Overview

The Radio Box class enables you to create a radio box, which is a group of items called buttons from which users can make a single selection. Each button consists of a text label that is preceded by a selection marker. Users can select either the text or the marker to turn on the button. When the button is on, it is highlighted. Radio buttons are always mutually exclusive so that only one button can be selected at any time.

Parent:

sashelp.fsp.widget.class

Class:

sashelp.fsp.RadioBox.class

Using the Radio Box Class

Example 1

This example checks for the active button by testing the variable associated with the radio box name, RBOX1. Each button is assigned a numeric value equal to its button number and links to a program section like the one for STATE1.
Example 2

This example returns the current active button by calling the SCL function ISACTIVE:

```scl
button=isactive('rbox1');
put 'The active button is:' button=
```

Example 3

These examples show some different ways to assign values to a radio box using the Fill Type field in the Attributes window.

Filling a Radio Box with Items from an SCL Array

From the Fill Type field in the Attributes window, select SCL array and enter the name of the array variable, DAYS.

In the SCL code, declare and initialize the array variable:

```scl
array days[5] $9 ('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday');
```
Filling a Radio Box with Items from an SCL List

From the Fill Type field in the Attributes window, select SCL List and enter the name of the SCL list, DAYLIST.

In the SCL code, create the SCL list. This example creates the SCL list DAYLIST from the array, DAYS:

/* declare the array variable, DAYS */
array days[5] $9 ('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday');

INIT:
/* create the SCL list from the array */
daylist = makelist(5);
do i = 1 to dim(days);
   daylist = setitemc(daylist, days[i], i);
end;
return;

Filling a Radio Box with Items from an SCL Text String

From the Fill Type field in the Attributes window, select SCL Text String and enter the name of the SCL variable, DAYSTR.

In the SCL code, assign values to the SCL variable:

INIT:
daystr='Monday Tuesday Wednesday Thursday Friday';
return'

Filling a Radio Box with Items from a LIST Entry in a Catalog

To start this example, create the LIST entry. One way is to issue this command from the catalog directory window:

edit days.list

Or, from the frame, select File --> New and create a new entry, DAYS, of type LIST.

In the LISTATTR window, specify 9 for Length, turn off CAPS and SORT, and select CASE-INSENSITIVE so that the names of the days display in mixed case. Use the END command and enter the names of the days in the LISTVALUES window:

Monday Tuesday Wednesday Thursday Friday

Close the LISTVALUES window to create DAYS.LIST.

From the Fill Type field in the Attributes window, select Catalog list entry and enter DAYS.LIST.
Methods

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

Dictionary

_activate

Turns on a specified radio button, or grays or ungrays it

Syntax
CALL NOTIFY (radio-box-name, '_activate', button);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>button</td>
<td>N</td>
<td>specifies the number of the button to which the method applies:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&gt;0 turns on the specified button</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;0 toggles gray or ungray the specified button</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 no effect</td>
</tr>
</tbody>
</table>

Details

If the button number is positive, the button is turned on and any previously selected button is turned off. If the button number is negative, the button is grayed or ungrayed, depending on its previous state. Users cannot select a grayed button.

See Also

_gray, _ungray, and _setValue in the Widget class

_getLabel

Returns the label assigned to a radio button

Inherited from Widget

Syntax
CALL NOTIFY (radio-box-name, '_getLabel', button, label);
### _getText

Returns the character value assigned to the active button in the radio box

Inherited from Widget

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>button</td>
<td>N</td>
<td>specifies the number of the button</td>
</tr>
<tr>
<td>label</td>
<td>C</td>
<td>returns the label assigned to the button</td>
</tr>
</tbody>
</table>

#### Example

_getLabel returns the label for the third button in RBOX1 and assigns it to the variable TEXT:

```plaintext
define length text $$ 24$

call notify('rbox1','_get_label_','3',text);

_msg_="the label assigned to the third button is : " || text;
```

