Overview

The Pie control supports visualization of one to three response variables by one to three independent variables. You can control the labels displayed for each pie slice, and include or exclude labels for each variable. The independent variable(s) in the chart are treated as Classification variables. Each category (Sector) of the independent variable is displayed as an instance of a Pie slice. Basic statistics such as Sum, Mean and Frequency are supported. Once created, the object is fully interactive: you can zoom the chart in and out; rotate or tilt it; display, hide, move, or size its legend; manipulate color; and more.

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

Ancestors:

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

Class:

sashelp.classes.Pie_c.class

Attributes “Attributes” on page 495
Methods “Methods” on page 515
Events “Events” on page 538
Using Multidimensional Pie Charts “Using Multidimensional Pie Charts” on page 490
Using Multidimensional Pie Charts

About Multidimensional Pie Charts

The Pie Chart object lets you create pie charts in two- or three-dimensional views. You can use one, two, or three independent variables, and up to four response (dependent) variables. You can view one pie chart at a time, or multiple charts that are organized in a row or in a grid.

The simplest pie chart displays a two-dimensional view of a chart. The independent variable determines what slices are in the pie, and the response variable determines the size of each slice:
You can enhance the analysis by changing to a three-dimensional view of the chart, and also analyzing a second response variable, whose values determine the height of each slice:

If you add a second independent variable, a separate pie chart is displayed for each of that variable's values. The charts are organized in a row. If you add a third independent variable, its values are used to organize the charts into columns, effectively forming a grid of pie charts:

---

**Creating a Multidimensional Pie Chart**

To create a multidimensional pie chart, open a frame and select a Pie 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 Pie Chart” on page 492
- Specify a viewing mode “Specifying a Viewing Mode for a Multidimensional Pie Chart” on page 492
- Specify the data dimensions and variables for the analysis. “Specifying Data Dimensions and Variables for Multidimensional Pie Charts” on page 492
Specifying Data for a Multidimensional Pie Chart

To specify the data for a multidimensional pie chart, use the dataSet or dataSetID attributes, or reference the SCL list that contains the data. Use the pie 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 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:

```plaintext```
pie1.dataSource='dataset';
pie1.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');
pie1.dataSource='datasetid';
pie1.dataSetID=dsid;
```

Specifying a Viewing Mode for a Multidimensional Pie Chart

You can display pie charts in two- or three-dimensional viewing modes. To set the viewing mode, set the viewMode attribute to either 2D or 3D (the default). If you set viewMode to 2D, you cannot use the heightVariable attribute for a response variable. If you specify a height variable, it is ignored.

For a pie object named pie1, the following code sets the viewing mode to 2D:

```plaintext```
pie1.viewMode = '2d';
```

Specifying Data Dimensions and Variables for Multidimensional Pie Charts

A multidimensional pie chart 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.

To specify the variables for a multidimensional chart, use the attributes sliceVariable to specify the independent variable and the sizeVariable to specify the response (dependent) variable. The independent variable determines what slices are in the pie, and the response variable determines the sizes of the slices. The object supports long variable names, and mixed-case variable names.
For a pie chart object named pie1, the following code sets a slice variable and a size variable:

```javascript
pie1.dataDimensions = '1d';
pie1.sliceVariable = 'country';
pie1.sizeVariable = 'sales';
```

Additional Response Variables. To add a second response variable to the analysis, use the heightVariable attribute, which determines the height of each pie slice. This attribute is only available if the viewMode attribute is set to 3D. You can add another response variable with the colorVariable attribute, available in both two- and three-dimensional viewing modes. The color variable's values determine the slice colors.

Additional Independent Variables. To add a second independent variable to the analysis, use the rowVariable attribute. A separate pie chart is displayed for each of the row variable's values, and the charts are organized in a row. To add a third independent variable, use the columnVariable attribute. A separate pie chart is displayed for each of the column variable's values, and the charts are organized in a grid.

