Unit 2: Elements of Visual Basic .net

Contents

Properties, Events and Methods of

Form, Label, TextBox, ListBox, Combo Box, Radio Button, Button, Check Box, Progress Bar, Date Time Picker, Calendar, Picture Box, HScrollbar, VScrollBar, Group Box, ToolTip, Timer

Controls

A *control* is a programming entity that has a graphical component. A control sits on a form and interacts with the user, providing information and possibly allowing the user to manipulate it. Text boxes, labels, buttons, scroll bars, drop - down lists, menu items, toolstrips, and just about everything else that you can see and interact with in a Windows application is a control.

Controls are an extremely important part of any interactive application. They give information to the user (Label, ToolTip, TreeView, PictureBox) and organize the information so that it ' s easier to understand (GroupBox, Panel, TabControl). They enable the user to enter data (TextBox, RichTextBox, ComboBox, MonthCalendar), select options (RadioButton, CheckBox, ListBox), tell the application to take action (Button, MenuStrip, ContextMenuStrip), and interact with objects outside of the application (OpenFileDialog, SaveFileDialog, PrintDocument, PrintPreviewDialog). Some controls also provide support for other controls (ImageList, ToolTip,ContextMenuStrip, ErrorProvider).

Form Control

In Visual Basic (any version), a window is known as a *form*. The Visual Basic Windows Form class is a descendant of the Control class. Forms also play a very central role in most Visual Basic applications. They are the largest graphical unit with which the user interacts directly. The user can minimize, restore, maximize, and close forms.

Properties of Form Control

1. **Name** Property: Indicates the name used in code to identify the object.
2. **FormBorderStyle** property: is used to set the border style of the form. Here are the possible values for that property:
   - **Fixed3D—** A fixed, three-dimensional border.
   - **FixedDialog—** A thick, fixed dialog-style border.
   - **FixedSingle—** A fixed, single-line border.
   - **FixedToolWindow—** A tool window border that is not resizable.
   - **None—** No border.
   - **Sizable—** A resizable border.
   - **SizableToolWindow—** A resizable tool window border.

3. **ControlBox** property :

   Forms usually come with minimizing and maximizing buttons, as well as a close box at upper right. To remove these buttons, you can set the form's **ControlBox** property to **False**. You can also remove the minimizing and maximizing buttons independently, with the **MaximizeBox** and **MinimizeBox** properties.

4. **StartPosition** property: is used to specify initial position of the form on the screen. Possible values are:
   - **CenterParent—** The form is centered within the bounds of its parent form.
   - **CenterScreen—** The form is centered on the current display and has the dimensions specified in the form's size.
   - **Manual—** The **Location** and **Size** properties of the form will determine its starting position.
   - **WindowsDefaultBounds—** The form is positioned at the Windows default location and has the bounds determined by Windows default.
   - **WindowsDefaultLocation—** The form is positioned at the Windows default location and has the dimensions specified in the form's size.
5. **Text** Property:
   Text Property of a form is for setting the caption you want to display in the title bar at the top.
6. **Visible** Property:
   The visible property is used to set the visibility of the form control. It takes only two values True (makes the form visible) and False (makes the form hidden).

Method:

Event	Description
gotFocus	The user brings the control into focus.
enter	The user moves focus to a control.
clicked	The user clicks a control.
leave	The user moves focus out of a control.
lostFocus	The user brings the control out of focus.
mouseDblClick	The user double-clicks a control.
mouseDown	The user releases the left mouse pointer button after clicking a control.
mouseEnter	The user moves the mouse pointer into the control area.
mouseLeave	The user moves the mouse pointer out of the control area.

mouseMove	The user moves the mouse pointer over the control.
mouseUp	The user presses the left mouse pointer button.

Example 1 :

Control Name	Property Name	Description
Form	Name	frmFirstForm
	Text	Form Demo
	Visible	True
	StartPosition	Center

Label Control

Label control is used to display Text on the Form. Main property of the label control is the text property which is used to set the text in the label. Drag and drop Label control from toolbox on the window Form.

Property

Property	Means
AutoSize	Sets/gets a value specifying if the control should be automatically resized to display all its contents.
BorderStyle	Sets/gets the border style for the control.
FlatStyle	Sets/gets the flat style appearance of the label control.
Image	Sets/gets the image that is displayed on a Label.
ImageAlign	Sets/gets the alignment of an image that is displayed in the control.
PreferredHeight	Gets the preferred height of the control.

