Overview 377
Using Multidimensional Histograms 378
About Multidimensional Histograms 378
Creating a Multidimensional Histogram 380
Specifying Data for a Multidimensional Histogram 380
Specifying the Histogram Data Dimensions and Viewing Mode 381
Specifying the Variables for a Multidimensional Histogram 382
Using Color in a Multidimensional Histogram 382
  Color Ranges in Multidimensional Histograms 383
Specifying Data Bins and Display Bins in a Multidimensional Histogram 383
Attributes 384
Dictionary 385
Methods 406
Events 430

Overview

The Histogram control creates a multidimensional histogram object, which lets you build a frequency plot of the data using one to three independent variables to create two- or three-dimensional visualizations. The variables can be attached to one of three axes. Histograms are ideal for large volumes of data. The data are divided into a grid of bins, which you define. The response value can be either the count of the number of observations falling in the bin, or the mean or sum of some other response variable. Once created, the object is fully interactive: you can zoom the histogram in and out; rotate or tilt it; display, hide, move, or size its legend; manipulate color; and more.

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

Ancestors:
  sashelp.fsp.object.class
    sashelp.fsp.widget.class

Class:
  sashelp.classes.Histogram_c.class

Attributes “Attributes” on page 384
Methods “Methods” on page 406
Events “Events” on page 430
Using Multidimensional Histograms

About Multidimensional Histograms

With a multidimensional histogram, you can create:

- Two-dimensional histograms with either one or two variables
- Three-dimensional histograms of either two or three variables

The simplest histogram displays a two-dimensional visualization of a single variable. For example, the following histogram shows the frequency for the single variable Age, specified for the X axis.
Adding a second variable to the analysis generates a two-dimensional visualization of two variables. For example, the following histogram shows the frequency for Age and Height, specified for the X and Z axes.

Adding a third variable to the analysis requires a three-dimensional visualization. The following example shows a histogram with variables Age, Weight, and Height, specified for the X, Y, and Z axes.
Creating a Multidimensional Histogram

To create a multidimensional histogram, open a frame and select a Histogram 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 Histogram” on page 380
- Specify the histogram’s data dimensions and viewing mode. “Specifying the Histogram Data Dimensions and Viewing Mode” on page 381
- Specify the variables for the analysis. “Specifying the Variables for a Multidimensional Histogram” on page 382

Specifying Data for a Multidimensional Histogram

To specify the data for a multidimensional histogram, use the dataSet or dataSetID attributes, or reference the SCL list that contains the data. Use the histogram object’s dataSource attribute to indicate the source of the data.
**Attribute** | **Specification**
--- | ---
**dataSet** | Specify the data set in the form libref.filename. You can use this attribute in the property sheet, or in SCL code.

**dataSetID** | 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.

For a histogram object named histogram1, the following code sets the dataSet attribute to Sashelp.Revhub:

```plaintext
histogram1.dataset = 'sashelp.revhub';
```

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

```plaintext
dsid=open('sashelp.revhub');
rc=where(dsid,'revenue gt 500000');
histogram1.dataSource='datasetid';
histogram1.dataSetID=dsid;
```

### Specifying the Histogram Data Dimensions and Viewing Mode

A multidimensional histogram can analyze one, two, or three independent variables. The number of data dimensions must correspond to the number of independent variables in the analysis. To set the data dimension, use the dataDimensions attribute. For information on specifying the variables, see “Specifying the Variables for a Multidimensional Histogram” on page 382.

Depending on the number of data dimensions, you can set the viewing mode to a two- or three-dimensional histogram. To set the viewing mode, use the viewMode attribute. The viewing mode affects which actions are available for interacting with the object.

The following table shows how to coordinate the data dimensions and viewing mode. The table after it describes the viewing modes.