The number of independent variables must correspond to the pie chart's data dimensions. For analyzing two data dimensions, you must use the rowVariable attribute for the second independent variable. If you specify more independent variables than are set on the dataDimensions attribute, the additional independent variables are not used in the pie chart.

The following table summarizes the pie object's variable attributes.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Sets</th>
<th>Determines</th>
</tr>
</thead>
<tbody>
<tr>
<td>sliceVariable</td>
<td>first independent variable</td>
<td>pie slices</td>
</tr>
<tr>
<td>sizeVariable</td>
<td>first response variable</td>
<td>size of pie slices</td>
</tr>
<tr>
<td>colorVariable</td>
<td>additional response variable</td>
<td>color of pie slices</td>
</tr>
<tr>
<td>heightVariable</td>
<td>additional response variable</td>
<td>height of pie slices</td>
</tr>
<tr>
<td></td>
<td>(available only if viewMode is 3D)</td>
<td></td>
</tr>
<tr>
<td>rowVariable</td>
<td>second independent variable</td>
<td>number of pie charts in a row of charts</td>
</tr>
<tr>
<td>columnVariable</td>
<td>third independent variable</td>
<td>number of columns in a grid of pie charts</td>
</tr>
</tbody>
</table>

---

### Using Color in a Multidimensional Pie Chart

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 mark, and legend</td>
</tr>
<tr>
<td>backgroundColor</td>
<td>Color outside of the chart axes but within the object's borders</td>
</tr>
<tr>
<td>borderColor</td>
<td>Color for the border around the pie object</td>
</tr>
<tr>
<td>borderTitleColor</td>
<td>Text color for the border's title</td>
</tr>
</tbody>
</table>
Color Ranges in Multidimensional Pie Charts

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Specifies</th>
</tr>
</thead>
<tbody>
<tr>
<td>colorVariable</td>
<td>Slice colors</td>
</tr>
<tr>
<td>gridColor</td>
<td>Color of the chart’s grid lines</td>
</tr>
<tr>
<td>highlightColor</td>
<td>Color a pie slice turns when it is selected</td>
</tr>
<tr>
<td>missingColor</td>
<td>Color used to represent missing values in the chart</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 pie object named pie1, the following code sets the backgroundColor attribute to blue:

```javascript
pie1.backgroundColor = 'blue';
```

**Color Ranges in Multidimensional Pie Charts**

By default, the bar colors in a multidimensional pie 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 a 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 color range to determine the specific color used for each bar. To assign the color variable, use the colorVariable attribute.

For example, for a pie object named pie1, the following code sets the colorVariable to Revenue, and the colorVariableStatistic to Mean:

```javascript
pie1.colorVariable = 'revenue';
pie1.colorVariableStatistic = 'mean';
```

**Setting the Number of Pie Slices in Multidimensional Pie Charts**

To set the number of slices that are displayed in a multidimensional pie chart, use the numberOfSlices attribute. For example, for an object names pie1, the following code limits the pie chart to four slices:

```javascript
pie1.numberofSlices = 4;
```
By default, the pie chart uses the values of the size variable to determine which values from the data are displayed. In this case, the three largest values for the size variable are represented as pie slices, and the fourth pie slice represents all other values for the size variable.

To control which variable's values determine the pie slices, use the groupingVariable attribute. Valid values are Slice, Size, Height, and Color.

Attributes

Attributes specified for the Pie 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 Pie control: attachedInterface, CBTFrameName, contentsUpdatedAttributes, help, helpText, model, and toolTipText.

