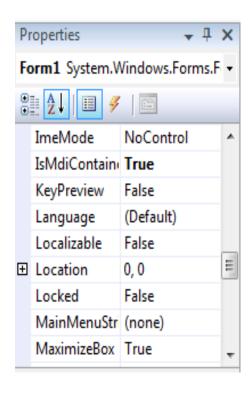


WINDOWS FORMS

FORM VÀ CÁC ĐỊNH DẠNG FORM

- Form gồm ba loại chính:
 - Multiple Document Interface (MDI): hay còn gọi là MDI Form (Form cha) cho phép trình bày các Form khác bên trong Parent Form.
 - Form nằm trong Parent Form là Child Form
 - Normal Form : là các form khi mở ra và không nằm trong Parent Form.

- MDI Form: form được gọi là MDI form khi properties IsMdiContainer được đặt là TRUE.
- TypeOfForm.shn



```
private void button1_Click(object sender, EventArgs e)
{
    Form frm = new Form2();
    frm.IsMdiContainer = true;
    frm.Show ();
}
```

• Child Form:

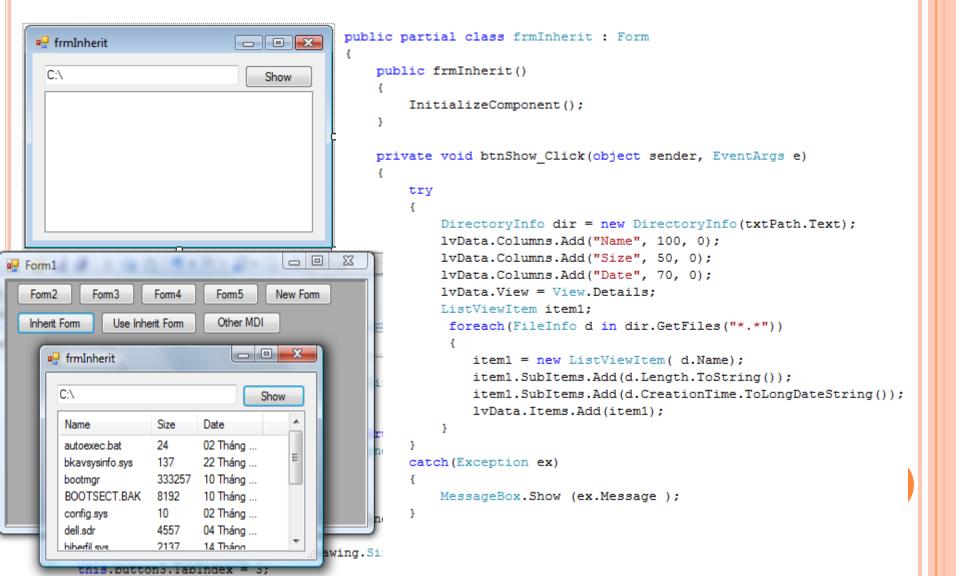
- Là Form khi được Load, nó sẽ nằm trong vùng làm việc của MDI Form.
- Để trở thành Child Form, thuộc tính MdiParent phải được khai báo trong code (không có trong cửa sổ Properties)
- Kích thước và không gian hoạt động phụ thuộc vào MDI Form.

```
private void button2_Click(object sender, EventArgs e)
{
    // khai bao va tao form
    Form frm = new Form3();
    // MDI form la form hien hanh
    frm.MdiParent = this;
    // dua form moi len man hinh
    frm.Show();
}
```

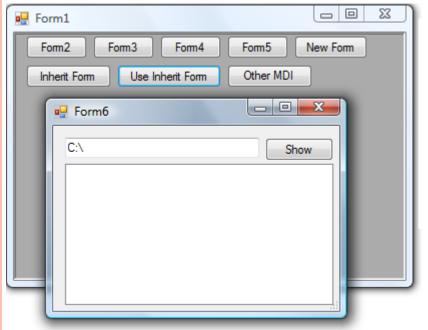
- Normal Form: là form khi load không là MDI hay Child form.
 - Dùng Show() cho phép form chuyển sang các form mở trước đó.
 - ShowDialog() sẽ mở dạng Modal, không cho phép chuyển sang các form khác đang mở

FORM KÉ THỦA

- Khi có nhiều form định dạng tương tự, các form có thể được thiết kế, tạo code và sau đó cho kế thừa vào các form khác.
 - Ví dụ: tạo form dùng để trình bày một list các file của một folder. Khi trình bày danh sách các mẫu tin, chỉ cần khai báo kế thừa từ Form trên và chỉ cần thay đổi định dạng.
 - Tạo form tên frmInherit, và 3 thành phần control như ListView, TextBox và Button.



o Một form kế thừa từ form Inherite:



```
public partial class Form6 : frmInherit
{
    public Form6()
    {
        InitializeComponent();
    }
    private void Form6_Load(object sender, EventArgs e)
    {
     }
}
```

CÁC HÌNH DẠNG CỦA FORM

- Mặc định của form là hình chữ nhật, nhưng chúng có thể được thay đổi tùy theo hình thức xuất hiện như hình tròn, ellipse hay hình bất kỳ.
- Dùng thuộc tính Region và đối tượng thuộc GraphicsPath
- Trong namespace : System.Drawing.Drawing2D



```
private void button1 Click(object sender, EventArgs e)
    Form frm = new frmCircle();
    GraphicsPath shape;
    shape = new GraphicsPath();
    shape.AddEllipse(15, 25, this.Width, this.Height);
    frm.Region = new Region(shape);
    frm.Show();
}
 private void button2 Click(object sender, EventArgs e)
     Form frm = new frmOthers();
     GraphicsPath shape;
     shape = new GraphicsPath();
     Point point1 = new Point(100, 50);
     Point point2 = new Point(250, 100);
     Point point3 = new Point(300, 200);
     Point point4 = new Point(180, 300);
     Point point5 = new Point(80, 200);
     Point point6 = new Point(30, 150);
     Point[] points = { point1, point2, point3, point4, point5, point6 };
     shape.AddCurve(points);
     frm.Region = new Region(shape);
     frm.Show();
 }
```

CÁC NHÓM ĐIỀU KHIỂN THÔNG THƯỜNG

• CheckBox :dùng cho chọn lựa Yes/No hay True/False.

o Thuộc tính:

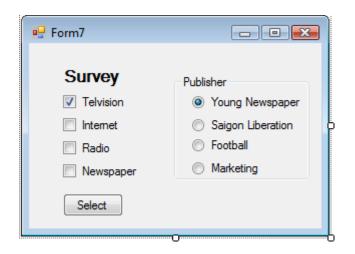
Flat Style: Kiểu đường viền

Apprearance: Hình dạng của Control

Checked: Trạng thái chọn True hay False

CheckState: Trạng thái đang chọn như Checked, Unchecked và Indeterminatte.

ThreeState: cho phép hay không 3 trạng thái trên



```
private void button1 Click(object sender, EventArgs e)
    string choose = "";
    foreach (Control chk in this.Controls)
        if (chk is CheckBox)
            if (((CheckBox)chk).Checked==true)
            choose += chk.Text + ",";
    MessageBox.Show("You are selected: " +choose );
```

• Event :

MouseClick: Khi user điều khiển bằng click Mouse.

CheckedChanged: khi User click vào checkBox

CheckStateChanged: khi CheckState thay đổi

EnabledChanged: khi thuộc tính Enabled thay đổi từ True sang False hay ngược lại

VisibleChanged: khi thuộc tính visible thay đổi từ True sang False hay ngược lại.

