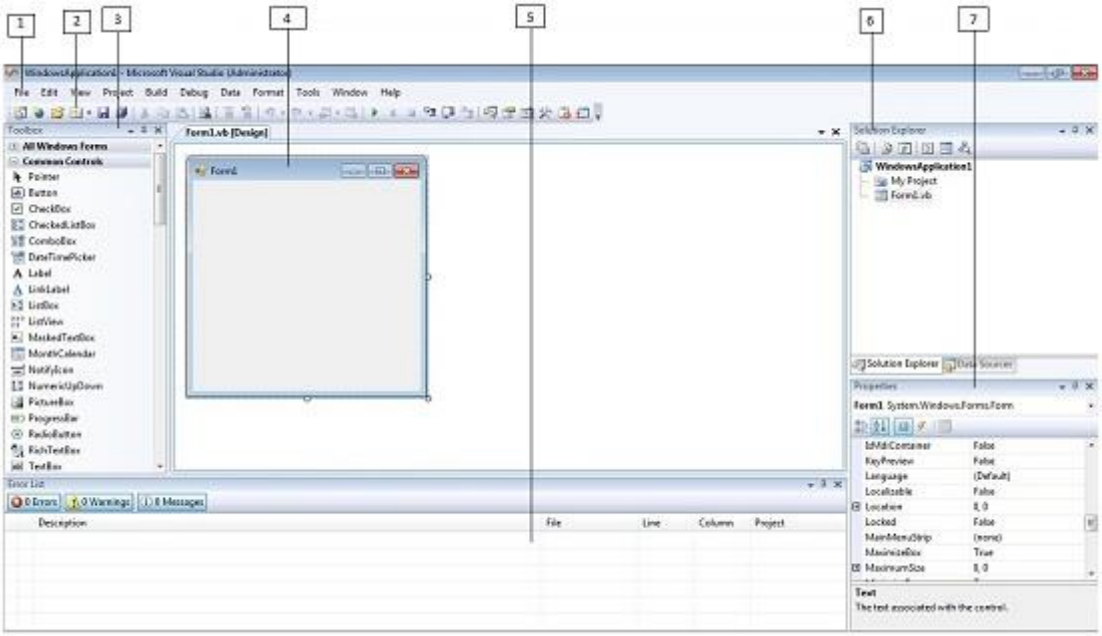
Visual Studio IDE

Visual Studio is a powerful and customizable programming environment that contains all the tools you need to build programs quickly and efficiently. It offers a set of tools that help you write and modify the code for your programs, and also detect and correct errors in your programs.

Before you start learning more about VB.NET programming, it is important to understand the development environment and identify some of the frequently using programming tools in Visual Studio IDE.



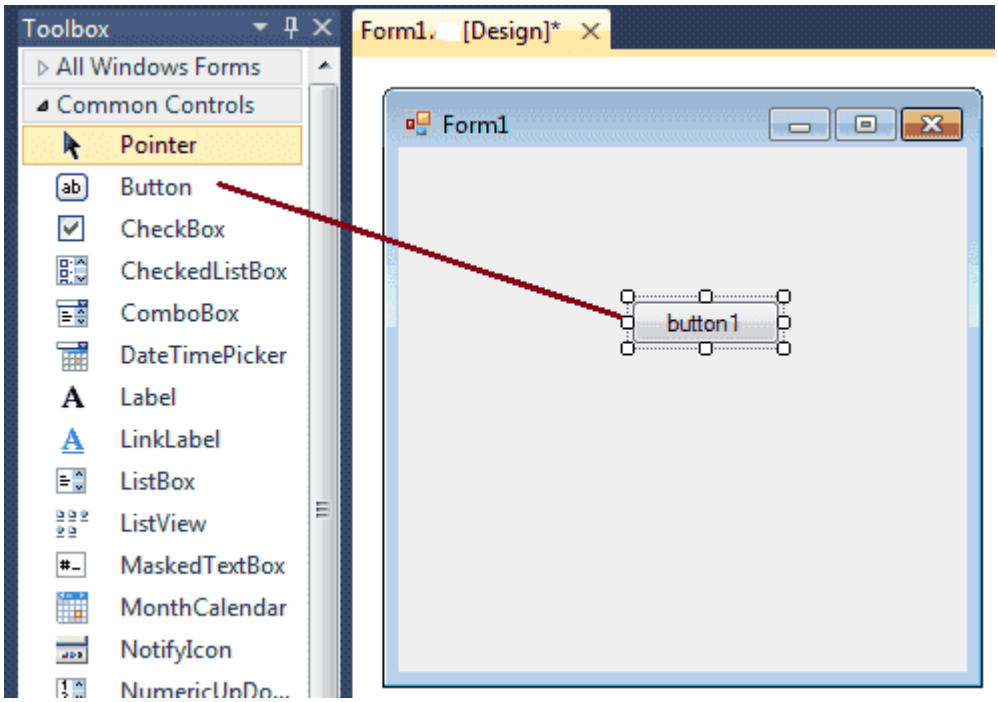
- 1. Menu Bar
- 2. Standard Toolbar
- 3. ToolBox
- 4. Forms Designer
- 5. Output Window
- 6. Solution Explorer
- 7. Properties Window