PreferredWidth	Gets the preferred width of the control.
Text	Sets/gets the text in the control.
TextAlign	Sets/gets the alignment of text in the control.
UseMnemonic	Sets/gets a value specifying if the control treats an ampersand character (&) in the control's Text property to be an access key character.

Example 2:

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.

The TextBox control accepts input from the user. It can also be used to display text. By default we can enter up to 2048 characters in a TextBox but if the Multiline property is set to True we can enter up to 32KB of text.

Noteworthy public properties of *TextBox* objects

Property	Description
AutoSize	Sets/gets a value specifying if the height of the control automatically adjusts when the font in the control is changed.
BackColor	Sets/gets the background color of the control.
BorderStyle	Sets/gets the border type of the text box control.
CanUndo	Returns a value specifying if the user can undo the previous operation.
ForeColor	Sets/gets the foreground color.
HideSelection	Sets/gets a value specifying if the selected text in the text box control remains highlighted when the text box loses focus.
Lines	Sets/gets the lines of text.
MaxLength	Sets/gets the maximum number of characters the user can type into the text box.
Modified	Indicates if the text box control has been modified by the user since the control was created or its contents were last set.
Multiline	Sets/gets a value specifying if this is a multiline text box control.
PasswordChar	Sets/gets the character used to mask characters of a password in a single-line text box.

ReadOnly	Sets/gets a value specifying if text in the text box is read-only.
ScrollBars	Sets/gets what scroll bars should appear in a multiline text box.
SelectedText	Sets/gets a value specifying the currently selected text in the control.
SelectionLength	Sets/gets the number of characters selected in the text box.
SelectionStart	Sets/gets the starting point of text selected in the text box.
Text	Sets/gets the current text in the text box.
TextAlign	Sets/gets how text is aligned in a text box control.
TextLength	Gets the length of text in the control.
WordWrap	Indicates if a multiline text box control automatically wraps words.

Noteworthy public events of *TextBox* objects

Methods	Means
AppendText	Appends text to the current text in the text box.
Clear	Clears all text from the text box.
ClearUndo	Clears information about the most recent operation of the text box.
Copy	Copies the selected text in the text box to the Clipboard.
Cut	Moves the selected text in the text box to the Clipboard.
Paste	Replaces the selected text in the text box with the contents of the Clipboard.
ScrollToCaret	Scrolls the text box to the caret position.
Select	Selects text in the text box.
SelectAll	Selects all text in the text box.
Undo	Undoes the last edit operation in the text box.

Noteworthy public events of *TextBox* objects.

Event	Means
AutoSizeChanged	Occurs when the value of the AutoSize property is changed.
Click	Occurs when the text box is clicked.
ReadOnlyChanged	Occurs when the value of the ReadOnly property is changed.

Example:

Control Name	Property	Value
Label1	Text	Enter Name
Label2	Text	Address
Label3	Text	AccountNo
Button1	Name	btnSend
	Text	Send
TextBox1	Name	txtName
TextBox2	Name	txtAddress
	MultiLine	True
TextBox3	Name	txtPassword
	PasswordChar	*

```
Public Class TextBoxDemo

    Private Sub btnSend_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnSend.Click
        MsgBox("Dear " & txtName.Text + " Your Account No is "
& txtAccountNo.Text)
    End Sub

End Class
```

Button Control

The Button control represents a standard Windows button. Button Control in VB.NET is used to trigger some action for the form. Just by clicking on the button the information is managed effectively in Visual Basic. Following table lists the Properties, Events, Methods used commonly with the Button Control

Properties:

Properties	Description
Visible	Property used to make the control visible or invisible
Enabled	This property is used to enable the control.
Width	This property is used to specify the width of the control.
Font	Property used to set the font properties like bold, Italic, Name and so on.
Height	Property used to specify the height.
Left	Property is set the X coordinate of the control.
BackColor	Property is used set the background color of the control.
Image	Property is used to set a background picture for the control.
DialogResult	Property is used to set or get the value returned to the parent from when the button is clicked.
Image	Property is used to set a a background picture for the control.
ImageAlign	Property is used to get or set alignment of the image.
ImageList	Property is used to set or get the Image List that contains the images displayed.