- CheckedListBox: dùng để hiển thị danh sách phần tử mà bên cạnh có biểu tượng CheckBox ứng với mỗi phần tử.
- CheckedListBox cũng có 3 trạng thái : Checked, Indeterminate(trạng thái này dùng trong code), Unchecked.

o Thuộc tính:

BorderStyle: Kiểu đường viền.

Multicolumn: Trình bày danh sách gồm nhiều cột.

ColumnWidth: Chiều rộng cột.

Item: Tập các phần tử, dùng method Add hay AddRange để thêm thành phần.

Selection Mode: Chọn một hay nhiều phần tử.

Sorted: True thì sẽ theo kiểu tăng dần.

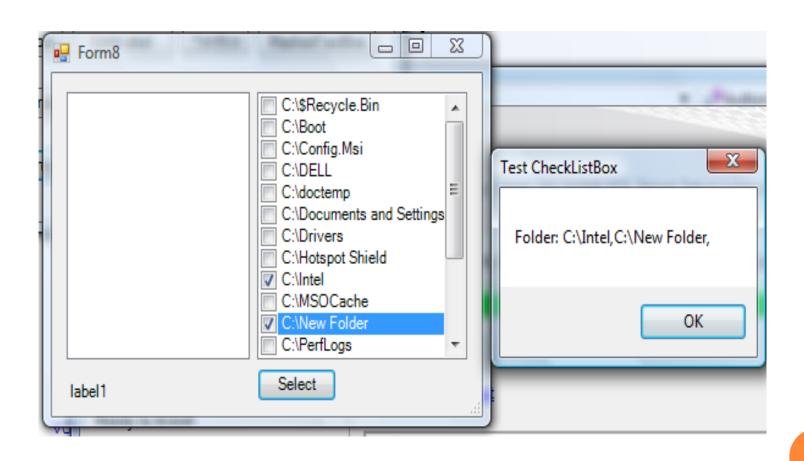
SelectedItems: Trả về tập các phần tử được chọn.

SelectedItem: Gán hay lấy giá trị object ứng với phần tử đang chọn.

Selected Value: Gán hay lấy giá trị ứng với phần tử kiểu object đang chọn

SelectedIndex: Gán hay lấy index ứng với phần tử đang chọn.

CheckedItems : Trả về tập phần tử được check



• Events:

ItemChecked: Khi user Click vào CheckBox control

SelectedIndexChanged: Khi index của phần tử bị thay đổi.

SelectedValueChanged: khi giá trị của phần tử được thay đổi trên control.

EnabledChanged: Khi thuộc tính Enabled đổi từ True sang False hay ngược lại

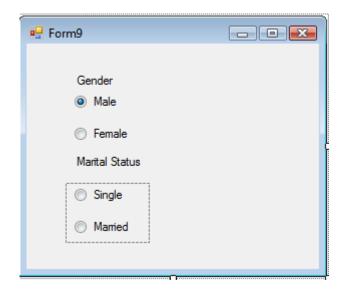
VisibleChanged : Khi thuộc tính Visible thay đổi giá trị từ True sang False hay ngược lại. • RadioButton: Khi CheckBox cho phép user chọn nhiều thành phần, thì control này chỉ cho phép chọn một trong các tùy chọn.

o Thuộc tính:

FlatStype: Kiểu đường viền.

Appearance: Hình dạng của control

Checked: Trạng thái chọn True /False



```
public Form9()
{
    InitializeComponent();
}

private void radioButton1_CheckedChanged(object sender, EventArgs e)
{
    if(radioButton1.Checked)
        MessageBox.Show("You are Male!");
    else
        MessageBox.Show("You are Female!");
}
```

• Event :

EnabledChanged: Khi thuộc tính Enabled đổi từ True sang False hay ngược lại

VisibleChanged: Khi thuộc tính Visible thay đổi giá trị từ True sang False hay ngược lại.

CÁC ĐIỀU KHIỂN ĐẶC BIỆT

- Tooltip, HelpProvider, ErrorProvider
- ProgressBar
- ListView
- TreeView
- DateTimePicker, MonthCalendar

 Tooltip: sẽ trình bày một window nhỏ cho các diễn giải ngán gọn khi người dùng đưa mouse vào control

o Thuộc tính:

BackColor: màu nền.

AutomaticDelay: tính bằng ms áp dụng cho 3 thuộc tính AutoPopDelay, InitialDelay, ReshowDelay

IsBalloon: mặc định là False, nếu là True thì window có các góc tròn.

AutoPopDelay: thời gian là ms từ khi window ToolTip xuất hiện cho đến khi kết thúc, nếu user đưa mouse vào controls.

InitialDelay : Thời gian là ms mà window ToolTip xuất hiện từ khi user đưa mouse vào Control

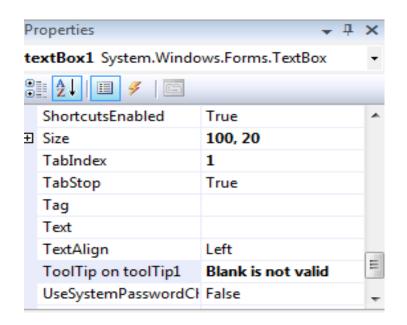
ReshowDelay: Thời gian là ms mà window ToolTip tắt từ khi user đưa mouse ra Control.

ToolTipTitle: Chuỗi xuất hiện bên cạnh biểu tượng.

ToolTipIcon :Biểu tượng xuất hiện bên cạnh chuỗi khai báo trong thuộc tính ToolTipTitle.







• Chúng ta có thể dùng code để thể hiện ToolTip thay vì trong phần điều khiển Properties bằng các method sau:

SetToolTip: Khai báo chuỗi cho control tương tự thuộc tính ToolTip on toolTip1

GetToolTip: Nhận chuỗi trã về từ thuộc tính ToolTip on toolTip1

RemoveAll: Loại bỏ các ToolTip text.

• ErrorProvider : cung cấp các giao tiếp với user các thông tin lỗi của Control trên form.

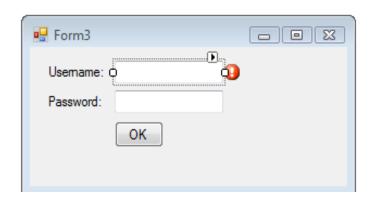
o Thuộc tính:

Icon: Biểu tượng sẽ xuất hiện bên cạnh control

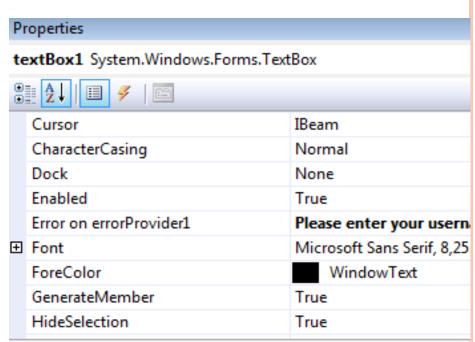
BlinkRate: thời gian là ms áp dụng cho khoảng cách giữa 2 lần nhấp nháy.

IsBalloon: Nếu là True thì window hiển thị có các góc tròn.