Visual Basic.NET IDE is built out of a collection of different windows. Some windows are used for writing code, some for designing interfaces, and others for getting a general overview of files or classes in your application.

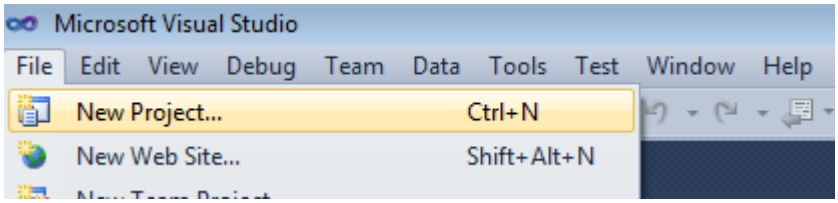
Visual Studio organizes your work in projects and solutions. A solution can contain more than one project, such as a DLL and an executable that references that DLL. From the following chapters you will learn how to use these Visual Studio features for your programming needs.

Windows Forms

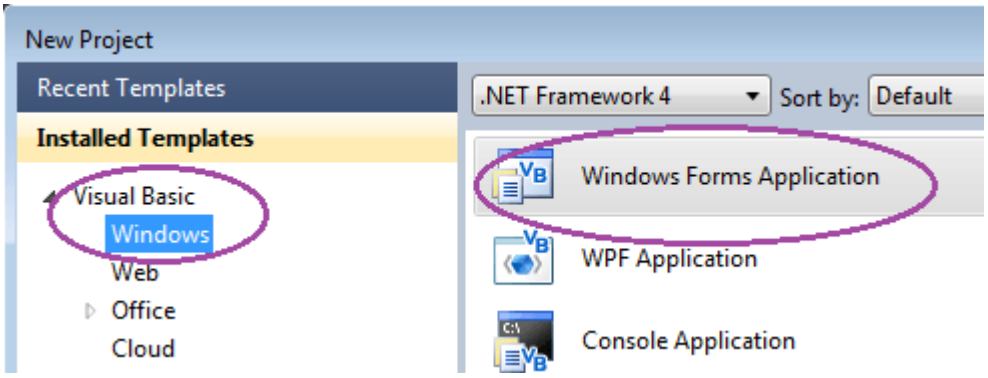
VB.Net programmers have made extensive use of forms to build user interfaces. Each time you create a Windows application, Visual Studio will display a default blank form, onto which you can drag and drop controls from the Visual Studio Toolbox window.



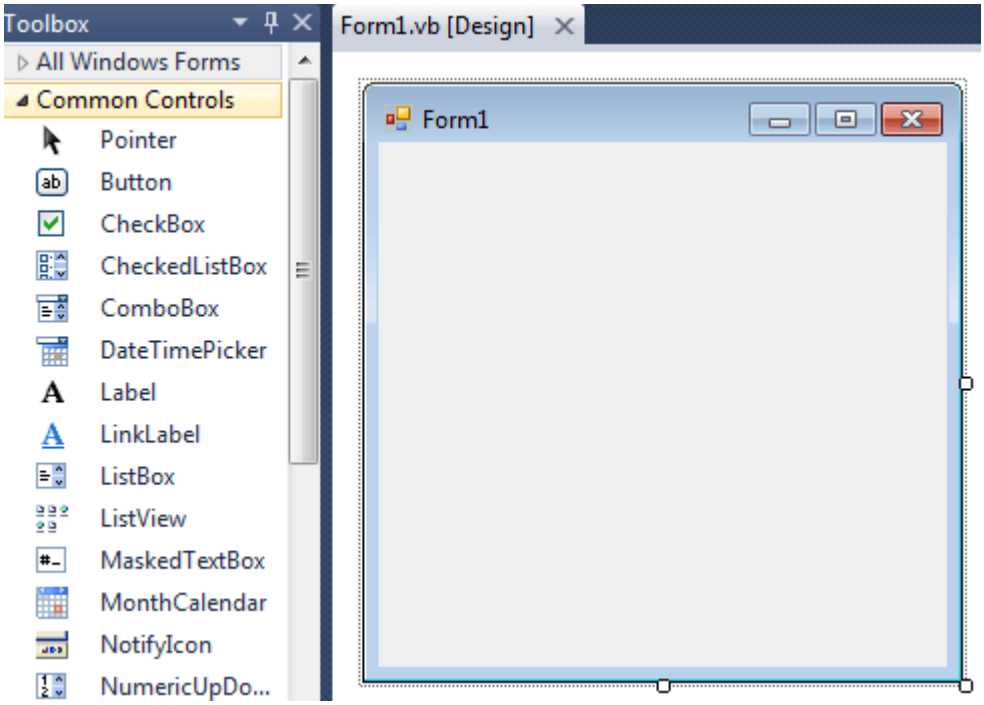
The first step is to start a new project and build a form. Open your Visual Studio and select File->NewProject and select Visual Basic from the New project dialog box and select Windows Froms Application. Enter your project name instead of WindowsApplication1 in the bottom of dialouge box and click OK button. The following picture shows how to crate a new Form in Visual Studio.