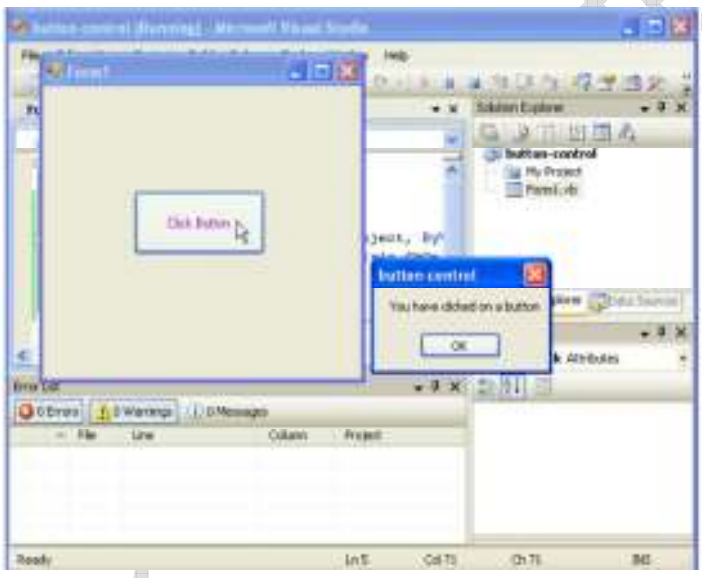
Methods:

Method	Description
PerformClick	Method used to generate a click event for the radio button.

Events:

Events	Description
Enter	Triggered when Command Button Control gets focus.
Click	Triggered when the control is clicked.
Leave	Triggered when control looses focus.

Example:



Properties

Control Name	Property	Value
Form	Name	frmButtonDemo
	Text	ButtonDemo
Button1	Text	Click Button

Name	btnClick

Explanation:

- Select the form on which the button resides.
- In the Properties window, set the form's Button property to the Button control's name.
- To respond to a button click, In the button's Click event handler write the code to run. Button1_Click must be bound to the control.
- **ForeColor**,**Text** properties of the button control are changed.
- **Click** event of the button is used to display a message.

Code:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnClick.Click
        MsgBox("You have clicked the Button", MsgBoxStyle.OkOnly)
    End Sub
```

Checkbox Control

CheckBox Control in VB.NET is used to select the choice by checking or not checking a check box. You use a checkbox to give the user an option, such as true/false or yes/no. The checkbox control can display an image or text or both. Checkbox is a selection control and user has to select the option to interact with it.

Following tables lists the commonly used Properties, Methods, Events with the Check Box Control is Visual Basic.net.

Properties	Description
Appearance	Property used to set or get the appearance of a checkbox.
AutoCheck	Property used to specify whether to change appearance automatically when the checkbox is clicked.
CheckAlign	Property is used to get or set horizontal and vertical alignment of a checkbox in a control.
Checked	Property is used to set or get the value indicating if the checkbox is in checked state.
CheckState	Property is used to set or get state of a three

	state checkbox.
FlatStyle	Property set or get flatstyle appearance of the checkbox.
Image	Property is used to get or set the image displayed in checkbox.
ImageAlign	Property is used to set or get alignment of the image on a checkbox.
ImageList	Property used to set or get the image list that contains the image displayed in a checkbox.
ImageIndex	Property used to set or get the image list index of the image displayed in the checkbox.
ThreeState	Property is used to specify if the checkbox will allow three check states rather than two.

Events:

Events	Description
AppearanceChanged	Triggered when the Appearance property is changed.
CheckedChanged	Triggered when the Checked property is changed.
CheckStateChanged	Triggered when the CheckState property is changed.

Example

Control name	Property Name	Value
Label1	Text	Select your Hobbies
Button1	Name	btnShowHobies
	Text	Show Hobbies
CheckBox1	Name	chkReading
	Text	Reading
CheckBox2	Name	chkWriting
	Text	Writing
CheckBox3	Name	chkDrawing
	Text	Drawing
CheckBox4	Name	chkPlaying
	Text	Playing

Code:

```
'EventName: Click of btnShowHobbies
    'Event Handler:btnShowHobbies_Click()
    'Parameters: ByVal sender As System.Object, ByVal e As
System.EventArgs
    'Return Type: -
```