Public Attributes

Dictionary

**actionMode**

Returns or sets the action mode

Type: Character

Initial Value: (Object)

Valid Values: Pick Probe Move Rotate Tilt Spin Zoom Legend ScrollLegend

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>-----------</td>
<td>---------------------------------------------------------------------------------------------------------------</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>
</tbody>
</table>

**animateMode**

Returns or sets the animation mode

Type: Character
Initial Value: (Object)
Valid Values: Filled Wireframe
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
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 for data mapping

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

The Pie 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

---

**columnDataMax**

Returns or sets the maximum value for the Column

Type: Numeric
**columnDataMin**

Returns or sets the minimum value for the Column

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

**columnVariable**

Returns or sets the column variable

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

**dataDimensions**

Returns or sets value for dataset dimensions

- **Type:** Character
- **Initial Value:** (Object)
- **Valid Values:** 1D 2D 3D
- **Category:** Appearance
**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

**defaultAttribute**

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

Type: Character
description

Returns the short description for the object

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

ggridColor

Returns or sets the grid color
Type: Character
Initial Value: (Object)
Valid Values:
Category: Appearance

groupingVariable

Returns or sets the grouping variable
Type: Character
**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

**heightVariableType**

Returns or sets the height variable type

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

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

- **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

**missingColor**

Returns or sets the color used to represent missing values

- **Type:** Character
**numberOfSlices**

Returns or sets the maximum number of slices in a pie.

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

**pickList**

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

- **Type**: List
- **Initial Value**: (Object)
- **Valid Values**: None
- **Category**: Data

**pieHeightScale**

Returns or sets the pie height scaling factor.

- **Type**: Numeric
- **Initial Value**: (Object)
- **Valid Values**: None
- **Category**: Appearance
pieHeightType

Returns or sets the pie height type used to display a pie.

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

rotationAngle

Returns or sets the amount to rotate.

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.

rowDataMax

Returns or sets the maximum value for the Row.

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

rowDataMin

Returns or sets the minimum value for the Row.

Type: Numeric
### rowVariable

Returns or sets the row variable

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

### scale

Returns or sets the scale value used for changing the object’s size

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  
Category: Appearance
showColorLabel

Returns or sets the color label state

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

showHeightLabel

Returns or sets the height label state

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

showLegend

Returns or sets the legend display state

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

showLegendValue

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

Type: Character
showSizeLabel

Returns or sets the size label state
Type: Character
Initial Value: (Object)
Valid Values: Yes No
Category: Appearance

showSliceLabel

Returns or sets the slice label state
Type: Character
Initial Value: (Object)
Valid Values: Yes No
Category: Appearance

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

sliceDataMax

Returns or sets the maximum value for the Slice

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

sliceDataMin

Returns or sets the minimum value for the Slice

Type: Numeric
**sliceVariable**

Returns or sets the slice variable

Type: Character

Initial Value: (Object)
Valid Values: \sashelp.classes.chartvalues.scl

Category: Data

---

**thresholdValueAbsolute**

Returns or sets the absolute threshold value to group slices

Type: Numeric

Initial Value: (Object)
Valid Values:

Category: Appearance

---

**thresholdValuePercent**

Returns or sets the threshold value in percent to group slices

Type: Numeric

Initial Value: (Object)
Valid Values:

Category: Appearance
**tickmarkFont**

Returns or sets the tick mark font

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

**tiltAngle**

Returns or sets the tilt angle

- **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 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
Category: Appearance
**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 Pie Control class are described here. See sashelp.fsp.Widget.class"Methods" on page 145 for inherited methods.