BlinkStyle: default là BlinkIfDifferentError, biểu tượng chỉ nhấp nháy khi lổi xảy ra khác với lỗi trước. Nếu để nhấp nháy liên tục AlwaysBlink







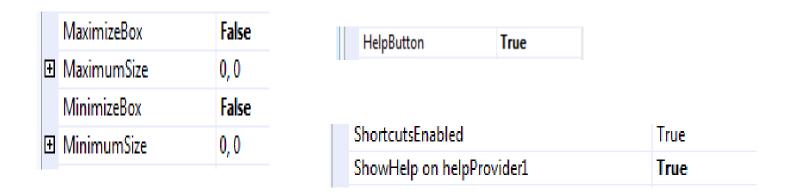
• Phương Thức:

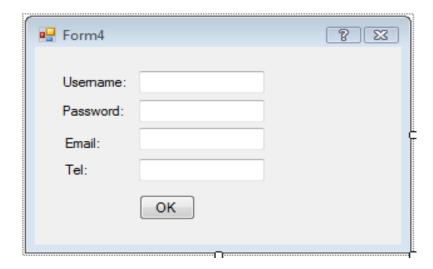
- SetError : Khai báo chuỗi cho control tương tự như thuộc tính Error on ErrorProvider.
- GetError : Nhận chuỗi từ thuộc tính Error on ErrorProvider của control
- Clear: bỏ các thông tin.

- HelpProvider: cung cấp cửa sổ trợ giúp bằng file hay URL.
- Mỗi đối tượng của HelpProvider nắm giữ tập các tham chiếu đến các control.
- HelpNameSpace chứa các khai báo để kết hợp sử dụng các file trợ giúp tương ứng với HelpProvider
- Ngoài ra còn sự kết hợp của phương thức SetHelpProvider và HelpNavigator,
- HelpProvider thể hiện các tiện ích thông qua nút
 ? Hay F1.
- Thuộc tính HelpNameSpace chỉ định tên file trợ giúp dạng .chm hay HTML

• Với HelpString cho phép hiện chuỗi khi user dùng nút?

• Để có nút?





- HelpNameSpace : chứa các thông tin đường dẫn như đến các file .chm hay HTML
 - Chon F1 HelpProvider .
 - Đưa đường dẫn vào HelpNameSpace.
 - Khi F1, sẽ chạy file trong đường dẫn

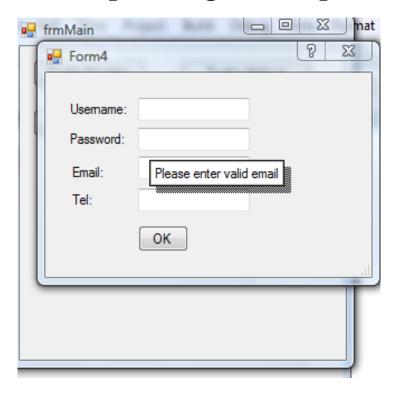


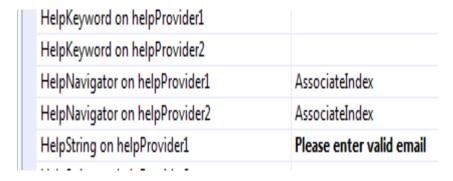
+	(ApplicationSettings)	
	(Name)	helpProvider1
	GenerateMember	True
	HelpNamespace	D:\SPKT\WINFORM
	Modifiers	Private
	Tag	

- HelpKeyWord chứa chức năng giúp đở theo KeyWord
- Ví dụ tại TextBox PassWord khi cần Help sẽ tìm theo KeyWord là Password

HelpKeyword on helpProvider1	Password
HelpKeyword on helpProvider2	
HelpNavigator on helpProvider1	Index
HalnNavinator on halnDrovidar?	AccordateIndev

 Như vậy khi dùng nút? Sẽ đưa ra thông báo tương ứng với những gì được nhập trong HelpString on helpProvider1



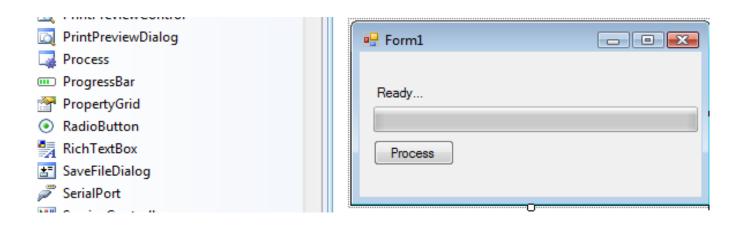


• Phương Thức:

- SetHelpKeyword :Khai báo chuỗi từ khóa dùng để tìm kiếm khi cửa số Help kích hoạt, tương tự thuộc tính HelpKeyword on helpProvider1
- SetHelpString: Khai báo chuỗi trình bày khi người sử dụng trỏ đến control sau khi nhấn nút? Tương tự thuộc tính HelpString on helpProvider1
- SetHelpNavigator: Khai báo được chọn khi của số Help kích hoạt tương tự HelpNavigator on helpProvider1.

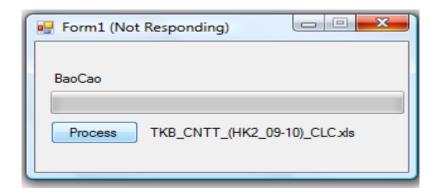
- PROGRESSBAR: thể hiện thời lượng đã thực hiện bằng 3 loại:
 - Blocks: từng khối xanh từ trái qua phải
 - Continuous : khối xanh từ trái qua phải
 - Marquee : cuộn từng khối xanh từ trái qua phải.

0

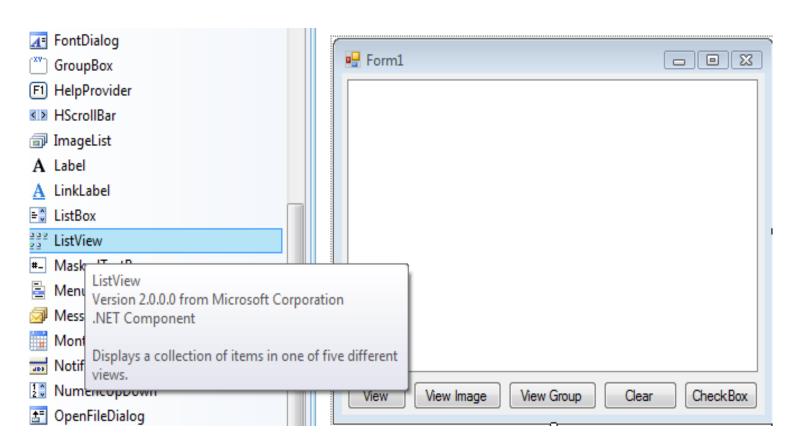


o Thuộc tính:

- Maximum : Giới hạn phần trên của ProgressBar
- Minimum: Giới hạn phần dưới
- Value: Giá trị hiện hành đang xử lý của ProgressBar control.
- Style : kiểu trình bày.
- Step: giá trị ứng với lần nhảy khi xử lý control



o LISTVIEW: Dùng để trình bày phần tử dạng danh sách với nhiều dạng khác nhau.