```
    'Description: is invoked when btnShowHobbies is clicked.
    Private Sub btnShowHobbies_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnShowHobbies.Click
        'Declare variables
        Dim msg As String = "Your Hobbies are : "
        'Check the hobbies of the user and display a message
        If chkReading.Checked Then
            msg = msg & "Reading"
        End If
        If chkWriting.Checked Then
            msg = msg & ",Writing"
        End If

        If chkDrawing.Checked Then
            msg = msg & ",Drawing"
        End If

        If chkPlaying.Checked Then
            msg = msg & ",Playing"
        End If
        MsgBox(msg)
    End Sub
```

Radio Buttons

Radio buttons, also called option buttons, are similar to checkboxes—the user can select and deselect them—except for two things: they are round where checkboxes are square, and you usually use radio buttons together in groups.

In fact, that's the functional difference between checkboxes and radio buttons—checkboxes can work independently, but radio buttons are intended to work in groups. When you select one radio button in a group, the others are automatically deselected. For example, although you might use checkboxes to select trimmings on a sandwich (of which there can be more than one), you would use radio buttons to let the user select one of a set of exclusive options, such as the current day of the week.

When the user selects one radio button in a group, the others clear automatically. All radio buttons in a given container, such as a form, make up a group. To create multiple groups on one form, you place each additional group in its own container, such as a group box or panel control.

Following table lists the Properties, Events, Methods used commonly with the Radio Button Control in Visual Basic.net 2008.

Properties:

Properties	Description
Text	Property used to display the text center aligned

	on the control.
Size	This property is used to specify the width, height of the control.
Font	Property used to set the font properties like bold, Italic, Name and so on.
Appearance	Property is used to set or get the value that determines the appearance.
Autocheck	Property is used to set or get whether the checked value, appearance change when clicked on the radio button.
Checked	Property is used to set or get the value indicating whether the radio button is checked.
FlatStyle	Property is used to set or get flatStyle appearance of the radio button.
Image	Property is used to set or get the image displayed on the radio button.
ImageAlign	Property is used to set or get the alignment of the image on the control.
ImageList	Property is used to set or get the images displayed on the control.

Methods:

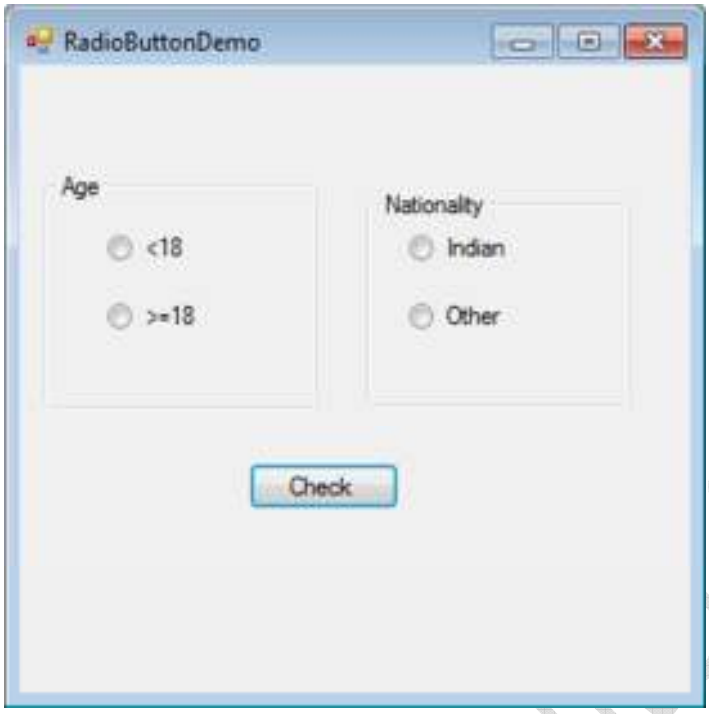
Method	Description
PerformClick	Method used to generate a click event for the radio button.

Events:

Events	Description
CheckedChanged	Triggered when the control is checked or changed.
AppearanceChanged	Triggered when the appearance property

changes.

Example:



Property Chart:

Control Name	Property	Value
Button1	Name	btnCheck
	Text	Check
RadioButton1	Name	rdbLess18
	Text	<18
RadioButton2	Name	rdbGreater18
	Text	>=18
RadioButton3	Name	rdbIndian
	Text	Indian
RadioButton4	Name	rdbOther
	Text	Other

Code:

