



ASP.NET

GV: ThS Phạm Thi Vương

Contents



Click to add Title



Click to add Title



Click to add Title



Click to add Title



Click to add Title

Lợi ích của việc sử dụng ASP

- ❖ Công nghệ Server-side scripting
- ❖ Tự động biên dịch
- ❖ Cho phép tạo các ứng dụng web đơn giản nhanh chóng và dễ dàng
- ❖ Tạo trang web có tích hợp nội dung động

Bất lợi của ASP

- ❖ Chỉ hỗ trợ 2 loại ngôn ngữ: VBScript và JavaScript
- ❖ Thông dịch mã lệnh ASP
- ❖ Pha trộn code, HTML và text
- ❖ Tương thích trình duyệt
- ❖ Không quản lý trạng thái trang web
- ❖ Cơ chế debug kém
- ❖ Tái sử dụng code kém

Các điểm nổi bật của ASP.NET

- ❖ Hỗ trợ đa ngôn ngữ: C#, VB.NET,...
- ❖ Biên dịch các trang trước, giúp làm tăng tốc độ thực hiện
- ❖ ASP code được phân ra độc lập với HTML và text
- ❖ Quản lý trạng thái trang web
- ❖ Có cơ chế hỗ trợ debug
- ❖ Hỗ trợ tái sử dụng code thông qua cơ chế kế thừa

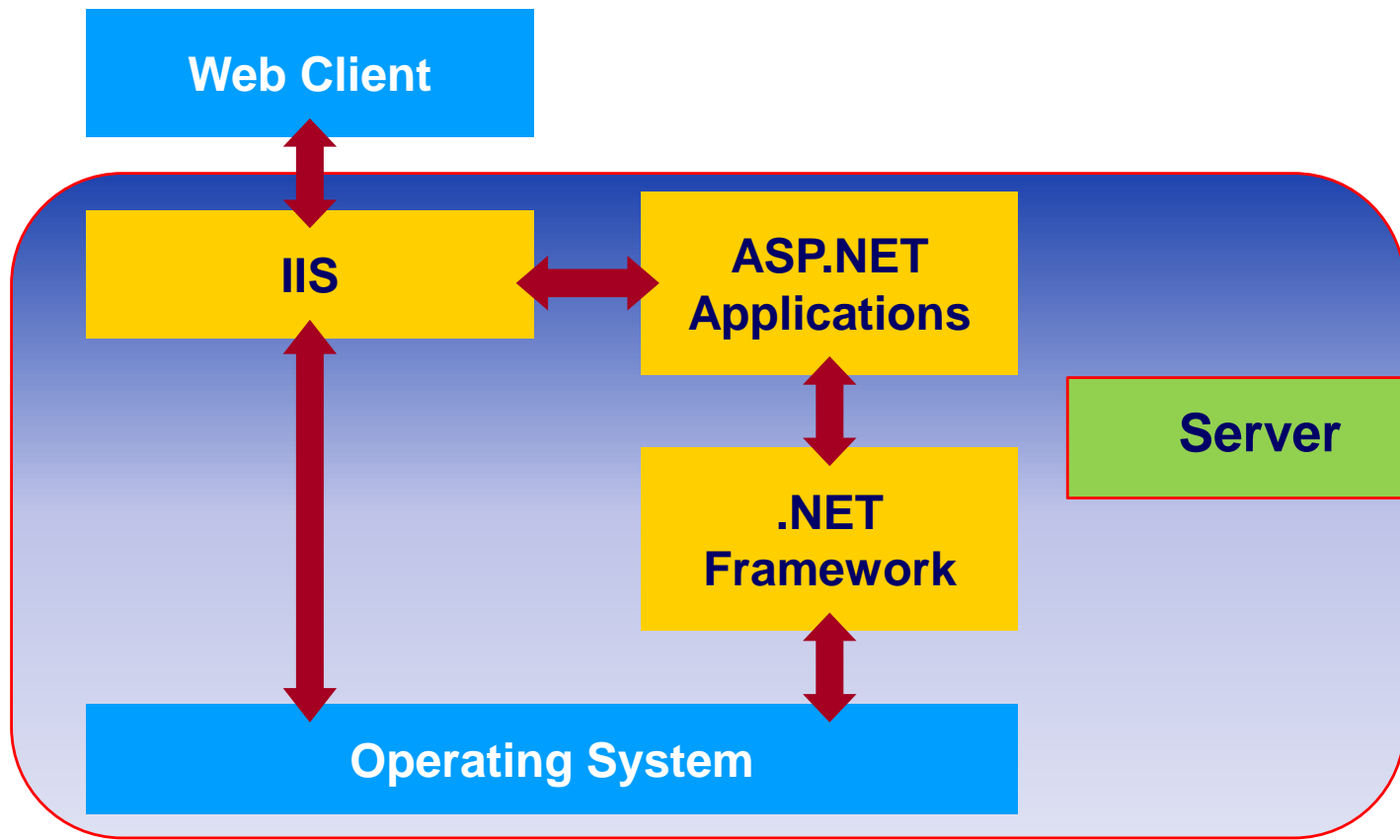
Các điểm nổi bật của ASP.NET

- ❖ Dễ dàng sử dụng C# hoặc VB.NET
- ❖ Sử dụng cơ chế server-side caching
- ❖ Tự động nhận dạng trình duyệt người dùng đang sử dụng
- ❖ Đi cùng với nhiều server control được xây dựng sẵn
- ❖ Global.asax hỗ trợ nhiều sự kiện hơn
- ❖ Web service : triển khai một hàm từ xa thông qua web

Giới thiệu về .NET Framework

- ❖ **.NET Framework là nền tảng cho Microsoft.NET Platform**
- ❖ **.NET Framework là môi trường cho việc xây dựng, triển khai và vận hành các ứng dụng Web cũng như Web Service**
- ❖ **.NET Framework chứa Common Language Runtime (CLR) và các lớp thư viện cung cấp các dịch vụ cơ sở để xây dựng các ứng dụng**

Cấu trúc một ứng dụng ASP.NET



Các thành phần của một ứng dụng Web

❖ Các file hiển thị thông tin

- Web Forms (.aspx), HTML, images, audio, video, ...

❖ Special Folder:

- App_Code, App_Data, App_Themes, ...

