
Common Dialog & User Control

Nguyễn Văn Phong

Nội Dung

- **Giới thiệu Common Dialog**
- OpenFileDialog
- SaveFileDialog
- FontDialog
- ColorDialog
- MessageBox
- User Control



Common Dialog

- Các dialog thường được sử dụng
 - Được warp thành các lớp trong FCL/BCL

OpenFileDialog

ColorDialog

**Common
Dialog**

SaveFileDialog

FontDialog

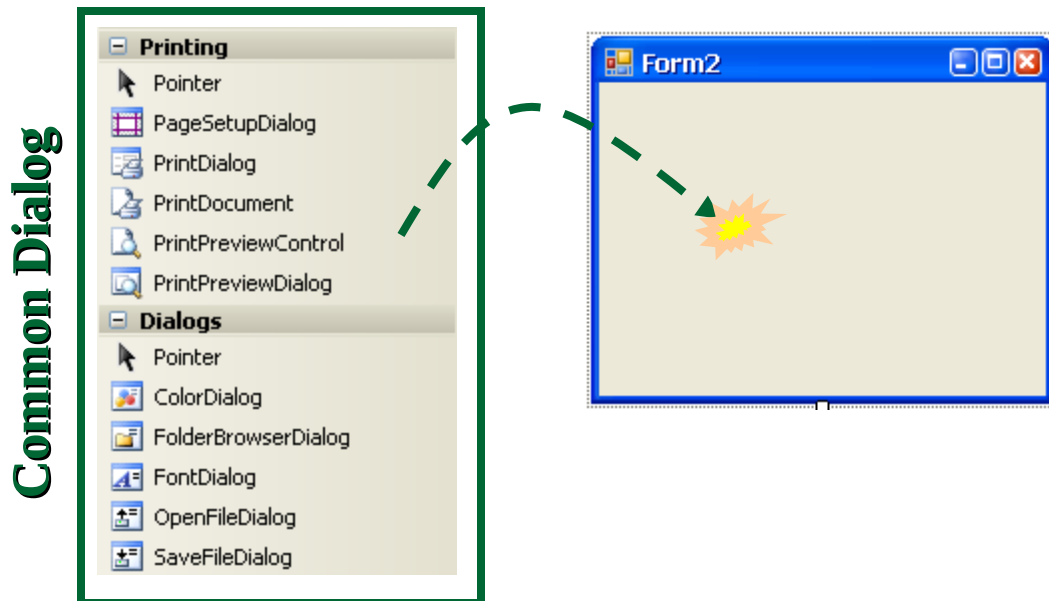
PrintDialog

.....

Common Dialog

■ Cách sử dụng

- 1. Kéo thả dialog vào form
 - VS sẽ tạo thể hiện chứa trong lớp form
- 2. Khai báo đối tượng và tạo thể hiện của lớp CD
 - VD: `OpenFileDialog oFile = new OpenFileDialog();`



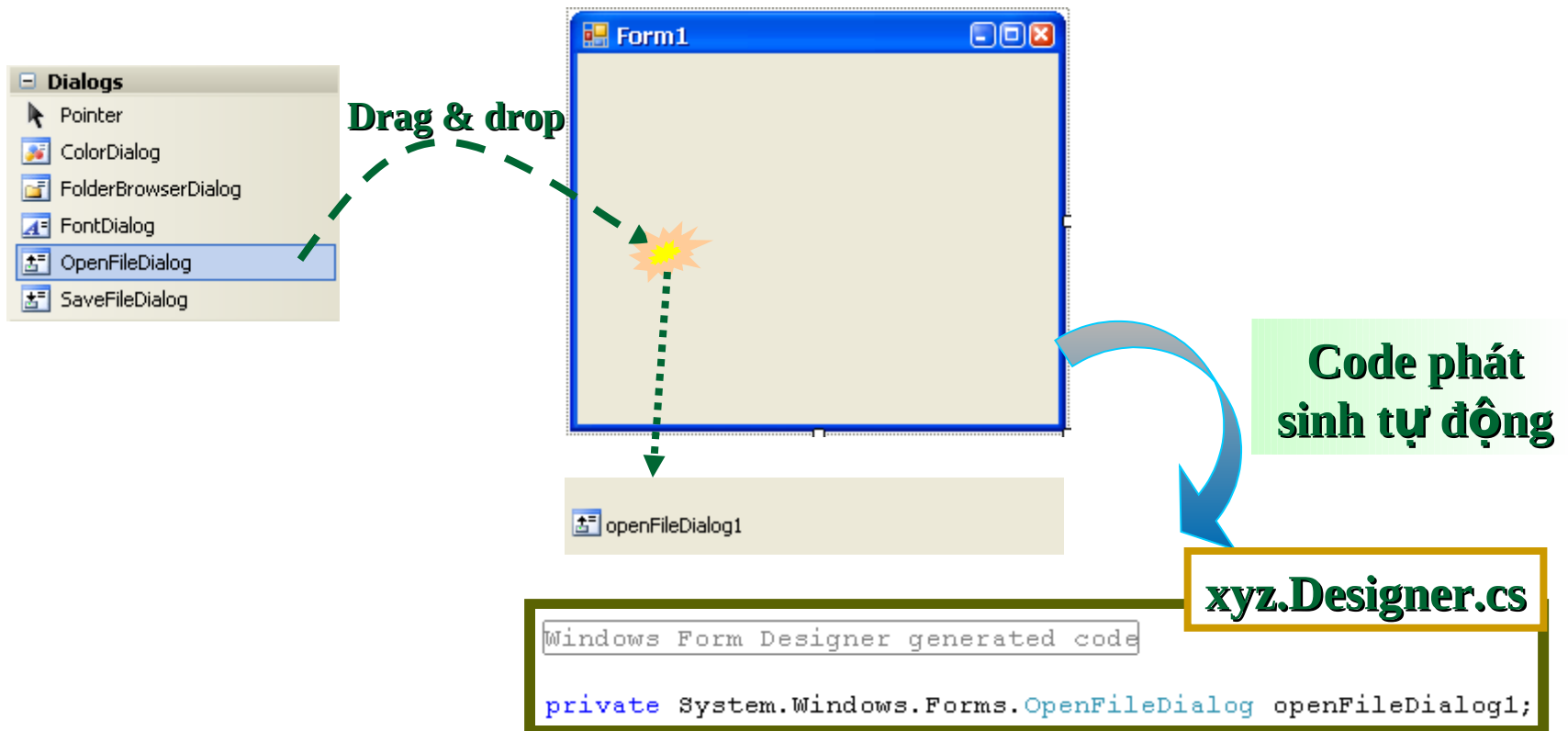
Nội Dung

- Giới thiệu Common Dialog
- **OpenFileDialog**
- SaveFileDialog
- FontDialog
- ColorDialog
- MessageBox
- User Control