<table>
<thead>
<tr>
<th>To analyze...</th>
<th>Set dataDimensions to...</th>
<th>Set viewMode to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>One variable</td>
<td>1D</td>
<td>2D</td>
</tr>
<tr>
<td>Two variables</td>
<td>2D</td>
<td>2D or 3D</td>
</tr>
<tr>
<td>Three variables</td>
<td>3D</td>
<td>3D or 3DPerspective</td>
</tr>
</tbody>
</table>

<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.</td>
</tr>
<tr>
<td>3D</td>
<td>creates a view that has height and width and uses shading to create the illusion of depth.</td>
</tr>
<tr>
<td>3DPerspective</td>
<td>creates a view that has height, width, and true depth.</td>
</tr>
</tbody>
</table>

For a histogram object named histogram1, the following code sets the data dimensions to 1D and the viewing mode to 2D:
Specifying the Variables for a Multidimensional Histogram

A multidimensional histogram can analyze from one to three independent variables. The number of variables must correspond to the data dimensions. If the number of variables specified exceeds the number of data dimensions, the excess variables are ignored. For information on setting the data dimensions, see “Specifying the Histogram Data Dimensions and Viewing Mode” on page 381.

To specify the variables, use the attributes XVariable, YVariable, and ZVariable. The histogram object supports long variable names, and mixed-case variable names.

<table>
<thead>
<tr>
<th>To analyze</th>
<th>Use attributes</th>
<th>dataDimensions must be</th>
</tr>
</thead>
<tbody>
<tr>
<td>One variable</td>
<td>XVariable</td>
<td>1D</td>
</tr>
<tr>
<td>Two variables</td>
<td>XVariable and ZVariable</td>
<td>2D</td>
</tr>
<tr>
<td>Three variables</td>
<td>XVariable, YVariable, and ZVariable</td>
<td>3D</td>
</tr>
</tbody>
</table>

For a histogram object named histogram1, the following code specifies variable Revenue as the value for the XVariable attribute:

```java
histogram1.XVariable = 'revenue';
```

Using Color in a Multidimensional Histogram

The following table lists the attributes you can use to control color in a multidimensional histogram. 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 axis, axis labels, tick marks, and legend.</td>
</tr>
<tr>
<td>backgroundColor</td>
<td>Color outside of the histogram axes but within the object’s border</td>
</tr>
<tr>
<td>borderColor</td>
<td>Color for the border around the histogram object</td>
</tr>
<tr>
<td>borderTitleColor</td>
<td>Text color for the border’s title</td>
</tr>
<tr>
<td>chartBackgroundColor</td>
<td>Color within the histogram axes</td>
</tr>
<tr>
<td>colorVariable</td>
<td>Bar colors in the histogram</td>
</tr>
<tr>
<td>faceColorMode</td>
<td>Whether bar colors affect the entire bar or only the bar tops</td>
</tr>
<tr>
<td>gridColor</td>
<td>Color of the histogram’s grid lines</td>
</tr>
</tbody>
</table>
### Histogram Control

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Specifies</th>
</tr>
</thead>
<tbody>
<tr>
<td>highlightColor</td>
<td>Color a bar turns when it is selected</td>
</tr>
<tr>
<td>missingColor</td>
<td>Color used to represent missing color in the histogram</td>
</tr>
</tbody>
</table>

For all the attributes except colorVariable, you can specify a SAS 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:

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

### Color Ranges in Multidimensional Histograms

By default, the bar colors in a multidimensional histogram range from red for the highest values to yellow for the lowest. The colors can represent the Sum, Mean, or Frequency statistic for the analysis variables. To specify the statistic to represent with 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 color variable whose values are indexed into the defined color range to determine the specific color used for each bar. To assign the color variable, use the colorVariable attribute. You can use the faceColorMode attribute to determine whether the color applies to the entire bar or just the bar tops.

For a histogram object named histogram1, the following code sets the colorVariable to Revenue, the colorVariableStatistic to Mean, and the faceColorMode to Top:

```javascript
histogram1.colorVariable = 'revenue';
histogram1.colorVariableStatistic = 'mean';
histogram1.faceColorMode = 'top';
```

### Specifying Data Bins and Display Bins in a Multidimensional Histogram

The data in a histogram are usually divided into bins that define the values ranges represented by each bar in the histogram. For example, if the data contain a range of ages from 1 to 80, and the data are divided into 8 bins, age values from 1 to 10 fall in the first bin, values 11 to 20 are in the second bin, and so on.

