Overview

The SAS/GRAPH Output class displays graphics output stored in a GRSEG catalog entry. This class enables you to present data in a visual form, configure output as push buttons, or define areas of the output, called hotspots, which can perform certain actions when they are selected by users. For example, on a United States map, you can make the state of North Carolina a hotspot that changes color when a user selects it.

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

Parent:

sashelp.fsp.widget.class

Class:

sashelp.fsp.Graph.class

Using the SAS/GRAPH Output Class

Subclassing the Segment Hotspot Class

Once you create a SAS/GRAPH output object, you can define hotspots for the object using a subclass of the SAS/GRAPH Output class, the Segment Hotspot class. The Segment Hotspot class provides a way to connect hotspots in your graphics output to labeled sections in your SCL program so that the labeled sections execute when the hotspot is selected.
The Segment Hotspot class is very similar to the Hotspot class; however, the Segment Hotspot class is available only in connection with the SAS/GRAPH Output class. Segment hotspots are defined only in terms of the SAS/GRAPH output they intersect. Segment hotspots are not true objects; they do not contain any methods of their own. All method calls to modify the segment hotspots must be made to the SAS/GRAPH output object that owns the hotspot.

Segment hotspots can be defined only for static GRSEG entries because segment hotspots are tightly coupled to the position of the graphic segment within the stream of segments stored in the GRSEG entry. For this reason, hotspots are removed when the GRSEG entry changes.

**PARENT:** SASHELP.FSP.WIDGET.CLASS
**CLASS:** SASHELP.FSP.SHOTSPOT.CLASS

---

**How to Define a Segment Hotspot**

To define a segment hotspot on your SAS/GRAPH output:

1. Position the cursor anywhere in the SAS/GRAPH output and from the pop-up menu select **Turn Hotspot Mode ON**.
2. Select the area on the output for which to create the hotspot. The hotspotted area is highlighted in the output.
3. Position the cursor on the hotspotted area and from the pop-up menu select **Hotspot Attributes**. The Segment Hotspot Attributes window opens.
4. Use the Segment Hotspot Attributes window to define the attributes for the hotspot.

**Segment Hotspot Attributes Window**

Sets the attributes of the hotspot and defines what happens when the hotspot is selected.

**Name**

specifies the hotspot's name. By default, the hotspot is named HOTn, where n is the nth hotspot of any class in the frame. Segment hotspots in the examples are named HOT1 or HOT2.

**Style**

specifies the highlight color and assigns the display attribute to apply to the hotspot.

**Display hotspot**

makes the hotspot visible when the application is executed.

**Additional Attributes**

accesses secondary windows in which you can modify additional attributes. All windows accessed through Additional Attributes are described in the Widget class.
Methods

Methods specific to the SAS/GRAPH Output class are described here. Inherited methods are described in the Object class and the Widget class.

Dictionary

_addHotspot

Adds a segment hotspot with the specified attributes to the graphics output

Syntax

CALL NOTIFY (graph-output-name, '_addHotspot', list-id);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list-id</td>
<td>N</td>
<td>specifies the identifier for an SCL list containing the hotspot attributes described in the following table:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>List item</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLOR</td>
<td>C</td>
<td>the color of the hotspot</td>
</tr>
<tr>
<td>FILLTYPE</td>
<td>N</td>
<td>the fill style of the highlight:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 empty (outline only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 filled</td>
</tr>
<tr>
<td>NAME</td>
<td>C</td>
<td>the name of the segment hotspot</td>
</tr>
<tr>
<td>SPOTID</td>
<td>N</td>
<td>the number of the graph segment where the hotspot is located</td>
</tr>
</tbody>
</table>

Details

- The list passed to the _addHotspot method must be explicitly created and filled using SCL list functions in the SCL program.
- _addHotspot can dynamically add a hotspot to the output object at run time. The hotspot is visible when it is added.
- To find the number of a graph segment, invoke the _getInfo method and retrieve the value returned in the SPOTID list item. You can use this number to position the hotspot by passing the value to the SPOTID item in the SCL list used by the _addHotspot method.
Example

This example creates an SCL list ATTRLIST and fills it with the attributes of a blue outline hotspot named HOT2, which is to be assigned to graph segment 5. The _addHotspot method adds the hotspot to GRAPH1.

```
attrlist = makelist();
setnitemc(attrlist, 'hot2', 'name');
setnitemc(attrlist, 'blue', 'color');
setnitemn(attrlist, 1, 'filltype');
setnitemn(attrlist, 5, 'spotid');
call notify('graph1', '_add_hotspot_',
    attrlist);
```

__clear

Removes the GRSEG entry that is displayed and removes any associated hotspots

**Syntax**

CALL NOTIFY (graph-output-name, '_clear');

**Details**

Although you can also clear the GRSEG entry by assigning a blank value to the SCL variable associated with the SAS/GRAPH output, the variable is not directly available when you call an SCL method or when you use CALL DISPLAY to call another frame. Therefore, it is better to use the _clear method to remove the GRSEG entry.

