

 무료 전자 책

배우기

winforms

Free unaffiliated eBook created from
Stack Overflow contributors.

#winforms

.....	1
1: winforms	2
.....	2
.....	2
Examples.....	2
Visual Studio WinForms	2
Windows Forms	2
.....	2
.....	3
.....	3
C # WinForms	4
VB.NET WinForms	5
2:	7
Examples.....	7
.....	7
.....	7
.....	8
CheckBox.....	9
.....	10
NumericUpDown.....	13
.....	15
3:	17
.....	17
.....	17
Examples.....	17
.....	17
4:	18
.....	18
HelpProvider	18
.....	18
HelpRequested	18
.....	18

MessgeBox CommonDialogs18
 18
 Examples.....18
 18
 MessageBox 19
 CHM ()19
 CHM19
 CHM19
 URL19
 F119
 CommonDialogs 20
 HelpRequested21
 21
 22
 CHM 22
 22
 () 22
 22
 URL 22
 22
 23
 Form HelpButtonClicked 23
5: **25**
 25
 Examples.....25
 25
 NumberBox..... 28
6: **36**
 36
 Examples.....36
 36

.....	36
.....	36
7:	38
Examples.....	38
.....	38
.....	38
.....	38
.....	38
.....	40

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [winforms](#)

It is an unofficial and free winforms ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official winforms.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: winforms

Windows Forms ("WinForms") .NET Framework GUI . [Win32 API](#) .NET Framework Windows .

WinForms . (, , ,) () .

Windows , WinForms . Control , , . Control . (Form , UserControl) (TableLayoutPanel , FlowLayoutPanel) .

WinForms .NET Framework (v1.0) (v4.5) . . [Build 2014 9 Microsoft](#) .

Windows Forms . .

, [Mono](#) Windows Forms .NET 2.0 Microsoft . WinForms Mono Windows API () .

:

- [MSDN Windows Forms](#)

Examples

Visual Studio WinForms

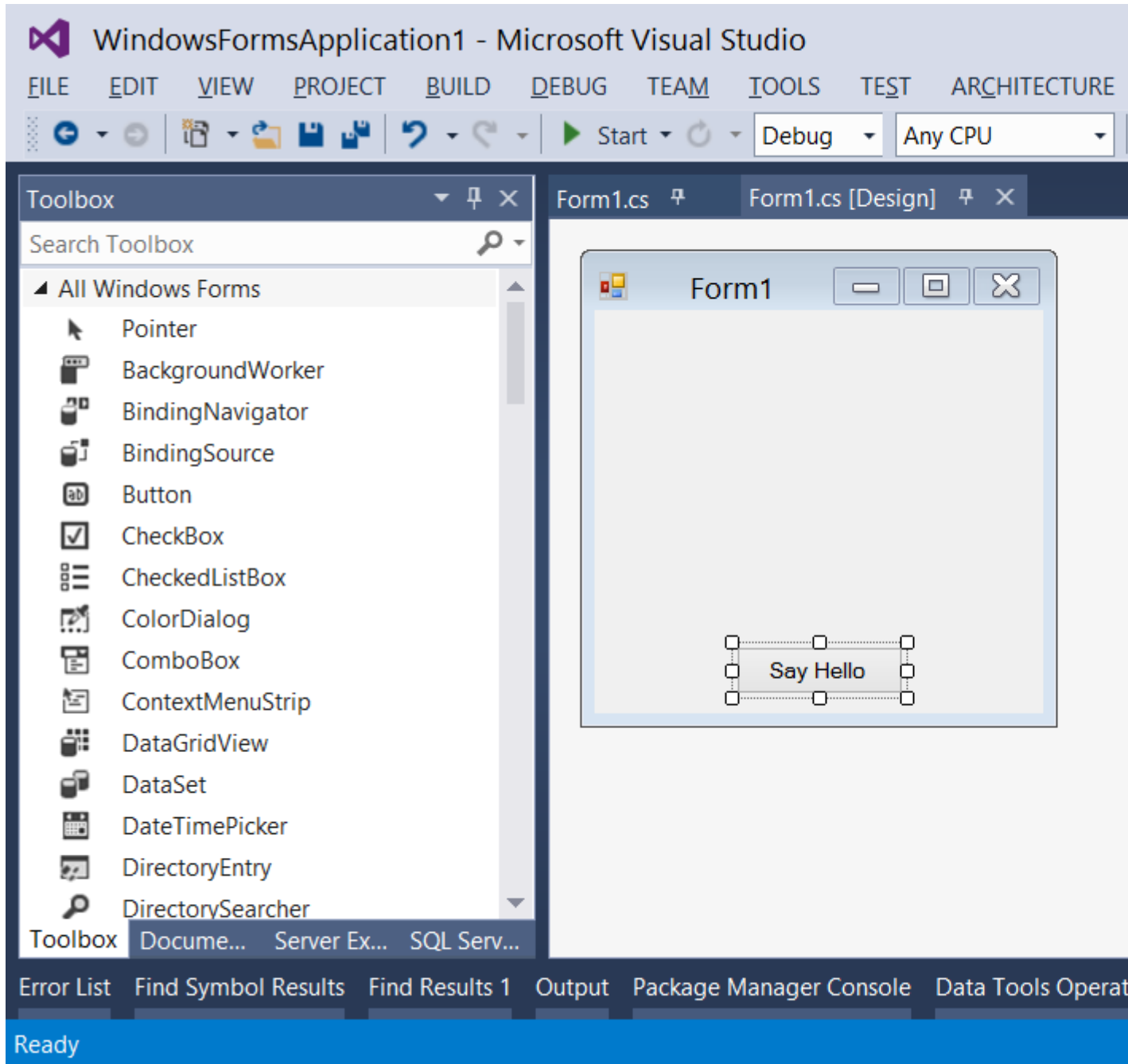
Visual Studio Windows Forms .

Windows Forms

1. Visual Studio .
2. . .
3. "Visual C #" "Visual Basic" .
4. .
5. **Windows Forms** .
6. .
7. .
8. .
9. Windows Forms Designer **Form1** .

1. **Button** .

2. . Text Say Hello .



1. Click

2. .

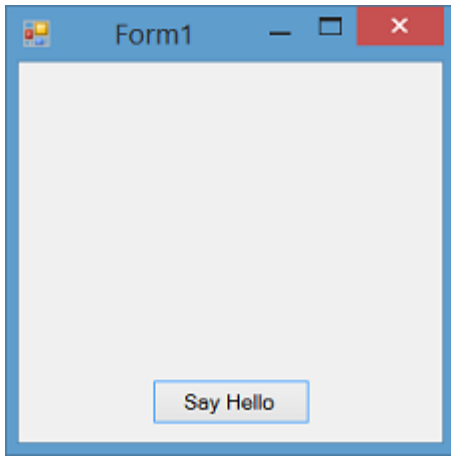
#

```
MessageBox.Show("Hello, World!");
```

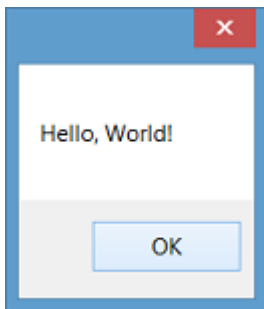
VB.NET

```
MessageBox.Show("Hello, World!");
```

1. F5



2. "Hello, World!" .



