLabelText**

Returns or sets the label text for the X axis

*Type: Character*

*Initial Value: (Object)*

*Valid Values:*

*Category: Appearance*

### XAxisValueLength

Returns or sets tick size for the X axis

*Type: Numeric*

*Initial Value: (Object)*

*Valid Values:*

*Category: Appearance*
**XDataMax**

Returns or sets the maximum value for the X-axis

Type: Numeric
Initial Value: (Object)
Valid Values: 
Category: Data

**XDataMin**

Returns or sets the minimum value for the X-axis

Type: Numeric
Initial Value: (Object)
Valid Values: 
Category: Data

**XGridSize**

Sets the data grid rows for Surface Plot

Type: Numeric
Initial Value: (Object)
Valid Values: 
Category: Appearance

**XScale**

Returns or sets the X scaling value
**XVariable**

Returns or sets the variable for the X axis

Type: Character
Initial Value: (Object)
Valid Values: \sashelp.classes.chartvalues.scl
Category: Data

---

**XVariableType**

Returns or sets the variable type for the X axis

Type: Character
Initial Value: (Object)
Valid Values: Numerical Categorical
Category: Data

---

**YAxisLabelText**

Returns or sets the label text for the Y axis

Type: Character
Initial Value: (Object)
Valid Values:
Category: Appearance
YAxisMode

Returns or sets the Y axis mode
Type: Character
Initial Value: (Object)
Valid Values: Relative Absolute
Category: Appearance

YAxisValueSpacing

Returns or sets the tick size for the Y axis
Type: Numeric
Initial Value: (Object)
Valid Values:
Category: Appearance

YDataMax

Returns or sets the maximum value for the Y-axis
Type: Numeric
Initial Value: (Object)
Valid Values:
Category: Data

YDataMin

Returns or sets the minimum value for the Y-axis
YScale

Returns or sets the Y scaling value

Type: Numeric
Initial Value: (Object)
Valid Values:
Category: Appearance

YVariable

Returns or sets the variable for the Y axis

Type: Character
Initial Value: (Object)
Valid Values: \sashelp.classes.chartvalues.sd
Category: Data

YVariableStatistic

Returns or sets the Y variable statistic

Type: Character
Initial Value: (Object)
Valid Values: Sum,Mean,Freq
Category: Data
**YVariableType**

Returns or sets the variable type for the Y axis

Type: Character
Initial Value: (Object)
Valid Values: Numerical Categorical
Category: Data

**ZAxisLabelText**

Returns or sets the label text for the Z axis

Type: Character
Initial Value: (Object)
Valid Values: 
Category: Appearance

**ZDataMax**

Returns or sets the maximum value for the Z-axis

Type: Numeric
Initial Value: (Object)
Valid Values: 
Category: Data

**ZDataMin**

Returns or sets the minimum value for the Z-axis

Type: Numeric  
Initial Value: (Object)  
Valid Values:  
Category: Data  

---

**ZScale**

Returns or sets the Z scaling value  
Type: Numeric  
Initial Value: (Object)  
Valid Values:  
Category: Appearance  

---

**ZVariable**

Returns or sets the variable for the Z axis  
Type: Character  
Initial Value: (Object)  
Valid Values: \sashelp.classes.chartvalues.sd  
Category: Data  

---

**ZVariableType**

Returns or sets the variable type for the Z axis  
Type: Character  
Initial Value: (Object)  
Valid Values: Numerical Categorical  
Category: Data
actionMode

Returns or sets the action mode

Type: Character

Initial Value: (Object)

Valid Values: Pick Probe Move Rotate Tilt Spin Zoom Legend ScrollLegend DataMove DataZoom Viewport Refline

Category: Appearance

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pick</td>
<td>Select a bar and add associated data to the pickList attribute. Selection is made by clicking the bar. Selection can be for a single bar, extended selection, or disabled, depending on setting for selectionMode attribute.</td>
</tr>
<tr>
<td>Probe</td>
<td>Display data associated with a bar. The data display is controlled by clicking and holding down the mouse button. As the mouse moves over different bars, the display changes.</td>
</tr>
<tr>
<td>Move</td>
<td>Move the chart within its borders. Movement is controlled by clicking and dragging the mouse.</td>
</tr>
<tr>
<td>Rotate</td>
<td>Rotate the chart around the vertical axis. Movement is controlled by horizontal mouse movement during a click and drag.</td>
</tr>
<tr>
<td>Tilt</td>
<td>Tilt the chart around the horizontal axis. Movement is controlled by vertical mouse movement during a click and drag.</td>
</tr>
<tr>
<td>Spin</td>
<td>Change both the tilt and the rotation angles. Movement is controlled by clicking and dragging the mouse.</td>
</tr>
<tr>
<td>Zoom</td>
<td>Zooms object in or out. Movement is controlled by clicking and dragging the mouse.</td>
</tr>
<tr>
<td>Legend</td>
<td>Moves legend, or changes legend size. To move, click in the legend center and drag. To size, click just inside border you want to size, and then drag.</td>
</tr>
<tr>
<td>ScrollLegend</td>
<td>Scrolls legend values if they can't all be displayed at once. Movement is controlled by vertical mouse movement during a click and drag.</td>
</tr>
<tr>
<td>DataMove</td>
<td>Scrolls the data within the axes. Only works for ViewMode=2D or 3D. For 3D, scrolls only in X-Axis. Also, only works after zooming the data using DataZoom or ViewPort. Movement is controlled by clicking and dragging the mouse.</td>
</tr>
<tr>
<td>DataZoom</td>
<td>Zooms into the data to see a subset of the bars more clearly. After a DataZoom, you can use DataMove. Movement is controlled by clicking and dragging the mouse.</td>
</tr>
</tbody>
</table>
Viewport
Zooms into part of the data by dragging a rectangle around it. After a Viewport, you can use DataMove. Movement is controlled by clicking and dragging the mouse.

Refline
Move a reference line. Movement is controlled by clicking and dragging the mouse.

animateMode

Returns or sets the animation mode

Type: Character
Initial Value: (Object)
Valid Values: Filled Wireframe
Category: Appearance

autoScale

Returns or sets the auto scale state, which determines whether chart size is automatically scaled to fit within its borders.

Type: Character
Initial Value: (Object)
Valid Values: On Off
Category: Appearance

axisColor

Returns or sets the tickmark, axis, label, and legend color

Type: Character
Initial Value: (Object)
Valid Values:
Category: Appearance
axisLabelFont

Returns or sets the axis label font

Type: List
Initial Value: (Object)
Valid Values:
Category: Appearance

chartBackgroundColor

Returns or sets the chart background color

Type: Character
Initial Value: (Object)
Valid Values:
Category: Appearance

chartStyle

Returns or sets the chart style

Type: Character
Initial Value: (Object)
Valid Values: Filled Wireframe
Category: Appearance

chartType

Returns or sets the chart type
**colorList**

Returns or sets the color range for data mapping

Type: List
Initial Value: (Object)
Valid Values:
Category: Appearance

---

**colorRangeObject**

Returns or sets the color range object

Type: Character
Initial Value: (Object)
Valid Values:
Category: Appearance

The Scatter control does not support the color attributes (Blinking, Highlight, Reverse, Underline, HiRev, and None) that are available on a colorRangeObject.

---

**colorSource**

Returns or sets the color source

Type: Character
Initial Value: (Object)
Valid Values: ColorList,ColorRangeObject
Category: Appearance
**colorVariable**

Returns or sets the color variable

Type: Character  
Initial Value: (Object)  
Valid Values: \sashelp.classes.chartvalues.scl  
Category: Data

**colorVariableStatistic**

Returns or sets the color variable statistic

Type: Character  
Initial Value: (Object)  
Valid Values: Sum,Mean,Freq  
Category: Data

**colorVariableType**

Returns or sets the color variable type

Type: Character  
Initial Value: (Object)  
Valid Values: Numerical Categorical  
Category: Data

**dataSet**

Returns or sets the dataset to be used
**dataSetID**

Returns or sets the ID value for the dataset to be charted

Type: Numeric
Initial Value: (Object)
Valid Values:
Category: Data

**dataSource**

Returns or sets the data source

Type: Character
Initial Value: (Object)
Valid Values: Dataset, dataSetID
Category: Data

**defaultAttribute**

Returns or sets the attribute name used for the object's default attribute

Type: Character
Initial Value: (Object)
Valid Values: \\sashelp\.classes\.defaultattributevalues\.scl
Category: Data
**dragInfo**

Returns or sets the information that is transferred from the control when a drag operation occurs

Type: List
Initial Value: (Object)
Valid Values:
Category: Drag & Drop

**dropInfo**

Returns or sets the information that defines the data representations that can be dropped on the control

Type: List
Initial Value: (Object)
Valid Values:
Category: Drag & Drop

**footnote1**

Returns or sets the footnote1 attributes

Type: SASHELP.CLASSES.CTEXTLABEL_C.CLASS
Initial Value: (Object)
Valid Values:
Category: Appearance

**footnote2**

Returns or sets the footnote2 attributes
gridColor

Returns or sets the color of the grid that is formed if tick marks are extended through the plot area

Type: Character
Initial Value: (Object)
Valid Values:
Category: Appearance

highlightColor

Returns or sets the selection color used when bars are selected with actionMode set to Pick

Type: Character
Initial Value: (Object)
Valid Values:
Category: Appearance

labelVariable

Returns or sets the label variable

Type: Character
Initial Value: (Object)
Valid Values: \sashelp.classes.chartvalues.scl
Category: Data
labelVariableColor

Returns or sets the color for the label variable

Type: Character
Initial Value: (Object)
Valid Values:
Category: Appearance

labelVariableFont

Returns or sets the font for the label variable

Type: List
Initial Value: (Object)
Valid Values:
Category: Appearance

legendFont

Returns or sets the legend font

Type: List
Initial Value: (Object)
Valid Values:
Category: Appearance

legendType

Returns or sets the legend type
**markerColor**

Returns or sets the marker color if color variable not assigned

Type: Character
Initial Value: (Object)
Valid Values: Bar List
Category: Appearance

**markerDrawingMode**

Returns or sets the marker drawing mode

Type: Character
Initial Value: (Object)
Valid Values: Actual Absolute
Category: Appearance

**markerShape**

Returns or sets the marker shape if shape variable not assigned

Type: Character
Initial Value: (Object)
Valid Values: Square, Triangle, Circle, Plus, X, Minus, Diamond, Hexagon
Category: Appearance
**markerShapeOrder**

Returns or sets the marker shape order

- **Type:** List
- **Initial Value:** (Object)
- **Valid Values:**
- **Category:** Appearance

**markerSize**

Returns or sets the marker size if size variable not assigned

- **Type:** Numeric
- **Initial Value:** (Object)
- **Valid Values:**
- **Category:** Appearance

**markerSizeMode**

Returns or sets the marker size mode

- **Type:** Character
- **Initial Value:** (Object)
- **Valid Values:** Relative Absolute
- **Category:** Appearance

