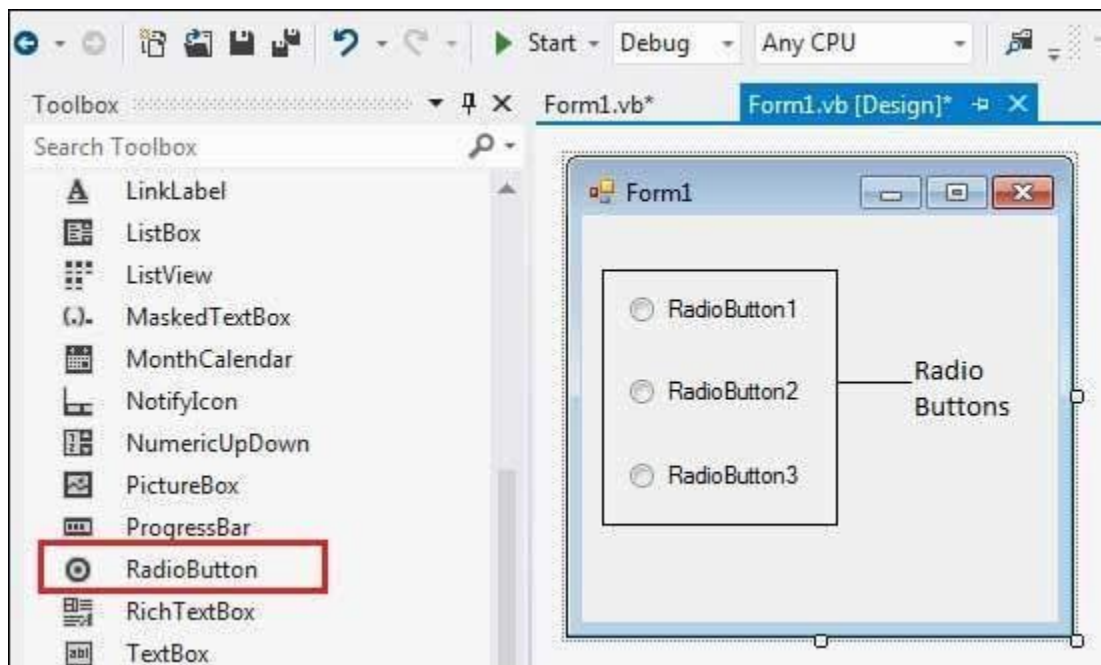# VB.Net - RadioButton Control

The RadioButton control is used to provide a set of mutually exclusive options. The user can select one radio button in a group. If you need to place more than one group of radio buttons in the same form, you should place them in different container controls like a GroupBox control.

Let's create three radio buttons by dragging RadioButton controls from the Toolbox and dropping on the form.

The *Checked* property of the radio button is used to set the state of a radio button. You can display text, image or both on radio button control. You can also change the appearance of the radio button control by using the *Appearance* property.

Properties of the RadioButton Control

The following are some of the commonly used properties of the RadioButton control −

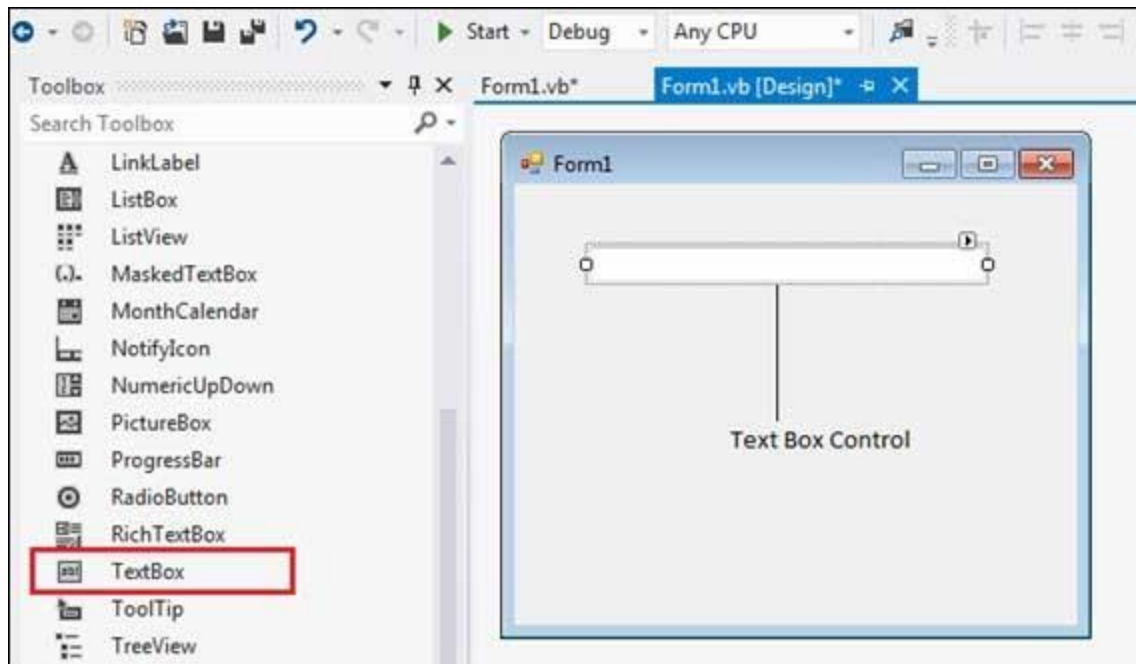| Sr.No.	Property & Description
1	**Appearance** Gets or sets a value determining the appearance of the radio button.
2	**AutoCheck** Gets or sets a value indicating whether the Checked value and the appearance of the control automatically change when the control is clicked.
3	**CheckAlign** Gets or sets the location of the check box portion of the radio button.
4	**Checked** Gets or sets a value indicating whether the control is checked.
5	**Text** Gets or sets the caption for a radio button.
6	**TabStop** Gets or sets a value indicating whether a user can give focus to the RadioButton control using the TAB key.