__deleteHotspot

Removes the specified hotspot from the graphics output

**Syntax**

CALL NOTIFY (graph-output-name, '_deleteHotspot', hotspot-name);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hotspot-name</td>
<td>C</td>
<td>specifies the name of the segment hotspot to remove</td>
</tr>
</tbody>
</table>

**Details**

The _deleteHotspot method can dynamically remove hotspots from a SAS/GRAPH output object at run time.
_edit

Invokes the SAS/GRAPH graphics editor on the GRSEG entry that is displayed

Syntax
CALL NOTIFY (graph-output-name, '_edit');

Details
The SAS/GRAPH graphics editor enables you to edit the GRSEG entry interactively. Any changes will be visible in the region upon return from the editor. For information on using the graphics editor, see SAS/GRAPH Software: Graphics Editor, Version 6, First Edition. All other program execution is suspended until you exit from the graphics editor.

_getContort

Returns the setting indicating whether the graphics output is automatically resized to fit the containing region

Syntax
CALL NOTIFY (graph-output-name, '_getContort', resize);

Argument | Type | Description
--- | --- | ---
resize | C | returns a value indicating whether resizing is on or off:

'N' off; the graph is scaled to fit the region containing the graph

'Y' on; the graph is scaled based on the current magnification factor.

See Also
_getMagnification

_getHotspotAttr

Returns the highlighting style of a specified hotspot
**Syntax**

CALL NOTIFY (graph-output-name, '_getHotspotAttr', hotspot-name, attr);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hotspot-name</td>
<td>C</td>
<td>specifies the name of the segment hotspot</td>
</tr>
<tr>
<td>attr</td>
<td>N</td>
<td>returns the fill style of the highlight:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 the hotspot outlines the graph segment (empty)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 the hotspot totally replaces the graph segment (filled)</td>
</tr>
</tbody>
</table>

**_getHotspotColor**

Returns the color of a specified segment hotspot

**Syntax**

CALL NOTIFY (graph-output-name, '_getHotspotColor', hotspot-name, color);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hotspot-name</td>
<td>C</td>
<td>specifies the name of the segment hotspot</td>
</tr>
<tr>
<td>color</td>
<td>C</td>
<td>returns the color of the segment hotspot</td>
</tr>
</tbody>
</table>

**_getInfo**

Returns descriptive information about the last point selected on the graphics output

**Syntax**

CALL NOTIFY (graph-output-name, '_getInfo', list-id);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list-id</td>
<td>N</td>
<td>returns the identifier of the SCL list containing information about the last point selected on the graphics output. The SCL list contains the following items:</td>
</tr>
</tbody>
</table>
List Item | Type | Description
--- | --- | ---
SPOT | N | a list defining the hotspot, if the selected point is a hotspot; if the point is not a hotspot, the value of SPOT is 0
SPOTID | N | the number of the graph segment within the graph, beginning with 1. If no segment has been selected, the value of SPOTID is 0.
SPOTTYPE | N | a number identifying the type of graph segment selected:
TEXT | C | the text of the segment, if the selected point contains text
X | N | the X coordinate of the selection, in pixels
Y | N | the Y coordinate of the selection, in pixels

Example

The following example displays a picture called ARROWS in the region named ARROW. The picture consists of four segments representing four arrows. No hotspots are defined. The _getInfo method returns information about where in the picture a click occurred. That information is used to start a different application, depending on which arrow the user clicked.

```plaintext
length info 8;

INIT:
    arrow='sashelp.eisgrph.arrows.grseg';
    return;

ARROW:
    if (_status_='$') then return;
    call notify('arrow','_get_info_',info);
    segment=getnitemn(info,'spotid');

    /* check to see if we really clicked on an arrow */
    if (segment=0) then return;

    /* run the selected application */
    select (segment);
    when (1) call display('north.frame');
    when (2) call display('west.frame');
    when (3) call display('east.frame');
    when (4) call display('south.frame');
    end;
    return;
```

_getMagnification

Returns the magnification factor at which the graphics output is drawn
Syntax

CALL NOTIFY (graph-output-name, '_getMagnification', factor);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>factor</td>
<td>N</td>
<td>returns the percentage by which the graph is magnified</td>
</tr>
</tbody>
</table>

Details

The magnification factor has no effect if the graph is resized to fit the region with the _setContort method.