**maximumMarkerSize**

Returns or sets the maximum marker size
minimumMarkerSize

Returns or sets the minimum marker size

Type: Numeric
Initial Value: (Object)
Valid Values:
Category: Appearance

missingColor

Returns or sets the color used to represent missing values

Type: Character
Initial Value: (Object)
Valid Values:
Category: Appearance

outlineColor

Returns or sets the marker outline color

Type: Character
Initial Value: (Object)
Valid Values:
Category: Appearance
### pickHighlightMode

Returns or sets the pick highlight mode, which determines whether the outline or the entire bar is colored upon selection

Type: Character
Initial Value: (Object)
Valid Values: Outline Filled
Category: Appearance

### pickList

Returns or sets the pick list, which stores data associated with a selected bar with actionMode set to Pick

Type: List
Initial Value: (Object)
Valid Values:
Category: Data

### projectionAngle

Returns or sets projection angle, which is the angle of the projected surface when viewMode is set to 3D

Type: Numeric
Initial Value: (Object)
Valid Values:
Category: Appearance

Valid values for the projectionAngle are between 30 and 150 degrees. If you set a value outside of this range, the closest value within the range is used instead.
**referenceLineRestriction**

Returns or sets the reference line mode

Type: Character
Initial Value: (Object)
Valid Values: No, Yes
Category: Data

**responseStatistics**

Returns or sets the response statistics

Type: Character
Initial Value: (Object)
Valid Values: Yes, No
Category: Data

**rotationAngle**

Returns or sets the angle to rotate around the vertical axis when viewMode is set to 3DPerspective. If viewMode is not 3DPerspective, rotation isn't possible and this attribute is ignored.

Type: Numeric
Initial Value: (Object)
Valid Values:
Category: Appearance

Valid values for the rotationAngle are between -180 and +180 degrees. If you set a value outside of this range, the closest value within the range is used instead.

**scale**

Returns or sets the scale value for changing the object's size
**shapeVariableStatistic**

Returns or sets the shape variable statistic

Type: Character
Initial Value: (Object)
Valid Values: Sum, Mean, Freq
Category: Data

---

**shapeVariable**

Returns or sets the shape variable

Type: Character
Initial Value: (Object)
Valid Values: \sashelp.classes.chartvalues.sd
Category: Data

---

**selectionMode**

Returns or sets the selection mode state, which determines whether multiple selections are allowed when actionMode is set to Pick

Type: Character
Initial Value: (Object)
Valid Values: None/Single/Extended
Category: Appearance

---

**Shape Chart Control**

Type: Numeric
Initial Value: (Object)
Valid Values: None
Category: Appearance
shapeVariableType

Returns or sets the shape variable type

Type: Character
Initial Value: (Object)
Valid Values: Numerical Categorical
Category: Data

showGrid

Returns or sets the auto grid state

Type: Character
Initial Value: (Object)
Valid Values: Yes No
Category: Appearance

showLabelOutline

Returns or sets the label outline state

Type: Character
Initial Value: (Object)
Valid Values: Yes No
Category: Appearance

showLegend

Returns or sets the legend display state
**sizeVariable**

Returns or sets the size variable

Type: Character

Initial Value: (Object)

Valid Values: \sashelp.classes.chartvalues.scl

Category: Data
sizeVariableStatistic

Returns or sets the size variable statistic
Type: Character
Initial Value: (Object)
Valid Values: Sum, Mean, Freq
Category: Data

sizeVariableType

Returns or sets the size variable type
Type: Character
Initial Value: (Object)
Valid Values: Numerical, Categorical
Category: Data

sortMode

Returns or sets the sort type
Type: Character
Initial Value: (Object)
Valid Values: None, Formatted Ascending, Unformatted Ascending
Category: Appearance

surfaceComplexity

Returns or sets the surface complexity
Scatter Chart Control

### tickmarkFont

Returns or sets the tick mark font

- **Type:** List
- **Initial Value:** (Object)
- **Valid Values:**
- **Category:** Appearance

### tiltAngle

Returns or sets the angle to tilt around the horizontal axis when viewMode is set to 3D Perspective. If viewMode is not 3D Perspective, tilting isn’t possible and this attribute is ignored.

- **Type:** Numeric
- **Initial Value:** (Object)
- **Valid Values:**
- **Category:** Appearance

Valid values for the tiltAngle are between -180 and 0 degrees. If you set a value outside of this range, the closest value within the range is used instead.

### title1

Returns or sets the title1 attributes

- **Type:** SASHELP.CLASSES.CTEXTLABEL_C.CLASS
- **Initial Value:** (Object)
- **Valid Values:**
- **Category:** Appearance
**title2**

Returns or sets the title2 attributes

Type: SASHELP.CLASSES.CTEXTLABEL_C.CLASS

Initial Value: (Object)

Valid Values:

Category: Appearance

**title3**

Returns or sets the title3 attributes

Type: SASHELP.CLASSES.CTEXTLABEL_C.CLASS

Initial Value: (Object)

Valid Values:

Category: Appearance

**title4**

Returns or sets the title4 attributes

Type: SASHELP.CLASSES.CTEXTLABEL_C.CLASS

Initial Value: (Object)

Valid Values:

Category: Appearance
viewMode

Returns or sets the view mode

Type: Character
Initial Value: (Object)
Valid Values: 2D 3D 3D Perspective
Category: Appearance