### _getStation

Returns the number of the radio button that has the specified label

#### Syntax

CALL NOTIFY (radio-box-name, '_getStation', button-text, button);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>button-text</td>
<td>C</td>
<td>specifies the label assigned to the button. This argument is case sensitive.</td>
</tr>
<tr>
<td>button</td>
<td>N</td>
<td>returns the number of the button (if the label does not exist, it contains a 0)</td>
</tr>
</tbody>
</table>

#### Example

_getStation returns the number of the button containing the label Basket and assigns the number to the variable BUTTON:

```plaintext
call notify('rbox1','_get_station_','Basket',button);
```

### _getText

Returns the character value assigned to the active button in the radio box

Inherited from Widget
Syntax
CALL NOTIFY (radio-box-name, '_getText', text-value);

Argument | Type | Description
---|---|---
text-value | C | returns the character value assigned to the active button. If the active button does not have a value, then the button's label is returned.

Details
If the radio box button 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 button that has a numeric value.

Example
If the return value Printing was assigned to the radio box, _getText returns the value Printing and assigns it to the variable TEXT:

```
length text $ 24;
call notify('rbox1', '_get_text_', text);
if (text='Printing') then
  _msg_='You have selected the printing option.';
```

_getValue

Returns the numeric value assigned to the active button in the radio box

Inherited from Widget

Syntax
CALL NOTIFY (radio-box-name, '_getValue', button-value);

Argument | Type | Description
---|---|---
button-value | N | returns the numeric value assigned to the active button. If the active button does not have a value, then the number of the active button is returned.

Details
If a radio box 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 radio box that has a character value.
**Example**

`_getValue` returns the value of the active station in RBOX1 and assigns it to the variable `VALUE`. Each station in RBOX1 represents a different number of guests at a dinner party.

```plaintext
call notify('rbox1','_get_value_',value);
_msg_='You have made reservations for ' || value || ' guests.';
```

---

**_gray**

Grays a single button or an entire radio box

---

**Syntax**

CALL NOTIFY (radio-box-name, '_gray', button-number);

---

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>button-number</td>
<td>N</td>
<td>specifies the number of the button to gray. If you omit button-number, the entire radio box is grayed.</td>
</tr>
</tbody>
</table>

**Details**

Unlike `_gray` in the Widget class, this method has the optional button-number argument.

---

**_isActive**

Returns the number of the currently selected radio button

---

**Syntax**

CALL NOTIFY (radio-box-name, '_isActive', button-number);

---

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>button-number</td>
<td>N</td>
<td>returns the number of the selected button</td>
</tr>
</tbody>
</table>

**Details**

`_isActive` returns the active button number even if the specified button is grayed.
Example

_isActive assigns the number of the selected button in RBOX1 to the variable BUTTON:

```plaintext
call notify('rbox1','_is_active_',button);
_msg_ = 'The active button is: ' || button;
```

_isGray

Reports whether a specified button or the entire radio box is grayed

Syntax

CALL NOTIFY (radio-box-name, '_isGray', is-grayed<, button-number>);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| is-grayed     | N    | returns a value indicating the status of the button(s):
|               |      | 1 the specified button is grayed                 |
|               |      | 0 the specified button is not grayed. If button-number is omitted, no button in the radio box is grayed. |
|               |      | -n all buttons are grayed, where n is the total number of buttons |
|               |      | m m buttons are grayed                           |
| button-number | N    | specifies the button number. If button-number is included, is-grayed returns either 0 or 1. |

Details

Unlike _isGray in the Widget class, this method has the optional button-number argument and may return values of –n and m for the is-grayed argument.

Example

_isGray determines whether the third button in RBOX1 is currently grayed:

```plaintext
call notify('rbox1','_is_gray_',isgray,3);
if isgray=1 then
    _msg_='The third button is currently grayed.';
else
    _msg_='The third button is not currently grayed.';
```
orea BOX Class

_repopulate

Refills a radio box with the current values of the source that populates the radio box

Syntax
CALL NOTIFY (radio-box-name, '_repopulate');

Details
The radio box is refilled based on the value of Fill type in the Radio Box Attributes window.

