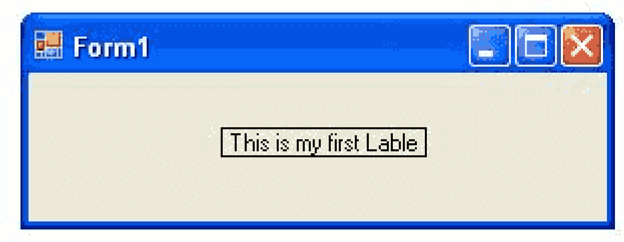
C# Label Control

Labels are one of the most frequently used C# control. We can use the Label control to display text in a set location on the page. Label controls can also be used to add descriptive text to a Form to provide the user with helpful information. The Label class is defined in the System.Windows.Forms namespace.



Add a Label control to the form - Click Label in the Toolbox and drag it over the forms Designer and drop it in the desired location.

If you want to change the display text of the Label, you have to set a new text to the Text property of Label.

```
label1.Text = "This is my first Label";
```

In addition to displaying text, the Label control can also display an image using the Image property, or a combination of the ImageIndex and ImageList properties.

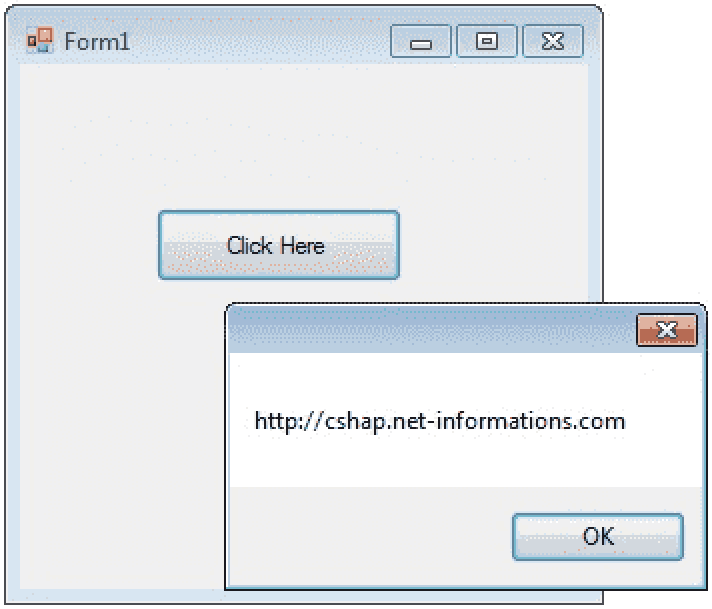
```
label1.Image = Image.FromFile("C:\\testimage.jpg");
```

The following C# source code shows how to set some properties of the Label through coding.

```
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)
        {
            label1.Text = "This is my first Lable";
            label1.BorderStyle = BorderStyle.FixedSingle;
            label1.TextAlign = ContentAlignment.MiddleCenter;
        }
    }
}
```

C# Button Control

Windows Forms controls are reusable components that encapsulate user interface functionality and are used in client side Windows applications. A button is a control, which is an interactive component that enables users to communicate with an application. The Button class inherits directly from the ButtonBase class. A Button can be clicked by using the mouse, ENTER key, or SPACEBAR if the button has focus.



http://cshap.net-informations.com

When you want to change display text of the Button , you can change the Text property of the button.

```
button1.Text = "Click Here";
```

Similarly if you want to load an Image to a Button control , you can code like this

