VB.NET - PROGRESSBAR CONTROL

It represents a Windows progress bar control. It is used to provide visual feedback to your users about the status of some task. It shows a bar that fills in from left to right as the operation progresses.

Let's click on a ProgressBar control from the Toolbox and place it on the form.
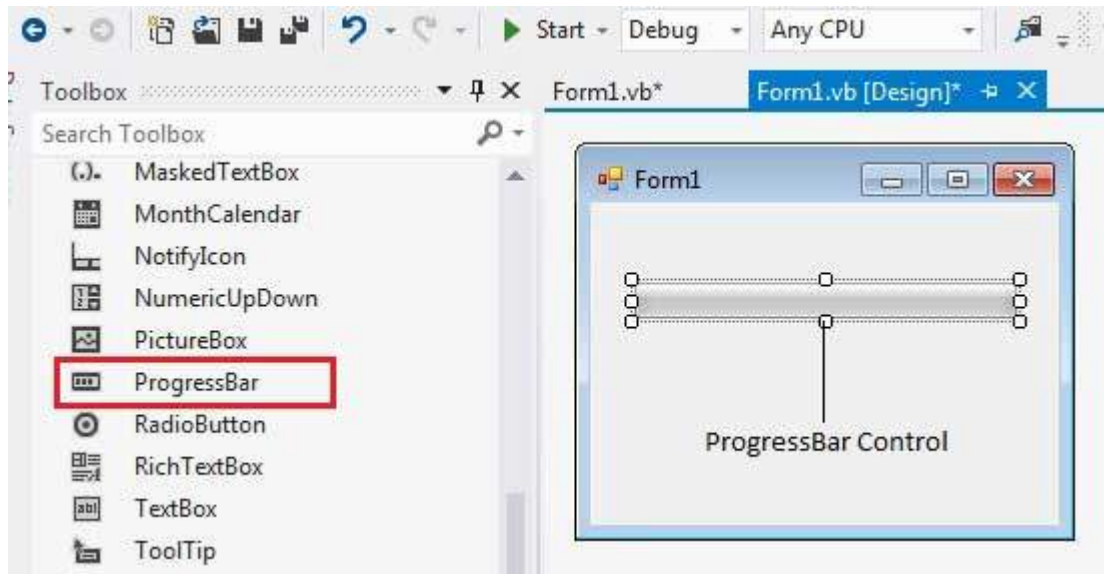
The main properties of a progress bar are *Value*, *Maximum* and *Minimum*. The Minimum and Maximum properties are used to set the minimum and maximum values that the progress bar can display. The Value property specifies the current position of the progress bar.

The ProgressBar control is typically used when an application performs tasks such as copying files or printing documents. To a user the application might look unresponsive if there is no visual cue. In such cases, using the ProgressBar allows the programmer to provide a visual status of progress.

Properties of the ProgressBar Control

The following are some of the commonly used properties of the ProgressBar control:

S.N	Property	Description
1	**AllowDrop**	Overrides Control.AllowDrop.
2	**BackgroundImage**	Gets or sets the background image for the ProgressBar control.
3	**BackgroundImageLayout**	Gets or sets the layout of the background image of the progress bar.
4	**CausesValidation**	Gets or sets a value indicating whether the control, when it receives focus, causes validation to be performed on any controls that require validation.
5	**Font**	Gets or sets the font of text in the ProgressBar.
6	**ImeMode**	Gets or sets the input method editor *IME* for the ProgressBar.
7	**ImeModeBase**	Gets or sets the IME mode of a control.
8	**MarqueeAnimationSpeed**	Gets or sets the time period, in milliseconds, that it takes the progress block to scroll across the progress bar.

9	**Maximum**	Gets or sets the maximum value of the range of the control.
10	**Minimum**	Gets or sets the minimum value of the range of the control.
11	**Padding**	Gets or sets the space between the edges of a ProgressBar control and its contents.
12	**RightToLeftLayout**	Gets or sets a value indicating whether the ProgressBar and any text it contains is displayed from right to left.
13	**Step**	Gets or sets the amount by which a call to the PerformStep method increases the current position of the progress bar.
14	**Style**	Gets or sets the manner in which progress should be indicated on the progress bar.
15	**Value**	Gets or sets the current position of the progress bar.