OpenFileDialog

- Sử dụng để chọn file lưu trên đĩa
- Cách sử dụng từ Toolbox



OpenFileDialog

Code phát sinh của VS

xyz.Designer.cs

```
private void InitializeComponent()
{
    this.openFileDialog1 = new System.Windows.Forms.OpenFileDialog();
    this.SuspendLayout();
    //
    // openFileDialog1
    //
    this.openFileDialog1.FileName = "openFileDialog1";
    //
    // Form1
    //
    this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
    this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
    this.ClientSize = new System.Drawing.Size(262, 202);
    this.Name = "Form1";
    this.Text = "Form1";
    this.Load += new System.EventHandler(this.Form1_Load);
    this.ResumeLayout(false);
}

#endregion

private System.Windows.Forms.OpenFileDialog openFileDialog1;
```

Tạo thể hiện

Hiển thị Form

openFileDialog1.ShowDialog();

OpenFileDialog

- Code tự viết – không dùng Toolbox

```
// tạo thể hiện OpenFileDialog
OpenFileDialog OFileDlg = new OpenFileDialog();
// thiết lập các option
// cho phép chọn nhiều file
OFileDlg.Multiselect = true;
// lọc chỉ mở những file C# source code
OFileDlg.Filter = "C# source code|*.cs";

DialogResult result;
// hiển thị và lấy giá trị trả về
result = OFileDlg.ShowDialog();
// xử lý tiếp theo...
```



OpenFileDialog

Thuộc tính, phương thức

Thuộc tính

FileName	Lấy tên file được chọn
FileNames	Lấy tên tất cả các file được chọn
Filter	Xác định kiểu file cần mở
InitialDirectory	Thư mục khởi tạo
Multiselect	Cho phép chọn nhiều file
Title	Tiêu đề của dialog

Phương thức

ShowDialog	Hiển thị dialog
------------	-----------------

Sự kiện

FileOk Xuất hiện khi user kích vào OK



Nội Dung

- Giới thiệu Common Dialog
- OpenFileDialog
- **SaveFileDialog**
- FontDialog
- ColorDialog
- MessageBox
- User Control



SaveFileDialog

- Sử dụng để tạo file trên đĩa.
- Cách sử dụng
 - Sử dụng SaveFileDialog component trên Toolbox
 - Tương tự như OpenFileDialog!
 - Tạo thể hiện của lớp SaveFileDialog

```
// tạo thể hiện Save File Dialog
SaveFileDialog saveDlg = new SaveFileDialog();
// save dưới dạng txt
saveDlg.Filter = "Text File|*.txt";
// hiển thị dialog và lấy giá trị trả về của dialog
DialogResult result = saveDlg.ShowDialog();
// nếu user chọn button "OK"
if (result == DialogResult.OK)
{
    // TO DO...
}
```



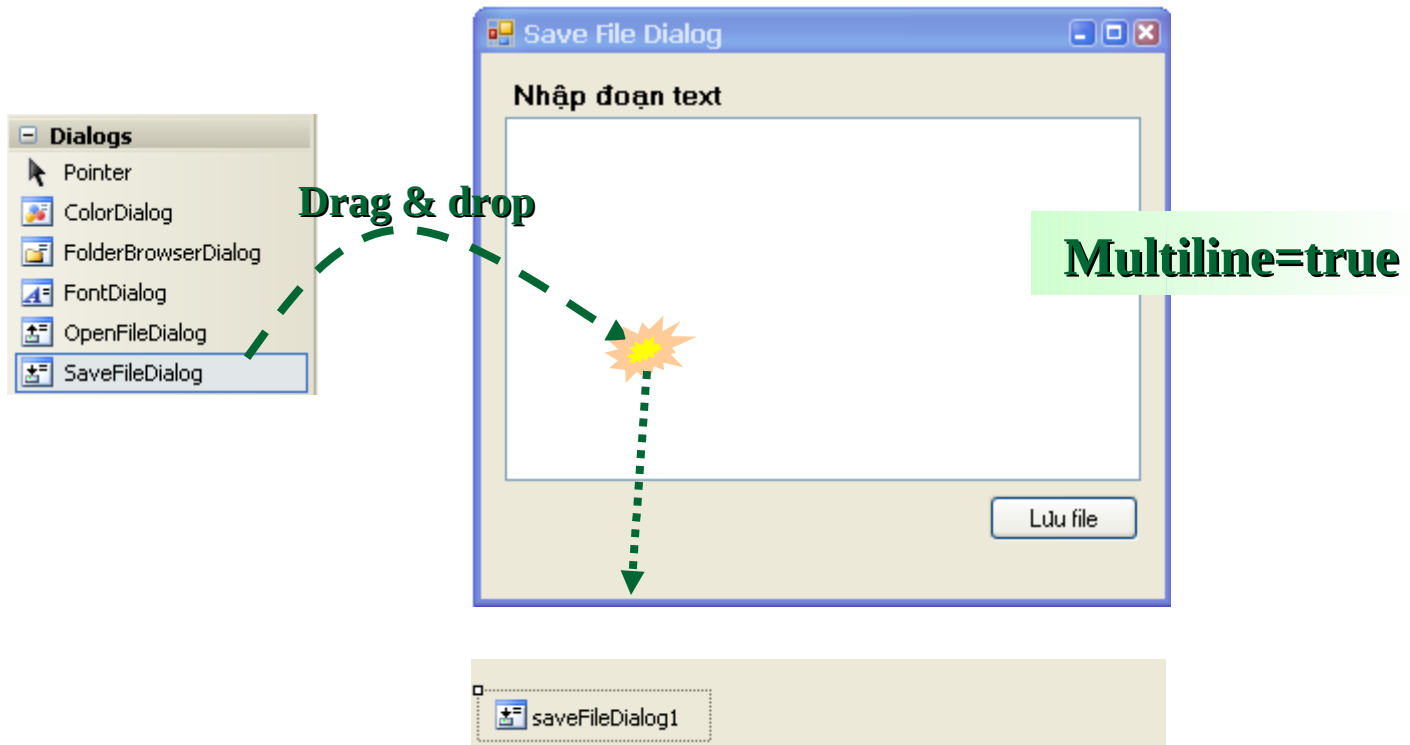
SaveFileDialog

- Demo: nhập văn bản vào textbox, sau đó lưu xuống file *.txt.
 - Tạo ứng dụng dạng Windows Form có các control
 - 1 label: caption của textbox
 - 1 textbox: chứa nội dung text do user nhập
 - 1 button: gọi SaveFileDialog và lưu file
 - 1 SaveFileDialog: khai báo dialog SaveFile.