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>Height and width</td>
</tr>
<tr>
<td>3D</td>
<td>Height, width, and the illusion of depth</td>
</tr>
<tr>
<td>3D Perspective</td>
<td>Height, width, and true depth</td>
</tr>
</tbody>
</table>

viewpointDistance

Returns or sets the distance of the eye point from the viewpoint

Type: Numeric
Initial Value: (Object)
Valid Values:
Category: Appearance

viewpointX

Returns or sets the X coordinate viewpoint location

Type: Numeric
Initial Value: (Object)
Valid Values:
Category: Appearance
**viewpointY**

Returns or sets the Y coordinate viewpoint location

Type: Numeric  
Initial Value: (Object)  
Valid Values:  
Category: Appearance

**viewpointZ**

Returns or sets the X coordinate viewpoint location

Type: Numeric  
Initial Value: (Object)  
Valid Values:  
Category: Appearance

**viewportStackDepth**

Returns the number of active viewports on the viewport stack

Type: Numeric  
Initial Value: 0  
Valid Values:  
Category: Appearance

The viewportStackDepth attribute is read-only at runtime; you cannot set it, and it is not used in build mode. At runtime, viewportStackDepth is automatically incremented when the actionMode attribute is set to Viewport and you zoom in on part of the data by dragging a rectangle around it. Each time you drag a rectangle, viewportStackDepth is incremented by 1.

To return to the previous viewport, use the _popPreviousViewport method.

**Methods**

Methods specified for the Scatter Control class are described here. See sashelp.fsp.Widget.class"Methods” on page 145 for inherited methods.
Public Methods

## _addReferenceLine

Create a reference line

### Syntax

```
objectName_addReferenceLine( axis, refId, vtype, value, label );
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>axis</td>
<td>Character</td>
<td>I</td>
<td>refine axis. The axis variable must be numeric.</td>
</tr>
<tr>
<td>refId</td>
<td>Numeric</td>
<td>O</td>
<td>unique ID returned for the reference line. Used to query or delete the line</td>
</tr>
<tr>
<td>vtype</td>
<td>Character</td>
<td>I</td>
<td>indicates whether the value parameter specifies a data value or a location on the axis. Use keyword Value for data values, or keyword Location for a location given as a percentage of the distance along the axis.</td>
</tr>
<tr>
<td>Argument</td>
<td>Type</td>
<td>Use</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>-----</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>value</td>
<td>Numeric</td>
<td>I</td>
<td>specifies the reference line's initial position on the axis. If the vtype parameter is set to Value, use a data value to position the line; for example, specify 5 to position the line at axis value 5. If the vtype parameter is set to Location, use a percentage to position the line; for example, specify .5 to position the line in the middle of the axis. Location values are expected to range from 0 to 1.</td>
</tr>
<tr>
<td>label</td>
<td>Character</td>
<td>I</td>
<td>label to display for the reference line to identify it for the user</td>
</tr>
</tbody>
</table>

**_binit***

Initializes an object at design time

**Syntax**

```c
objectName_binit();
```

**_deleteReferenceLine***

Delete a reference line

**Syntax**

```c
objectName_deleteReferenceLine( refid );
```
### _getPickList_

**Return the picklist**

**Syntax**

```javascript
objectName_getPickList( picklist );
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pickList</td>
<td>List</td>
<td>I</td>
<td>list of values stored when plot element was selected with actionMode set to Pick.</td>
</tr>
</tbody>
</table>

### _getPickedItemCount_

**Return the number of items in the picklist**

**Syntax**

```javascript
objectName_getPickedItemCount( pickitems );
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pickitems</td>
<td>Numeric</td>
<td>O</td>
<td>number of values stored when plot element was selected with actionMode set to Pick.</td>
</tr>
</tbody>
</table>
得到了一个表，它列出了各种对象的名称，每个对象的名称后跟一个点号，然后是一个函数名称，再跟一个括号，括号中包含两个参数。表的列包括：Argument（参数），Type（类型），Use（使用），和Description（描述）。

表中包含两个参数的函数，分别是refId和value。refId用于指定参考线的ID，而value则用于指定当前参考线的值。

根据表中的信息，参考线值函数的语法如下：
```
ObjectName_getReferenceLineValue( refId, value );
```

函数的参数如下：
- **refId**: 数字类型的参数，用于指定参考线的ID。
- **value**: 数字类型的参数，用于指定当前参考线的值。

下面是一个函数的示例，用于初始化一个对象：
```
ObjectName_init( );
```

**_init**

函数的名称是ObjectName_init，它没有参数。

下面是一个函数的示例，用于将当前视图设置为之前的视图：
```
ObjectName_popPreviousViewport( );
```

**_popPreviousViewport**

函数的名称是ObjectName_popPreviousViewport，它没有参数。
_print

print

Syntax

objectName_print();

_resetView

Set the view back to the default setting

Syntax

objectName_resetView();

_setReferenceLineLabel

Sets the label associated with the specified refline

Syntax

objectName_setReferenceLineLabel(refid, label);
### _setReferenceLineValue_

**Set a reference line value**

**Syntax**

```plaintext
objectName_setReferenceLineValue( refId, value);
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>refId</td>
<td>Numeric</td>
<td>I</td>
<td>refId of the reference line whose label will be set</td>
</tr>
<tr>
<td>label</td>
<td>Character</td>
<td>I</td>
<td>the label to assign to the reference line</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>refId</td>
<td>Numeric</td>
<td>I</td>
<td>refId of the reference line whose value will be set</td>
</tr>
<tr>
<td>value</td>
<td>Character</td>
<td>I</td>
<td>the data value to use for setting the reference line's position. The value cannot be a location percentage. To set a location, delete the reference line and add a new one with _addReferenceLine.</td>
</tr>
</tbody>
</table>