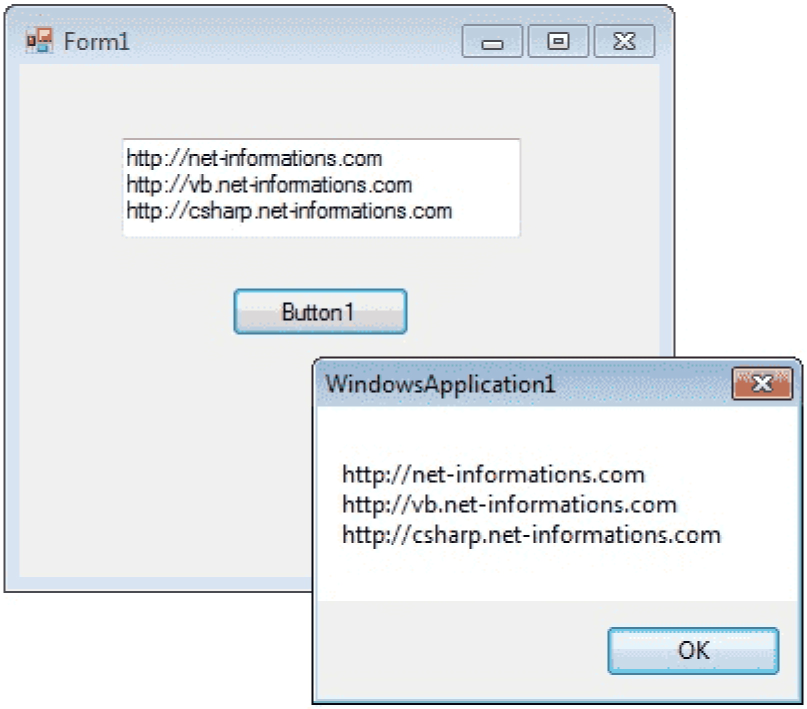
```
button1.Image = Image.FromFile("C:\\testimage.jpg");
```

The following C# source code shows how to change the button Text property while Form loading event and to display a message box when pressing a Button Control.

```
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)
        {
            button1.Text = "Click Here";
        }

        private void button1_Click(object sender, EventArgs e)
        {
            MessageBox.Show("http://cshap.net-informations.com");
        }
    }
}
```

C# TextBox Control

A TextBox control is used to display, or accept as input, a single line of text. This control has additional functionality that is not found in the standard Windows text box control, including multiline editing and password character masking.



http://net-informations.com
http://vb.net-informations.com
http://csharp.net-informations.com

http://net-informations.com
http://vb.net-informations.com
http://csharp.net-informations.com

A text box object is used to display text on a form or to get user input while a C# program is running. In a text box, a user can type data or paste it into the control from the clipboard.

For displaying a text in a TextBox control , you can code like this

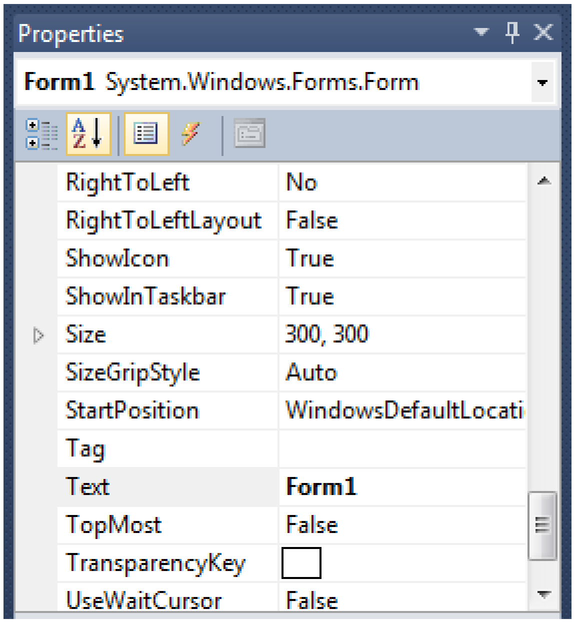
```
textBox1.Text = "http://csharp.net-informations.com";
```

You can also collect the input value from a TextBox control to a variable like this way

```
string var;
var = textBox1.Text;
```

C# TextBox Properties

You can set TextBox properties through Property window or through program. You can open Properties window by pressing F4 or right click on a control and select Properties menu item



Form1 System.Windows.Forms.Form

RightToLeft	No
RightToLeftLayout	False
ShowIcon	True
ShowInTaskbar	True
Size	300, 300
SizeGripStyle	Auto
StartPosition	WindowsDefaultLocati
Tag	
Text	**Form1**
TopMost	False
TransparencyKey	□
UseWaitCursor	False

The below code set a textbox width as 250 and height as 50 through source code.

```
textBox1.Width = 250;
textBox1.Height = 50;
```

Background Color and Foreground Color

You can set background color and foreground color through property window and programmatically.

```
textBox1.BackColor = Color.Blue;
textBox1.ForeColor = Color.White;
```

Textbox BorderStyle

You can set 3 different types of border style for textbox, they are None, FixedSingle and fixed3d.

