

# SWBBtnBar Control

Displays a vertical button bar using flat (pop-up) push buttons that can be categorized into visual groups.

## Remarks

The SWBBtnBar control creates a vertical button bar that utilizes graphical item buttons that may or may not contain a caption, and remain flat until the mouse is passed over them. These items are grouped together in one or more groups. At run-time, the user can switch between groups by clicking on the group button at the top of each group.

The control allows design-time or run-time configuration of its properties and customization of the graphical display including options to show captions under each button, set button caption text color, and background color for the entire control. The size of the buttons can also be customized (although all buttons must be the same size). Button states can be changed at run-time to flat (normal/visible), disabled, or invisible.

## [Creating a button bar with SWBBtnBar](#)

## Properties

[BackColor](#)  
[BorderHGap](#)  
[BorderVGap](#)  
[BorderStyle](#)  
[CurrentGroup](#)  
[GroupBtnVGap](#)  
[GroupBtnHeight](#)  
[Groups](#)  
[ImageList](#)  
[Items](#)  
[ItemHeight](#)  
[ItemWidth](#)  
[ItemTextColor](#)  
[ItemCaptionPos](#)  
[ItemVertGap](#)  
[ScrollBtnHGap](#)  
[ScrollBtnVGap](#)  
[ScrollBtnSize](#)  
[ShowCaptions](#)

## Methods

[AddGroup](#)  
[AddItem](#)  
[ChangeGroup](#)

[Clear](#)

[HitTest](#)

[Refresh](#)

[RemoveGroup](#)

[RemoveItem](#)

## **Events**

[GroupClick](#)

[ItemClick](#)

## Creating a Button Bar

To create a button bar with the SWBBtnBar control:

- 1 Place the SWBBtnBar on a form. Set the alignment if desired. In most cases, the alignment will be vbAlignLeft.
- 2 Place an ImageList control on the form. Add images to the imagelist controls for the buttons you will be creating.
- 3 Right-click on the button bar and select Properties from the popup menu.
- 4 On the General property page, select the image list control from the pulldown combo.
- 5 In the General Item Properties group, set the height and width of the items (buttons) in twips. You can also select to show or hide captions below the items by checking or clearing the Show Captions checkbox.
- 6 First, add a group by typing the name of a group in the Caption field within the Groups section of the property page. You must also enter a "key", or unique string value by which this group will be referenced at run-time. Once you have entered both of these values, click Append to add this Group to the button bar. You can now add more groups by typing a new group caption and group key value, and clicking Append (to add the group after the current group), or Insert (to insert the group before the current group).
- 7 Next, create an item (button) in one of the groups. Use the Select Group arrow buttons in the Groups section to select the group you want to add the item to. Then, in the Items section of the property page, type the caption for this item in the Caption field, and a unique key (string) value in the Key field. Every button on the button bar must have a unique key. Then select an image for this item from the image list by using the arrow buttons next to the Image viewer under the Caption and Key fields. When you have selected an image for this item, click Append to add this item to the current group. You can now add additional items by entering the Caption and Key values, and clicking Append (to add the item after the currently selected item) or Insert (to add the item before the currently selected item) to add the item to the current group. To edit an item, or select a new "current" item, use the Select Item arrow buttons in the Items section of the property page.
- 8 When you have added all groups and items you need, click OK to update the control.

## **BackColor Property**

**Type:** OLE\_COLOR

**Access:** Read/Write

### **Description:**

Sets or returns the color of the background of the button bar.

## **BorderHGap Property**

**Type:** Integer

**Access:** Read/Write

### **Description:**

Sets or returns the horizontal gap (in twips) between group buttons and the left and right edges of the control.

## **BorderVGap Property**

**Type:** Integer

**Access:** Read/Write

### **Description:**

Sets or returns the vertical gap (in twips) between group buttons and the top and bottom edges of the control.

