Overview

The List Box Control displays a scrollable lists of text items from which users can make one or more selections. List boxes are helpful when users must select from a large and/or frequently updated list of items.

Ancestors:

sashelp.fsp.Object.class
sashelp.fsp.Widget.class
sashelp.classes.Afcontrol.class

Class:

sashelp.classes.Listbox_c.class

Using the List Box Control

Key properties “Key Properties to set” on page 437
List box items “Setting list box items” on page 438
Build-time attributes “Build-time attributes” on page 438

Key Properties to set

selectedItem Specify which of the list items you want selected by default.
attribute “selectedItem” on page 441
(default attribute)

Build-time attributes

Using SCL to specify list box items 438

Attributes 439
Dictionary 439
Methods 443
Events 447
Event Handlers 448
Interfaces 448
You can also query this attribute to find out what the user has selected. For example, the following line of SCL code queries the item that is currently selected:

```scl
choice = listbox1.selectedItem;
```

Note: The following line of code is also valid since the `selectedItem` attribute is the default attribute:

```scl
choice = listbox1;
```

### Setting list box items

You must tell every list box control what items to display. Use one of the following ways to add items to a list box control:

- You can set the `items` attribute in the Properties window. When you click the value cell for the `items` attribute, the Items window appears where you can enter each of the items you want.
- You can populate the list box with items from a list model by setting the value of the `model` attribute in the Properties window to the name of the list model.
- You can also populate the list box with items from a list model by dragging the model from the Components window onto the list box in the frame.
- You can specify the value of the `item` attribute in the list box at run time. See Using SCL to specify list box items for more information.

### Build-time attributes

The following attribute is a build-time attribute, meaning it can only be set at build-time if you choose to use it:

- `selectionMode` on page 442

### Using SCL to specify list box items

You can specify list box items using an SCL program. For example, the following code creates a list and assigns the list to a list box:

```scl
petList = makelist(); rc = insertc(petList, 'Dog', -1); rc = insertc(petList, 'Cat', -1); rc = insertc(petList, 'Bird', -1); listbox1.items = petList;
```
You could just as easily fill a list box with a list of items that already exists in a model that you have instantiated. The following code assigns color list model items to a list box:

```plaintext
listbox1.items = colorlist1.items;
```

### Attributes

Public attributes specified for the List Box Control class are described here. See `sashelp.classes.Afcontrol.class` for inherited attributes.

### Dictionary

#### defaultAttribute

Returns or sets the attribute name used for the object's default attribute

- **Type:** Character
- **Valid Values:** `\sashelp.classes.defaultattributevalues.scl`

#### description

Returns the short description for the object

- **Type:** Character
- **Valid Values:**

#### dragInfo

Returns or sets the information that is transferred from the control when a drag operation occurs

- **Type:** List
- **Editor:** `SASHELP.FSP.DRAGDROPINFOEDITOR.FRAME`
**dropInfo**

Returns or sets the information that defines the data representations that can be dropped on the control

Type: List
Valid Values:
Editor: SASHELP.FSP.DRAGDROPINFOEDITOR.FRAME

**font**

Returns or sets the font applied to the text

Type: List
Valid Values:
Editor: sashelp.classes.fonteditor3.scl

**items**

Returns or sets the items to display

Type: List
Valid Values:
Editor: sashelp.classes.stringListEditor.frame

**rows**

Returns the number of rows in the list

Type: Numeric
Valid Values:

---

**selectedCount**

Returns the number of items selected

Type: Numeric

Valid Values:

---

**selectedIndex**

Returns or sets the index of the selected item

Type: Numeric

Valid Values:

---

**selectedIndexes**

Returns or sets the selected items in the list by index number

Type: List

Valid Values:

---

**selectedItem**

Returns or sets the selected item

Type: Character

Valid Values: \sashelp.classes.itemsValues.scl
**selectedItems**

Returns the selected items

**Type:** List

**Valid Values:**

**selectionMode**

Returns or sets whether to allow multiple selections

**Type:** Character

**Valid Values:** Single Selection, Multiple Selections, Extended Selections

**textColor**

Returns or sets the foreground color applied to the text

**Type:** Character

**Valid Values:** (All SAS-supported color values)

**Editor:** sashelp.classes.colorEditor.frame

**textOrientation**

Returns or sets how the text in the input area is written

**Type:** Character

**Valid Values:** Yes, No
List Box Control

Methods

Methods specified for the List Box Control class are described here. See sashelp.classes.Afcontrol.class for inherited methods.

