Overview

The CSF control lets you create a Critical Success Factor object, which is a graphic that visually represents a value in a range of data. There are three representations of a CSF: Classic, Digital, and Speedometer. In each of them, an indicator shows the current value of the CSF. This value can be obtained from a SAS data set variable or an SCL variable.

The range of data used by the CSF is depicted by segments, with each segment containing a beginning and ending value, and possibly a different color. The number of segments and their corresponding colors are defined in a RANGE entry that is associated with the CSF.

The Classic CSF presents a half-moon graphic, with the segments depicted from the minimum at the left rotating around to the maximum at the right. A hub is drawn in the center, and an indicator arm points out to the edge of the graphic to where the current value is located in the range. Segment values are labeled around the edge of the CSF, and a legend may be located at any corner of the CSF or centered above or below it.

The Digital CSF presents a simple segmented bar across the top of the graphic, with a digital readout below it representing the current value of the CSF. An indicator bar is drawn at the location along the bar where the current value is located. Segment values are labeled across the top of the bar.

The Speedometer CSF presents an arc of segments rising from the left up to straight up, with a digital readout of the current value shown inside the arc. Segment values are labeled along the inside of the arc.

Note: To use the Critical Success Factor control, you must have SAS/GRAPH software licensed at your site.
Using Critical Success Factor Controls

The CSF provides a default RANGE entry of three segments equally spaced between 0 and 100, with the colors of red, yellow, and green, respectively. Alternately, you can assign another valid RANGE entry for the CSF to use. The current value of the CSF is by default 0, and may be set by assigning a SAS data set and appropriate variable to the CSF, where the CSF will read out the first value for that variable in the data set and assign it to the current value. Alternately, you can leave the data set value NULL and directly assign the current value of the CSF, from whatever source you want, using the value attribute.

The CSF allows you to modify many visual attributes, such as location of the legend, fonts for the legend, value, and labels, and shading (none, simple, and shaded). All color and font assignments are standard SAS color and font interfaces.

If the current value is MISSING or falls below the minimum value in the current range, the indicator is placed at the minimum location on the CSF. If the current value is greater than the maximum value in the current range, the indicator is placed at the maximum location.

The Label attributes are for the numbers displayed for each of the segments. The Legend attributes are for displaying the variable name or additional annotation about the chosen variable. The Value attributes are for displaying the actual current value.

Attributes

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

Public Attributes

Dictionary

arrowColor

Returns or sets the color of the arrow in the Classic CSF

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

Returns or sets the color of the backdrop in the Digital CSF

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

**dataSet**

Returns or sets the dataset to be used to set the value. Once the dataSet and variableName from that dataSet have been chosen, the CSF will read the first entry of that variable from the dataSet and set the value.

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

**dataSetID**

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

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

Returns or sets the digital readout display. Used with the Digital and Speedometer CSF’s. The default is on. If off, the Digital CSF only displays the segmented bar, and the Speedometer only displays the arc of labeled segments with the value displayed below.

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

graphStyle

Returns or sets the graphic representation to display. Three styles are available: None, Simple, and Shaded. With None the CSF is draw in a flat 2D style. With Simple, backdrops and bevels are added to simulate depth. With Shaded, a lighting calculation is performed on the backdrops and bevels to simulate a light falling on the 3D object.

Type: Character
Initial Value: (Object)
Valid Values: None Simple Shaded
Category: graphType

graphType

Returns or sets the graphic representation to display. The three choices are Classic, Digital, and Speedometer.

Type: Character
Initial Value: (Object)
Valid Values: Classic Digital Speedometer
Category: Appearance
**hubColor**

Returns or sets the color of the hub in the Classic CSF

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

---

**indicator**

Returns or sets the indicator to use in the Classic CSF. The choices are used in the Classic CSF, and are Arrow, Spear, Harpoon, and Line. They specify how the indicator appears.