3. Visual Studio .

C # WinForms

1. (:) .

```
using System;
using System.ComponentModel;
using System.Drawing;
using System.Windows.Forms;

namespace SampleApp
{
    public class MainForm : Form
    {
        private Button btnHello;

        // The form's constructor: this initializes the form and its controls.
        public MainForm()
        {
            // Set the form's caption, which will appear in the title bar.
            this.Text = "MainForm";

            // Create a button control and set its properties.
            btnHello = new Button();
            btnHello.Location = new Point(89, 12);
            btnHello.Name = "btnHello";
            btnHello.Size = new Size(105, 30);
            btnHello.Text = "Say Hello";

            // Wire up an event handler to the button's "Click" event
```



```

        // (see the code in the btnHello_Click function below).
        btnHello.Click += new EventHandler(btnHello_Click);

        // Add the button to the form's control collection,
        // so that it will appear on the form.
        this.Controls.Add(btnHello);
    }

    // When the button is clicked, display a message.
    private void btnHello_Click(object sender, EventArgs e)
    {
        MessageBox.Show("Hello, World!");
    }

    // This is the main entry point for the application.
    // All C# applications have one and only one of these methods.
    [STAThread]
    static void Main()
    {
        Application.EnableVisualStyles();
        Application.Run(new MainForm());
    }
}

```

2. / . (X:\MainForm.cs .

3. C# .

```
%WINDIR%\Microsoft.NET\Framework64\v4.0.30319\csc.exe /target:winexe "X:\MainForm.cs"
```

```
: .NET Framework C# %WINDIR%\Microsoft.NET .C# C# .
```

4. MainForm.exe . Explorer .

VB.NET WinForms

1. (:) .

```

Imports System.ComponentModel
Imports System.Drawing
Imports System.Windows.Forms

Namespace SampleApp
    Public Class MainForm : Inherits Form
        Private btnHello As Button

        ' The form's constructor: this initializes the form and its controls.
        Public Sub New()
            ' Set the form's caption, which will appear in the title bar.
            Me.Text = "MainForm"

            ' Create a button control and set its properties.
            btnHello = New Button()
            btnHello.Location = New Point(89, 12)
            btnHello.Name = "btnHello"
        End Sub
    End Class
End Namespace

```

```

    btnHello.Size = New Size(105, 30)
    btnHello.Text = "Say Hello"

    ' Wire up an event handler to the button's "Click" event
    ' (see the code in the btnHello_Click function below).
    AddHandler btnHello.Click, New EventHandler(AddressOf btnHello_Click)

    ' Add the button to the form's control collection,
    ' so that it will appear on the form.
    Me.Controls.Add(btnHello)
End Sub

' When the button is clicked, display a message.
Private Sub btnHello_Click(sender As Object, e As EventArgs)
    MessageBox.Show("Hello, World!")
End Sub

' This is the main entry point for the application.
' All VB.NET applications have one and only one of these methods.
<STAThread> _
Public Shared Sub Main()
    Application.EnableVisualStyles()
    Application.Run(New MainForm())
End Sub
End Class
End Namespace

```

2. / . (X:\MainForm.vb .

3. VB.NET .

```
%WINDIR%\Microsoft.NET\Framework64\v4.0.30319\vbc.exe /target:winexe "X:\MainForm.vb"
```

```
: .NET Framework VB.NET %WINDIR%\Microsoft.NET . VB.NET Hello World .
```

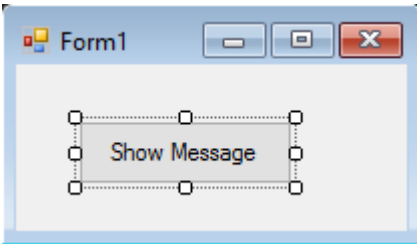
4. MainForm.exe . Explorer .

winforms : <https://riptutorial.com/ko/winforms/topic/1018/winforms->

2:

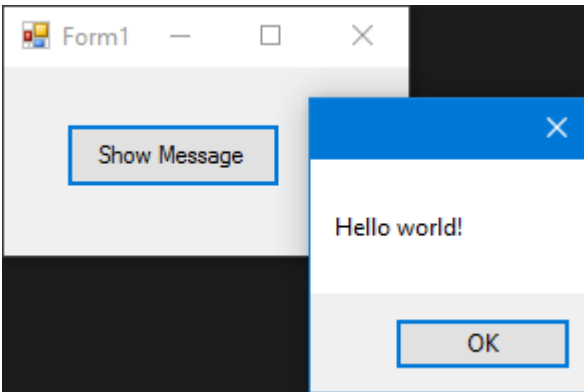
Examples

Message box . cmdShowMessage .

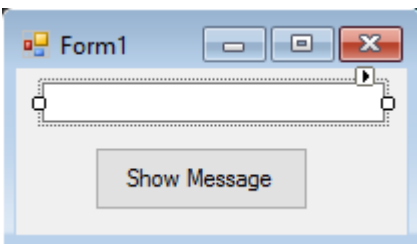


Visual Studio click . MessageBox .

```
private void cmdShowMessage_Click(object sender, EventArgs e)
{
    MessageBox.Show("Hello world!");
}
```



messagebox . .



```
private void cmdShowMessage_Click(object sender, EventArgs e)
```

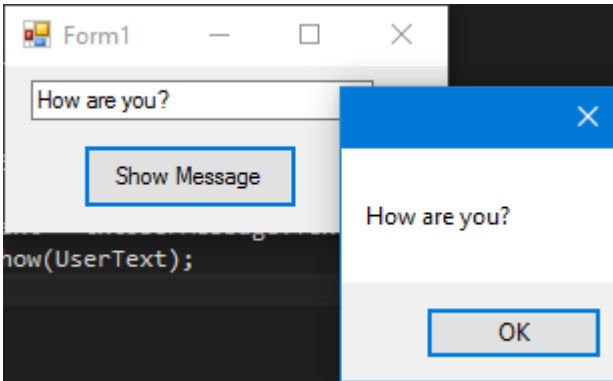
```

{
    string UserText = txtUserMessage.Text;
    MessageBox.Show(UserText);
}

```

.Text , .

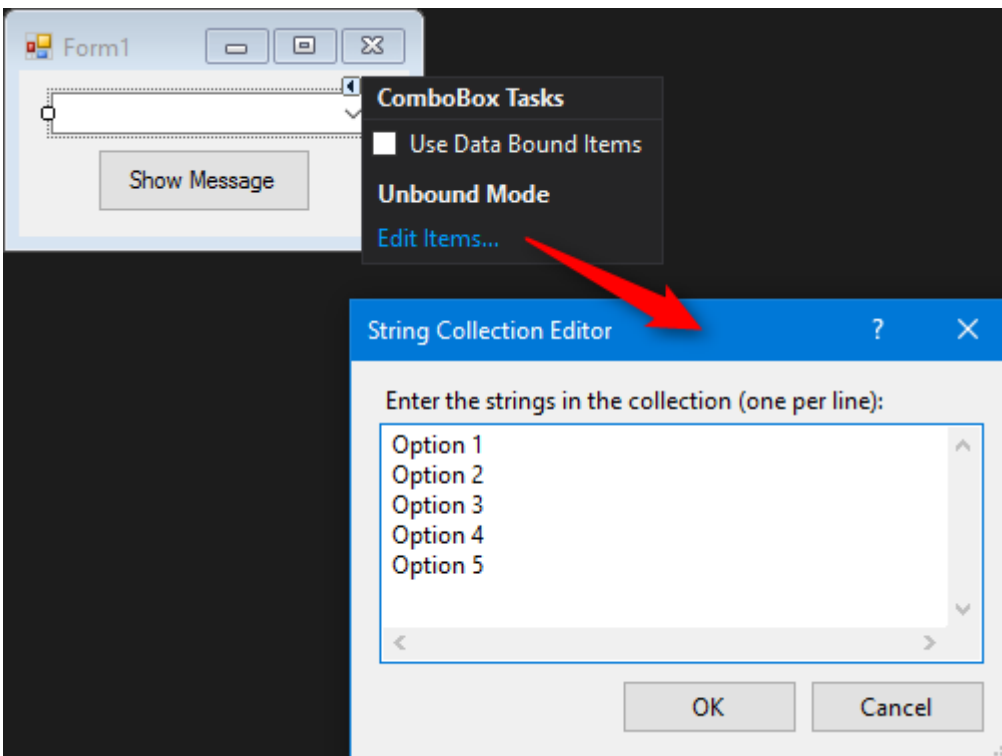
. MessageBox :



ComboBox .

messagebox .

. Items .



click .

```

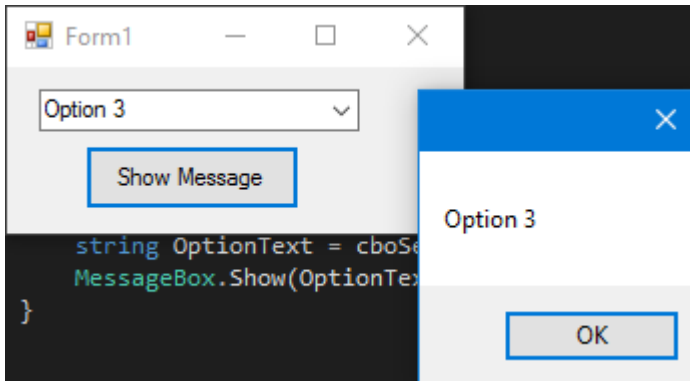
private void cmdShowMessage_Click(object sender, EventArgs e)
{
    string OptionText = cboSelectOption.SelectedItem.ToString();
    MessageBox.Show(OptionText);
}

```

```
}
```

```
SelectedItem . ToString() .
```

.



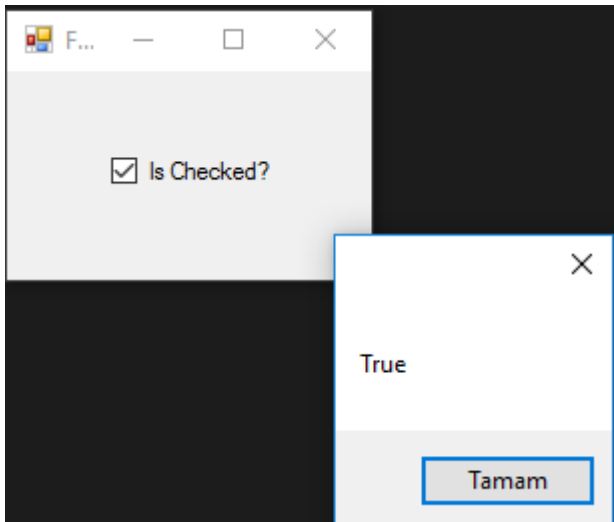
```
SelectionChangeCommitted . SelectedIndexChanged .
```

CheckBox

Checkbox "Are you ok?" specific boolean ..

```
CheckedChanged . CheckedChanged check .
```

"Check Is?" CheckBox ..



```
CheckedChanged MessageBox .
```

```
private void checkBox1_CheckedChanged(object sender, EventArgs e)  
{  
    bool IsChecked = checkBox1.Checked;  
    MessageBox.Show(IsChecked.ToString());  
}
```

CheckBox -> IsChecked true .

CheckBox -> IsChecked false .

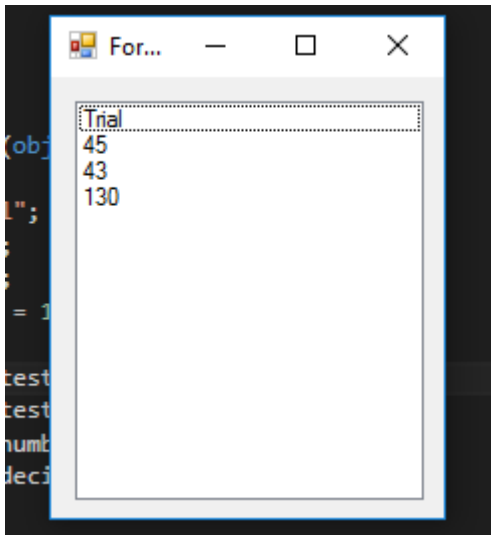
Listbox .Listbox Combobox Combobox . .Listbox; .

Listbox

```
private void Form3_Load(object sender, EventArgs e)
{
    string test = "Trial";
    string test2 = "45";
    int numberTest = 43;
    decimal decimalTest = 130;

    listBox1.Items.Add(test);
    listBox1.Items.Add(test2);
    listBox1.Items.Add(numberTest);
    listBox1.Items.Add(decimalTest);
}
```

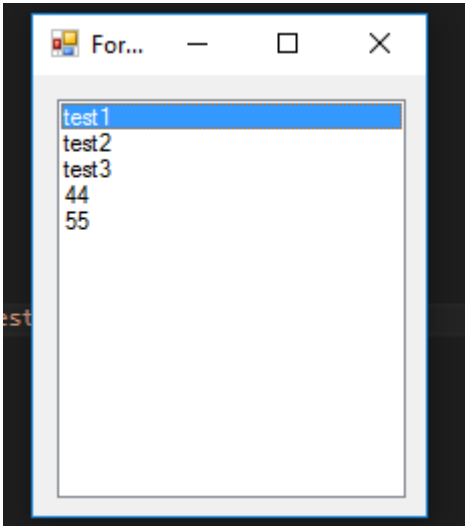
;



datasources .

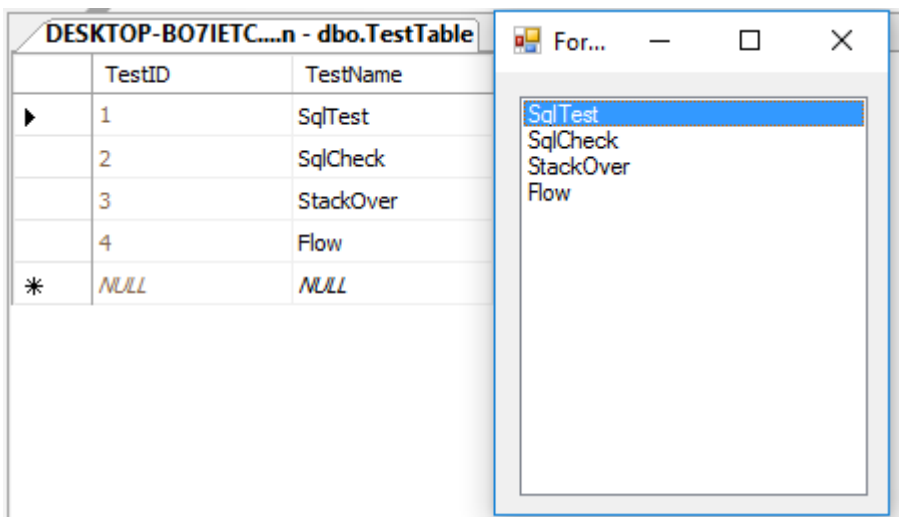
```
private void Form3_Load(object sender, EventArgs e)
{
    List<string> TestList = new List<string> { "test1", "test2", "test3", "44", "55"
};
    listBox1.DataSource = TestList;
}
```

;



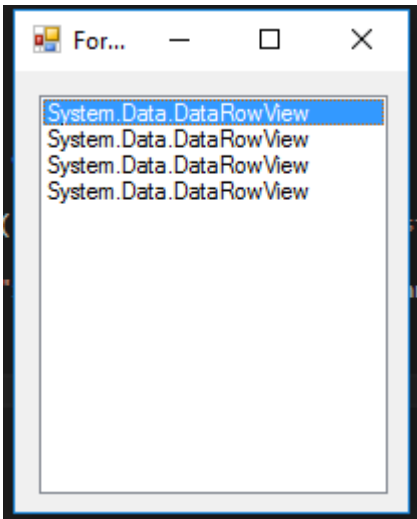
```
private void Form3_Load(object sender, EventArgs e)
{
    SqlConnection Connection = new
SqlConnection("Server=serverName;Database=db;Trusted_Connection=True;"); //Connetion to MS-
SQL(RDBMS)
    Connection.Open(); //Connection open
    SqlDataAdapter Adapter = new SqlDataAdapter("Select * From TestTable",
Connection); // Get all records from TestTable.
    DataTable DT = new DataTable();
    Adapter.Fill(DT); // Fill records to DataTable.
    listBox1.DataSource = DT; // DataTable is the datasource.
    listBox1.ValueMember = "TestID";
    listBox1.DisplayMember= "TestName";
}
}
```

;



SQL listBox ValueMember DisplayMember .

