

Listbox Control in VB.NET

Example of Listbox Control in VB.NET

<http://www.thecodegallery.com/VBNET/ListboxExample.php>

Listbox Control is a rectangular box type structure which allows user to display list of items from which user can select one or more items at a time. It also allows user to sort items of the list in particular order. User can add new items, remove selected or all items from Listbox at run time or design time.



Properties of Listbox Control in VB.NET

MultiColumn It is used to specify whether Listbox supports multiple columns or not. It has Boolean value. Default value is false.

ColumnWidth It is used to specify width of each column in MultiColumn Listbox.

Items It represents collection of items contained in Listbox control.

Sorted It is used to specify whether items of ListBox are sorted in alphabetical order or not. It has Boolean value. Default value is false.

SelectionMode It is used to get or set SelectionMode of ListBox. It determines how user can select the items of ListBox. It can have one of the following four options:

(1) None: No Selection is allowed

(2) One: User can select only one item at a time.

(3) MultiSimple: User can select or deselect item just by mouse click or pressing spacebar.

(4) MultiExtended: User can select or deselect items by holding Ctrl key and mouse click. User can also select or deselect items by pressing Shift key and mouse click or arrow key. Default value is One.

ScrollAlwaysVisible It is used to specify whether Scroll Bars is always associated with ListBox or Not regardless of number of items present in ListBox. It has Boolean value. Default value is false.

HorizontalExtent It is used to get or set width in pixel, by which a ListBox can be scrolled horizontally. It works only when Horizontal Scrollbar Property is set to true.

HorizontalScrollbar It is used to specify whether ListBox can have Horizontal Scroll Bar or Not, If Number of Items in ListBox are not accommodated in specified width. It has Boolean value. Default value is False.

SelectedIndex It is used to get or set zero based index of the item currently selected in ListBox.

SelectedItem It is used to get or set item currently selected in ListBox.

SelectedItems It is used to get collection of multiple items currently selected in

ListBox.

SelectedIndices It is used to get collection of zero based indexes of all items currently selected in ListBox.

Enable It is used to specify whether ListBox Control is enabled or not at runtime. It has Boolean value. Default value is true.

Visible It is used to specify whether ListBox Control is visible or not at runtime. It has Boolean value. Default value is true.

TabStop It is used to specify whether user can use TAB key to set focus on ListBox Control or not. It has Boolean value. Default value is true.

Methods of Listbox Control in VB.NET

Method **Purpose**

ClearSelected It is used to unselect all the items that are currently selected in ListBox.

FindString It is used to find first occurrences of an item in the ListBox that partially match with string specified as an argument. If an item is found then it returns zero based index of that item, otherwise it returns -1. The search performed by this method is case insensitive.

FindStringExact It is used to find first occurrences of an item in the ListBox that exactly match with string specified as an argument. If an item is found then it returns zero based index of that item, otherwise it returns -1. The search performed by this method is case insensitive.

GetSelected It is used to determine whether an item whose index is passed as an

argument is selected or not. It returns Boolean value.

SetSelected It is used to select or deselect an item whose index is passed as an argument.

Example:

`ListBox1.SetSelected (1, true)` will select second item of ListBox.

Events of Listbox Control in VB.NET

SelectedIndexChanged It is the default event of ListBox Control. It fires each time a selected Item in the ListBox is changed.

Methods of Item Collection of ListBox

Add It is used to add an item at the end of ListBox collection.

Example:

`ListBox1.Items.Add(Item)`

Insert It is used to insert an item at specific index position.

Example:

`ListBox1.Items.Insert(Index, Item)`

Clear It is used to remove all items from ListBox collection.

Example:

`ListBox1.Items.Clear()`

CopyTo It is used to copy all the Items of ListBox into an array that is passed as an argument. It also accepts an index position as an argument to specify from which index position within array copy will start.

Example:

```
ListBox1.Items.CopyTo(ArrayName, StartIndex)
```

Remove It is used to remove first occurrence of an item that matches with value passed as an argument.

Example:

```
ListBox1.Items.Remove(Value)
```

RemoveAt It is used to remove an item from specific index position in ListBox.

Example:

```
ListBox1.RemoveAt (1) will remove second Item of ListBox.
```

Contains It is used to determine whether a ListBox's Item Collection contains an item that is passed as an argument. It returns Boolean value. If an item is found then it returns true otherwise false.

Example:

```
If ListBox1.Items.Contains (Value) Then
```

```
MsgBox("Item is Found")
```

```
Else
```

```
MsgBox("Item is Not Found")
```

```
End If
```

Properties of Item Collection of ListBox

Count It is used to get number of items available in ListBox's Item Collection.