Methods of the RadioButton Control

The following are some of the commonly used methods of the RadioButton control −

Sr.No.	Method Name & Description
1	**PerformClick** Generates a Click event for the control, simulating a click by a user

VB.Net - TextBox Control

Text box controls allow entering text on a form at runtime. By default, it takes a single line of text, however, you can make it accept multiple texts and even add scroll bars to it.

Let's create a text box by dragging a Text Box control from the Toolbox and dropping it on the form.

The Properties of the TextBox Control

The following are some of the commonly used properties of the TextBox control −

Sr.No.	Property & Description
1	**AcceptsReturn** Gets or sets a value indicating whether pressing ENTER in a multiline TextBox control creates a new line of text in the control or activates the default button for the form.
2	**AutoCompleteCustomSource** Gets or sets a custom System.Collections.Specialized.StringCollection to use when the AutoCompleteSourceproperty is set to CustomSource.

3	**AutoCompleteMode**
	Gets or sets an option that controls how automatic completion works for the TextBox.
4	**AutoCompleteSource**
	Gets or sets a value specifying the source of complete strings used for automatic completion.
5	**CharacterCasing**
	Gets or sets whether the TextBox control modifies the case of characters as they are typed.
6	**Font**
	Gets or sets the font of the text displayed by the control.
7	**FontHeight**
	Gets or sets the height of the font of the control.
8	**ForeColor**
	Gets or sets the foreground color of the control.
9	**Lines**
	Gets or sets the lines of text in a text box control.
10	**Multiline**
	Gets or sets a value indicating whether this is a multiline TextBox control.

11	**PasswordChar**
	Gets or sets the character used to mask characters of a password in a single-line TextBox control.
12	**ReadOnly**
	Gets or sets a value indicating whether text in the text box is read-only.
13	**ScrollBars**
	Gets or sets which scroll bars should appear in a multiline TextBox control. This property has values − - None - Horizontal - Vertical - Both
14	**TabIndex**
	Gets or sets the tab order of the control within its container.
15	**Text**
	Gets or sets the current text in the TextBox.
16	**TextAlign**
	Gets or sets how text is aligned in a TextBox control. This property has values − Left

	• Right • Center
17	**TextLength** Gets the length of text in the control.
18	**WordWrap** Indicates whether a multiline text box control automatically wraps words to the beginning of the next line when necessary.

The Methods of the TextBox Control

The following are some of the commonly used methods of the TextBox control −

Sr.No.	Method Name & Description
1	**AppendText** Appends text to the current text of a text box.
2	**Clear** Clears all text from the text box control.
3	**Copy** Copies the current selection in the text box to the **Clipboard**.
4	**Cut**

		Moves the current selection in the text box to the **Clipboard**.
5	**Paste**	Replaces the current selection in the text box with the contents of the **Clipboard**.
6	**Paste(String)**	Sets the selected text to the specified text without clearing the undo buffer.
7	**ResetText**	Resets the Text property to its default value.
8	**ToString**	Returns a string that represents the TextBoxBase control.
9	**Undo**	Undoes the last edit operation in the text box.

Masked Textbox Control in VB.NET

MaskedTextBox provides all the facility of simple TextBox Control.

It allows user to validate input by specifying mask property. Mask property is used to specify pattern that must be followed while entering text into MaskedTextBox. It does not allow user to enter text that does not comply with the pattern specified in the Mask property.

Thus MaskedTextBox control is used to restrict invalid input from user.

Properties of Masked Textbox Control in VB.NET

Property	Purpose
AllowPromptAsInput	It is used to specify weather Prompt character can be entered as valid input character in MaskedTextBox or not. It has Boolean value. Default value is true.
AsciiOnly	It is used to specify weather only ASCII characters can be entered as valid input character in MaskedTextBox or not. It has Boolean value. Default value is false.
BackColor	It is used to get or set BackColor of the MaskedTextBox.
BeepOnError	It is used to specify weather control will generate system beep sound on each invalid character

	input or not. It has Boolean value. Default value is false.
ContextMenuStrip	It is used to specify the name of shortcut menu that is displayed when user right clicks on the MaskedTextBox.
CutCopyMaskFormat	While cut or copy MaskedTextBox contents into clipboard this property is used to specify weather to: Exclude Prompt And Literal Include Prompt Include Literals Include Prompt And Literal. Default value is Include Literals.
Enabled	It is used to specify weather MaskedTextBox Control is enabled or not at run time. It has Boolean value. Default value is true.
Font	It is used to set Font Face, Font Style, Font Size and Effects of the text associated with MaskedTextBox Control.
ForeColor	It is used to get or set Fore color of the text associated with

	MaskedTextBox Control.
HidePromptOnLeave	It is used to specify weather Prompt character is displayed or hide when focus is lost from MaskedTextBox. It has Boolean value. Default value is false.
HideSelection	It is used to specify weather text selection should be hidden or not when MaskedTextBox lose its focus. It has Boolean value. Default value is true.
InsertKeyMode	It is used to get or set method used for input characters in MaskedTextBox. It has following 3 options: (1) Default (2) Insert (3) Overwrite
Mask	It is used to get or set format string which determines weather characters entered in MaskedTextBox are valid or not.
PasswordChar	It is used to get or set a character that is displayed for each input characters in MaskedTextBox.
PromptChar	It is used to get or set Prompt character for MaskedTextBox

	Control. This character is displayed in MaskedTextBox when user has not entered any character.
ReadOnly	It is used to specify weather text associated with MaskedTextBox is ReadOnly or not. It has Boolean value. Default value is false.
RejectInputOnFirstFailure	It is used to specify weather input is rejected or not when entered characters are not complying with the pattern specified in mask property. It has Boolean value. Default value is false.
ResetOnPrompt	It is used to specify weather to reset and skip the current position when the input character is same as prompt character. It has Boolean value. Default value is true.
ResetOnSpace	It is used to specify weather to reset and skip the current position when the input is space character. It has Boolean value. Default value is false.
Text	It is used to get or set text associated with MaskedTextBox Control.

TextAlign	It is used to get or alignment of the text associated with MaskedTextBox Control.
TabIndex	It is used to get or set Tab order of the MaskedTextBox.
TabStop	It is used to specify weather user can use TAB key to set focus on MaskedTextBox Control or not. It has Boolean value. Default value is true.
TextMaskFormat	It is used to get or set value which determines the text that is returned from MaskedTextBox Control can contains literal, prompt , both or none. It has following 4 options: Exclude Prompt And Literal Include Prompt Include Literals Include Prompt And Literal. Default value is Include Literals.
UseSystemPasswordCharacter	It is used to specify weather each input character that is entered in MaskedTextBox is displayed as System Password character or not. It has Boolean value. Default value is false.
Visible	It is used to specify weather MaskedTextBox Control is visible

	or not at run time. It has Boolean value. Default value is true.

Methods of Masked Textbox Control in VB.NET

Method	Purpose
Append Text	It is used to append text at the end of current text in MaskedTextBox Control.
Clear	It is used to clear all text from MaskedTextBox Control.
Cut	It is used to move current selection of MaskedTextBox into clipboard.
Copy	It is used to copies selected text of MaskedTextBox in clipboard.
Paste	It is used to replace current selection of MaskedTextBox by contents of clipboard. It is also used to move contents of Clipboard to MaskedTextBox control where cursor is currently located.
Select	It is used to select specific text from MaskedTextBox.
SelectAll	It is used to select all text of MaskedTextBox.

DeselectAll	It is used to deselect all text selected in MaskedTextBox.
Show	It is used to show MaskedTextBox at run time.
Hide	It is used to hide MaskedTextBox at run time.
Focus	It is used to set input focus on MaskedTextBox at run time.

RichTextBox Control in VB.NET

RichTextBox Control allows user to display, input, edit and format text   information.
RichTextBox Control supports advance formatting features as compared to TextBox. Using RichTextBox user can format only selected portion of the text. User can also format paragraph using RichTextBox   Control.
RichTextBox Control also allows user to save as well as load file of RTF format and Standard ASCII format.

Properties of RichTextBox Control in VB.NET

Property	Purpose
AutoWordSelection	It is used to specify weather automatic word selection is ON or OFF. It has Boolean value. Default value is false.

BackColor	It is used to get or set background color of the RichTextBox.
ContextMenuStrip	It is used to specify the name of shortcut menu that is displayed when user right clicks on the RichTextBox.
DetectUrls	It is used to specify weather URL that is entered in RichTextBox is automatically displayed as hyperlink or not. It has Boolean value. Default value is true.
Enabled	It is used to specify weather RichTextBox Control is enabled or not at run time. It has Boolean value. Default value is true.
Font	It is used to set Font Face, Font Style, Font Size and Effects of the text associated with RichTextBox Control.
ForeColor	It is used to get or set Fore color of the text associated with RichTextBox Control.
HideSelection	It is used to specify weather text selection should be hidden or not when RichTextBox lose its focus. It has Boolean value. Default value is true.
MaxLength	It is used to get or set maximum number of characters that can be entered in RichTextBox. Default value is 2147483647.

Multiline	It is used to specify weather RichTextBox can be expanded to enter more than one line of text or not. It has Boolean value. Default value is true.
ReadOnly	It is used to specify weather text associated with RichTextBox is ReadOnly or not. It has Boolean value. Default value is false.
RightMargin	It is used to get or set right margin of the text in RichTextBox.
ScrollBars	It is used to get or set type of scrollbars to be added with RichTextBox control. It has following 4 options: (1) None (2) Horizontal (2) Vertical (3) Both Default value is Both.
Size	It is used to get or set height and width of RichTextBox control in pixel.
Text	It is used to get or set text associated with RichTextBox Control.
TabIndex	It is used to get or set Tab order of the RichTextBox.
TabStop	It is used to specify weather user can use TAB key to set focus on RichTextBox Control or not. It has Boolean value. Default value is

	true.
Visible	It is used to specify weather RichTextBox Control is visible or not at run time. It has Boolean value. Default value is true.
WordWrap	It is used to specify weather line will be automatically word wrapped while entering multiple line of text in RichTextBox control or not. It has Boolean value. Default value is true.
ZoomFactor	It is used to get or set current zoom level of RichTextBox.
SelectedRtf	It is used to get or set currently selected RTF formatted text in RichTextBox.
SelectedText	It is used to get or set currently selected text in RichTextBox.
SelectionAlignment	It is used to get or set horizontal alignment of the text selected in RichTextBox.
SelectionBackColor	It is used to get or set BackColor of the text selected in RichTextBox.
SelectionBullet	It is used to get or set value which determines weather bullet style should be applied to selected text or not.
SelectionColor	It is used to get or set Fore Color of the text

	selected in RichTextBox.
SelectionFont	It is used to get or set font face, font style, and font size of the text selected in RichTextBox.
SelectionLength	It is used to get or set number of characters selected in the RichTextBox.
SelectionStart	It is used to get or set starting point of the text selected in the RichTextBox.

Methods of RichTextBox Control in VB.NET

Method	Purpose
Cut	It is used to move current selection of RichTextBox into clipboard.
Copy	It is used to copies selected text of RichTextBox in clipboard.
Paste	It is used to replace current selection of TextBox by contents of clipboard. It is also used to move contents of Clipboard to RichTextBox control where cursor is currently located.
Select	It is used to select specific text from RichTextBox.
SelectAll	It is used to select all text of RichTextBox.

DeselectAll	It is used to deselect all text selected in RichTextBox.
Clear	It is used to clear all text from RichTextBox Control.
AppendText	It is used to append text at the end of current text in RichTextBox Control.
ClearUndo	It is used to clear information from the Undo buffer of the RichTextBox.
Find	It is used to find starting position of first occurrence of a string in the RichTextBox control. If a string is not found then it returns -1.
SaveFile	It is used to save contents of RichTextBox in to Rich Text Format (RTF) file.
LoadFile	It is used to loads a Rich Text Format (RTF) file or standard ASCII text file into RichTextBox Control.
Undo	It is used to undo last edit operation of RichTextBox.
Redo	It is used to redo the last operation that is undo using undo method.