For each histogram variable, there are two attributes for controlling the bins. One attribute determines the number of bins used to divide the actual data values. The other attribute determines the number of bins currently displayed in the histogram; the number of bins is equal to the number of bars displayed.
### Attributes

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XDataBinExponent,</td>
<td>Determine the number of bins used to divide data values in the X, Y, and Z variables.</td>
</tr>
<tr>
<td>YDataBinExponent,</td>
<td></td>
</tr>
<tr>
<td>ZDataBinExponent</td>
<td></td>
</tr>
</tbody>
</table>

| XDisplayBinExponent,          | Determine the number of bins (bars) displayed in the histogram. The setting for the display bin cannot exceed the setting for the data bin. For example, if the data are divided into 8 bins, you cannot display more than 8 bars in the histogram. |
| YDisplayBinExponent,          |                                                                                                                                               |
| ZDisplayBinExponent           |                                                                                                                                               |

The value you specify for a bin attribute is used as an exponential value of 2. Thus, if you specify 2, there will be 4 bins. If you specify 3, there will be 8 bins, and so on.

The XDataBinExponent, YDataBinExponent, ZDataBinExponent attributes determine how many bins are used to divide data values for the corresponding variables. For example, a setting of XDataBinExponent = 3 divides the data values for the X variable into 8 bins. If the X variable is age, and the age values in the data range from 1 to 80, age values from 1 to 10 fall in the first bin, values 11 to 20 are in the second bin, and so on.

The XDisplayBinExponent, YDisplayBinExponent, ZDisplayBinExponent attributes determine how many bars are displayed in the histogram to represent the full range of data. For example, a setting of XDisplayBinExponent = 2 displays 4 bars in the histogram. If the X variable is age, and the age values in the data range from 1 to 80, age values from 1 to 20 are represented by the first bar, values 21 to 40 are represented by the second, and so on.

The number of data bins can always be evenly distributed among the number of display bins because the number of bins of both types is always a power of 2, and the number of display bins cannot exceed the number of data bins. This means that you can choose how to optimize your application for displaying histograms. The trade off is between performance and memory.

Changing the number of data bins forces a data read because the data values must be read if they are to be divided into the appropriate bins. Changing the number of display bins, however, does not force a data read. Thus, you can improve performance by setting the largest number of data bins you anticipate using for each variable. You might want to do this for improving performance in a client/server application where you have to download the data from the server to the client application. You might be able to download the data only once, for example, when initializing the application.

However, the more data bins you have, the more memory the application needs to keep all the bins in memory. Thus, to optimize the application for memory use rather than performance, you might choose to use the smallest number of data bins you expect to need for most application functions, and set a higher number of data bins only when you actually need the increased number.

---

**Attributes**

Attributes specified for the Histogram 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 Histogram 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 mark size for the X axis
Type: Numeric
Initial Value: (Object)
Valid Values:
Category: Appearance

XDataBinExponent

Returns or sets the X data bin exponent
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

---

**XDisplayBinExponent**

Returns or sets the X display bin exponent

Type: Numeric

Initial Value: (Object)

Valid Values:

Category: Appearance

---

**XVariable**

Returns or sets the variable for the X axis

Type: Character
Initial Value: (Object)
Valid Values: \sashelp.classes.chartvalues.sc
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

**YAxisValueSpacing**

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

Returns or sets the Y data bin exponent

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

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

**YDisplayBinExponent**

Returns or sets the Y display bin exponent

Type: Numeric
Histogram Control

YVariable

Returns or sets the variable for the Y axis
Type: Character
Initial Value: (Object)
Valid Values: \sashelp.classes.chartvalues.scl
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
**ZDataBinExponent**

Returns or sets the Z data bin exponent

Type: Numeric
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

**ZDisplayBinExponent**

Returns or sets the Z display bin exponent

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.scl
Category:  Data

---

**ZVariableType**

Returns or sets the variable 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 Refline
Category:  Appearance
### 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
Histogram Control

**chartBackgroundColor**

Returns or sets the chart background color

- **Type:** Character
- **Initial Value:** (Object)
- **Valid Values:**
- **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
**colorList**