### Public Methods

#### _binit

Initializes an object at design time

________

**Syntax**

```objectName._binit( );```

#### _explodeSlice

Sets the slice to exploded state

________

**Syntax**

```objectName._explodeSlice( arg1 );```  
```objectName._explodeSlice( );```  

<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></td>
</tr>
</tbody>
</table>
_getPickList

Return the 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 slice was selected with actionMode set to Pick.</td>
</tr>
</tbody>
</table>

_getPickedItemCount

Return the number of items in the picklist

Syntax

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 slice was selected with actionMode set to Pick.</td>
</tr>
</tbody>
</table>

_hideSlice

Sets the slice hidden state
Syntax

objectName_hideSlice( arg1 );
objectName_hideSlice( );

<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></td>
</tr>
</tbody>
</table>

_init

Initializes an object

Syntax

objectName_init( );

_print

Print

Syntax

objectName_print( );

_resetView

Set the view back to the default setting
Syntax

objectName_resetView();

__setTextAlignment

Set text alignment

Syntax

objectName_setTextAlignment( arg1, arg2 );

<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></td>
</tr>
<tr>
<td>arg2</td>
<td>Numeric</td>
<td>I</td>
<td></td>
</tr>
</tbody>
</table>

__setTextColor

Set text color

Syntax

objectName_setTextColor( arg1, arg2 );

<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></td>
</tr>
<tr>
<td>arg2</td>
<td>Numeric</td>
<td>I</td>
<td></td>
</tr>
</tbody>
</table>
_setTextFont

Set text font

Syntax

objectName_setTextFont( arg1, arg2 );

<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></td>
</tr>
<tr>
<td>arg2</td>
<td>Numeric</td>
<td>I</td>
<td></td>
</tr>
</tbody>
</table>

_setTextLabel

Set text label

Syntax

objectName_setTextLabel( arg1, arg2 );

<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></td>
</tr>
<tr>
<td>arg2</td>
<td>Numeric</td>
<td>I</td>
<td></td>
</tr>
</tbody>
</table>

_showSlice

Sets the slice visible state

Syntax

objectName_showSlice( arg1 );
objectName_showSlice( );
**unexplodeSlice**

Sets the slice to unexploded state

Syntax

```plaintext
objectName_unexplodeSlice( arg1 );
objectName_unexplodeSlice( );
```

<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></td>
</tr>
</tbody>
</table>

**_setcamActionMode**

sets the actionMode attribute

Syntax

```plaintext
objectName_setcamActionMode( mode );
```

**_setcamAnimateMode**

sets the animateMode attribute

Syntax

```plaintext
objectName_setcamAnimateMode( mode );
```
**_setcamAxisColor_**

sets the axisColor attribute

----------

**Syntax**

```
objectName_setcamAxisColor( color );
```

**_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>font</td>
</tr>
</tbody>
</table>

**_setcamColorList_**

sets the colorList attribute

----------

**Syntax**

```
objectName_setcamColorList( colorList );
```
### _setcamColorRangeObject

Invoked when the **colorRangeObject** attribute is changed

#### Syntax

```
return = objectName_setcamColorRangeObject( object );
```

### _setcamColorSource

Invoked when the **colorSource** attribute is changed

#### Syntax

```
objectName_setcamColorSource( source );
```

### _setcamColorVariable

Sets the **colorVariable**

#### Syntax

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

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

Syntax

```
objectName_setcamColorVariableStatistic( stat );
```

**_setcamColorVariableType**

sets the color variable type

Syntax

```
objectName_setcamColorVariableType( type );
```

**_setcamColumnDataMax**

sets the columnDataMax attribute

Syntax

```
objectName_setcamColumnDataMax( max );
```

**_setcamColumnDataMin**

sets the columnDataMin attribute
_setcamColumnVariable

sets the column variable

Syntax

objectName_setcamColumnDataMin( min );

_setcamColumnVariable

sets the column variable

Syntax

objectName_setcamColumnVariable( varName );

_setcamDataDimensions

sets the dataDimension: 1D, 2D, or 3D

Syntax

objectName_setcamDataDimensions( dimension );

_setcamDataSetID

sets the data set id

Syntax

objectName_setcamDataSetID( id );
_setcamDataSetName

sets the data set name

Syntax

dobjectName_setcamDataSetName( dsname );

_setcamDataSource

sets the data source

Syntax

dobjectName_setcamDataSource( source );

_setcamGridColor

sets the grid color

Syntax

doobjectName_setcamGridColor( color );

_setcamGroupingVariable

sets the variable to use for grouping: Size, Slice, Height, or Color
_setcamHeightVariable

sets the height variable

__________________________

**Syntax**

objectName_setcamGroupingVariable( arg );

__________________________

_setcamHeightVariable

sets the height variable

__________________________

**Syntax**

objectName_setcamHeightVariable( varName );

__________________________

_setcamHeightVariableStatistic

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

__________________________

**Syntax**

objectName_setcamHeightVariableStatistic( stat );

__________________________

_setcamHeightVariableType

sets the height variable's type

__________________________

**Syntax**

objectName_setcamHeightVariableType( type );
_setcamHighlightColor

sets the highlightColor attribute

Syntax

objectName_setcamHighlightColor( color );

_setcamLegendFont

sets the legendFont attribute

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 );
_setcamMissingColor

sets the missingColor attribute

Syntax

objectName_setcamMissingColor( color );

_setcamNumberOfSlices

sets the numberOfSlices attribute

Syntax

objectName_setcamNumberOfSlices( num );

_setcamPickList

sets the pickList attribute

Syntax

objectName_setcamPickList( arg1 );

<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>values for the pick list</td>
</tr>
</tbody>
</table>

_setcamPieHeightScale

sets the pieHeightScale attribute
**Syntax**

