Overview

The Block class enables you to create blocks, which are rectangular, text-based objects with three components. Blocks are useful for representing menu choices, or for representing icons on systems that do not support icons.

Blocks are composed of three parts: the rectangular area that contains the label, the label itself, and a shadow that makes the block appear three dimensional.

Parent:

sashelp.fsp.widget.class

Class:

sashelp.fsp.Block.class

Using the Block Class

Assume you want an entry called TENNIS.FRAME to execute when a block, BLOCK1, is selected. This requires the following steps:

1. In the Value on Selection window, enter `tennis.frame` in the Value field.
2. Include the following code in your SCL program:

   ```sas
   BLOCK1:
   call display(block1);
   return;
   ```

Alternatively, you can accomplish the same thing by accessing the Command Processing window and assigning `afappl c=sasuser.test.tennis.frame` to the Execute SAS commands on selection field. In this case, the SCL code shown above is not needed. When the block is selected, SASUSER.TEST.TENNIS.FRAME executes.
Methods

Methods specific to the Block class are described here. Inherited methods are described in the Object class and the Widget class.

Dictionary

_getLabel

Returns the label assigned to a block

Syntax
CALL NOTIFY (block-name, '_getLabel', label);

Argument | Type | Description
---|---|---
lable | C | returns the label for the block

_getText

Returns the character value assigned to a block

Value: Inherited from Widget

Syntax
CALL NOTIFY (block-name, '_getText', text-value);

Argument | Type | Description
---|---|---
text-value | C | returns the character value assigned to the block. This value is blank if the block is not selected.

Details

If a block has been assigned a numeric value in the Value on Selection window, the _getText method causes a program halt. Use the _getValue method to return the value of a block that has a numeric value.
Example

If the value `print` was assigned to the block in the Value on Selection window, `_getText` assigns the value `print` to the SCL variable `TEXT`:

```plaintext
length text $ 24;
call notify('block1','_get_text_',text);
if (text='print') then
    _msg_='The report will be printed.';
```

**_getValue**

Returns the numeric value assigned to a block

Inherited: from Widget

**Syntax**

CALL NOTIFY (block-name, '_getValue', block-value);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>block-value</td>
<td>N</td>
<td>returns the numeric value assigned to the block. This value is blank if the block is not selected.</td>
</tr>
</tbody>
</table>

**Details**

If a block has been assigned a character value in the Value on Selection window, the `_getValue` method causes a program halt. Use the `_getText` method to return the value of a block that has a character value.

**Example**

If the value 4 was assigned to the block in the Value on Selection window, `_getValue` returns the value 4 and assigns it to the variable `VALUE`:

```plaintext
call notify('block1','_get_value_',value);
if (value=4) then
    _msg_='You have made reservations for four.';
```

**_setColor**

Assigns colors to a block’s components (block, label, and shadow)

Restrictions: Some systems do not support changing the colors of blocks.

**Syntax**

CALL NOTIFY (block-name, '_setColor', block-color, label-color, shadow-color);
### Argument Type Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>block-color</td>
<td>C</td>
<td>specifies the color of the block</td>
</tr>
<tr>
<td>label-color</td>
<td>C</td>
<td>specifies the color of the label</td>
</tr>
<tr>
<td>shadow-color</td>
<td>C</td>
<td>specifies the color of the shadow</td>
</tr>
</tbody>
</table>

#### Example

In this example, `_setColor` changes the color of the block to green, the label to white, and the shadow to yellow.

```plaintext
call notify('block1', '_set_color_', 'green', 'white', 'yellow');
```

---

### `_setColorGroup`

Assigns a predefined color scheme to a block

**Restrictions:** Some devices do not support changing the background color.

#### Syntax

```plaintext
CALL NOTIFY (block-name, '_setColorGroup', group-number);
```

#### Argument Type Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>group-number</td>
<td>N</td>
<td>specifies the number of the color combination to apply to the block.</td>
</tr>
</tbody>
</table>

#### Details

Only three of the five items available in each color combination affect a block: TEXT colors the block label, ICON colors the block itself, and SHADOW colors the block shadow.

#### Example

`_setColorGroup` uses color group 7 to make the label yellow, the block blue, and the shadow gray.

```plaintext
call notify('block1', '_set_color_group_', 7);
```
_setLabel

Assigns a label to a block

Syntax
CALL NOTIFY (block-name, '_setLabel', label);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>label</td>
<td>C</td>
<td>specifies the label to assign</td>
</tr>
</tbody>
</table>

Details
_setLabel assigns the value specified in label as the new label for the block. If the block does not have a return value, the new label also becomes the new return value.