Returns or sets the color range for data mapping

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

**colorRangeObject**

Returns or sets the color range object

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

The Histogram 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: Data

**colorVariable**

Returns or sets the color variable

Type: Character
Histogram Control

**dataSet**

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

---

**dataDimensions**

Returns or sets the data dimensions

Type: Character
Initial Value: (Object)
Valid Values: 1D 2D 3D
Category: Data

---

**dataSet**

Returns or sets the dataset to be used

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

**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
Histogram Control

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

faceColorMode

Returns or sets the face color mode
Type: Character
Initial Value: (Object)
Valid Values: Top All
Category: Appearance

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

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

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

heightVariable

Returns or sets the height variable

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

heightVariableStatistic

Returns or sets the height variable statistic

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

highlightColor

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

Type: Character
Histogram Control

missingColor

Returns or sets the color used to represent missing values
Type: Character
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
Type: Character
Initial Value: (Object)
Valid Values: Bar,List
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.
**Histogram Control**

**selectionMode**

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

**rotationAngle**

Returns or sets the angle to rotate around the vertical axis when viewMode is set to 3D Perspective. If viewMode is not 3D Perspective, 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 size of the histogram

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

**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
showGrid

Shows or hides the grid that is formed if tick marks are extended through the chart area

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

showLegend

Determines whether the legend is displayed

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

showLegendValue

Returns or sets the display of selected data values in the legend

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

Returns or sets the sort type

Type: Character
Initial Value: (Object)
Valid Values: None Ascending
Category: Appearance

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
Return or sets the title2 attributes

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

Return or sets the title3 attributes

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

Return or sets the title4 class 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, 3DPerspective
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>3DPerspective</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 Z coordinate viewpoint location

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

Methods

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

Public Methods

_addReferenceLine

Add reference line value

Syntax

objectName_addReferenceLine( axis, refid, vtypevalue,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>refline axis. The axis variable must be numeric.</td>
</tr>
<tr>
<td>refld</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>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

objectName._binit();

__deleteReferenceLine

Delete reference line value

Syntax

objectName._deleteReferenceLine(refid);

<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 for the reference line to be deleted</td>
</tr>
</tbody>
</table>

__getPickList

Get picklist

Syntax

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 chart element was selected with actionMode set to Pick.</td>
</tr>
</tbody>
</table>
getJSONReferenceLineValue

Return reference line value

Syntax

objectName_getReferenceLineValue( 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 value will be returned</td>
</tr>
<tr>
<td>value</td>
<td>Numeric</td>
<td>O</td>
<td>the data value at the reference line's current position</td>
</tr>
</tbody>
</table>

_init

Initializes an object

Syntax

objectName_init( );

_print

Print

Syntax

objectName_print( );
_resetView

Reset View

Syntax

objectName._resetView();

_setReferenceLineValue

Set reference line value

Syntax

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 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>

_setcamAxisColor

Sets the axisColor attribute
### setcamAxisColor

**Syntax**

```
objectName_setcamAxisColor( color );
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>color</td>
<td>Character</td>
<td>I</td>
<td>color to set</td>
</tr>
</tbody>
</table>

### setcamActionMode

Sets the actionMode attribute

**Syntax**

```
objectName_setcamActionMode( mode );
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Character</td>
<td>I</td>
<td>the action mode to set on the actionMode attribute</td>
</tr>
</tbody>
</table>

### setcamAnimateMode

sets the animationMode attribute

**Syntax**

```
objectName_setcamAnimateMode( mode );
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Character</td>
<td>I</td>
<td>the animation mode to set on the animationMode attribute</td>
</tr>
</tbody>
</table>
**_setcamAutoScale**

sets the autoScale attribute

---

**Syntax**

```
objectName_setcamAutoScale( arg1 );
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>arg1</td>
<td>Character</td>
<td>I</td>
<td>specify Yes or No</td>
</tr>
</tbody>
</table>