,

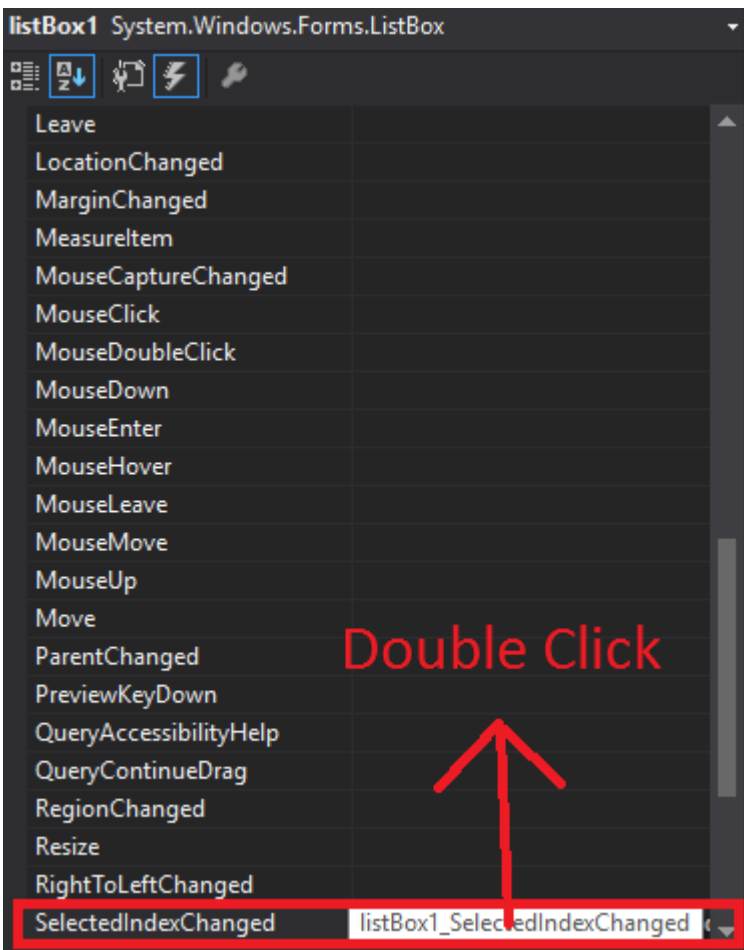


;

SelectedIndex_Changed;

```
private void Form3_Load(object sender, EventArgs e)
{
    List<string> DataList = new List<string> {"test1" , "test2" , "test3" , "44" ,
"45" };
    listBox1.DataSource = TestList;
}
```

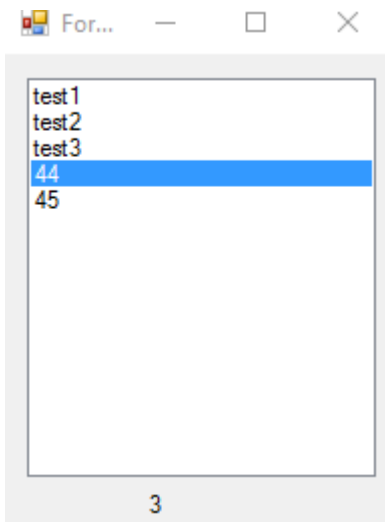
Listbox F4 .



Visual Studio codebehind listBox1_SelectedIndexChanged .

```
private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    int Index = listBox1.SelectedIndex;
    label1.Text = Index.ToString();
}
```

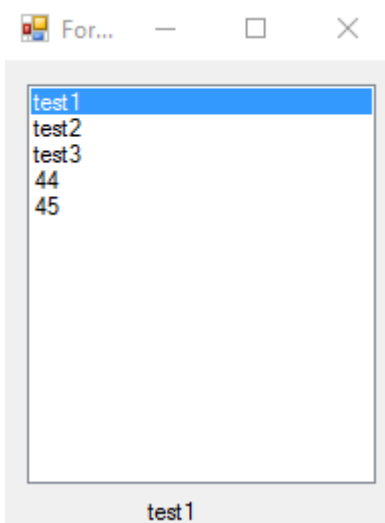
SelectedIndex_Changed; ()



SelectedValue_Changed; (SelectedIndex_Changed .)

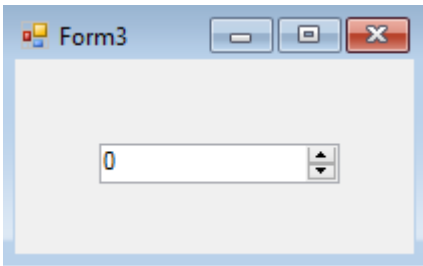
```
private void listBox1_SelectedValueChanged(object sender, EventArgs e)
{
    label1.Text = listBox1.SelectedValue.ToString();
}
```

;



NumericUpDown

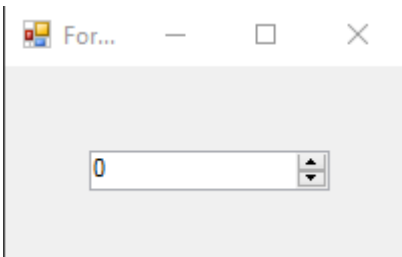
NumericUpDown TextBox



Form_Load .

```
private void Form3_Load(object sender, EventArgs e)
{
    numericUpDown1.Maximum = 10;
    numericUpDown1.Minimum = -10;
}
```

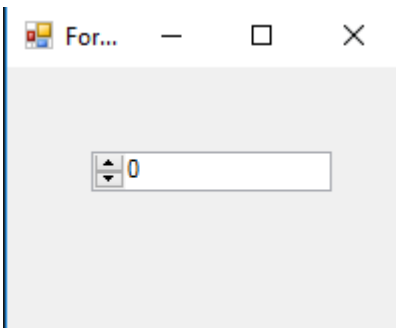
;



UpDownAlign .

```
private void Form3_Load(object sender, EventArgs e)
{
    numericUpDown1.UpDownAlign = LeftRightAlignment.Left;
}
```

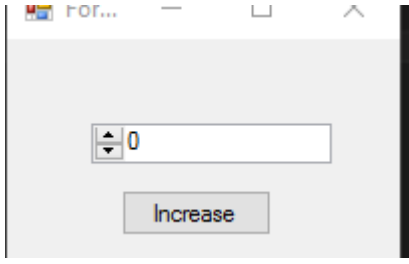
;



UpButton() UpButton() . (.button .)

```
private void button1_Click(object sender, EventArgs e)
{
    numericUpDown1.UpButton();
}
```

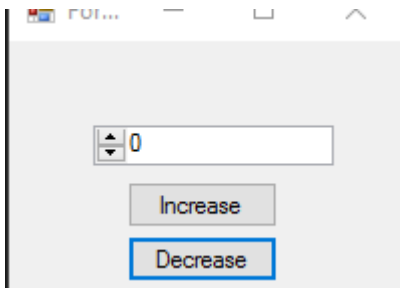
**



```
DownButton() .( . button .)
```

```
private void button2_Click(object sender, EventArgs e)
{
    numericUpDown1.DownButton();
}
```

;

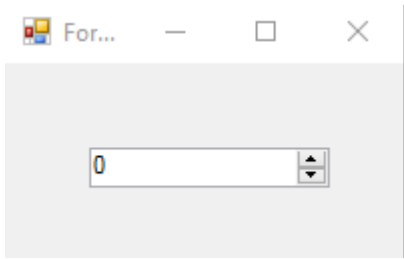


ValueChanged;

.

```
private void numericUpDown1_ValueChanged(object sender, EventArgs e)
{
    decimal result = numericUpDown1.Value; // it will get the current value
    if (result == numericUpDown1.Maximum) // if value equals Maximum value that we set
in Form_Load.
    {
        label1.Text = result.ToString() + " MAX!"; // it will add "MAX" at the end of
the label
    }
    else if (result == numericUpDown1.Minimum) // if value equals Minimum value that
we set in Form_Load.
    {
        label1.Text = result.ToString() + " MIN!"; // it will add "MIN" at the end of
the label
    }
    else
    {
        label1.Text = result.ToString(); // If Not Max or Min, it will show only the
number.
    }
}
```

;



: <https://riptutorial.com/ko/winforms/topic/5816/>-

3:

propertyName	.
dataSource	Object.
dataMember	.
formattingEnabled	.
updateMode	()
nullValue	DBNull .
formatInfo	IFormatProvider .

<https://msdn.microsoft.com/en-us/library/ef2xyb33.aspx> . !

Examples

System.Windows.Forms.Binding DataBindings . Add () - .

```
textBox.DataBindings.Add( "Text", dataObj, "MyProperty" );
```

, . dataObj.MyProperty changeevent textBox.Text . textBox.TextChanged
dataObj.MyProperty .

: <https://riptutorial.com/ko/winforms/topic/7362/>-

4:

Windows Forms . CHM URL . , .

HelpProvider

[HelpProvider](#) . F1 .

-
- CHM (, ,)
- URL

[Help](#) .

-
- CHM (, ,)
- URL

HelpRequested

[HelpRequested](#) Control Form F1 .

Form . ? HelpProvider .

MessageBox CommonDialogs

MessageBox , OpenFileDialog , SaveDialog ColorDialog .

[ToolTip](#) . ToolTip .

HelpProvider Help HTML (.chm) HTML . , , . .HTML HTML 1.x .HTML
"HTML " [Microsoft HTML](#) HTML .

Examples

Help Class HTML Help 1.0 . (.chm) HTML HTML . , , . .Microsoft HTML Help
Workshop HTML Help 1.x .

#

```
Help.ShowHelp(this, helpProviderMain.HelpNamespace);
```

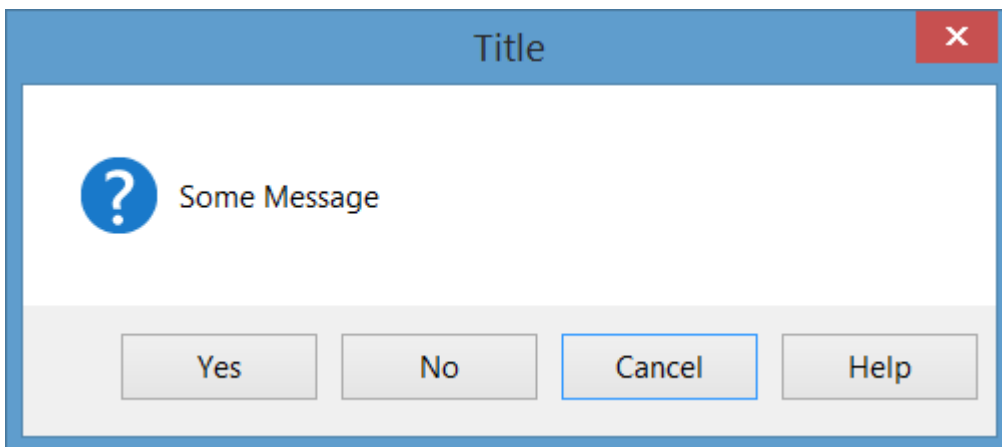
VB.NET

```
Help.ShowHelp(Me, hlpProviderMain.HelpNamespace)
```

MessageBox

```
.Help MessageBox . [] F1 CHM URL MessageBox . .
```

```
MessageBox .
```



CHM ()

```
MessageBox.Show("Some Message", "Title", MessageBoxButtons.YesNoCancel,  
    MessageBoxIcon.Question, MessageBoxDefaultButton.Button3, 0,  
    "help.chm", HelpNavigator.KeywordIndex, "SomeKeyword");
```

CHM

```
MessageBox.Show("Some Message", "Title", MessageBoxButtons.YesNoCancel,  
    MessageBoxIcon.Question, MessageBoxDefaultButton.Button3, 0,  
    "help.chm", HelpNavigator.Topic, "/SomePath/SomePage.html");
```

CHM

```
MessageBox.Show("Some Message", "Title", MessageBoxButtons.YesNoCancel,  
    MessageBoxIcon.Question, MessageBoxDefaultButton.Button3, 0,  
    "help.chm");
```

URL .

```
MessageBox.Show("Some Message", "Title", MessageBoxButtons.YesNoCancel,  
    MessageBoxIcon.Question, MessageBoxDefaultButton.Button3, 0,  
    "http://example.com");
```

F1

MessageBox parent `HelpRequested` .

```
private void Form1_HelpRequested(object sender, HelpEventArgs hlpevent)
{
    // Perform custom action, for example show a custom help form
    var f = new Form();
    f.ShowDialog();
}
```

MessageBox with Help .

```
MessageBox.Show("Some Message", "Title", MessageBoxButtons.YesNoCancel,
    MessageBoxIcon.Question, MessageBoxDefaultButton.Button3, 0, true);
```

:

```
MessageBox.Show("Some Message", "Title", MessageBoxButtons.YesNoCancel,
    MessageBoxIcon.Question, MessageBoxDefaultButton.Button3, 0, false);
```

