

ItemImage

Image to appear in an item.

Usage

```
Set_Property(OLECtrlEntID, "OLE.ItemImage[group; item]", Value)
```

Values

Value can be any positive or negative whole number:

Default: 0

Indices

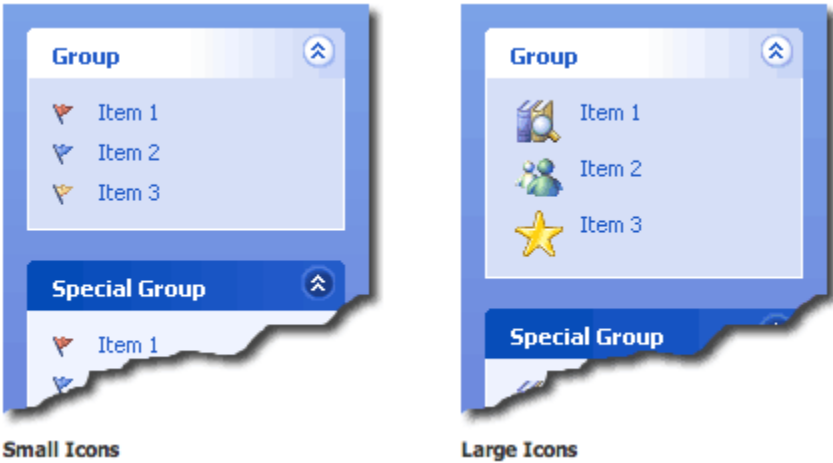
Index	Description
group	Index to an existing group
item	Index to an existing item within the group

Remarks

The ItemImage property sets an items image. The value of this property is an index into the item image list specified in the [ImageList](#) property. If there is no item image list, then this property is ignored. Set this property to zero to display no image.

The image is aligned according to the [Layout](#) property. If the layout is "Images", then the image appear by itself. The "ImagesWithText" layout displays the image on the left with the text to the right. A layout of "ImagesWithTextBelow" will center the image with the text centered below it.

The size of the image will determine the size of the item as seen below:



Dynamic Index

Every indexed property, including this one, supports dynamic indexing. There are three ways to index a ShortcutBar property: single, ranged, and all. Single indexing operates on one element at a time while ranged and all indexing operates on several elements at once, reducing the amount of code required.

Single

Single indexing is simple. Just pass a single integer to index an existing element.

Ranged

Ranged indexing operates on one or more properties. Instead of passing a single integer, you pass two integers delimited by a dash '-' character. For instance, to operate on elements 3, 4, 5 and 6 you pass '3-6'.

All

All indexing operates on all existing elements at once. Instead of passing a numerical index, you pass the word "All".

Setting multiple properties

Use ranged or all indexing to set the same value for multiple properties at once. This is particularly useful for boolean properties, such as enabling/disabling settings for several properties at once. It can also be useful for initializing or resetting a control.

Getting multiple properties

Getting multiple properties is tricky because it is possible to be getting multiple properties with different values. The ShortcutBar control resolves this by using a same/indeterminate paradigm. If all values of multiple properties are the same, then that value is returned. If there is at least one different value, then an indeterminate value is returned. In most cases the returned value is "", except for boolean properties. They return a value of 2, which is the value of OI Check Boxes in their indeterminate states.

Example

```
// Set an image for an item
Set_Property(@Window:".OLE_SHORTCUT", "OLE.ItemImage[1; 1]", 2)

// Remove all images from all items
Set_Property(@Window:".OLE_SHORTCUT", "OLE.ItemImage[All; All]", 0)
```

See Also

[GroupImage](#), [Layout](#)