_getValue

Returns either the identifier of an SCL list containing descriptive information about the specified segment hotspot or the name of the current GRSEG entry

Syntax

CALL NOTIFY (graph-output-name, '_getValue', name, hot-list-id);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>C</td>
<td>specifies the name of the segment hotspot if the hot-list-id variable is included; otherwise, returns the name of the current GRSEG entry</td>
</tr>
<tr>
<td>hot-list-id</td>
<td>N</td>
<td>returns the identifier of an SCL list containing information about the currently selected hotspot. The SCL list contains the following items:</td>
</tr>
</tbody>
</table>

List Item | Type | Description                                                                 |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>'SPOTID'</td>
<td>N</td>
<td>the number of the graph segment within the graph, beginning with 1. If no segment has been selected, the value of SPOTID is 0.</td>
</tr>
<tr>
<td>'SPOTTYPE'</td>
<td>N</td>
<td>a number identifying the type of graph segment containing the hotspot; for a description of the values, see _getInfo in this class.</td>
</tr>
<tr>
<td>'TEXT'</td>
<td>C</td>
<td>the text of the selected segment</td>
</tr>
</tbody>
</table>
List Item | Type | Description
---|---|---
'X' | N | the X coordinate of the selection, in pixels
'Y' | N | the Y coordinate of the selection, in pixels

Details
The name argument must refer to a segment hotspot and not to a regular hotspot. The _getValue method automatically creates the list returned in hot-list-id. Do not attempt to pass an already existing list to this method.

_hideHotspots

Removes highlighting from one or more segment hotspots

Syntax
CALL NOTIFY (graph-output-name, '_hideHotspots', hotspot-name-1..., hotspot-name-n);

Argument | Type | Description
---|---|---
hotspot-name | C | specifies the name of a segment hotspot to unhighlight

Details
When a hotspot's highlighting is removed, the highlighted area reverts to the object's color, and the hotspot outline is removed from the display.

If multiple hotspot names are supplied, only highlighting for the specified hotspots is removed.

See Also
_hideShowHotspots

_hscroll

Scrolls graphics output horizontally

Syntax
CALL NOTIFY (graph-output-name, '_hscroll', <unit, num-units>);
<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td></td>
<td>specifies the scrolling unit:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'MAX'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'PAGE'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'HALF'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'COLUMN'</td>
</tr>
<tr>
<td>num-units</td>
<td>N</td>
<td>specifies the number of units to scroll. To scroll right, specify a positive number; to scroll left, specify a negative number. The default is 1</td>
</tr>
</tbody>
</table>

**Details**

The _hscroll method is useful when the graphics output is not scaled to fit the region and a large graph, or multiple graphs, may be partially hidden. You can use the _hscroll method to scroll graphics output whether or not the scroll bars are turned on. This method is useful for switching between multiple pictures rendered into the same GRSEG entry. For example, if you use the GREPLAY procedure to create a template containing four graphs and you bring the templated graph into the SAS/GRAPH output object, use _hscroll to scroll from one graph to another.

**Example**

This example shows different ways of scrolling a graph horizontally:

```sas
/* scroll all the way right */
call notify('graph1', '_hscroll_', 'max', 1);

/* scroll left one page */
call notify('graph1', '_hscroll_', 'page', -1);

/* scroll right one page */
call notify('graph1', '_hscroll_', 'page', 1);

/* scroll left one column */
call notify('graph1', '_hscroll_', 'column', -1);

/* scroll right one column */
call notify('graph1', '_hscroll_', 'column', 1);
```

**_setContort**

Turns on or off automatic resizing of the graphics output to fit the containing region
Syntax
CALL NOTIFY (graph-output-name, '_setContort', resize);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resize</td>
<td>C</td>
<td>specifies the setting:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'Y' turns resizing on; the graph is scaled to fit the region</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'N' turns resizing off; the graph is scaled based on its natural size and the current magnification factor.</td>
</tr>
</tbody>
</table>

See Also
_setMagnification

_setGraph
Assigns the graphics output (GRSEG entry) to display in a SAS/GRAPH output object

Syntax
CALL NOTIFY (graph-output-name, '_setGraph', entry-name);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>entry-name</td>
<td>C</td>
<td>specifies the two- or four-level name of the GRSEG entry</td>
</tr>
</tbody>
</table>

Details
Instead of using the _setGraph method, you could name the GRSEG entry by assigning a value to the variable associated with the object. However, because the value of the variable is not available within a method or another frame, the _setGraph method is the preferred way to assign the graph.

