

VB.NET - BASIC CONTROLS

http://www.tutorialspoint.com/vb.net/vb.net_basic_controls.htm

Copyright © tutorialspoint.com

An object is a type of user interface element you create on a Visual Basic form by using a toolbox control. In fact, in Visual Basic, the form itself is an object. Every Visual Basic control consists of three important elements:

- **Properties** which describe the object,
- **Methods** cause an object to do something and
- **Events** are what happens when an object does something.

Control Properties

All the Visual Basic Objects can be moved, resized or customized by setting their properties. A property is a value or characteristic held by a Visual Basic object, such as Caption or Fore Color.

Properties can be set at design time by using the Properties window or at run time by using statements in the program code.

```
Object. Property = Value
```

Where

- **Object** is the name of the object you're customizing.
- **Property** is the characteristic you want to change.
- **Value** is the new property setting.

For example,

```
Form1.Caption = "Hello"
```

You can set any of the form properties using Properties Window. Most of the properties can be set or read during application execution. You can refer to Microsoft documentation for a complete list of properties associated with different controls and restrictions applied to them.

Control Methods

A method is a procedure created as a member of a class and they cause an object to do something. Methods are used to access or manipulate the characteristics of an object or a variable. There are mainly two categories of methods you will use in your classes:

- If you are using a control such as one of those provided by the Toolbox, you can call any of its public methods. The requirements of such a method depend on the class being used.
- If none of the existing methods can perform your desired task, you can add a method to a class.

For example, the *MessageBox* control has a method named *Show*, which is called in the code snippet below:

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
        Handles Button1.Click
            MessageBox.Show("Hello, World")
        End Sub
End Class
```

Control Events

An event is a signal that informs an application that something important has occurred. For example, when a user clicks a control on a form, the form can raise a **Click** event and call a procedure that handles the event. There are various types of events associated with a Form like click, double click, close, load, resize, etc.

Following is the default structure of a form **Load** event handler subroutine. You can see this code by double clicking the code which will give you a complete list of the all events associated with Form control:

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    'event handler code goes here
End Sub
```

Here, **Handles MyBase.Load** indicates that **Form1_Load** subroutine handles **Load** event. Similar way, you can check stub code for click, double click. If you want to initialize some variables like properties, etc., then you will keep such code inside Form1_Load subroutine. Here, important point to note is the name of the event handler, which is by default Form1_Load, but you can change this name based on your naming convention you use in your application programming.

Basic Controls

VB.Net provides a huge variety of controls that help you to create rich user interface. Functionalities of all these controls are defined in the respective control classes. The control classes are defined in the **System.Windows.Forms** namespace.

The following table lists some of the commonly used controls:

| S.N. | Widget & Description |
|------|--|
| 1 | Forms The container for all the controls that make up the user interface. |
| 2 | TextBox It represents a Windows text box control. |
| 3 | Label It represents a standard Windows label. |
| 4 | Button It represents a Windows button control. |
| 5 | ListBox It represents a Windows control to display a list of items. |
| 6 | ComboBox It represents a Windows combo box control. |
| 7 | |

[RadioButton](#)

It enables the user to select a single option from a group of choices when paired with other RadioButton controls.

8

[CheckBox](#)

It represents a Windows CheckBox.

9

[PictureBox](#)

It represents a Windows picture box control for displaying an image.

10

[ProgressBar](#)

It represents a Windows progress bar control.

11

[ScrollBar](#)

It Implements the basic functionality of a scroll bar control.

12

[DateTimePicker](#)

It represents a Windows control that allows the user to select a date and a time and to display the date and time with a specified format.

13

[TreeView](#)

It displays a hierarchical collection of labeled items, each represented by a TreeNode.

14

[ListView](#)

It represents a Windows list view control, which displays a collection of items that can be displayed using one of four different views.