Using the Scroll Bar class, you can create objects associated with a scrollable area indicating that more information is available in a particular direction. Scroll bars enable users to move back and forth through lists, text, or a range of data values.

To move through a list or text with the scroll bars, users move the thumb, the rectangle between the two arrows. The thumb can be moved by:

- clicking on the arrows
- clicking on either side of the thumb
- dragging the thumb toward one of the arrows.

As users move the thumb, the value returned by the scroll bar changes based on the thumb's position. Clicking on either side of the thumb moves one page size in the appropriate direction. Clicking on the arrows at the ends of the scroll bar moves the thumb one increment in the appropriate direction. The maximum value is visually represented on the scroll bar with the thumb in the far right position for a horizontal scroll bar and with the thumb on the bottom for a vertical scroll bar. The minimum value is visually represented on the scroll bar with the thumb in the far left position for a horizontal scroll bar and with the thumb on the top for a vertical scroll bar. A scroll bar is vertical if you size it so that the vertical dimension is longer than the horizontal dimension; it is horizontal if you size it so that the horizontal dimension is longer than the vertical dimension.

Use a scroll bar in your applications when all the information is not fully visible within a window in a particular direction or when the information in a window can grow beyond the window's borders. For example, provide scroll bars for an application in which a text object can grow as the user works with it.

PARENT: SASHELP.FSP.WIDGET.CLASS
CLASS: sashelp.fsp.ScrollBar.class

Methods

Methods specific to the Scroll Bar class are described here. Inherited methods are described in the Object class and the Widget class.
Dictionary

_getInc

Returns the number by which the values in the scroll bar’s range are incremented

Syntax

CALL NOTIFY (scroll-bar-name, '_getInc', increment);

Argument Type Description
increment N returns the size of the scroll bar’s increments

Example

The following example uses a scroll bar to obtain data points between -2.20 to +2.20 in increments of 0.10. A Zoom button, when pressed, halves the current increment value, and an Unzoom button doubles the increment value.

INIT:
    call notify('sbar1','_set_min_',-2.20);
    call notify('sbar1','_set_max_',2.20);
    call notify('sbar1','_set_inc_',0.10);
    sbar1 = 0;
    return;

ZOOM:
    /* halve the current increment */
    call notify('sbar1','_get_inc_',inc);
    call notify('sbar1','_set_inc_',inc/2);
    return;

UNZOOM:
    /* double the current increment */
    call notify('sbar1','_get_inc_',inc);
    call notify('sbar1','_set_inc_',inc*2);
    return;

MAIN:
    /* The numeric text entry field named */
    /* SVALUE shows the numeric value of */
    /* the scroll bar. */
    svalue=sbar1;
    return;
### _getMax

Returns the maximum value in a scroll bar's range

---

**Syntax**

```call notify (scroll-bar-name, '_getMax', max-val);```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>max-val</td>
<td>N</td>
<td>returns the maximum value in the scroll bar's range</td>
</tr>
</tbody>
</table>

**Example**

The following example, which combines a critical success factor and a scroll bar to produce a dynamically changeable CSF display, restricts the maximum value of the CSF to the maximum value in the scroll bar's range:

```
INIT:
    call notify('sbar1','_get_max_',csf);
    sbar1=csf;
    return;

SBAR1:
    csf=sbar1;
    return;
```

### _getMin

Returns the minimum value in a scroll bar's range

---

**Syntax**

```call notify (scroll-bar-name, '_getMin', min-val);```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>min-val</td>
<td>N</td>
<td>returns the minimum value in the scroll bar's range</td>
</tr>
</tbody>
</table>

**Example**

The following example, which combines a critical success factor and a scroll bar to produce a dynamically changeable CSF display, restricts the minimum value of the CSF to the minimum value in the scroll bar's range:
INIT:
    call notify('sbar1','_get_min_',csf);
    sbar1=csf;
    return;

SBAR1:
    csf=sbar1;
    return;

_getSize

Returns the size of a scroll bar's thumb

Syntax

CALL NOTIFY (scroll-bar-name, '_getSize', thumb-size);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>thumb-size</td>
<td>N</td>
<td>returns the size of the scroll bar's thumb in multiples of the increment value</td>
</tr>
</tbody>
</table>

Details

The value returned by the _getSize method is the page size, or the amount of data that is automatically scrolled through when users click either side of the thumb in the scroll bar.

Example

The following example uses a scroll bar to obtain data points within the range of \(-2.20\) to \(+2.20\) in increments of 0.10. A Zoom button, when pressed, halves the current increment value, and an UnZoom button doubles the increment value.

INIT:
    call notify('sbar1','_set_min_','-2.20');
    call notify('sbar1','_set_max_','2.20');
    call notify('sbar1','_set_inc_','0.10');
    call notify('sbar1','_set_size_','5');
    sbar1 = 0;
    return;

ZOOM:
    /* halve the current increment */
    /* doubling the thumb size leaves it unchanged visually */
    call notify('sbar1','_get_inc_','inc');
    call notify('sbar1','_set_inc_','inc/2');
    call notify('sbar1','_get_size_','thumb');
call notify('sbar1','_set_size_',thumb*2);
return;

UNZOOM:
    /* double the current increment */
    /* halving the thumb size leaves */
    /* it unchanged visually */
    call notify('sbar1','_get_inc_',inc);
    call notify('sbar1','_set_inc_',inc*2);
    call notify('sbar1','_get_size_',thumb);
    call notify('sbar1','_set_size_',thumb/2);
return;

_getValue

Returns the value (thumb position) of a scroll bar

Syntax
CALL NOTIFY (scroll-bar-name, '_getValue', scroll-bar-value);

Argument Type Description
scroll-bar-value N returns the thumb position of the scroll bar

Details
_getValue is inherited from the Widget class.