SaveFileDialog

- Thiết kế form



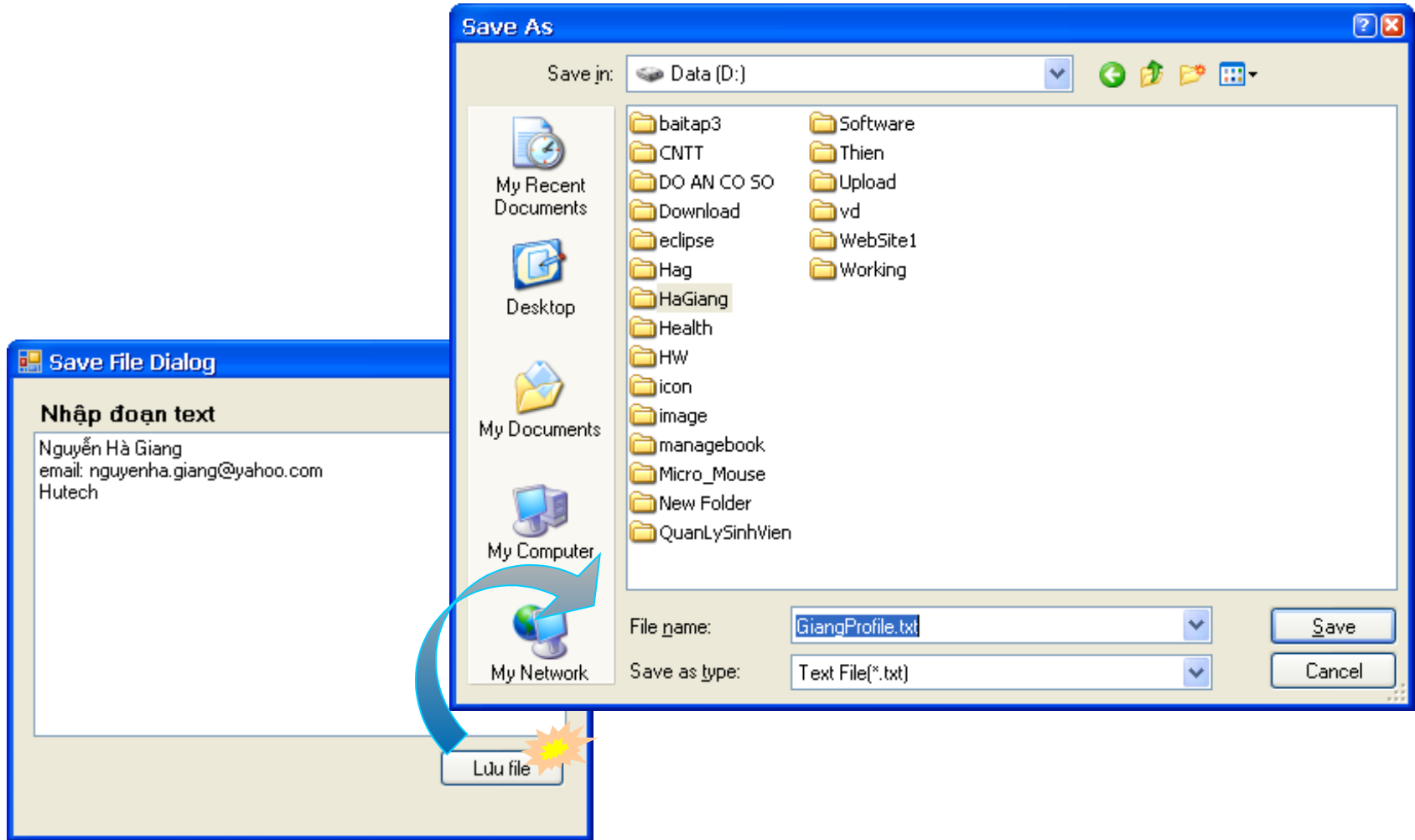
SaveFileDialog

- Viết phần xử lý cho button “Lưu file”

System.I

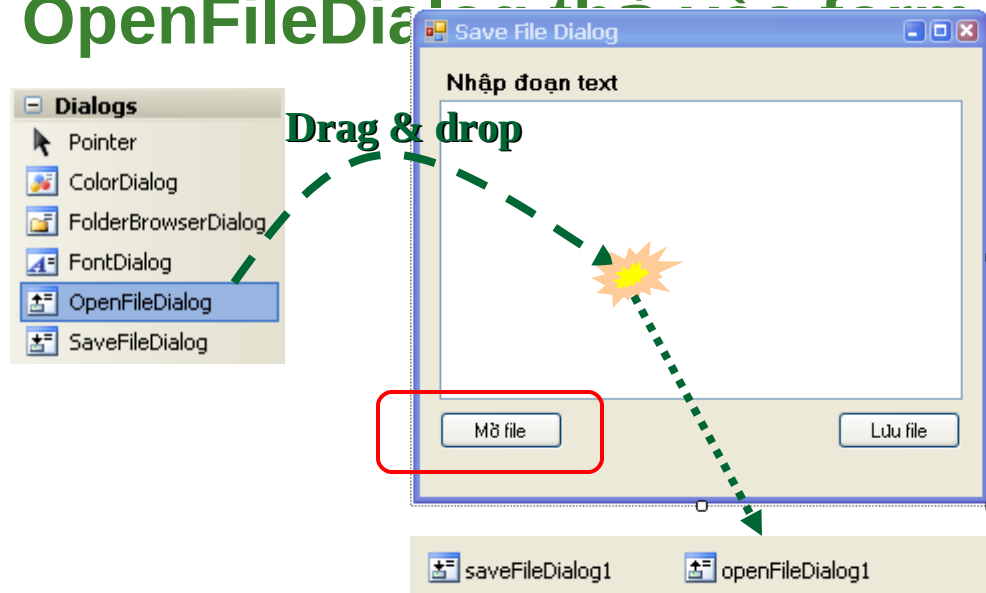
```
21 // phần xử lý cho button lưu
22 // lưu dạng text
23 saveFileDialog1.Filter = "Text File(*.txt)|*.txt";
24
25 if (saveFileDialog1.ShowDialog() == DialogResult.OK)
26 {
27     // sau khi user nhập tên file cần lưu và nhấn "OK"
28
29     // tạo stream để đọc ghi file
30     Stream myStream = saveFileDialog1.OpenFile();
31     // tạo đối tượng ghi vào file
32     StreamWriter writer = new StreamWriter(myStream);
33     // lưu nội dung text của textbox1 vào file
34     writer.Write(textBox1.Text);
35
36     writer.Close();
37     myStream.Close();
38 }
```