❖ Source code xử lý logic của chương trình

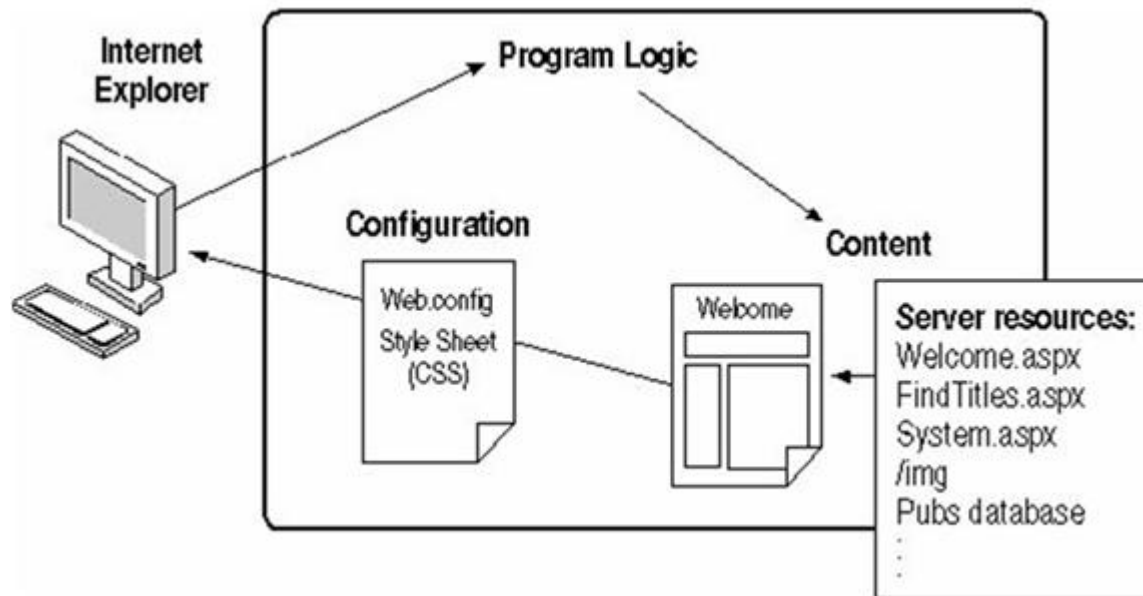
- .cs , .vb, ...

❖ Các file cấu hình

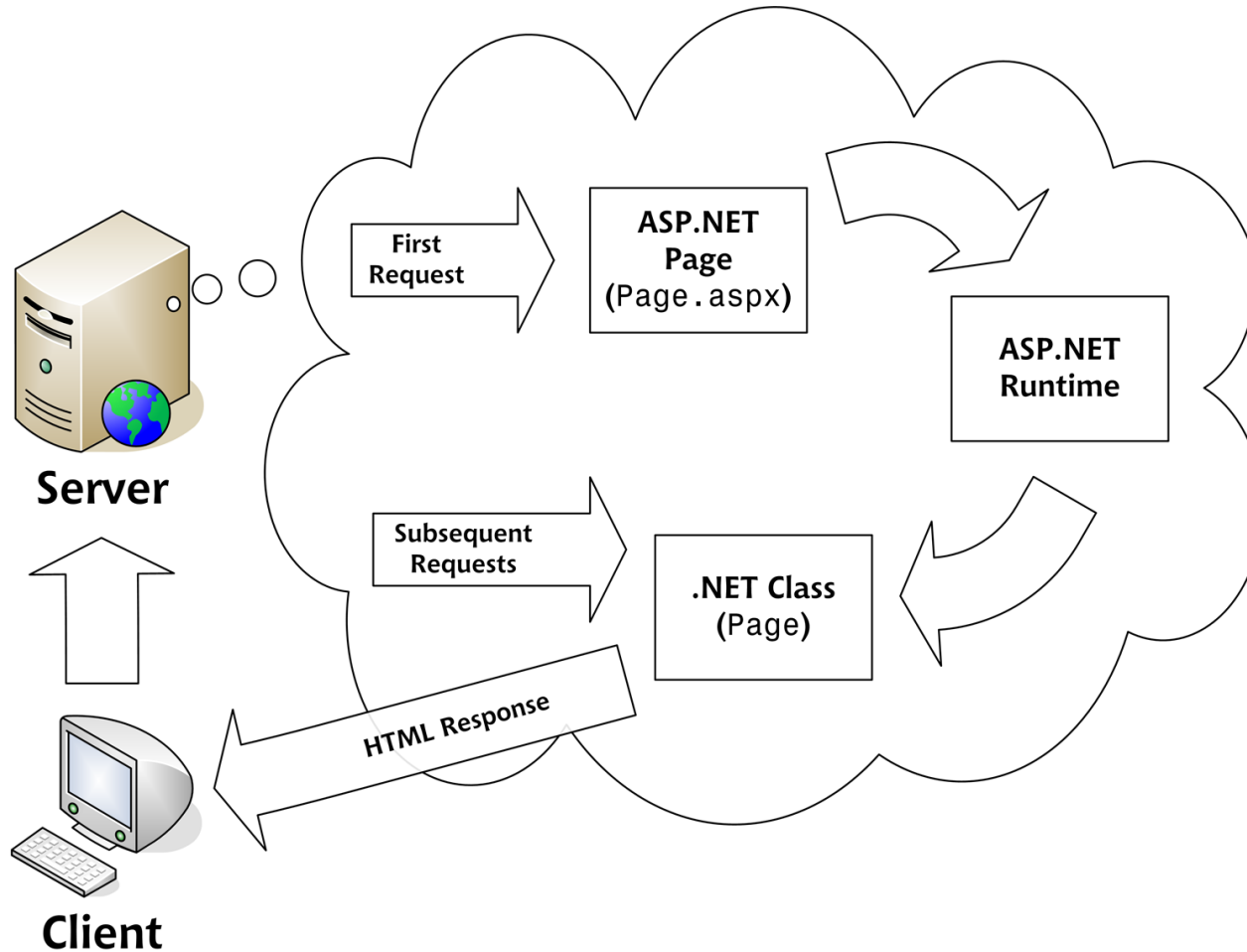
- Web.config , Style sheets (css)

Các thành phần của một ứng dụng Web

- ❖ Trong một ứng dụng Web hoàn chỉnh, các phần thực thi của Web Form được lưu trong các file .dll và chạy trên server thông qua điều khiển của IIS



Cơ chế xử lý một trang ASP.NET



Cơ chế xử lý một trang ASP.NET

❖ Các bước xử lý

- Khi client request một trang ASP.NET từ trình duyệt
- Một HTTP request được gửi tới IIS trên Server
- Web Server sẽ chuyển request này đến Asp.net runtime
- Asp.net runtime chịu trách nhiệm tìm và load nội dung trang aspx được yêu cầu và biên dịch nó thành 1 lớp .NET để xử lý request
- Lớp này sau đó sẽ phát sinh nội dung mã HTML trả về trình duyệt của người dùng

Cơ chế xử lý một trang ASP.NET

- ❖ Người dùng thực hiện các thao tác trên trang web được trả về. Nếu các thao tác này đòi hỏi các xử lý tại server, thì trang này sẽ được gửi lại (post-back) về server. Thông tin trả về chứa các control ẩn chứa các thông tin về thao tác thực hiện của người dùng trên trang.
- ❖ Tại server, trang aspx được load lại, nhưng chỉ các trường ẩn mới được đọc và các sự kiện tương ứng mới được xử lý.
- ❖ Kết quả lại được gửi lại về browser.