**_setcamAxisLabelFont**

sets the font for the axis label

---

**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>SCL list defining the font</td>
</tr>
</tbody>
</table>

**_setcamBinDimensions**

description

---

**Syntax**

```
objectName_setcamBinDimensions( arg1 );
```
**_setcamChartBackgroundColor_**

sets the chartBackgroundColor attribute

---

**Syntax**

```
objectName_setcamChartBackgroundColor( color );
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>color</td>
<td>Character</td>
<td>I</td>
<td>the color to set for the background</td>
</tr>
</tbody>
</table>

**_setcamColorList_**

Invoked when the colorList attribute is changed

---

**Syntax**

```
return = objectName_setcamColorList( arg1 );
```

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

**_setcamColorRangeObject_**

Invoked when the colorRangeObject attribute is changed
_setcamColorSource

Invoked when the colorSource attribute is changed

Syntax

return = objectName_setcamColorSource( arg1 );

_setcamColorVariable

sets the colorVariable

Syntax

objectName_setcamColorVariable( varName );

_setcamColorVariableStatistic

sets the colorVariableStatistic: Sum, Mean, or Freq

Syntax

objectName_setcamColorVariableStatistic( stat );
**_setcamDataSetID**

sets the dataSetID

Syntax

```
objectName_setcamDataSetID( id );
```

**_setcamDataSetName**

sets the dataSet name

Syntax

```
objectName_setcamDataSetName( dsname );
```

**_setcamFaceColorMode**

sets the faceColorMode

Syntax

```
objectName_setcamFaceColorMode( mode );
```

**_setcamGridColor**

sets the grid color


Syntax

objectName_setcamGridColor( color );

_setcamHeightVariable

sets the height variable

Syntax

objectName_setcamHeightVariable( varName );

_setcamHeightVariableStatistic

sets the height variable's statistic

Syntax

objectName_setcamHeightVariableStatistic( stat );

_setcamHighlightColor

sets the height variable's color

Syntax

objectName_setcamHighlightColor( color );
_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>arg1</td>
<td>List</td>
<td>I</td>
<td>font</td>
</tr>
</tbody>
</table>

_setcamLegendType

Invoked when the legendType attribute is changed

Syntax

objectName_setcamLegendType( type );

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>Character</td>
<td>I</td>
<td>specify Bar or List</td>
</tr>
</tbody>
</table>

_setcamMissingColor

Invoked when the missingColor attribute is changed

Syntax

objectName_setcamMissingColor( color );
_setcamPickHighlightMode

sets the pickHighlightMode attribute

Syntax

objectName_setcamPickHighlightMode( mode );

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Character</td>
<td>I</td>
<td>specifies Outline or Filled</td>
</tr>
</tbody>
</table>

_setcamPickList

sets the pickList attribute

Syntax

objectName_setcamPickList( list );

_setcamProjectionAngle

sets the projection angle

Syntax

objectName_setcamProjectionAngle( angle );
**_setcamReferenceLineRestriction_**

sets the reference line restriction

---

**Syntax**

```
objectName_setcamReferenceLineRestriction( arg1 );
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>arg1</td>
<td>Character</td>
<td>I</td>
<td>specifies Yes or No</td>
</tr>
</tbody>
</table>

---

**_setcamRotationAngle_**

sets the rotation angle

---

**Syntax**

```
objectName_setcamRotationAngle( angle );
```

---

**_setcamScale_**

sets the scale

---

**Syntax**

```
objectName_setcamScale( scale );
```
_setcamSelectionMode

sets the selectionMode attribute

Syntax

ObjectName_setcamSelectionMode(arg1);

_setcamShowGrid

shows or hides grid

Syntax

ObjectName_setcamShowGrid(arg1);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>arg1</td>
<td>Character</td>
<td>I</td>
<td>specifies Yes or No</td>
</tr>
</tbody>
</table>

_setcamShowLegend

sets showLegend attribute

Syntax

ObjectName_setcamShowLegend(arg1);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>arg1</td>
<td>Character</td>
<td>I</td>
<td>accepts Yes or No</td>
</tr>
</tbody>
</table>
__setcamShowLegendValue