NumericUpDown Control in VB.NET

NumericUpDown Control is a rectangle box with Up and Down arrow key. It is used to represent a numeric value that can be incremented or

decremented between specific range. You can also enter numeric value directly in the NumericUpDown Control between specific range if its readonly property is false.

Properties of NumericUpDown Control

Various Properties of NumericUpDown Control are:

Property Name	Description
Name	It is used to specify name of the NumericUpDown Control.
DecimalPlaces	It is used to specify number of digits to be display after decimal point. Default value is 0.
Hexadecimal	It is used to determine weather numeric value in the NumericUpDown Control is display in Hexadecimal form or not. It has boolean value. Default value is false.
Increment	It is used to specify the amount by which the value of NumericUpDown Control is Incremented or Decremented each time when Up arrow or Down arrow is clicked.
InterceptArrowKeys	It is used to determine weather value in the NumericUpDown control can be increment or decrement using UP and DOWN arrow key or not. It has boolean value. Default value is true.
Maximum	It is used to specify maximum value for the NumericUpDown Control.
Minimum	It is used to specify minimum value for the NumericUpDown Control.
ReadOnly	It is used to determine weather Numeric value in the NumericUpDown Control is ReadOnly or Not. It has boolean value. Default value is false.

Value	It is used to Get or Set current value of NumericUpDown Control.
ThousandsSeparator	It is used to determine weather thousands separator will be inserted after every three decimal digits or not. It has boolean value. Default value is false.
TextAlign	It is used to set alignment of the Numeric value in the NumericUpDown Control. It can have one of the following three values: left,right or Center. Default value is left.
UpDownAlign	It is used to set alignment of the **UPDOWN** arrow in the NumericUpDown Control. It can have one of the following two values: left,right. Default value is right.

Methods of NumericUpDown Control

Various Methods of NumericUpDown Control are:

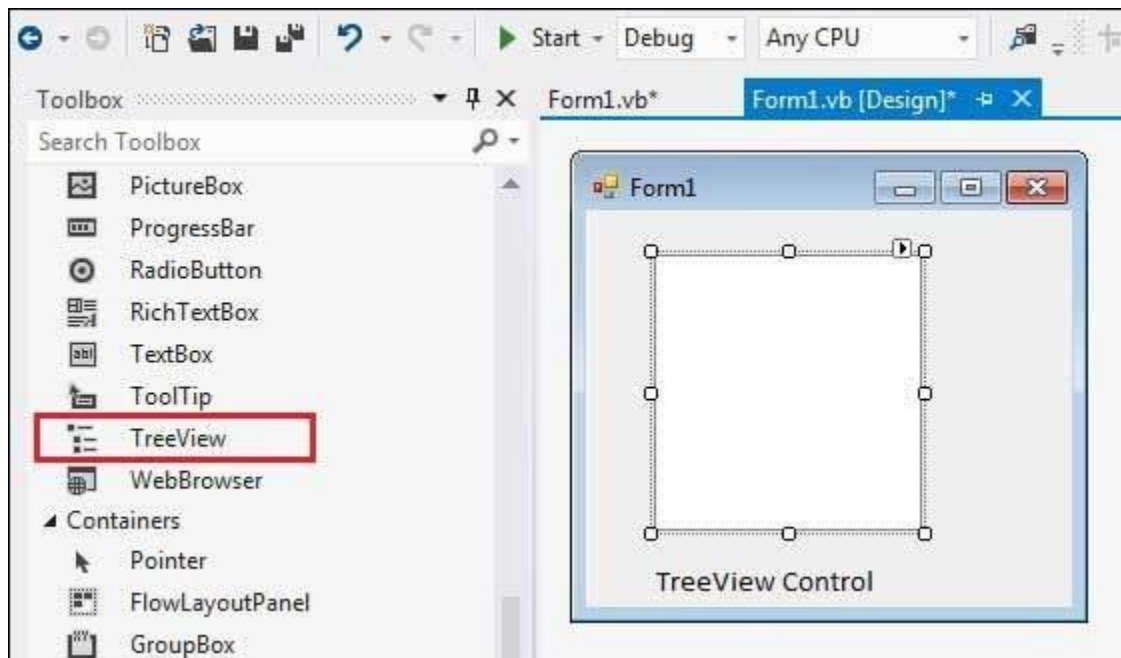| Method Name	Description
UpButton	It is used to Increment the current value of NumericUpDown Control by the value specified in the Increment Property.
DownButton	It is used to Decrement the current value of NumericUpDown Control by the value specified in the Increment Property.

Event of NumericUpDown Control

Various Events of NumericUpDown Control are:

Event Name	Description
ValueChanged	This event fires each time when the value in the NumericUpDown Control changed.

VB.Net - TreeView Control

–

The TreeView control is used to display hierarchical representations of items similar to the ways the files and folders are displayed in the left pane of the Windows Explorer. Each node may contain one or more child nodes.

Let's click on a TreeView control from the Toolbox and place it on the form.

Properties of the TreeView Control

The following are some of the commonly used properties of the TreeView control −

Sr.No.	Property & Description
1	**BackColor** Gets or sets the background color for the control.
2	**BackgroundImage** Gets or set the background image for the TreeView control.
3	**BackgroundImageLayout** Gets or sets the layout of the background image for the TreeView control.
4	**BorderStyle** Gets or sets the border style of the tree view control.
5	**CheckBoxes** Gets or sets a value indicating whether check boxes are displayed next to the tree nodes in the tree view control.
6	**DataBindings** Gets the data bindings for the control.
7	**Font**

	Gets or sets the font of the text displayed by the control.
8	**FontHeight** Gets or sets the height of the font of the control.
9	**ForeColor** The current foreground color for this control, which is the color the control uses to draw its text.
10	**ItemHeight** Gets or sets the height of each tree node in the tree view control.
11	**Nodes** Gets the collection of tree nodes that are assigned to the tree view control.
12	**PathSeparator** Gets or sets the delimiter string that the tree node path uses.
13	**RightToLeftLayout** Gets or sets a value that indicates whether the TreeView should be laid out from right-to-left.
14	**Scrollable** Gets or sets a value indicating whether the tree view control displays scroll bars when they are needed.