## BorderStyle Property

**Type:** Integer

**Access:** Read/Write

### Description:

Sets or returns the the style of the control. By setting this control to bsSunken, the control will take on a 3D appearance. This property controls only the outside border of the control and has no effect on group or item button appearance.

<b>bsFlat</b>	1	Displays a flat control edge.
<b>bsSunken</b>	2	Displays a 3D (sunken) control edge.

## **CurrentGroup Property**

**Type:** String

**Access:** Read-Only

### **Description:**

Returns the key to the currently selected (currently visible) group. This property will return an empty string ("") if no group has been selected yet (although some may have been created). Use the *ChangeGroup* method to force this property to initialize.



## **GroupBtnVGap Property**

**Type:** Integer

**Access:** Read/Write

### **Description:**

Sets or returns the vertical gap (in twips) between group buttons shown in the button bar.

## **GroupBtnHeight Property**

**Type:** Integer

**Access:** Read/Write

### **Description:**

Sets or returns the vertical height (in twips) of the group buttons. To control the width of the group buttons, change adjust the BorderHGap property.

## **Groups Property**

**Type:** Collection of CItem Type

**Access:** Read/Write

### **Description:**

Allows setting or retrieving of properties of the individual Groups on the button bar. Note that the groups within the collection are numbered starting at 1.

[Group Properties](#)

## Group Properties

**Caption**      The text caption that appears within the Group button.

**Key**            The *unique* string key used to identify this Group. Once a group is added, it's key value should not be changed. Changing it may cause the button bar to function improperly.

## **ImageList Property**

**Type:** String

**Access:** Read/Write

### **Description:**

Sets or returns the name of the imagelist control that this button bar will retrieve its item images from. Note that this property is a string that should be assigned the *Name* property of an imagelist control, and cannot be set to reference the control itself.

## **Items Property**

**Type:** Collection of CItem Type

**Access:** Read/Write

### **Description:**

Allows setting or retrieving of properties of the individual Items on the button bar. Note that the items within the collection are numbered starting at 1.

[Item Properties](#)

## Item Properties

State	Setting	Description
<b>isUp</b>	1	The button is visually raised.
<b>isDown</b>	2	The button is visually sunken..
<b>isFlat</b>	3	The button is visually flat.
<b>isDisabled</b>	4	The button is disabled.
<b>isInvisible</b>	5	The button is not shown on the toolbar.

When setting the state, use `isFlat` to change a disabled or invisible button back to visible again.

**Caption**      The text caption that appears below the button's picture.

**Key**            The *unique* string key used to identify this button. Once an item is added, it's key value should not be changed. Changing it may cause the button bar to function improperly.

**ImageIndex**   The integer used to identify the image number within the specified `ImageList` that will be used on this button (1 - n).

**Group**                The key of the group this button belongs to.

## **ItemHeight Property**

**Type:** Integer

**Access:** Read/Write

### **Description:**

Sets or returns the height of the item buttons. All items in all groups must be the same size.



## **ItemWidth Property**

**Type:** Integer

**Access:** Read/Write

### **Description:**

Sets or returns the width of the item buttons. All items in all groups must be the same size.

## **ItemTextColor Property**

**Type:** OLE\_COLOR

**Access:** Read/Write

### **Description:**

Sets or returns the color of the item caption text for all items.

## ItemCaptionPos Property

**Type:** Integer

**Access:** Read/Write

### Description:

Sets or returns the position of the item button captions. They can be by setting this property to one of the constants below:

<b>icBottom</b>	1	Places the captions below the item buttons
<b>icRight</b>	2	Places the captions to the right of the item buttons

## **ItemVertGap Property**

**Type:** Integer

**Access:** Read/Write

### **Description:**

Sets or returns the vertical gap (in twips) between item buttons shown in the button bar.

## **ScrollBtnHGap Property**

**Type:** Integer

**Access:** Read/Write

### **Description:**

Sets or returns the horizontal gap (in twips) between the scroll buttons and the right edge of the control.