Invoked when the showLegendValue attribute is changed

Syntax

return = 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>

__setcamSortMode

sets sortMode attribute

Syntax

objectName__setcamSortMode(arg1);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>arg1</td>
<td>Character</td>
<td>I</td>
<td>accepts None or Ascending</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>arg1</td>
<td>List</td>
<td>l</td>
<td>font</td>
</tr>
</tbody>
</table>

### _setcamTiltAngle_

**Sets the tilt angle**

**Syntax**

```plaintext
objectName_setcamTiltAngle( angle );
```

### _setcamViewMode_

**Sets the viewMode attribute**

**Syntax**

```plaintext
objectName_setcamViewMode( mode );
```

### _setcamViewpointDistance_

**Sets the viewpointDistance attribute**

**Syntax**

```plaintext
objectName_setcamViewpointDistance( arg1 );
```
_setcamViewpointX

sets the viewpointX attribute

Syntax

objectName_setcamViewpointX( coord );

<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>x coordinate</td>
</tr>
</tbody>
</table>

_setcamViewpointY

sets the viewpointY attribute

Syntax

objectName_setcamViewpointY( coord );

<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>y coordinate</td>
</tr>
</tbody>
</table>

_setcamViewpointZ

sets the viewpointZ attribute

Syntax

objectName_setcamViewpointZ( coord );
<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>l</td>
<td>z coordinate</td>
</tr>
</tbody>
</table>

### _setcamXAxisLabelText_

sets x-axis label text

**Syntax**

```
objectName_setcamXAxisLabelText( text );
```

### _setcamXAxisValueLength_

sets XAxisValueLength attribute

**Syntax**

```
objectName_setcamXAxisValueLength( length );
```

### _setcamXDataBinExponent_

sets XDataBinExponent attribute

**Syntax**

```
objectName_setcamXDataBinExponent( exponent );
```
_setcamXDataMax

sets XDataMax attribute

Syntax

objectName_setcamXDataMax( max );

_setcamXDataMin

sets XDataMin attribute

Syntax

objectName_setcamXDataMin( min );

_setcamXDisplayBinExponent

sets XDisplayBinExponent attribute

Syntax

objectName_setcamXDisplayBinExponent( exponent );

_setcamXVariable

sets x variable
Chapter 15

Syntax

_objectName._setcamXVariable( varName );

_setcamXVariableType

sets XVariableType attribute

Syntax

_objectName._setcamXVariableType( type );

_setcamYAxisLabelText

sets y-axis label text

Syntax

_objectName._setcamYAxisLabelText( text );

_setcamYAxisValueSpacing

sets YAxisValueSpacing

Syntax

_objectName._setcamYAxisValueSpacing( space );
**_setcamYDataBinExponent**

sets YDataBinExponent attribute

----------

**Syntax**

```
ObjectName_setcamYDataBinExponent( exponent );
```

**_setcamYDataMax**

sets YDataMax attribute

----------

**Syntax**

```
ObjectName_setcamYDataMax( max );
```

**_setcamYDataMin**

sets YDataMin attribute

----------

**Syntax**

```
ObjectName_setcamYDataMin( min );
```

**_setcamYDisplayBinExponent**

sets YDisplayBinExponent attribute
Syntax

_objectName_setcamYDisplayBinExponent( exponent );

__setcamYVariable

sets y variable

Syntax

_objectName_setcamYVariable( varName );

__setcamYVariableType

sets type for y variable

Syntax

_objectName_setcamYVariableType( type );

__setcamZAxisLabelText

sets label text for z axis

Syntax

_objectName_setcamZAxisLabelText( text );
_setcamZDataBinExponent

sets ZDataBinExponent attribute

Syntax

objectName_setcamZDataBinExponent( exponent );

_____________________________________________________

_setcamZDataMax

sets ZDataMax attribute

Syntax

objectName_setcamZDataMax( max );

_____________________________________________________

_setcamZDataMin

sets ZDataMin attribute

Syntax

objectName_setcamZDataMin( min );

_____________________________________________________

_setcamZDisplayBinExponent

sets ZDisplayBinExponent attribute
_setcamZVariable

sets z variable

Syntax

objectName_setcamZDisplayBinExponent( exponent );

_setcamZVariable

sets z variable

Syntax

objectName_setcamZVariable( varName );

