Overview

The Critical Success Factor class enables you to create a critical success factor (CSF). A CSF is a graphic that visually represents the position of some value in a range of data. An arrow indicates the current value of the CSF. This value can be obtained from a SAS data set variable or an SCL variable. The range depicted in the CSF may be divided into segments, and each segment may have a different color and attribute. The number of segments and their corresponding colors are defined in a RANGE entry that is associated with the CSF.

Parent:

sashelp.fsp.widget.class

Class:

sashelp.fsp.CSF.class

Methods

Methods specific to the Critical Success Factor class are described here. Inherited methods are described in the Object class and the Widget class.

Dictionary

_getValue

Returns the current value of the critical success factor

Inherited from Widget
Syntax
CALL NOTIFY (csf-name, '_getValue', csf-value);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>csf-value</td>
<td>N</td>
<td>returns the value of the critical success factor</td>
</tr>
</tbody>
</table>

Details
The value returned by the _getValue method is visually represented as the location of the arrow on the CSF picture.

$selectFormat

Selects the format for displaying minimum and maximum values

Syntax
CALL NOTIFY (csf-name, '_selectFormat', type);

<table>
<thead>
<tr>
<th>Where ...</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>C</td>
<td>specifies the type of format to use. Valid values are DATA for the format data set and RANGE for the format specified in the RANGE entry.</td>
</tr>
</tbody>
</table>

_setColor

Assigns a color to parts of the CSF that are not controlled via the RANGE entry

Restrictions Some systems do not support using methods to change the colors of a critical success factor.

Syntax
CALL NOTIFY (csf-name, '_setColor', piece-type, color-name);
Critical Success Factor Class

_setDsname

Assigns the name of the data set containing the current value of the CSF

**Syntax**

CALL NOTIFY (csf-name, `_setDsname`, data-set-name);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data-set-name</td>
<td>C</td>
<td>specifies the name of the data set</td>
</tr>
</tbody>
</table>

_setDepVar

Assigns the name of a variable that contains the current value of the CSF

**Syntax**

CALL NOTIFY (csf-name, `_setDepVar`, dep-var-name);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dep-var-name</td>
<td>C</td>
<td>specifies the name of the variable</td>
</tr>
</tbody>
</table>

_setColor

specifies which of these pieces of the CSF to assign the color to:

- `'HUB'` Semicircle at the base of the arrow
- `'ARROW'` Arrow pointing to current value
- `'OUTLINE'` Outline of range segments and hub
- `'TEXT'` Current range labels
- `'TVALUE'` Current value.

**Example**

The _setColor method sets the colors to red for the CSF named CSF1:

```plaintext
    call notify('csf1','_set_color_','HUB','RED');
    call notify('csf1','_set_color_','ARROW','RED');
    call notify('csf1','_set_color_','OUTLINE','RED');
    call notify('csf1','_set_color_','TEXT','RED');
```
Argument | Type | Description
---|---|---
data-set-name | C | specifies the name of the SAS data set

Details

- If you changed the data set name for a previously drawn CSF, the CSF is redrawn using the new data set.
- If you changed the data in the data set, call the _update method to get the CSF to redraw with the changed data.

__setFont

Assigns a font for labeling a CSF

Syntax

CALL NOTIFY (csf-name, '_setFont', font-name);

<table>
<thead>
<tr>
<th>Where ...</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| font-name | C | specifies the SAS/GRAPH font. For valid fonts, see SAS/GRAPH Software: Reference

__setRange

Assigns a RANGE entry

Syntax

CALL NOTIFY (csf-name, '_setRange', range-object-name);

<table>
<thead>
<tr>
<th>Where ...</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| range-object-name | C | specifies the four-level name of the RANGE entry to assign to the CSF

Details

A RANGE entry defines the number of segments in the CSF’s range and the color and attribute for each segment.
**_setSize_**

Assigns the height of the graphic text that appears on the CSF

---

**Syntax**

CALL NOTIFY (csf-name, '_setSize', size);

Where ...

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>specifies the height of the graphic text in units of character cells. Values are 0.001 through 3.000.</td>
</tr>
</tbody>
</table>

**Example**

The _setSize_ method changes the height of the labels for CSF1 to 2.5 character cells:

```
call notify('csf1', '_set_size_', 2.5);
```

---

**_setValue_**

Assigns a value to a CSF

Inherited from Widget

---

**Syntax**

CALL NOTIFY (csf-name, '_setValue', csf-value);

Argument Type Description

| csf-value | N | specifies the value to assign to the CSF |

**Details**

If you want to use _setValue from the INIT section of the frame and you have set the data set name and the variable in the Attributes window, first clear the data set name and variable by calling _setDsname and _setDepVar with empty strings. If you fail to do this, the value for the CSF is retrieved from the data set. Note that this also applies to calls to _setValue prior to a call to _update.

```
INIT:
    call notify('csf', '_setDsname', '');
    call notify('csf', '_setDepVar', '');
    call notify('csf', '_setVALUE', 30);
    return;
```
or

        BUTTON:
          call notify('csf', '_setDsname', '');
          call notify('csf', '_setDepVar', '');
          call notify('csf', '_setValue', 25);
          call notify('csf', '_update');
          return;

Example

The _setValue method changes the value of CSF1 from 1 to 100:

MAIN:
  do i = 1 to 100;
    csf = i;
    /* You can also use this line */
    /* CALL NOTIFY('csf1', '_setValue', i); */
    refresh;
  end;
return;