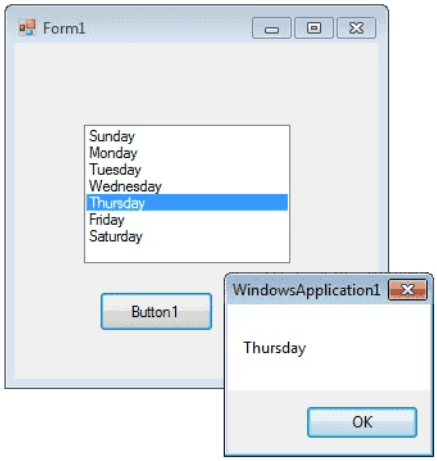
C# ListBox Control

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

The ListBox control enables you to display a list of items to the user that the user can select by clicking.



In addition to display and selection functionality, the ListBox also provides features that enable you to efficiently add items to the ListBox and to find text within the items of the list. 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.

```
listBox1.Items.Add("Sunday");
```

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

```
string var;
var = listBox1.Text;
```

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 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 C# 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.

```
<span class="text-node">
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)
        {
            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;
        }
        private void button1_Click(object sender, EventArgs e)
        {
            foreach (Object obj in listBox1.SelectedItems )
            {
                MessageBox.Show(obj.ToString ());
            }
        }
    }
}
</span>
```

How to bind a ListBox to a List ?

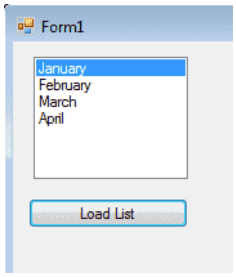
First you should create a fresh List Object and add items to the List.

```
<span class="text-node">
List<string> nList = new List<string>();
nList.Add("January");
nList.Add("February");
nList.Add("March");
nList.Add("April");
</string></string></span>
```

The next step is to bind this List to the Listbox. In order to do that you should set datasource of the Listbox.

listBox1.DataSource = nList;

Full Source code




```
<span class="text-node">
private void button1_Click(object sender, EventArgs e)
{
    List<string> nList = new List<string>();
    nList.Add("January");
    nList.Add("February");
    nList.Add("March");
    nList.Add("April");
    listBox1.DataSource = nList;
}
</string></string></span>
```

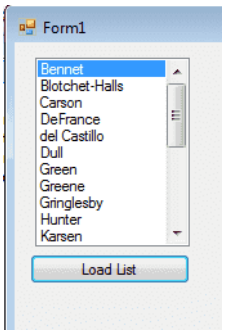
How to bind a listbox to database values ?

First you should create a connection string and fetch data from database to a Dataset.

```
<span class="text-node">
connetionString = "Data Source=ServerName;Initial Catalog=databasename;User ID=userid;Password=yourpassword";
sql = "select au_id,au_lname from authors";
</span>
```

After that you should set Listbox datasoure as Dataset.

```
<span class="text-node">
listBox1.DataSource = ds.Tables[0];
listBox1.ValueMember = "au_id";
listBox1.DisplayMember = "au_lname";
</span>
```




```
<span class="text-node">
using System;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            string connetionString = null;
            SqlConnection connection;
            SqlCommand command;
            SqlDataAdapter adapter = new SqlDataAdapter();
            DataSet ds = new DataSet();
            int i = 0;
            string sql = null;
            //connetionString = "Data Source=ServerName;Initial Catalog=databasename;User ID=userid;Password=yourpasswo
            //sql = "select au_id,au_lname from authors";
            connection = new SqlConnection(connetionString);
            try
            {
                connection.Open();
                command = new SqlCommand(sql, connection);
                adapter.SelectCommand = command;
                adapter.Fill(ds);
                adapter.Dispose();
                command.Dispose();
                connection.Close();
                listBox1.DataSource = ds.Tables[0];
                listBox1.ValueMember = "au_id";
                listBox1.DisplayMember = "au_lname";
            }
            catch (Exception ex)
            {
                MessageBox.Show("Cannot open connection ! ");
            }
        }
    }
}
</span>
```

How to refresh DataSource of a ListBox ?

How to clear the Listbox if its already binded with datasource ?

When you want to clear the Listbox, if the ListBox already binded with Datasource, you have to set the Datasource of Listbox as null.

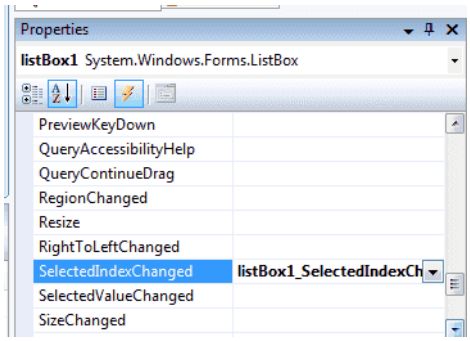
listBox1.DataSource = null;

How to SelectedIndexChanged event in ListBox ?

This event is fired when the item selection is changed in a ListBox. You can use this event in a situation that you want select an item from your listbox and accodring to this selection you can perform other programming needs.

You can add the event handler using the Properties Window and selecting the Event icon and double-clicking on SelectedIndexChanged as you can see in following image.

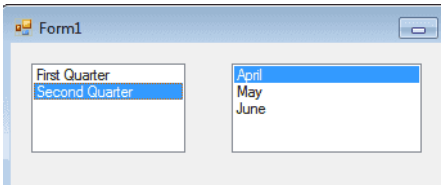
The event will fire again when you select a new item. You can write your code within SelectedIndexChanged event . When you double click on ListBox the code will automatically come in you code editor like the following image.




```
private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    //your code here
}
```

From the following example you can understand how to fire the SelectedIndexChanged event

First you should drag two listboxes on your Form. First listbox you should set the List as Datasource, the List contents follows:




```
<span class="text-node">
List<string> nList = new List<string>();
nList.Add("First Quarter");
nList.Add("Second Quarter");
</string></string></span>
```

When you load this form you can see the listbox is populated with List and displayed first quarter and second quarter. When you click the "Fist Quarter" the next listbox is populated with first quarter months and when you click "Second Quarter" you can see the second listbox is changed to second quarter months. From the following program you can understand how this happened.

Next : C# Checked ListBox Control

Download Source Code

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;
using System.Collections.Generic;


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

        List  < string > fQ = new List  < string > ();
        List  < string > sQ = new List  < string > ();

        private void Form1_Load(object sender, EventArgs e)
        {
            fQ.Add("January");
            fQ.Add("February");
            fQ.Add("March");

            sQ.Add("April");
            sQ.Add("May");
            sQ.Add("June");

            List < string > nList = new List  < string > ();

            nList.Add("First Quarter");
            nList.Add("Second Quarter");

            listBox1.DataSource = nList;
        }

        private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
        {
            if (listBox1.SelectedIndex == 0)
            {
                listBox2.DataSource = null;
                listBox2.DataSource = fQ;
            }
            else if (listBox1.SelectedIndex == 1)
            {
                listBox2.DataSource = null;
                listBox2.DataSource = sQ;
            }
        }


    }
}
```