Cấu trúc một Web Form aspx

❖ Một Web Form bao gồm các thành phần:

- Directives
- Code Declaration Blocks
- Code Render Blocks
- Web Controls
- Server-side comments
- Literal Text và HTML Tags

Code Declaration Blocks và Code Render Blocks có thể đặt trực tiếp trên WebForm hoặc tách biệt trong file Code Behind

Cấu trúc một Web Form aspx

```
<%@ Page Language="C#" %>
```

Page directives

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html>
```

```
<head>
```

```
<title>Sample Page</title>
```

Literal text và html tags

```
<script runat="server">
```

```
void Page_Load()
```

```
{
```

```
    messageLabel.Text = "Hello World";
```

```
}
```

```
</script>
```

Code declaration Blocks

```
</head>
```

```
<body>
```

```
<!-- comment here --%>
```

Server-side comments

```
<form runat="server">
```

```
<p>
```

```
<asp:Label id="messageLabel" runat="server" />
```

Web controls

```
</p>
```

```
<p>
```

```
<!-- Declare the title as string and set it --%>
```

```
<% string Title = "This is generated by a code render " +
```

```
"block."; %>
```

```
<%= Title %>
```

Code render Blocks

```
</p>
```

```
</form>
```

```
</body>
```

```
</html>
```

Directives

- ❖ Chứa các chỉ thị cho biết cách thức Web Form được biên dịch
- ❖ Được khai báo trong `<%@ ... %>` và có thể đặt tại bất kỳ vị trí nào trên Web Form
- ❖ Một số thuộc tính quan trọng: **Language**, **AutoEventWireup**, **CodeFile**, ...

Ví dụ:

```
<%@ Language="C#" AutoEventWireup="true"  
CodeFile="Default.aspx.cs" %>
```


Code **declaration** blocks

- ❖ Được khai báo nếu phần xử lý logic của chương trình được thể hiện ngay trong Web Form (không sử dụng code behind)
- ❖ Khai báo các phương thức hoặc các hàm xử lý sự kiện

Ví dụ:

```
<script runat="server">
    void mySub()
    {
        // Code here
    }
    void Page_Load( ... )
    {
        // Code here
    }
</script>
```

Code **Render** Blocks

- ❖ Là các đoạn code được thực thi khi một trang được nạp hoặc trả nội dung về phía người dùng.
- ❖ Bao gồm 2 loại:
 - Inline Code
 - Inline Expression

Code **Render** Blocks - **Inline Code**

- ❖ Bao gồm các lệnh xử lý trên server nhưng không trả nội dung về phía trình duyệt.
- ❖ Thường được sử dụng để **khai báo biến**
- ❖ Được khai báo trong cặp thẻ **<% ... %>**

Ví dụ:

<%

```
string Title = "This is generated by a code render  
block.";
```

%>

Code **Render** Blocks - **Inline Expression**

- ❖ Code xử lý trả thông tin về trình duyệt.
- ❖ Thông tin trả về có thể là nội dung của một biến hoặc kết quả của việc gọi thực hiện một phương thức
- ❖ Được khai báo trong cặp thẻ **<%= ... %>**

Ví dụ:

<%

string Title = "This is generated by a code render block.";

%>

<%= Title %> hoặc **<%= mySub() %>**

Web Controls

❖ Bao gồm 3 loại:

- Html Control
- Html Server Control
- Asp.net Server Control

❖ Được khai báo trong thẻ

`<form runat="server" > ... </form>`

Ví dụ:

`<asp:Label ID="Label1" runat="server" Text="Text Content" />`

`<asp:TextBox ID="TextBox1" runat="server" Text="Enter text here" />`

Server-side comments

- ❖ Thể hiện các ghi chú trên Web Form
- ❖ Sử dụng một trong 2 dạng:
 - **Html Comment** : `<!-- comment -->`
 - **Asp.net Comment**: `<%-- comment --%>`
- ❖ Html comment sẽ được gởi về trình duyệt do đó không thích hợp để comment nội dung Asp.net server-side code
- ❖ Html comment được dùng để ẩn thông tin đối với trình duyệt nhưng sẽ được xử lý bởi Asp.net runtime

Server-side comments

❖ Ví dụ:

```
<!--
```

```
<% string Title = "This is generated by a code render block.";
```

```
%>
```

```
<%= Title %>
```

```
-->
```

Kết xuất tại trình duyệt:

```
<!--
```

This is generated by a code render block.

```
-->
```

Literal Text và HTML Tags

- ❖ Cung cấp cấu trúc định dạng thông tin trang web (thông qua các thẻ Html) cùng với nội dung hiển thị tĩnh (literal text)
- ❖ Nếu không có thành phần này, trang web sẽ không có cấu trúc và trình duyệt sẽ không hiển thị được

Literal Text và HTML Tags

Ví dụ:

```
<%@ Page Language="C#" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<title>Sample Page</title>
<script runat="server">
void Page_Load()
{
    messageLabel.Text = "Hello World";
}
</script>
</head>
<body>
<form runat="server">
    <p>
        <asp:Label id="messageLabel" runat="server" />
    </p>
    <p>
        <!-- Declare the title as string and set it --%>
        <% string Title = "This is generated by a code render " +
        "block."; %>
        <%= Title %>
    </p>
</form>
</body>
</html>
```

Web Form được xây dựng với code behind

- ❖ Code behind: là file mã nguồn (C#, VB.net) chứa khai báo lớp có nhiệm vụ xử lý các logic nghiệp vụ của chương trình hay các sự kiện xảy ra khi người dùng tương tác với WebForm
- ❖ Tên của lớp thường trùng với tên của WebForm
Ví dụ: nếu tên WebForm là `index` → tên class sẽ là `index`
- ❖ Tất cả các class xử lý sự kiện trên WebForm đều kế thừa từ lớp **System.Web.UI.Page**
- ❖ Tất cả các class đều chứa hàm **Page_Load** tự động gọi thực hiện khi WebForm nhận được request từ client
- ❖ Hàm **Page_Load** dùng để khởi tạo nội dung của các control trên WebForm

Web Form được xây dựng với code behind

❖ Ví dụ:

```
using System;
using System.Data;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Page.IsPostBack == false)
        {
            TextBox1.Text = "initialized data";
        }
    }
}
```

LÀM VIỆC VỚI CONTROL



❖ Server Control

❖ ASP.NET Server control vs HTML control

❖ Simple control

- Label, Button (Button, LinkButton, ImageButton), TextBox
- List Control (ListBox, DropDownList, Table, DataGrid, DataList, Repeater)

❖ Validation control

❖ Một số control khác

Server Control

- ❖ **Server control là những control mà Web server (IIS) có thể “hiểu được”.**
- ❖ **Các loại server control**
 - HTML Server Control
 - ASP.NET Server Control
- ❖ **Dùng để thể hiện giao diện web**

HTML Server Control

- ❖ HTML Server control là những tag do HTML tạo ra
- ❖ Duy trì tương thích với các tag HTML cũ
- ❖ Thêm vào thuộc tính run at = “server”
- ❖ Tất cả HTML Server Control phải được đặt trong tag <form> với thuộc tính run at = “server”

ASP.NET Server Control

- ❖ ASP.NET Server Control là những tag đặc biệt của riêng ASP.NET
- ❖ Các control này cũng sẽ được xử lý trên server, và đòi hỏi phải có thuộc tính runat = “server”
- ❖ Không tương ứng với HTML tag nào
- ❖ Có thể dùng để thể hiện các thành phần phức tạp

ASP.NET Server control vs HTML control

Tính năng	ASP.NET Server control	HTML control
Server event	Kích hoạt được một số sự kiện cụ thể trên Server	Chỉ có thể kích hoạt các sự kiện mức trang trên server (post-back)
Quản lý trạng thái	Dữ liệu nhập vào control được lưu giữ lại sau mỗi request	Dữ liệu không được lưu giữ lại, phải tự lưu và điền vào sử dụng script
Tương thích	Tự động nhận diện loại trình duyệt và tạo hiển thị cho phù hợp	Không tự động nhận diện trình duyệt
Các thuộc tính	.NET Framework cung cấp một tập các thuộc tính cho mỗi control, cho phép thay đổi phần hiển thị và hành vi thông qua mã lệnh	Chỉ có các thuộc tính chuẩn của HTML

Tại sao sử dụng HTML control ?

❖ Sử dụng HTML control khi

- Nâng cấp từ ASP
- Không phải tất cả các control đều cần các sự kiện server-side hoặc quản lý trạng thái

Khác biệt trong HTML tag

❖ Server control:

- `<asp:controlnameid="some_id" runat="server"/>`

❖ HTML control

- HTML tag

