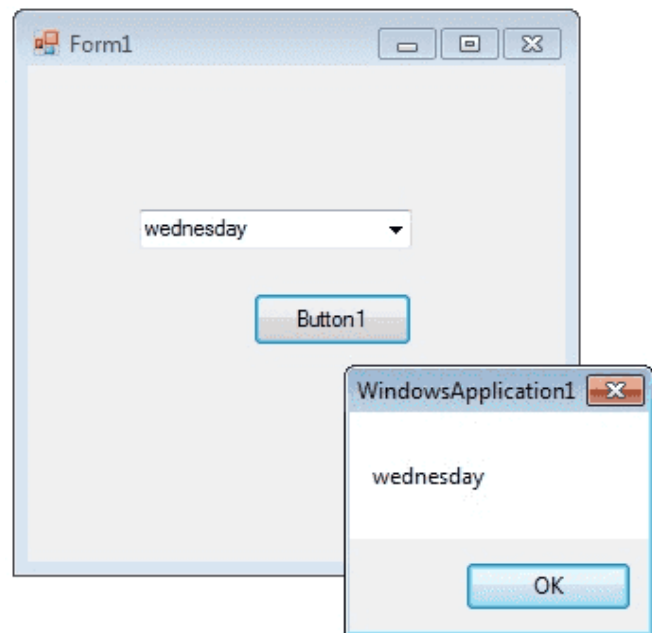


C# ComboBox Control

csharp.net-informations.com/gui/cs-combobox.htm

C# 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. A ComboBox displays a text box combined with a ListBox, which enables the user to select items from the list or enter a new value .

The user can type a value in the text field or click the button to display a drop down list. You can add individual objects with the Add method. You can delete items with the Remove method or clear the entire list with the Clear method.



How add a item to combobox

```
<span class="text-node">
comboBox1.Items.Add("Sunday");
comboBox1.Items.Add("Monday");
comboBox1.Items.Add("Tuesday");
</span>
```

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

```
<span class="text-node">
string var;
var = comboBox1.Text;
    Or
var item = this.comboBox1.GetItemText(this.comboBox1.SelectedItem);
MessageBox.Show(item);
</span>
```

How to remove an item from ComboBox

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);
```

The above code will remove the second item from the combobox.

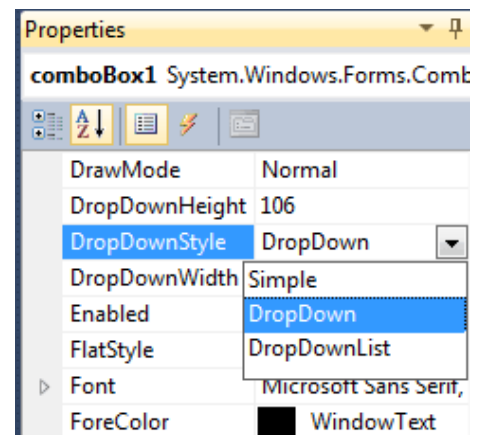
```
comboBox1.Items.Remove("Friday");
```

The above code will remove the item "Friday" from the combobox.

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 Selected Value

How to set the selected item in a comboBox

You can display selected item in a combobox in two ways.

```
<span class="text-node">
comboBox1.Items.Add("test1");
comboBox1.Items.Add("test2");
comboBox1.Items.Add("test3");
comboBox1.SelectedItem = "test3";
or
comboBox1.SelectedIndex = comboBox1.FindStringExact("test3");
</span>
```

ComboBox DataSource Property

How to populate a combo box with a DataSet ?

You can Programmatically Binding DataSource to ComboBox in a simple way..

Consider an sql string like...."select au_id,au_lname from authors";

Make a datasource and bind it like the following...

```
<span class="text-node">
```

```

comboBox1.DataSource = ds.Tables[0];
comboBox1.ValueMember = "au_id";
comboBox1.DisplayMember = "au_lname";
</span>

```

Combobox SelectedIndexChanged event

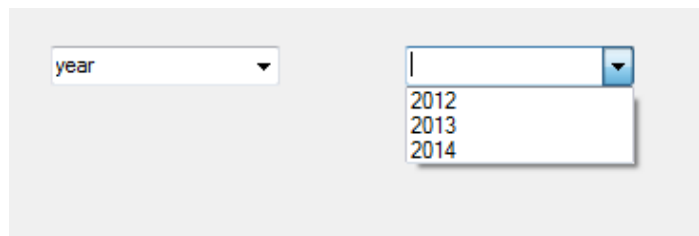
The SelectedIndexChanged event of a combobox fire when you change the slected item in a combobox. If you want to do something when you change the selection, you can write the program on SelectedIndexChanged event. From the following code you can understand how to set values in the SelectedIndexChanged event of a combobox. Drag and drop two combobox on the Form and copy and paste the following source code.

```

<span class="text-node">
using System;
using System.Windows.Forms;
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            comboBox1.Items.Add("weekdays");
            comboBox1.Items.Add("year");
        }
        private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
        {
            comboBox2.Items.Clear();
            if (comboBox1.SelectedItem == "weekdays")
            {
                comboBox2.Items.Add("Sunday");
                comboBox2.Items.Add("Monday");
                comboBox2.Items.Add("Tuesday");
            }
            else if (comboBox1.SelectedItem == "year")
            {
                comboBox2.Items.Add("2012");
                comboBox2.Items.Add("2013");
                comboBox2.Items.Add("2014");
            }
        }
    }
}
</span>

```

Output



ComboBox Databinding

You can bind data to a Combobox from various resources like Dataset, List, Enum, Dictionary etc. From the following link you can study more about ... [ComboBox Databinding](#)

ComboBox Default Value

How to set a default value for a Combo Box

You can set combobox default value by using SelectedIndex property

```
<span class="text-node">  
comboBox1.SelectedIndex = 6;  
</span>
```

Above code set 6th item as combobox default value

ComboBox readonly

How to make a combobox read only

You can make a ComboBox readonly, that means a user cannot write in a combo box but he can select the given items, 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.

Autocomplete ComboBox

From the version Visual Studio 2005, some of the controls support Autocomplete feature including the ComboBox controls. From the following link you can see how to make [Autocomplete ComboBox](#)

ComboBox Example

The following C# source code add seven days in a week to a combo box while load event of a Windows Form and int Button click event it displays the selected text in the Combo Box.

```

using System;
using System.Drawing;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            private void Form1_Load(object sender, EventArgs e)
            {
                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.SelectedIndex = comboBox1.FindStringExact("Sunday");

            }

            private void button1_Click(object sender, EventArgs e)
            {
                string var;
                var = comboBox1.Text;
                MessageBox.Show(var);
            }
        }
    }
}

```