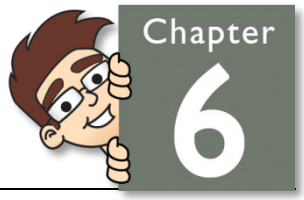
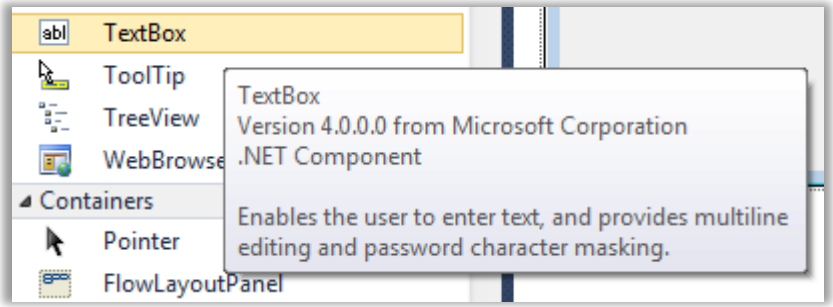


Chapter Six: User Input

In this chapter, we will take a more in-depth look at the different controls that are used to retrieve user input. You will learn about text boxes, list and combo boxes, and radio buttons and check boxes.

Lesson One: Text Boxes

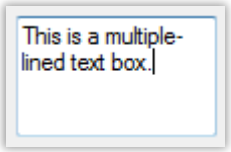
A common user input control is the text box, which allows the user to enter a line of text into a program. The text box control is found in the Toolbox pane of your IDE when you are looking at the Form Design tab. Look under the heading "Common Controls" to find the "TextBox" item.



To add this control to your form, you can double-click it in the Toolbox or click and drag it from the Toolbox to the Form Design screen. You can click on the control's outline and then drag the mouse to resize or move the text box on the form.

The C# data type representing a text box in code is called **TextBox**. There are many different properties for the **TextBox** control. To view the properties, you can click on the control and then look at the Property Sheet panel. If the panel is not shown on the screen, just right-click the text box within the form and choose "Properties" from the menu. Here is a list of some of the more important properties:

Property Name	Description
(Name)	This is the name of the control. You will use this name to access the control's information in a program, so make sure this is a descriptive name that follows variable-naming rules.
Text	This is the value displayed in the control. You can use this property at design time to set the default text for the control. At runtime, you can use this property to read in the text that the user has entered, or set this value to your own string.
Enabled	This Boolean value determines whether or not the user can use this control on the form. If it is set to **false**, the text box will be grayed-out and the user will not be able to use it.
Visible	This Boolean value determines whether or not the user can see this control on the form. If this value is **false**, the text box will not appear on the form
ForeColor and **BackColor**	These properties set the foreground (text) and background colors for the control

Multiline	This property determines if more than one line of text is allowed in the control. If **false**, only one line is permitted. If **true**, the text box can be resized to hold many lines.	

There are many other properties for the **TextBox** control. To see what any of these properties do, just click on the property name and then look at the descriptive text at the bottom of the Properties panel.

If you double-click on the text box control in the form design window, the form code window will appear and a **TextChanged**() event handler function will be automatically created for you.

```
private void textBox1_TextChanged(object sender, EventArgs e)
{

}
```

This function will be called any time the user changes the text in the box. This is useful when you need to process the user input character-by-character for some reason. Or, if you need to update another area of the form to reflect the text in a text box control, you will want to know when the user has changed that text.

Using the TextBox Variable

Every **TextBox** on the form has a **(Name)** property, and that property determines a variable name for the control within your code. In the example above we used "textBox1" as the **(Name)** property. You can access all of the control's properties in code by writing "**textBox1.<property name>**".

The **Text** property is used to get or set the value in a text box. To retrieve the information a user has typed in, you can set a **string** variable equal to the **Text** property:

```
string input = textBox1.Text;   // gets current textBox1 contents
```

To set or display information in the text box control, you would just set the **Text** property to a string:

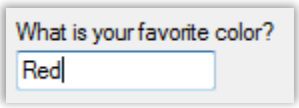
```
textBox1.Text = "Blue";         // set new textBox1 contents
```

These statements are valid from within any of your form methods, including the **TextChanged**() event handler demonstrated above and other event handlers such as button click events.

The text box control will hold only one line of text by default. If you want to change the height of the text box to hold more than one text line, you will need to find the **Multiline** property in the Property panel and

change its value to **true**. Then you can click and drag the control to change its height. If you do not change this **Multiline** property, you will not be able to resize the text box's height.

If you decide to use the text box control to hold multiple lines of text, or you think the user may enter more characters than can be shown on the screen, you may want to add scroll bars to the control. Scroll bars are a set of arrows with a slider in between. A text box can contain horizontal scroll bars, vertical scroll bars or both. Horizontal scroll bars will scroll the text left and right. Vertical scroll bars will scroll up and down. Scroll bars can be enabled or disabled from the Properties sheet.

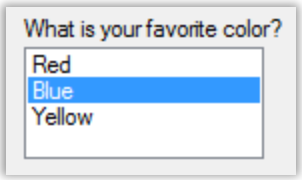


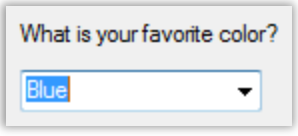
Finally, it is extremely good Windows design to use a descriptive label for any text box controls on your form. Without this, the user will not know what type of information you are expecting! So remember to drag a **Label** control from the Toolbox and position it above or to the side of the text box. Then change the **Text** property of the label with a descriptive string. You do not normally access labels from within the code at runtime. Therefore it is not usually necessary to give the label controls good variable names, because the variables will not be accessed from elsewhere in the program. If you want to change the labels while the program is running for some reason then go ahead and follow standard variable naming procedures.

Lesson Two: List Boxes and Combo Boxes

The list box control and the combo box control are useful user input controls. These controls both offer the user a choice of a list of items.

The list box control is used to display multiple items that can be selected by the user. A user can select items in the list by clicking on them. The list box can be handy when you want the user to make one or more selections from a pre-defined list of choices.





The combo box control is very similar to the list box control. Both controls allow a user to select from a list of items. A combo box, however, will only show the currently selected item in the text window. The rest of the list is hidden until the user hits a down arrow on the right. The first image shows the combo box control in its default state, while the second image shows the combo box's full drop-down list. The combo box is very useful in situations where you need to have the user select a single item from a list, but you do not have the space on your form to hold a full list box.

You will find the list box and combo box controls in the Toolbox under "Common Controls" just like the text box. Follow the same drag-n-drop procedure to add either control to your form.