15	**SelectedImageIndex**
	Gets or sets the image list index value of the image that is displayed when a tree node is selected.
16	**SelectedImageKey**
	Gets or sets the key of the default image shown when a TreeNode is in a selected state.
17	**SelectedNode**
	Gets or sets the tree node that is currently selected in the tree view control.
18	**ShowLines**
	Gets or sets a value indicating whether lines are drawn between tree nodes in the tree view control.
19	**ShowNodeToolTips**
	Gets or sets a value indicating ToolTips are shown when the mouse pointer hovers over a TreeNode.
20	**ShowPlusMinus**
	Gets or sets a value indicating whether plus-sign (+) and minus-sign (-) buttons are displayed next to tree nodes that contain child tree nodes.
21	**ShowRootLines**
	Gets or sets a value indicating whether lines are drawn between the tree nodes that are at the root of the tree

		view.
22	**Sorted**	
	Gets or sets a value indicating whether the tree nodes in the tree view are sorted.	
23	**StateImageList**	
	Gets or sets the image list that is used to indicate the state of the TreeView and its nodes.	
24	**Text**	
	Gets or sets the text of the TreeView.	
25	**TopNode**	
	Gets or sets the first fully-visible tree node in the tree view control.	
26	**TreeViewNodeSorter**	
	Gets or sets the implementation of IComparer to perform a custom sort of the TreeView nodes.	
27	**VisibleCount**	
	Gets the number of tree nodes that can be fully visible in the tree view control.	

Methods of the TreeView Control

The following are some of the commonly used methods of the TreeView control −

Sr.No.	Method Name & Description
1	**CollapseAll** Collapses all the nodes including all child nodes in the tree view control.
2	**ExpandAll** Expands all the nodes.
3	**GetNodeAt** Gets the node at the specified location.
4	**GetNodeCount** Gets the number of tree nodes.
5	**Sort** Sorts all the items in the tree view control.
6	**ToString** Returns a string containing the name of the control.

ToolTip Control in VB.NET

ToolTip Control is used to display brief description of the control whenever focus is set on that control. Brief description contains

information such as for which purpose this control is used or which type of input should be accepted by that control.

Properties of ToolTip Control

Various Properties of ToolTip Control are:

Property Name	Description
Name	It is used to specify name of the ToolTip Control.
Active	It is used to determine weather ToolTip Control will be Active or Not. It has boolean value. Default value is true. ToolTip will appear only when this property is set to true.
InitialDelay	It is used to specify amount of time the mouse pointer should remain on the control to display ToolTip text.
AutoPopDelay	It is used to specify amount of time the ToolTip text remains visible if mouse pointer remains on the control.
IsBallon	It is used to determine weather ToolTip will be displayed in the form of ballon or not. It has boolean value. Default value is false.
ToolTipIcon	It is used to set icon to be displayed in the ToolTip. It can have following values: **None,Info,Error,Warning**. Default value is None.
ToolTipTitle	It is used to set title to be displayed in each ToolTip.
UseAnimation	It is used to determine weather animations are used when ToolTip is display and Hide. It has boolean value.Default value is true.

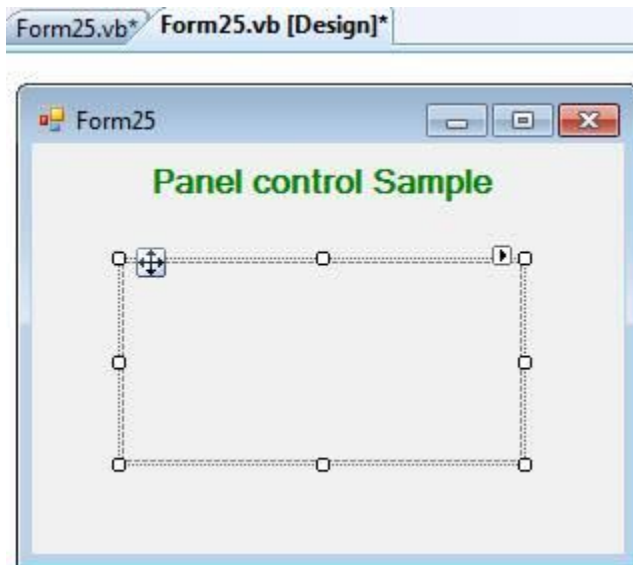|	
ShowAlways	It is used to determine weather ToolTip will be displayed always even if the parent window is not active. It has boolean value. Default value is false.
StripAmpersands	It is used to determine weather ampersand (&) will be displayed in the ToolTip text or not. It has boolean value. Default value is false. If it is set to true then ampersand (&) will not display in the ToolTip text.

Methods of ToolTip Control

Various Methods of ToolTip Control are:

Method Name	Description
SetToolTip	It is used to specify ToolTip text to be display when mouse pointer is on particular control. **Syntax:** ToolTip1.SetToolTip(ControlName,"ToolTipText")
GetToolTip	It is used to retrieve ToolTip text associated with particular control. **Syntax:** ToolTip1.GetToolTip(ControlName)
RemoveAll	It is used to remove all ToolTip text associated with ToolTip Control. **Syntax:** ToolTip1.RemoveAll()

The Panel control is a container of other controls. The Panel control is displayed by default without any borders at run time.
How to use Panel control

Drag and drop Panel control from toolbox on the window Form.