_issel

Reports whether a specified radio button is currently selected

Syntax
CALL NOTIFY (radio-box-name, '_issel', button-number, status);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>button-number</td>
<td>N</td>
<td>specifies the number of the button</td>
</tr>
<tr>
<td>status</td>
<td>N</td>
<td>returns a value indicating whether the button is currently selected:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 currently selected</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 not currently selected</td>
</tr>
</tbody>
</table>

Details
_issel returns a value even if the specified button is grayed.

Example
_issel determines whether the third button in RBOX1 is currently selected:

    call notify('rbox1','_issel_','_issel_','3',issel);
    if issel=1 then
        _msg_='The third button is currently selected.';
    else
        _msg_='The third button is not currently selected.';
The _update method for radio boxes works the same as the _repopulate method if the radio boxes are not in an extended table. If radio boxes are in an extended table, the _update method works the same as the _refresh method.

Note: The _repopulate method performs some processing that must take place before a radio box is repopulated. Therefore, if you write your own method to customize the processing, you must call the _repopulate method of the parent class as the first command in the method you write. Use the CALL SUPER routine to call the _repopulate method of the parent class:

```scl
POPULATE:
  method;
  call super(_self_,'_repopulate_');
  /* scl code to repopulate radio box */
  endmethod;
```

**Example**

This example shows how the _repopulate method uses the current values of the source to fill the radio box. Assume that the Attributes window specifies that RBOX1 is initially filled with an SCL array named ALPHA.

```scl
/* Create the array that fills the radio box */
array alpha $ ('a' 'b' 'c');

/* Create a second array with the same values */
array base $ ('a' 'b' 'c');

/* Each time the radio box is selected, increment the counter */
/* and append the counter value to the base values. */
/* Copy the new value into ALPHA. */
MAIN:
  counter+1;
  do i = 1 to dim(alpha);
    alpha(i) = base(i) || compress
       (put(counter, 2.));
  end;

  /* Refill RBOX1 with the current values of ALPHA */
  call notify('rbox1', '_repopulate_');
  return;
```

When the window opens, the values in the radio box are a, b, c. After the user clicks on a button once, the MAIN section runs, the SCL array ALPHA is updated, and the radio box is repopulated with the new values of ALPHA: a1, b1, c1.
_setColor

Assigns a color to the radio box

Restrictions  Some systems do not support changing the color of a radio box.

Syntax

CALL NOTIFY (radio-box-name, '_setColor', label-color);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>label-color</td>
<td>C</td>
<td>specifies the color of the labels for each button in the radio box and of the button for the active button.</td>
</tr>
</tbody>
</table>

_setLabel

Assigns a label to a specified radio button

Syntax

CALL NOTIFY (radio-box-name, '_setLabel', button-text, button-number);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>button-text</td>
<td>C</td>
<td>specifies the new label assigned to the radio button</td>
</tr>
<tr>
<td>button-number</td>
<td>N</td>
<td>specifies the button number</td>
</tr>
</tbody>
</table>

Details

_setLabel assigns the value specified in button-text as the new label for the radio box. If the radio box does not have a return value, the new label also becomes the new return value. Return values for the radio buttons can be specified if the radio box has been filled from the Enter Values field in the Attributes window.

Example

_setLabel assigns the label Bucket to the third button in RBOX1:

call notify('rbox1', '_set_label_', 'Bucket', 3);
_ungray

Ungrays a specified button or the entire radio box

Syntax
CALL NOTIFY (radio-box-name, '_ungray'<, button-number>);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>button-number</td>
<td>N</td>
<td>specifies the number of the button to ungray. If you omit button-number, the entire radio box is ungrayed.</td>
</tr>
</tbody>
</table>

Details
Unlike _ungray in the Widget class, this method has the optional button-number argument.