---

title

Returns or sets the title

Type: Character

Valid Values:

---

titleHeight

Returns the height of the title

Type: Numeric

Valid Values:

---

topRow

Returns or sets the row number displayed as the top row

Type: Numeric

Valid Values:
_deselectAll

Deselects all items in the list

Syntax

objectName_deselectAll();

__find

Finds a string in the list of items

Syntax

objectName_find( startRow, text );
objectName_find( startRow, text, caseSensitive );

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>startRow</td>
<td>Numeric</td>
<td>Update</td>
<td>Specifies the starting row of the search and returns the row index if found</td>
</tr>
<tr>
<td>text</td>
<td>Character</td>
<td>Input</td>
<td>Specifies the text string to search for</td>
</tr>
<tr>
<td>caseSensitive</td>
<td>Character</td>
<td>Input</td>
<td>Specifies whether to perform a case sensitive search</td>
</tr>
</tbody>
</table>

__getText

Returns information about the specified row

Syntax

objectName_getText( row, text );
**List Box Control**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>row</td>
<td>Numeric</td>
<td>Input</td>
<td>Specifies the index of the row to retrieve</td>
</tr>
<tr>
<td>text</td>
<td>Character</td>
<td>Output</td>
<td>Returns the text for the specified row</td>
</tr>
</tbody>
</table>

**_isSelected**

Returns whether the specified row is currently selected

**Syntax**

objectName_isSelected( rowIndex, selected );

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rowIndex</td>
<td>Numeric</td>
<td>Input</td>
<td>Specifies the index of the row to check</td>
</tr>
<tr>
<td>selected</td>
<td>Numeric</td>
<td>Output</td>
<td>Returns the selection status of the specified row: 1 if selected, 0 if not selected</td>
</tr>
</tbody>
</table>

**_onClick**

Invoked when the click event occurs

**Syntax**

objectName_onClick( selectedRow );

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>selectedRow</td>
<td>Numeric</td>
<td>Input</td>
<td>contains the index of the selected row</td>
</tr>
</tbody>
</table>
_onDoubleClick

Invoked when the doubleClick event occurs

Syntax

objectName_onDoubleClick( selectedRow );

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>selectedRow</td>
<td>Numeric</td>
<td>Input</td>
<td>contains the index of the selected row</td>
</tr>
</tbody>
</table>

_onPopup

Invoked when the popup event occurs

Syntax

objectName_onPopup( );

_selectAll

Selects all items in list

Syntax

objectName_selectAll( );
_showPopup

Displays a pop-up menu containing items in a list

Syntax

objectName_showPopup( popupItemList, selectedIndex );

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>popupItemList</td>
<td>List</td>
<td>Input</td>
<td>specifies the items to display in the popup</td>
</tr>
<tr>
<td>selectedIndex</td>
<td>Numeric</td>
<td>Update</td>
<td>specifies the index of the item selected from the menu</td>
</tr>
</tbody>
</table>

Events

Events specified for the List Box Control class are described here. See sashelp.classes.Afcontrol.class for inherited events.

click
  Occurs on a click
doubleClick
  Occurs on a double-click
font changed
  Occurs when the font attribute is changed
items changed
  Occurs when the items attribute is changed
popup
  Occurs on a popup request
rows changed
  Occurs when the rows attribute is changed
selectedIndex changed
  Occurs when the selectedIndex attribute is changed
selectedIndexes changed
  Occurs when the selectedIndexes attribute is changed
Event Handlers

Event handlers specified for the List Box Control class are described here. See sashelp.classes.Afcontrol.class for inherited event handlers.

<table>
<thead>
<tr>
<th>Event</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.click</td>
<td>_onClick</td>
<td>Executes when the click event occurs</td>
</tr>
<tr>
<td>.doubleClick</td>
<td>_onDoubleClick</td>
<td>Executes when the doubleClick event occurs</td>
</tr>
<tr>
<td>.popup</td>
<td>_onPopup</td>
<td>Executes when the popup event occurs</td>
</tr>
</tbody>
</table>

Interfaces

Interfaces specified for the List Box Control class are described here. See sashelp.classes.Afcontrol.class for inherited interfaces.

Required Interfaces
sashelp.classes.staticstringlist.intrface