### _setcamShowLegendValue_

**Syntax**

```plaintext
objectName_setcamShowLegendValue( arg1);
```
<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return</td>
<td>Numeric</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>arg1</td>
<td>Character</td>
<td>U</td>
<td>accepts Yes or No</td>
</tr>
</tbody>
</table>

**_showLegendValue**

Show the legend value

---

**Syntax**

```
objectName_showLegendValue();
```

---

**_setcamActionMode**

sets the actionMode attribute

---

**Syntax**

```
objectName_setcamActionMode( mode );
```

---

**_setcamAnimateMode**

sets the animateMode attribute

---

**Syntax**

```
objectName_setcamAnimateMode( mode );
```
_setcamAutoScale

sets the autoScale attribute

Syntax

objectName_setcamAutoScale( scale );

_setcamAxisColor

sets the axis color

Syntax

objectName_setcamAxisColor( color );

_setcamAxisLabelFont

sets the axis label font

Syntax

objectName_setcamAxisLabelFont( font );

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>font</td>
<td>List</td>
<td>I</td>
<td>font</td>
</tr>
</tbody>
</table>

_setcamChartBackgroundColor

sets the chartBackgroundColor attribute
Syntax
objectName._setcamChartBackgroundColor( color );

_setcamChartStyle

sets the chartStyle attribute

Syntax
objectName._setcamChartStyle( style );

_setcamChartType

sets the chartType attribute

Syntax
objectName._setcamChartType( type );

_setcamColorList

sets the colorList attribute

Syntax
objectName._setcamColorList( colorList );
## _setcamColorRangeObject

sets the colorRangeObject attribute

**Syntax**

```
objectName_setcamColorRangeObject( object );
```

## _setcamColorSource

sets the colorSource attribute

**Syntax**

```
objectName_setcamColorSource( source );
```

## _setcamColorVariable

sets the color variable

**Syntax**

```
objectName_setcamColorVariable( varName );
```
_setcamColorVariableStatistic

sets the color variable's statistic: Mean, Sum, or Freq

Syntax

ObjectName_setcamColorVariableStatistic( stat );

_setcamColorVariableType

sets the color variable's type

Syntax

ObjectName_setcamColorVariableType( type );

_setcamDataSetID

sets the dataSetID attribute

Syntax

ObjectName_setcamDataSetID( id );

_setcamDataSetName

sets the dataSetName attribute
Syntax

objectName_setcamDataSource(dsname);

_setcamDataSource

sets the dataSource attribute

Syntax

objectName_setcamDataSource(source);

_setcamGridColor

sets the gridColor attribute

Syntax

objectName_setcamGridColor(color);

_setcamHighlightColor

sets the highlightColor attribute

Syntax

objectName_setcamHighlightColor(color);
_setcamLabelColor

Set the LabelVariable color

Syntax

objectName_setcamLabelColor( color );

_argument

_setcamLabelFont

Set the LabelVariable font

Syntax

objectName_setcamLabelFont( font );

Argument

<table>
<thead>
<tr>
<th>font</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>font</td>
<td>List</td>
<td>I</td>
<td>font</td>
</tr>
</tbody>
</table>

_setcamLabelVariable

sets the label variable

Syntax

objectName_setcamLabelVariable( varName );
_setcamLegendFont

sets the legend font

Syntax

objectName_setcamLegendFont( font );

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>font</td>
<td>List</td>
<td>I</td>
<td>font</td>
</tr>
</tbody>
</table>

_setcamLegendType

sets the legend type: Bar or List

Syntax

objectName_setcamLegendType( type );

_setcamMarkerColor

sets the markerColor attribute

Syntax

objectName_setcamMarkerColor( color );

_setcamMarkerDrawingMode

sets the markerDrawingMode attribute: Actual or Absolute
**Syntax**

objectName_setcamMarkerDrawingMode( mode );

---

**setcamMarkerShape**

sets the markerShape attribute

---

**Syntax**

objectName_setcamMarkerShape( shape );

---

**setcamMarkerShapeOrder**

Set the marker shape order

---

**Syntax**

objectName_setcamMarkerShapeOrder( order );

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>order</td>
<td>List</td>
<td>I</td>
<td>order in which to use shapes</td>
</tr>
</tbody>
</table>

---

**setcamMarkerSize**

sets the markerSize attribute
_setcamMarkerSizeMode

sets the markerSizeMode attribute: Relative or Absolute

Syntax

objectName_setcamMarkerSizeMode( mode );

_setcamMaximumMarkerSize

sets the maximumMarkerSize attribute

Syntax

objectName_setcamMaximumMarkerSize( max );

_setcamMinimumMarkerSize

sets the minimumMarkerSize attribute

Syntax

objectName_setcamMinimumMarkerSize( min );
**_setcamMissingColor**

sets the missingColor attribute

Syntax

```
objectName_setcamMissingColor( color );
```

**_setcamOutlineColor**

sets the outlineColor attribute

Syntax

```
objectName_setcamOutlineColor( color );
```

**_setcamPickHighlightMode**

sets the pickHighlightMode attribute

Syntax

```
objectName_setcamPickHighlightMode( mode );
```

**_setcamPickList**

sets the pickList attribute
**Syntax**

```
objectName_setcamPickList(values);
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>values</td>
<td>List</td>
<td>I</td>
<td>list of values stored in pickList</td>
</tr>
</tbody>
</table>

**_setcamProjectionAngle_**

sets the `projectionAngle` attribute

**Syntax**

```
objectName_setcamProjectionAngle(angle);
```

**_setcamReferenceLineRestriction_**

sets the `referenceLineRestriction` attribute: accepts Yes or No

**Syntax**