```cpp
objectName_setcamPieHeightScale( scale );
```

**_setcamPieHeightType**

sets the pieHeightType attribute: Absolute or Relative

**Syntax**

```cpp
objectName_setcamPieHeightType( type );
```

**_setcamPieStartAngle**

sets the pie start angle

**Syntax**

```cpp
objectName_setcamPieStartAngle( angle );
```

**_setcamProjectionAngle**

sets the projection angle

**Syntax**

```cpp
objectName_setcamProjectionAngle( angle );
```
_setcamRotationAngle

sets the rotation angle

Syntax

objectName_setcamRotationAngle( angle );

_setcamRowDataMax

sets the rowDataMax attribute

Syntax

objectName_setcamRowDataMax( max );

_setcamRowDataMin

sets the rowDataMin attribute

Syntax

objectName_setcamRowDataMin( min );

_setcamRowVariable

sets the row variable
Syntax
objectName_setcamRowVariable( varName );

_setcamScale
sets the scale attribute

Syntax
objectName_setcamScale( scale );

_setcamSelectionMode
sets the selectionMode attribute

Syntax
objectName_setcamSelectionMode( mode );

_setcamShowColorLabel
shows or hides label for color variable: accepts Yes or No

Syntax
objectName_setcamShowColorLabel( arg1 );
_setcamShowHeightLabel

shows or hides label for height variable: accepts Yes or No

Syntax

objectName_setcamShowHeightLabel( arg1 );

_setcamShowLegend

shows or hides legend: accepts Yes or No

Syntax

objectName_setcamShowLegend( arg1 );

_setcamShowLegendValue

Invoked when the showLegendValue attribute is changed: accepts Yes or No

Syntax

objectName_setcamShowLegendValue( arg1 );

_setcamShowSizeLabel

Invoked when the showSizeLabel attribute is changed: accepts Yes or No
Pie Chart Control

_setcamSizeVariable

Syntax

objectName_setcamSizeVariable( varName );

_setcamShowSliceLabel

shows or hides slice label: accepts Yes or No

Syntax

objectName_setcamShowSliceLabel( arg1 );

_setcamSideColor

sets side color

Syntax

objectName_setcamSideColor( color );

_setcamSizeVariable

sets size variable

Syntax

objectName_setcamSizeVariable( varName );
_setcamSizeVariableStatistic

sets size variable's statistic: Mean, Sum, or Freq

Syntax

objectName_setcamSizeVariableStatistic( stat );

_setcamSizeVariableType

sets size variable’s type

Syntax

objectName_setcamSizeVariableType( type );

_setcamSliceDataMax

sets sliceDataMax attribute

Syntax

objectName_setcamSliceDataMax( max );

_setcamSliceDataMin

sets the sliceDataMin attribute
### _setcamThresholdValuePercent

**Syntax**

```plaintext
objectName_setcamThresholdValuePercent( percent );
```

_setscamSliceVariable

sets the slice variable

### _setcamThresholdValueAbsolute

**Syntax**