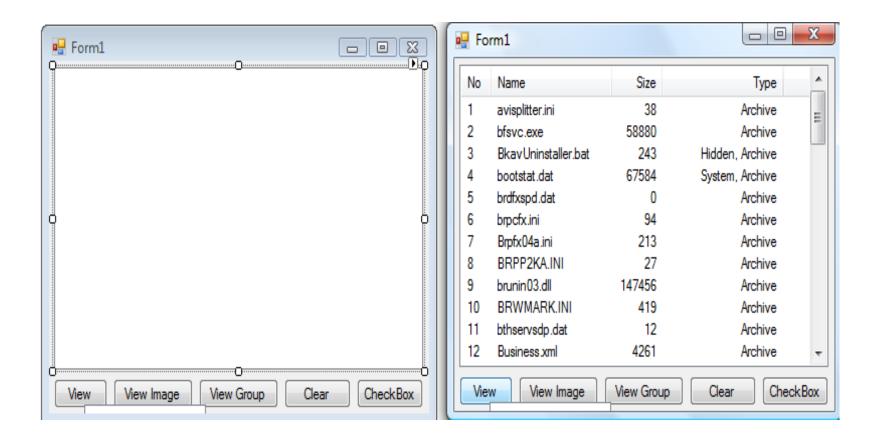


o Thuộc tính:

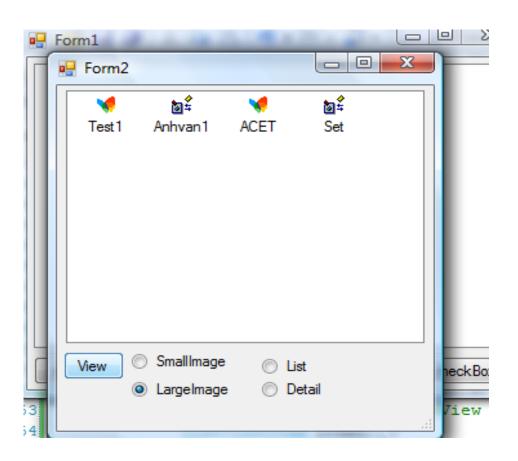
- AllowColumnReorder : Cho phép sắp xếp cột trên ListView ở chế độ thực thi.
- CheckBoxes: Xuất hiện CheckBox bên cạnh từng phần tử của ListView
- Columns: Khai báo số cột (có tiêu đề)
- FullRowSelect : True sẽ cho phép tô màu theo hàng của phần tử được chọn.
- Group : Khai báo nhóm để phân tách loại phần tử sau khi trình bày trên ListView
- Label Edit: True nếu cho phép user thay đối chuỗi của mỗi phần tử.
- LabelWrap: True thì chuỗi sẽ tự động xuống hàng khi hết chiều dài.
- MultiSelect: True cho phép chọn nhiều phần tử,

- LargeImageList :Đối tượng ImageList chứa danh sách các Image theo index từ 0--- N-1 (N là số lượng hình) dùng trong trường hợp thuộc tính View là LargeIcon
- SmallImageList : tương tự trên nhưng View là SmallIcon
- Sorting: Asccending khi sắp xếp theo tăng dần.
- View: Chế độ trình bày gồm : List; Details; LargeIcon; SmallIcon; Title.
- SelectedItems : Trả về danh sách các phần tử được chọn.
- CheckedItems : Trả về danh sách các phần tử được Check.

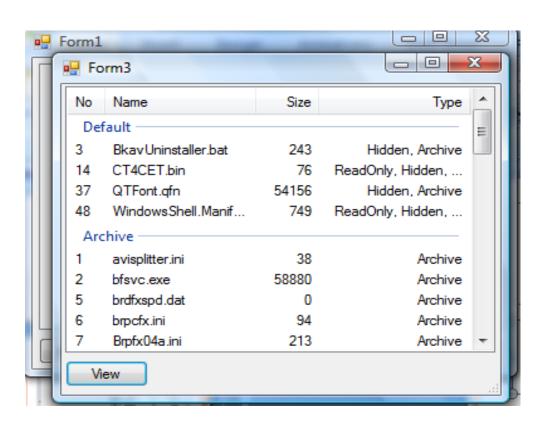
Ví dụ: Liệt kê danh sách tập tin



o Chọn hình thức trình bày:



Nhóm phần tử:



• Phương thức :

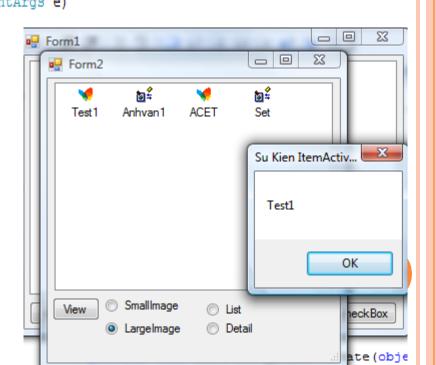
- Clear : Xóa các phần tử và cột trên ListView
- ArrangeIcons: Sắp xếp phần tử trình bày trên ListView theo giá trị.

```
private void button4_Click(object sender, EventArgs e)
{
    listView1.Clear();
}
```

• Event :

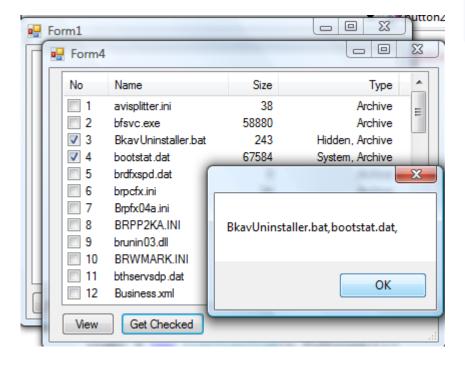
- SelectedIndexChanged : Khi user thay đổi phần tử được chọn trên ListView.
- ItemActivate: Khi chọn phần tử trên ListView
- ItemChecked: Khi Check vào CheckBox trên ListView.
- Ví dụ dùng ItemActivate để lấy giá trị của phần tử đang chọn.

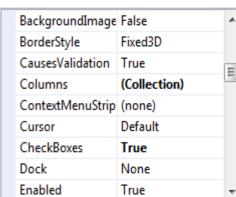
```
private void listView1_ItemActivate(object sender, EventArgs e)
{
    ListViewItem item = listView1.FocusedItem;
    MessageBox.Show(item.Text, "Su Kien ItemActivate");
}
```



• Sử dụng thuộc tính CheckBoxes và CheckItems trong ListView.

```
int i = 0;
listView1.FullRowSelect = true;
// listView1.CheckBoxes = true;
listView1.View = View.Details;
ListViewItem item1;
```

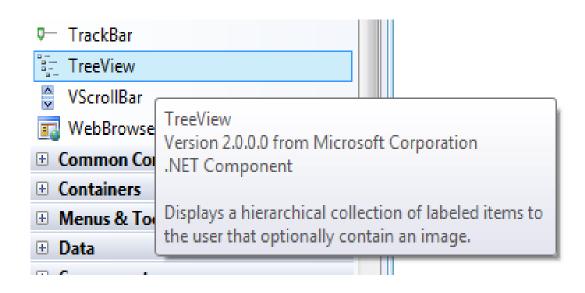




```
private void button2_Click(object sender, EventArgs e)
{
    string strChecked="";
    foreach (ListViewItem item in listView1.CheckedItems)
    {
        strChecked += item.SubItems[1].Text + ",";
    }
    // Đua đổi tượng được chọn ra
    if (!strChecked.Equals(""))
        MessageBox.Show(strChecked);
}
```

TREEVIEW

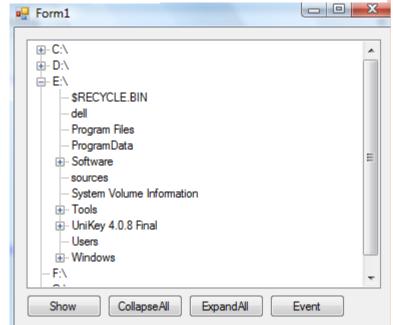
 Điều khiển TreeView dùng để trình bày danh sách phần tử phân theo Node- hình cây. Trong đó mỗi đối tượng TreeView nắm giữ danh sách các Node con.



o Thuộc tính:

- CheckBox : cho phép CheckBox xuất hiện tại các node
- Nodes: Khai báo số Node và tiêu đề.
- FullRowSelected : True sẽ cho phép tô màu theo hàng của phần tử được chọn.
- ShowLine: True sẽ cho phép đường viền đi theo các node
- LabelEdit: True thi user sẽ có khả năng thay đối nhản của Node.
- ShowPlusMinus : True sẽ xuất hiện + và –
- ShowRoorLine: True sẽ cho phép xuất hiện từ gốc.
- ImageList : Đối tượng ImageList chứa danh sách các Image đánh số từ 0---N-1.
- ImageIndex : Chỉ mục Image từ 0—N-1 tương ứng với Node.
- SelectImageIndex : Chỉ mục Image từ 0—N-1 tương ứng với Node được chọn.

```
void GetDisk()
   int i = 0; // lấy danh sách HDD
   foreach (string d in Directory.GetLogicalDrives())
    {// đưa vào Node
        this.treeView1.Nodes.Add(d);
        GetFolder(d, i); // gọi GetFolder để lấy Folder của HDD
        i++;
}// GetFolder chứa tham số tham chiếu đến HDD và vi trí của Node
void GetFolder(string name, int level)
-
    try
    {
        int level1 = 0:
        foreach (string d in Directory.GetDirectories(name))
            this.treeView1.Nodes[level].Nodes.Add(d.Substring(3));
            GetFile(d, level, level1); // lấy các file trong Folder
            level1++;
        }
    }
```



• Phương thức:

- CollapseAll: Thu gọn các node lại
- ExpandAll: Triển Khai các node

```
private void button2_Click(object sender, EventArgs e)
{
    treeView1.CollapseAll();
}

private void button3_Click(object sender, EventArgs e)
{
    treeView1.ExpandAll();
}
```

• Event :

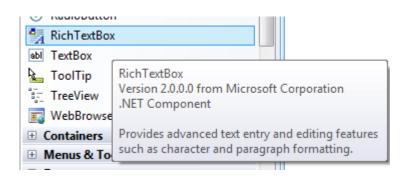
- AfterCheck: Xåy ra sau khi user check vào CheckBox.
- AfterCollapse: Xảy ra sau khi thu gọn các nodes
- AfterSelect : Xåy ra sau khi user click trên node
- BeforeCheck : Xảy ra trước khi user check vào CheckBox
- BeforeCollapse:
- BeforeExpand:
- BeforeSelect:
- Click: Xåy ra khi user click trên node
- DoubleClick : Xåy ra khi double-click trên node

Ví dụ dùng AfterSelect

```
private void button1 Click(object sender, EventArgs e)
    string SQL = "select name from ";
    SQL+="master.dbo.sysdatabases";
    GetDatabase(SQL);
private void treeView1 AfterSelect(object sender, TreeViewEventArgs e)
    string SQL = "select id, name, crdate from ";
    SQL += e.Node.Text +".dbo.sysobjects";
    SQL += " where type='U' order by name";
    GetTable(SQL);
```

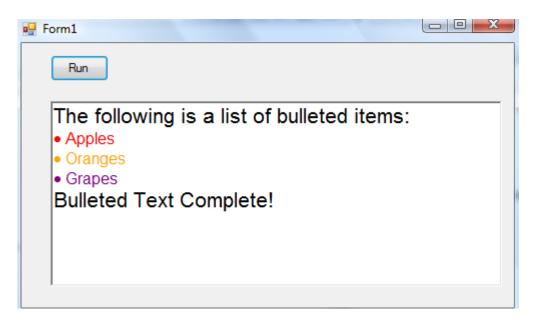
RICHTEXTBOX

- RichTextBox cung cấp khả năng nhập và chỉnh sửa các đoạn text.
- Control này cũng cung cấp khả năng và các đặc tính định dạng cao hơn dạng TextBox cơ bản.
- Text có thể được gán trực tiếp cho Control này hay có thể được nạp từ dạng Rich Text Format (.rtf) hay dạng văn bảng dạng thường (.txt).
- Text trong Control có thể được gán ký tự hay được định dạng.



- RichTextBox cung cấp các thuộc tính giúp tùy biến và thay đổi định dạng Text như:
 - SelectionFont
 - SelectionBullet
 - SelectedText
 - SelectionColour
- Ví dụ: richTextBox.cs

```
richTextBox1.Clear();
// Set the font for the opening text to a larger Arial font;
richTextBox1.SelectionFont = new Font("Arial", 16);
// Assign the introduction text to the RichTextBox control.
richTextBox1.SelectedText = "The following is a list of bulleted items:" + "\n";
// Set the Font for the first item to a smaller size Arial font.
richTextBox1.SelectionFont = new Font("Arial", 12);
// Specify that the following items are to be added to a bulleted list.
richTextBox1.SelectionBullet = true;
// Set the color of the item text.
richTextBox1.SelectionColor = Color.Red;
// Assign the text to the bulleted item.
richTextBox1.SelectedText = "Apples" + "\n";
```



- o RichTextBox cũng cung cấp các method thực hiện các chức năng như đọc (LoadFile) và ghi file (SaveFile) dưới dạng (.rtf) và ASCII text file.
- Phương thức nạp chồng Find cũng được cung cấp nhằm tìm kiếm chuỗi hay các ký tự trong Text
- o ví dụ:
 - richTextBox1.cs

	Name	Description
≟ ≬	<u>LoadFile(String)</u>	Loads a rich text format (RTF) or standard ASCII text file into the RichTextBox control.
≟ ≬	<u>LoadFile(Stream, RichTextBoxStreamType)</u>	Loads the contents of an existing data stream into the <u>RichTextBox</u> control.
≟ ≬	<u>LoadFile(String, RichTextBoxStreamType)</u>	Loads a specific type of file into the <u>RichTextBox</u> control.

```
OpenFileDialog openFile1 = new OpenFileDialog();

// Initialize the OpenFileDialog to look for RTF files.
openFile1.DefaultExt = "*.rtf";
openFile1.Filter = "RTF Files|*.rtf";

// Determine whether the user selected a file from the OpenFileDialog.
if (openFile1.ShowDialog() == System.Windows.Forms.DialogResult.OK && openFile1.FileName.Length > 0)
{
    // Load the contents of the file into the RichTextBox.
    richTextBox1.LoadFile(openFile1.FileName);
}
```

Name	Description
SaveFile(String)	Saves the contents of the RichTextBox to a rich text format (RTF) file.
SaveFile(Stream, RichTextBoxStreamType)	Saves the contents of a RichTextBox control to an open data stream.
SaveFile(String, RichTextBoxStreamType)	Saves the contents of the RichTextBox to a specific type of file.

```
// Create a SaveFileDialog to request a path and file name to save to.
SaveFileDialog saveFile1 = new SaveFileDialog();

// Initialize the SaveFileDialog to specify the RTF extention for the file.
saveFile1.DefaultExt = "*.rtf";
saveFile1.Filter = "RTF Files|*.rtf";

// Determine whether the user selected a file name from the saveFileDialog.
if (saveFile1.ShowDialog() == System.Windows.Forms.DialogResult.OK &&
    saveFile1.FileName.Length > 0)
{
    // Save the contents of the RichTextBox into the file.
    richTextBox1.SaveFile(saveFile1.FileName);
}
```

```
Name
                                  Description
Find(Char[])
                                  Searches the text of a RichTextBox control for the first instance of a character from a list of characters.
Find(String)
                                  Searches the text in a RichTextBox control for a string.
                                  Searches the text of a RichTextBox control, at a specific starting point, for the first instance of a character from a list of
Find(Char[], Int32)
                                  characters.
Find(String, RichTextBoxFinds)
                                  Searches the text in a RichTextBox control for a string with specific options applied to the search.
Find(Char[], Int32, Int32)
                                  Searches a range of text in a RichTextBox control for the first instance of a character from a list of characters.
Find(String, Int32, RichTextBoxFinds)
                                  Searches the text in a RichTextBox control for a string at a specific location within the control and with specific options
                                  applied to the search.
Find(String, Int32, Int32,
                                  Searches the text in a RichTextBox control for a string within a range of text within the control and with specific options
RichTextBoxFinds)
                                  applied to the search.
private void button3 Click(object sender, EventArgs e)
     MessageBox.Show(FindMyText(new char[] { 'W', 'i', 'n', 'f', 'o', 'r', 'm' }).ToString(), "Tim thay");
public int FindMyText(char[] text)
     // Initialize the return value to false by default.
     int returnValue = -1:
     // Ensure that a search string has been specified and a valid start point.
     if (text.Length > 0)
          // Obtain the location of the first character found in the control
          // that matches any of the characters in the char array.
          int indexToText = richTextBox1.Find(text);
          // Determine whether the text was found in richTextBox1.
          if (indexToText >= 0)
                // Return the location of the character.
                returnValue = indexToText +1:
     return returnValue:
```

```
public bool FindMyText(string text)
   // Initialize the return value to false by default.
   bool returnValue = false:
   // Ensure a search string has been specified.
    if (text.Length > 0)
       // Obtain the location of the search string in richTextBox1.
       int indexToText = richTextBox1.Find(text);
       // Determine whether the text was found in richTextBox1.
        if (indexToText >= 0)
           returnValue = true:
    return returnValue:
private void button4 Click(object sender, EventArgs e)
   MessageBox.Show(FindMyText("Winform").ToString(), "Tim thay");
```

