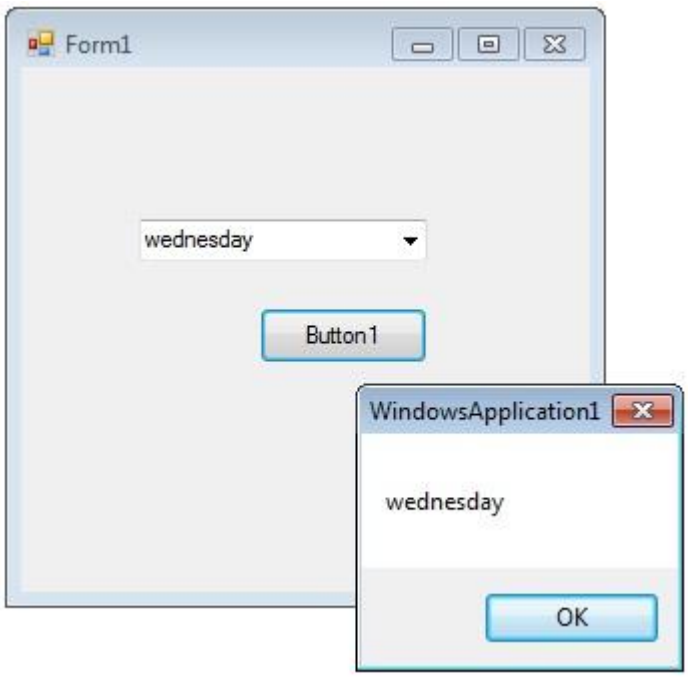
ComboBox Control

VB.Net controls are located in the Toolbox of the development environment, and you use them to create objects on a form with a simple series of mouse clicks and dragging motions. The ComboBox control , which lets the user choose one of several choices.



The user can type a value in the text field or click the button to display a drop down list. In addition to display and selection functionality, the ComboBox also provides features that enable you to efficiently add items to the ComboBox.

Add item to combobox

```
ComboBox1.Items.Add("Sunday")

ComboBox1.Items.Add("Monday")

ComboBox1.Items.Add("Tuesday")
```

How to set the selected item in a comboBox

You can set which item should shown while it displaying in the form for first time.

```
comboBox1.Items.Add("test1")

comboBox1.Items.Add("test2")

comboBox1.Items.Add("test3")

comboBox1.SelectedItem = "test3"
```

or

ComboBox1.SelectedItem = ComboBox1.Items(1)

or

comboBox1.SelectedIndex = comboBox1.FindStringExact("test3")

ComboBox SelectedItem
How to retrieve value from ComboBox

If you want to retrieve the displayed item to a string variable , you can code like this

Dim var As String

var = ComboBox1.Text

Or

Dim item = Me.comboBox1.GetItemText(Me.comboBox1.SelectedItem)

MessageBox.Show(item)

How to remove an item from ComboBox in VB.Net

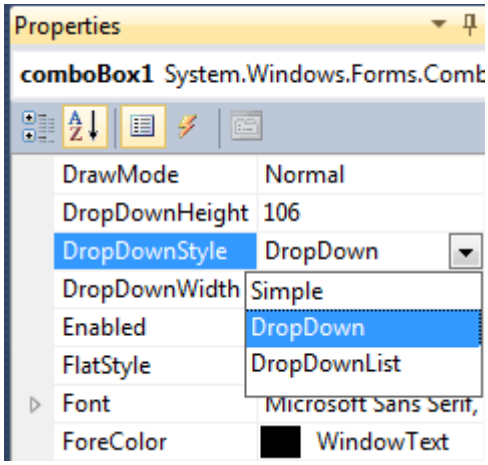
You can remove items from a combobox in two ways. You can remove item at a the specified index or giving a specified item by name.

ComboBox1.Items.RemoveAt(1)

Or

ComboBox1.Items.Remove("Friday")

DropDownStyle



The DropDownStyle property specifies whether the list is always displayed or whether the list is displayed in a drop down. The DropDownStyle property also specifies whether the text portion can be edited.

```
ComboBox1.DropDownStyle = ComboBoxStyle.DropDown
```