```plaintext
objectName_setcamThresholdValueAbsolute( num );
```

### _setcamThresholdValuePercent

**Syntax**

```plaintext
objectName_setcamThresholdValuePercent( percent );
```
_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>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 );

_argument Type Use Description
arg1 Numeric I x coordinate

_setcamViewpointX

sets the viewpointX attribute

Syntax

objectName_setcamViewpointX( coord );

_argument Type Use Description
arg1 Numeric I x coordinate

_setcamViewpointY

sets the viewpointY attribute

Syntax

objectName_setcamViewpointY( coordinate );

_argument Type Use Description
arg1 Numeric I y coordinate
_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>I</td>
<td>z coordinate</td>
</tr>
</tbody>
</table>

---

**Events**

Events specified for the Pie 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>actionMode changed</td>
<td>Yes</td>
<td>Occurs when actionMode attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>animateMode changed</td>
<td>Yes</td>
<td>Occurs when animateMode attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>axisColor changed</td>
<td>Yes</td>
<td>Occurs when axisColor attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>axisLabelFont changed</td>
<td>Yes</td>
<td>Occurs when axisLabelFont attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>colorList changed</td>
<td>Yes</td>
<td>Occurs when colorList attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>colorRangeObject changed</td>
<td>Yes</td>
<td>Occurs when colorRangeObject attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>colorSource changed</td>
<td>Yes</td>
<td>Occurs when colorSource attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>Event Name</td>
<td>Trigger</td>
<td>Occurrence</td>
<td>Status</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------</td>
<td>----------------------</td>
<td>---------</td>
</tr>
<tr>
<td>colorVariable changed</td>
<td>Yes</td>
<td>Occurs when colorVariable attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>colorVariableStatistic changed</td>
<td>Yes</td>
<td>Occurs when colorVariableStatistic attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>colorVariableType changed</td>
<td>Yes</td>
<td>Occurs when colorVariableType attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>columnDataMax changed</td>
<td>Yes</td>
<td>Occurs when columnDataMax attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>columnDataMin changed</td>
<td>Yes</td>
<td>Occurs when columnDataMin attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>columnVariable changed</td>
<td>Yes</td>
<td>Occurs when columnVariable attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>dataDimensions changed</td>
<td>Yes</td>
<td>Occurs when dataDimensions attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>dataSet changed</td>
<td>Yes</td>
<td>Occurs when dataSet attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>dataSetID changed</td>
<td>Yes</td>
<td>Occurs when dataSetID attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>dataSource changed</td>
<td>Yes</td>
<td>Occurs when dataSource attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>footnote1 changed</td>
<td>Yes</td>
<td>Occurs when footnote1 attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>footnote2 changed</td>
<td>Yes</td>
<td>Occurs when footnote2 attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>gridColor changed</td>
<td>Yes</td>
<td>Occurs when gridColor attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>Attribute</td>
<td>Condition</td>
<td>Description</td>
<td>Status</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------</td>
<td>-------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>groupingVariable changed</td>
<td>Yes</td>
<td>Occurs when groupingVariable attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>heightVariable changed</td>
<td>Yes</td>
<td>Occurs when heightVariable attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>heightVariableStatistic changed</td>
<td>Yes</td>
<td>Occurs when heightVariableStatistic attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>heightVariableType changed</td>
<td>Yes</td>
<td>Occurs when heightVariableType attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>highlightColor changed</td>
<td>Yes</td>
<td>Occurs when highlightColor attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>legendFont changed</td>
<td>Yes</td>