• Note: DialogResult: trả về các giá trị từ Dialog control hay các hộp thoại.

Member name	Description
None	Nothing is returned from the dialog box. This means that the modal dialog continues running.
OK	The dialog box return value is OK (usually sent from a button labeled OK).
Cancel	The dialog box return value is Cancel (usually sent from a button labeled Cancel).
Abort	The dialog box return value is Abort (usually sent from a button labeled Abort).
Retry	The dialog box return value is Retry (usually sent from a button labeled Retry).
Ignore	The dialog box return value is Ignore (usually sent from a button labeled Ignore).
Yes	The dialog box return value is Yes (usually sent from a button labeled Yes).
No	The dialog box return value is No (usually sent from a button labeled No).

DATE & MONTH

 DateTime: tạo các thể hiện về thời gian như giờ và ngày tháng

```
DateTime.Now; 5/16/2006 1:05:13 AM
DateTime.Now.ToString(); 5/16/2006 1:05:13 AM
DateTime.Now.ToShortTimeString() 11:40 AM
DateTime.Now.ToShortDateString() 5/16/2006
DateTime.Now.ToLongTimeString() 11:40:13 AM
DateTime.Now.ToLongDateString() Tuesday, May 16, 2006

DateTime date1 = new DateTime(2008, 3, 1, 7, 0, 0);
outputBlock.Text += date1.ToString() + "\n";
```

```
private void button1_Click(object sender, EventArgs e)
{

   DateTime date1 = new DateTime(2008, 3, 1, 7, 0, 0);
   DateTime today = DateTime.Now;
   label2 .Text = (date1.ToString());
   label1.Text = (today.ToString());
}
```

- DateTimePicker control cho phép người điều khiển làm việc với thời gian như ngày và giờ.
- Thuộc tính:
 - customFormat: định dạng do người dùng tùy biến.
 - Format: Định dạng do hệ thống cung cấp
 - *MaxDate*: 31/12/9998 là max có thể chọn
 - Mindate: chọn giá trị nhỏ nhất của ngày tháng năm
 - Check: True khi CheckBox sẽ check nếu ShowCheckBox xác định là True
 - showCheckBox: True sẽ kích hoạt chức năng CheckBox
 - ShowUpDown: False nếu muốn theo hình thức DropDown.
 - Value: gán hay lấy giá trị về thời gian của DateTimePicker

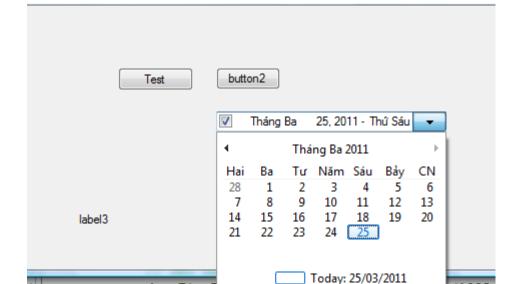
VÍ Dụ:

```
// Create a new DateTimePicker control and initialize it.
DateTimePicker dateTimePicker1 = new DateTimePicker();

// Set the MinDate and MaxDate.
dateTimePicker1.MinDate = new DateTime(1985, 6, 20);
dateTimePicker1.MaxDate = DateTime.Today;

// Set the CustomFormat string.
dateTimePicker1.CustomFormat = "MMMM dd, yyyy - dddd";
dateTimePicker1.Format = DateTimePickerFormat.Custom;

// Show the CheckBox and display the control as an up-down control.
dateTimePicker1.ShowCheckBox = true;
// dateTimePicker1.ShowUpDown = true;
```



Examples of DateTime for Given Formats

DateTime.Now.ToString("dddd, MMMM dd yyyy") - Tuesday, May 16 2006
DateTime.Now.ToString("ddd, MMMM d "'"yy") - Tue, May 16 '06
DateTime.Now.ToString("dddd, MMMM dd") - Tuesday, May 16
DateTime.Now.ToString("M/yy") - 5/06
DateTime.Now.ToString("dd-MM-yy") - 16-05-06

d	Displays one- or two-digit day.	If the day is 8, this format will display the day as "8" instead of "08".
h	Displays one- or two-digit hour in 12-hour format.	If the time is 1:10 PM, this format will display the hour as "1" instead of "13".
Н	Displays one- or two-digit hour in 24-hour format.	If the time 1:30 PM, this format will display the hour as "13" instead of "1".
m	Displays one- or two-digit minute.	If the time 2:05 PM, this format will display the minutes as "5" instead of "05".
М	Displays one- or two-digit number indicating the month.	If the month is 9, this format will display the month as "9" instead of "09".
s	Displays one- or two-digit seconds.	If the time is 1:45:02 P.M, this format will display the seconds as "2" instead of "02".
У	Displays one-digit year.	If the year is 2008, this format will display the day as "8" instead of "2008".

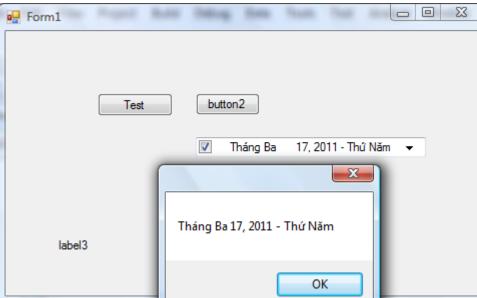
```
// Set the CustomFormat string.
dateTimePicker1.CustomFormat = "MMMM dd, yyyy - dddd";
dateTimePicker1.Format = DateTimePickerFormat.Custom;
```

Name	Description
Custom	Used to display the date or time value in custom format.
Long	Used to display the date or time value in the long date format, which is set by the user's operating system.
Short	Used to display the date or time value in the short date format, which is set by the user's operating system
Time	Used to display the date or time value in the time format, which is set by the user's operating system.

• Event:

- CloseUp: khi user kết thúc chọn giá trị
- FormatChanged: khi user thay đổi định dạng
- ValueChanged: khi user chọn giá trị khác với giá trị trước đó.

```
private void dateTimePicker1_ValueChanged(object sender, EventArgs e)
{
    MessageBox.Show(dateTimePicker1.Text.ToString());
}
```



MONTHCALENDAR

• Month Calendar cho phép trình bày tương tự Date Time Picker, cho phép user chọn giá trị thời gian như một lịch tháng và thể hiện với các giá trị cho trước.