SaveFileDialog



SaveFileDialog

- Mở rộng bổ sung phần đọc file text từ demo trên
 - Open file *.txt và hiển thị nội dung của file trên TextBox.
- Bổ sung thêm button “Mở file” và kéo thành phần OpenFileDialog



SaveFileDialog

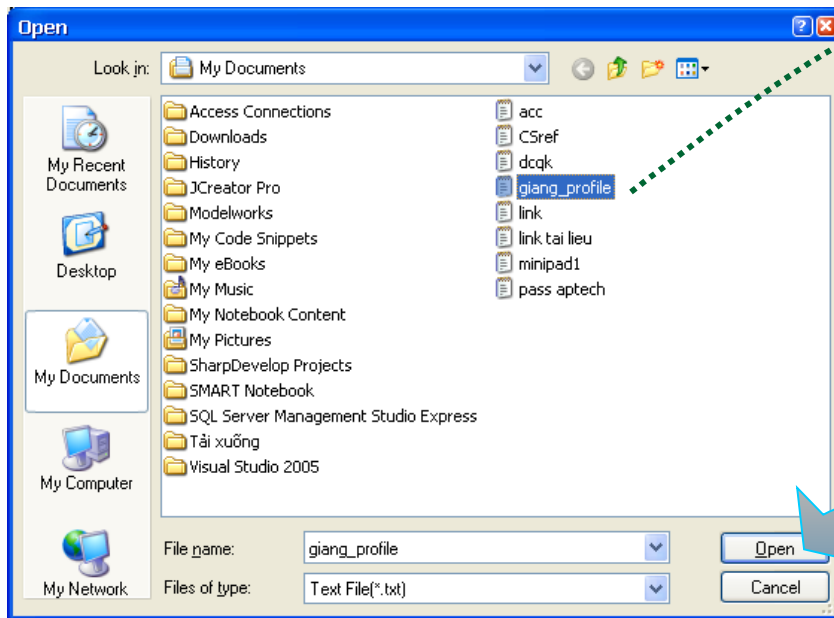
- Viết phần xử lý button “Mở file”

```
43 // chức năng mở file txt và hiển thị nội dung
44 openFileDialog1.Filter = "Text File (*.txt)|*.txt";
45
46 if (openFileDialog1.ShowDialog() == DialogResult.OK)
47 {
48     // tạo stream đọc/ghi file
49     Stream myStream = openFileDialog1.OpenFile();
50     // tạo đối tượng đọc file
51     StreamReader reader = new StreamReader(myStream);
52     // đọc toàn bộ nội dung vào textbox1
53     textBox1.Text = reader.ReadToEnd();
54
55     reader.Close();
56     myStream.Close();
57 }
```

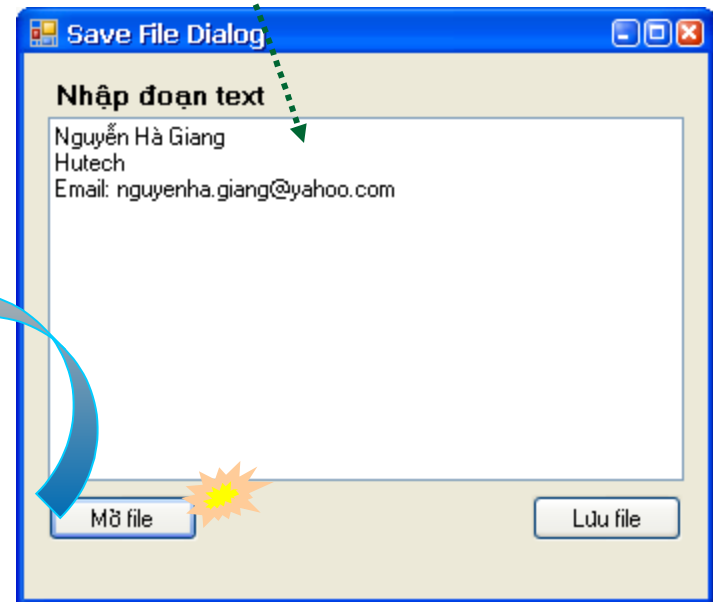


SaveFileDialog

■ Demo



Hiển thị nội
dung file



Nội Dung

- Giới thiệu Common Dialog
- OpenFileDialog
- SaveFileDialog
- **FontDialog**
- ColorDialog
- MessageBox
- User Control



FontDialog

- Chức năng hiển thị hộp thoại chọn font chữ được install trong máy
 - Trong ứng dụng làm việc với document, đồ họa...
- Sử dụng FontDialog
 - Từ ToolBox kéo thả FontDialog vào Form
 - Sử dụng trong lúc thiết kế
 - Khai báo thể hiện FontDialog và hiển thị
 - Viết code



FontDialog

Thuộc tính, phương thức thường dùng

Thuộc tính

Font	Get/set font được chọn
Color	Get/set màu của font
ShowColor	Hiển thị hay không hiển thị chọn màu
ShowApply	Hiển thị/ không button Apply

Phương thức

ShowDialog	Hiển thị dialog ra màn hình
------------	-----------------------------