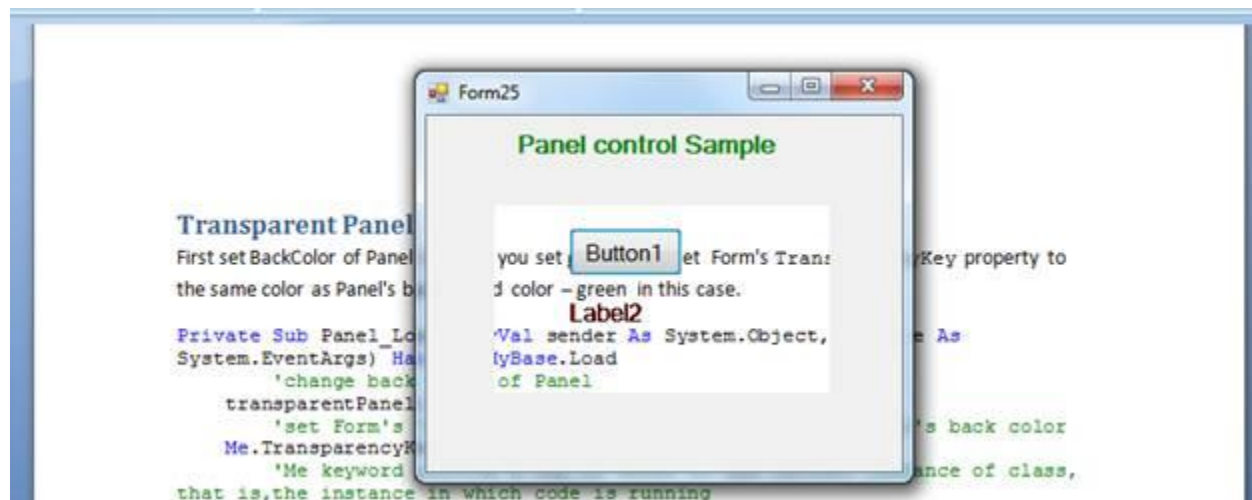Collection of control can be placed in side Panel.

Transparent Panel

First set BackColor of Panel suppose you set green then set Form's TransparencyKey property to the same color as Panel's background color –red in this case.

```
Private Sub Form25_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        'change back color of Panel
        Panel1.BackColor = Color.Red
        'set Form's TransparencyKey to the same
color as Panel's back color
        Me.TransparencyKey = Color.Red
        'Me keyword provide a way to refer to the
current instance of class, that is,the instance in
which code is running

 End Sub
```

Now when you run the application then panel will be transparent.

GroupBox Control in VB.NET

GroupBox control is used to group other controls of VB.NET. GroupBox control having a frame to indicate boundary and a text to indicate header or title. Generally GroupBox control is used as a container for Radio Button. When Radio Buttons are grouped using GroupBox, user can select one RadioButton from each GroupBox.

Properties of GroupBox Control in VB.NET

Property	Purpose
BackColor	It is used to get or set background color of the GroupBox.
BackgroundImage	It is used to get or set background Image of the GroupBox.
BackgroundImageLayout	It is used to get or set background Image layout of the GroupBox. It has one of the following values: None, Tile, Centre, Stretch, Zoom
Font	It is used to get or set font Style, Font Size, Font Face of the text contained in GroupBox Control.
ForeColor	It is used to get or set color of the text

	contained in GroupBox Control.
Enabled	It is used to specify weather GroupBox Control is enabled or not. It has Boolean value. Default value is true.
Visible	It is used to specify weather GroupBox Control is visible or not at run time. It has Boolean value. Default value is true.
Text	It is used to get or set Title or Header Text of the GroupBox Control.

Methods of GroupBox Control in VB.NET

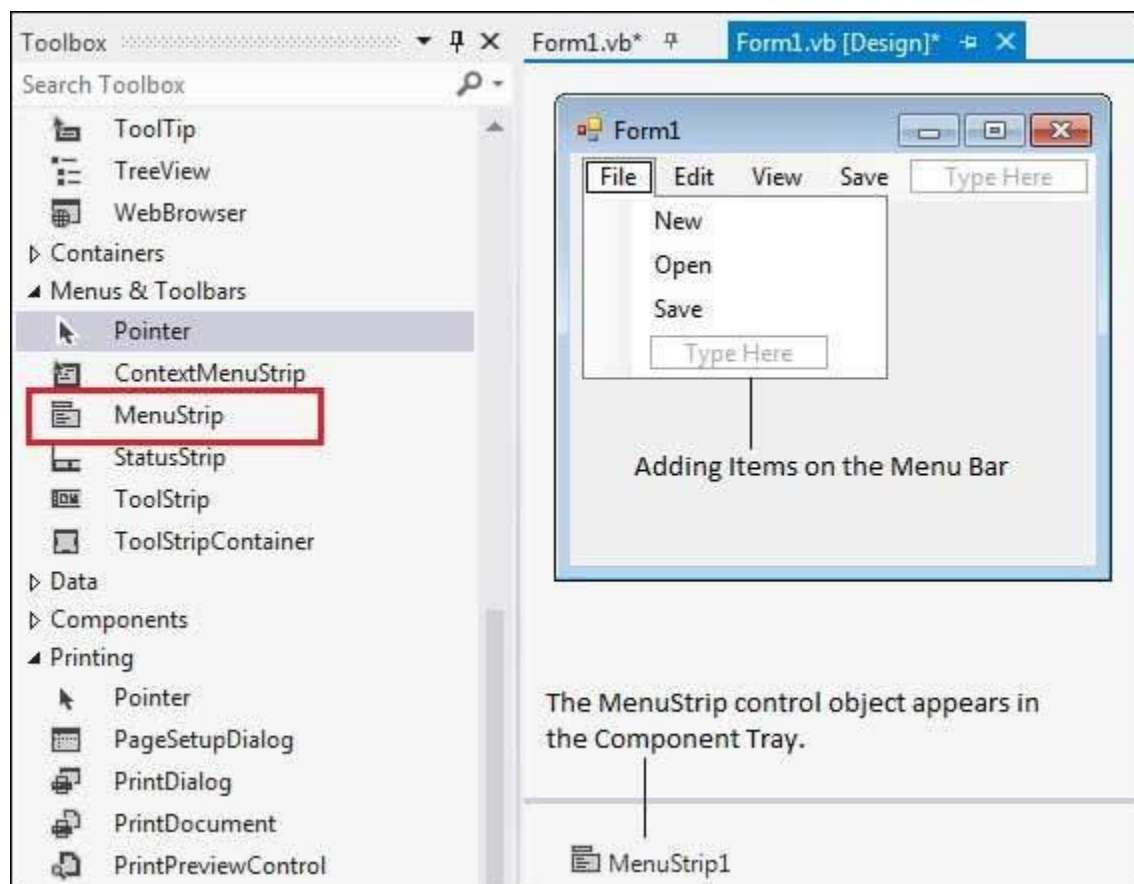| Method	Purpose
Show	It is used to show GroupBox Control.
Hide	It is used to Hide GroupBox Control at run time.
Focus	It is used to set cursor or focus on GroupBox Control.