• Thuộc tính:

- *FirstDayofWeek*: Ngày đầu tiên mặc định là ngày khai báo trong hệ thống.
- *MaxDate*: tối đa là 31/12/9998
- MinDate: chọn ra tối thiểu của lịch
- **SelectionRange**: Khoảng thời gian từ lúc bắt đầu đến kết thúc.
- **ShowToday:** True thì biểu tượng và ngày hiện tại sẽ xuất hiện.
- ShowTodayCircle: Đường viền ngày hiện tại
- ToDayDate: Gán hay lấy giá trị hiện tại

EVENT

- DateChanged: Khi user dùng giá trị khác với trị trước đó.
- DateSelected: Khi chọn giá trị trên Control

```
private void monthCalendar1_DateSelected(object sender, System.Windows.Forms.DateRangeEventArgs e)
{
    // Show the start and end dates in the text box.
    textBox1.Text = "Date Selected: Start = " +
        e.Start.ToShortDateString() + " : End = " + e.End.ToShortDateString();
}

private void monthCalendar1_DateChanged(object sender, System.Windows.Forms.DateRangeEventArgs e)
{
    // Show the start and end dates in the text box.
    textBox1.Text = "Date Changed: Start = " +
        e.Start.ToShortDateString() + " : End = " + e.End.ToShortDateString();
}
```

BOLD DISPLAY

Tháng Ba 2011

Н	ai	Ba	Tư	Năm	Sáu	Bảy	CN	
		1	2	3	4	5	6	14
	7	8	9	10	11	12	13	15
					18			
2	1	22	23	24	25	26	27	17
2	8	29	30	31				18

MULTIPLE MONTH

// Configure the calendar to display 3 rows by 4 columns of months.
monthCalendar1.CalendarDimensions = new System.Drawing.Size(4, 3);

◀ Tháng Giêng 2011							èng 2011 Tháng Hai 2011									Tháng Ba 2011									Tháng Tư 2011 →							
	Hai	Ba	Tư	Năm	Sáu	Bảy	CN		Hai	Ba	Tư	Năm	Sáu	Bảy	CN		Hai	Ba	Tư	Năm	Sáu	Bảy	CN		Hai	Ba	Tư	Năm	Sáu	Bảy	CN	
1	27	28	29	30	31	1	2	6		1	2	3	4	5	6	10		1	2	3	4	5	6	14					1	2	3	
2	3	4	5	6	7	8	9	7	7	8	9	10	11	12	13	11	7	8	9	10	11	12	13	15	4	5	6	7	8	9	10	
3	10	11	12	13	14	15	16	8	14	15	16	17	18	19	20	12	14	15	16	17	18	19	20	16	11	12	13	14	15	16	17	
4	17	18	19	20	21	22	23	9	21	22	23	24	25	26	27	13	21	22	23	24	25	26	27	17	18	19	20	21	22	23	24	
5 6	24 31	25	26	27	28	29	30	10	28							14	28	29	30	31				18	25	26	27	28	29	30		
		1	Tháng	Năm 2	2011			Tháng Sáu 2011							Tháng Bảy 2011						Tháng Tám 2011											
	Hai	Ba	Tư	Năm	Sáu	Bảy	CN		Hai	Ba	Tư	Năm	Sáu	Bảy	CN		Hai	Ba	Tư	Năm	Sáu	Bảy	CN		Hai	Ba	Tư	Năm	Sáu	Bảy	CI	
18							1	23			1	2	3	4	5	27					1	2	3									
19	2	3	4	5	6	7	8	24	6	7	8	9	10	11	12	28	4	5	6	7	8	9	10	32	1	2	3	4	5	6		
20	9	10	11	12	13	14	15	25	13	14	15	16	17	18	19	29	11	12	13	14	15	16	17	33	8	9	10	11	12	13	1	
21	16	17	18	19	20	21	22	26	20	21	22	23	24	25	26	30	18	19	20	21	22	23	24	34	15	16	17	18	19	20	2:	
22	23	24	25	26	27	28	29	27	27	28	29	30				31	25	26	27	28	29	30	31	35	22	23	24	25	26	27	28	
23	30	31																						36	29	30	31					
		T	l Tháng	Chín 2	2011					T	háng	Mười :	2011					Thá	ing M	ười Mớ	et 2011					The	áng M	lười Ha	i 2011			
	Hai	Ba	Tư	Năm	Sáu	Bảy	CN		Hai	Ba	Tư	Năm	Sáu	Bảy	CN		Hai	Ba	Tư	Năm	Sáu	Bảy	CN		Hai	Ba	Tư	Năm	Sáu	Bảy	CN	
36				1	2	3	4	40						1	2	45		1	2	3	4	5	6	49				1	2	3	4	
37	5	6	7	8	9	10	11	41	3	4	5	6	7	8	9	46	7	8	9	10	11	12	13	50	5	6	7	8	9	10	11	
38	12	13	14	15	16	17	18	42	10	11	12	13	14	15	16	47	14	15	16	17	18	19	20	51	12	13	14	15	16	17	18	
39	19	20	21	22	23	24	25	43	17	18	19	20	21	22	23	48	21	22	23	24	25	26	27	52	19	20	21	22	23	24	2	
40 l	26	27	28	29	30			44	24	25	26	27	28	29	30	49	28	29	30					53	26	27	28	29	30	31		
								45	31															53								

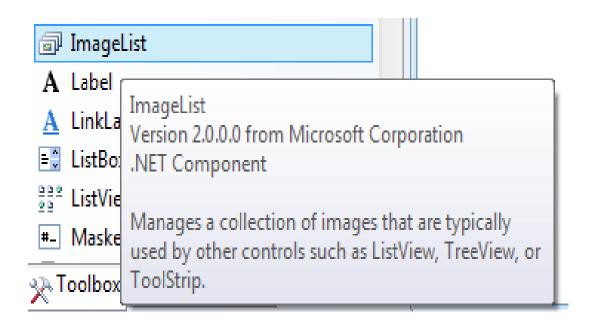
TIMER

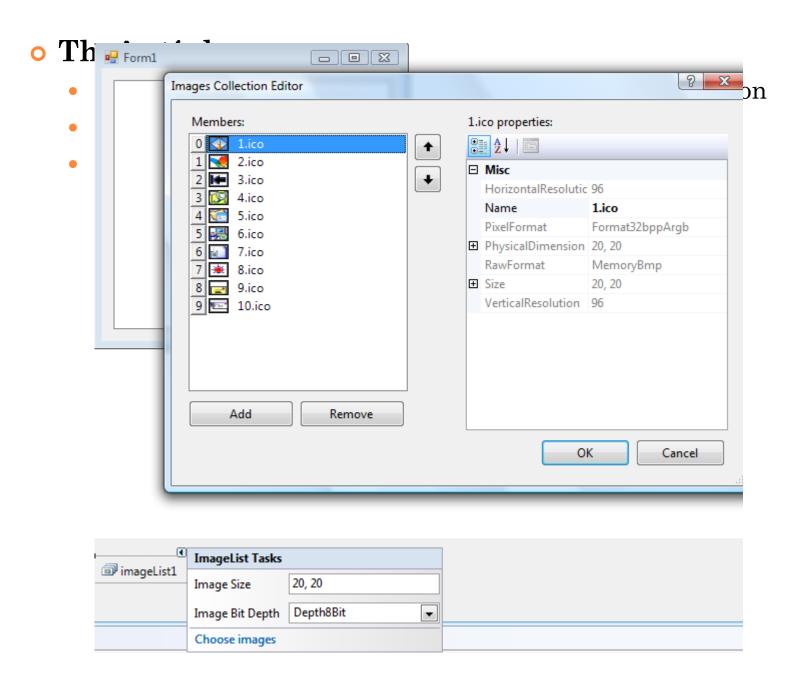
- Dùng để hiện thị cũng như tính toán thời gian trong Window form
- Thuộc tính:
 - Enable: xác định Timer đang chạy
 - Interval: xác định bằng ms giữa các khoảng thời gian đánh dấu
- Method:
 - Start/Stop: bắt đầu hay dừng Timer
- Event:
 - Tick: Khi giờ được trôi qua

```
public string GetTime() {
    string TimeInString = "";
    int hour = DateTime.Now.Hour;
   int min = DateTime.Now.Minute;
    int sec = DateTime.Now.Second;
    TimeInString = (hour < 10) ? "0" + hour.ToString() : hour.ToString();
    TimeInString +=":"+((min <10)?"0"+min.ToString ():min.ToString ());
    TimeInString += ":" + ((sec < 10) ? "0" + sec.ToString() : sec.ToString());
    return TimeInString;
public void Timer Tick(object sender, EventArgs eArgs)
    if (sender == timer1)
       label1.Text = GetTime();
    }
                                   - X
}
                         Form1
```