_setcamZVariableType

sets type for z variable

Syntax

objectName_setcamZVariableType( type );

Events

Events specified for the Histogram 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>XDataBinExponent changed</td>
<td>Yes</td>
<td>Occurs when XDataBinExponent 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>XDisplayBinExponent changed</td>
<td>Yes</td>
<td>Occurs when XDisplayBinExponent 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>XVariableType changed</td>
<td>Yes</td>
<td>Occurs when XVariableType attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>YAxisLabelText changed</td>
<td>Yes</td>
<td>Occurs when YAxisLabelText attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>YAxisValueSpacing changed</td>
<td>Yes</td>
<td>Occurs when YAxisValueSpacing attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>YDataBinExponent changed</td>
<td>Yes</td>
<td>Occurs when YDataBinExponent attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>YDataMax changed</td>
<td>Yes</td>
<td>Occurs when YDataMax attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>YDataMin changed</td>
<td>Yes</td>
<td>Occurs when YDataMin attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>Event Description</td>
<td>Occurrence</td>
<td>Occurs when</td>
<td>Status</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>------------</td>
<td>------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>YDisplayBinExponent changed</td>
<td>Yes</td>
<td>YDisplayBinExponent attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>YVariable changed</td>
<td>Yes</td>
<td>YVariable attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>YVariableType changed</td>
<td>Yes</td>
<td>YVariableType attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>ZAxisLabelText changed</td>
<td>Yes</td>
<td>ZAxisLabelText attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>ZDataBinExponent changed</td>
<td>Yes</td>
<td>ZDataBinExponent attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>ZDataMax changed</td>
<td>Yes</td>
<td>ZDataMax attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>ZDataMin changed</td>
<td>Yes</td>
<td>ZDataMin attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>ZDisplayBinExponent changed</td>
<td>Yes</td>
<td>ZDisplayBinExponent attribute value is changed</td>
<td>Untrapped</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>Event Description</td>
<td>Triggered</td>
<td>Occurrence</td>
<td>Event Details</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----------</td>
<td>------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>axisColor changed</td>
<td>Yes</td>
<td>Occurs</td>
<td>when axisColor attribute value is changed</td>
</tr>
<tr>
<td>axisLabelFont changed</td>
<td>Yes</td>
<td>Occurs</td>
<td>when axisLabelFont attribute value is changed</td>
</tr>
<tr>
<td>chartBackgroundColor changed</td>
<td>Yes</td>
<td>Occurs</td>
<td>when chartBackgroundColor attribute value is changed</td>
</tr>
<tr>
<td>colorList changed</td>
<td>Yes</td>
<td>Occurs</td>
<td>when colorList attribute value is changed</td>
</tr>
<tr>
<td>colorRangeObject changed</td>
<td>Yes</td>
<td>Occurs</td>
<td>when colorRangeObject attribute value is changed</td>
</tr>
<tr>
<td>colorSource changed</td>
<td>Yes</td>
<td>Occurs</td>
<td>when colorSource attribute value is changed</td>
</tr>
<tr>
<td>colorVariable changed</td>
<td>Yes</td>
<td>Occurs</td>
<td>when colorVariable attribute value is changed</td>
</tr>
<tr>
<td>colorVariableStatistic changed</td>
<td>Yes</td>
<td>Occurs</td>
<td>when colorVariableStatistic attribute value is changed</td>
</tr>
<tr>
<td>dataDimensions changed</td>
<td>Yes</td>
<td>Occurs</td>
<td>when dataDimensions attribute value is changed</td>
</tr>
<tr>
<td>dataSet changed</td>
<td>Yes</td>
<td>Occurs</td>
<td>when dataSet attribute value is changed</td>
</tr>
<tr>
<td>dataSetID changed</td>
<td>Yes</td>
<td>Occurs</td>
<td>when dataSetID attribute value is changed</td>