Type: Character  
Initial Value: (Object)  
Valid Values: Arrow Spear Harpoon Line  
Category:

---

**labelColor**

Returns or sets the color of the label in the CSF

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

---

**labelFont**

Returns or sets the font used to display the label

Type: List
**labelFormat**

Returns or sets the format to use when displaying the labels

**Type:** Character

<table>
<thead>
<tr>
<th>Initial Value</th>
<th>Valid Values</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Object)</td>
<td>Range Dataset</td>
<td>Misc</td>
</tr>
</tbody>
</table>

**legendColor**

Returns or sets the color of the legend in the CSF

**Type:** Character

<table>
<thead>
<tr>
<th>Initial Value</th>
<th>Valid Values</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Object)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**legendFont**

Returns or sets the font used to display the legend

**Type:** List

<table>
<thead>
<tr>
<th>Initial Value</th>
<th>Valid Values</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Object)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
legendPosition

Returns or sets the position of the legend in the widget

Type: Character
Initial Value: (Object)
Valid Values: TopLeft TopCenter TopRight BottomLeft BottomCenter BottomRight
Category:

legendText

Returns or sets the legend label string

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

outlineColor

Returns or sets the color of the outline in the CSF

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

range

Returns or sets the range entry to be used. A RANGE entry name that specifies the three- or four-level name that defines the number of segments in the range and the color and attribute for each segment. For more information about RANGE entries, refer to other SAS/AF documentation.
shaftSize

Returns or sets the width of the shaft of the indicator. The width of the indicator shaft in the Classic CSF. The default is 4, with a range from 1 to 10.

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

value

Returns or sets the current value of the CSF. If the dataSet is NULL, this attribute is used to set the current value of the CSF. If the dataSet is valid, this attribute is ignored.

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

valueColor

Returns or sets the color of the value in the CSF

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

Returns or sets the font used to display the value

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

valueFormat

Returns or sets the format to use when displaying the values

Type: Character
Initial Value: (Object)
Valid Values: Range Dataset
Category: Misc

variableName

Returns or sets the variable in the dataset to be used. If the dataSet is NULL, the value of the CSF is taken from the value attribute. If the dataSet is valid and the variableName is specified from the dataSet, the value of the CSF is the first entry from this variable in this dataSet.

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

Methods specified for the Critical Success Factor Control class are described here. See sashelp.fsp.Widget.class “Methods” on page 145 for inherited methods.

Public Methods

__print

Print the CSF

Syntax

objectName__print( );

__setcamArrowColor

Set the color of the Arrow.

Syntax

objectName_setcamArrowColor( color );

__setcamBackdropColor

Set the color of the Backdrop.

Syntax

objectName_setcamBackdropColor( color );
_setcamDataSet

Invoked when the dataSet attribute is changed

**Syntax**

```
objectName_setcamDataSet( dsname );
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dsname</td>
<td>Character</td>
<td>I</td>
<td>Set the Dataset name on the CSF</td>
</tr>
</tbody>
</table>

_setcamDataSetID

Set the dataset ID on the CSF.

**Syntax**

```
objectName_setcamDataSetID( id );
```

_setcamDigitalSwitch

Set the digital text to be on/off.

**Syntax**

```
objectName_setcamDigitalSwitch( arg1 );
```
### _setcamDsId

Set the dataset ID on the CSF.

**Syntax**

```c
objectName_setcamDsId(id);
```

### _setcamDsName

Set the dataset Name on the CSF.

**Syntax**

```c
objectName_setcamDsname(dsname);
```

### _setcamFont

Set the Font.

**Syntax**

```c
objectName_setcamFont(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>

### _setcamGraphStyle

Set the Style of the CSF.
**Syntax**

```
objectName_setcamGraphStyle(style);
```

---

**_setcamGraphType**

Set the Type of the CSF.

---

**Syntax**

```
objectName_setcamGraphType(type);
```

---

**_setcamHubColor**

Set the color of the Hub.

---

**Syntax**

```
objectName_setcamHubColor(color);
```

---

**_setcamIndicatorType**

Set the type of the Indicator.