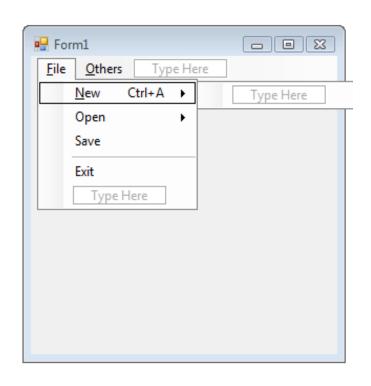
MENUES

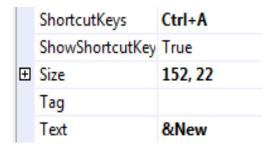
• ImageList: Quản lý Image cho các điều khiển khác như: ListView, TreeView hay ToolBar.





• MenuStrip: dùng để thiết kế Menu trên Form với các điều khiển ToolStripMenuItem- menu con, ToolStripComboBox, ToolStripSeparator- gạch phân tách và ToolStripTextBox- TextBox

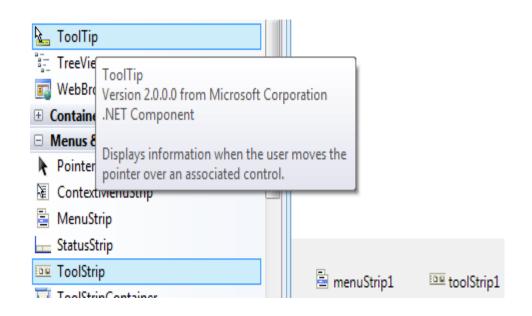


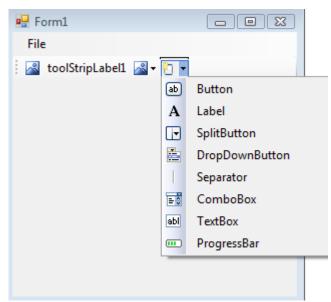


• Thuộc tính:

- AllowItemReorder: sắp xếp các phần tử trên Menu khi nhấn Alt
- *TextDirection*: chọn hình thức trình bày: nằm ngang, thẳng đứng hay ngược
- Items: chứa danh sách các menu khác như ToolStripMenuItem, ToolStripComboBox...
- RightToLeft: "No" trình bày từ trái sang phải.

• ToolStripMenuItem: trình bày các tùy chọn trên MenuStrip





• Event :

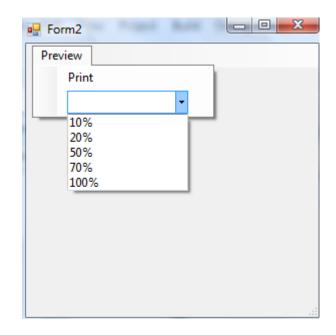
- CheckedChange: khi trạng thái của CheckBox thay đổi.
- Click: Khi click vào Control
- **DoubleClick**: Khi click hai lần vào Control.

Ví dụ khai báo và tạo đối tượng của Control:

```
void CreateControls()
{
    MenuStrip ms = new MenuStrip();
    ToolStripMenuItem mm = new ToolStripMenuItem();
    mm.Text = "&File";
    mm.Click += new System.EventHandler(ToolStripMenuItem1_Click);
    ms.Items.Add(mm);
    ms.Items.Add("&Edit");
    ms.Items.Add("&View");
    ms.Items.Add("&Windows");
    this.Controls.Add(ms);
}
```

• ToolStripComboBox: Cho phép trình bày menu như một dạng ComboBox.

```
ToolStripTextBox tt = new ToolStripTextBox();
tt.Text = "*.*";
ToolStripButton bt = new ToolStripButton();
bt.Text="Go";
ToolStripComboBox tc = new ToolStripComboBox();
tc.Text = "Folder";
tc.Items.Add("Folder");
tc.Items.Add("File");
ts.Items.Add(tt);
ts.Items.Add(bt);
ts.Items.Add(tc);
this.Controls.Add(ts);
```



• Thuộc tính :

- **DropDownStyle:** hình thức trình bày danh sách phần tử.
- MaxDropDownListItem: số lượng phần tử cần thêm, mặc định là 8 phần tử.
- Items: danh sách các phần tử trong Control
- **SelectedItems**: Phần tử được chọn trên control
- SelectedText: Text của item được chọn trên control
- **TooltipText**: Chuỗi Tip xuất hiện khi di chuột vào thành phần của control.

• ContextMenuStrip: thiết kế dạng pop-up cho phép xuất hiện điều khiển *ToolStripMenuItem* dạng shortcut khi chuột phải vào Control hay Form.

ContextMenuStrip

StatusStrip

ToolStrip

ToolStripContair

Data

Displays a shortcut menu when the user right-clicks the associated control.

Components

Dialogs

ContextMenuStrip

Version 2.0.0.0 from Microsoft Corporation

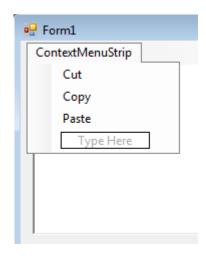
NET Component

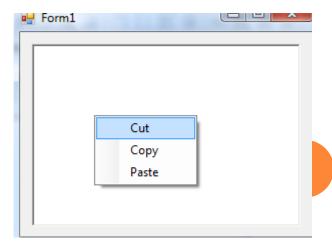
Displays a shortcut menu when the user right-clicks the associated control.

Components

Components

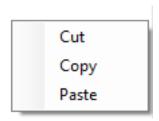
ContextMenuStrip1

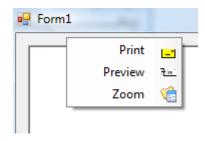




o Thuộc tính:

- **Items**: danh sách các điều khiển như *ToolStripMenuItems, ToolStripComboBox...*
- RightToLeft: Trình bày menu từ trái sang phải.
- ShowImageMargin: Che giấu hay trình bày hình ảnh.
- Ví dụ: tạo hai control cho Form và RichTextBox
 - Khi R_Click vào Form
 - R_Click vào TextBox





```
private void toolStripMenuItem3_Click(object sender, EventArgs e)
{
    richTextBox1.Text= Clipboard.GetText();
}
void CreateControls()
{

    ContextMenuStrip cms = new
    ContextMenuStrip();
    cms.Name = "Shortcut";
    cms.Items.Add("Copy");
    cms.Items.Add("Cut");
    cms.Items.Add("Paste");
    this.Controls.Add(cms);
}
```