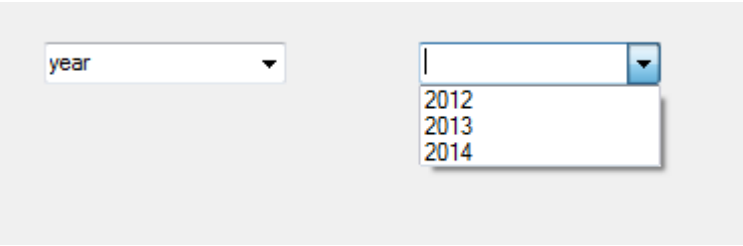
Combobox SelectedIndexChanged event

The SelectedIndexChanged event of a combobox fire when your selection change in the combobox. If you want some actionsg when you change the selection, you can write the code on SelectedIndexChanged event. From the following vb.net code you can understand how to set values in the SelectedIndexChanged event of a combobox. Drag and drop two combobox on the Form and copy and paste the following source code.

```
Public Class Form1

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        ComboBox1.Items.Add("weekdays")

        ComboBox1.Items.Add("year")

    End Sub

    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged

        ComboBox2.Items.Clear()

        If ComboBox1.SelectedItem = "weekdays" Then

            ComboBox2.Items.Add("Sunday")

            ComboBox2.Items.Add("Monday")

            ComboBox2.Items.Add("Tuesday")

        ElseIf ComboBox1.SelectedItem = "year" Then

            ComboBox2.Items.Add("2012")

            ComboBox2.Items.Add("2013")

            ComboBox2.Items.Add("2014")

        End If

    End Sub
```

End Class

Output



ComboBox Default Value

How to set a default value for a Combo Box

You can set combobox default value in VB.Net by using SelectedIndex property

comboBox1.SelectedIndex = 6

ComboBox readonly

How to make a combobox read only

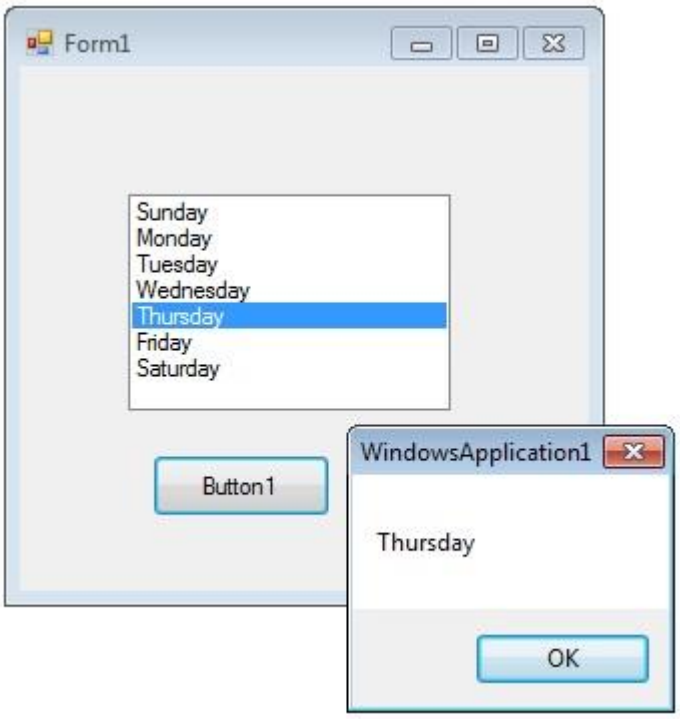
You can make a ComboBox readonly in VB.Net, that means a user cannot write in a combobox but he can select the given values, in two ways. By default, DropDownStyle property of a Combobox is DropDown. In this case user can enter values to combobox. When you change the DropDownStyle property to DropDownList, the Combobox will become read only and user can not enter values to combobox. Second method, if you want the combobox completely read only, you can set comboBox1.Enabled = false.

```
Public Class Form1
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        ComboBox1.Items.Add("Sunday")
        ComboBox1.Items.Add("Monday")
        ComboBox1.Items.Add("Tuesday")
        ComboBox1.Items.Add("wednesday")
        ComboBox1.Items.Add("Thursday")
        ComboBox1.Items.Add("Friday")
        ComboBox1.Items.Add("Saturday")

        ComboBox1.SelectedItem = ComboBox1.Items(3)
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
```

```
        Dim var As String
        var = ComboBox1.Text
        MsgBox(var)
    End Sub
End Class
```

ListBox Control

VB.Net provides several mechanisms for gathering input in a program. A Windows Forms ListBox control displays a list of choices which the user can select from.



You can use the Add or Insert method to add items to a list box. The Add method adds new items at the end of an unsorted list box. The Insert method allows you to specify where to insert the item you are adding.

```
ListBox1.Items.Add("Sunday")
```

The SelectionMode property determines how many items in the list can be selected at a time. A ListBox control can provide single or multiple selections using the SelectionMode property .

If you want to retrieve a single selected item to a variable , you can code like this.

```
Dim var As String
```

```
var = ListBox1.SelectedItem
```

If you change the selection mode property to multiple select , then you will retrieve a collection of items from ListBox1.SelectedItems property.