Select project type from New project dialog Box.

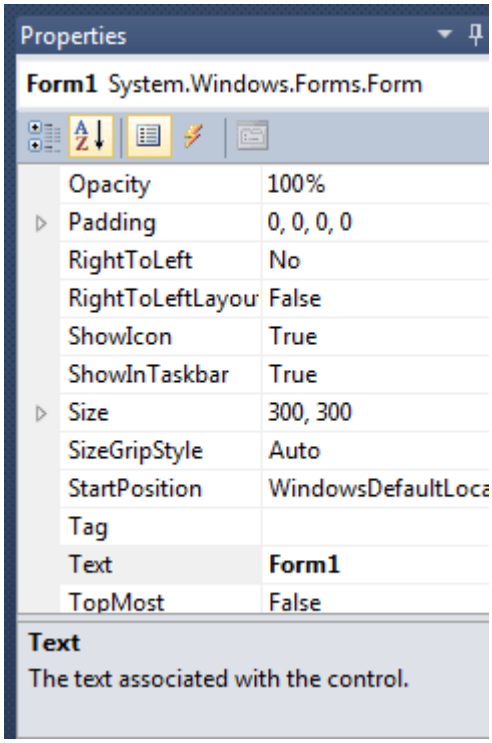


When you add a Windows Form to your project, many of the forms properties are set by default. Although these values are convenient, they will not always suit your programming needs. The following picture shows how is the default Form look like.



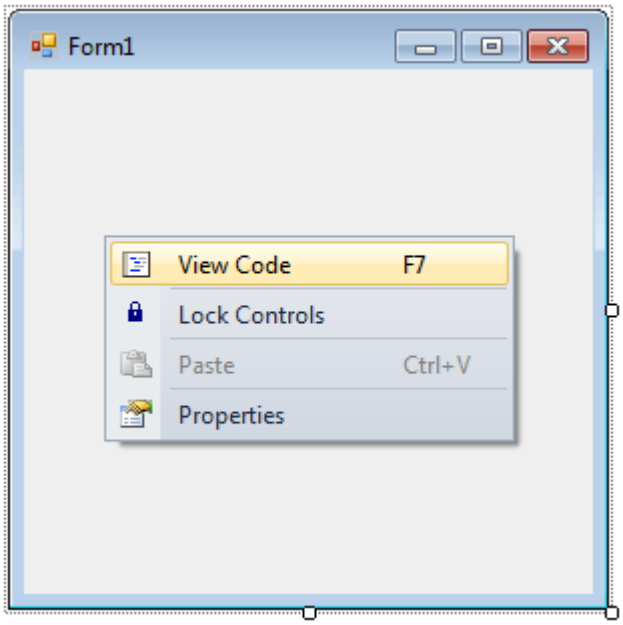
At the top of the form there is a title bar which displays the forms title. Form1 is the default name, you can change the name to your convenience . The title bar also includes the control box, which holds the minimize, maximize, and close buttons.

If you want to set any properties of the Form, you can use Visual Studio Property window to change it.



For example , to change the forms title from Form1 to MyForm, click on Form1 and move to the right side down Properties window, set Text property to MyForm. Then you can see the Title of the form is changed. Likewise you can set any properties of Form through Properties window.

You can also set the properties of the Form1 through coding. For coding, you should right-click the design surface or code window and then clicking View Code.



When you right click on Form then you will get code behind window, there you can write your code.