---

**Syntax**

```
objectName_setcamIndicatorType(type);
```
_setcamLabelColor

Set the color of the Label.

Syntax

objectName_setcamLabelColor( color );

_setcamLabelFont

Set the font for the Label.

Syntax

objectName_setcamLabelFont( 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>

_setcamLabelFormat

Set the format of the Label.

Syntax

objectName_setcamLabelFormat( format );

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>format</td>
<td>Numeric</td>
<td>I</td>
<td>format</td>
</tr>
</tbody>
</table>
**_setcamLegendColor_**

Set the color of the Legend.

Syntax

```
objectName_setcamLegendColor( color );
```

**_setcamLegendFont_**

Set the font for the Legend.

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>

**_setcamLegendPosition_**

Set the position of the Legend.

Syntax

```
objectName_setcamLegendPosition( pos );
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pos</td>
<td>Character</td>
<td>I</td>
<td>position</td>
</tr>
</tbody>
</table>
_setcamLegendText

Set the text for the Legend.

Syntax

objectName_setcamLegendText( text );

_setcamOutlineColor

Set the color of the Outline.

Syntax

objectName_setcamOutlineColor( color );

_setcamRange

Set the range on the CSF.

Syntax

objectName_setcamRange( range );

_setcamShaftSize

Set the Indicator shaft size.
**Syntax**

```
objectName_setcamShaftSize( size );
```

---

**_setcamValue**

Set the value of the CSF

---

**Syntax**

```
objectName_setcamValue( input );
```

---

**_setcamValueColor**

Set the color of the Value.

---

**Syntax**

```
objectName_setcamValueColor( color );
```

---

**_setcamValueFont**

Set the font for the Value.

---

**Syntax**

```
objectName_setcamValueFont( font );
```
_setcamValueFormat

Set the format of the Value.

Syntax

objectId._setcamValueFormat( format );

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

_setcamVariableName

Set the variable name on the CSF.

Syntax

objectId._setcamVariableName( varName );

Events

Events specified for the Critical Success Factor 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>arrowColor changed</td>
<td>Yes</td>
<td>Occurs when arrowColor attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>backdropColor changed</td>
<td>Yes</td>
<td>Occurs when backdropColor 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>digitalSwitch changed</td>
<td>Yes</td>
<td>Occurs when digitalSwitch attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>graphStyle changed</td>
<td>Yes</td>
<td>Occurs when graphStyle attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>graphType changed</td>
<td>Yes</td>
<td>Occurs when graphType attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>hubColor changed</td>
<td>Yes</td>
<td>Occurs when hubColor attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>indicator changed</td>
<td>Yes</td>
<td>Occurs when indicator attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>labelColor changed</td>
<td>Yes</td>
<td>Occurs when labelColor attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>labelFont changed</td>
<td>Yes</td>
<td>Occurs when labelFont attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>labelFormat changed</td>
<td>Yes</td>
<td>Occurs when labelFormat attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>legendColor changed</td>
<td>Yes</td>
<td>Occurs when legendColor 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>legendPosition changed</td>
<td>Yes</td>
<td>Occurs when legendPosition attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>Event</td>
<td>Occurs when</td>
<td>Description</td>
<td>Trapped</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------</td>
<td>--------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>legendText changed</td>
<td>Yes</td>
<td>legendText attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>outlineColor changed</td>
<td>Yes</td>
<td>outlineColor attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>range changed</td>
<td>Yes</td>
<td>range attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>shaftSize changed</td>
<td>Yes</td>
<td>shaftSize attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>value changed</td>
<td>Yes</td>
<td>value attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>valueColor changed</td>
<td>Yes</td>
<td>valueColor attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>valueFont changed</td>
<td>Yes</td>
<td>valueFont attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>valueFormat changed</td>
<td>Yes</td>
<td>valueFormat attribute value is changed</td>
<td>Untrapped</td>
</tr>
<tr>
<td>variableName changed</td>
<td>Yes</td>
<td>variableName attribute value is changed</td>
<td>Untrapped</td>
</tr>
</tbody>
</table>