VB.Net - MenuStrip Control

Advertisements

The **MenuStrip** control represents the container for the menu structure.

The MenuStrip control works as the top-level container for the menu structure. The ToolStripMenuItem class and the ToolStripDropDownMenu class provide the functionalities to create menu items, sub menus and drop-down menus.

The following diagram shows adding a MenuStrip control on the form −

Properties of the MenuStrip Control

The following are some of the commonly used properties of the MenuStrip control −

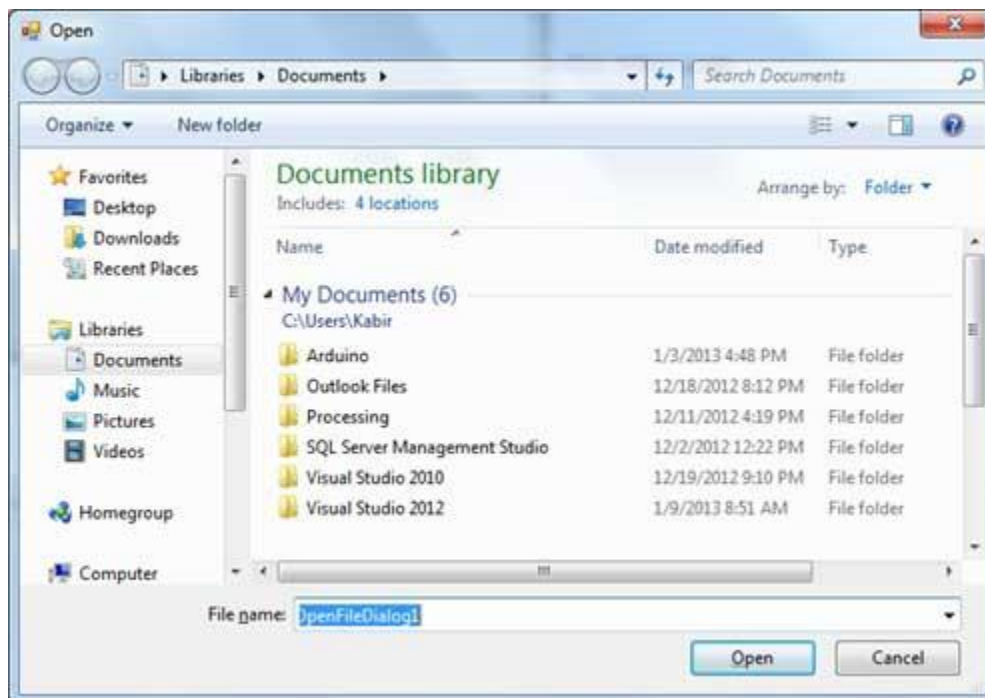| Sr.No.	Property & Description
1	**CanOverflow** Gets or sets a value indicating whether the MenuStrip supports overflow functionality.
2	**GripStyle** Gets or sets the visibility of the grip used to reposition the control.
3	**MdiWindowListItem** Gets or sets the ToolStripMenuItem that is used to display a list of Multiple-document interface (MDI) child forms.
4	**ShowItemToolTips** Gets or sets a value indicating whether ToolTips are shown for the MenuStrip.
5	**Stretch** Gets or sets a value indicating whether the MenuStrip stretches from end to end in its container.

VB.Net - OpenFileDialog Control

The **OpenFileDialog** control prompts the user to open a file and allows the user to select a file to open. The user can check if the

file exists and then open it. The OpenFileDialog control class inherits from the abstract class **FileDialog**.

If the ShowReadOnly property is set to True, then a read-only check box appears in the dialog box. You can also set the ReadOnlyChecked property to True, so that the read-only check box appears checked.

Following is the Open File dialog box −

Properties of the OpenFileDialog Control

The following are some of the commonly used properties of the OpenFileDialog control −

Sr.No.	Property & Description
1	**AddExtension** Gets or sets a value indicating whether the dialog box automatically adds an extension to a file name if the user

	omits the extension.
2	**AutoUpgradeEnabled** Gets or sets a value indicating whether this FileDialog instance should automatically upgrade appearance and behavior when running on Windows Vista.
3	**CheckFileExists** Gets or sets a value indicating whether the dialog box displays a warning if the user specifies a file name that does not exist.
4	**CheckPathExists** Gets or sets a value indicating whether the dialog box displays a warning if the user specifies a path that does not exist.
5	**CustomPlaces** Gets the custom places collection for this FileDialog instance.
6	**DefaultExt** Gets or sets the default file name extension.
7	**DereferenceLinks** Gets or sets a value indicating whether the dialog box returns the location of the file referenced by the shortcut or whether it returns the location of the shortcut (.lnk).

8	**FileName**
	Gets or sets a string containing the file name selected in the file dialog box.
9	**FileNames**
	Gets the file names of all selected files in the dialog box.
10	**Filter**
	Gets or sets the current file name filter string, which determines the choices that appear in the "Save as file type" or "Files of type" box in the dialog box.
11	**FilterIndex**
	Gets or sets the index of the filter currently selected in the file dialog box.
12	**InitialDirectory**
	Gets or sets the initial directory displayed by the file dialog box.
13	**Multiselect**
	Gets or sets a value indicating whether the dialog box allows multiple files to be selected.
14	**ReadOnlyChecked**
	Gets or sets a value indicating whether the read-only check box is selected.