For example , if you want to change the back color of the form to Brown , you can code like this.

```
Private Sub Form1_Load(ByVal sender As System.Object,

  ByVal e As System.EventArgs) Handles MyBase.Load

    Me.BackColor = Color.Brown

End Sub
```

Label Control

Microsoft Visual Studio .NET controls are the graphical tools you use to build the user interface of a VB.Net program. Labels are one of the most frequently used Visual Basic control.

A Label control lets you place descriptive text , where the text does not need to be changed by the user. The Label class is defined in the System.Windows.Forms namespace.

Add a Label control to the form. Click Label in the Toolbox and drag it over the forms Designer and drop it in the desired location.

If you want to change the display text of the Label, you have to set a new text to the Text property of Label.

```
Label1.Text = "This is my first Label"
```

You can load Image in Label control , if you want to load an Image in the Lable control you can code like this

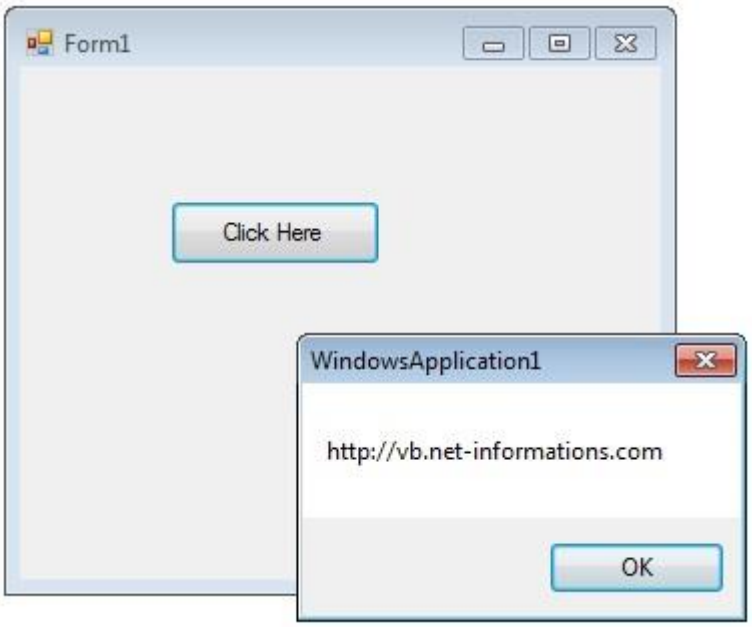
```
Label1.Image = Image.FromFile("C:\testimage.jpg")
```

The following source code shows how to set some properties of the Label through coding.

```
Public Class Form1
   Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
      Label1.Text = "This is my first Label"
      Label1.BorderStyle = BorderStyle.FixedSingle
      Label1.TextAlign = ContentAlignment.MiddleCenter
   End Sub
End Class
```

Button Control

Windows Forms controls are reusable components that encapsulate user interface functionality and are used in client side Windows applications. A Button is a control, which is an interactive component that enables users to communicate with an application which we click and release to perform some actions.



The Button control represents a standard button that reacts to a Click event. A Button can be clicked by using the mouse, ENTER key, or SPACEBAR if the button has focus.

When you want to change display text of the Button , you can change the Text property of the button.

```
Button1.Text = "My first Button"
```

Similarly if you want to load an Image to a Button control , you can code like this.

```
Button1.Image = Image.FromFile("C:\testimage.jpg")
```

The following vb.net source code shows how to change the button Text property while Form loading event and to display a message box when pressing a Button Control.

```
Public Class Form1

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Button1.Text = "Click Here"
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        MsgBox("http://vb.net-informations.com")
    End Sub

End Class
```

TextBox Control

VB.Net provides several mechanisms for gathering input in a program. A TextBox control is used to display, or accept as input, a single line of text.

VB.Net programmers make extensive use of the TextBox control to let the user view or enter large amount of text. A text box object is used to display text on a form or to get user input while a VB.Net program is running. In a text box, a user can type data or paste it into the control from the clipboard.

