

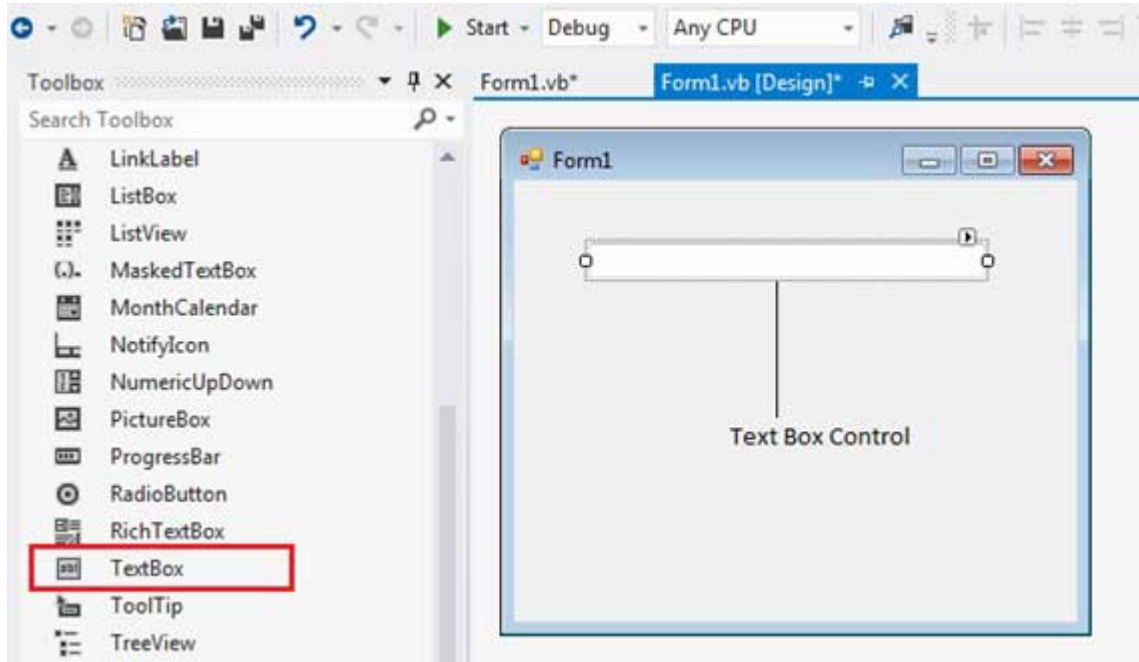
VB.NET - TEXTBOX CONTROL

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

Copyright © tutorialspoint.com

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:

S.N	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 AutoCompleteSource property 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: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • 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:

S.N	Method Name & Description
1	AppendText Append 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

PasteString

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.

Events of the TextBox Control

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

S.N	Event	Description
1	Click	Occurs when the control is clicked.
2	DoubleClick	Occurs when the control is double-clicked.
3	TextAlignChanged	Occurs when the TextAlign property value changes.

Example

In this example, we create three text boxes and use the Click event of a button to display the entered text using a message box. Take the following steps:

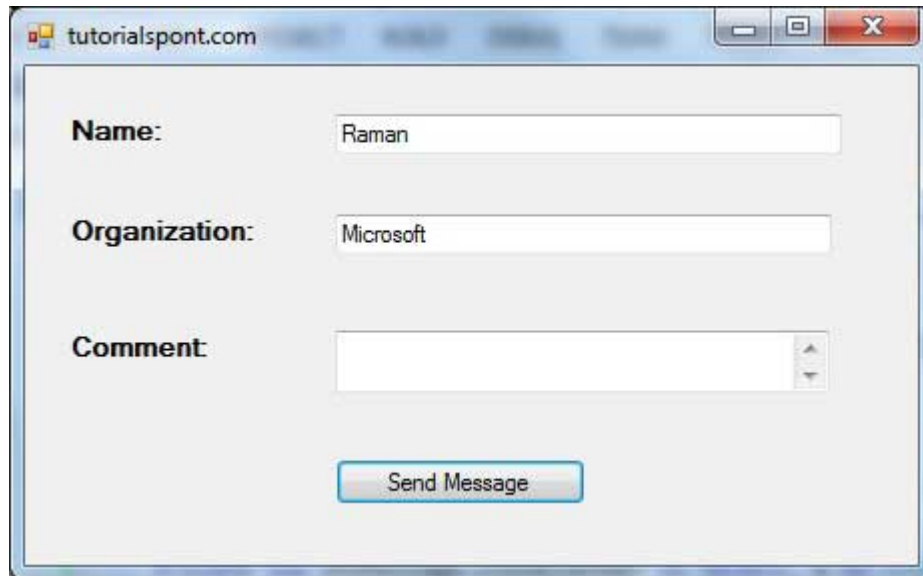
- Drag and drop three Label controls and three TextBox controls on the form.
- Change the texts on the labels to: Name, Organization and Comments, respectively.
- Change the names of the text boxes to txtName, txtOrg and txtComment, respectively.
- Drag and drop a button control on the form. Set its name to btnMessage and its text property to 'Send Message'.
- Click the button to add the Click event in the code window and add the following code.

```
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 = "tutorialspont.com"
End Sub

Private Sub btnMessage_Click(sender As Object, e As EventArgs) _
Handles btnMessage.Click
    MessageBox.Show("Thank you " + txtName.Text + " from " + txtOrg.Text)
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:



Clicking the Send Message button would show the following message box:



Loading [Mathjax]/jax/output/HTML-CSS/jax.js