```
{ <asp:TextBox id="txtText" runat="server"></asp:TextBox>  
  <INPUT type="text" id="textfield1">
```

```
{ <asp:Button id="btnShow" runat="server"  
  Text="Show"></asp:Button>  
  <INPUT type="button" value="Show">
```

Server Control & HTML Control

	Server control	HTML control
Hiển thị Text	Label, TextBox, Literal	Label, Text Field, Text Area, Password Field
Hiển thị Table	Table, DataGrid	Table
List	DropDownList, ListBox, DataList, Repeater	List Box, Dropdown
Thực hiện lệnh	Button, LinkButton, ImageButton	Button, Reset Button, Submit Button
Đặt giá trị	CheckBox, CheckBoxList, RadioButton, RadioButtonList	Checkbox, Radio Button
Hiển thị Image	Image, ImageButton	Image
Liên kết	Hyperlink	Anchor <a>

Server Control & HTML Control

Group control	Panel, Placeholder	Flow Layout, Grid Layout
Ngày tháng	Calendar	none
Quảng cáo	AdRotator	none
Đường kẻ	none	Horizontal Rule
Lấy tên file từ client	none	File Field
Lưu dữ liệu trên trang	(sử dụng quản lý trạng thái)	Input Hidden
Kiểm tra tính đúng đắn của dữ liệu nhập	RequiredFieldValidator, CompareValidator, RangeValidator, RegularExpressionValidator, CustomValidator, ValidationSummary	none (sử dụng client script)

Label, Buttons, TextBox - HTML tag

- **Label**

- `<asp:Label id="Label1" runat="server">Please input text</asp:Label>`

- **Buttons (Button, LinkButton, ImageButton)**

- `<asp:Button id="Button1" runat="server" Text="Button"></asp:Button>`

- `<asp:LinkButton id="LinkButton1" runat="server">LinkButton</asp:LinkButton>`

- `<asp:ImageButton id="ImageButton1" runat="server"></asp:ImageButton>`

- **TextBox**

- `<asp:TextBox id="TextBox1" runat="server"></asp:TextBox>`

Chỉnh sửa Thuộc tính lúc Thiết kế

Button

Properties

Button1 System.Web.UI.WebControls.Button

(DataBindings)	
(ID)	Button1
AccessKey	
BackColor	<input type="text"/>
BorderColor	<input type="text"/>
BorderStyle	NotSet
BorderWidth	
CausesValidation	True
CommandArgument	
CommandName	
CssClass	
Enabled	True
EnableViewState	True
Font	
ForeColor	<input type="text"/>
Height	
TabIndex	0
Text	Button
ToolTip	
Visible	True
Width	

Properties

LinkButton1 System.Web.UI.WebControls.LinkButton

(DataBindings)	
(ID)	LinkButton1
AccessKey	
BackColor	<input type="text"/>
BorderColor	<input type="text"/>
BorderStyle	NotSet
BorderWidth	
CausesValidation	True
CommandArgument	
CommandName	
CssClass	
Enabled	True
EnableViewState	True
Font	
ForeColor	<input type="text"/>
Height	
TabIndex	0
Text	LinkButton
ToolTip	
Visible	True
Width	

Link Button

Image Button

Properties

ImageButton1 System.Web.UI.WebControls.ImageButton

(DataBindings)	
(ID)	ImageButton1
AccessKey	
AlternateText	
BackColor	<input type="text"/>
BorderColor	<input type="text"/>
BorderStyle	NotSet
BorderWidth	
CausesValidation	True
CommandArgument	
CommandName	
CssClass	
EnableViewState	True
ForeColor	<input type="text"/>
Height	
ImageAlign	NotSet
ImageUrl	<input type="text"/>
TabIndex	0
ToolTip	
Visible	True
Width	

Một số thuộc tính quan trọng

- Label, Buttons
 - Thuộc tính Text
- TextBox

Thuộc tính	Sử dụng để
Text	Lấy/Đặt dữ liệu cho TextBox.
TextMode	SingleLine, MultiLine (scrollable), Hoặc Password.
Enabled	Enable/Disable TextBox
Visible	Show/Hide TextBox
ReadOnly	Ngăn không cho người dùng thay đổi dữ liệu trong TextBox.
AutoPostBack	Khi được thiết lập là True, sự kiện TextChanged trong TextBox sẽ tự động kích hoạt post-back về server (không phải được kích hoạt khi nội dung TextBox thay đổi)

List Control

ListBox, DropDownList, Table

Control	Sử dụng khi
ListBox	Hiển thị danh sách dữ liệu read-only đơn giản, sử dụng scroll
DropDownList	Hiển thị danh sách dữ liệu read-only đơn giản, sử dụng cửa sổ xổ xuống
Table	Hiển thị thông tin dưới dạng dòng và cột. Table control cho phép xây dựng các bảng động bằng mã lệnh sử dụng các thuộc tính tập hợp TableRows và TableCells

List Control: HTML tag

❖ ListBox

- `<asp:ListBox id="ListBox1" runat="server"></asp:ListBox>`

❖ DropDownList

- `<asp:DropDownList id="DropDownList1" runat="server"></asp:DropDownList>`

❖ Table

- `<asp:Table id="Table1" runat="server" Width="100px" Height="70px"></asp:Table>`

Chỉnh sửa Thuộc tính lúc Thiết kế

Properties

ListBox1 System.Web.UI.WebControls.ListBox

(DataBindings)

(ID) **ListBox1**

AccessKey

AutoPostBack False

BackColor

CssClass

DataMember

DataSource

DataTextField

DataTextFormatStri

DataValueField

Enabled True

EnableViewState True

Font

ForeColor

Height

Items (Collection)

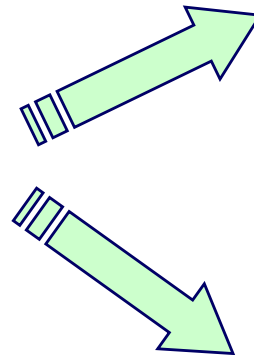
Rows 4

SelectionMode Single

TabIndex 0

Visible True

Width



ListItem Collection Editor

Members:

0	dong 1
1	Dong 2

↑

↓

Add Remove

Dong 2 Properties:

Selected	False
Text	Dong 2
Value	

OK Cancel Help

TableRow Collection Editor

Members:

0	TableRow
1	TableRow
2	TableRow

↑

↓

Add Remove

TableCell Collection Editor

Members:

0	TableCell
1	TableCell

↑

↓

Add Remove

TableCell Properties:

Appearance

BackColor	Silver
BorderColor	
BorderStyle	NotSet
BorderWidth	
ColumnSpan	0
CssClass	

Font

ForeColor	
Text	

Behavior

AccessKey	
-----------	--

OK Cancel Help

Thêm các mục dữ liệu vào thời điểm chạy ứng dụng

❖ **ListBox và DropDownList:**

- Sử dụng phương thức Add và danh sách Items của control
- Ví dụ:

```
protected void btnShow_Click(object sender,  
    EventArgs e)  
{  
    ListBox1.Items.Add(txtSource.Text) ;  
    DropDownList1.Items.Add(txtSource.Text) ;  
}
```

Lấy mục dữ liệu được chọn

- ❖ Dùng thuộc tính **SelectedItem** để lấy mục dữ liệu được chọn hiện tại trong List

```
protected void Page_Load(object sender, EventArgs e)
{
    // Test if there is a selected item.
    if (ListBox1.SelectedItem != null)
        // Display the selected item.
        Label1.Text = "The selected item is: " +
                      ListBox1.SelectedItem.Text;
    else
        Label1.Text = "No item is selected.";
}
```

Một số control khác

- ❖ **Lấy và thiết lập giá trị**
 - RadioButton, RadioButtonList, CheckBox, CheckBoxList
- ❖ **Gom nhóm**
 - Panel
- ❖ **Hiển thị Hình ảnh và Quảng cáo**
 - Background, Foreground, Image, AdRotator
- ❖ **Lấy thông tin Ngày tháng**
 - Calendar
- ❖ **Lấy Tập tin từ Client**
 - File Field HTML control

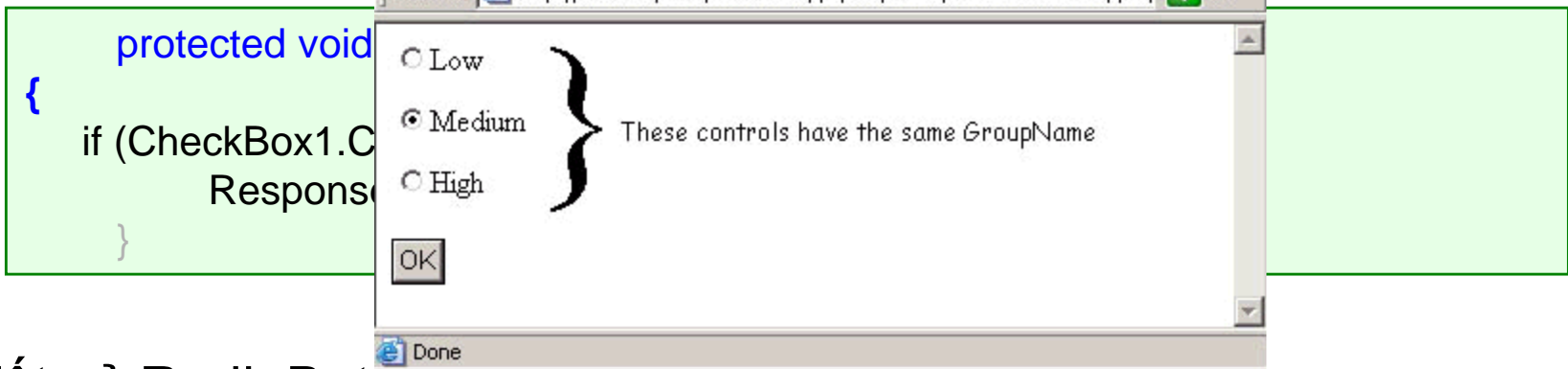
Một số control khác

RadioButton, CheckBox

❖ HTML tag

- `<asp:RadioButton id="R1" runat="server"></asp:RadioButton>`
- `<asp:CheckBox id="C1" runat="server"></asp:CheckBox>`

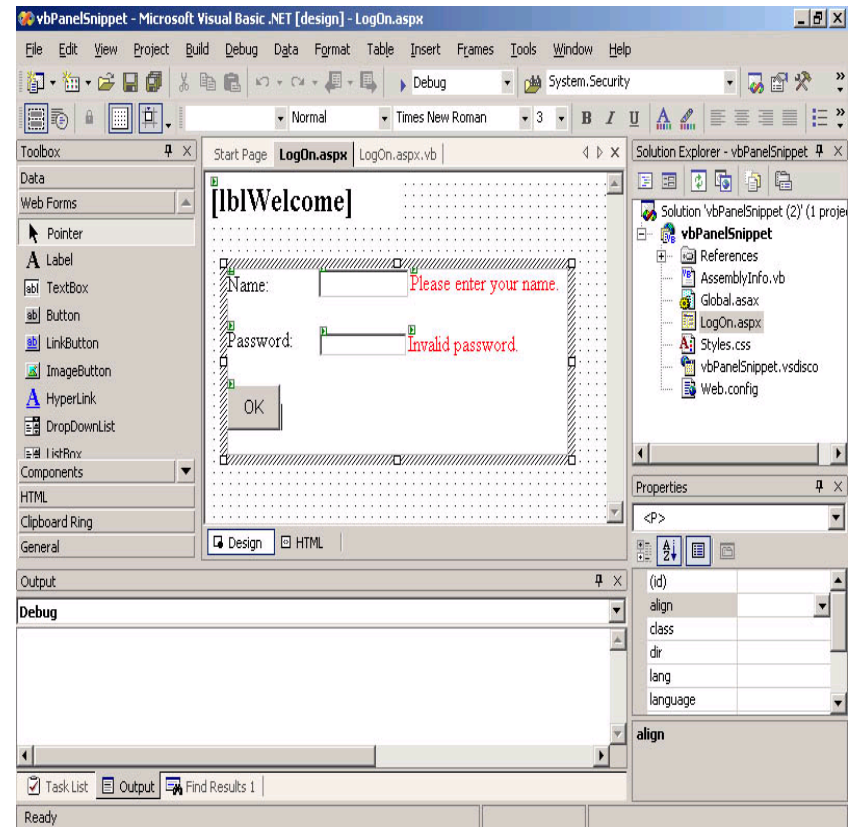