For displaying a text in a TextBox control , you can code like this.

```
TextBox1.Text = "Welcome"
```

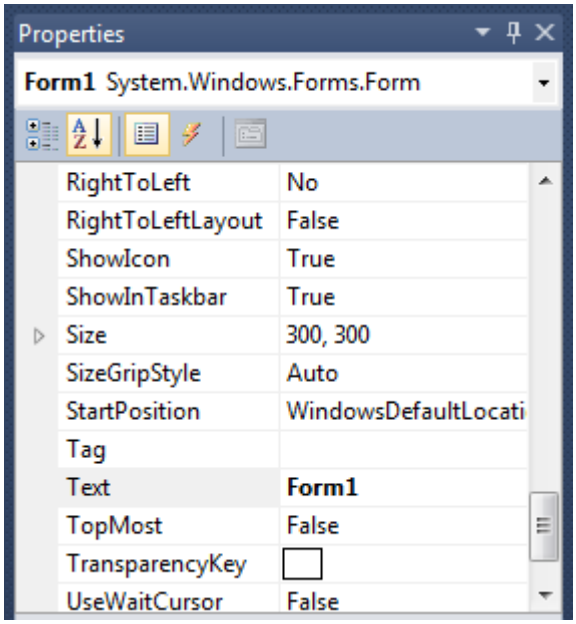
You can also collect the input value from a TextBox control to a variable like this way.

```
Dim var As String
```

var = TextBox1.Text

VB.Net TextBox Properties

You can set TextBox properties through Property window or through program. Normally Property window is located under the solution explorer. You can open Properties window by pressing F4 or right click on a control and select Properties menu item.



The below code set a textbox width as 150 and height as 25 through source code.

TextBox1.Width = 150

TextBox1.Height = 25

Background Color and Foreground Color

You can set Textbox background color and foreground color through property window, also you can set it programmatically.

TextBox1.BackColor = Color.Red

TextBox1.ForeColor = Color.Gray

TextBox BorderStyle

You can set 3 different types of border style for textbox in vb.net, they are None, FixedSingle and fixed3d.

TextBox1.BorderStyle = BorderStyle.FixedSingle

VB.Net Textbox Maximum Length

Maximum Length property sets the maximum number of characters or words the user can input into the text box control. That means you can limit the user input by this property.

TextBox1.MaxLength = 25

How to ReadOnly Textbox

when a program wants to prevent a user from changing the text that appears in a text box, the program can set the controls Readonly property is to True.

TextBox1.ReadOnly = True

Multiline TextBox in vb.net

By default TextBox accept single line of characters , If you need to enter more than one line in a TextBox control, you should change the Multiline property is to True.

TextBox1.Multiline = True

Textbox passowrd character

Sometimes you want a textbox to receive password from the user. In order to keep the password confidential, you can set the PasswordChar property of a textbox to a specific character.

TextBox1.PasswordChar = "*"

The above code set the PasswordChar to * , so when the user enter password then it display only * instead of typed characters.

How to Newline in TextBox

You can add new line in a textbox using two ways.

TextBox1.Text += "some text here" + "\r\n"

Or

TextBox1.Text += "some text here" + Environment.NewLine

VB.Net TextBox Events

TextBox Keydown event

Keydown event occurs when a key is pressed while the control has focus.

e.g.

```
Private Sub TextBox1_KeyDown(ByVal sender As System.Object, ByVal e As System.Windows.Forms.KeyEventArgs) Handles TextBox1.KeyDown

    If e.KeyCode = Keys.Enter Then

        MessageBox.Show("Enter key pressed")

    ElseIf e.KeyCode = Keys.Escape Then

        MessageBox.Show("Escape key pressed")

    End If

End Sub
```

TextChanged Event

TextChanged Event is raised if the Text property is changed by either through program modification or user input.

e.g.

```
Private Sub TextBox1_TextChanged(ByVal sender As System.Object,

        ByVal e As System.EventArgs) Handles TextBox1.TextChanged

  Label1.Text = TextBox1.Text

End Sub
```

How to retrieve integer values from textbox ?

VB.Net String to Integer conversion

```
Dim i As Integer

i = Integer.Parse(TextBox1.Text)
```

Parse method Converts the string representation of a number to its integer equivalent.

VB.Net String to Double conversion

```
Dim dbl As Double

dbl = Double.Parse(TextBox1.Text)
```

```
Public Class Form1

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        TextBox1.Width = 200
        TextBox1.Height = 50
        TextBox1.Multiline = True
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
        Dim var As String
        var = TextBox1.Text
        MsgBox(var)
    End Sub

End Class
```

How to allow only numbers in a textbox

```
Public Class Form1
  Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress
    If (Not Char.IsControl(e.KeyChar) _
        AndAlso (Not Char.IsDigit(e.KeyChar) _
        AndAlso (e.KeyChar <> Microsoft.VisualBasic.ChrW(46)))) Then
      e.Handled = True
    End If
  End Sub
  Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles TextBox1.TextChanged
```

```
        If System.Text.RegularExpressions.Regex.IsMatch(TextBox1.Text, "[  ^ 0-
9]") Then
            TextBox1.Text = ""
        End If
    End Sub
End Class
```