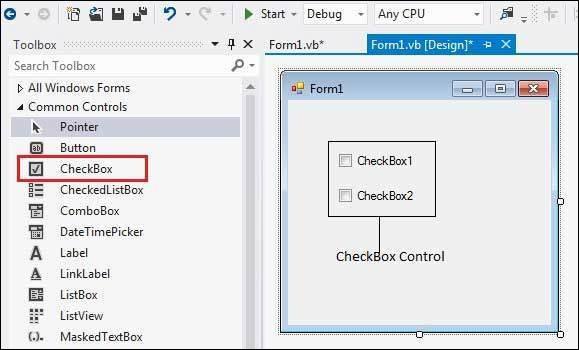
**CheckBox Control**

The CheckBox control allows the user to set true/false or yes/no type options. The user can select or deselect it. When a check box is selected it has the value True, and when it is cleared, it holds the value False.

Let's create two check boxes by dragging CheckBox controls from the Toolbox and dropping on the form.



The CheckBox control has three states, checked, unchecked and indeterminate. In the indeterminate state, the check box is grayed out. To enable the indeterminate state, the ThreeState property of the check box is set to be True.

**Properties of the CheckBox Control**

The following are some of the commonly used properties of the CheckBox control −

1. Appearance

Gets or sets a value determining the appearance of the check box.

2. AutoCheck

Gets or sets a value indicating whether the Checked or CheckedState value and the appearance of the control automatically change when the check box is selected.

3.CheckAlign

Gets or sets the horizontal and vertical alignment of the check mark on the check box.

4.Checked

Gets or sets a value indicating whether the check box is selected.

5.CheckState

Gets or sets the state of a check box.

6.Text

Gets or sets the caption of a check box.

7.ThreeState

Gets or sets a value indicating whether or not a check box should allow three check states rather than two.

**Events of the CheckBox Control**

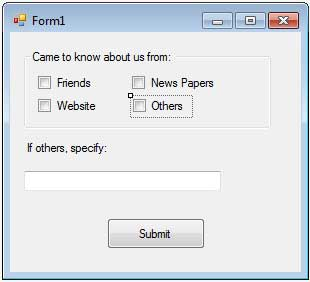
The following are some of the commonly used events of the CheckBox control −

|  |  |
| --- | --- |
| **Sr.No.** | **Event & Description** |
| 1 | **AppearanceChanged**  Occurs when the value of the Appearance property of the check box is changed. |
| 2 | **CheckedChanged**  Occurs when the value of the Checked property of the CheckBox control is changed. |
| 3 | **CheckStateChanged**  Occurs when the value of the CheckState property of the CheckBox control is changed. |

**Example**

In this example, let us add four check boxes in a group box. The check boxes will allow the users to choose the source from which they came to know about the organization. If the user chooses the check box with text "others", then the user is asked to specify and a text box is provided to give input. When the user clicks the Submit button, he/she gets an appropriate message.

The form in design view



Let's put the following code in the code editor window.

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 = "Checkbox Control"

Label1.Visible = False

TextBox1.Visible = False

TextBox1.Multiline = True

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) \_ Handles Button1.Click

Dim str As String

str = " "

If CheckBox1.Checked = True Then

str &= CheckBox1.Text

str &= " "

End If

If CheckBox2.Checked = True Then

str &= CheckBox2.Text

str &= " "

End If

If CheckBox3.Checked = True Then

str &= CheckBox3.Text

str &= " "

End If

If CheckBox4.Checked = True Then

str &= TextBox1.Text

str &= " "

End If

If str <> Nothing Then

MsgBox(str + vbLf + "Thank you")

End If

End Sub

Private Sub CheckBox4\_CheckedChanged(sender As Object, \_ e As EventArgs) Handles CheckBox4.CheckedChanged

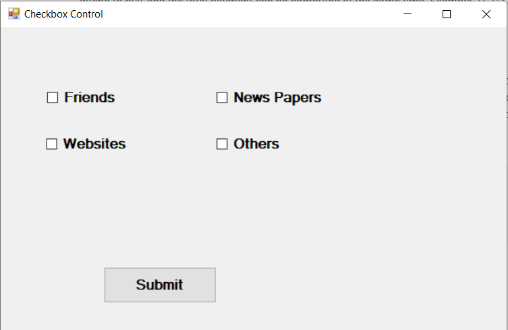
Label1.Visible = True

TextBox1.Visible = True

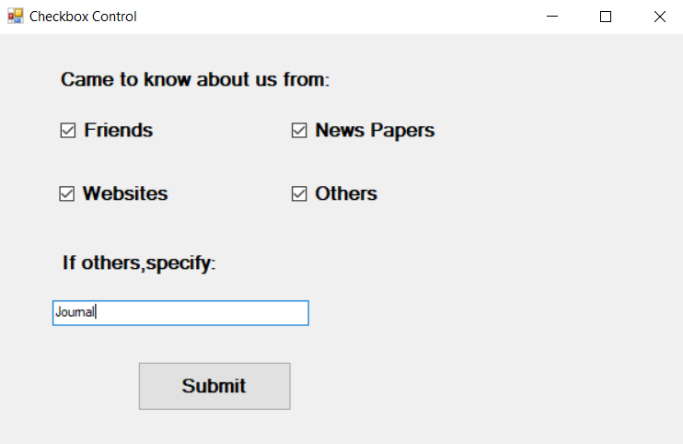
End Sub

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.



Checking all the boxes



Clicking the Submit button