❖ Sử dụng thuộc tính GroupName để gộp các control có cùng giá trị thiết lập



- Tất cả RadioButton phải có cùng một GroupName

Panel

- ❖ Kéo thả control Panel vào Web form.
- ❖ Kéo các control khác lên trên Panel để gom nhóm



Hình ảnh và Quảng cáo

❖ Hình nền

- Sử dụng thuộc tính Background của Web form
- Sử dụng thuộc tính BackImageUrl của Panel control

❖ Hình ảnh

- Sử dụng Image control

❖ Button bằng hình ảnh

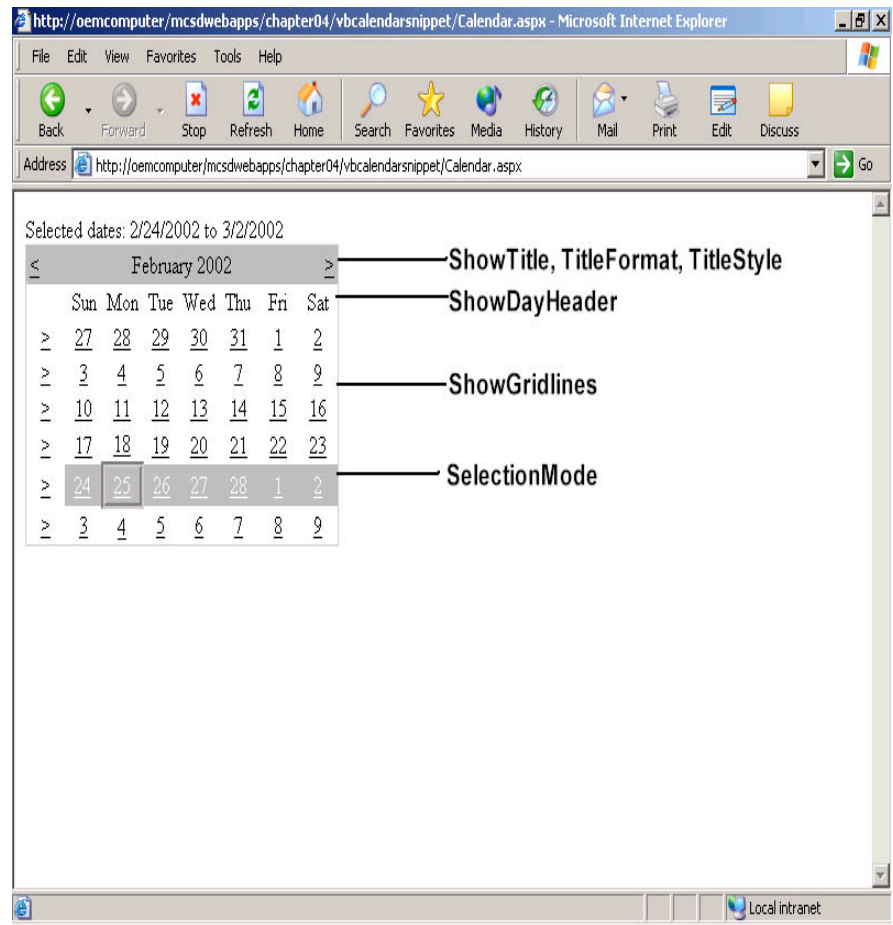
- Sử dụng ImageButton control

❖ Quảng cáo

- Sử dụng AdRotator control

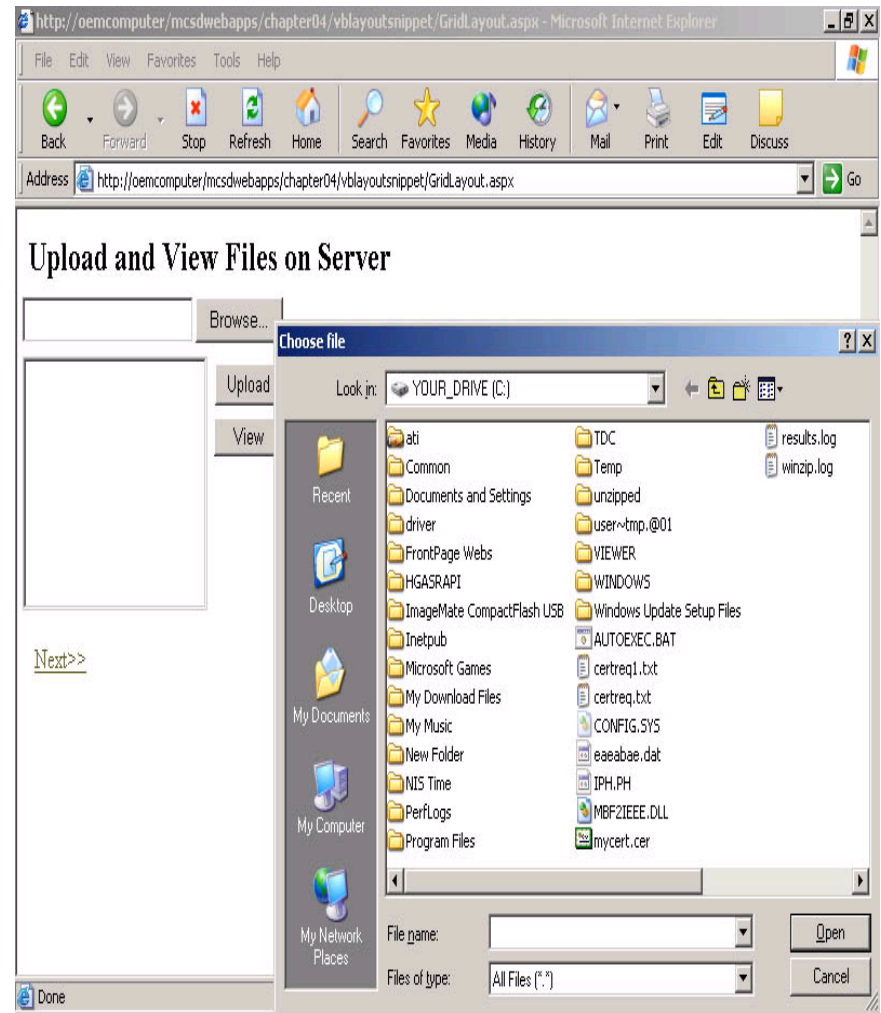
Calendar

- ❖ Sử dụng Calendar control để lấy thông tin về Ngày tháng
- ❖ Để lấy hoặc thiết lập giá trị ngày tháng trên Calendar control, sử dụng hàm xử lý sự kiện `SelectionChanged` và thuộc tính `SelectedDate` hoặc `SelectedDates`



File Field HTML control

- ❖ Sử dụng File Field HTML control để upload file từ client lên server
- ❖ File Field HTML control = Text Field HTML control + Submit Button HTML control
- ❖ Nhấn vào Browse button sẽ hiển thị cửa sổ cho phép chọn đường dẫn đến các file muốn upload trên máy client



Sử dụng Css đối với các asp.net controls

❖ Cách 1:

- Khai báo trực tiếp trong thuộc tính **style** (inline css) của thẻ asp.net controls

Ví dụ:

```
<asp:label id="labMsg" runat="server" Text="hello world"  
style="font-style: italic; text-decoration: underline; color: Red" />
```

- Sử dụng các thuộc tính định dạng built-in được hỗ trợ đối với asp.net controls tương ứng

```
<asp:label id="labMsg" runat="server" ForeColor="Blue" />
```

Sử dụng Css đối với các asp.net controls

❖ Cách 2:

- Thay đổi nội dung thuộc tính Style (inline css) của đối tượng server control tương ứng

Ví dụ:

```
<asp:label id="labMsg" runat="server" Text="hello world" style="font-style: italic; text-decoration: underline; color: Red" />
```

Code behind:

```
labMsg.Style["font-style"] = "italic" ;  
labMsg.Style["text-decoration"] = "underline" ;  
labMsg.Style["color"] = "Red" ;
```

Sử dụng Css đối với các asp.net controls

❖ Cách 3:

- Thiết lập giá trị thuộc tính **CssClass** của đối tượng server control tương ứng (embedded , external css)

Ví dụ:

```
.myStyle {  
    font-style: bold;  
    text-decoration: line-through;  
    color: green; }
```

Code behind:

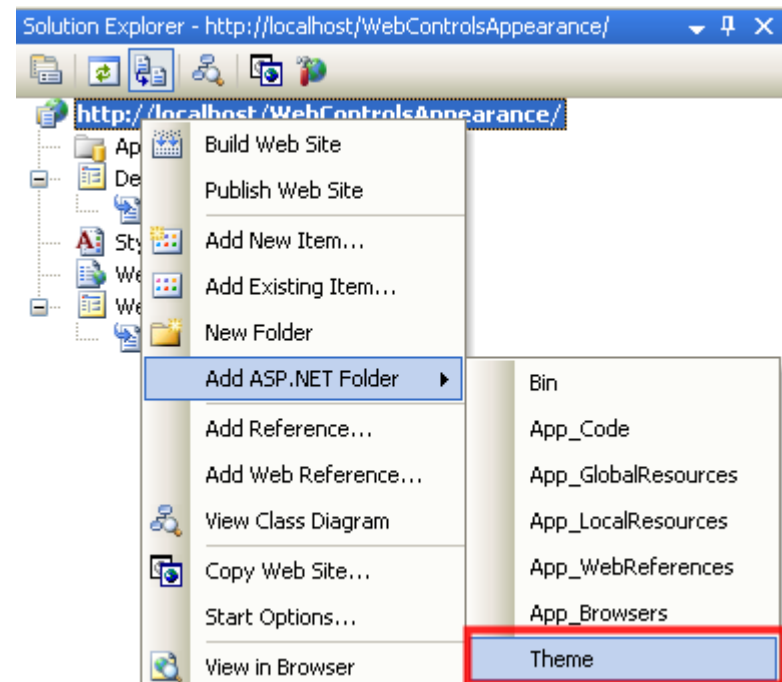
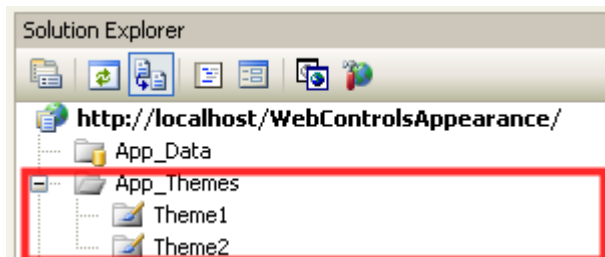
```
labMsg.Style.Clear( );  
labMsg.CssClass = "myStyle";
```

Theme, Skin

- ❖ Theme, skins cho phép xây dựng các tập kiểu định dạng hiển thị (style) có thể áp dụng đối với mọi asp.net server controls trong toàn bộ site
- ❖ Có thể xây dựng các style áp dụng đối với các control đơn giản (Label, TextBox...) và phức tạp (GridView ..) mà css không áp dụng được
- ❖ Mỗi theme có thể xem như 1 giao diện của trang web
- ❖ Giúp tạo ra giao diện nhất quán cho toàn bộ trang web

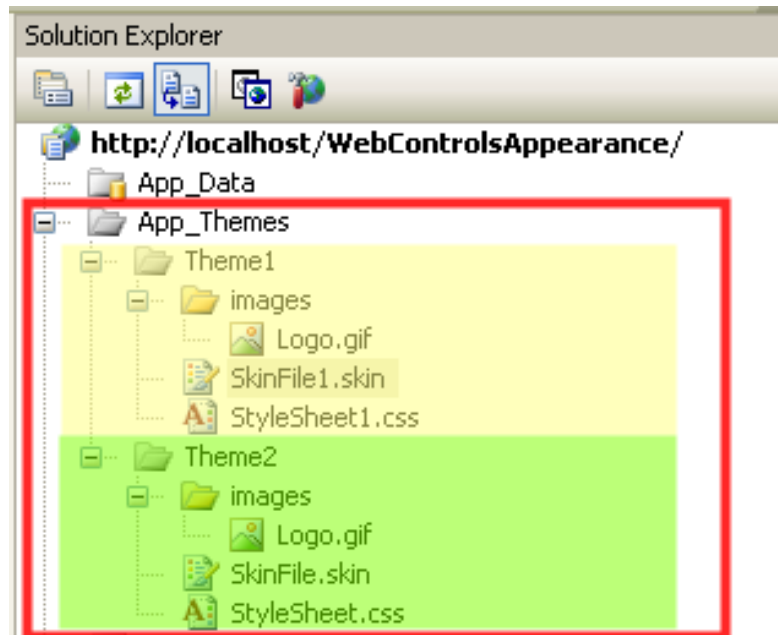
Theme

- ❖ Một asp.net website có thể định nghĩa nhiều theme.
- ❖ Tất cả các theme phải đặt trong thư mục **App_Themes**



Theme

❖ Mỗi theme có thể định nghĩa nhiều **skin** file, **css** file, hình ảnh, ...



Skin

❖ Skin file: mô tả tập các kiểu định dạng hiển thị của các asp.net server controls

Ví dụ:

Định nghĩa style áp dụng đối với mọi Label và TextBox trong trang web

```
<asp:Label runat="server" ForeColor="Blue" Font-Size="10pt" Font-Name="Verdana" />
```

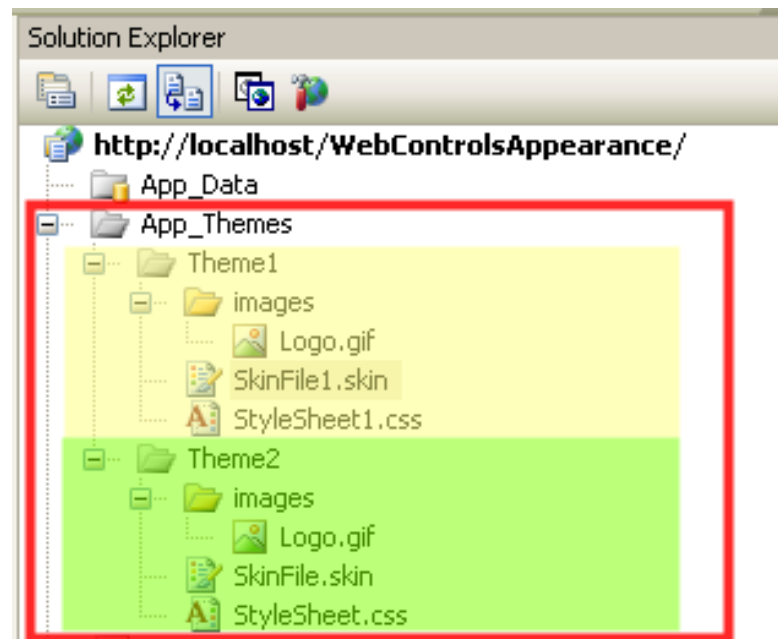
```
<asp:TextBox runat="server" BackColor="#FFFFC0" ForeColor="Green" />
```

Các định dạng trên không chứa thuộc tính ID hay bất kỳ thuộc tính nào liên quan đến một control cụ thể (Ví dụ: Text ...)

Theme, Skin

❖ Sử dụng Theme trong WebForm

`<%@ Page ... Theme="Theme1" ...`



Theme, Skin

❖ Sử dụng Theme đối với mọi WebForm trong site

Trong file **web.config**, bổ sung:

```
<system.web>
```

```
...
```

```
<pages theme="Theme1" />
```

```
...
```

```
</system.web>
```

→ Tất cả các WebForm khi thêm mới vào site đều sử dụng **Theme1** như là theme mặc định. Nếu WebForm này khai báo thuộc tính Theme → Theme trong WebForm sẽ được ưu tiên sử dụng

Overriding Theme

- ❖ Giả sử trong **theme1** định nghĩa style của TextBox

```
<asp:TextBox runat="server" BackColor="#FFFFC0"  
ForeColor="Green" />
```

- ❖ Trong WebForm định nghĩa 1 TextBox với định dạng