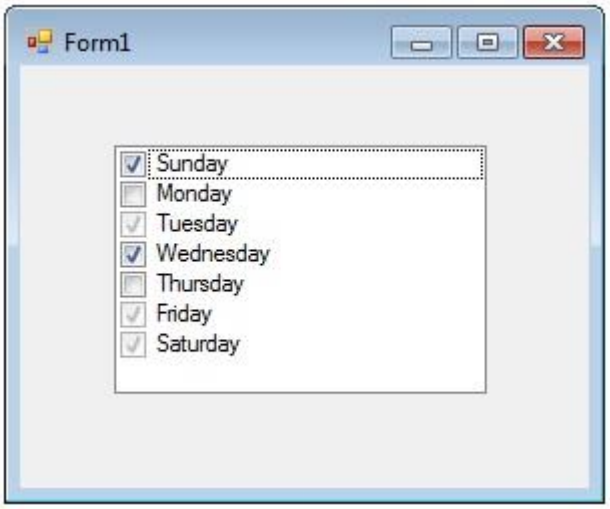
```
ListBox1.SelectionMode = SelectionMode.MultiSimple
```

The following VB.Net program initially fill seven days in a week while in the form load event and set the selection mode property to MultiSimple. At the Button click event it will display the selected items.

```
Public Class Form1
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        ListBox1.Items.Add("Sunday")
        ListBox1.Items.Add("Monday")
        ListBox1.Items.Add("Tuesday")
        ListBox1.Items.Add("Wednesday")
        ListBox1.Items.Add("Thursday")
        ListBox1.Items.Add("Friday")
        ListBox1.Items.Add("Saturday")

        ListBox1.SelectionMode = SelectionMode.MultiSimple
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

        Dim obj As Object
        For Each obj In ListBox1.SelectedItems
            MsgBox(obj.ToString)
        Next

    End Sub
End Class
```

Checked ListBox Control

The CheckedListBox control gives you all the capability of a list box and also allows you to display a check mark next to the items in the list box.



To add objects to the list at run time, assign an array of object references with the AddRange method. The list then displays the default string value for each object.

```
Dim days As String() =  {"Sunday", "Monday", "Tuesday"}

checkedListBox1.Items.AddRange(days)
```

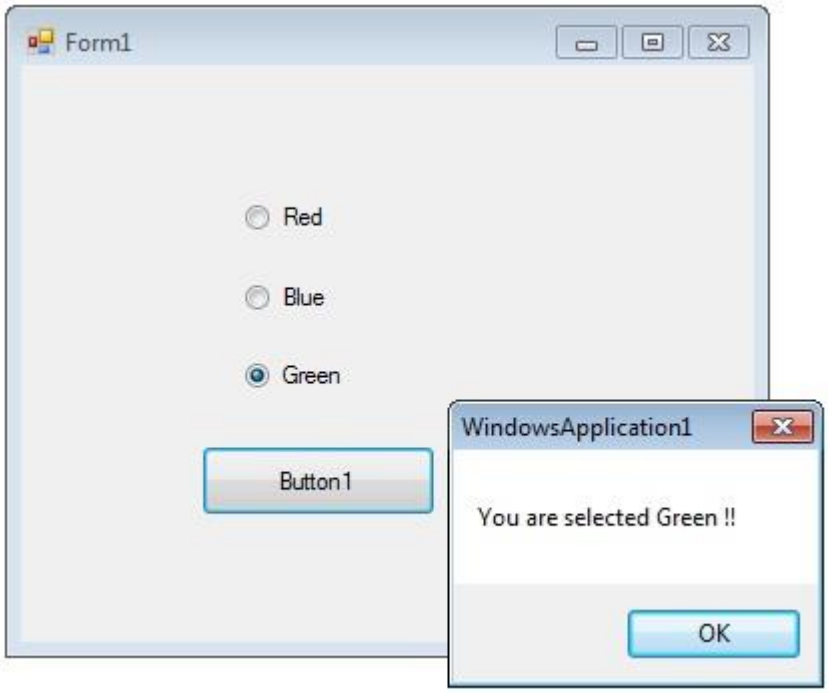
You can add individual items to the list with the Add method. The CheckedListBox object supports three states through the CheckState enumeration: Checked, Indeterminate, and Unchecked.

```
CheckedListBox1.Items.Add("Sunday", CheckState.Checked)

CheckedListBox1.Items.Add("Monday", CheckState.Unchecked)

CheckedListBox1.Items.Add("Tuesday", CheckState.Indeterminate)
```

```
Public Class Form1

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        CheckedListBox1.Items.Add("Sunday", CheckState.Checked)
        CheckedListBox1.Items.Add("Monday", CheckState.Unchecked)
        CheckedListBox1.Items.Add("Tuesday", CheckState.Indeterminate)
        CheckedListBox1.Items.Add("Wednesday", CheckState.Checked)
        CheckedListBox1.Items.Add("Thursday", CheckState.Unchecked)
        CheckedListBox1.Items.Add("Friday", CheckState.Indeterminate)
        CheckedListBox1.Items.Add("Saturday", CheckState.Indeterminate)
    End Sub

End Class
```