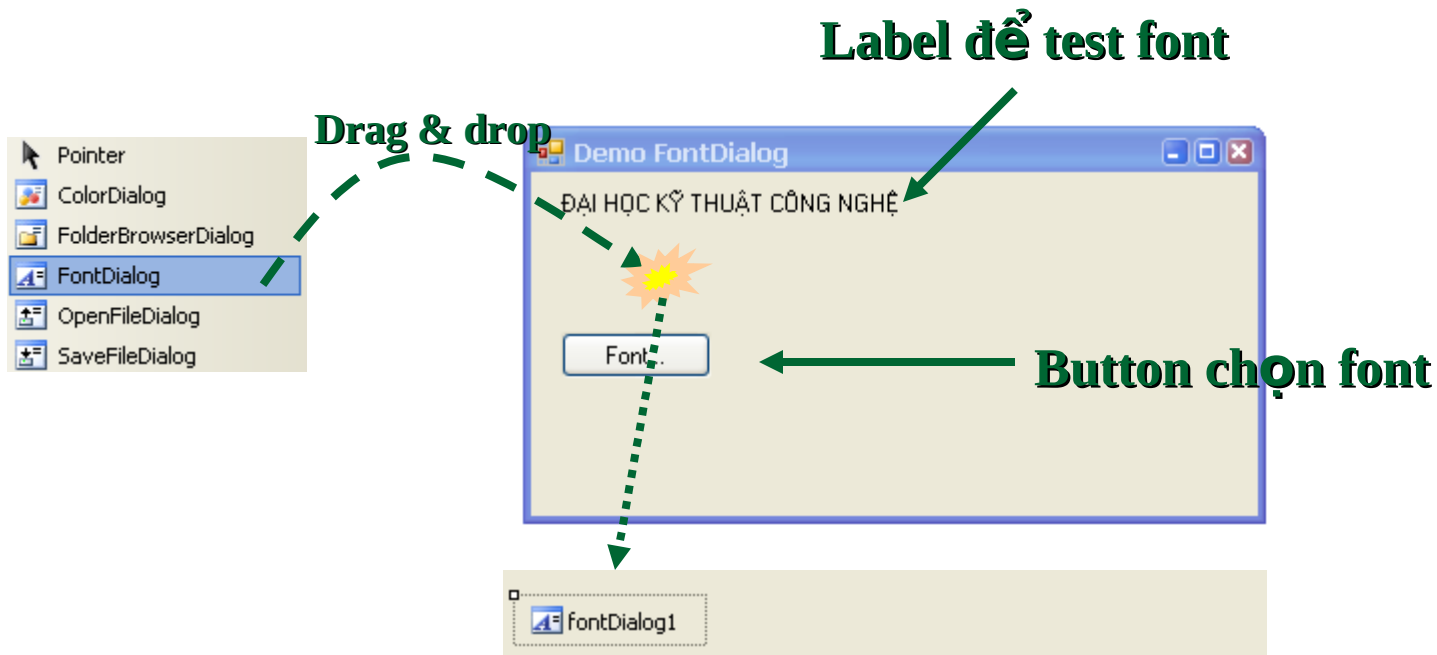
Sự kiện

Apply	Kích hoạt khi user chọn apply
-------	-------------------------------



FontDialog

- Demo: gọi FontDialog thiết lập font cho control
- Tạo Form có dạng sau



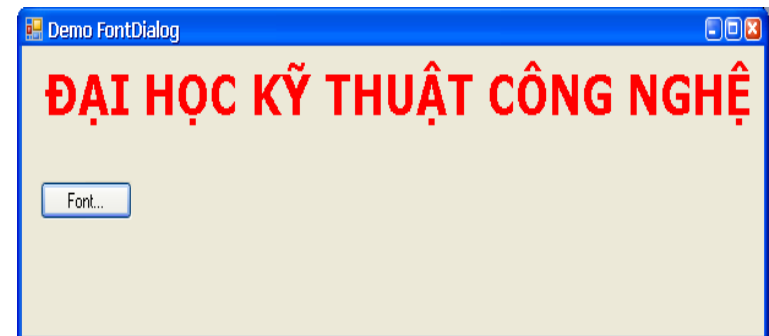
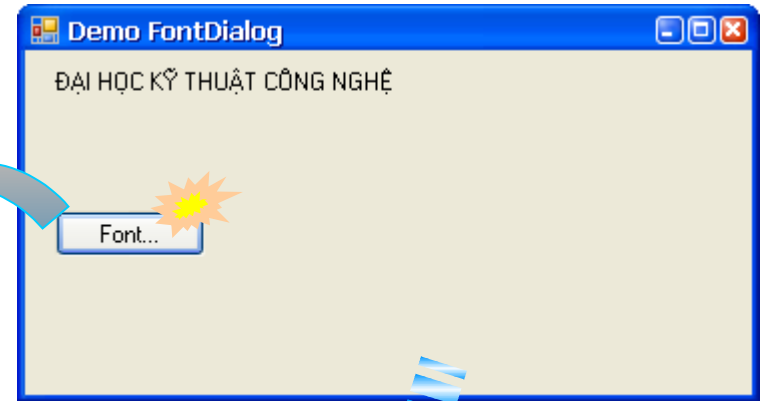
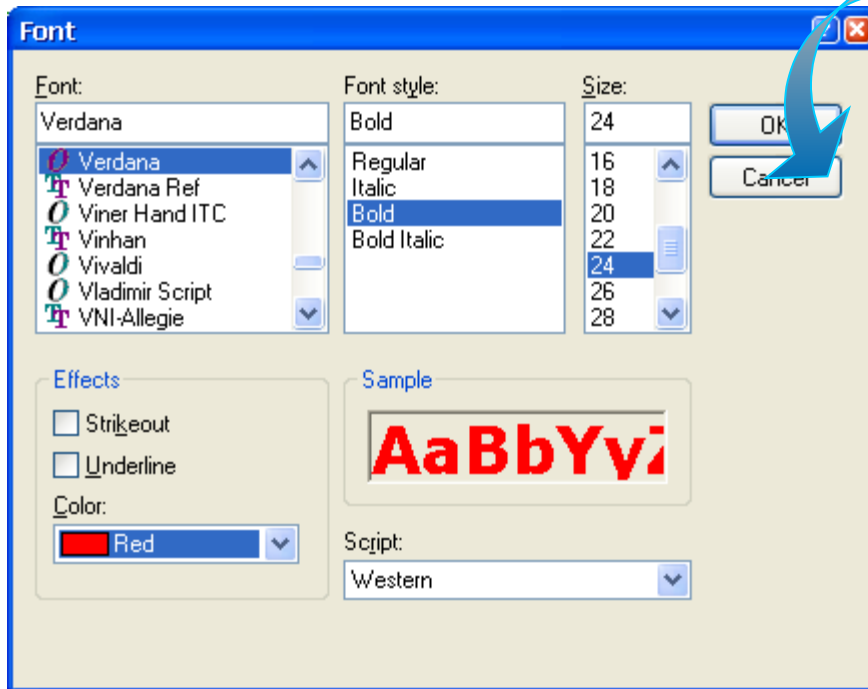
FontDialog

- Phần xử lý cho button “Font”

```
21 // hiển thị chức năng chọn màu
22 fontDialog1.ShowColor = true;
23 // chọn màu mặc định là màu label
24 fontDialog1.Color = label1.ForeColor;
25 // thiết lập font mặc định là font của label
26 fontDialog1.Font = label1.Font;
27
28 if (fontDialog1.ShowDialog() == DialogResult.OK)
29 {
30     // thiết lập font mới cho label
31     label1.Font = fontDialog1.Font;
32     // thiết lập màu mới cho label
33     label1.ForeColor = fontDialog1.Color;
34 }
```