15	**RestoreDirectory**
	Gets or sets a value indicating whether the dialog box restores the current directory before closing.
16	**SafeFileName**
	Gets the file name and extension for the file selected in the dialog box. The file name does not include the path.
17	**SafeFileNames**
	Gets an array of file names and extensions for all the selected files in the dialog box. The file names do not include the path.
18	**ShowHelp**
	Gets or sets a value indicating whether the Help button is displayed in the file dialog box.
19	**ShowReadOnly**
	Gets or sets a value indicating whether the dialog box contains a read-only check box.
20	**SupportMultiDottedExtensions**
	Gets or sets whether the dialog box supports displaying and saving files that have multiple file name extensions.
21	**Title**
	Gets or sets the file dialog box title.

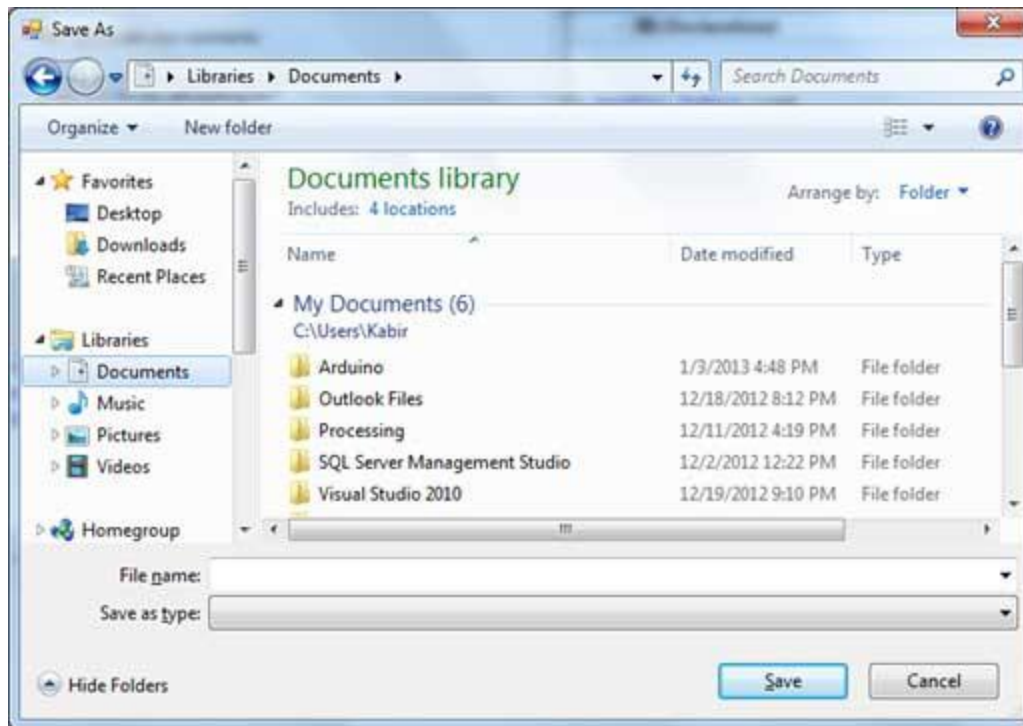| 22 | **ValidateNames** |
| | Gets or sets a value indicating whether the dialog box accepts only valid Win32 file names. |

Methods of the OpenFileDialog Control

The following are some of the commonly used methods of the OpenFileDialog control −

Sr.No.	Method Name & Description
1	**OpenFile**
	Opens the file selected by the user, with read-only permission. Th specified by the FileName property.
2	**Reset**
	Resets all options to their default value.

VB.Net - SaveFileDialog Control

The **SaveFileDialog** control prompts the user to select a location for saving a file and allows the user to specify the name of the file to save data. The SaveFileDialog control class inherits from the abstract class FileDialog.

Following is the Save File dialog box −

Properties of the SaveFileDialog Control

The following are some of the commonly used properties of the SaveFileDialog control −

Sr.No.	Property & Description
1	**AddExtension** Gets or sets a value indicating whether the dialog box automatically adds an extension to a file name if the user omits the extension.
2	**CheckFileExists** Gets or sets a value indicating whether the dialog box displays a warning if the user specifies a file name that does not exist.

3	**CheckPathExists**
	Gets or sets a value indicating whether the dialog box displays a warning if the user specifies a path that does not exist.
4	**CreatePrompt**
	Gets or sets a value indicating whether the dialog box prompts the user for permission to create a file if the user specifies a file that does not exist.
5	**DefaultExt**
	Gets or sets the default file name extension.
6	**DereferenceLinks**
	Gets or sets a value indicating whether the dialog box returns the location of the file referenced by the shortcut or whether it returns the location of the shortcut (.lnk).
7	**FileName**
	Gets or sets a string containing the file name selected in the file dialog box.
8	**FileNames**
	Gets the file names of all selected files in the dialog box.
9	**Filter**
	Gets or sets the current file name filter string, which determines the choices that appear in the "Save as file

	type" or "Files of type" box in the dialog box.
10	**FilterIndex** Gets or sets the index of the filter currently selected in the file dialog box.
11	**InitialDirectory** Gets or sets the initial directory displayed by the file dialog box.
12	**OverwritePrompt** Gets or sets a value indicating whether the Save As dialog box displays a warning if the user specifies a file name that already exists.
13	**RestoreDirectory** Gets or sets a value indicating whether the dialog box restores the current directory before closing.
14	**ShowHelp** Gets or sets a value indicating whether the Help button is displayed in the file dialog box.
15	**SupportMultiDottedExtensions** Gets or sets whether the dialog box supports displaying and saving files that have multiple file name extensions.
16	**Title**

	Gets or sets the file dialog box title.
17	**ValidateNames** Gets or sets a value indicating whether the dialog box accepts only valid Win32 file names.

Methods of the SaveFileDialog Control

The following are some of the commonly used methods of the SaveFileDialog control −

Sr.No.	Method Name & Description
1	**OpenFile** Opens the file with read/write permission.
2	**Reset** Resets all dialog box options to their default values