RadioButton Control

A radio button or option button is a type of graphical user interface element that allows the user to choose only one of a predefined set of options. When a user clicks on a radio button, it becomes checked, and all other radio buttons with same group become unchecked.



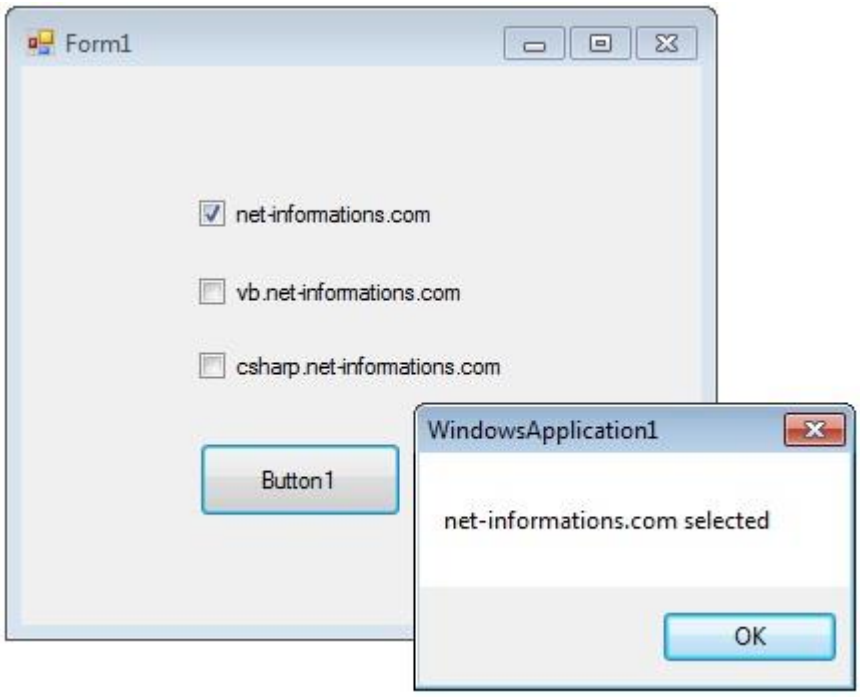
The radio button and the check box are used for different functions. Use a radio button when you want the user to choose only one option. When you want the user to choose all appropriate options, use a check box. Like check boxes, radio buttons support a Checked property that indicates whether the radio button is selected.

```
Public Class Form1

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        RadioButton1.Checked = True
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        If RadioButton1.Checked = True Then
            MsgBox("You are selected Red !! ")
            Exit Sub
        ElseIf RadioButton2.Checked = True Then
            MsgBox("You are selected Blue !! ")
```

```
        Exit Sub
      Else
        MsgBox("You are selected Green !! ")
        Exit Sub
      End If
    End Sub

End Class
```

CheckBox Control

CheckBoxes allow the user to make multiple selections from a number of options. You can click a check box to select it and click it again to deselect it.



CheckBoxes comes with a caption, which you can set in the Text property.

CheckBox1.Text = "Welcome"

You can use the CheckBox control ThreeState property to direct the control to return the Checked, Unchecked, and Indeterminate values. You need to set the check boxs ThreeState property to True to indicate that you want it to support three states.

CheckBox1.ThreeState = True

To apply the same property settings to multiple CheckBox controls, use the Style property. The following VB.Net program shows how to find a checkbox is selected or not.

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim msg As String = ""

        If CheckBox1.Checked = True Then
            msg = "net-informations.com"
        End If

        If CheckBox2.Checked = True Then
            msg = msg & "   vb.net-informations.com"
        End If

        If CheckBox3.Checked = True Then
            msg = msg & "   csharp.net-informations.com"
        End If


        If msg.Length > 0 Then
            MsgBox(msg & " selected ")
        Else
            MsgBox("No checkbox selected")
        End If

        CheckBox1.ThreeState = True
    End Sub

End Class
```