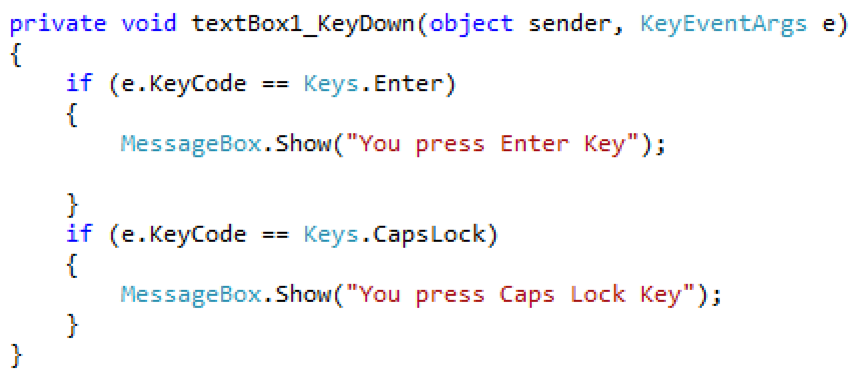
```
textBox1.BorderStyle = BorderStyle.Fixed3D;
```

TextBox Events

Keydown event

You can capture which key is pressed by the user using KeyDown event

e.g.

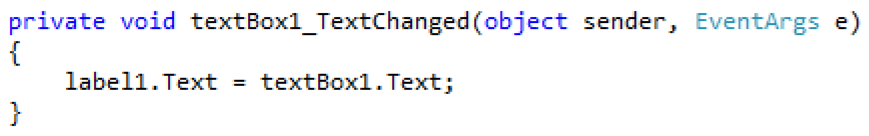


```
private void textBox1_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        MessageBox.Show("You press Enter Key");

    }
    if (e.KeyCode == Keys.CapsLock)
    {
        MessageBox.Show("You press Caps Lock Key");
    }
}
```

TextChanged Event

When user input or setting the Text property to a new value raises the TextChanged event

e.g.



```
private void textBox1_TextChanged(object sender, EventArgs e)
{
    label1.Text = textBox1.Text;
}
```

Textbox Maximum Length

Sets the maximum number of characters or words the user can input into the text box control.

```
textBox1.MaxLength = 40;
```

Textbox ReadOnly

When a program wants to prevent a user from changing the text that appears in a text box, the program can set the controls Read-only property is to True.

```
textBox1.ReadOnly = true;
```

Multiline TextBox

You can use the Multiline and ScrollBars properties to enable multiple lines of text to be displayed or entered.

```
textBox1.Multiline = true;
```

Textbox passowrd character

TextBox controls can also be used to accept passwords and other sensitive information. You can use the PasswordChar property to mask characters entered in a single line version of the control

```
textBox1.PasswordChar = '*';
```

The above code set the PasswordChar to * , so when the user enter password then it display only * instead of typed characters.

How to Newline in a TextBox

You can add new line in a textbox using many ways.

```
textBox1.Text += "your text" + "\r\n";
```

or

```
textBox1.Text += "your text" + Environment.NewLine;
```

How to retrieve integer values from textbox ?

```
int i;
i = int.Parse (textBox1.Text);
```

Parse method Converts the string representation of a number to its integer equivalent.

String to Float conversion

```
float  i;
i = float.Parse (textBox1.Text);
```

String to Double conversion

```
double   i;
i = float.Parse (textBox1.Text);
```

How to allow only numbers in a textbox

Many of us have faced a situation where we want the user to enter a number in a TextBox. Click the following link that are going to make a Numeric Textbox which will accept only numeric values; if there are any values except numeric. More about.... How do I make a textbox that only accepts numbers

From the following C# source code you can see some important property settings to a TextBox control.

```
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)
        {
            textBox1.Width = 250;
            textBox1.Height = 50;
            textBox1.Multiline = true;
            textBox1.BackColor = Color.Blue;
            textBox1.ForeColor = Color.White;
            textBox1.BorderStyle = BorderStyle.Fixed3D;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            string var;
            var = textBox1.Text;
            MessageBox.Show(var);
        }
    }
}
```