Note: The _setGraph method removes any segment hotspots that are previously defined for the SAS/GRAPH output object in which the GRSEG entry is displayed.

Example
_setGraph displays the graph named SASHELP.EISGRPH.ARROWS.GRSEG in the region named ARROW:

call notify('arrow', '_set_graph_',
            'sashelp.eisgrph.arrows.grseg');
_setHotspotAttr

Assigns a fill style to display in a hotspot when it is selected

Syntax
CALL NOTIFY (graph-output-name, '_setHotspotAttr', hotspot-name, attr);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hotspot-name</td>
<td>C</td>
<td>specifies the name of the segment hotspot</td>
</tr>
<tr>
<td>attr</td>
<td>N</td>
<td>specifies the fill style of the highlight:</td>
</tr>
</tbody>
</table>

Example
_setHotspotAttr draws an outline around the graph segment when the hotspot HOT1 is selected:

call notify('graph1', '_set_hotspot_attr', 'hot1', 1);

_setHotspotColor

Assigns a color to display in an active hotspot

Syntax
CALL NOTIFY (graph-output-name, '_setHotspotColor', hotspot-name, color);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hotspot-name</td>
<td>C</td>
<td>specifies the name of the hotspot</td>
</tr>
<tr>
<td>color</td>
<td>C</td>
<td>specifies a highlight color</td>
</tr>
</tbody>
</table>

_setMagnification

Sets the magnification factor with which the graphics output is drawn

Syntax
CALL NOTIFY (graph-output-name, '_setMagnification', factor);
### factor

N specifies the percentage by which to magnify the graph. The default is 100 percent.

#### Details

A value of 200.0 doubles the size of the graphics output horizontally and vertically. The aspect ratio of the graph is preserved. Note that the magnification factor is not in effect when the graph is contorted (see `_setCONTORT`). Also, the magnification factor affects the natural size of the graph, not the size of the region containing the graph.

#### See Also

- `_setContort`

### `_showHotspots`

Highlights or unhighlights one or more segment hotspots

#### Syntax

```plaintext
CALL NOTIFY (graph-output-name, '_showHotspots', action<, hotspot-name-1<..., hotspot-name-n>>);
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
<td>C</td>
<td>specifies the action to apply:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘OFF’ unhighlights the hotspot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘ON’ highlights the hotspot (the default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘TOGGLE’ reverses the current highlighting status of the hotspot (the default)</td>
</tr>
<tr>
<td>hotspot-name</td>
<td>C</td>
<td>specifies the name of the affected hotspot</td>
</tr>
</tbody>
</table>

#### Details

`_showHotspots` turns segment highlights on or off for some or all of the hotspots in the graph. The segments contain the color and fill style specified in the Hotspot Attributes window. Use this method to show selectable areas of SAS/GRAPH output.

#### Examples

In these examples, `_showHotspots` changes the highlight setting of one or more hotspots using the current color and fill style defined for each segment hotspot:

```plaintext
/* reverse the current highlight setting for all hotspots */
```
call notify('graph1','_show_hotspots_');

    /* highlight hotspots HOT2 and HOT3 */
    call notify('graph2','_show_hotspots_','on','hot2','hot3');

    /* turn off highlight for all hotspots */
    call notify('graph2','_show_hotspots_','off');

    /* reverse the current highlight setting for HOT2 */
    call notify('graph2','_show_hotspots_','toggle','hot2');

_vscroll

Scrolls vertically graphics output that is larger than the containing region

Syntax

CALL NOTIFY (graph-output-name '_vscroll', <units<, num-units>>);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>units</td>
<td>C</td>
<td>specifies the scrolling unit: 'MAX' 'PAGE' 'HALF' 'ROW'</td>
</tr>
<tr>
<td>num-units</td>
<td>N</td>
<td>specifies the number of units to scroll. To scroll down, specify a positive number; to scroll up, specify a negative number.</td>
</tr>
</tbody>
</table>

Details

You can scroll graphics output with the _vscroll method whether or not the scroll bars are turned on. This method is useful for switching between multiple pictures rendered into the same GRSEG entry.