_setInc

Assigns the number by which the values in the scroll bar's range are incremented

Syntax
CALL NOTIFY (scroll-bar-name, '_setInc', increment);

Argument Type Description
increment N specifies the size of the scroll bar's increments

Details
Only data points that are multiples of the increment away from the minimum value are returned by the scroll bar.
The increment value defines the amount of data that is automatically scrolled through when users click the arrows at the ends of the scroll bar.

**Example**

The following example uses a scroll bar to obtain data points within the range of -2.20 to +2.20 in increments of 0.10.

```plaintext
INIT:
   call notify('sbar1','_set_min_','-2.20');
   call notify('sbar1','_set_max_','2.20');
   call notify('sbar1','_set_inc_','0.10');
   sbar1 = 0;
   svalue = sbar1;
return;

MAIN:
   /* The numeric text entry field named */
   /* SVALUE shows the numeric value of */
   /* the scroll bar. */
   svalue=sbar1;
return;
```

**_setMax**

Assigns the maximum value in a scroll bar's range

**Syntax**

```plaintext
CALL NOTIFY (scroll-bar-name, '_setMax', max-val);
```

**Example**

See the example in the _setMin method for an example that uses the _setMax method.

**_setMin**

Assigns the minimum value in a scroll bar's range
Syntax

**CALL NOTIFY** (scroll-bar-name, '_setMin', min-val);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>min-val</td>
<td>N</td>
<td>specifies the minimum value in the scroll bar’s range</td>
</tr>
</tbody>
</table>

Example

The following example, which combines a critical success factor and a scroll bar to produce a dynamically changeable CSF display, restricts the range of the CSF by increasing the minimum value in the scroll bar’s range to 30 and decreasing the maximum value in the scroll bar’s range to 60:

```
INIT:
    csf=30;
    call notify('sbar1','_set_min_',30);
    call notify('sbar1','_set_max_',60);
    sbar1=30;
    return;

SBAR1:
    csf=sbar1;
    return;
```

_setSize

Assigns the width, in increment units, of a scroll bar’s thumb

Syntax

**CALL NOTIFY** (scroll-bar-name, '_setSize', thumb-size);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>thumb-size</td>
<td>N</td>
<td>specifies the width of the scroll bar’s thumb, in increment units</td>
</tr>
</tbody>
</table>

Details

The _setSize method defines the amount of change made to the value of the scroll bar when the user clicks to either the right or left of the thumb.

Example

The following example uses a scroll bar to obtain data points within the range of -2.20 to +2.20 in increments of 0.10. This example defines a page size of 0.25.
INIT:
    call notify('sbar1','_set_min_',-2.20);
call notify('sbar1','_set_max_','2.20');
call notify('sbar1','_set_inc_','0.10');
call notify('sbar1','_set_size_','5');
sbar1=0;
  \  
svalue=sbar1;
return;

/* */
MAIN:
    /* The numeric text entry field named */
    /* SVALUE shows the numeric value of */
    /* the scroll bar. */
    svalue=sbar1;
return;

_setValue

Assigns a value (thumb position) to a scroll bar

Syntax

CALL NOTIFY (scroll-bar-name, '_setValue', scroll-bar-value);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>scroll-bar-value</td>
<td>N</td>
<td>specifies the value (thumb position) to assign to the scroll bar</td>
</tr>
</tbody>
</table>

Details

_setVALUE is inherited from Widget. The _setValue method assigns only values that fall on or between the scroll bar’s minimum and maximum values. For example, if a value is specified that is greater than the scroll bar’s maximum value, the _setValue method assigns the maximum value. However, if the value is set directly (via the SCL variable), the value is not adjusted to the scroll bar’s range until the FRAME entry is refreshed.

Examples

Example 1: Changing the value of the scroll bar

BUTTON:
    do i=1 to 100;
    sbar1=i;
    /* this _set_value_ method produces */
    /* the same results as sbar1=i. */
    /* call notify('sbar1',
        '_set_value_','i'); */
refresh;
end;
return;

Example 2: SCL program behind scroll bar  
Assume you have a FRAME entry that contains a scroll bar with default minimum and maximum values (min=0, max=100), 3 numeric text entry fields, and a push button. The FRAME entry has the following SCL program:

FIELD1:
  sbar1=field1;
  return;

FIELD2:
  call notify('sbar1','_set_value_',field2);
  return;

BUTTON1:
  return;
MAIN:
  call notify('sbar1','_get_value_',field3);
  return;

When 200 is entered in FIELD1, 200 is displayed in FIELD3. When 200 is entered in FIELD2, 100 is displayed in FIELD3. When 200 is entered in FIELD1 and the button (BUTTON1) is selected, 100 is displayed in FIELD3.