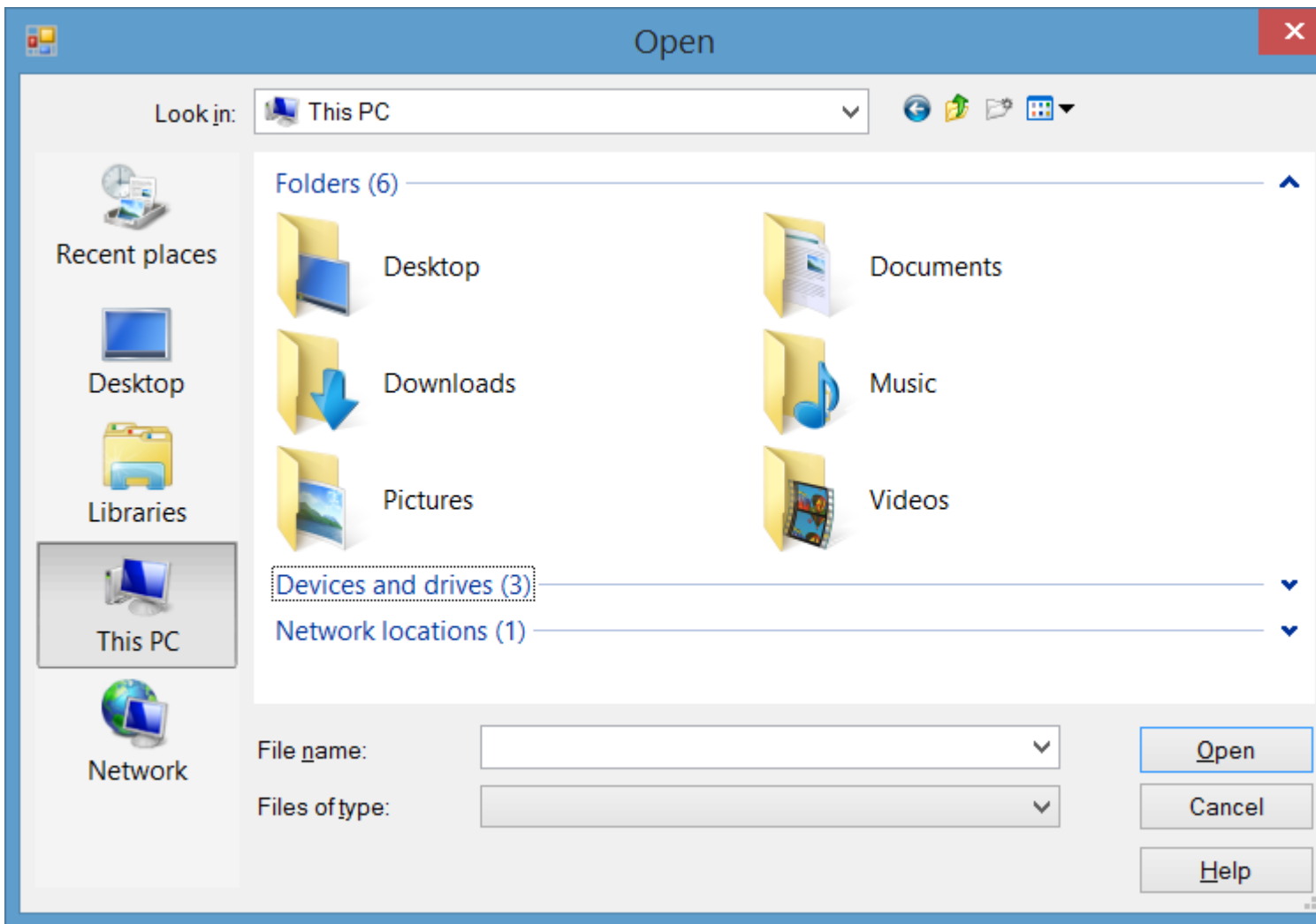
CommonDialogs

OpenFileDialog, SaveFileDialog ColorDialog . ShowHelp true `HelpRequest` .

```
void openFileDialog1_HelpRequest(object sender, EventArgs e)
{
    //Perform custom action
    Help.ShowHelp(this, "Http://example.com");
}
```

- ShowHelp true true .
- Help **F1** .

Help OpenFileDialog :



HelpRequested

F1 (?) [HelpRequested](#) .

.

[HelpRequested](#) . [Handled](#) true .

HelpRequested F1 [textBox1](#) .

```
private void Form1_HelpRequested(object sender, HelpEventArgs hlpevent)
{
    var c = this.ActiveControl;
    if(c!=null)
        MessageBox.Show(c.Name);
}
private void textBox1_HelpRequested(object sender, HelpEventArgs hlpevent)
{
    hlpevent.Handled = true;
    MessageBox.Show("Help request handled and will not bubble up");
}
```

URL [Help](#) CHM .

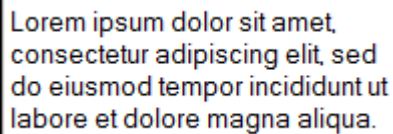
[Help](#)

-
- CHM (, ,)
- URL

`Help.ShowPopup` .

```
private void control_MouseClick(object sender, MouseEventArgs e)
{
    var c = (Control)sender;
    var help = "Lorem ipsum dolor sit amet, consectetur adipiscing elit, " +
               "sed do eiusmod tempor incididunt ut labore et dolore magna aliqua."
    if (c != null)
        Help.ShowPopup(c, "Lorem ipsum dolor sit amet.", c.PointToScreen(e.Location));
}
```

:



CHM

`Help.ShowHelp` CHM , , .

```
Help.ShowHelp(this, "Help.chm");
```

()

```
Help.ShowHelp(this, "Help.chm", HelpNavigator.Index, "SomeKeyword");
```

```
Help.ShowHelp(this, "Help.chm", HelpNavigator.Topic, "/SomePath/SomePage.html");
```

URL

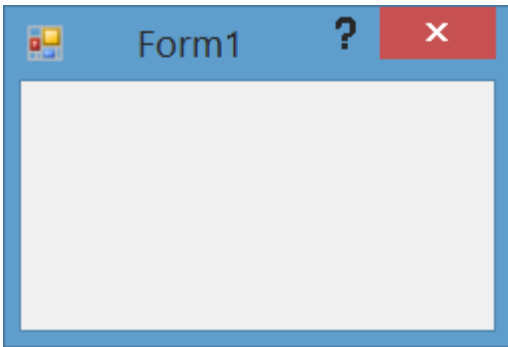
`ShowHelp` URL .

```
Help.ShowHelp(this, "Http://example.com");
```

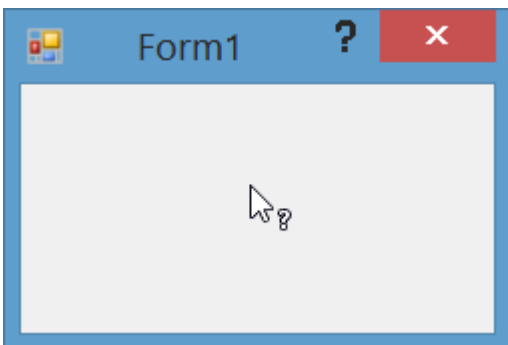
Form . .

1. form `HelpButton` true true .
2. `MinimizeBox` `MaximizeBox` false .

Form



Help ? :



```
Control Form HelpRequested HelpProvider HelpProvider .
```

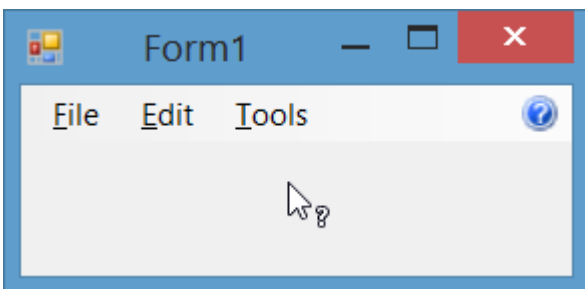
```
Form MinimizeBox MaximizeBox true , Form .
```

```
MenuStrip . MenuStrip ToolStripMenuItem Click .
```

```
private const int WM_SYSCOMMAND = 0x0112;
private const int SC_CONTEXTHELP = 0xF180;
[System.Runtime.InteropServices.DllImport("user32.dll")]
static extern IntPtr SendMessage(IntPtr hWnd, int Msg, int wParam, int lParam);
private void helpToolStripMenuItem_Click(object sender, EventArgs e)
{
    SendMessage(this.Handle, WM_SYSCOMMAND, SC_CONTEXTHELP, 0);
}
```

```
: Button button1.Capture = false; ToolStripMenuItem .
```

? .



Form HelpButtonClicked

HelpButton HelpButtonClicked . args Cancel true .

```
private void Form1_HelpButtonClicked(object sender, CancelEventArgs e)
{
    e.Cancel = true;
    //Perform some custom action
    MessageBox.Show("Some Custom Help");
}
```

: <https://riptutorial.com/ko/winforms/topic/3285/>-

5:

. : . .

Examples

WinForms WPF . " " .

WinForms .

TextBox . TextBox Windows . TextBox . .

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace StackOverflowDocumentation
{
    public class SOTextBox : TextBox
    {
        public SOTextBox() : base()
        {
            base.BackColor = SOUserPreferences.BackColor;
            base.ForeColor = SOUserPreferences.ForeColor;
        }
        protected override void OnEnter(EventArgs e)
        {
            base.BackColor = SOUserPreferences.FocusColor;
            base.OnEnter(e);
        }
        protected override void OnLeave(EventArgs e)
        {
            base.BackColor = SOUserPreferences.BackColor;
            base.OnLeave(e);
        }
    }
}
```

. **TextBox** .

. :

```
using System;
using System.Threading;
using Microsoft.Win32;
using System.Globalization;
using System.Data;
using System.Drawing;

namespace StackOverflowDocumentation
{
    public class SOUserPreferences
```

```

{
    private static string language;
    private static string logPath;
    private static int formBackCol;
    private static int formForeCol;
    private static int backCol;
    private static int foreCol;
    private static int focusCol;

    static SOUserPreferences()
    {
        try
        {
            RegistryKey HKCU = Registry.CurrentUser;
            RegistryKey kSOPrefs = HKCU.OpenSubKey("SOPrefs");
            if (kSOPrefs != null)
            {
                language = kSOPrefs.GetValue("Language", "EN").ToString();
                logPath = kSOPrefs.GetValue("LogPath", "c:\\windows\\logs\\").ToString();
                formForeCol = int.Parse(kSOPrefs.GetValue("FormForeColor", "-2147483630").ToString());
                formBackCol = int.Parse(kSOPrefs.GetValue("FormBackColor", "-2147483633").ToString());
                foreCol = int.Parse(kSOPrefs.GetValue("ForeColor", "-2147483640").ToString());
                backCol = int.Parse(kSOPrefs.GetValue("BackColor", "-2147483643").ToString());
                focusCol = int.Parse(kSOPrefs.GetValue("FocusColor", "-2147483643").ToString());
            }
            else
            {
                language = "EN";
                logPath = "c:\\windows\\logs\\";
                formForeCol = -2147483630;
                formBackCol = -2147483633;
                foreCol = -2147483640;
                backCol = -2147483643;
                focusCol = -2147483643;
            }
        }
        catch (Exception ex)
        {
            //handle exception here;
        }
    }

    public static string Language
    {
        get
        {
            return language;
        }
        set
        {
            language = value;
        }
    }

    public static string LogPath
    {

```

```

    get
    {
        return logPath;
    }
    set
    {
        logPath = value;
    }
}

public static Color FormBackColor
{
    get
    {
        return ColorTranslator.FromOle(formBackCol);
    }
    set
    {
        formBackCol = ColorTranslator.ToOle(value);
    }
}

public static Color FormForeColor
{
    get
    {
        return ColorTranslator.FromOle(formForeCol);
    }
    set
    {
        formForeCol = ColorTranslator.ToOle(value);
    }
}

public static Color BackColor
{
    get
    {
        return ColorTranslator.FromOle(backCol);
    }
    set
    {
        backCol = ColorTranslator.ToOle(value);
    }
}

public static Color ForeColor
{
    get
    {
        return ColorTranslator.FromOle(foreCol);
    }
    set
    {
        foreCol = ColorTranslator.ToOle(value);
    }
}

public static Color FocusColor
{
    get

```

```

        {
            return ColorTranslator.FromOle(focusCol);
        }
        set
        {
            focusCol = ColorTranslator.ToOle(value);
        }
    }
}
}

```

Windowssave () ,

. . .

NumberBox

. . . .

```

using System;
using System.Windows.Forms;
using System.Globalization;

namespace StackOverflowDocumentation
{
    public class SONumberBox : SOTextBox
    {
        private int decPlaces;
        private int extraDecPlaces;
        private bool perCent;
        private bool useThouSep = true;
        private string decSep = ".";
        private string thouSep = ",";
        private double numVal;

        public SONumberBox() : base()

        {
        }

        public bool PerCent
        {
            get
            {
                return perCent;
            }
            set
            {
                perCent = value;
            }
        }

        public double Value
        {
            get
            {
                return numVal;
            }
        }
    }
}

```



```

    set
    {
        numVal = value;
        if (perCent)
        {
            double test = numVal * 100.0;
            this.Text = FormatNumber(test) + "%";
        }
        else
        {
            this.Text = FormatNumber(value);
        }
    }
}
public bool UseThousandSeparator
{
    get
    {
        return useThouSep;
    }
    set
    {
        useThouSep = value;
    }
}
public int DecimalPlaces
{
    get
    {
        return decPlaces;
    }
    set
    {
        decPlaces = value;
    }
}
public int ExtraDecimalPlaces
{
    get
    {
        return extraDecPlaces;
    }
    set
    {
        extraDecPlaces = value;
    }
}
protected override void OnTextChanged(EventArgs e)
{
    string newVal = this.Text;
    int len = newVal.Length;
    if (len == 0)
    {
        return;
    }
    bool neg = false;
    if (len > 1)
    {
        if (newVal.Substring(0, 1) == "-")
        {
            newVal = newVal.Substring(1, len - 1);
        }
    }
}

```

```

        len = newVal.Length;
        neg = true;
    }
}
double val = 1.0;
string endChar = newVal.Substring(newVal.Length - 1);
switch (endChar)
{
    case "M":
    case "m":
        if (len > 1)
        {
            val = double.Parse(newVal.Substring(0, len - 1)) * 1000000.0;
        }
        else
        {
            val *= 1000000.0;
        }
        if (neg)
        {
            val = -val;
        }
        this.Text = FormatNumber(val);
        break;
    case "T":
    case "t":
        if (len > 1)
        {
            val = double.Parse(newVal.Substring(0, len - 1)) * 1000.0;
        }
        else
        {
            val *= 1000.0;
        }
        if (neg)
        {
            val = -val;
        }
        this.Text = FormatNumber(val);
        break;
}

base.OnTextChanged(e);
}
protected override void OnKeyPress(KeyPressEventArgs e)
{
    bool handled = false;
    switch (e.KeyChar)
    {
        case '-':
            if (this.Text.Length == 0)
            {
                break;
            }
            else if (this.SelectionStart == 0)
            {
                //negative being inserted first
                break;
            }
            else
            {

```

```

        handled = true;
        break;
    }
    case '1':
    case '2':
    case '3':
    case '4':
    case '5':
    case '6':
    case '7':
    case '8':
    case '9':
    case '0':
    case (char)Keys.Back:
        break;
    case 'M':
    case 'm':
    case 'T':
    case 't':
    case '%':
        //check last pos
        int l = this.Text.Length;
        int sT = this.SelectionStart;
        int sL = this.SelectionLength;
        if ((sT + sL) != l)
        {
            handled = true;
        }
        break;
    default:
        string thisChar = e.KeyChar.ToString();
        if (thisChar == decSep)
        {
            char[] txt = this.Text.ToCharArray();
            for (int i = 0; i < txt.Length; i++)
            {
                if (txt[i].ToString() == decSep)
                {
                    handled = true;
                    break;
                }
            }
            break;
        }
        else if (thisChar != thouSep)
        {
            handled = true;
        }
        break;
    }

    if (!handled)
    {
        base.OnKeyPress(e);
    }
    else
    {
        e.Handled = true;
    }
}

```

```

protected override void OnLeave(EventArgs e)
{
    string tmp = this.Text;
    if (tmp == "")
    {
        tmp = "0";
        numVal = NumberLostFocus(ref tmp);
        this.Text = tmp;
    }
    if (tmp.Substring(tmp.Length - 1) == "%")
    {
        tmp = tmp.Substring(0, tmp.Length - 1);
        numVal = 0.0;
        numVal = NumberLostFocus(ref tmp) / 100.0;
        double test = numVal * 100.0;
        this.Text = FormatNumber(test) + "%";
    }
    else if (perCent)
    {
        numVal = NumberLostFocus(ref tmp);
        double test = numVal * 100.0;
        this.Text = FormatNumber(test) + "%";
    }
    else
    {
        numVal = NumberLostFocus(ref tmp);
        this.Text = tmp;
    }
    base.OnLeave(e);
}
private string FormatNumber(double amount)
{
    NumberFormatInfo nF = new NumberFormatInfo();
    nF.NumberDecimalSeparator = decSep;
    nF.NumberGroupSeparator = thouSep;

    string decFormat;
    if (useThouSep)
    {
        decFormat = "#,##0";
    }
    else
    {
        decFormat = "#0";
    }
    if (decPlaces > 0)
    {
        decFormat += ".";
        for (int i = 0; i < decPlaces; i++)
        {
            decFormat += "0";
        }
        if (extraDecPlaces > 0)
        {
            for (int i = 0; i < extraDecPlaces; i++)
            {
                decFormat += "#";
            }
        }
    }
    else if (extraDecPlaces > 0)

```

```

    {
        decFormat += ".";
        for (int i = 0; i < extraDecPlaces; i++)
        {
            decFormat += "#";
        }
    }
    return (amount.ToString(decFormat, nF));
}
private double NumberLostFocus(ref string amountBox)
{
    if (amountBox.Substring(0, 1) == decSep)
        amountBox = "0" + amountBox;
    NumberFormatInfo nF = new NumberFormatInfo();
    nF.NumberDecimalSeparator = decSep;
    nF.NumberGroupSeparator = thouSep;

    try
    {
        double d = 0.0;
        int l = amountBox.Length;
        if (l > 0)
        {
            char[] c = amountBox.ToCharArray();
            char endChar = c[l - 1];

            switch (endChar)
            {
                case '0':
                case '1':
                case '2':
                case '3':
                case '4':
                case '5':
                case '6':
                case '7':
                case '8':
                case '9':
                    {
                        stripNonNumerics(ref amountBox);
                        d = Double.Parse(amountBox, nF);
                        break;
                    }
                case 'm':
                case 'M':
                    {
                        if (amountBox.Length == 1)
                            d = 1000000.0;
                        else
                        {
                            string s = amountBox.Substring(0, l - 1);
                            stripNonNumerics(ref s);
                            d = Double.Parse(s, nF) * 1000000.0;
                        }
                        break;
                    }
                case 't':
                case 'T':
                    {

```

```

        if (amountBox.Length == 1)
            d = 1000.0;
        else
        {
            string s = amountBox.Substring(0, l - 1);
            stripNonNumerics(ref s);
            d = Double.Parse(s, nF) * 1000.0;
        }
        break;
    }
default:
    {
        //remove offending char
        string s = amountBox.Substring(0, l - 1);
        if (s.Length > 0)
        {
            stripNonNumerics(ref s);
            d = Double.Parse(s, nF);
        }
        else
            d = 0.0;
        break;
    }
}
}
amountBox = FormatNumber(d);
return (d);
}
catch (Exception e)
{
    //handle exception here;
    return 0.0;
}
}
private void stripNonNumerics(ref string amountBox)
{
    bool dSFound = false;
    char[] tmp = decSep.ToCharArray();
    char dS = tmp[0];
    string cleanNum = "";
    int l = amountBox.Length;
    if (l > 0)
    {
        char[] c = amountBox.ToCharArray();
        for (int i = 0; i < l; i++)
        {
            char b = c[i];
            switch (b)
            {
                case '0':
                case '1':
                case '2':
                case '3':
                case '4':
                case '5':
                case '6':
                case '7':
                case '8':
                case '9':
                    cleanNum += b;
                    break;
            }
        }
    }
}

```

```

        case '-':
            if (i == 0)
                cleanNum += b;
            break;
        default:
            if ((b == dS) && (!dSFound))
            {
                dSFound = true;
                cleanNum += b;
            }
            break;
    }
}
amountBox = cleanNum;
}
}

```

. , , .10M 10,000,000.00 (). . .

: <https://riptutorial.com/ko/winforms/topic/6476/>

6:

WinForms .

Examples

WinForms .

- -

```
Form1 aForm1Instance = new Form1();
aForm1Instance.Show();
```

- -

```
Form2 aForm2Instance = new Form2();
aForm2Instance.ShowDialog();
```

. (1) . . ()

(MDI) ().

. Close ?

.

```
theGlobalInstance.Close();
theGlobalInstance.Dispose();
theGlobalInstance = null;
```

Application.OpenForms .

Close .

```
Form2 toClose = Application.OpenForms.OfType<Form2>().FirstOrDefault();
if(toClose != null)
{
    toClose.Close();
    toClose.Dispose();
}
```

ShowDialog DialogResult . DialogResult enum .

DialogResult (DialogResult.None) . WinForm ShowDialog .

```
private cmdClose_Click(object sender, EventArgs e)
{
    this.DialogResult = DialogResult.Cancel;
}
```


ShowDialog . ShowDialog() () using .

OpenFileDialog , , .

```
using (var form = new OpenFileDialog())
{
    DialogResult result = form.ShowDialog();
    if (result == DialogResult.OK)
    {
        MessageBox.Show("Selected file is: " + form.FileName);
    }
}
```

DialogResult . DialogResult . DialogResult .

, OK DialogResult.OK ShowDialog() DialogResult.OK .

: <https://riptutorial.com/ko/winforms/topic/8768/-->

7:

Examples

```
var source = new AutoCompleteStringCollection();

// Add your collection of strings.
source.AddRange(new[] { "Guybrush Threepwood", "LeChuck" });

var textBox = new TextBox
{
    AutoCompleteCustomSource = source,
    AutoCompleteMode = AutoCompleteMode.SuggestAppend,
    AutoCompleteSource = AutoCompleteSource.CustomSource
};

form.Controls.Add(textBox);
```

Global

`AutoCompleteMode.SuggestAppend` `Append Suggest` .

```
textBox.KeyPress += (sender, e) => e.Handled = !char.IsControl(e.KeyChar) &&
!char.IsDigit(e.KeyChar);
```

`TextBox` `Handle true` .

/ `TextChanged` .

```
textBox.TextChanged += (sender, e) => textBox.Text = Regex.Match(textBox.Text, @"\d+").Value
```

NumericUpDown

```
textBox.SelectionStart = textBox.TextLength;
textBox.ScrollToCaret();
```

`SelectionStart 0` .

#

```
private void Form_load(object sender, EventArgs e)
{
    textBox.Text = "Place Holder text...";
}

private void textBox_Enter(object sender, EventArgs e)
```

```

{
    if(textBox.Text == "Place Holder text...")
    {
        textBox.Text = "";
    }
}

private void textBox_Leave(object sender, EventArgs e)
{
    if(textBox.Text.Trim() == "")
    {
        textBox.Text = "Place Holder text...";
    }
}

```

VB.NET

```

Private Sub Form_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    textBox.Text = "Place Holder text..."
End Sub

Private Sub textBox_GotFocus(sender as Object,e as EventArgs) Handles textBox.GotFocus
    if Trim(textBox.Text) = "Place Holder text..." Then
        textBox.Text = ""
    End If
End Sub

Private Sub textBox_LostFocus(sender as Object,e as EventArgs) Handles textBox.LostFocus
    if Trim(textBox.Text) = "" Then
        textBox.Text = "Place Holder text..."
    End If
End Sub

```

: <https://riptutorial.com/ko/winforms/topic/4674/>

S. No		Contributors
1	winforms	4444 , Bjørn-Roger Kringsjå , Chris Shao , Cody Gray , Community , Reza Aghaei
2		Aimnox , Berkay , help-info.de , Jeff Bridgman
3		Kai Thoma
4		help-info.de , Reza Aghaei
5		Balagurunathan Marimuthu
6		Cody Gray , Jeff Bridgman , Steve
7		gplumb , Jones Joseph , Stefano d'Antonio