```
'Event name: Click Event of btnCheck
    'Event handler name: btnCheck_Click
    'Paramters: ByVal sender As System.Object, ByVal e As
System.EventArgs
    'Return Type: NA
    'Description: gets invoked when the btncheck button is
clicked
    Private Sub btnCheck_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnCheck.Click
        'check if radio buttons >18 and Indian are checked
        'if yes show You are eligible to vote message
        If rdbGreater18.Checked And rdbIndian.Checked Then
            MsgBox(" You are eligible to vote")
        End If
        'check if radio buttons <18 and Other are checked
        'if yes show You are too young to vote and not an
Indian
        If rdbLess18.Checked And rdbOther.Checked Then
            MsgBox(" You are too young to vote and not an
Indian")
        End If
        'check if radio buttons >18 and Other are checked
        'if yes show You are not an Indian
        If rdbGreater18.Checked And rdbOther.Checked Then
            MsgBox(" You are not an Indian")
        End If
        'check if radio buttons <18 and Indian are checked
        'if yes show You are too young  to vote
        If rdbLess18.Checked And rdbIndian.Checked Then
            MsgBox(" You are too young  to vote")
        End If
    End Sub
```

ListBoxControl

ListBox Control is used to display a list of items, from which the user can select single or multiple choices by clicking on them. It allows the programmer to add items at design time by using the properties window, or at the runtime. If there are too many items to display at once, a scroll bar automatically appears to let the user scroll through the list.

Following table lists the Properties,Events, Methods used commonly with the ***ListBox Control*** in Visual Basic.net 2008.

Properties:

Properties	Description

ColumnWidth	Property used to set or get the column width.
DrawMode	Property used to set or get the drawing mode for the listbox.
HorizontalExtent	Property used to set or get the width for scrolling horizontally.
HorizontalSCrollBar	Property used to set or get if a horizontal scroll bar is displayed.
ItemHeight	Property used to set or get items height.
Items	Property returns a collection of the items.
MultiColumn	Property used to set or get if multiple columns are supported.
ScrollAlwaysVisible	Property used to set or get if a vertical scroll bar is always shown.
SelectedIndex	Property used to set or get the index of the currently selected item.
SelectedIndices	Property used to get a collection with indices of all selected item.
SelectedItem	Property used to get a selected item.
SelectedItems	Property used to get a selected items.
SelectionMode	Property used to get or set the selection mode.

Methods:

Method	Description
ClearSelected	Method used to set the focus for control.
EndUpdate	Method used to resume visual updating of the list box.
FindString	Method used to find the first item that begins with the indicated string.
FindStringExact	Method used to find first item in the list box that matches the indicated string.

GetItemHeight	Method used to return the height of an item.
GetSelected	Method used to return true if the indicated item is selected.
IndexFromPoint	Method used to return the index of the item at the given coordinates.
SetSelected	Method used to select or deselect the idndicated item.

Events:

Events	Description
SelectedIndexChanged	Triggered when the SelectedIndex property has changed.

Example :

In this example, we will fill up a list box with items, retrieve the total number of items in the list box, sort the list box, remove some items and clear the entire list box.

Design the Form:

Add the following code in the code editor window:

```
Public Class Form1
   Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
      ' Set the caption bar text of the form.
      Me.Text = "ListBox Demo"
      ' creating multi-column and multiselect list box
      ListBox1.MultiColumn = True
      ListBox1.SelectionMode = SelectionMode.MultiExtended
   End Sub
   'populates the list
   Private Sub Button1_Click_1(sender As Object, e As EventArgs) _
          Handles Button1.Click
      ListBox1.Items.Add("Safety")
      ListBox1.Items.Add("Security")
      ListBox1.Items.Add("Governance")
      ListBox1.Items.Add("Good Music")
      ListBox1.Items.Add("Good Movies")
      ListBox1.Items.Add("Good Books")
      ListBox1.Items.Add("Education")
      ListBox1.Items.Add("Roads")
      ListBox1.Items.Add("Health")
      ListBox1.Items.Add("Food for all")
      ListBox1.Items.Add("Shelter for all")
      ListBox1.Items.Add("Industrialisation")
      ListBox1.Items.Add("Peace")
      ListBox1.Items.Add("Liberty")
      ListBox1.Items.Add("Freedom of Speech")
   End Sub
   'sorting the list
   Private Sub Button2_Click(sender As Object, e As EventArgs) _
          Handles Button2.Click
      ListBox1.Sorted = True
   End Sub
   'clears the list
   Private Sub Button3_Click(sender As Object, e As EventArgs) _
          Handles Button3.Click
      ListBox1.Items.Clear()
   End Sub
   'removing the selected item
   Private Sub Button4_Click(sender As Object, e As EventArgs) _
          Handles Button4.Click
      ListBox1.Items.Remove(ListBox1.SelectedItem.ToString)
   End Sub
   'counting the numer of items
   Private Sub Button5_Click(sender As Object, e As EventArgs) _
          Handles Button5.Click
      Label1.Text = ListBox1.Items.Count
   End Sub
   'displaying the selected item on the third label
   Private Sub ListBox1_SelectedIndexChanged(sender As Object, e As
EventArgs) _
           Handles ListBox1.SelectedIndexChanged
      Label3.Text = ListBox1.SelectedItem.ToString()
   End Sub
End Class
```

Date Time Picker Control

DateTimePicker Control is used select a date and time for a date range specified using a **MinDate** and **MaxDate** properties.

Date Time Picker Properties

Following tables list commonly used Properties, Methods, Events of the Date Time Picker Control in Visual Basic.net 2008.

Properties:

Properties	Description
CalendarFont	Property to set the font for calendar.
CalendarForeColor	Property to set or get the foreground color for the calendar.
CalendarMonthBackGround	Property to set or get the background color for the calendar month.
CalendarTitleBacKColor	Property to get or set background color for the calendar title.
CalendarTitleForeColor	Property to get or set foreground color for the calendar title.
BackColor	Property to set the background color for the control.
BindingContext	Property used to set or get the binding context for the control.
Checked	Property used to get or set a value indicating whether the Value property has been set a valid date/time value that can be updated.
Format	Property used to set or get the format of the date and time displayed.
MaxDate	Property used to set or get the maximum date that can be selected using the control.
MinDate	Property used to set or get the minimum date that can be selected using the control.
ShowCheckBox	Property used to set or get a value to decide whether to display a check box to the left of the

	selected date.
Value	Property used to set or get the date, time value assigned to the control.
Width	Property set width of the control.
RightToLeft	Property specifies a value to know whether the text appears from right to left.
Locked	Prevents the control being moved at design time.

Methods:

Method	Description
DoDragDrop	Method used to begin Drag Drop Operation.
Equals	Method used to check if two instances of an object are equal.
FindForm	Method used to retreive the form the control is on.
Focus	Method used to set the input focus on the control.
Hide	Method used to hide the control.