FontDialog



FontDialog


- Bổ sung chức năng “Apply” của FontDialog
 - Khi FontDialog đang hiển thị cho phép user kích chọn nút Apply ⇒ label thay đổi font theo.
- Cách thực hiện
 - Khai báo có hiển thị button Apply cho FontDialog
 - `fontDialog1.ShowApply = true`
 - Đăng ký trình xử lý sự kiện cho button “Apply”
 - Tạo trình xử lý cho sự kiện “Apply” của FontDialog
 - Đăng ký trình xử lý cho sự kiện Apply của FontDialog



FontDialog

■ Code minh họa

```
28 // hiển thị button Apply
29 fontDialog1.ShowApply = true;
30
31 // đăng ký sự kiện khi button Apply được nhấn
32 fontDialog1.Apply += new EventHandler(ChangeFont);
```



```
46 void ChangeFont(object sender, EventArgs e)
47 {
48     // thiết lập font mới cho label
49     label1.Font = fontDialog1.Font;
50     // thiết lập màu mới cho label
51     label1.ForeColor = fontDialog1.Color;
52 }
```

Nội Dung

- Giới thiệu Common Dialog
- OpenFileDialog
- SaveFileDialog
- FontDialog
- **ColorDialog**
- MessageBox
- User Control



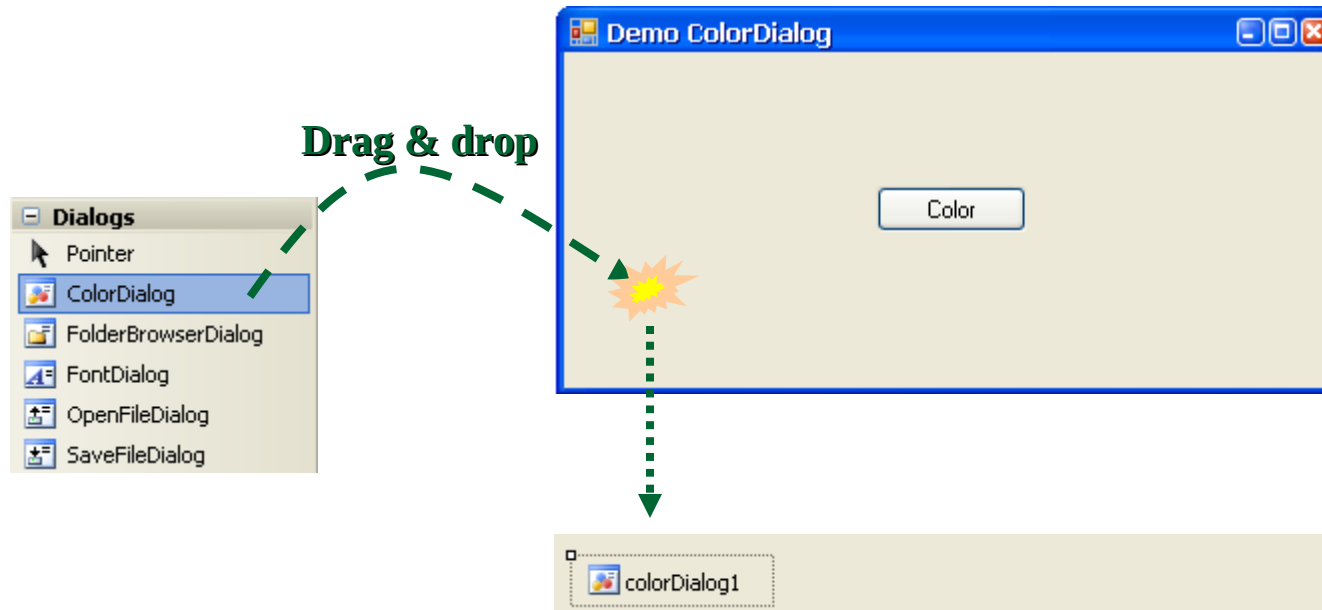
ColorDialog

- **Hiển thị chức năng chọn màu**
- **Sử dụng theo 2 cách**
 - Kéo ColorDialog từ Toolbox thả vào form...
 - Viết code tạo đối tượng...
- **ColorDialog sử dụng giá trị màu có kiểu Color**
 - **Màu ARGB**
 - Alpha
 - Red
 - Green
 - Blue



ColorDialog

- Demo chọn màu: chọn màu nền cho form



ColorDialog

■ Phần xử lý cho button “Color”

```
if (colorDialog1.ShowDialog() == DialogResult.OK)
{
    // lấy bề mặt vẽ của form
    Graphics g = this.CreateGraphics();
    // hình chữ nhật của form
    Rectangle rect = new Rectangle(0,0,Width,Height);
    // tạo brush gradient: có màu được chọn -> white
    LinearGradientBrush brush = new LinearGradientBrush(rect,
        colorDialog1.Color, Color.White,
        LinearGradientMode.Horizontal);
    // tô nền của form
    g.FillRectangle(brush, rect);
}
```



Nội Dung

- ❑ Giới thiệu Common Dialog
- ❑ OpenFileDialog
- ❑ SaveFileDialog
- ❑ FontDialog
- ❑ ColorDialog
- ❑ **MessageBox**
- ❑ User Control



MessageBox

- **Hiển thị hộp thoại chứa thông tin chỉ dẫn đến user**
- **Bao gồm các phần**
 - Text
 - Button
 - Symbol
- **Lớp MessageBox có phương thức tĩnh Show để hiển thị dialog.**
 - **Bắt giá trị trả về để biết phản ứng của user**



MessageBox

- Có khoảng 21 phiên bản của Show
- Một số phiên bản thường sử dụng
 - DialogResult Show(string text);
 - DialogResult Show(string text, string caption);
 - DialogResult Show(string text, string caption, MessageBoxButtons button);
 - DialogResult Show(string text, string caption, MessageBoxButtons button, MessageBoxIcon icon);
 - ...