## **ScrollBtnVGap Property**

**Type:** Integer

**Access:** Read/Write

**Description:**

Sets or returns the vertical gap (in twips) between the scroll buttons and the nearest group button.

## **ScrollBtnSize Property**

**Type:** Integer

**Access:** Read/Write

### **Description:**

Sets or returns the vertical and horizontal size (in twips) of the scroll buttons. These buttons are always square. That is, this measurement always represents both the vertical and horizontal size of the scroll button.

## **ShowCaptions Property**

**Type:** Boolean

**Access:** Read/Write

### **Description:**

Sets or returns the visibility of the text captions that appear below each item's image.



## AddGroup Method

`SWBBtnBar1.AddGroup    Caption as String, Key as String,    Optional Position as Integer`

Adds a group to the button bar. Once a group is added to the button bar, it becomes a member of the Groups collection, through which these properties can be retrieved or updated (except for Key, as noted).

### Arguments:

**Caption**        The caption that will appear below the item's image.

**Key**            The unique string that can be used to identify this item. The default is the empty string (""). Once a group is added, its key value should not be changed. Changing it may cause the button bar to function improperly.

**Position**        (Optional) The ordinal position of this item in the button bar. The default is last (bottom-most).

## AddItem Method

```
SWBBtnBar1.AddButton GroupKey as String, Caption as String, Key as  
String, ImageIndex as Integer, Optional Position as Integer
```

Adds an item to the button bar. Once an item is added to the button bar, it becomes a member of the Items collection, through which these properties can be retrieved or updated (except for Key, as noted).

### Arguments:

**Group**                      The group that this item will appear in.

**Caption**                  The caption that will appear below the item's image.

**Key**                      The unique string that can be used to identify this item. The default is the empty string (""). Once an item is added, its key value should not be changed. Changing it may cause the button bar to function improperly.

**ImageIndex**              The number of the image within the ImageList that will be used for this item.

**Position**              (Optional) The ordinal position of this item in the button bar. The default is last (bottom-most).

## ChangeGroup Method

`SWBBtnBar.ChangeGroup    GroupKey as String`

This method changes the current group to the group specified by GroupKey, and displays that group in the button bar.

## Clear Method

`SWBBtnBar.Clear`

This method removes *all* groups and their associated items from the button bar.

## HitTest Method

SWBBtnBar1.HitTest X as Single, Y as Single

Returns the *Key* property of the group or item button positioned at the coordinates specified by X and Y.

This method is most often called from within a MouseUp or MouseDown event, where the X and Y parameters are passed to the event, identifying the location of the mouse at the time the mouse button was clicked. This method can be called during one of these events to identify the object under the mouse cursor when the mouse button was clicked.

This method returns an empty string ("" ) if coordinates specified are not within a group button or an item button.

### Arguments:

- X**     The X coordinate of the mouse cursor.
- Y**     The Y coordinate of the mouse cursor.

## Refresh Method

`SWBBtnBar.Refresh`

This method causes the entire button bar image to be cleared and then redrawn. This method will not change the current group.

## RemoveGroup Method

`SWBBtnBar1.RemoveGroup GroupKey as String`

Removes the specified group and all of its associated items from the button bar.

### Arguments:

**GroupKey** The key of the group to be removed.

## RemoveItem Method

```
SWBBtnBar1.RemoveItem ItemKey as String
```

Removes the specified item from the button bar.

### Arguments:

**ItemKey**      The key of the item to be removed.



## **GroupClick Event**

`GroupClick(GroupKey as String)`

Occurs whenever the user changes groups by clicking on a group button.

### **Arguments:**

**GroupKey**                The key of the group that was clicked (the newly selected group).

## **ItemClick Event**

`ItemClick(ItemKey as String)`

Occurs whenever the user selects an item by clicking on an item button.

### **Arguments:**

**ItemKey**      The key of the item button that was clicked.