```
objectName_setcamReferenceLineRestriction(arg1);
```

**_setcamResponseStatistics_**

sets the response statistic: Mean, Sum, or Freq
**_setcamResponseStatistics**

Syntax

```
objectName._setcamResponseStatistics(stat);
```

**_setcamRotationAngle**

sets the rotation angle

Syntax

```
objectName._setcamRotationAngle(angle);
```

**_setcamScale**

sets the scale attribute

Syntax

```
objectName._setcamScale(scale);
```

**_setcamSelectionMode**

sets the selectionMode attribute

Syntax

```
objectName._setcamSelectionMode(mode);
```
_setcamShapeVariable

sets the shape variable

Syntax

objectName._setcamShapeVariable(varName);

_setcamShapeVariableStatistic

sets the shape variable statistic: Mean, Sum, or Freq

Syntax

objectName._setcamShapeVariableStatistic(stat);

_setcamShapeVariableType

sets the shape variable's type

Syntax

objectName._setcamShapeVariableType(type);

_setcamShowGrid

sets the showGrid attribute: accepts Yes or No
Syntax
objectName_setcamShowGrid( arg1 );

_setcamShowLegend
sets the showLegend attribute: accepts Yes or No

Syntax
objectName_setcamShowLegend( arg1 );

_setcamShowMarkerCenterPoint
sets the showMarkerCenterPoint attribute: accepts Yes or No

Syntax
objectName_setcamShowMarkerCenterPoint( arg1 );

_setcamSizeVariable
sets the size variable

Syntax
objectName_setcamSizeVariable( varName );
_setcamSizeVariableStatistic

sets the size variable’s statistic: Mean, Sum, or Freq

-------

Syntax

objectName_setcamSizeVariableStatistic(stat);

_setcamSizeVariableType

sets the size variable’s type

-------

Syntax

objectName_setcamSizeVariableType(type);

_setcamSortMode

sets the sortMode attribute

-------

Syntax

objectName_setcamSortMode(mode);

_setcamSurfaceComplexity

sets the surfaceComplexity attribute
Syntax

objectName_setcamSurfaceComplexity( arg1 );

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>arg1</td>
<td>Numeric</td>
<td>I</td>
<td>complexity level</td>
</tr>
</tbody>
</table>

_setcamTickmarkFont

sets the tick mark font

Syntax

objectName_setcamTickmarkFont( font );

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>font</td>
<td>List</td>
<td>I</td>
<td>font</td>
</tr>
</tbody>
</table>

_setcamTiltAngle

sets the tilt angle

Syntax

objectName_setcamTiltAngle( angle );

_setcamViewMode

sets the viewMode attribute
Syntax

objectName_setcamViewMode( mode );

_setcamViewpointDistance

sets the viewpointDistance attribute

Syntax

objectName_setcamViewpointDistance( distance );

_setcamViewpointX

sets the viewpointX coordinate

Syntax

objectName_setcamViewpointX( coordinate );

_setcamViewpointY

sets the viewpointY coordinate

Syntax

objectName_setcamViewpointY( coordinate );
_setcamViewpointZ

description

Syntax

objectName_setcamViewpointZ( arg1 );

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>arg1</td>
<td>Numeric</td>
<td>I</td>
<td></td>
</tr>
</tbody>
</table>

_setcamXAxisLabelText

sets the XAxisLabelText attribute

Syntax

objectName_setcamXAxisLabelText( text );

_setcamXAxisValueLength

sets the XAxisValueLength attribute

Syntax

objectName_setcamXAxisValueLength( length );
_setcamXDataMax

sets the XDataMax attribute

-------------------

Syntax

objectName_setcamXDataMax( max );

-------------------

_setcamXDataMin

sets the XDataMin attribute

-------------------

Syntax

objectName_setcamXDataMin( min );

-------------------

_setcamXGridSize

sets the XGridSize attribute

-------------------

Syntax

objectName_setcamXGridSize( size );

-------------------

_setcamXScale

sets the XScale attribute
**Syntax**

```
ObjectName_setcamXScale( scale );
```

---

**_setcamXVariable**

sets the x variable

---

**Syntax**

```
ObjectName_setcamXVariable( varName );
```

---

**_setcamXVariableType**

sets the x variable's type

---

**Syntax**

```
ObjectName_setcamXVariableType( type );
```

---

**_setcamYAxisLabelText**

sets the YAxisLabelText attribute

---

**Syntax**

```
ObjectName_setcamYAxisLabelText( text );
```
**_setcamYAxisMode**

sets the YAxisMode attribute

Syntax

```
objectName_setcamYAxisMode( mode );
```

**_setcamYAxisValueSpacing**

sets the YAxisValueSpacing attribute

Syntax

```
objectName_setcamYAxisValueSpacing( spacing );
```

**_setcamYDataMax**

sets the YDataMax attribute

Syntax

```
objectName_setcamYDataMax( max );
```

**_setcamYDataMin**

sets the YDataMin attribute
**Syntax**

`objectName_setcamYDataMin( min );`

---

### _setcamYScale

sets the YScale attribute

---

**Syntax**

`objectName_setcamYScale( scale );`

---

### _setcamYVariable

sets the YVariable attribute

---

**Syntax**

`objectName_setcamYVariable( varName );`

---

### _setcamYVariableStatistic

sets the Y variable’s statistic: Mean, Sum, or Freq

---

**Syntax**

`objectName_setcamYVariableStatistic( stat );`
_setcamYVariableType

sets the YVariableType attribute

-------------

Syntax

objectName_setcamYVariableType( type );

__________________________

_setcamZAxisLabelText

sets the z axis label text

-------------

Syntax

objectName_setcamZAxisLabelText( text );

__________________________

_setcamZDataMax

sets the ZDataMax attribute

-------------

Syntax

objectName_setcamZDataMax( max );

__________________________

_setcamZDataMin

sets the ZDataMin attribute
__setcamZDataMin__

Syntax