</tr>
<tr>
<td>dataSource changed</td>
<td>Yes</td>
<td>Occurs</td>
<td>when dataSource attribute value is changed</td>
</tr>
<tr>
<td>faceColorMode changed</td>
<td>Yes</td>
<td>Occurs</td>
<td>when faceColorMode attribute value is changed</td>
</tr>
<tr>
<td>footnote1 changed</td>
<td>Yes</td>
<td>Occurs</td>
<td>when footnote1 attribute value is changed</td>
</tr>
<tr>
<td>Event Description</td>
<td>Indicator</td>
<td>Occurs when</td>
<td>Attribute Value Changed</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----------</td>
<td>-------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>footnote2 changed</td>
<td>Yes</td>
<td>footnote2</td>
<td>Yes</td>
</tr>
<tr>
<td>gridColor changed</td>
<td>Yes</td>
<td>Untrapped</td>
<td>Yes</td>
</tr>
<tr>
<td>heightVariable changed</td>
<td>Yes</td>
<td>Untrapped</td>
<td>Yes</td>
</tr>
<tr>
<td>heightVariableStatistic changed</td>
<td>Yes</td>
<td>Untrapped</td>
<td>Yes</td>
</tr>
<tr>
<td>highlightColor changed</td>
<td>Yes</td>
<td>Untrapped</td>
<td>Yes</td>
</tr>
<tr>
<td>legendFont changed</td>
<td>Yes</td>
<td>Untrapped</td>
<td>Yes</td>
</tr>
<tr>
<td>legendType changed</td>
<td>Yes</td>
<td>Untrapped</td>
<td>Yes</td>
</tr>
<tr>
<td>missingColor changed</td>
<td>Yes</td>
<td>Untrapped</td>
<td>Yes</td>
</tr>
<tr>
<td>pickHighlightMode changed</td>
<td>Yes</td>
<td>Untrapped</td>
<td>Yes</td>
</tr>
<tr>
<td>pickList changed</td>
<td>Yes</td>
<td>Untrapped</td>
<td>Yes</td>
</tr>
<tr>
<td>projectionAngle changed</td>
<td>Yes</td>
<td>Untrapped</td>
<td>Yes</td>
</tr>
<tr>
<td>referenceLineRestriction changed</td>
<td>Yes</td>
<td>Untrapped</td>
<td>Yes</td>
</tr>
<tr>
<td>rotationAngle changed</td>
<td>Yes</td>
<td>Untrapped</td>
<td>Yes</td>
</tr>
<tr>
<td>Event</td>
<td>Action</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>scale changed</td>
<td>Yes</td>
<td>Occurs when scale attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>selectionMode changed</td>
<td>Yes</td>
<td>Occurs when selectionMode attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>showGrid changed</td>
<td>Yes</td>
<td>Occurs when showGrid attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>showLegend changed</td>
<td>Yes</td>
<td>Occurs when showLegend attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>showLegendValue changed</td>
<td>Yes</td>
<td>Occurs when showLegendValue attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>sortMode changed</td>
<td>Yes</td>
<td>Occurs when sortMode attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>tickmarkFont changed</td>
<td>Yes</td>
<td>Occurs when tickmarkFont attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>tiltAngle changed</td>
<td>Yes</td>
<td>Occurs when tiltAngle attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>title1 changed</td>
<td>Yes</td>
<td>Occurs when title1 attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>title2 changed</td>
<td>Yes</td>
<td>Occurs when title2 attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>title3 changed</td>
<td>Yes</td>
<td>Occurs when title3 attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>title4 changed</td>
<td>Yes</td>
<td>Occurs when title4 attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>viewMode changed</td>
<td>Yes</td>
<td>Occurs when viewMode attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>viewpointDistance changed</td>
<td>Yes</td>
<td>Occurs when viewpointDistance attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>Event</td>
<td>Occurs when</td>
<td>Untrapped</td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>viewpointX changed</td>
<td>viewpointX attribute value is changed</td>
<td>Untrapped</td>
<td></td>
</tr>
<tr>
<td>viewpointY changed</td>
<td>viewpointY attribute value is changed</td>
<td>Untrapped</td>
<td></td>
</tr>
<tr>
<td>viewpointZ changed</td>
<td>viewpointZ attribute value is changed</td>
<td>Untrapped</td>
<td></td>
</tr>
</tbody>
</table>