MessageBox





- Các button hiển thị theo message
 - Cho phép user chọn lựa các phản ứng với message
 - Được định nghĩa trong MessageBoxButtons

OK	MessageBoxButtons
OKCancel	
AbortRetryIgnore	
YesNoCancel	
YesNo	
RetryCancel	



MessageBox

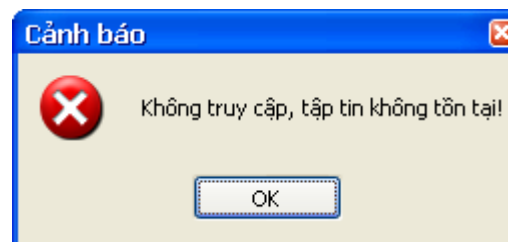
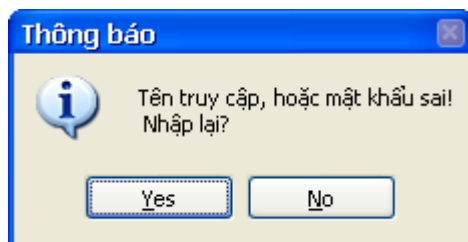
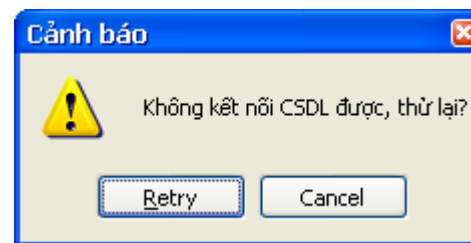
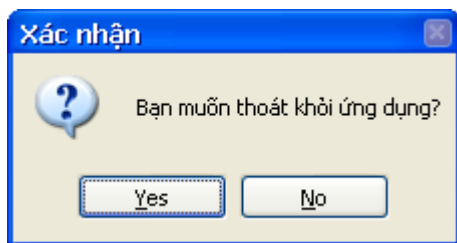
- **MessageBoxIcon**: định nghĩa các icon kèm theo message

	MessageBoxIcon.Asterisk MessageBoxIcon.Information
	MessageBoxIcon.Error MessageBoxIcon.Stop MessageBoxIcon.Hand
	MessageBoxIcon.Exclamation MessageBoxIcon.Warning
	MessageBoxIcon.Question
	MessageBoxIcon.None



MessageBox

- Một số các MessageBox minh họa



Nội Dung

- ❑ Giới thiệu Common Dialog
- ❑ OpenFileDialog
- ❑ SaveFileDialog
- ❑ FontDialog
- ❑ ColorDialog
- ❑ MessageBox
- ❑ **User Control**



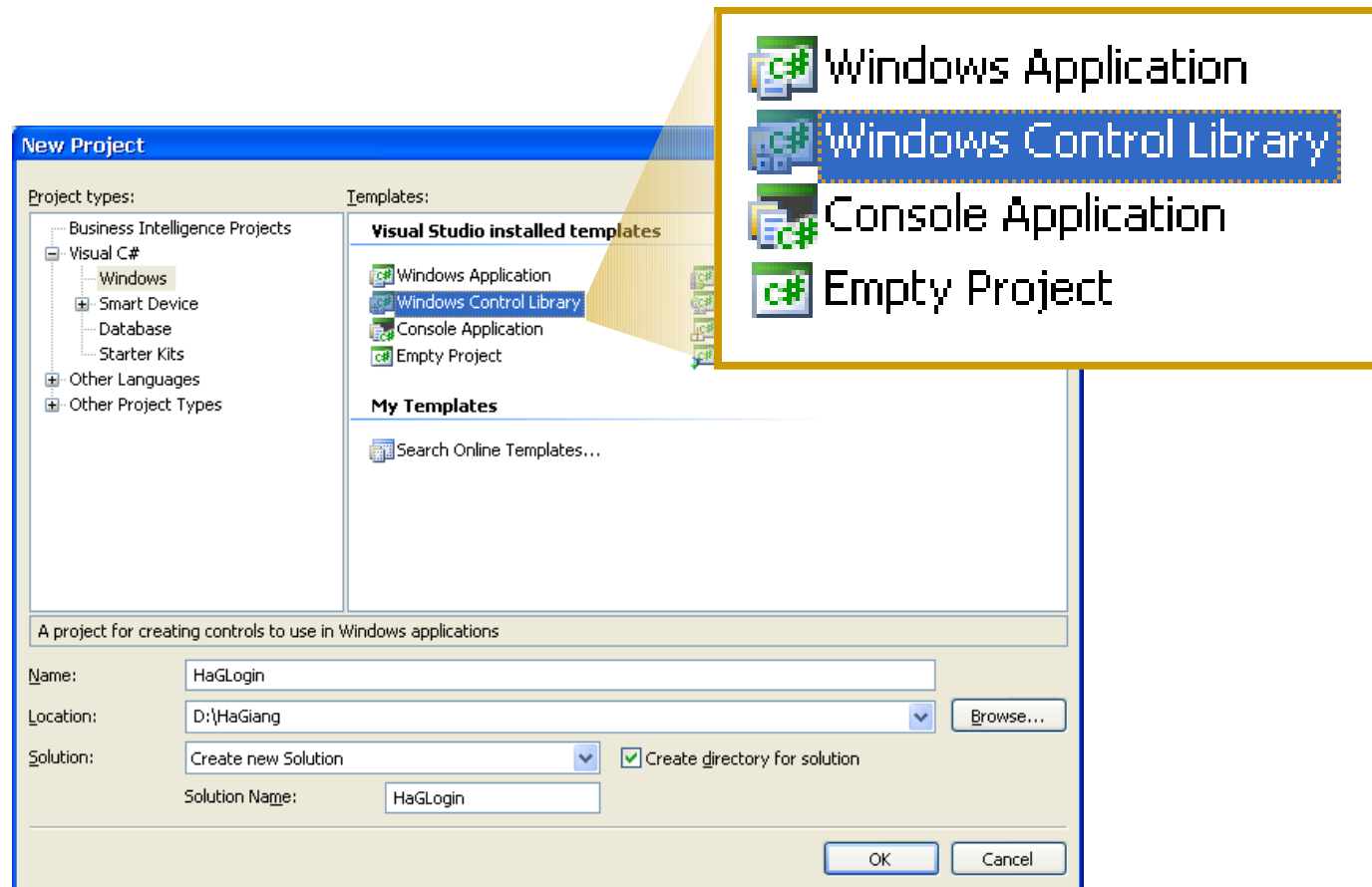
User Control

- Công việc kéo thả nhóm control thường sử dụng ⇒ nhàm chán
- User control phục vụ mục đích tái sử dụng
- Vấn đề:
 - Nhóm control phục vụ cho **việc login** thường xuất hiện trong ứng dụng
- Yêu cầu:
 - Tạo nhóm control phục vụ cho việc login
 - Nhóm control này bao gồm
 - 2 label
 - 2 textbox