```plaintext
objectName_setcamZDataMin( min );
```

__setcamZScale__

sets the ZScale attribute

Syntax

```plaintext
objectName_setcamZScale( scale );
```

__setcamZVariable__

sets the z variable

Syntax

```plaintext
objectName_setcamZVariable( varName );
```

__setcamZVariableType__

sets the z variable's type

Syntax

```plaintext
objectName_setcamZVariableType( type );
```
_showLabelOutline

Sets display of label outline: accepts Yes or No

Syntax

objectName._showLabelOutline(outline);

Events

Events specified for the Scatter Control class are described here.

System Events

<table>
<thead>
<tr>
<th>Event</th>
<th>Enabled</th>
<th>Description</th>
<th>Handler Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>XAxisLabelText changed</td>
<td>Yes</td>
<td>Occurs when XAxisLabelText attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>XAxisValueLength changed</td>
<td>Yes</td>
<td>Occurs when XAxisValueLength attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>XDataMax changed</td>
<td>Yes</td>
<td>Occurs when XDataMax attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>XDataMin changed</td>
<td>Yes</td>
<td>Occurs when XDataMin attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>XGridSize changed</td>
<td>Yes</td>
<td>Occurs when XGridSize attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>XScale changed</td>
<td>Yes</td>
<td>Occurs when XScale attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>XVariable changed</td>
<td>Yes</td>
<td>Occurs when XVariable attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>Event Description</td>
<td>Occurs when</td>
<td>Yes/No</td>
<td>Untrapped</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>-------------</td>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td>XVariableType changed</td>
<td></td>
<td>Yes</td>
<td>Untrapped</td>
</tr>
<tr>
<td>YAxisLabelText changed</td>
<td></td>
<td>Yes</td>
<td>Untrapped</td>
</tr>
<tr>
<td>YAxisMode changed</td>
<td></td>
<td>Yes</td>
<td>Untrapped</td>
</tr>
<tr>
<td>YAxisValueSpacing changed</td>
<td></td>
<td>Yes</td>
<td>Untrapped</td>
</tr>
<tr>
<td>YDataMax changed</td>
<td></td>
<td>Yes</td>
<td>Untrapped</td>
</tr>
<tr>
<td>YDataMin changed</td>
<td></td>
<td>Yes</td>
<td>Untrapped</td>
</tr>
<tr>
<td>YScale changed</td>
<td></td>
<td>Yes</td>
<td>Untrapped</td>
</tr>
<tr>
<td>YVariable changed</td>
<td></td>
<td>Yes</td>
<td>Untrapped</td>
</tr>
<tr>
<td>YVariableStatistic changed</td>
<td></td>
<td>Yes</td>
<td>Untrapped</td>
</tr>
<tr>
<td>YVariableType changed</td>
<td></td>
<td>Yes</td>
<td>Untrapped</td>
</tr>
<tr>
<td>ZAxisLabelText changed</td>
<td></td>
<td>Yes</td>
<td>Untrapped</td>
</tr>
<tr>
<td>ZDataMax changed</td>
<td></td>
<td>Yes</td>
<td>Untrapped</td>
</tr>
<tr>
<td>ZDataMin changed</td>
<td></td>
<td>Yes</td>
<td>Untrapped</td>
</tr>
<tr>
<td>ZScale changed</td>
<td></td>
<td>Yes</td>
<td>Untrapped</td>
</tr>
<tr>
<td>Event Type</td>
<td>YES</td>
<td>Occurs when</td>
<td>Untrapped</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----</td>
<td>----------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>ZVariable changed</td>
<td>Yes</td>
<td>ZVariable attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>ZVariableType changed</td>
<td>Yes</td>
<td>ZVariableType attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>actionMode changed</td>
<td>Yes</td>
<td>actionMode attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>animateMode changed</td>
<td>Yes</td>
<td>animateMode attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>autoScale changed</td>
<td>Yes</td>
<td>autoScale attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>axisColor changed</td>
<td>Yes</td>
<td>axisColor attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>axisLabelFont changed</td>
<td>Yes</td>
<td>axisLabelFont attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>chartBackgroundColor changed</td>
<td>Yes</td>
<td>chartBackgroundColor attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>chartStyle changed</td>
<td>Yes</td>
<td>chartStyle attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>chartType changed</td>
<td>Yes</td>
<td>chartType attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>colorList changed</td>
<td>Yes</td>
<td>colorList attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>colorRangeObject changed</td>
<td>Yes</td>
<td>colorRangeObject attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>colorSource changed</td>
<td>Yes</td>
<td>colorSource attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>colorVariable changed</td>
<td>Yes</td>
<td>colorVariable attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>Event Name</td>
<td>Occurs When</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>colorVariableStatistic changed</td>
<td>colorVariableStatistic attribute value is changed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>colorVariableType changed</td>
<td>colorVariableType attribute value is changed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dataSet changed</td>
<td>dataSet attribute value is changed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dataSetID changed</td>
<td>dataSetID attribute value is changed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dataSource changed</td>
<td>dataSource attribute value is changed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>footnote1 changed</td>
<td>footnote1 attribute value is changed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>footnote2 changed</td>
<td>footnote2 attribute value is changed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gridColor changed</td>