Methods of the ProgressBar Control

The following are some of the commonly used methods of the ProgressBar control:

S.N	Method Name & Description
1	**Increment** Increments the current position of the ProgressBar control by specified amount.
2	**PerformStep** Increments the value by the specified step.
3	**ResetText** Resets the Text property to its default value.
4	**ToString** Returns a string that represents the progress bar control.

Events of the ProgressBar Control

The following are some of the commonly used events of the ProgressBar control:

S.N	Event	Description
1	**BackgroundImageChanged**	Occurs when the value of the BackgroundImage property changes.
2	**BackgroundImageLayoutChanged**	Occurs when the value of the BackgroundImageLayout property changes.
3	**CausesValidationChanged**	Occurs when the value of the CausesValidation property changes.

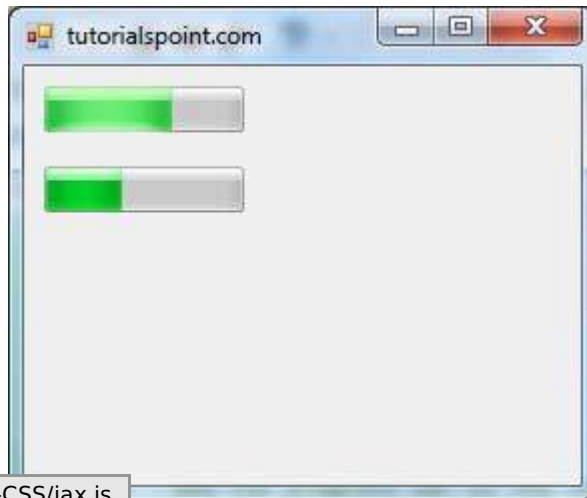| 4	**Click**	Occurs when the control is clicked.
5	**DoubleClick**	Occurs when the user double-clicks the control.
6	**Enter**	Occurs when focus enters the control.
7	**FontChanged**	Occurs when the value of the Font property changes.
8	**ImeModeChanged**	Occurs when the value of the ImeMode property changes.
9	**KeyDown**	Occurs when the user presses a key while the control has focus.
10	**KeyPress**	Occurs when the user presses a key while the control has focus.
11	**KeyUp**	Occurs when the user releases a key while the control has focus.
12	**Leave**	Occurs when focus leaves the ProgressBar control.
13	**MouseClick**	Occurs when the control is clicked by the mouse.
14	**MouseDoubleClick**	Occurs when the user double-clicks the control.
15	**PaddingChanged**	Occurs when the value of the Padding property changes.
16	**Paint**	Occurs when the ProgressBar is drawn.
17	**RightToLeftLayoutChanged**	Occurs when the RightToLeftLayout property changes.
18	**TabStopChanged**	Occurs when the TabStop property changes.
18	**TextChanged**	Occurs when the Text property changes.

Example

In this example, let us create a progress bar at runtime. Let's double click on the Form and put the follow code in the opened window.

```vbnet
Public Class Form1
   Private Sub Form1_Load(sender As Object, e As EventArgs) _
      Handles MyBase.Load
      'create two progress bars
      Dim ProgressBar1 As ProgressBar
      Dim ProgressBar2 As ProgressBar
      ProgressBar1 = New ProgressBar()
      ProgressBar2 = New ProgressBar()
      'set position
      ProgressBar1.Location = New Point(10, 10)
      ProgressBar2.Location = New Point(10, 50)
      'set values
      ProgressBar1.Minimum = 0
      ProgressBar1.Maximum = 200
      ProgressBar1.Value = 130
      ProgressBar2.Minimum = 0
      ProgressBar2.Maximum = 100
      ProgressBar2.Value = 40
      'add the progress bar to the form
      Me.Controls.Add(ProgressBar1)
      Me.Controls.Add(ProgressBar2)
      ' Set the caption bar text of the form.
      Me.Text = "tutorialspoint.com"
```

```
    End Sub
End Class
```

When the above code is executed and run using **Start** button available at the Microsoft Visual Studio tool bar, it will show the following window: