

Chương 2: Lập trình ứng dụng windows Forms



I MỤC TIÊU

- Vận dụng các control để thiết kế ứng dụng
- Xử lý các sự kiện với các control

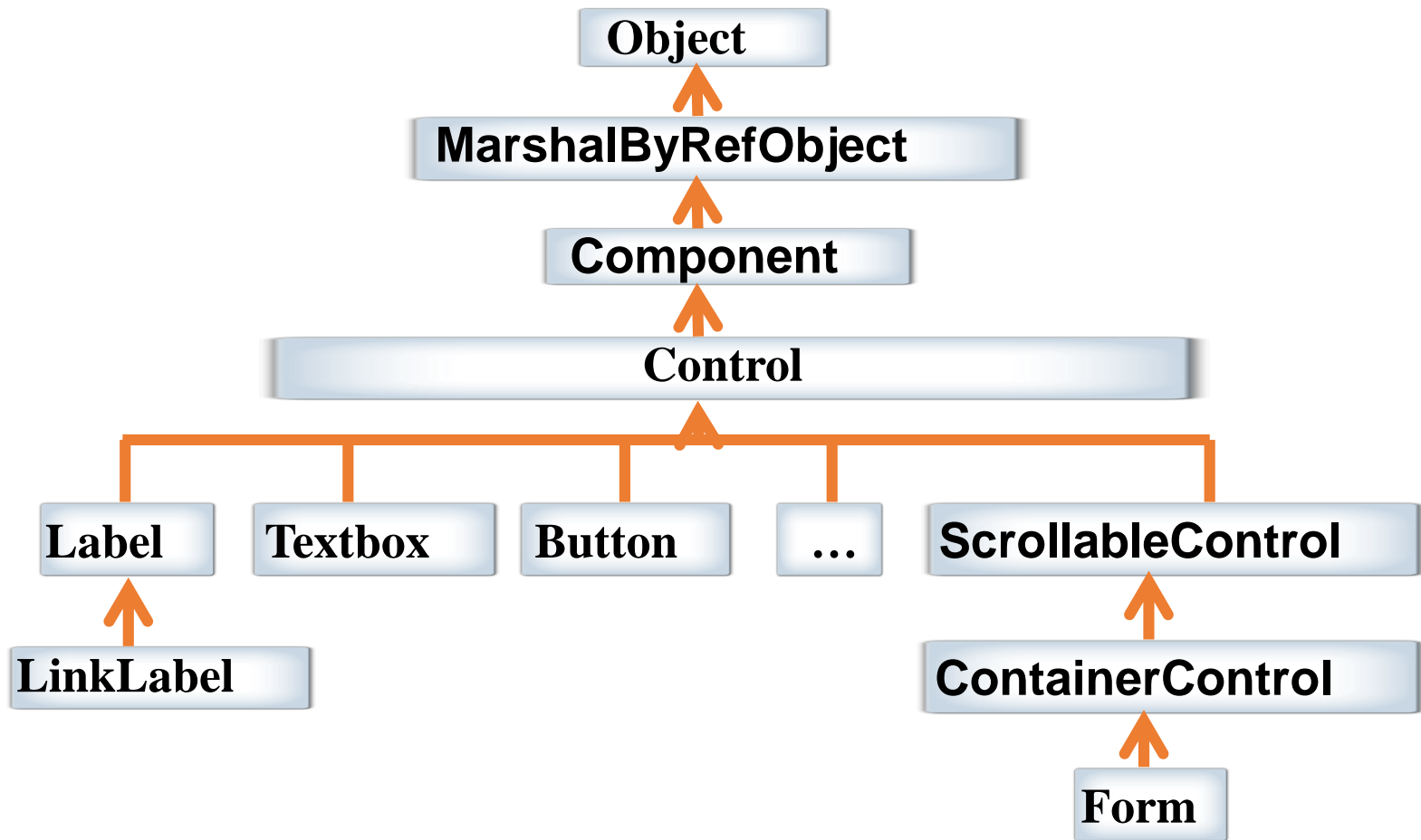


I NỘI DUNG

- Các thành phần ứng dụng windows form
- Sử dụng các control để thiết kế ứng dụng
- Sử dụng các điều khiển chuẩn



Các thành phần của Windows Forms



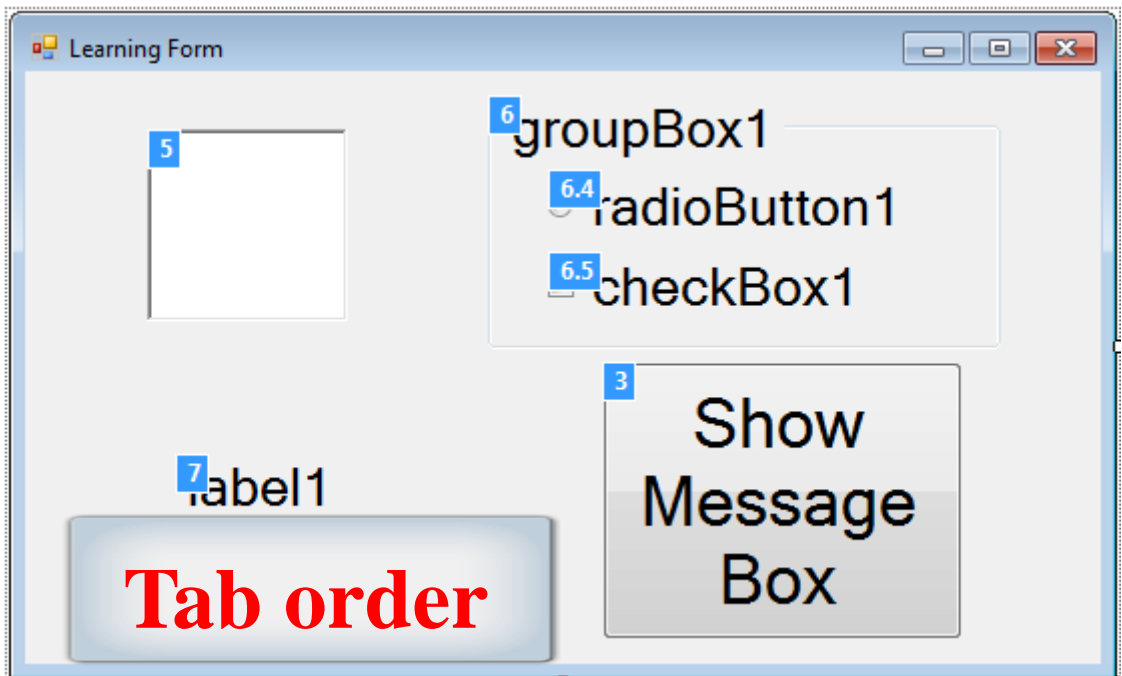
Windows Form

**Create
Form**

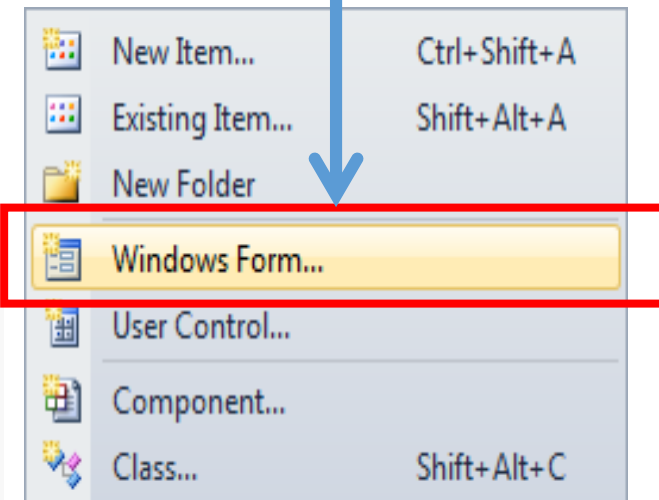
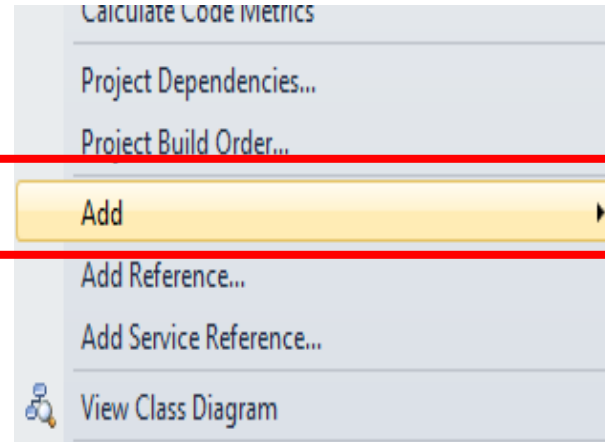
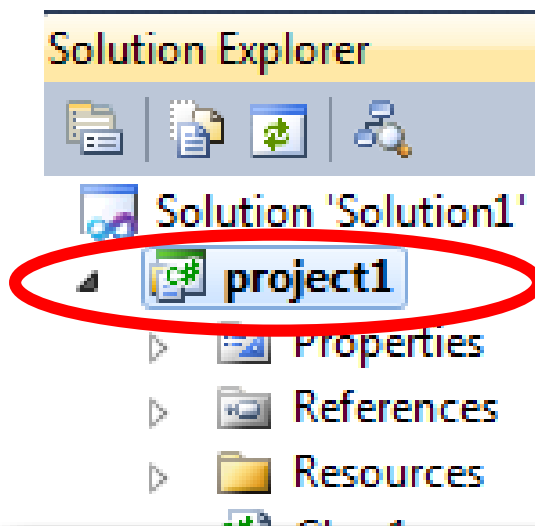
**Popular
Properties**

**Popular
Events**

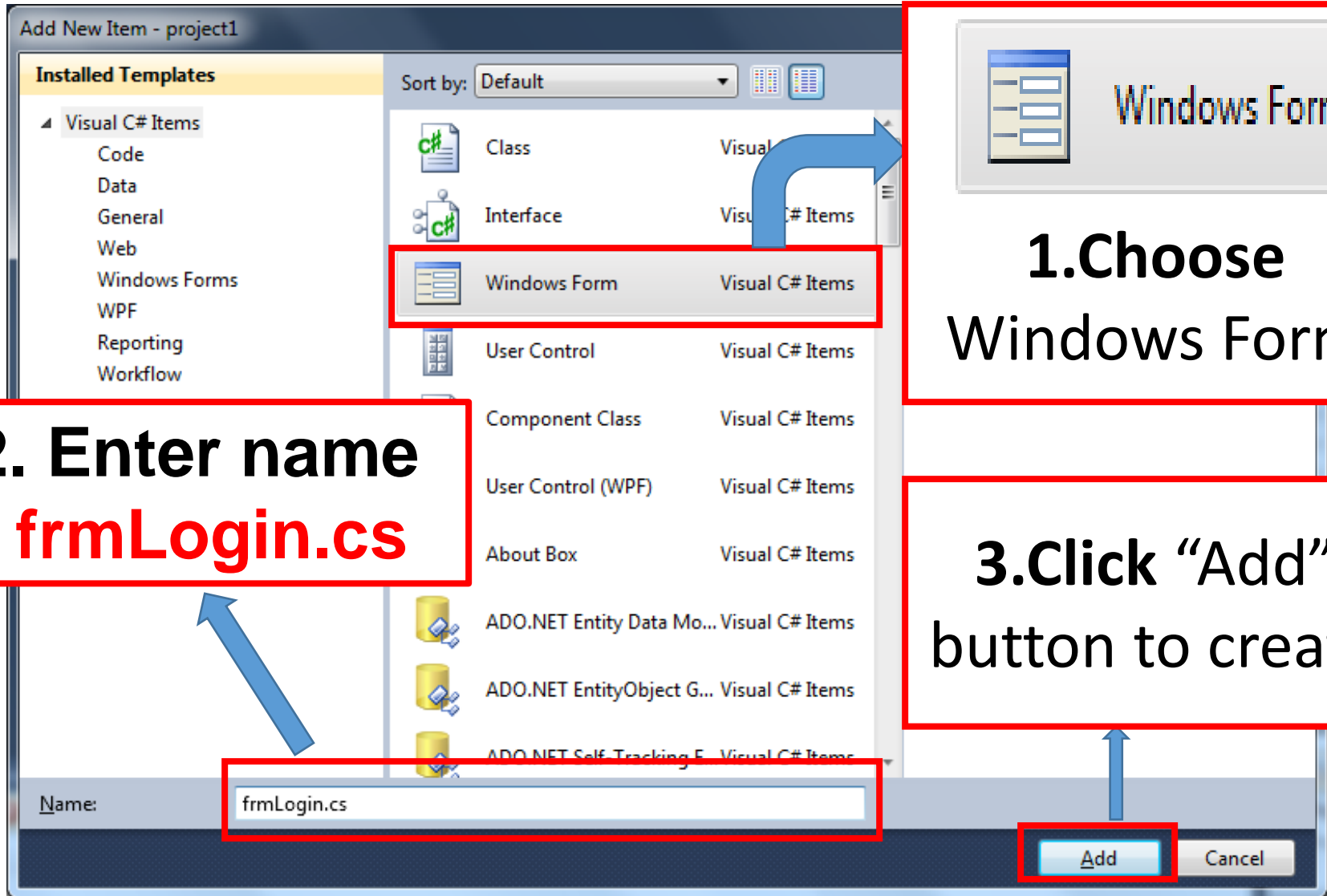
Represents a window or dialog box that makes up an application's user interface



Create Form



1. Right Click on Project Name
2. Choose Add Item
3. Choose Windows Form...



frmLogin.cs

frmLogin.cs [Design]

frmLogin

0 Errors

0 Warnings

0 Messages

	D	File	Line	Column	Project

Solution Explorer

Solution 'Solution1' (3 projects)

project1

Properties

References

Resources

frmLogin.cs

Properties

frmLogin System.Windows.Forms.Form

RightToLeftLayout

ShowIcon

ShowInTaskbar

Size

SizeGripStyle

StartPosition

Tag

Text

TopMost

TransparencyKey

False

True

True

358, 274

Auto

WindowsDefaultLocation

frmLogin

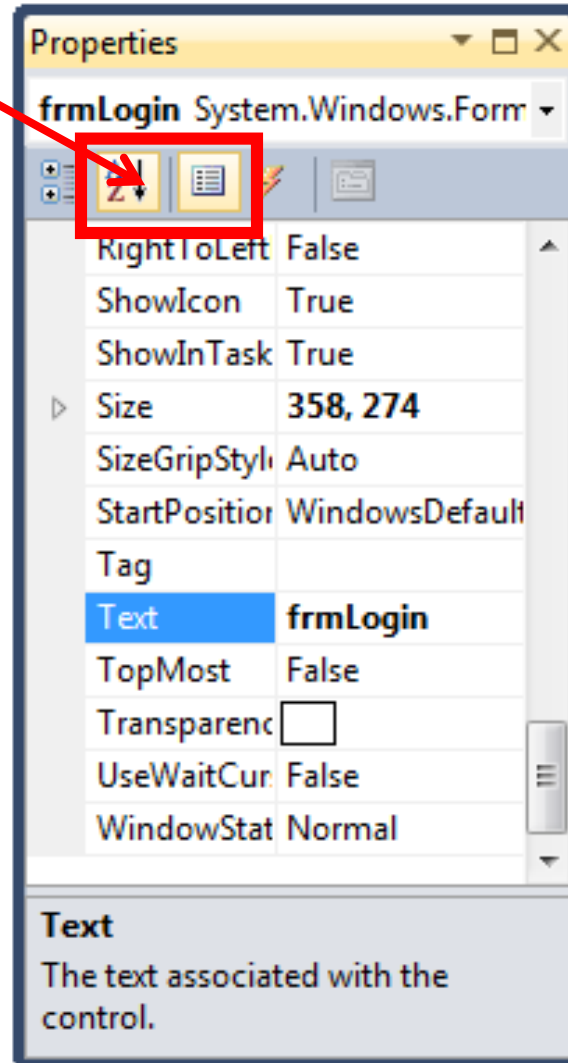
False

SizeGripStyle

Determines when the SizeGrip will be displayed for the form.



Popular Properties



Name	Description
Text	Gets or sets the text associated with this control
Name	Gets or sets the Name
Size	Gets or sets the size of the form.
WindowState	Gets or sets the form's window state. Normal; Minimized; Maximized
Font	Gets or sets the font of the text displayed by the control



Name	Description
StartPosition	<p>Gets or sets the starting position of the form at run time.</p> <p>Manual</p> <p>CenterScreen</p> <p>WindowsDefaultLocation</p> <p>WindowsDefaultBounds</p> <p>CenterParent</p>
TopMost	<p>Gets or sets a value indicating whether the form should be displayed as a topmost form.</p>



Name	Description
FormBorderStyle	<p>Gets or sets the border style of the form</p> <p>None</p> <p>FixedSingle</p> <p>Fixed3D</p> <p>FixedDialog</p> <p>Sizable</p> <p>FixedToolWindow</p> <p>SizableToolWindow</p>



Name	Description
BackColor	Gets or sets the background color for the control
BackGroundImage	Gets or sets the background image displayed in the control
MainMenuStrip	Gets or sets the primary menu container for the form
ForceColor	Gets or sets the foreground color of the control



Name	Description
Cursor	Gets or sets the cursor that is displayed when the mouse pointer is over the control
Location	Gets or sets the Point that represents the upper-left corner of the Form in screen coordinates
Icon	Gets or sets the icon for the form.
Opacity	Gets or sets the opacity level of the form



Name	Description
AcceptButton	Gets or sets the button on the form that is clicked when the user presses the ENTER key.
CancelButton	Gets or sets the button control that is clicked when the user presses the ESC key

(Name)	Form1
AcceptButton	btnSend
AccessibleDescription	(none)
AccessibleName	btnExit
AccessibleRole	btnSend
AllowDrop	linkLabel1

BackgroundImageLayout	Tile
CancelButton	btnExit
CausesValidation	(none)
ContextMenuStrip	btnExit
ControlBox	btnSend
Cursor	linkLabel1
DoubleBuffered	False



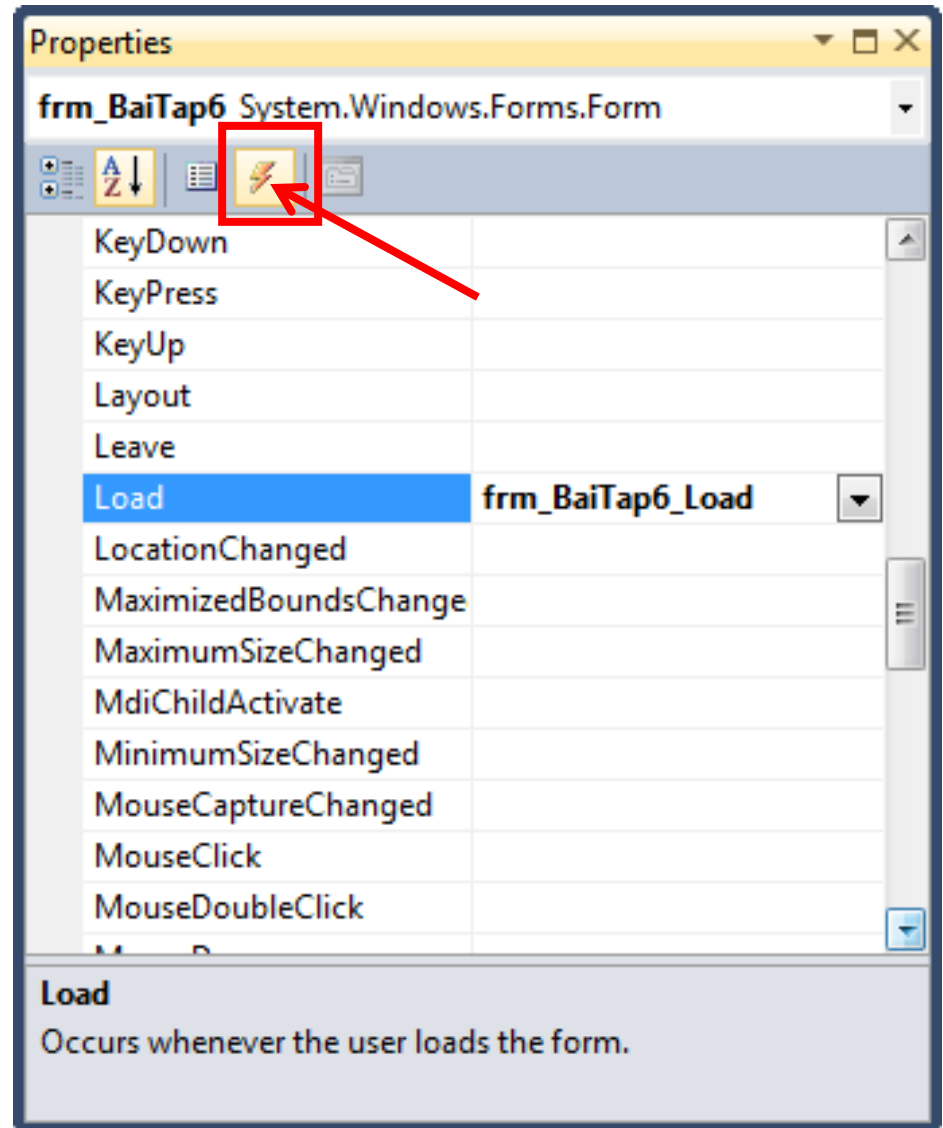
DEMO

Windows Forms



Popular Events

For Windows Forms & Another Controls



Name	Description
Click	Occurs when the control is clicked. (Inherited from Control.)
DoubleClick	Occurs when the control is double-clicked. (Inherited from Control.)
Load	Occurs before a form is displayed for the first time.
FormClosing	Occurs before the form is closed.
FormClosed	Occurs after the form is closed.



Name	Description
KeyDown	Occurs when a key is pressed while the control has focus. (Inherited from Control.)
KeyPress	Occurs when a key is pressed while the control has focus. (Inherited from Control.)
KeyUp	Occurs when a key is released while the control has focus. (Inherited from Control.)



Name	Description
MouseClicked	Occurs when the control is clicked by the mouse. (Inherited from Control.)
MouseDoubleClick	Occurs when the control is double clicked by the mouse. (Inherited from Control.)
MouseDown	Occurs when the mouse pointer is over the control and a mouse button is pressed.(Control)



Name	Description
MouseEnter	Occurs when the mouse pointer enters the control.(Control)
MouseHover	Occurs when the mouse pointer rests on the control(Control)
MouseLeave	Occurs when the mouse pointer leaves the control. (Control)
MouseMove	Occurs when the mouse pointer is moved over the control. (Control)



Name	Description
MouseUp	Occurs when the mouse pointer is over the control and a mouse button is released.(Control)
MouseWheel	Occurs when the mouse wheel moves while the control has focus.(Control)

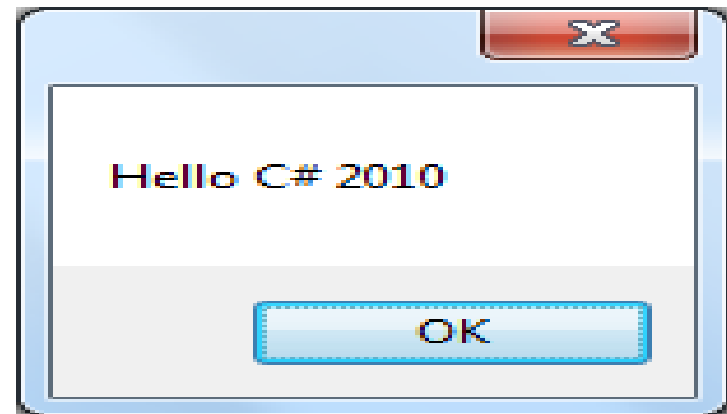
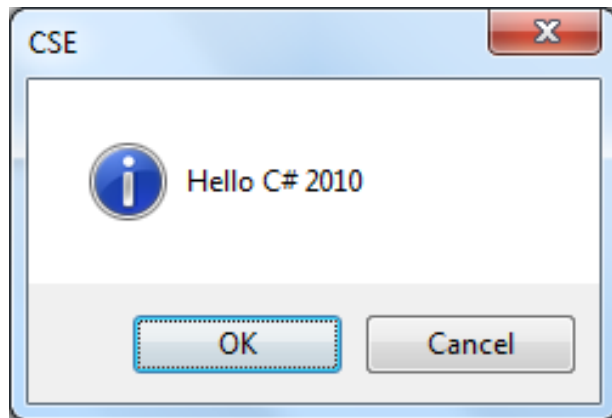
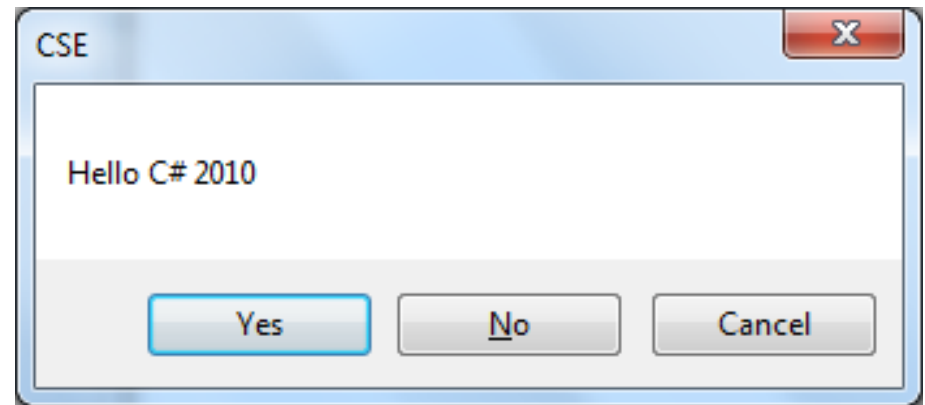
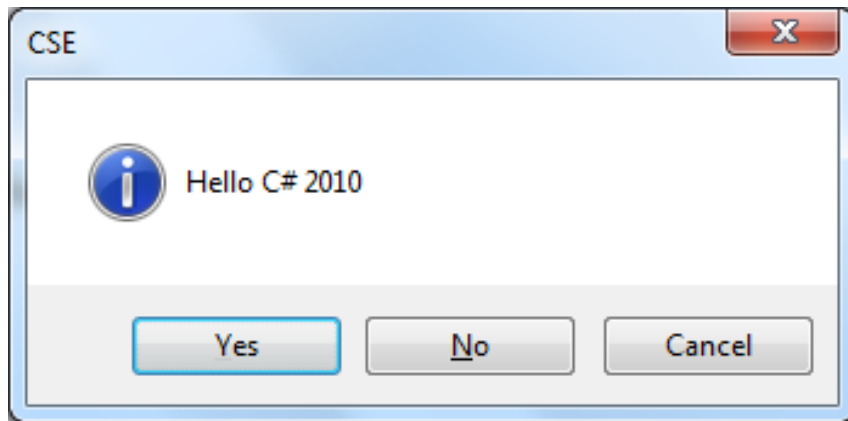


DEMO

Events



MessageBox class



Name	Description
Show(string)	Show a message box with text.
Show(string,string)	Show a message box with text and caption.
Show(string,string, MessageBoxButtons)	Show a message box with text, caption, and buttons.
Show(string,string, MessageBoxButtons, MessageBoxIcon)	Show a message box with text, caption, buttons, and icon.



```
MessageBox.Show("Hello 2017");
```

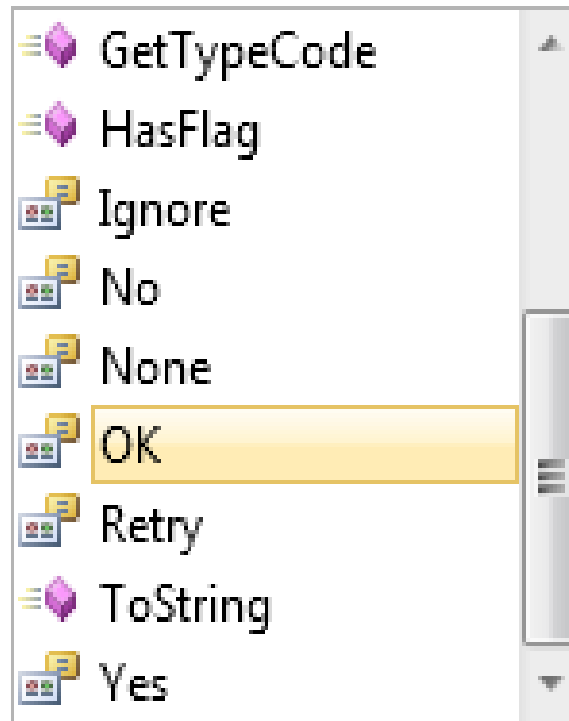
```
MessageBox.Show("Hello 2017", "Title");
```

```
MessageBox.Show("Hello 2017", "Title",  
    MessageBoxButtons.YesNoCancel);
```

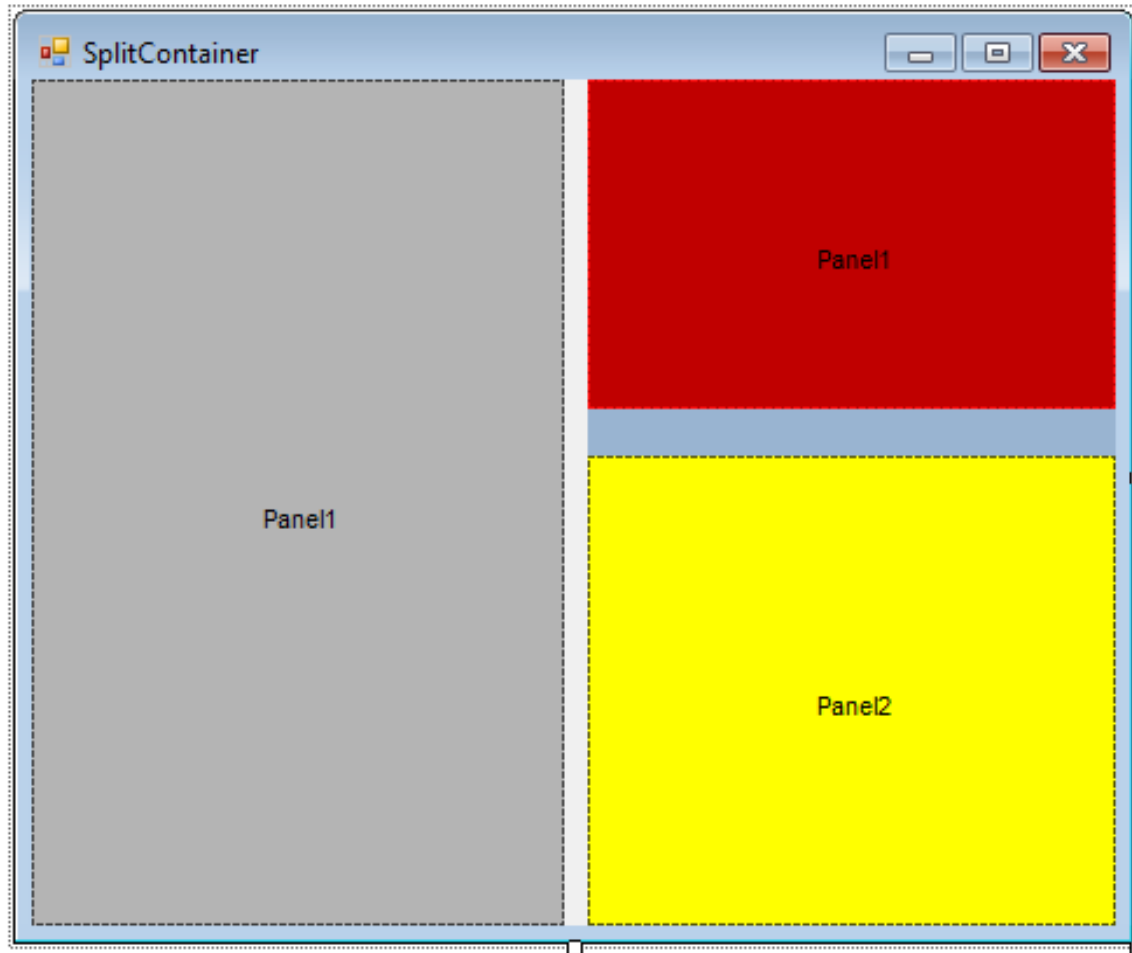
```
MessageBox.Show("Hello 2017", "Title",  
    MessageBoxButtons.OK,  
    MessageBoxIcon.Question);
```



```
DialogResult ret= MessageBox.Show("Hello Tèo 2011",  
                                   "Title tèo",  
                                   MessageBoxButtons.OK,  
                                   MessageBoxIcon.Question);  
  
if(ret==DialogResult.
```



Panel & SplitContainer

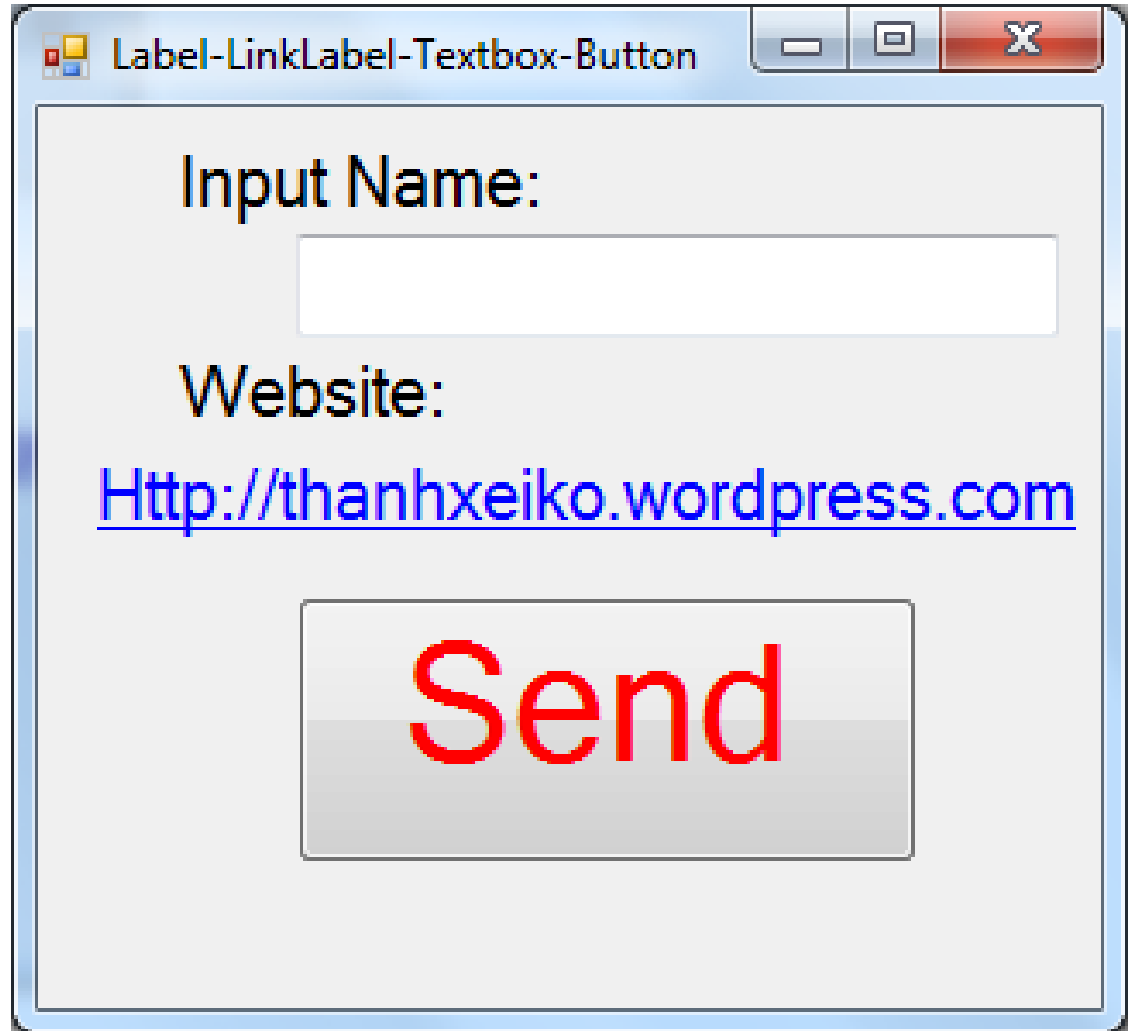


Label

LinkLabel

TextBox

Button



The screenshot shows a Windows application window with the title bar 'Label-LinkLabel-Textbox-Button'. The window contains the following elements:

- A label 'Input Name:' followed by a text box.
- A label 'Website:' followed by a link label displaying the URL [Http://thanhxeiko.wordpress.com](http://thanhxeiko.wordpress.com).
- A button labeled 'Send'.



LinkLabel Events

Name	Description
LinkClicked	<p>The LinkClicked event is handled to perform tasks when the user clicks on a link in the control.</p> <p>It is passed an instance of the <u>LinkLabelLinkClickedEventArgs</u> class</p>



```
private void linkLabel1_LinkClicked(object sender,  
                                   LinkLabelLinkClickedEventArgs e)  
{ switch (e.Button)  
  {  
    case MouseButtons.Left:  
      System.Diagnostics.Process.Start(linkLabel1.Text);  
      break;  
    case MouseButtons.Right:  
      MessageBox.Show("Right Click");  
      break;  
    case MouseButtons.Middle:  
      MessageBox.Show("Middle Click", "Title",  
                      MessageBoxButtons.YesNoCancel,  
                      MessageBoxIcon.Information);  
      break;  
  }  
}
```



How add LinkLabel control & event at Runtime?



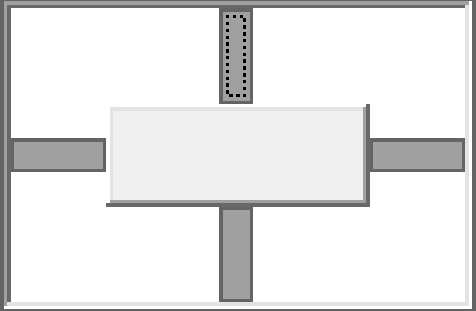

```

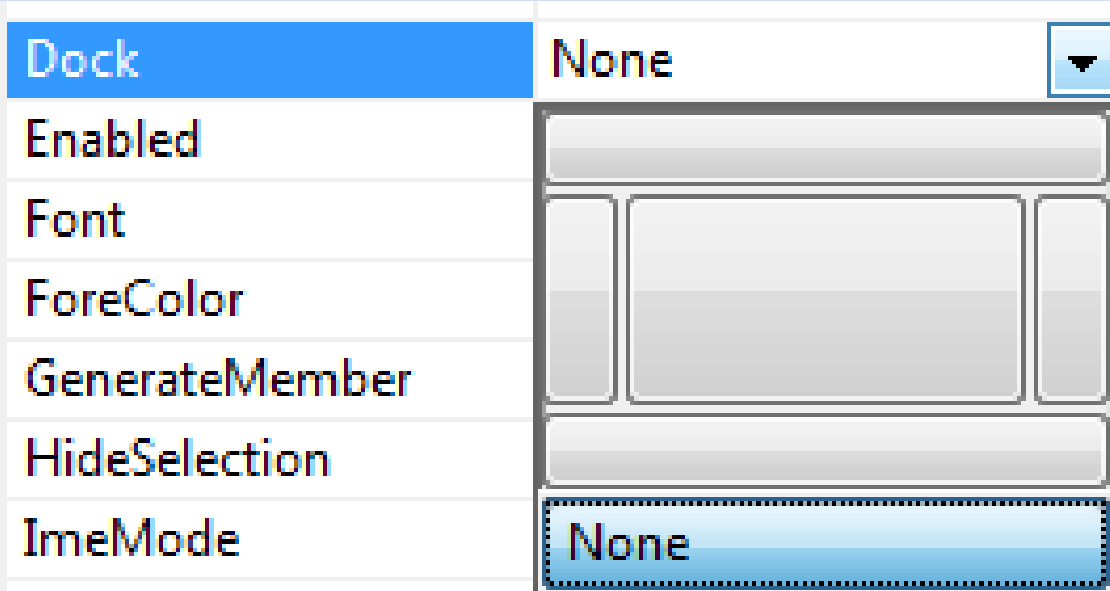
private LinkLabel linkTeo;
private void Form1_Load(object sender, EventArgs e)
{ //Create the linklabel
    linkTeo = new LinkLabel();
    //set location for linklabel
linkTeo.Location =new System.Drawing.Point(100, 100);
    //Set the text for linklabel
linkTeo.Text= "http://thanhxeiko.wordpress.com";
    //Add the controls to the form
this.Controls.Add(linkTeo);
    //Add an event at runtime
linkTeo.LinkClicked += linkTeo_Clicked;
}
private void linkTeo_Clicked(object sender,
    LinkLabelLinkClickedEventArgs e)
{ if (sender is LinkLabel)
    MessageBox.Show(((LinkLabel)sender).Text);
    MessageBox.Show("Add event at runtime");
}

```



TextBox – Properties - Events

Name	Description
Anchor	<p>Gets or sets the edges of the container to which a control is bound and determines how a control is resized with its parent</p> <p>Top Left Right Bottom</p> <div><div><div>Anchor</div><div>AutoCompleteCustom</div><div>AutoCompleteMode</div><div>AutoCompleteSource</div><div>BackColor</div><div>BorderStyle</div></div><div>Top, Bottom, Left, Right ▼</div></div>  A diagram showing a rectangular control within a container. Four thick gray lines extend from the center of each side of the rectangle: one pointing up to the top edge, one pointing down to the bottom edge, one pointing left to the left edge, and one pointing right to the right edge. This illustrates how the control is anchored to all four edges of its container.

Name	Description
Dock	<p data-bbox="388 225 1860 606">Gets or sets which control borders are docked to its parent control and determines how a control is resized with its parent</p> <div data-bbox="388 635 656 1206"> <p>None</p> <p>Top</p> <p>Left</p> <p>Right</p> <p>Bottom</p> <p>Fill</p> </div> 

Name	Description
Multiline	Gets or sets a value indicating whether this is a
PasswordChar	Gets or sets the character used to mask characters of a password in a single-line TextBox control
ReadOnly	Gets or sets a value indicating whether text in the text box is read-only.



Name	Description
UseSystemPasswordChar	Gets or sets a value indicating whether the text in the TextBox control should appear as the default password character
Visible	Gets or sets a value indicating whether the control and all its child controls are displayed.
Enabled	Gets or sets a value indicating whether the control can respond to user interaction.

Name	Description
TextChanged	Occurs when the Text property value changes.

```
private void txtName_TextChanged  
(object sender, EventArgs e)  
{  
    this.Text = txtName.Text;  
}
```

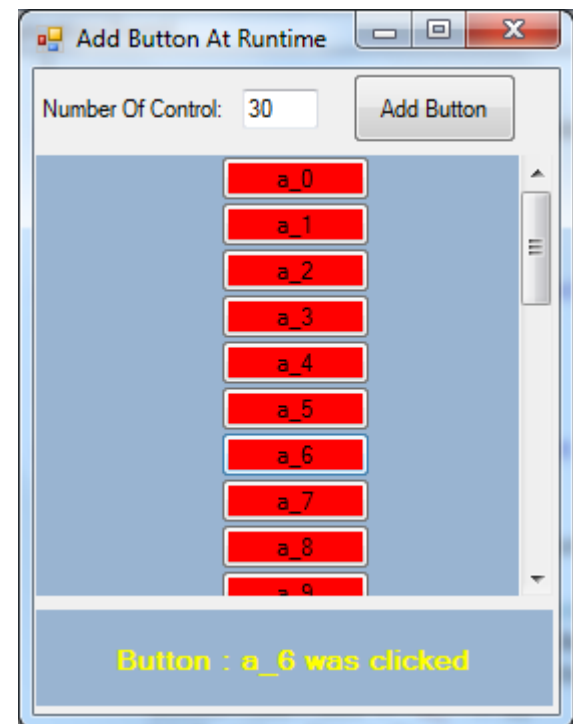
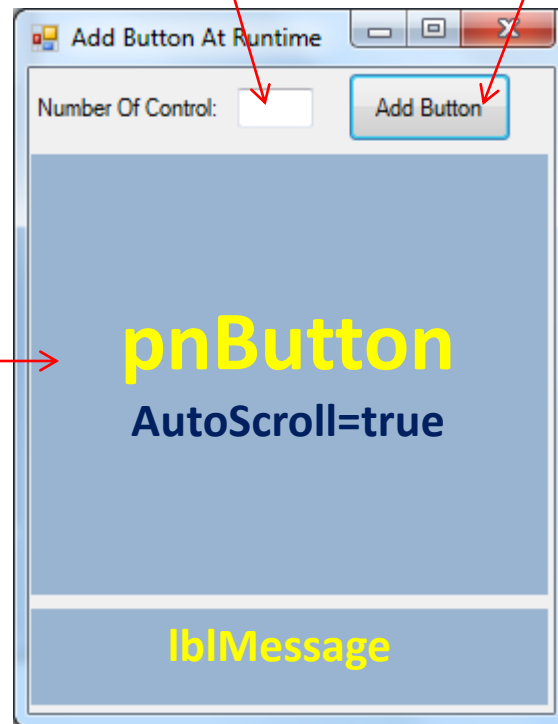


Add Button & Event at Runtime

txtNumberControl

btnAddButton

- Containers
- Pointer
- FlowLayoutPanel
- GroupBox
- Panel
- SplitContainer
- TabControl
- TableLayoutPanel



```

private void btnAddButton_Click(object sender, EventArgs e)
{
    pnButton.Controls.Clear();
    for (int i = 0; i < Int32.Parse(txtNumberControl.Text); i++)
    {
        Button btnRuntime = new Button();
        btnRuntime.BackColor = Color.Red;
        btnRuntime.Location = new System.Drawing.Point
            (pnButton.Width/2-btnRuntime.Width/2,
             i * btnRuntime.Height);
        btnRuntime.Text = "a_" + i;
        btnRuntime.Tag = i;
        btnRuntime.Click += btnRuntime_click;
        pnButton.Controls.Add(btnRuntime);
    }
}

private void btnRuntime_click(object sender, EventArgs e)
{
    Button btn = (Button)sender;
    lblMessage.Text = "Button : " + btn.Text + " was clicked";
}

```



GroupBox & RadioButton

Name	Description
Checked	Gets or sets a value indicating whether the control is checked
CheckAlign	Gets or sets the location of the check box portion of the RadioButton

```
if (radRed.Checked == true)
{
    //do something for Red
}
```

```
radBlue.CheckAlign =
ContentAlignment.MiddleLeft;
```

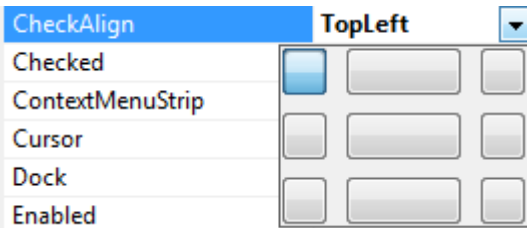


GroupBox & RadioButton

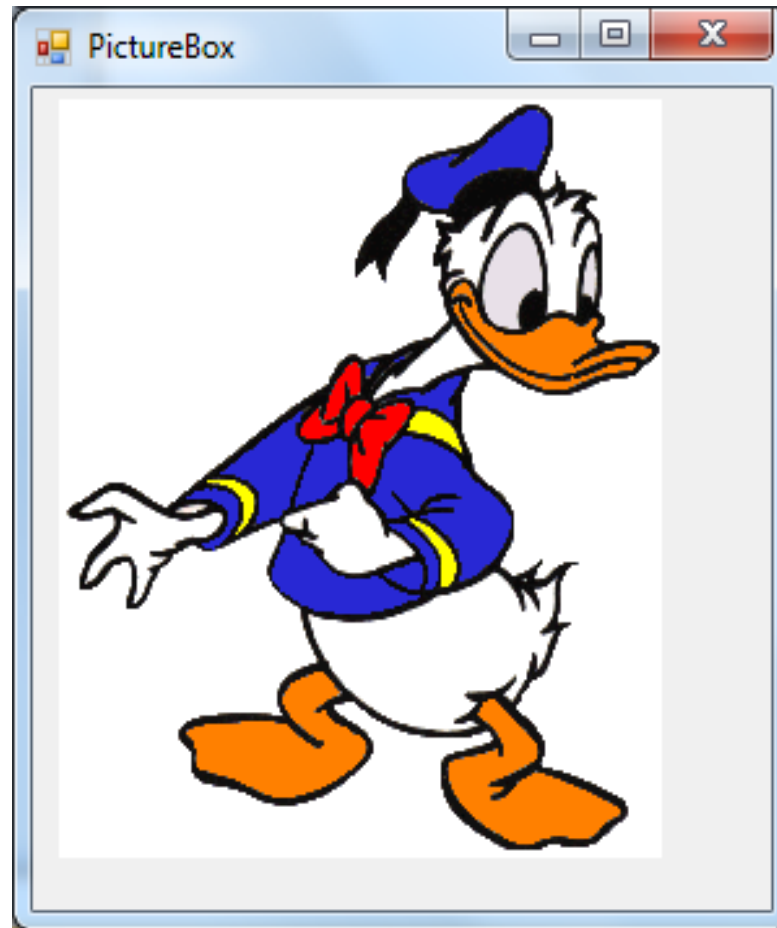
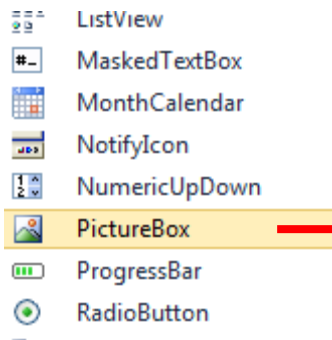


```
private void frmGroupBoxRadio_Load
(object sender, EventArgs e)
{
    radRed.CheckedChanged +=
        rad_CheckedChanged;
    radGreen.CheckedChanged +=
        rad_CheckedChanged;
    radBlue.CheckedChanged +=
        rad_CheckedChanged;
}
```

```
private void rad_CheckedChanged
(object sender, EventArgs e)
{
    RadioButton rad = (RadioButton)sender;
    pnColor.BackColor = rad.ForeColor;
}
```

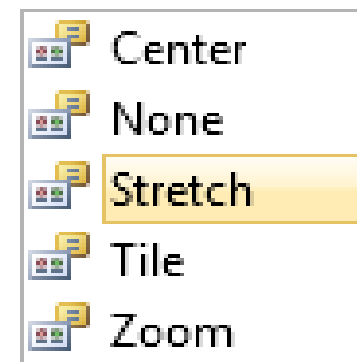


PictureBox control

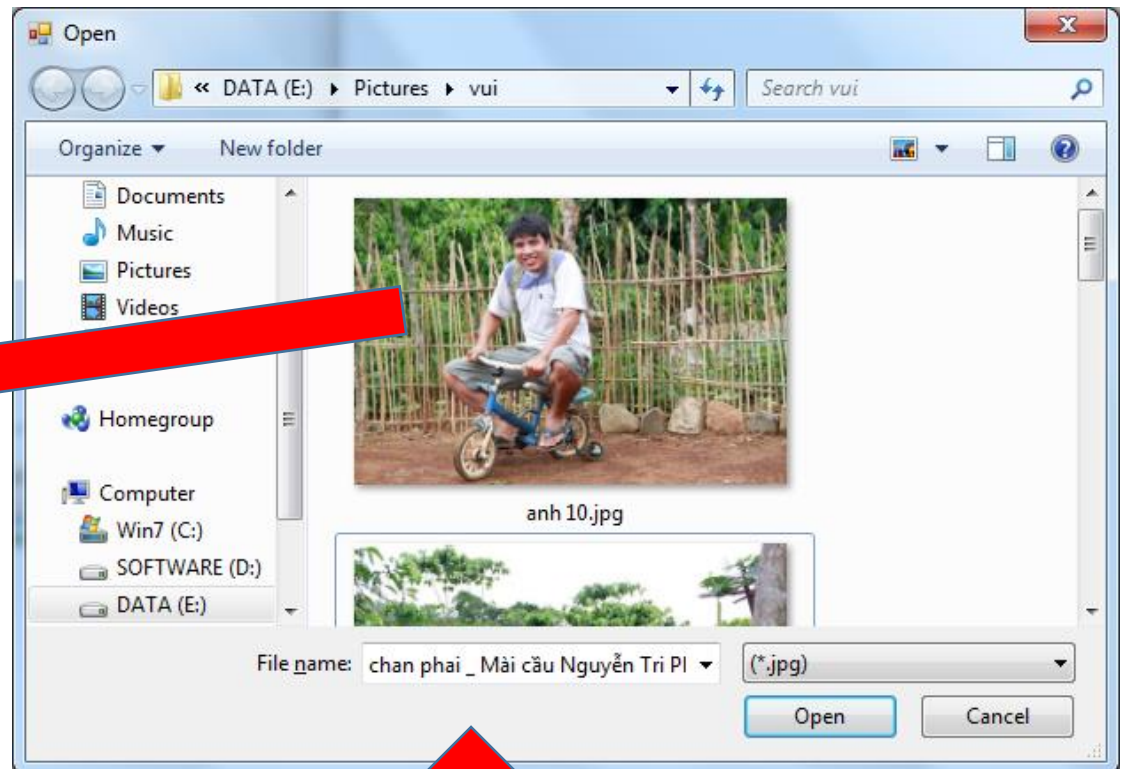


Name	Description
BackgroundImage	Gets or sets the background image displayed in the control
BackgroundImageLayout	Gets or sets the background image layout as defined in the ImageLayout enumeration

```
picDemo.BackgroundImageLayout=ImageLayout.
```

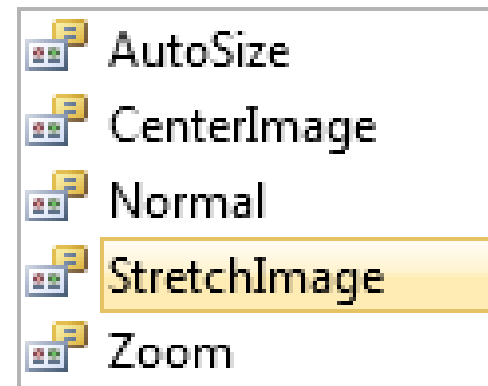


Get Picture from OpenFileDialog



Name	Description
Image	Gets or sets the image that is displayed by PictureBox
SizeMode	Indicates how the image is displayed.

```
picDemo.SizeMode=PictureBoxSizeMode.
```



```
private void btnOpenPic_Click
(object sender, EventArgs e)
{
    OpenFileDialog fileOpenDlg = new OpenFileDialog();
    fileOpenDlg.Filter = "(*.jpg)|*.jpg|(*.doc)|*.doc";
    if (fileOpenDlg.ShowDialog() == DialogResult.OK)
    {
        //picDemo.Image = new Bitmap(fileOpenDlg.OpenFile());
        //Or
        picDemo.Image = Image.FromFile(fileOpenDlg.FileName);
        //Or
        //picDemo.BackgroundImage = new Bitmap(fileOpenDlg.OpenFile());
    }
}
```



| Bài tập áp dụng

16/09/2024

Tổng quan về lập trình ứng dụng

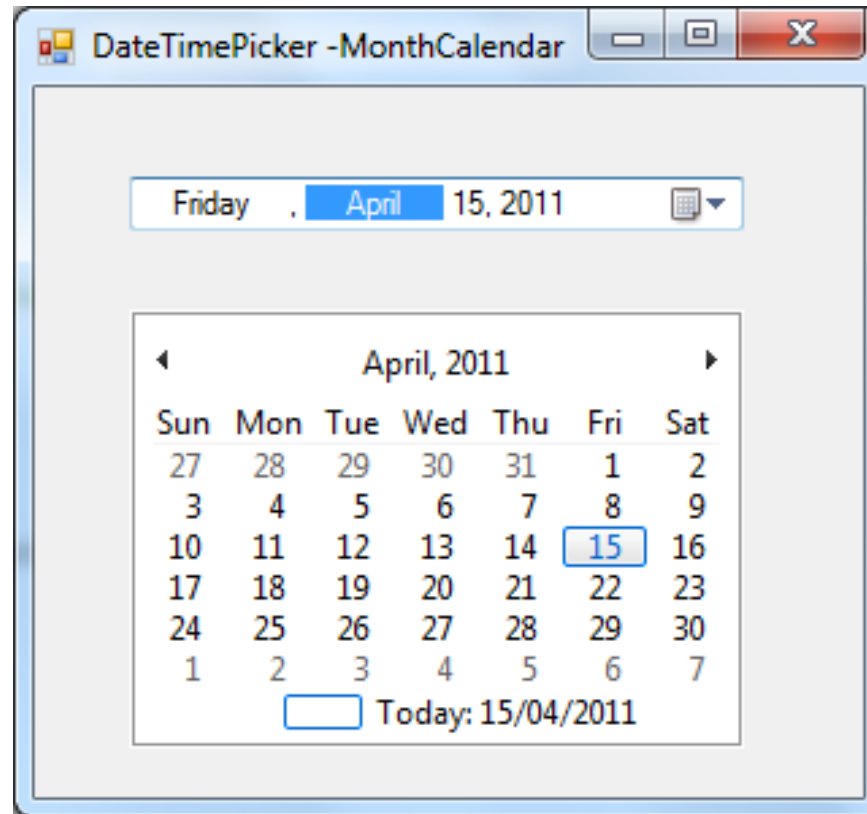
48



FACULTY OF INFORMATION
TECHNOLOGY
THU DUC COLLEGE OF
TECHNOLOGY



DateTimePicker & MonthCalendar



Name	Description
Format	Gets or sets the format of the date and time displayed in the control
CustomFormat	Gets or sets the custom date/time format string
Value	Gets or sets the date/time value assigned to the control.

dateTimePicker1.**Format** = DateTimePickerFormat.**Custom**;
dateTimePicker1.**CustomFormat** = "**MMMM dd, yyyy - dddd**"

Next Slide to see list Custom Format



Format string	Description
d	The one- or two-digit day.
dd	The two-digit day. Single-digit day values are preceded by a 0.
ddd	The three-character day-of-week abbreviation.
dddd	The full day-of-week name.
h	The one- or two-digit hour in 12-hour format.
hh	The two-digit hour in 12-hour format. Single digit values are preceded by a 0.
H	The one- or two-digit hour in 24-hour format.



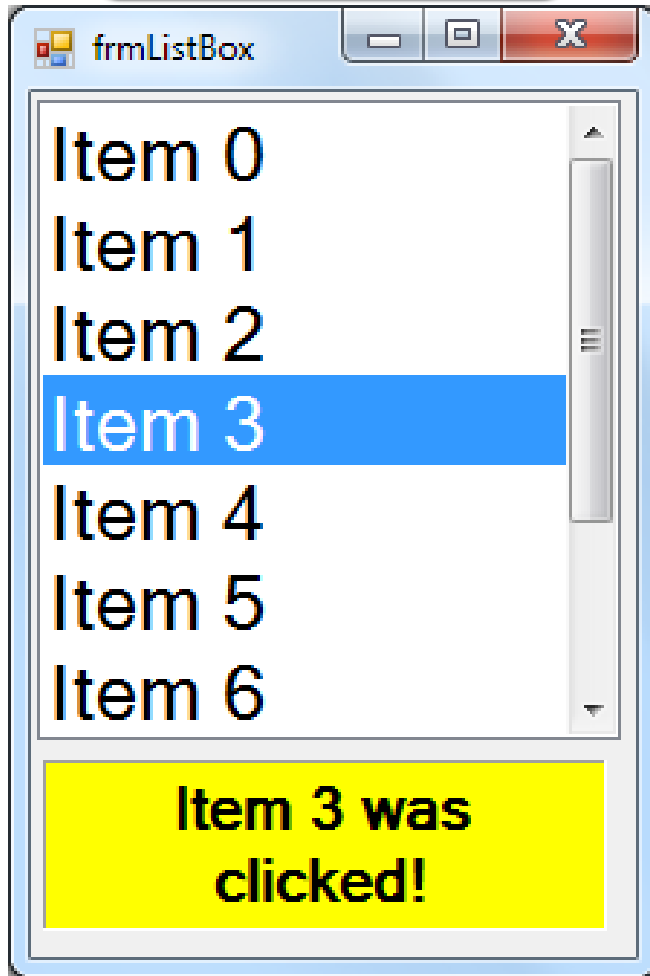
Format string	Description
HH	The two-digit hour in 24-hour format. Single digit values are preceded by a 0.
m	The one- or two-digit minute.
mm	The two-digit minute. Single digit values are preceded by a 0.
M	The one- or two-digit month number.
mm	The two-digit minute. Single digit values are preceded by a 0.
M	The one- or two-digit month number.
MM	The two-digit month number. Single digit values are preceded by a 0.
MMM	The three-character month abbreviation.



Format string	Description
MMMM	The full month name.
s	The one- or two-digit seconds.
ss	The two-digit seconds. Single digit values are preceded by a 0.
t	The one-letter A.M./P.M. abbreviation (A.M. is displayed as "A").
tt	The two-letter A.M./P.M. abbreviation (A.M. is displayed as "AM").
y	The one-digit year (2001 is displayed as "1").
yy	The last two digits of the year (2001 is displayed as "01").
yyyy	The full year (2001 is displayed as "2001").



ListBox



```
private void  
frmListBox_Load(object sender,  
EventArgs e)  
{ listBox1.Items.Clear();  
  for (int i = 0; i < 10; i++)  
    listBox1.Items.Add("Item " + i);  
}  
private void  
listBox1_SelectedIndexChanged  
(object sender, EventArgs e)  
{  
  lblMessage.Text = listBox1.Text  
    + " was clicked!";  
}
```

And We can use **AddRange** method to add data:

```
private void frmListBox_Load(object sender, EventArgs e)
{
    string[] strArr = new string[] {
        "Tèo", "Tí", "Bin", "Bo"};
    listBox1.Items.AddRange(strArr);
}
```



```

public class CStudent
{
    private string m_strID;
    private string m_strName;
    public CStudent(string strID,
                    string strName)
    {
        this.m_strID = strID;
        this.m_strName = strName;
    }
    public string ID
    {
        get { return this.m_strID; }
        set { this.m_strID = value; }
    }
    public string Name
    {
        get { return this.m_strName; }
        set { this.m_strName = value; }
    }
}

```




```
using System.Collections;
```

Also We can use DataSource to display data

```
ArrayList arr = new ArrayList();  
for(int i=0;i<10;i++)  
{arr.Add(new CStudent("ID_"+i,"Name  
"+i));  
}  
listBox1.DataSource = arr;  
listBox1.ValueMember = "ID";  
listBox1.DisplayMember = "Name";
```

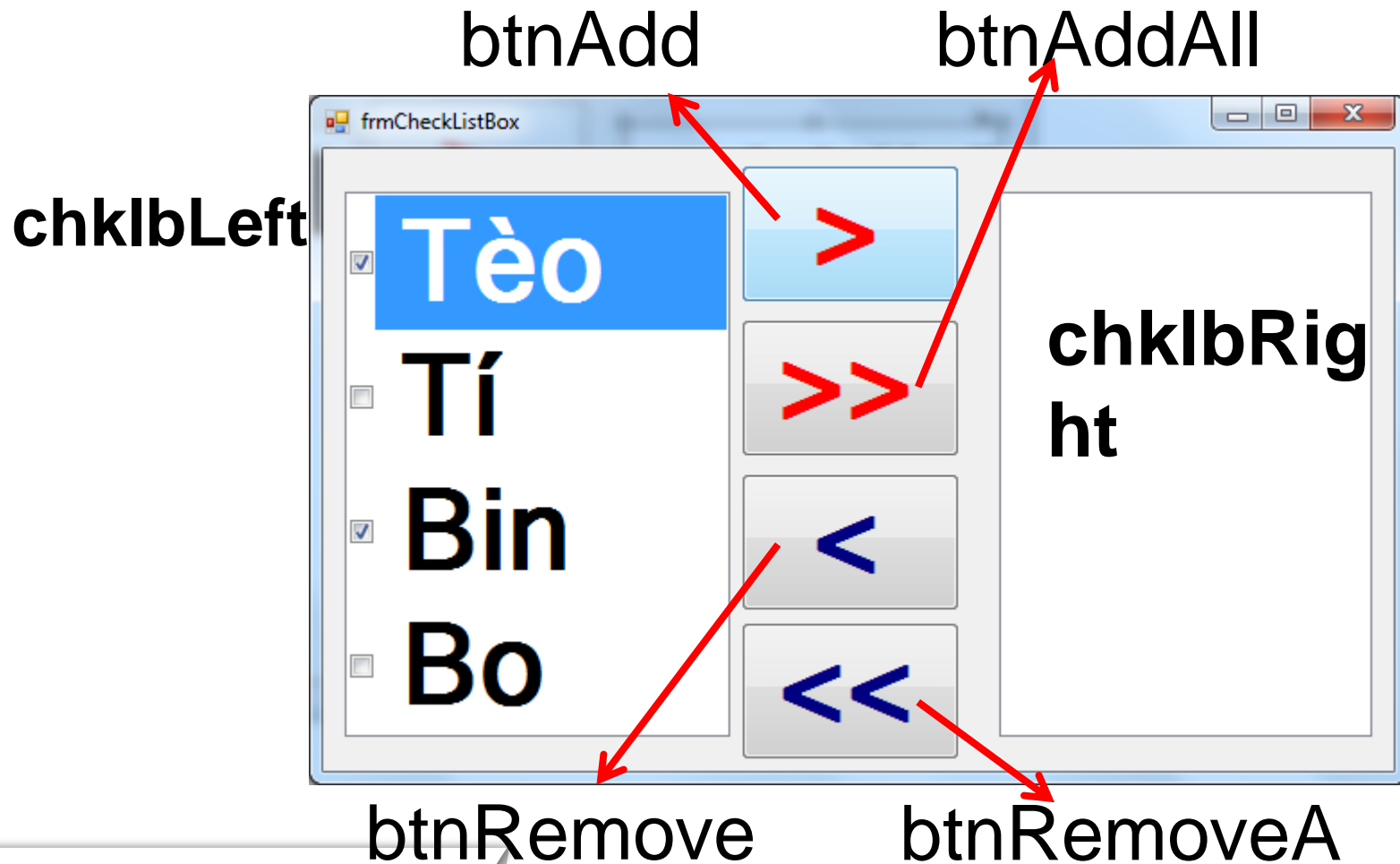


To get **object** from Listbox, We could use coding below:

```
if (listBox1.SelectedItem != null)
{
    CStudent svTeo = (CStudent )
                    listBox1.SelectedItem;
    lblMessage.Text = aStudent.Name
                    + " Was selected";
}
```



CheckedListBox



Use **Items** to add data

```
private void frmCheckListBox_Load  
(object sender, EventArgs e)  
{ chk1bLeft.Items.Add("Tèo");  
  chk1bLeft.Items.Add("Tí");  
  chk1bLeft.Items.Add("Bin");  
  chk1bLeft.Items.Add("Bo");  
}
```

Or we could use **AddRange**

```
chk1bLeft.Items.AddRange(new string[] {  
  "Tèo", "Tí", "Bin", "Bo"});
```



How to process Items Checked???

Case 1:

```
CheckedListBox.CheckedIndexCollection  
indexCollection = chkIbLeft.CheckedIndices;  
string strChecked = "";  
foreach (int i in indexCollection)  
{  
    strChecked += i + " ";  
}  
MessageBox.Show(strChecked);
```



How to process Items Checked???

Case 2:

```
CheckedListBox.CheckedItemCollection items  
= chk1bLeft.CheckedItems;  
string strChecked = "";  
foreach (string s in items)  
{  
    strChecked += s + ";";  
}  
MessageBox.Show(strChecked);
```



How to process Items Checked???

Case 3:

```
string strChecked = "";  
for (int i = 0; i <  
chklbLeft.Items.Count; i++){  
    if (chklbLeft.GetItemChecked(i))  
    {  
        //Process Item checked here  
    }  
}
```



Go back CheckListBox Example:

```
private void btnAdd_Click
(object sender, EventArgs e)
{
    foreach(int i in chk1bLeft.CheckedIndices)
    {
        chk1bRight.Items.Add(chk1bLeft.Items[i]);
    }
    foreach (string s in chk1bRight.Items)
    {chk1bLeft.Items.Remove(s);}
}
```




```
private void btnAddAll_Click
(object sender, EventArgs e)
{
    chk1bRight.Items.AddRange
    (chk1bLeft.Items);
    chk1bLeft.Items.Clear();
}
```



```
private void btnRemove_Click
(object sender, EventArgs e)
{
    foreach (string s in
        chk1bRight.CheckedItems)
        chk1bLeft.Items.Add(s);
    foreach(string s in      chk1bLeft.Items)
        chk1bRight.Items.Remove(s);
}
```



```
private void btnRemoveAll_Click  
(object sender, EventArgs e)  
{  
    chk1bLeft.Items.AddRange  
        (chk1bRight.Items);  
  
    chk1bRight.Items.Clear();  
}
```



| Bài tập áp dụng

16/09/2024

Tổng quan về lập trình ứng dụng

68



FACULTY OF INFORMATION
TECHNOLOGY
THU DUC COLLEGE OF
TECHNOLOGY



Menu (2 ways to use)

MainMenu

MenuStrip

MenuItem

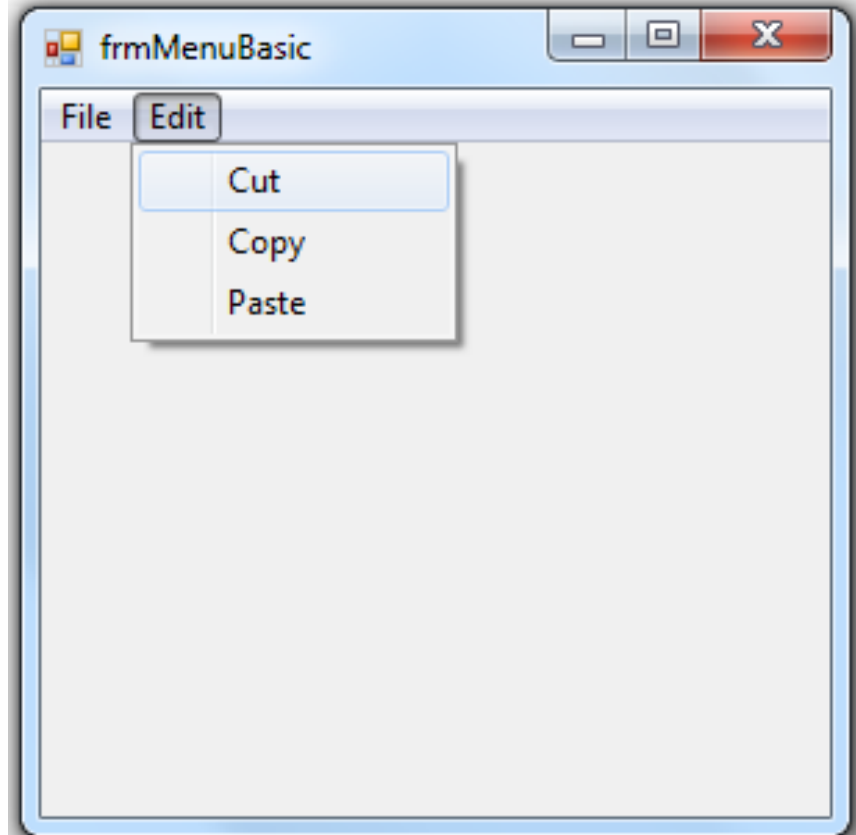
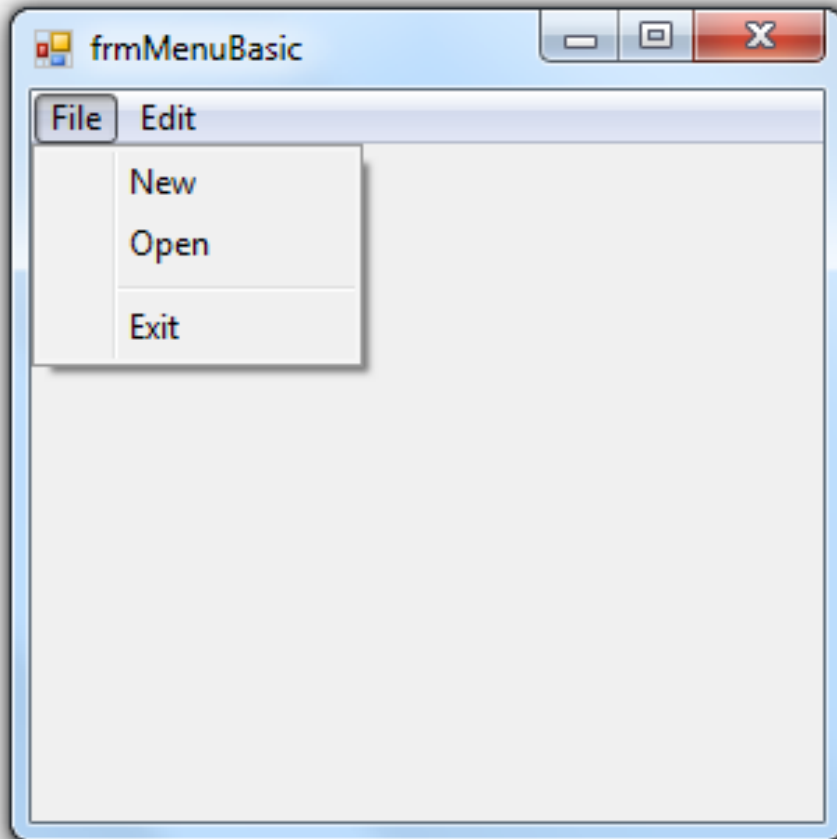
ToolStripMenuItem

**Menu
Property**

**MainMenuStrip
Property**



Coding to create Menu at Runtime



```
private MainMenu mainMenuBar;  
private MenuItem menuFile, menuEdit,  
menuFileNew, menuFileOpen, menuFileExit,  
menuEditCut, menuEditCopy, menuEditPaste;  
private void createMenu()  
{  
mainMenuBar = new MainMenu();  
this.Menu = mainMenuBar;  
menuFile=new MenuItem("File");  
menuFileNew = new MenuItem("New");  
menuFileOpen = new MenuItem("Open");  
menuFileExit = new MenuItem("Exit");  
menuFile.MenuItems.Add(menuFileNew);  
menuFile.MenuItems.Add(menuFileOpen);
```



```
private void createMenu(){.....
menuFile.MenuItems.Add("-");
menuFile.MenuItems.Add(menuFileExit);
mainMenuBar.MenuItems.Add(menuFile);
menuEdit = new MenuItem("Edit");
menuEditCut = new MenuItem("Cut");
menuEditCopy = new MenuItem("Copy");
menuEditPaste = new MenuItem("Paste");
menuEdit.MenuItems.AddRange(new MenuItem[] {
menuEditCut,menuEditCopy,menuEditPaste});
mainMenuBar.MenuItems.Add(menuEdit);
attachEvents();}
```




```
private void attachEvents()
```

```
{  
menuFileNew.Click += process_MenuClick;  
menuFileOpen.Click += process_MenuClick;  
menuFileExit.Click += process_MenuClick;  
menuEditCut.Click += process_MenuClick;  
menuEditCopy.Click += process_MenuClick;  
menuEditPaste.Click += process_MenuClick;  
}
```

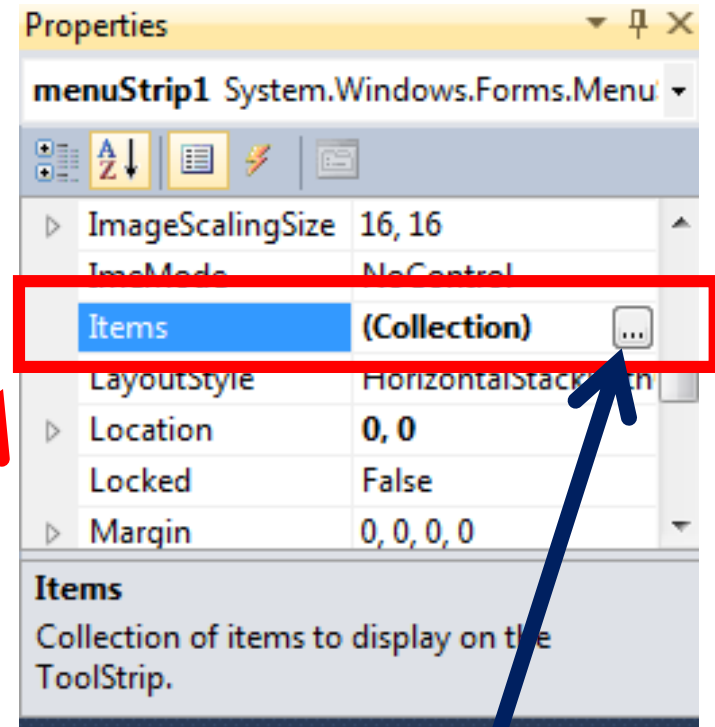
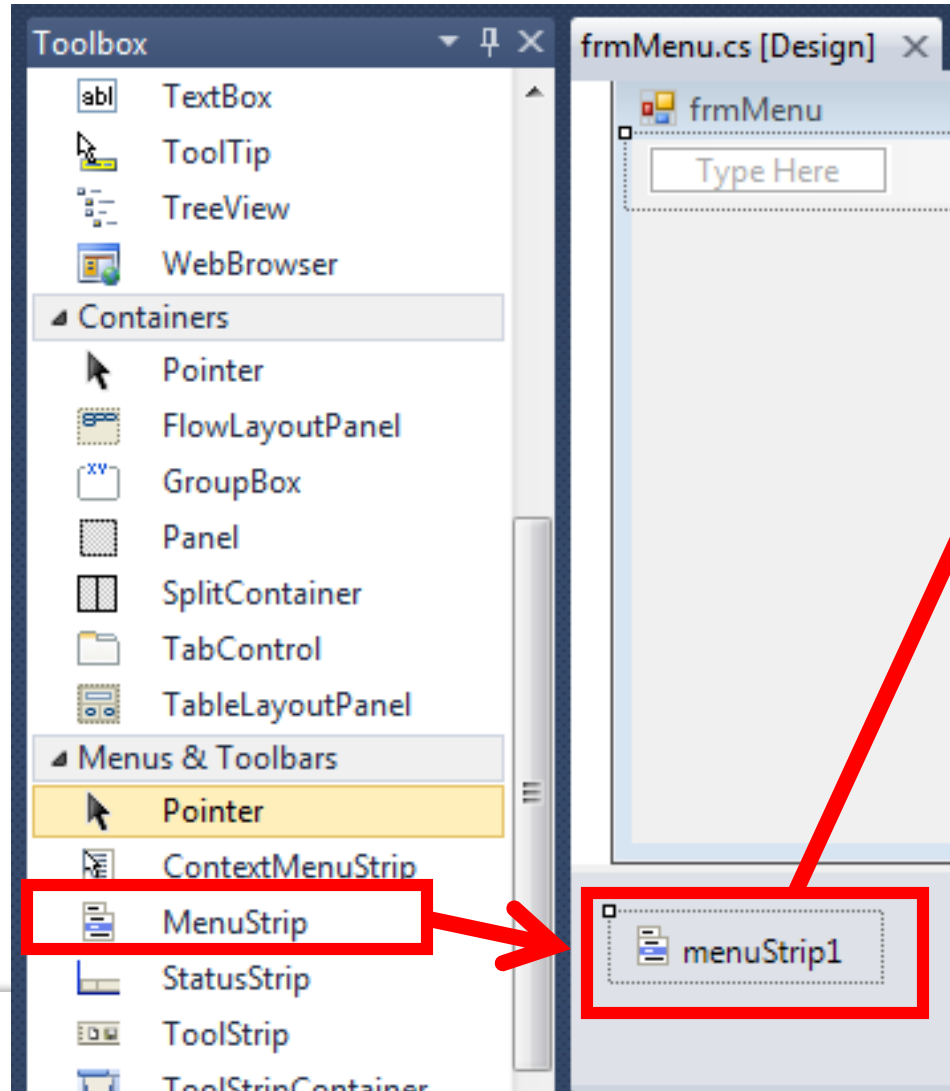


```
private void process_MenuClick
(object sender, EventArgs e)
{
    if (sender.Equals(menuFileExit))
    {
        Application.Exit();
    }
}
```

```
private void frmMenuBasic_Load
(object sender, EventArgs e)
{createMenu();
}
```



Designer



**Click on button
to add MenuItem**

Click Add button to add new MenuItem

Select item and add to list below:

MenuItem

Add

Members:

menuStrip1

menuFile

menuEdit

Text to Display

Click here to add
Sub MenuItem

MenuItem's Name

ToolStripMenuItem

menuFile



RightToLeft

No

RightToLeftAutoMirrorIm: False

ShortcutKeyDisplayString

Text

File

TextAlign

MiddleCenter

TextDirection

Horizontal

TextImageRelation

ImageBeforeText

Behavior

Data

(ApplicationSettings)

DropDown

(none)

DropDownItems

(Collection)

Tag

Design

(Name)

menuFile

GenerateMember

True

OK

Cancel

Select item and add to list below:

MenuItem Add

Members:

- ToolStripDropDownMenu
 - menuFileNew
 - menuFileOpen
 - toolStripMenuItem3
 - menuFileExit

Image Icon

Text to Display

Click here to add
Sub MenuItem

MenuItem's Name

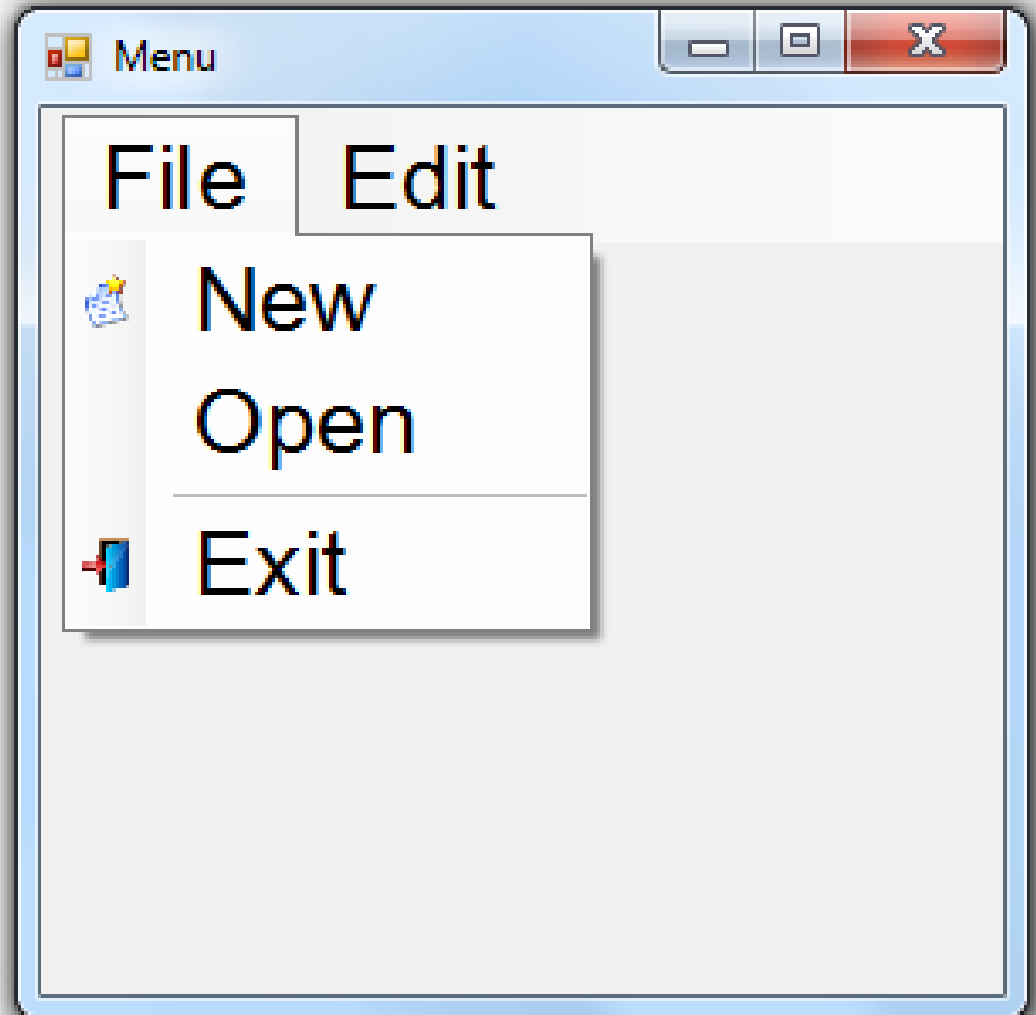
ToolStripMenuItem menuFileNew

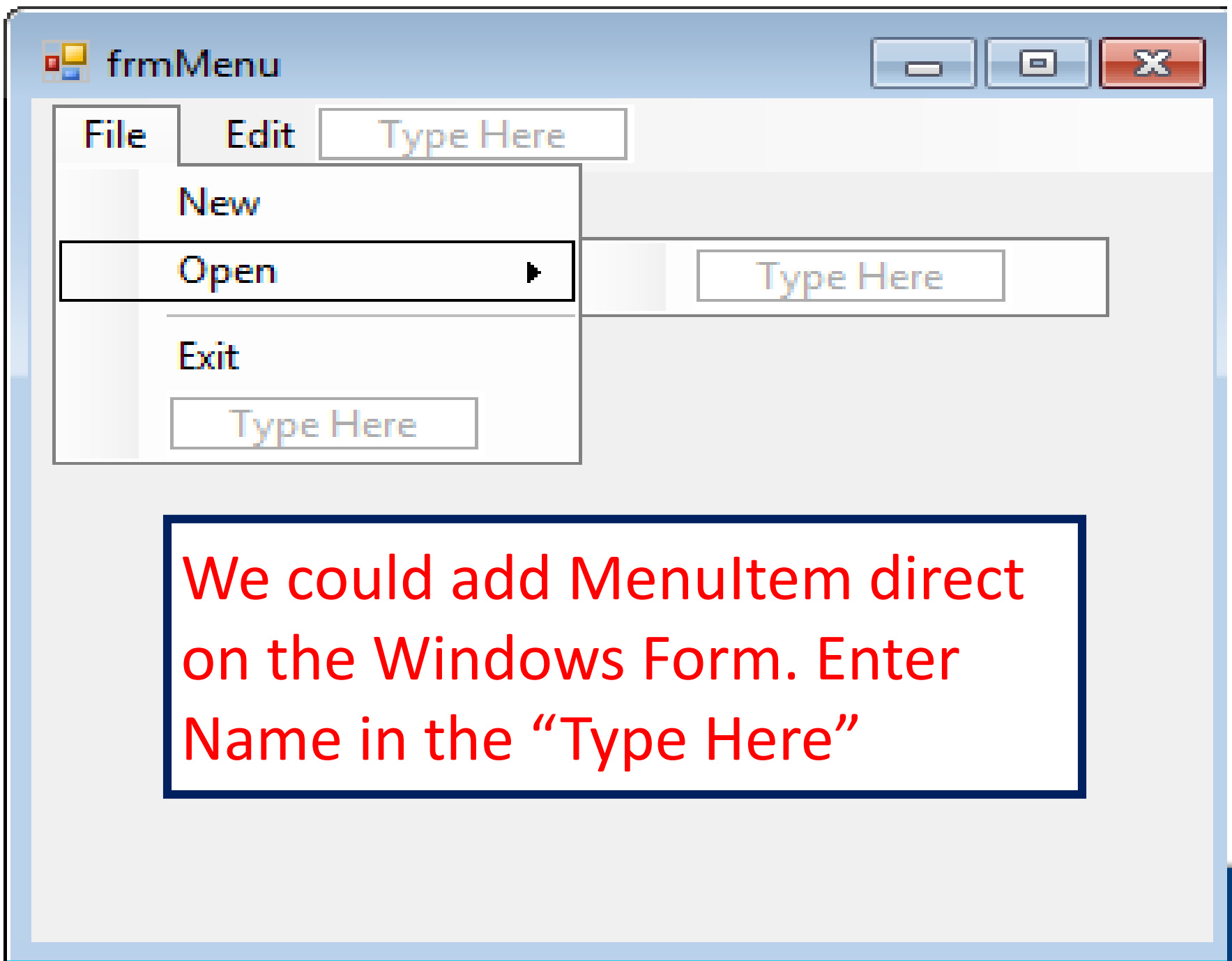
Image	System.Drawing...
ImageAlign	MiddleCenter
ImageScaling	SizeToFit
ImageTransparentColor	<input type="checkbox"/>
RightToLeft	No
RightToLeftAutoMi	False
ShortcutKeyDisplay	
Text	New
TextAlign	MiddleCenter
TextDirection	Horizontal
TextImageRelation	ImageBeforeText
Behavior	
Data	
(ApplicationSetting	
DropDown	(none)
DropDownItems	(Collection)
Tag	
Design	
(Name)	menuFileNew

OK

Cancel

MenuStrip





We could add MenuItem direct on the Windows Form. Enter Name in the "Type Here"

Add Event
for each
MenuItem

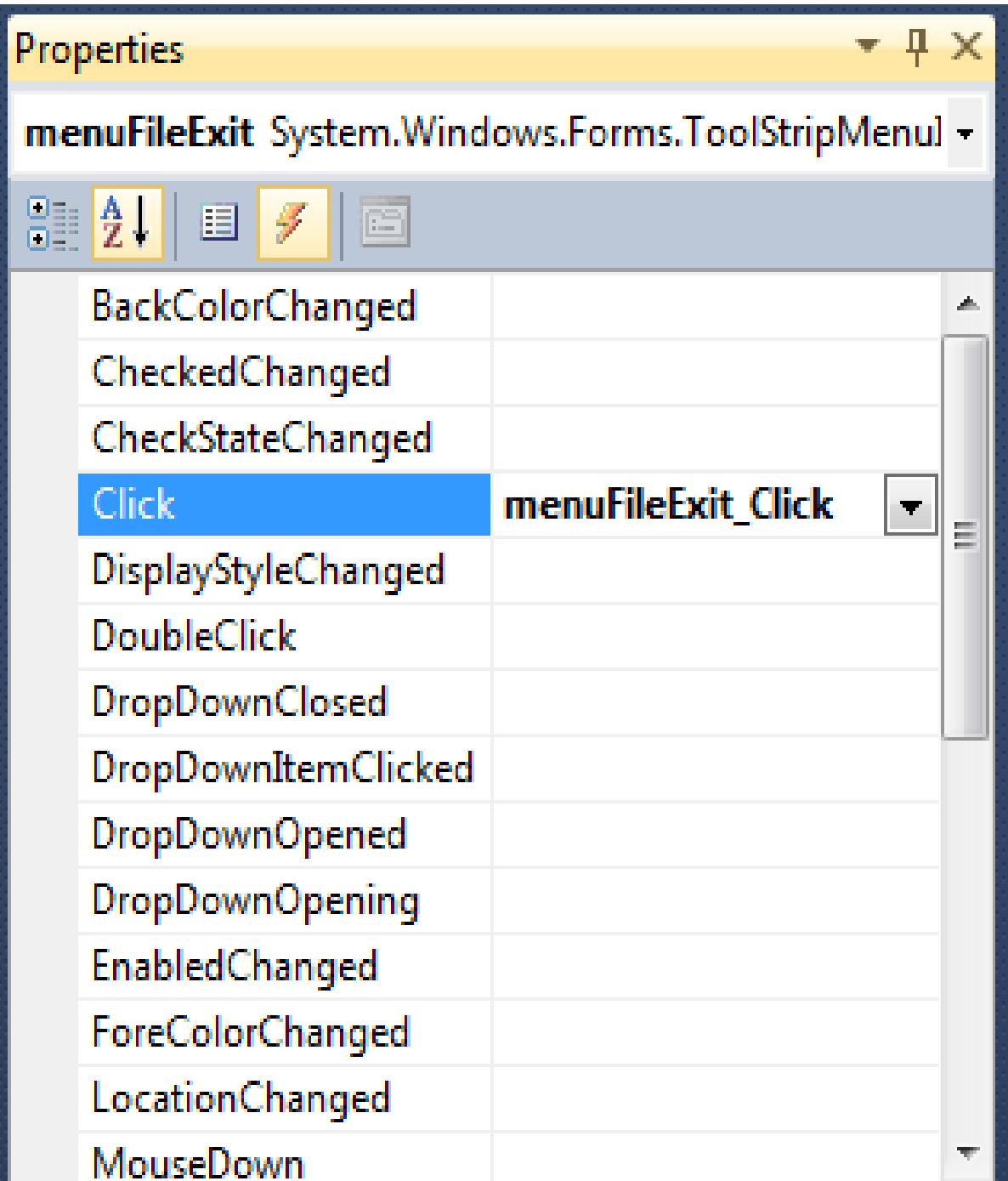


Image List

Toolbox

► Data

▲ Components



Pointer



BackgroundWorker



DirectoryEntry



DirectorySearcher



ErrorProvider



EventLog



FileSystemWatcher



ImageList



MessageQueue



frmMenuStrip

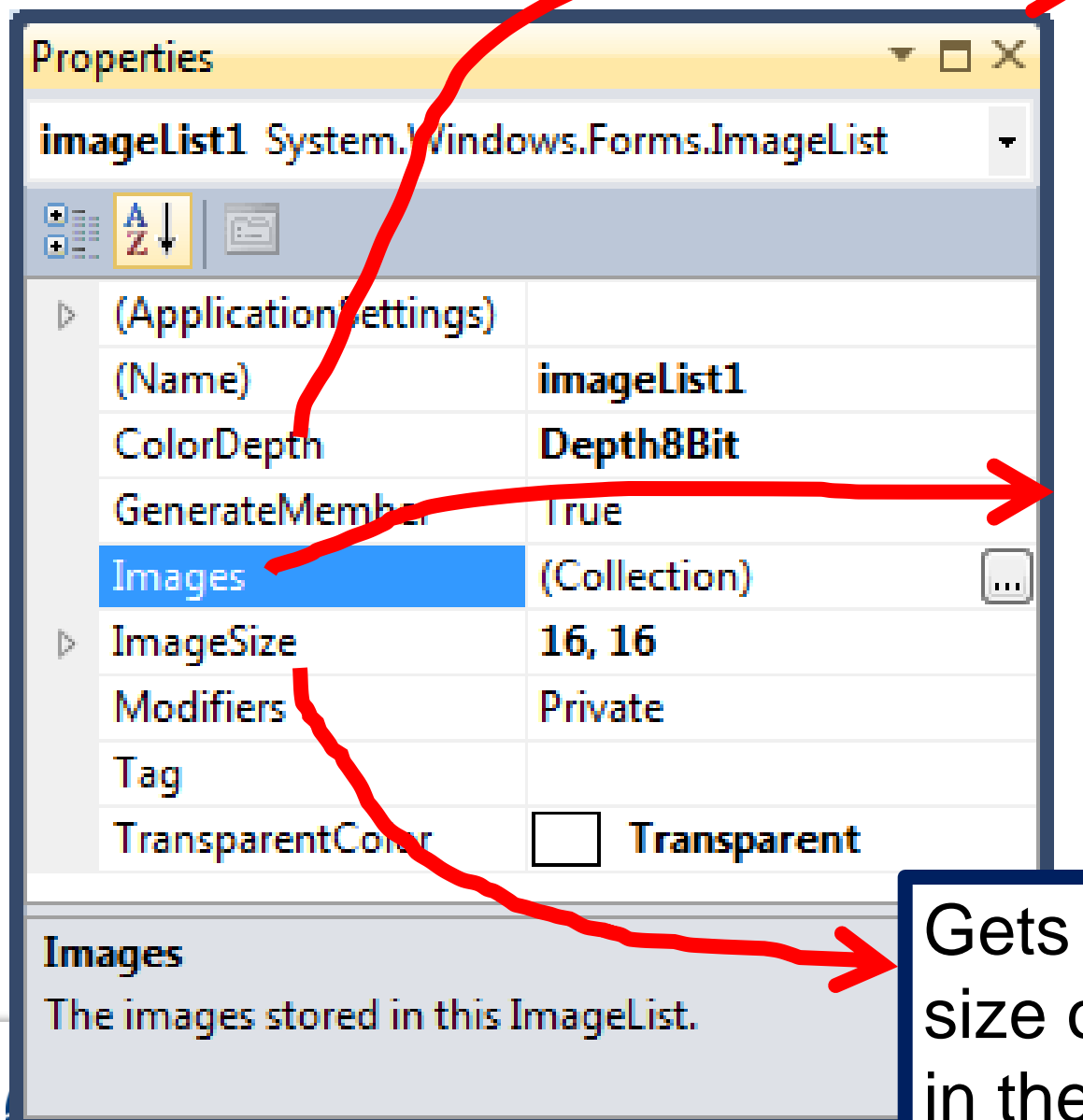


Drag & Drop



imageList1





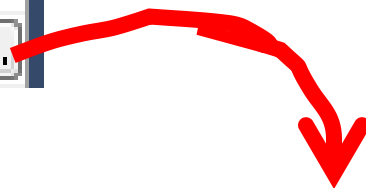
Gets the color depth of the image list

Gets the ImageList. ImageCollection for this image list. See Next Slide

Gets or sets the size of the images in the image list



Images

(Collection)



Images Collection Editor

Members:

0		add.png
1		Exit.png






Click **Add** button
to insert Image
into Collection



Add

Remove

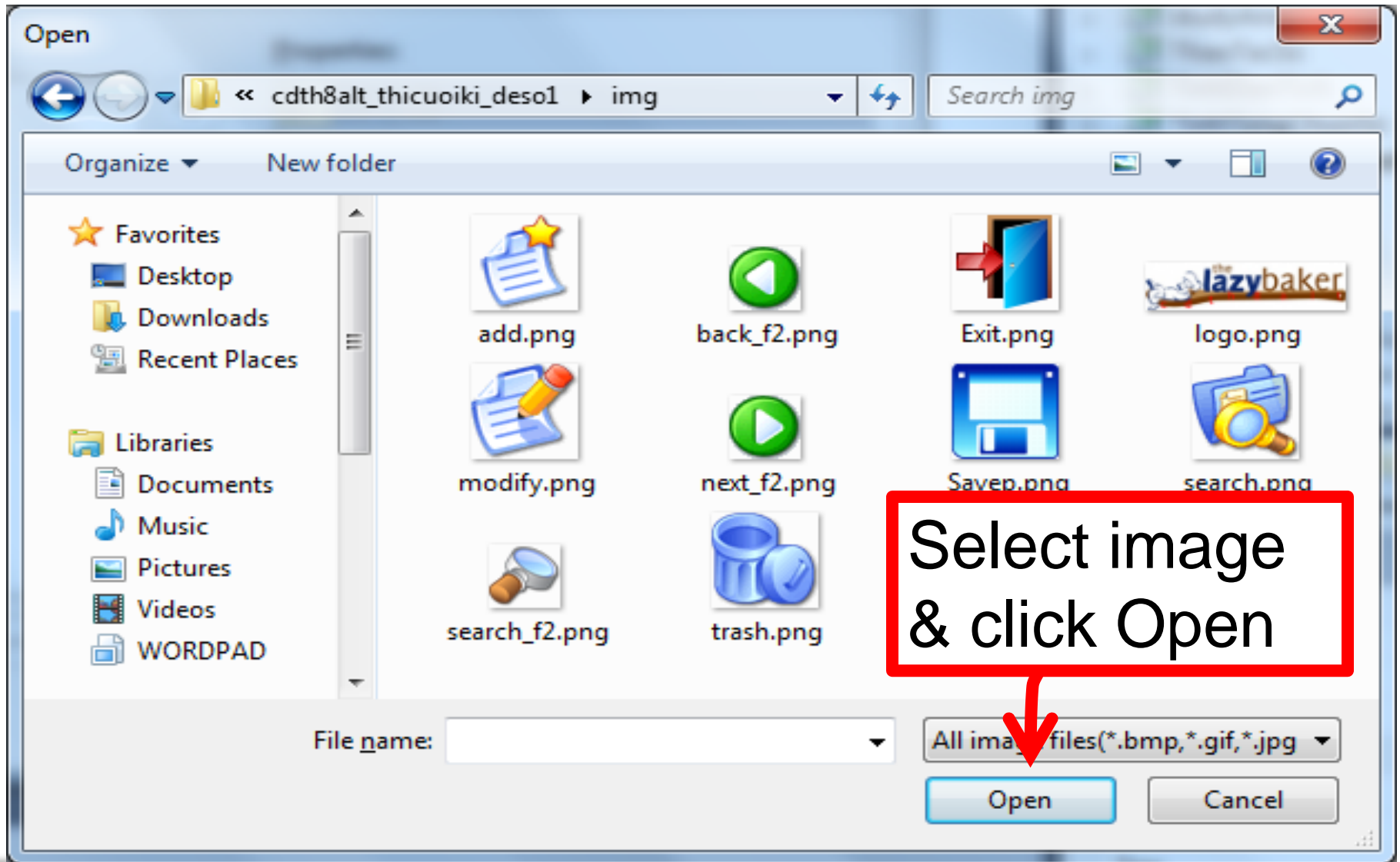
add.png properties:

		
Misc		
HorizontalResol	71.9836	
Name	add.png	
PhysicalDimens	48, 48	
PixelFormat	Format32bppArgb	
RawFormat	Png	
Size	48, 48	
VerticalResolutio	71.9836	

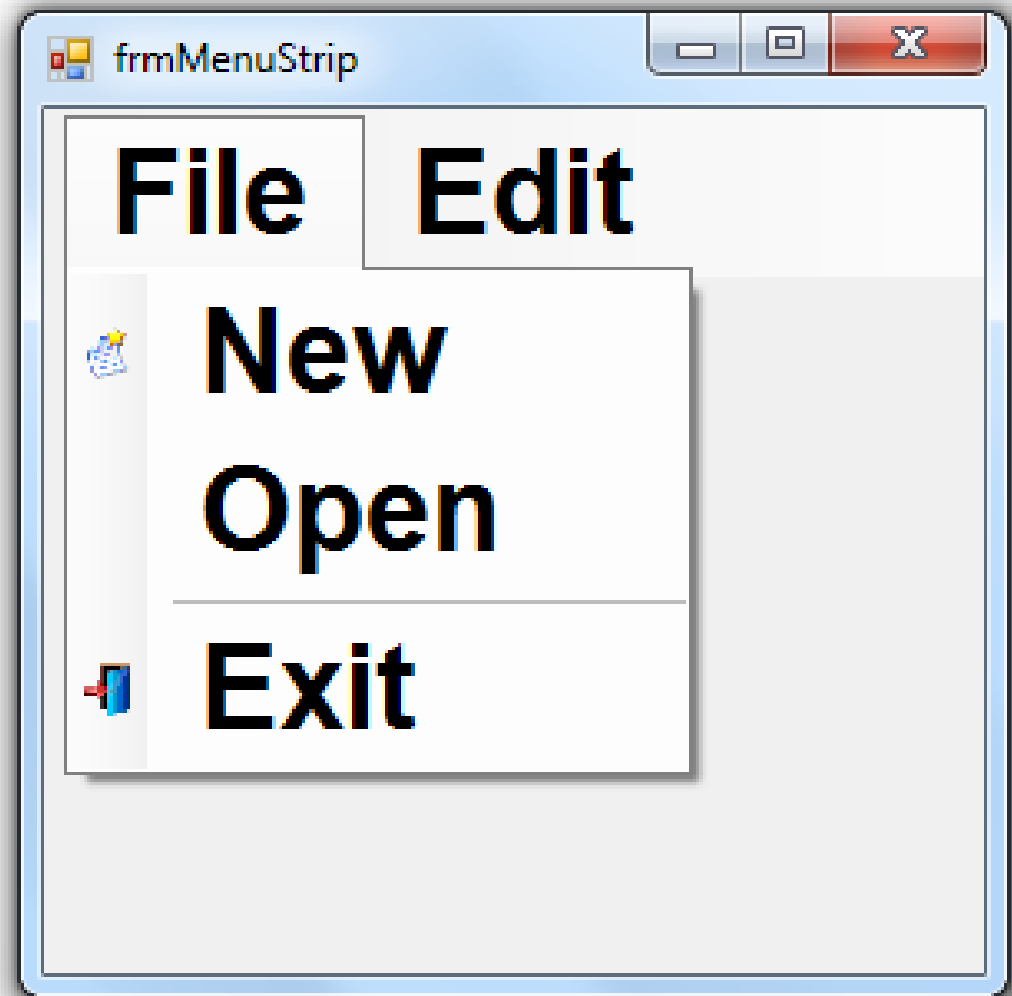
OK

Cancel





MenuStrip *At Runtime*



```
private MenuStrip menuBar;  
private ToolStripMenuItem  
menuFile, menuEdit, menuFileNew,  
menuFileOpen, menuFileExit,  
menuEditCut, menuEditCopy,  
menuEditPaste;
```



```
private void createMenu()  
{  
    menuBar = new MenuStrip();  
    menuBar.Font = new Font("arial", 36,  
        FontStyle.Bold, GraphicsUnit.Pixel);  
    this.MainMenuStrip = menuBar;  
    menuFile = new  
        ToolStripMenuItem("File");  
    menuFileNew = new  
        ToolStripMenuItem("New");  
    menuFileNew.Image =  
        imageList1.Images[0];  
}
```



```
menuFileOpen = new  
ToolStripMenuItem("Open");  
ToolStripSeparator sp = new  
ToolStripSeparator();  
menuFileExit = new  
ToolStripMenuItem("Exit");  
menuFileExit.Image =  
imageList1.Images[1];  
menuFile.DropDownItems.Add(  
menuFileNew);  
menuFile.DropDownItems.Add(  
menuFileOpen);
```




```
menuFile.DropDownItems.Add(sp);  
menuFile.DropDownItems.Add(  
menuFileExit);  
menuEdit = new  
ToolStripMenuItem("Edit");  
menuEditCut = new  
ToolStripMenuItem("Cut");  
menuEditCopy = new  
ToolStripMenuItem("Copy");  
menuEditPaste = new  
ToolStripMenuItem("Paste");
```



```
menuEdit.DropDownItems.AddRange(new  
ToolStripItem[] {  
menuEditCut,menuEditCopy,  
menuEditPaste});  
menuBar.Items.AddRange(new  
ToolStripItem[] {  
menuFile,menuEdit});  
this.Controls.Add(menuBar);  
attachEvents();  
}  
  
private void attachEvents()  
{menuFileExit.Click += processClick; }
```

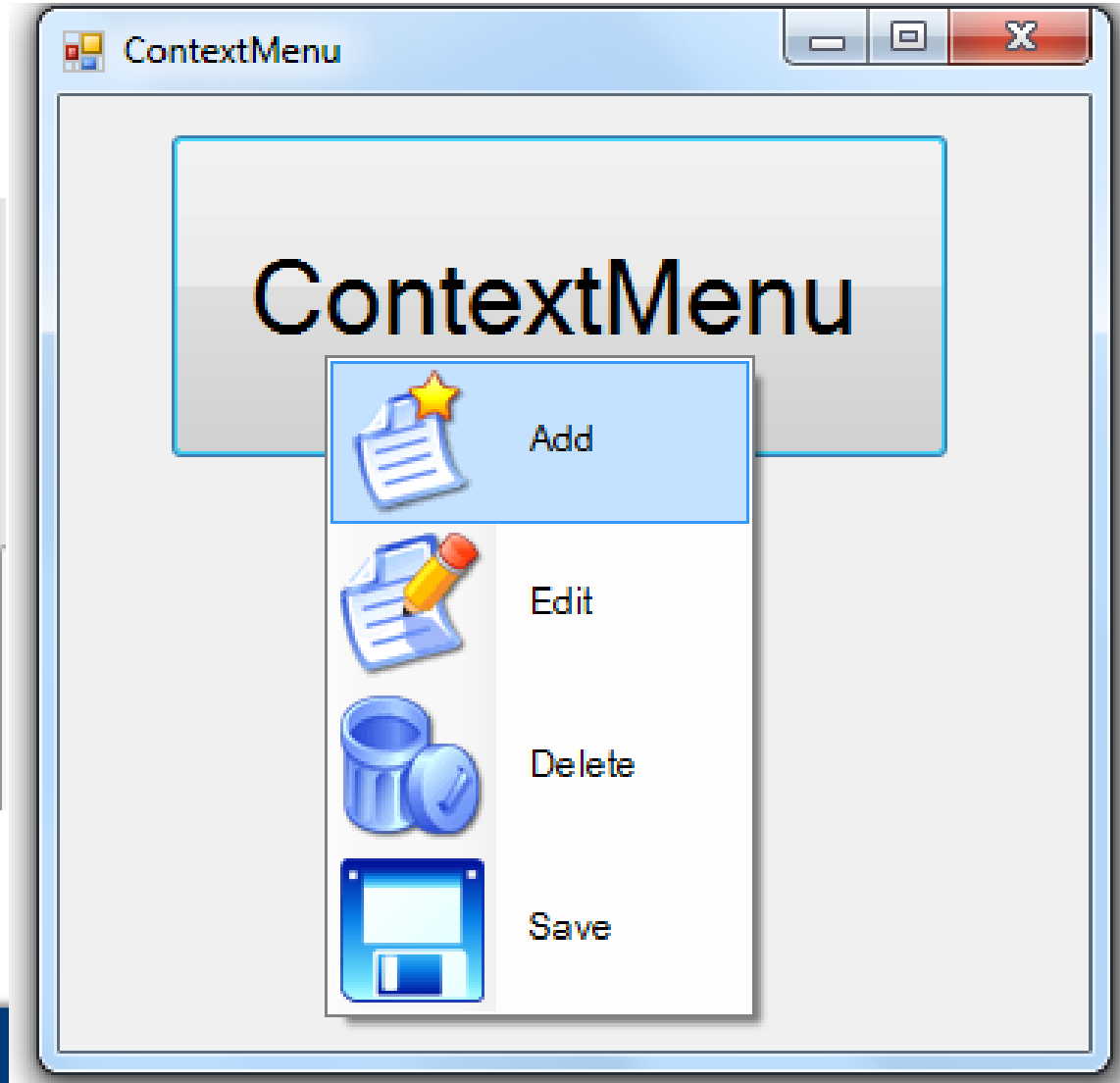
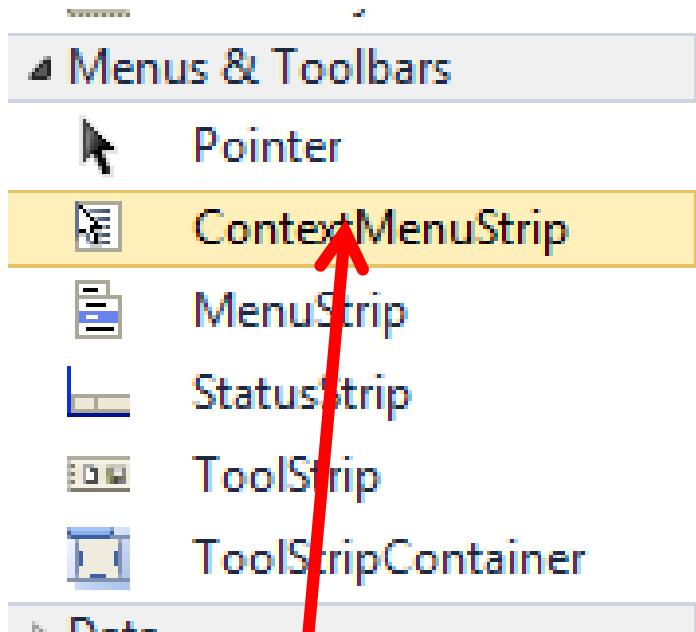


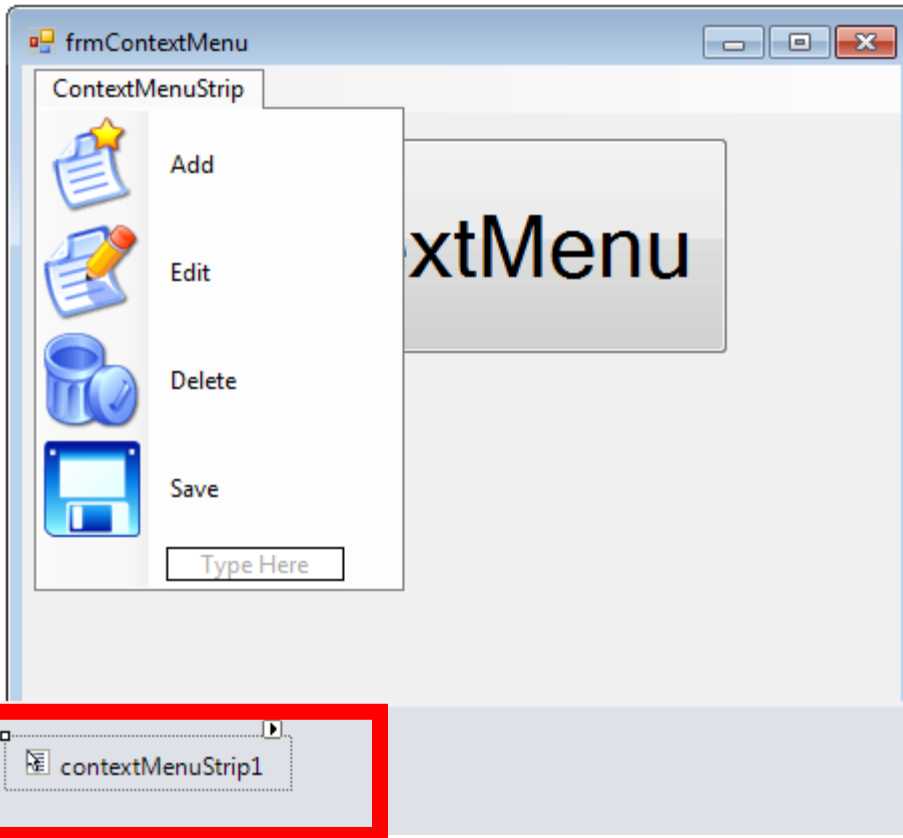
```
private void processClick
(object o, EventArgs e)
{
    if (o.Equals(menuFileExit))
        Application.Exit();
}

private void frmMenuStrip_Load
(object sender, EventArgs e)
{
    createMenu();
}
```



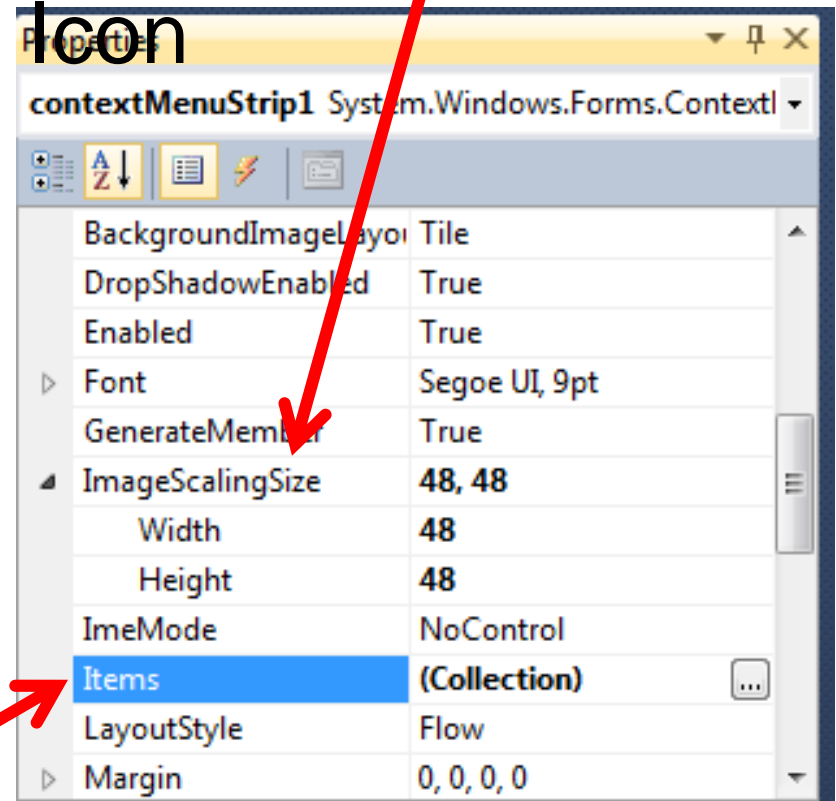
ContextMenuStrip





ImageScalingSize to
set Width, Height

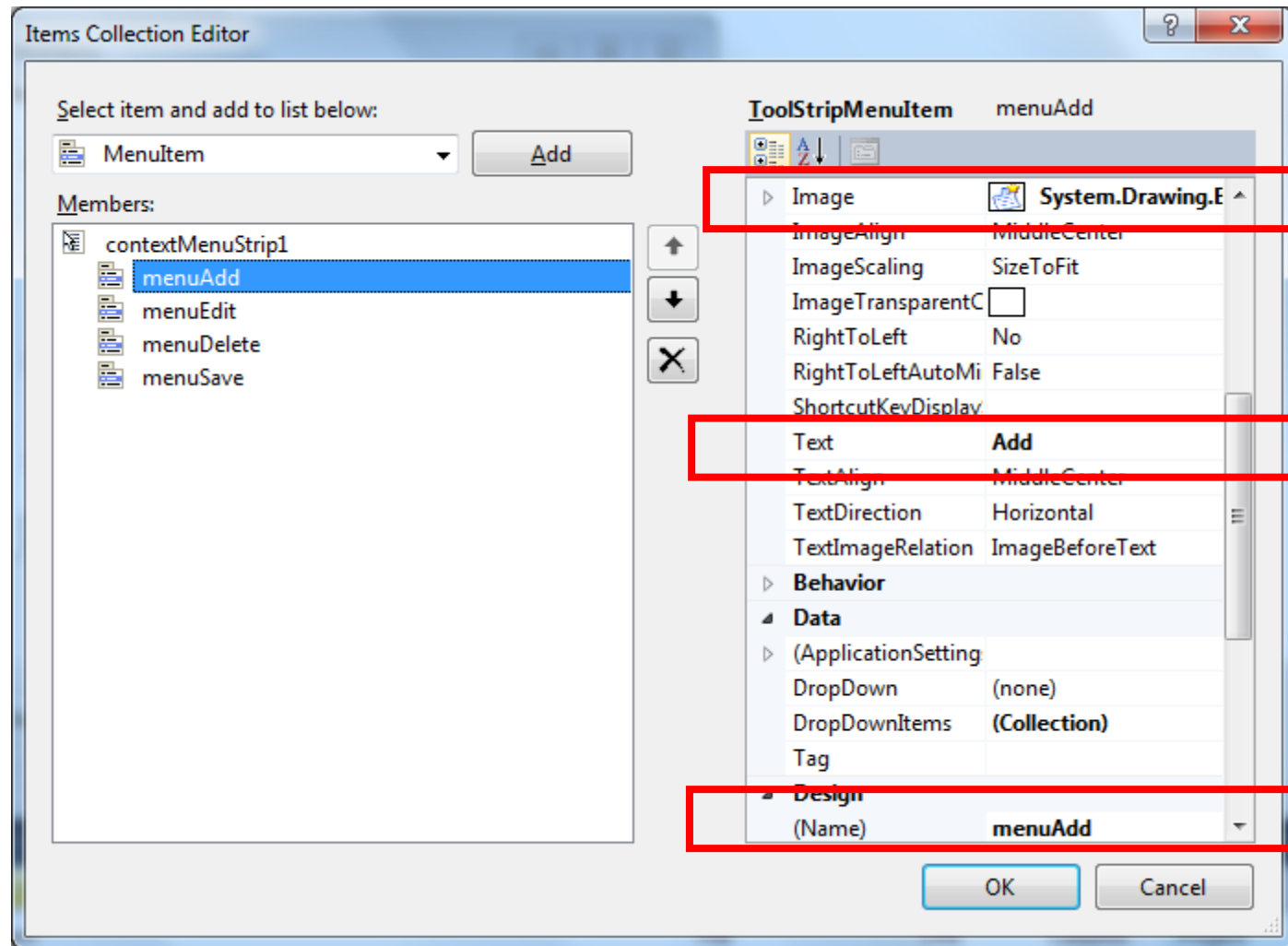
Icon

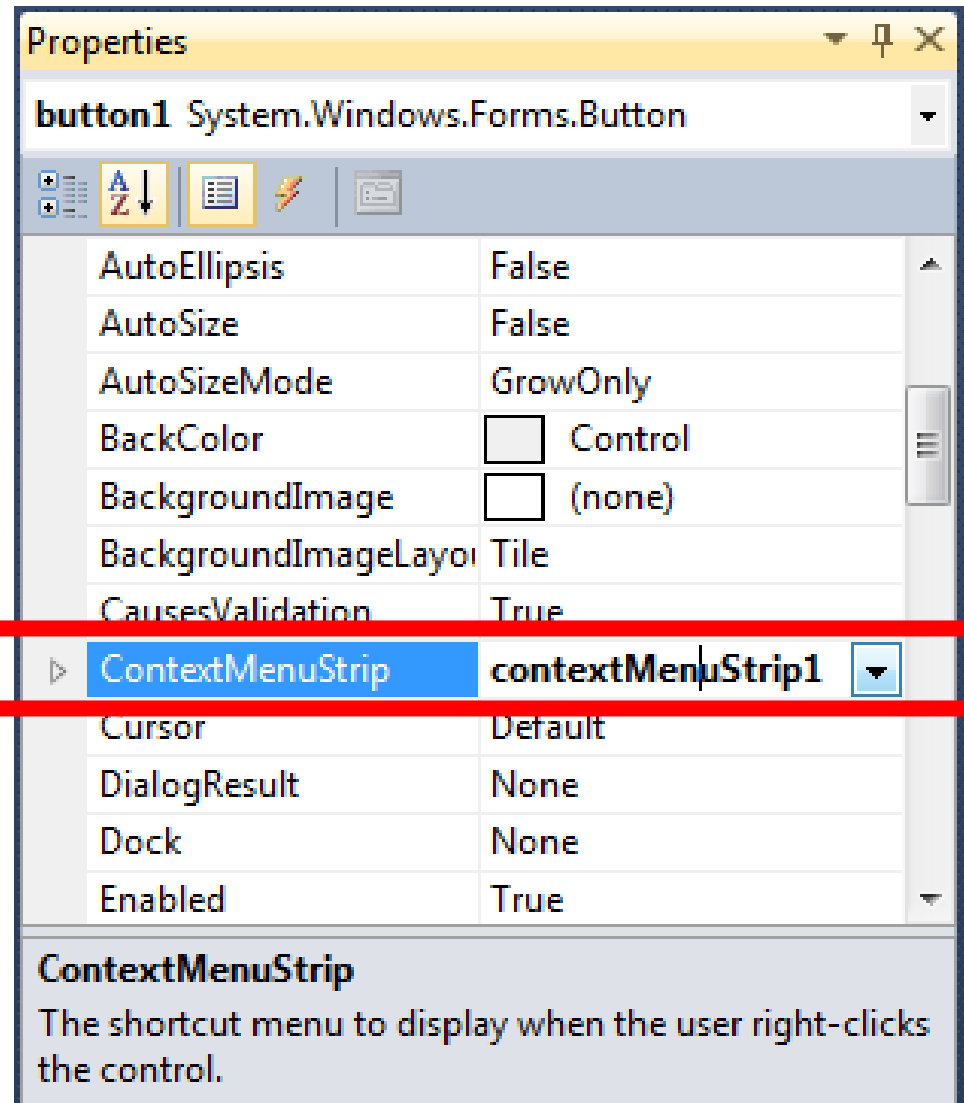
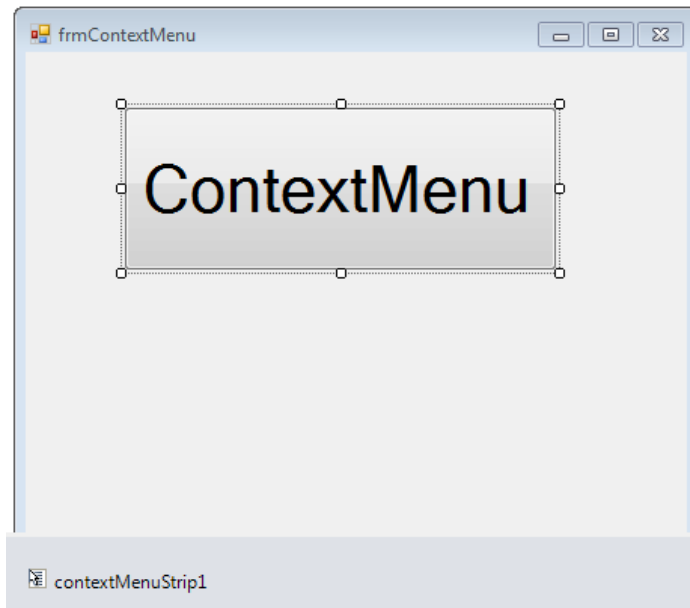


Click Items (Collection)
to add MenuItem



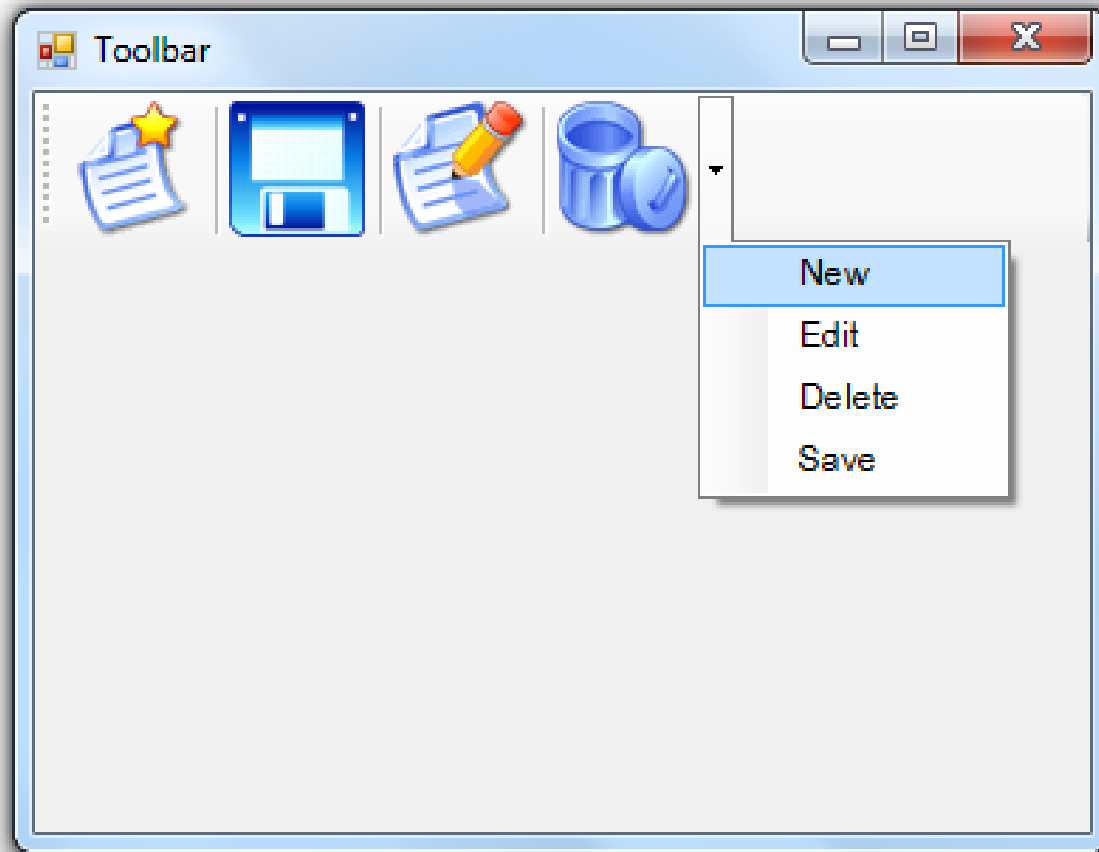
As the same
MenuStrip



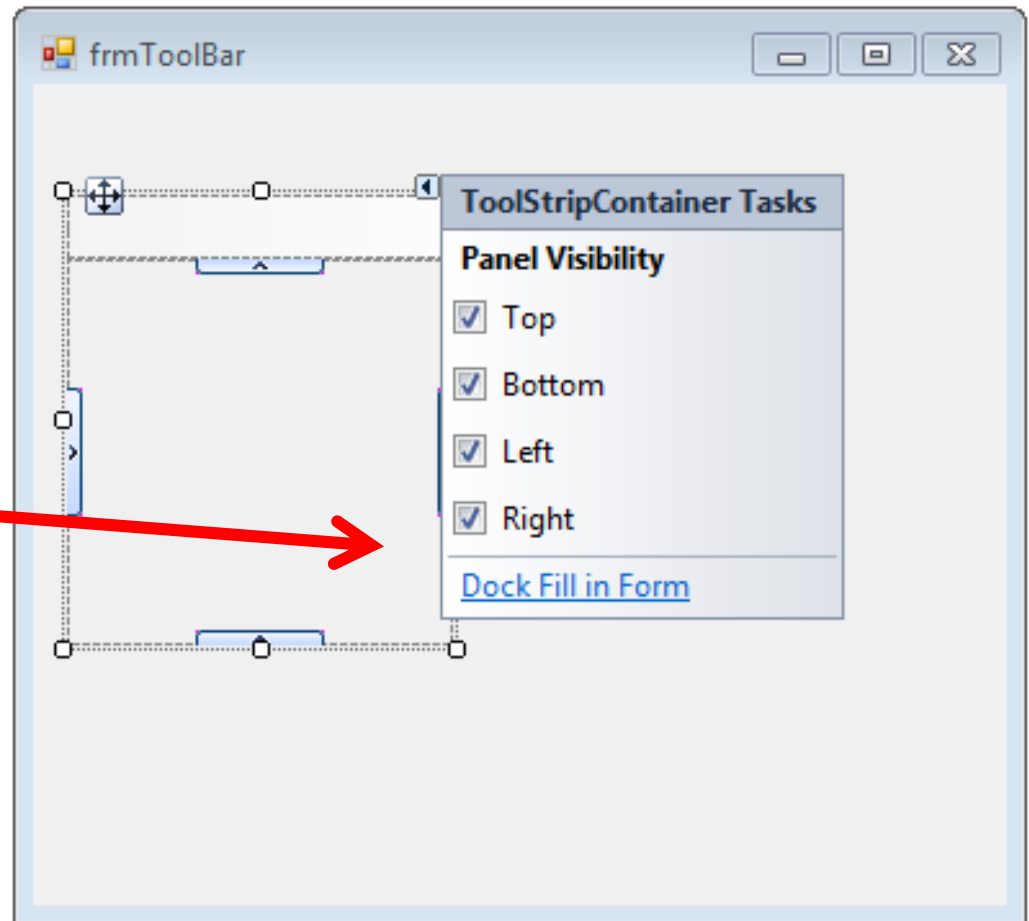
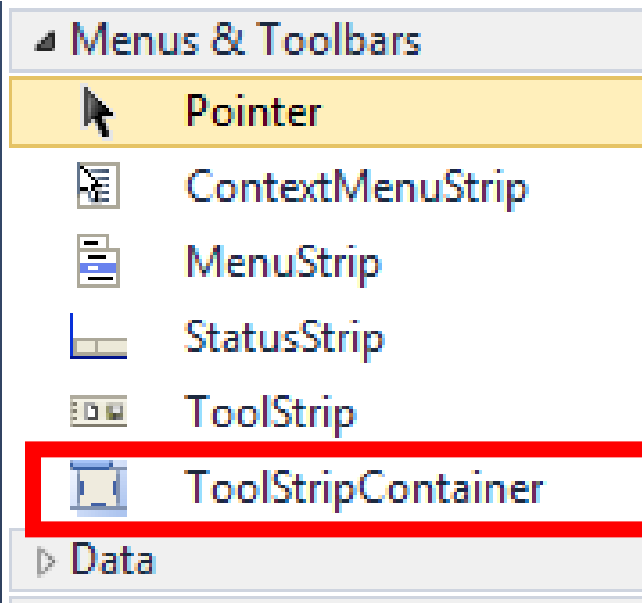


1. Choose button
2. Set **ContextMenuStrip**
3. Attach event as the same **MenuStrip**

ToolStripContainer & ToolStrip

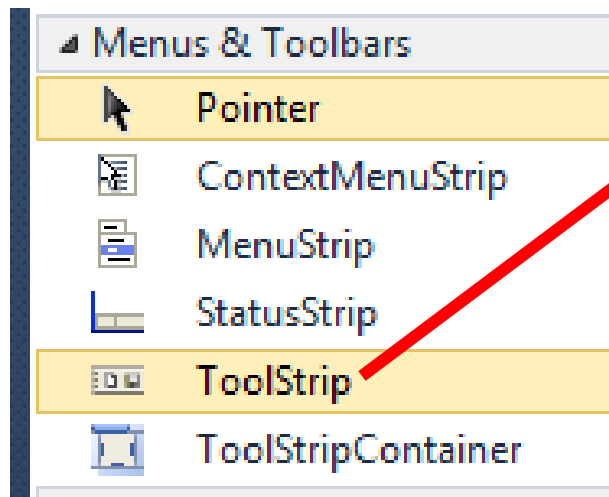


ToolStripContainer

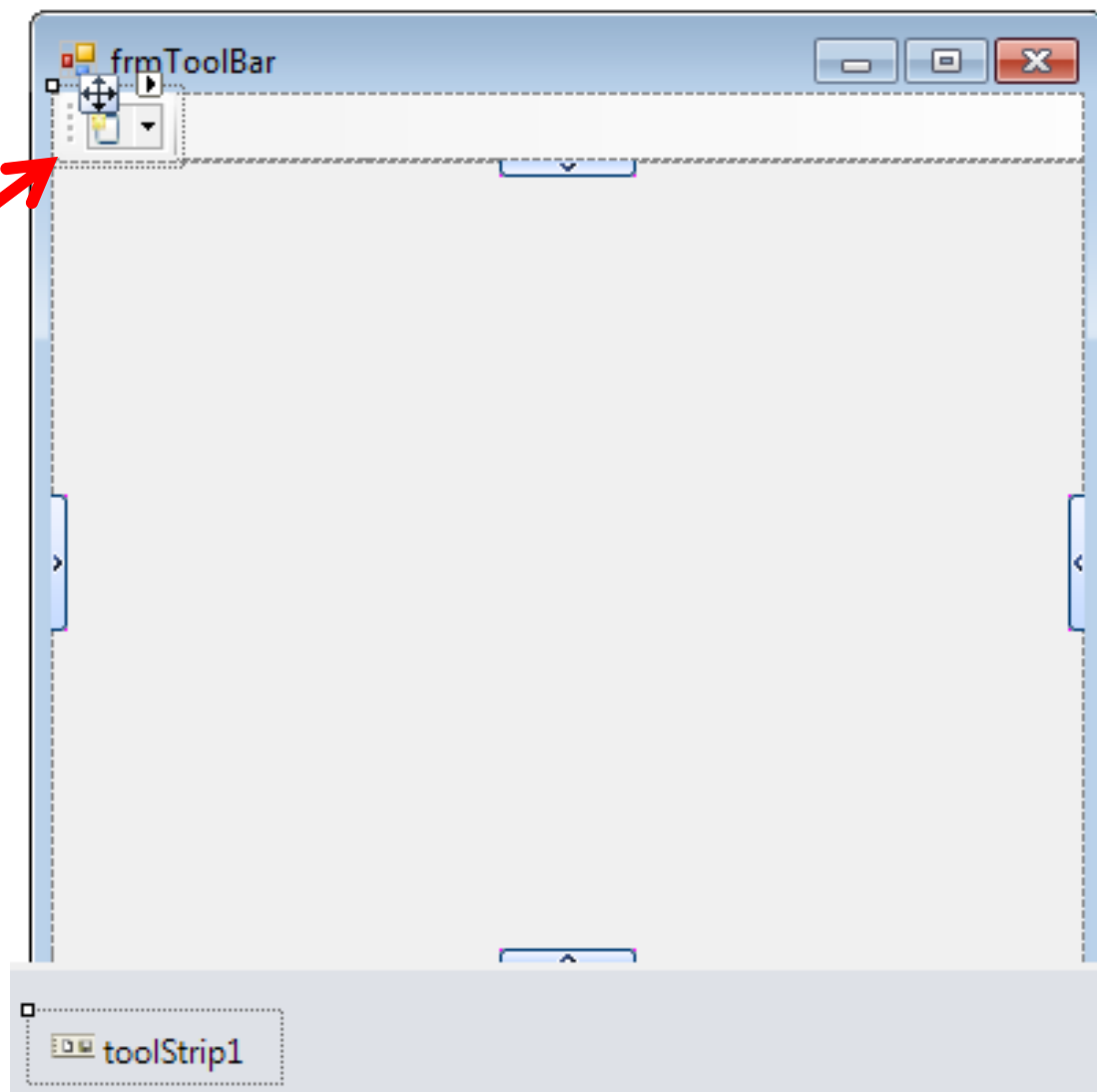


ToolStripContainer Provides panels on each side of the form and a central panel that can hold one or more controls





Drag & Drop
ToolStrip into
the Form



As the
same
MenuStrip

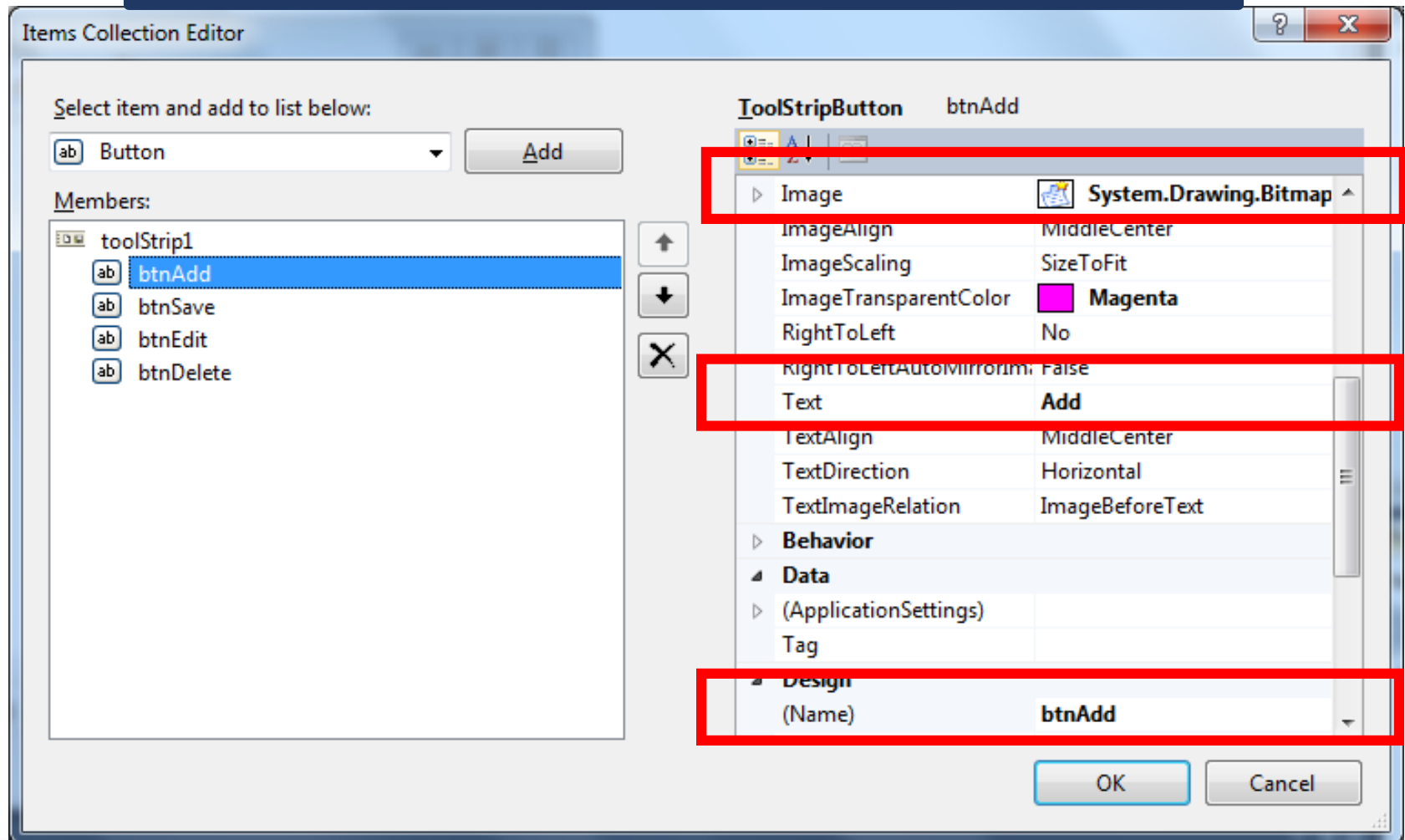
Properties

toolStrip1 System.Windows.Forms.ToolStrip

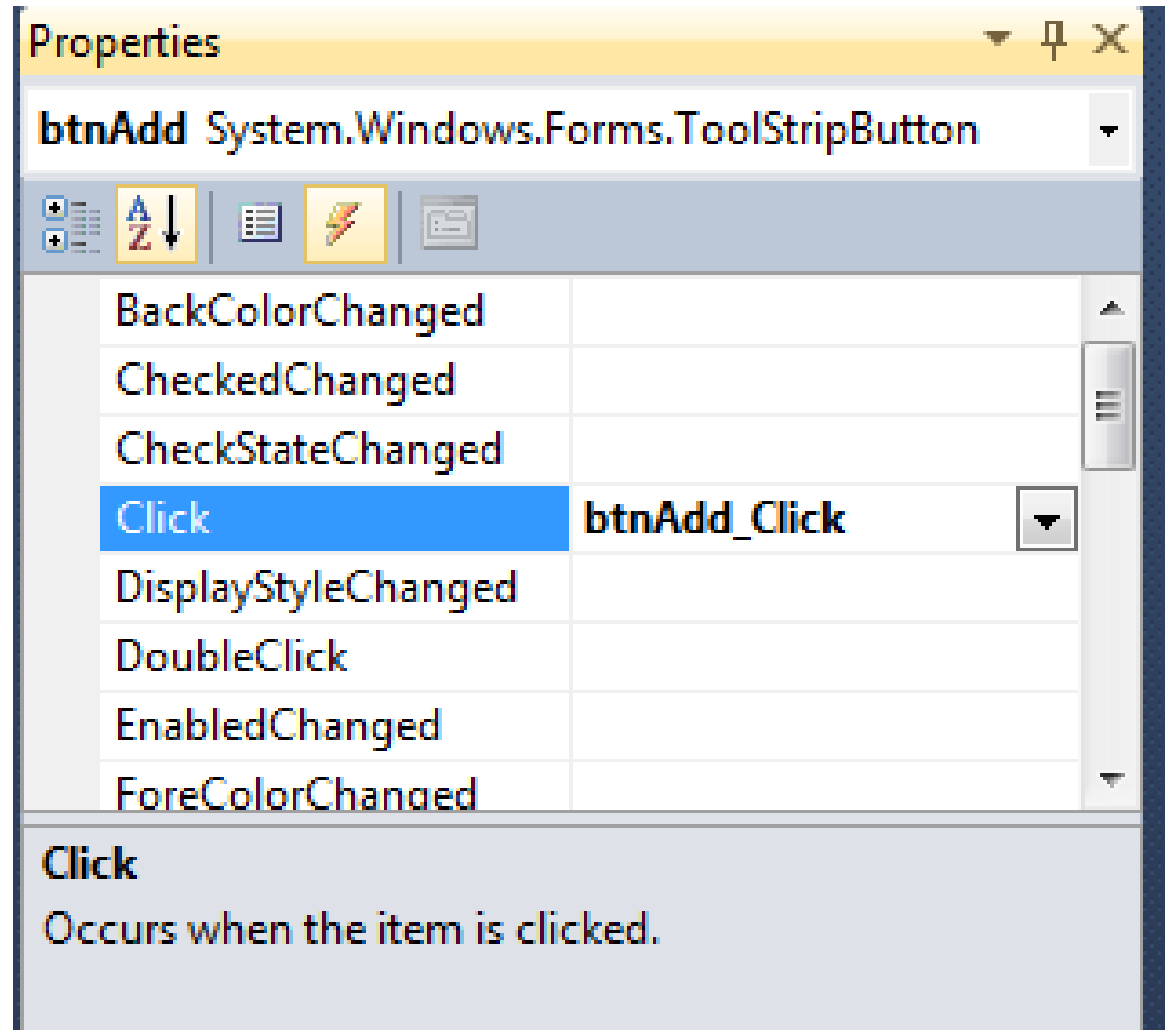
Icons: Add, Remove, Undo, Redo, Copy, Paste, Find, Help

GripStyle	Visible
ImageScalingSize	16, 16
Width	16
Height	16
ImeMode	NoControl
Items	(Collection)
LayoutStyle	HorizontalStackWithOverf
Location	3, 0

Add Items Collection for ToolBar



Attach Events for each Item



| Bài tập áp dụng

16/09/2024

Tổng quan về lập trình ứng dụng

103



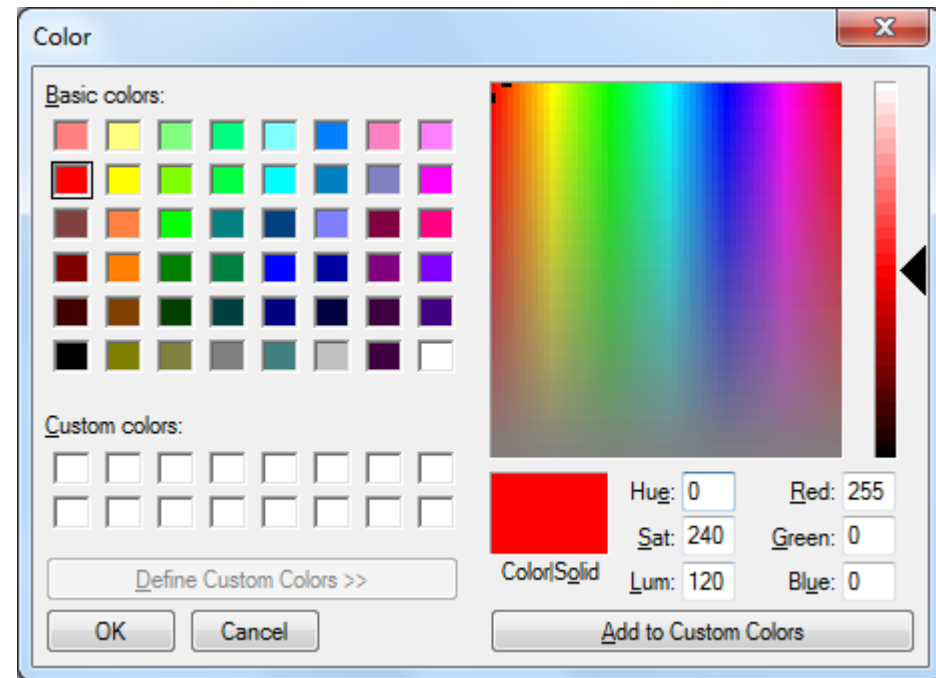
FACULTY OF INFORMATION
TECHNOLOGY
THU DUC COLLEGE OF
TECHNOLOGY



ColorDialog



btnColor



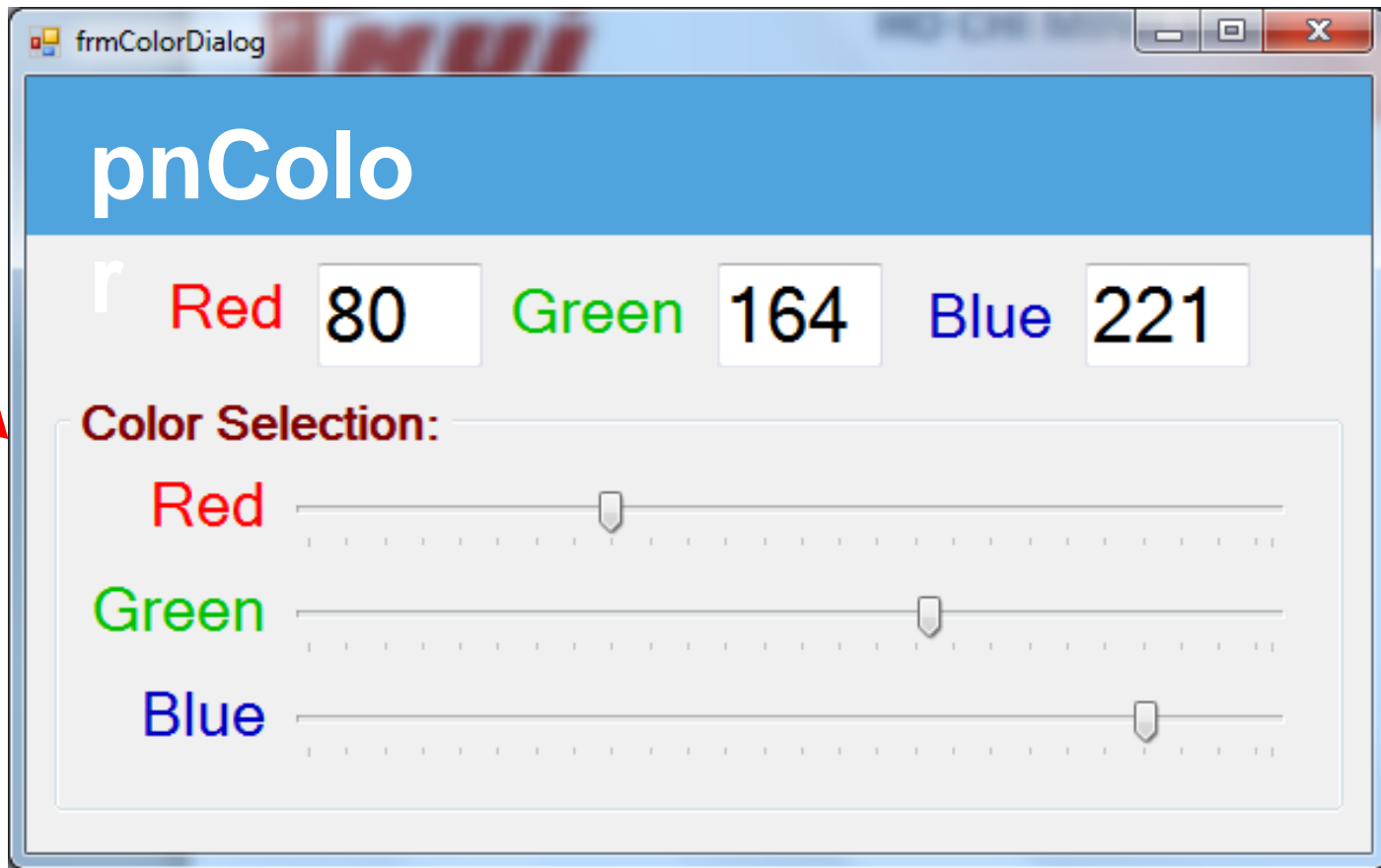
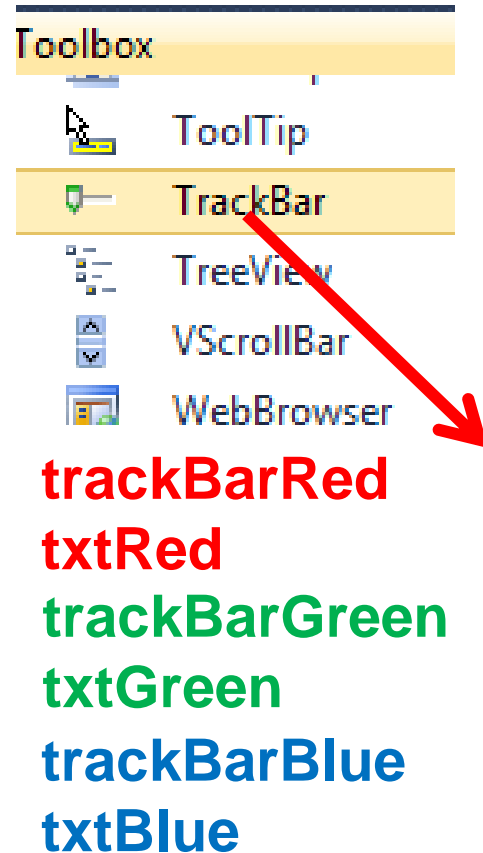
**How to use
ColorDialog?**


```
private void btnColor_Click
(object sender, EventArgs e)
{
    ColorDialog cldlg= new ColorDialog();
    cldlg.Color = pnColor.BackColor;

    if(cldlg.ShowDialog()==DialogResult.OK)
        pnColor.BackColor = cldlg.Color;
}
```



If you want to set color by the **RED**, **GREEN**, **BLUE**. Please see figure below:



For each
trackbar,
please set
properties as
the same
figure

Properties

trackBarRed System.Windows.Forms.TrackBar

Maximum	254
MaximumSize	0, 0
Minimum	0
MinimumSize	0, 0
Modifiers	Private
Orientation	Horizontal
RightToLeft	No
RightToLeftLayout	False
Size	441, 45
SmallChange	1
TabIndex	1
TabStop	True
Tag	
TickFrequency	10
TickStyle	BottomRight

Maximum
The maximum value for the position of the slider on the TrackBar.



```
private void setColor(){  
    int nRed = trackBarRed.Value;  
    int nGreen = trackBarGreen.Value;  
    int nBlue = trackBarBlue.Value;  
    txtRed.Text = nRed + "";  
    txtBlue.Text = nBlue + "";  
    txtGreen.Text = nGreen + "";  
    pnColor.BackColor = Color.FromArgb(nRed,  
    nGreen,nBlue);  
}
```



Create Event or trackbar control

```
private void processTrackBar  
(object sender, EventArgs e)  
{  
    setColor();  
}
```



```
private void frmColorDialog_Load
(object sender, EventArgs e)
{
    trackBarBlue.Scroll += processTrackBar;
    trackBarGreen.Scroll += processTrackBar;
    trackBarRed.Scroll += processTrackBar;
    setColor();
}
```



Name	Description
Maximum	The Maximum property sets the value of the track bar when the slider is all the way to the right
Minimum	Gets or sets the lower limit of the range this TrackBar is working with.
Value	Gets or sets a numeric value that represents the current position of the scroll box on the track bar



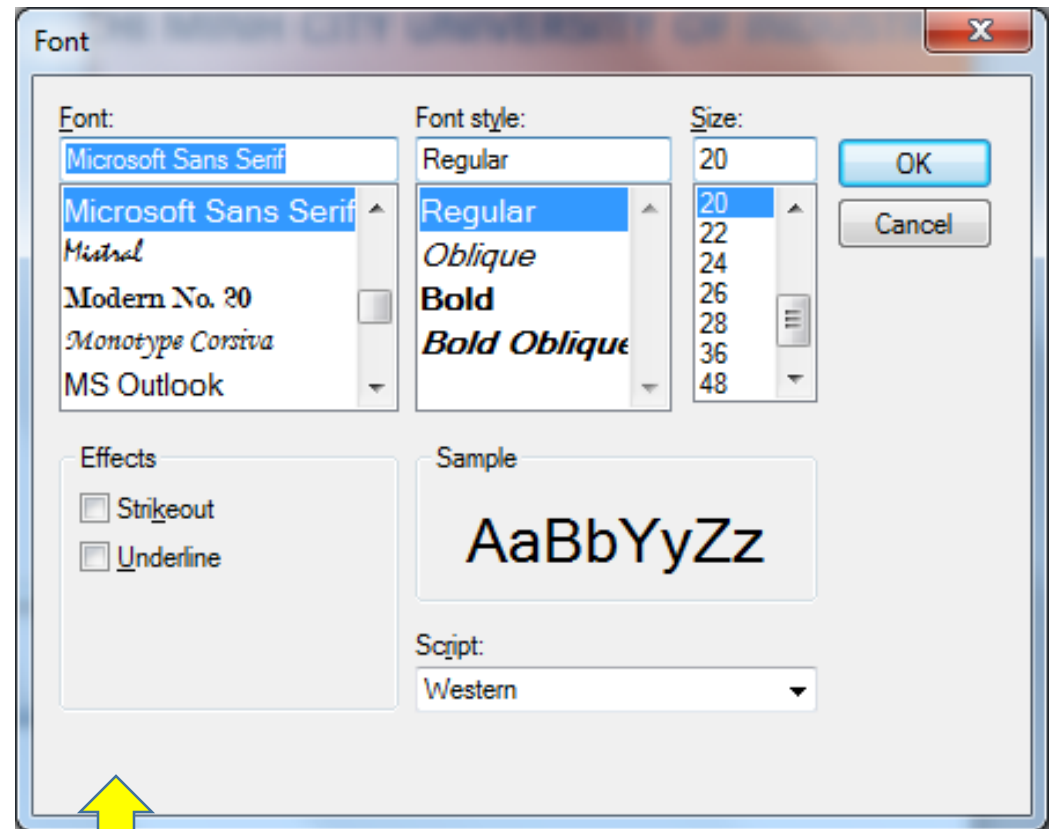
Name	Description
LargeChange	Sets how many positions to move if the bar is clicked on either side of the slider
SmallChange	sets how many positions to move if the keyboard arrows are used to move the slider
TickFrequency	Establishes how many positions are between each tick-mark



FontDialog



btnFont



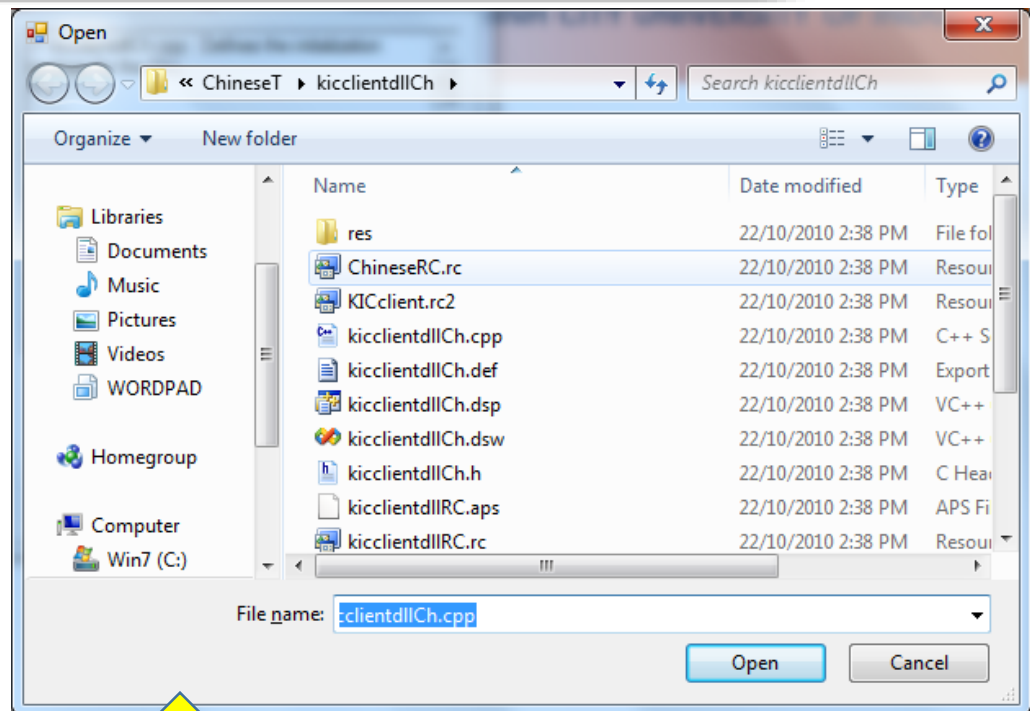
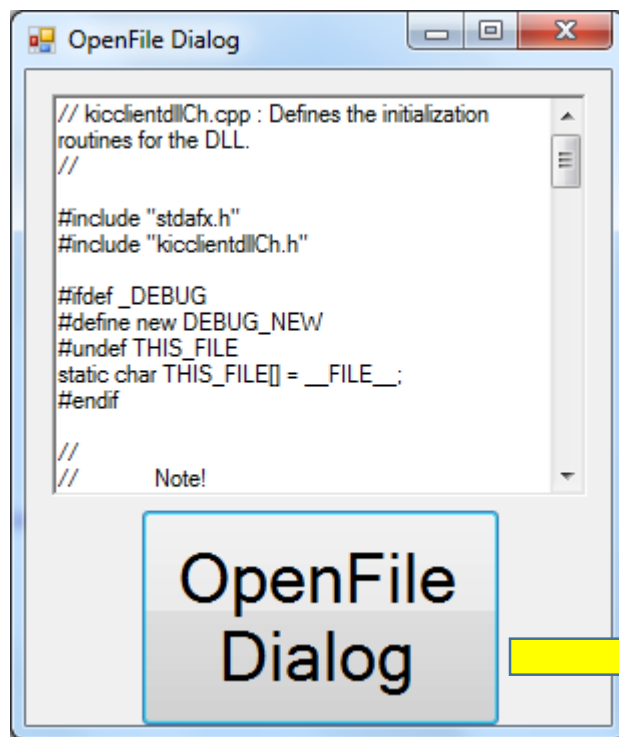
How to use
FontDialog?



```
private void btnFont_Click
(object sender, EventArgs e)
{
    FontDialog ftDialog = new
        FontDialog();
    ftDialog.Font = lblFont.Font;
    if (ftDialog.ShowDialog() ==
        DialogResult.OK)
    {
        lblFont.Font = ftDialog.Font;
    }
}
```



RichTextBox & StreamFile

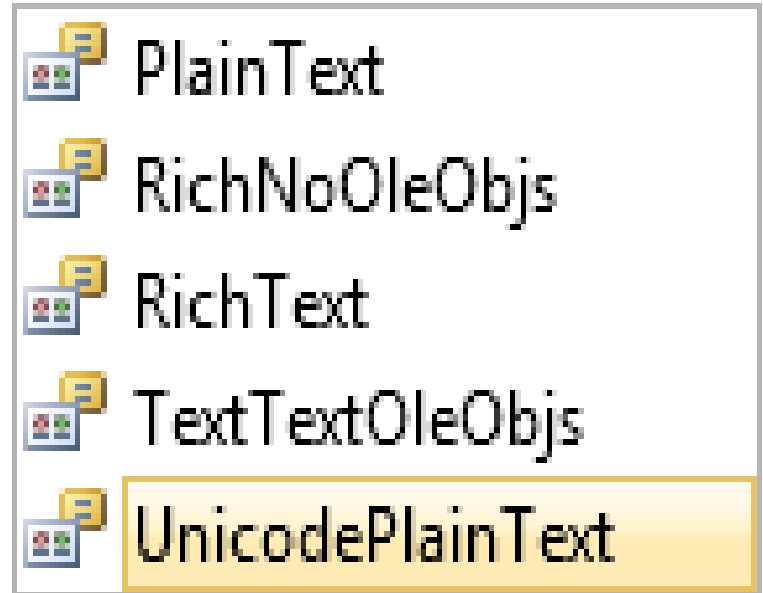


```
OpenFileDialog openFileDialog = new  
OpenFileDialog();  
openDlg.Filter = "(*.txt)|*.txt|(All)|*.*";  
if (openDlg.ShowDialog() ==  
DialogResult.OK)  
{  
    rtFile.LoadFile(openDlg.FileName,  
RichTextBoxStreamType.PlainText);  
}
```



```
richTxtFile.LoadFile(openDlg.FileName,  
    RichTextBoxStreamType.);
```

You could choose
StreamType



But We could use StreamFile
to read Data.

You should Know this

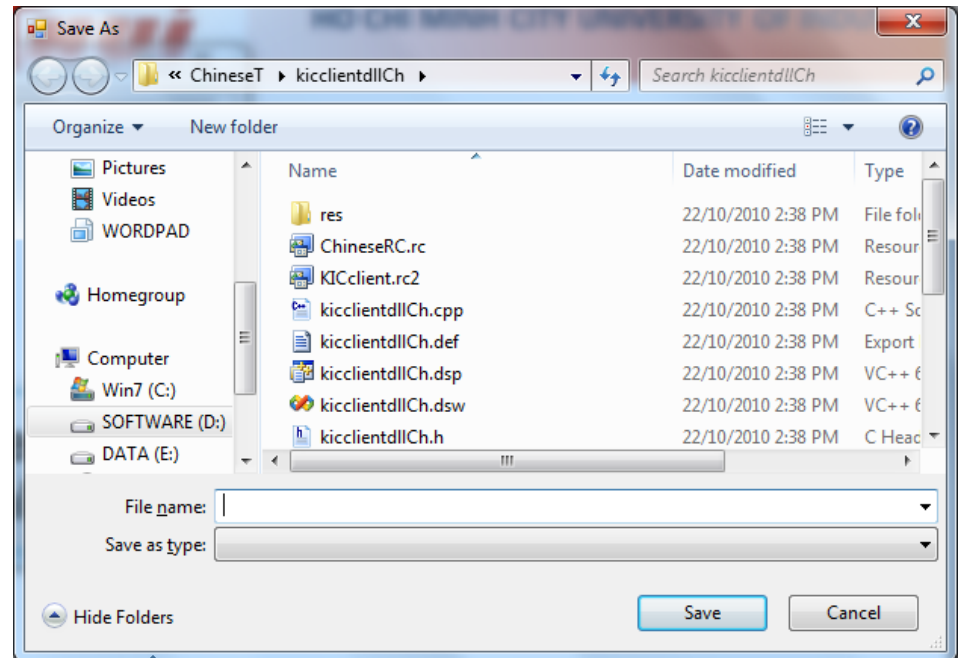
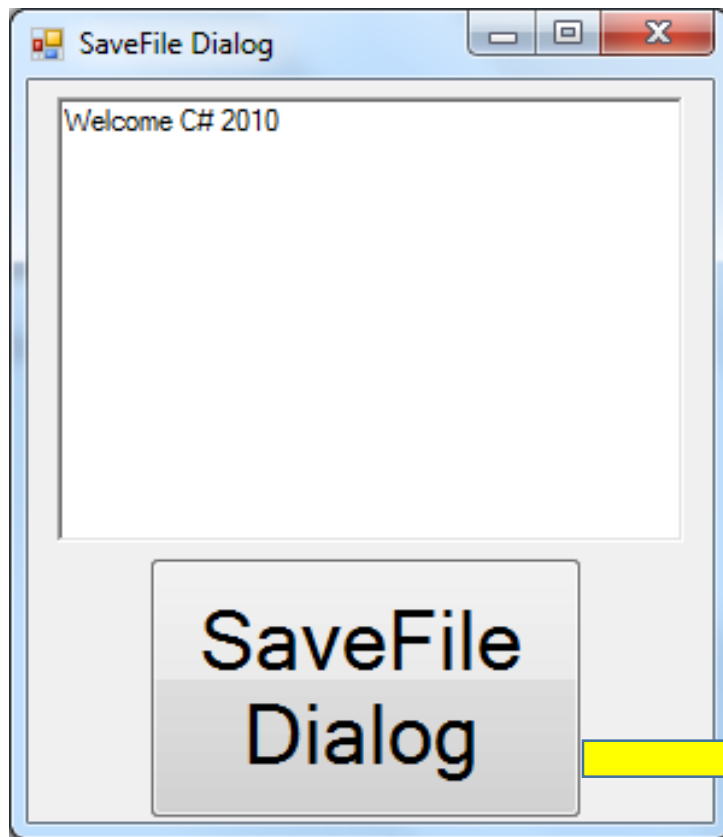


Add this command at the top File
using System.IO;

```
if (openDlg.ShowDialog() ==  
DialogResult.OK)  
{  
    //Open to read  
    Stream stream = openDlg.OpenFile();  
    StreamReader reader = new  
    StreamReader(stream);  
    richTxtFile.Text = reader.ReadToEnd();  
    reader.Close();  
}
```



RichTextBox & StreamFile




```
SaveFileDialog saveDlg = new
SaveFileDialog();
saveDlg.Filter = "(*.txt)|*.txt|(All)|*.*";
if (saveDlg.ShowDialog() ==
DialogResult.OK)
{
rtFile.SaveFile(saveDlg.FileName, RichTextBo
xStreamType.PlainText);
}
```



Use StreamWriter to Write Data.

```
if (saveDlg.ShowDialog() ==  
    DialogResult.OK)  
{  
    Stream stream = saveDlg.OpenFile();//Open  
    to Write  
    StreamWriter writer = new  
    StreamWriter(stream);  
    writer.WriteLine(rtFile.Text);  
    writer.Close();  
}
```



| Bài tập áp dụng

16/09/2024

Tổng quan về lập trình ứng dụng

123



FACULTY OF INFORMATION
TECHNOLOGY
THU DUC COLLEGE OF
TECHNOLOGY



List View

btnHeader

btnAdd

btnRemove

The screenshot shows a Windows form titled 'frmListView'. On the left side, there are three buttons: 'Load Header', 'Add One Item', and 'Remove Items'. The main area of the form contains a list view with three columns: 'ID', 'Name', and 'Birthday'. The list view is currently empty. Below the list view, there are three input fields: 'ID:', 'Name:', and 'Birthday:'. The 'ID' field contains the text 'txtID', the 'Name' field contains the text 'txtName', and the 'Birthday' field contains the text 'dateBirthDay'.

ID	Name	Birthday

ID: txtID

Name: txtName

Birthday: 09/05/2011 dateBirthDay



```
private void btnLoadHeader_Click
(object sender, EventArgs e)
{
    ColumnHeader hdID = new
        ColumnHeader();
    hdID.Text = "ID";
    ColumnHeader hdName = new
        ColumnHeader();
    hdName.Text = "Name";
    ColumnHeader hdBirthday = new
        ColumnHeader();
    hdBirthday.Text = "Birthday";
}
```



```
lvDataDn.GridLines = true;  
lvDataDn.FullRowSelect = true;  
lvDataDn.MultiSelect = true;  
lvDataDn.HideSelection = false;  
  
lvDataDn.View = View.Details;  
  
hdID.Width = 100;  
hdName.Width = 140; hdBirthday.Width = 140;  
lvDataDn.Columns.Clear();
```



```
lvDataDn.Columns.AddRange(new  
ColumnHeader[] {  
hdID,hdName,hdBirthday});  
}
```

At Design Time

?????



```
private void btnAdd_Click
(object sender, EventArgs e)
{
    ListViewItem itemID = new
        ListViewItem();
    itemID.Text = txtID.Text;
    ListViewItem.ListViewSubItem itemName =
    new
    ListViewItem.ListViewSubItem();
    itemName.Text=txtName.Text;
    itemID.SubItems.Add(itemName);
}
```




```
ListViewItem.ListViewSubItem itemBirthday =  
new ListViewItem.ListViewSubItem();  
    itemBirthday.Text =  
dateBirthDay.Value.Day + "/" +  
dateBirthDay.Value.Month + "/" +  
dateBirthDay.Value.Year;  
  
itemID.SubItems.Add(itemBirthday);  
    lvDataDn.Items.Add(itemID);  
}
```



```
private void btnRemove_Click
(object sender, EventArgs e)
{
    ListView.
    SelectedListViewItemCollection
    list = lvDataDn.SelectedItems;
    foreach (ListViewItem item in list)
        lvDataDn.Items.Remove(item);
}
```

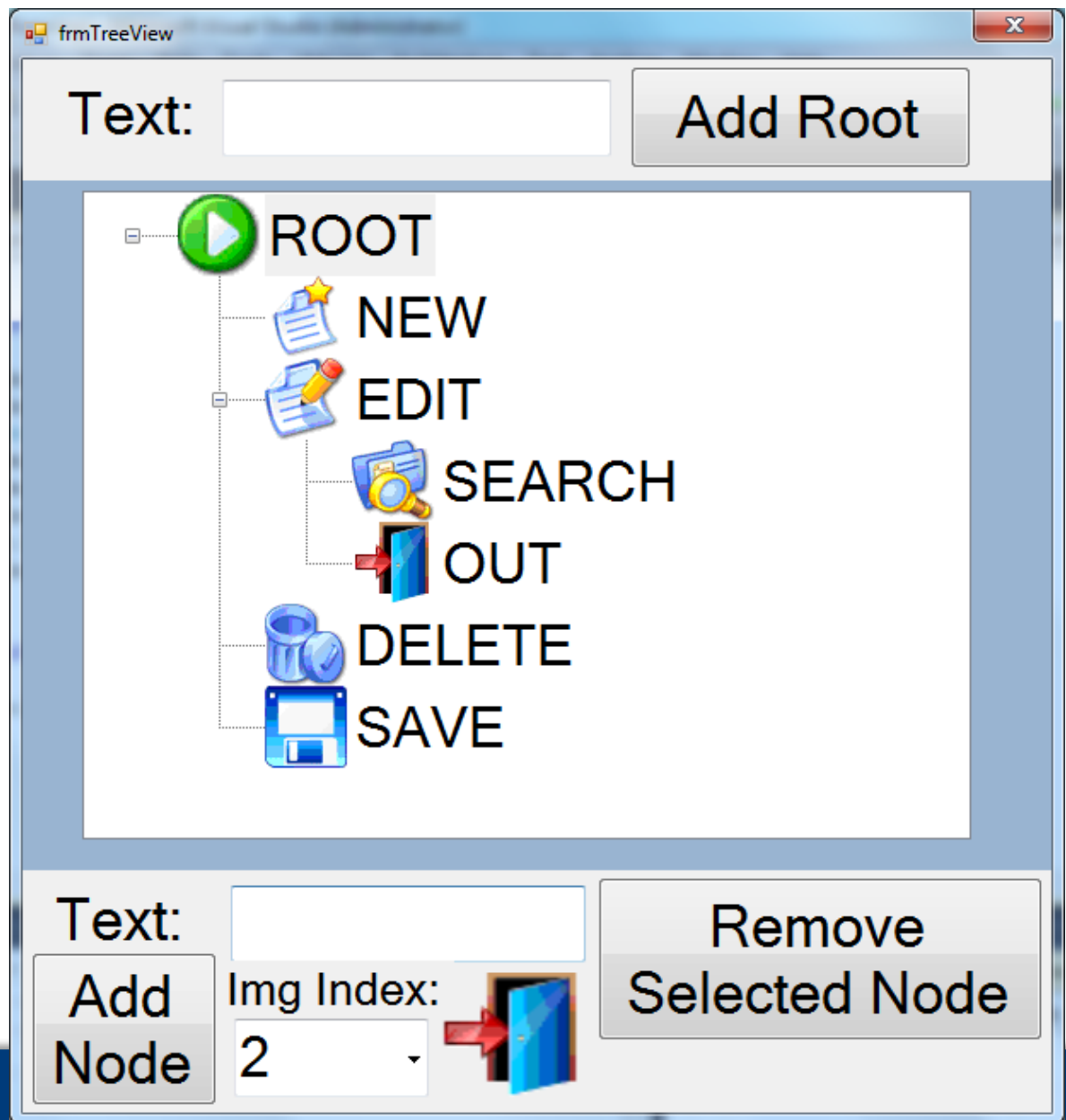


We could use Indices

```
ListView.SelectedIndexCollection  
indices =  
lvDataDn.SelectedIndices;  
foreach (int index in indices)  
{  
    //Process Index here  
}
```



Tree View



ImageList

ImageIndex

SelectedImageIndex

TreeNode

SelectedNode

Nodes
Add
Remove

Expand
Collaps

**Popular
Properties
& Method**



```
private void frmTreeView_Load
(object sender, EventArgs e)
{cboImageIndex.Items.Clear();
  for (int i = 0; i <
    imgList.Images.Count;i++ )
  {
    cboImageIndex.Items.Add(i);
  }
  tvSample.ImageList = imgList;
}
```



```
private void btnAddRoot_Click
(object sender, EventArgs e)
{tvSample.Nodes.Clear();
TreeNode rootNode = new
TreeNode(txtRoot.Text);
rootNode.ImageIndex =
Int32.Parse(cboImageIndex.Text);
rootNode.SelectedImageIndex =
rootNode.ImageIndex;
tvSample.Nodes.Add(rootNode);
txtRoot.Text = "";
}
```



```
private void  
cboImageIndex_SelectedIndexChanged  
(object sender, EventArgs e)  
{  
  
picShow.Image=imgList.Images[Int32  
.Parse(cboImageIndex.Text)];  
}
```




```
private void  
btnRemove_Click(object sender,  
EventArgs e)  
{  
    TreeNode tNode =  
        tvSample.SelectedNode;  
  
    tvSample.Nodes.Remove(tNode);  
}
```



| Bài tập áp dụng

16/09/2024

Tổng quan về lập trình ứng dụng

138



FACULTY OF INFORMATION
TECHNOLOGY
THU DUC COLLEGE OF
TECHNOLOGY