<td>gridColor attribute value is changed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>highlightColor changed</td>
<td>highlightColor attribute value is changed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>labelVariable changed</td>
<td>labelVariable attribute value is changed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>labelVariableColor changed</td>
<td>labelVariableColor attribute value is changed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>labelVariableFont changed</td>
<td>labelVariableFont attribute value is changed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>legendFont changed</td>
<td>legendFont attribute value is changed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>legendType changed</td>
<td>legendType attribute value is changed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event</td>
<td>Triggered</td>
<td>Occurs when</td>
<td>Untrapped</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------</td>
<td>-------------</td>
<td>-----------</td>
</tr>
<tr>
<td>markerColor changed</td>
<td>Yes</td>
<td>markerColor attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>markerDrawingMode changed</td>
<td>Yes</td>
<td>markerDrawingMode attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>markerShape changed</td>
<td>Yes</td>
<td>markerShape attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>markerShapeOrder changed</td>
<td>Yes</td>
<td>markerShapeOrder attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>markerSize changed</td>
<td>Yes</td>
<td>markerSize attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>markerSizeMode changed</td>
<td>Yes</td>
<td>markerSizeMode attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>maximumMarkerSize changed</td>
<td>Yes</td>
<td>maximumMarkerSize attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>minimumMarkerSize changed</td>
<td>Yes</td>
<td>minimumMarkerSize attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>missingColor changed</td>
<td>Yes</td>
<td>missingColor attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>outlineColor changed</td>
<td>Yes</td>
<td>outlineColor attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>pickHighlightMode changed</td>
<td>Yes</td>
<td>pickHighlightMode attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>pickList changed</td>
<td>Yes</td>
<td>pickList attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>projectionAngle changed</td>
<td>Yes</td>
<td>projectionAngle attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>Attribute Code</td>
<td>Changed</td>
<td>Occurs when</td>
<td>Attribute Value is Changed</td>
</tr>
<tr>
<td>---------------</td>
<td>---------</td>
<td>-------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>referenceLineRestriction</td>
<td>Yes</td>
<td>(\text{referenceLineRestriction})</td>
<td>changed</td>
</tr>
<tr>
<td>responseStatistics</td>
<td>Yes</td>
<td>(\text{responseStatistics})</td>
<td>changed</td>
</tr>
<tr>
<td>rotationAngle</td>
<td>Yes</td>
<td>(\text{rotationAngle})</td>
<td>changed</td>
</tr>
<tr>
<td>scale</td>
<td>Yes</td>
<td>(\text{scale})</td>
<td>changed</td>
</tr>
<tr>
<td>selectionMode</td>
<td>Yes</td>
<td>(\text{selectionMode})</td>
<td>changed</td>
</tr>
<tr>
<td>shapeVariable</td>
<td>Yes</td>
<td>(\text{shapeVariable})</td>
<td>changed</td>
</tr>
<tr>
<td>shapeVariableStatistic</td>
<td>Yes</td>
<td>(\text{shapeVariableStatistic})</td>
<td>changed</td>
</tr>
<tr>
<td>shapeVariableType</td>
<td>Yes</td>
<td>(\text{shapeVariableType})</td>
<td>changed</td>
</tr>
<tr>
<td>showGrid</td>
<td>Yes</td>
<td>(\text{showGrid})</td>
<td>changed</td>
</tr>
<tr>
<td>showLabelOutline</td>
<td>Yes</td>
<td>(\text{showLabelOutline})</td>
<td>changed</td>
</tr>
<tr>
<td>showLegend</td>
<td>Yes</td>
<td>(\text{showLegend})</td>
<td>changed</td>
</tr>
<tr>
<td>showLegendValue</td>
<td>Yes</td>
<td>(\text{showLegendValue})</td>
<td>changed</td>
</tr>
<tr>
<td>showMarkerCenterPoint</td>
<td>Yes</td>
<td>(\text{showMarkerCenterPoint})</td>
<td>changed</td>
</tr>
</tbody>
</table>
sizeVariable changed Yes Occurs when sizeVariable attribute value is changed Untrapped

sizeVariableStatistic changed Yes Occurs when sizeVariableStatistic attribute value is changed Untrapped

sizeVariableType changed Yes Occurs when sizeVariableType attribute value is changed Untrapped

sortMode changed Yes Occurs when sortMode attribute value is changed Untrapped

surfaceComplexity changed Yes Occurs when surfaceComplexity attribute value is changed Untrapped

tickmarkFont changed Yes Occurs when tickmarkFont attribute value is changed Untrapped

tiltAngle changed Yes Occurs when tiltAngle attribute value is changed Untrapped

title1 changed Yes Occurs when title1 attribute value is changed Untrapped

title2 changed Yes Occurs when title2 attribute value is changed Untrapped

title3 changed Yes Occurs when title3 attribute value is changed Untrapped

title4 changed Yes Occurs when title4 attribute value is changed Untrapped

viewMode changed Yes Occurs when viewMode attribute value is changed Untrapped

viewpointDistance changed Yes Occurs when viewpointDistance attribute value is changed Untrapped

viewpointX changed Yes Occurs when viewpointX attribute value is changed Untrapped
<table>
<thead>
<tr>
<th>Event</th>
<th>Occurs when</th>
<th>Untrapped</th>
</tr>
</thead>
<tbody>
<tr>
<td>viewpointY changed</td>
<td>viewpointY attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>viewpointZ changed</td>
<td>viewpointZ attribute value is changed</td>
<td>Untrapped</td>
</tr>
</tbody>
</table>