PointToScreen	Method used to calculate the location of the specified client point into screen coordinates.
ToString	Method used to convert this object into its equivalent String value.
Select	Method used to select the control.

Events:

Events	Description
DragDown	Triggered when the dropdown claendar appears.
CloseUp	Triggered when the dropdown calendar disappears.

FormatChanged	Triggered when the format property is changed.
ValueChanged	Triggered when the value property is chnaged.

Timer Control

Timer Control is used to set time intervals, this control is visible only at design time and not in the runtime.

Timer Control Properties

Following table lists the Timer control Properties, Events, Methods used commonly with the Timer Control in Visual Basic.net 2008.

Properties:

Properties	Description
Enabled	Property used to Get or set whether the timer is running.
Intrevale	Property used to set or get the time in millisecond between the timer clicks.

Methods:

Events	Description
Start	Method used to start timer.
Stop	Method used to stop timer.

Events:

Events	Description
Tick	Triggered when the time intreval has elapsed.

Picture Box Control

PictureBox Control is used to display an image of the file formats like png, gif, jpeg, bmp, wmf, ico, pcx, tga, tiff. Images can be loaded either at the design time or at runtime.

PictureBox Control Properties

Following table lists the Properties, Methods, Events used commonly with the Picture Box Control in Visual Basic.net 2008.

Properties:

Properties	Description
BackColor	Property gets or sets the bsckground color for the control.
SizeMode	Property used to Get or sets size options for the control.
BorderStyle	Property used to specify the border style for the control.
Font	Property to used to set or get the font style for the control.
Image	Property used to specify the image to be loaded either from a resource file or from a local location.
Visible	Property to used to specify whether to make the control visible.
WaitOnLoad	Property to used to wait till the a big image gets loaded.
Enabled	Property to used to enable or disable the control.

Events:

Events	Description
Resize	Triggered when the picture box is resized.
SizeModeChanged	Triggered when the SizeMode changes.

HScroll And VScroll Bar Control

VScroll Bar Control is used to view the portion outside the visible area by enabling scrolling vertically. HScroll Bar Control is used to view the portion outside the visible area by enabling scrolling horizontally. Both these ***scrollbar control*** consists of two arrows at end with a scroll bar.

Following table lists the ***Scrollbar Control Properties***,Events used commonly with the VScroll Bar and HScroll Bar Control in Visual Basic.net 2008.

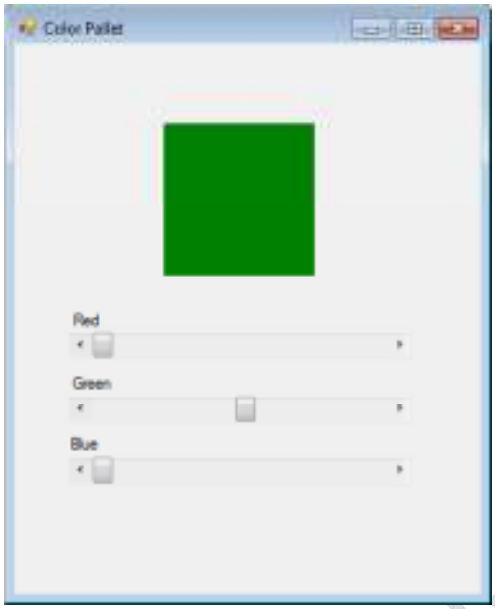
Properties:

Properties	Description
LargeChange	Property used to set or get the value added or subtracted from the value property when the scroll bar itself is clicked.
Maximum	Property used to set or get upper limit of the scrollable range.
Minimum	Property used to set or get lower limit of the scrollable range.
SmallChange	Property used to get or set the value added to or subtracted from the value property when the user clicks an arrow button.
Value	Property used to get or set value corresponding to current position of the scroll bar.

Events:

Events	Description
Scroll	Triggered when the scroll box is moved.
ValueChanged	Triggered when the Value property has changed, either by a scroll event or programmatically.

Example:



Property Chart

Control name	Property	Value
Form	Text	Color Pallet
PictureBox1	Name	pbPallet
HScrollBar1	Name	hsbRed
	Minimum	0
	Maximum	255
	Small Change	1
	Large Change	10
	Value	0
HScrollBar1	Name	hsbGreen
	Minimum	0
	Maximum	255
	Small Change	1
	Large Change	10
	Value	0
HScrollBar1	Name	hsbBlue

	Minimum	0
	Maximum	255
	Small Change	1
	Large Change	10
	Value	0
Label1	Text	Red
Label2	Text	Green
Label3	Text	Blue

Code:

```
Imports System.Drawing.Color

Public Class Assign7
    'Event Name: Scroll Event of hsbRed Control
    'Event handler: hsbRed_Scroll
    'Parameters: ByVal sender As System.Object, ByVal e As
System.Windows.Forms.ScrollEventArgs
    'Description: When the user scrolls the hsbRedControl
    Private Sub hsbRed_Scroll(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.ScrollEventArgs) Handles
hsbRed.Scroll
        pbPallet.BackColor = FromArgb(CInt(hsbRed.Value),
CInt(hsbGreen.Value), CInt(hsbBlue.Value))
    End Sub

    Private Sub hsbGreen_Scroll(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.ScrollEventArgs) Handles
hsbGreen.Scroll
        pbPallet.BackColor = FromArgb(CInt(hsbRed.Value),
CInt(hsbGreen.Value), CInt(hsbBlue.Value))
    End Sub

    Private Sub hsbBlue_Scroll(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.ScrollEventArgs) Handles
hsbBlue.Scroll
        pbPallet.BackColor = FromArgb(CInt(hsbRed.Value),
CInt(hsbGreen.Value), CInt(hsbBlue.Value))
    End Sub
End Class
```

Tooltip Control

Tooltip Control is used to display explanatory text when the mouse rest on a control or window. This control is helpful in providing a quick help to users.

Tool Tip Control Properties

Following table lists the Tooltip Control Properties, Events used commonly with the Tool Tip Control in Visual Basic.net 2008.

Properties:

Properties	Description
BackColor	Property used to Get or set background color for the control.
AutomaticDelay	Property used to set or get the automatic delsy for the tool tip.
AutoPopUpDelay	Property used to set a delay for automatic pop up of the control.
InitialDelay	Property used to set an initial delay for the control.
ToolTipIcon	Property used to choose the type of tool tip icon.
ShowAlways	Property used to get or set whether the tool tip should be appear when its parent control is inactive.

Events:

Events	Description
GetToolTip	Returns a tool tip text.
SetToolTip	Connects tool tip text with the tool tip.

Progress Bar Control

Progress Bar Control is used to display the progress of a task in Visual Basic. Mostly the progress bar is used while accessing the databases, or downloading or copying files from the network resources.

Progress Bar Control Properties

Following table lists the Properties, Events, Methods commonly used by the Progress Bar Control in Visual Basic.net 2008.

Properties:

Properties	Description
Font	This property sets the font properties for the control.
ForeColor	This property is used to set the forecolor for the object.
Maximum	This property is used to get or set the progress bars maximum value.
Minimum	This property is used to get or set the progress bars minimum value.
Value	Property is used to get or set the current value of the progress bar.
Increment	Property is used to increase the position of the progress bar by the given value.
PerformStep	Property is used to increment the value of the progress bar by the Steps property.

Methods:

Method	Description
Increment	Method used to increment the position of the progress bar by the given value.
PerformStep	Methods used to increment the value of the progress bar by Step property.

Month Calendar Control

This control is used to select a date in an application at runtime. Using this control calendar for one or more months can be displayed, so that a single date or a range of dates can be selected using this control.

Month Calendar Properties

Following tables lists the Properties, Events, Methods commonly used with the Month Calendar Control of Visual Basic.net 2008.

Properties:

Properties	Description
AnnuallyBoldedDate	Property holds an array of DateTime objects specifying which days should be bold.
BoldedDates	Property used to Get or sets an array of DateTime objects specifying which dates shpuld be bold.
CalendarDimensions	Property used to Get or set the number of columns.
FirstDayOfWeek	Property to get or set the first day of the week.
MaxDate	Property to get or set the maximum possible date.
MaxSelectionCount	Property holds the maximum number of days that can be selected.
MinDate	Property to get or set the minimum possible date.
ScrollChange	Property holds the scroll rate.
SelectionEnd	Property used to set or get the end date of the selected range.
SelectionRange	Property used to get the selected range of date from the control.
SelectionStart	Property used set or get the start date of the selected range.
SingleMonthSize	Property returns the minimum size in which to display a month.

ShowTodayCircle	Property set the todays date inside a circle.
TodaysDate	Property set or get the todays date.
TodayDateSet	Property specifies whether the Date Time property has been set.

Methods:

Method	Description
AddAnnuallyBoldedDate	Method used to add a day that will be displayed in bold annually.
AddBoldedDate	Method used to add a day that will be displayed in Bold.
AddMonthlyBoldedDate	Method used to add a day that will be displayed in bold monthly.
GetDisplayRange	Method used to get the date information that specifies the range displayed dates.
RemoveAllAnnuallyBoldedDates	Method used to remove all annually bolded dates.
RemoveAllBoldedDates	Method used to remove all non recurring bolded dates.
RemoveBoldedDate	Method used to remove a date from the calendars internal list o monthly bolded dates.
SetCalendarDimension	Method used to set the number of columns and rows.
SetDate	Method used to set the selected Date.
SetSelectionRange	Method used to set the selected dates to the given range of dates.
UpdateBoldedDates	Method used to redisplay the bolded dates.

Events:

Events	Description

Datechanged	Triggered when the date in the calendar control is changed.
DateSelected	Triggered when the control is double clicked.