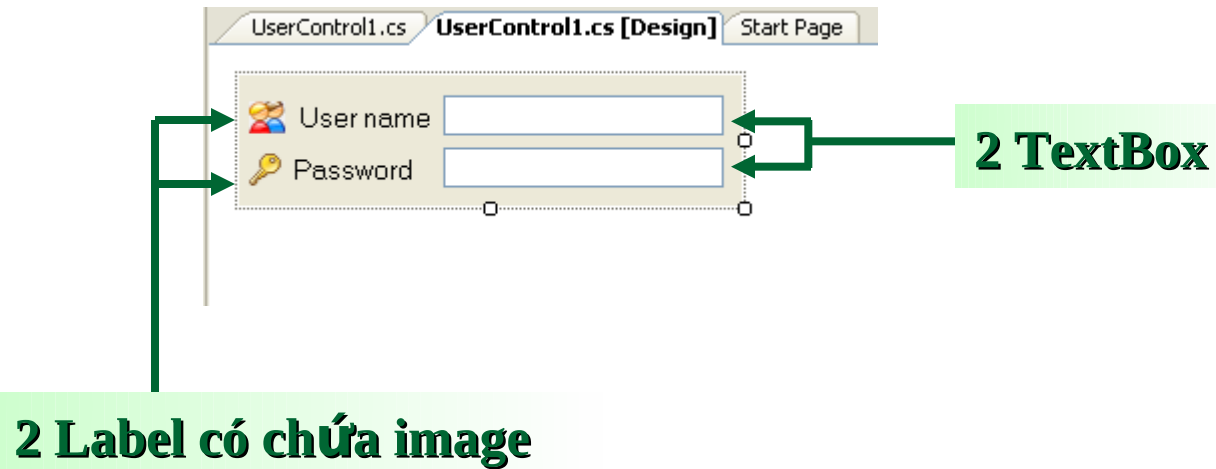
User Control

■ Bước 1: Tạo ứng dụng Windows Control Library



User Control

- **Bước 2: thiết kế user control như sau**



User Control

- Khai báo các property cho user control
 - Mỗi property gắn với textBox
 - UserName gắn với txtUID
 - Password gắn với txtPwd

```
[Category("Data"), Description("User Name")]
public string UserName
{
    get { return txtUID.Text; }
    set { txtUID.Text = value; }
}
[Category("Data"), Description("Password")]
public string Password
{
    get { return txtPwd.Text; }
    set { txtPwd.Text = value; }
}
```



User Control

- Biên dịch user control
- Kết quả được file assembly có phần mở rộng là DLL
- Sử dụng user control trong client app
 - Add vào phần References của project
 - Tạo đối tượng user control, add vào form...
 - Add vào ToolBox | Choose Items...
 - Kéo user control thả vào form, code phát sinh tự động



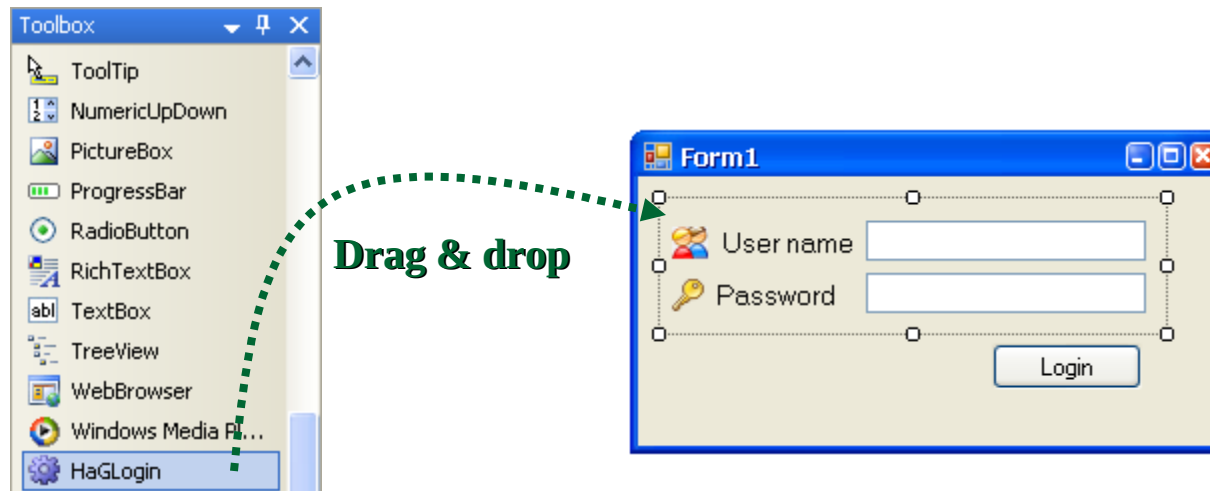
User Control

- Tạo ứng dụng Test User control
 - Tạo ứng dụng Windows Form
 - Thêm User Control vào ToolBox
 - Kích chuột phải vào ToolBox
 - Chọn chức năng Choose Items...
 - Chọn file DLL của User control vừa tạo
 - Kéo user control thả vào form



User Control

- Kéo thả user control vào form

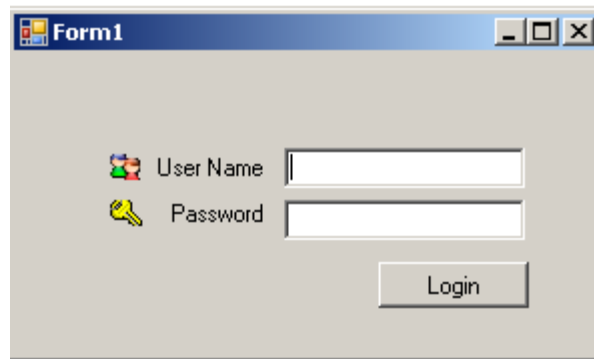


**Phát sinh code
trong designer.cs**

```
private Login.UserControl1 userControl11;  
private void InitializeComponent()  
{  
    this.userControl11 = new Login.UserControl1();  
    this.button1 = new System.Windows.Forms.Button();  
    this.SuspendLayout();  
}
```

User Control

- Sử dụng User Control như control bình thường trên form.
- Truy cập user control Login thông qua 2 property đã định nghĩa khi xây dựng control này
 - Username: là textBox User Name
 - Password: là nội dung của textBox Password



Tóm tắt

- **Common Dialog**
 - OpenFileDialog - SaveFileDialog
 - FontDialog - ColorDialog
- **MessageBox**
 - Khai báo và sử dụng
 - MessageBoxIcon
 - MessageBoxButtons
- **User control**
 - Xây dựng user control
 - Sử dụng user control