<td>Occurs when legendFont attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>legendType changed</td>
<td>Yes</td>
<td>Occurs when legendType attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>missingColor changed</td>
<td>Yes</td>
<td>Occurs when missingColor attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>numberOfSlices changed</td>
<td>Yes</td>
<td>Occurs when numberOfSlices attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>pickList changed</td>
<td>Yes</td>
<td>Occurs when pickList attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>pieHeightScale changed</td>
<td>Yes</td>
<td>Occurs when pieHeightScale attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>pieHeightType changed</td>
<td>Yes</td>
<td>Occurs when pieHeightType attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>projectionAngle changed</td>
<td>Yes</td>
<td>Occurs when projectionAngle attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>Event Name</td>
<td>Changeable</td>
<td>Occurs When</td>
<td>Untrapable</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------</td>
<td>-----------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>rotationAngle changed</td>
<td>Yes</td>
<td>Occurs when rotationAngle attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>rowDataMax changed</td>
<td>Yes</td>
<td>Occurs when rowDataMax attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>rowDataMin changed</td>
<td>Yes</td>
<td>Occurs when rowDataMin attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>rowVariable changed</td>
<td>Yes</td>
<td>Occurs when rowVariable attribute value is changed</td>
<td>Untrapped</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>showColorLabel changed</td>
<td>Yes</td>
<td>Occurs when showColorLabel attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>showHeightLabel changed</td>
<td>Yes</td>
<td>Occurs when showHeightLabel 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>showSizeLabel changed</td>
<td>Yes</td>
<td>Occurs when showSizeLabel attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>showSliceLabel changed</td>
<td>Yes</td>
<td>Occurs when showSliceLabel attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>sizeVariable changed</td>
<td>Yes</td>
<td>Occurs when sizeVariable attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>Event Description</td>
<td>Occurs when</td>
<td>Attribute Value Changed</td>
<td></td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-------------</td>
<td>-------------------------</td>
<td></td>
</tr>
<tr>
<td>sizeVariableStatistic changed</td>
<td>Yes</td>
<td>attribute value is changed</td>
<td></td>
</tr>
<tr>
<td>sizeVariableType changed</td>
<td>Yes</td>
<td>attribute value is changed</td>
<td></td>
</tr>
<tr>
<td>sliceDataMax changed</td>
<td>Yes</td>
<td>attribute value is changed</td>
<td></td>
</tr>
<tr>
<td>sliceDataMin changed</td>
<td>Yes</td>
<td>attribute value is changed</td>
<td></td>
</tr>
<tr>
<td>sliceVariable changed</td>
<td>Yes</td>
<td>attribute value is changed</td>
<td></td>
</tr>
<tr>
<td>thresholdValueAbsolute changed</td>
<td>Yes</td>
<td>attribute value is changed</td>
<td></td>
</tr>
<tr>
<td>thresholdValuePercent changed</td>
<td>Yes</td>
<td>attribute value is changed</td>
<td></td>
</tr>
<tr>
<td>tickmarkFont changed</td>
<td>Yes</td>
<td>attribute value is changed</td>
<td></td>
</tr>
<tr>
<td>tiltAngle changed</td>
<td>Yes</td>
<td>attribute value is changed</td>
<td></td>
</tr>
<tr>
<td>title1 changed</td>
<td>Yes</td>
<td>attribute value is changed</td>
<td></td>
</tr>
<tr>
<td>title2 changed</td>
<td>Yes</td>
<td>attribute value is changed</td>
<td></td>
</tr>
<tr>
<td>title3 changed</td>
<td>Yes</td>
<td>attribute value is changed</td>
<td></td>
</tr>
<tr>
<td>title4 changed</td>
<td>Yes</td>
<td>attribute value is changed</td>
<td></td>
</tr>
<tr>
<td>viewMode changed</td>
<td>Yes</td>
<td>attribute value is changed</td>
<td></td>
</tr>
<tr>
<td>Event</td>
<td>Enabled</td>
<td>Description</td>
<td>Status</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>viewpointDistance</td>
<td>Yes</td>
<td>Occurs when viewpointDistance attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>viewpointX changed</td>
<td>Yes</td>
<td>Occurs when viewpointX attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>viewpointY changed</td>
<td>Yes</td>
<td>Occurs when viewpointY attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>viewpointZ changed</td>
<td>Yes</td>
<td>Occurs when viewpointZ attribute value is changed</td>
<td>Untrapped</td>
</tr>
</tbody>
</table>