```
<asp:TextBox runat="server" BackColor="#FF8000"  
ForeColor="Fuchsia" ID="TextBox1" />
```

→ Style trong theme sẽ override style cụ thể của control trong WebForm

Overriding Theme

❖ Để override style định nghĩa trong theme

❖ Cách 1:

`<asp:TextBox ID="TextBox1" EnableTheming="false" ...`

❖ Cách 2:

`<%@ Page ... StyleSheetTheme="Theme1" ...`

- style trong **Theme** sẽ được sử dụng **sau khi** áp dụng style cụ thể của server control trong WebForm
- style trong **StyleSheetTheme** sẽ được sử dụng **trước khi** áp dụng style cụ thể của server control trong WebForm

Named skin

- ❖ Trong trường hợp một loại control cần thiết lập nhiều định dạng khác nhau, ta có thể phân biệt giữa các định dạng này thông qua thuộc tính **SkinID**

Ví dụ:

```
<asp:TextBox runat="server" BackColor="#FFFFC0" ForeColor="Green" />
```

```
<asp:TextBox runat="server" BackColor="#FF0000" ForeColor="White" SkinID="skin1" />
```

Sử dụng trong WebForm:

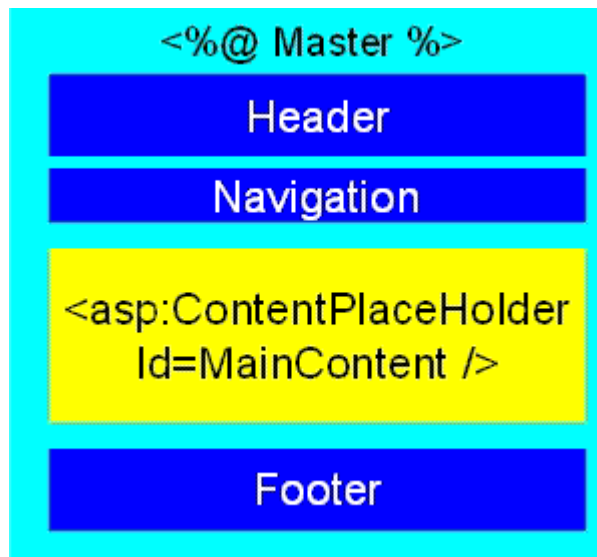
```
<asp:TextBox ID="TextBox1" runat="server" />
```

```
<asp:TextBox ID="TextBox1" runat="server" SkinID="skin1" />
```

Master Page

❖ **Master Page** cho phép định nghĩa layout template nhất quán cho các WebForm trong site

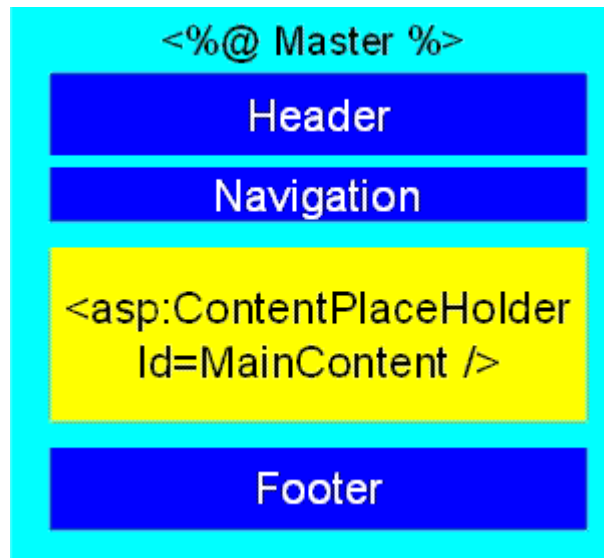
Ví dụ:



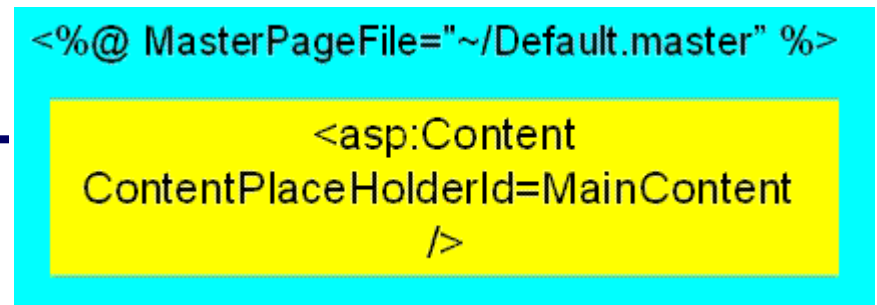
Master Page

Master Page

- ❖ **Content Page** là các **WebForm** kế thừa layout template mà Master Page đã định nghĩa và bổ sung thêm nội dung tương ứng với chức năng của WebForm này



Master Page (.master)

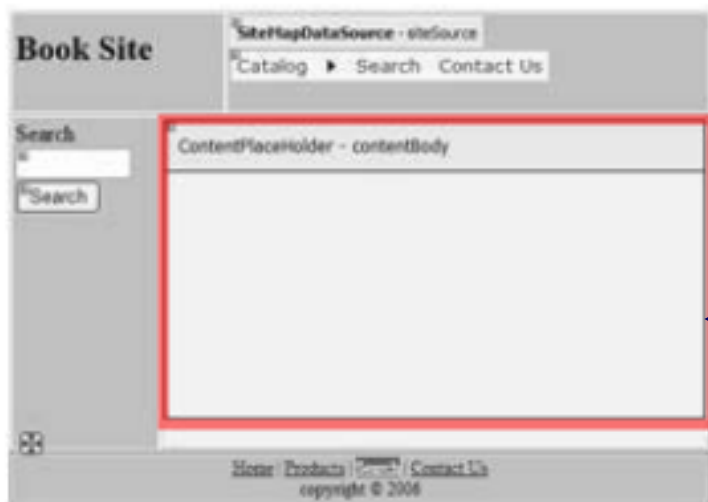


Content Page (.aspx)

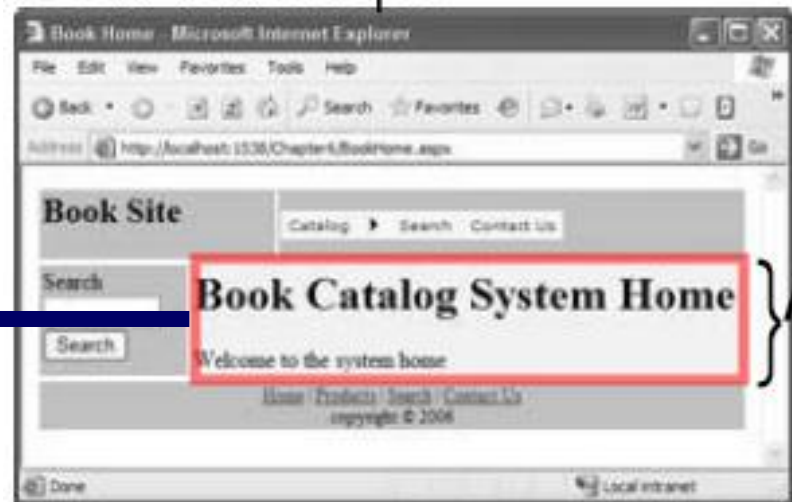
Master Page

- ❖ Master page sẽ định nghĩa các **PlaceHolderControl**.
- ❖ Content page sẽ chèn nội dung tương ứng vào các **PlaceHolderControl** trong

Books.Master



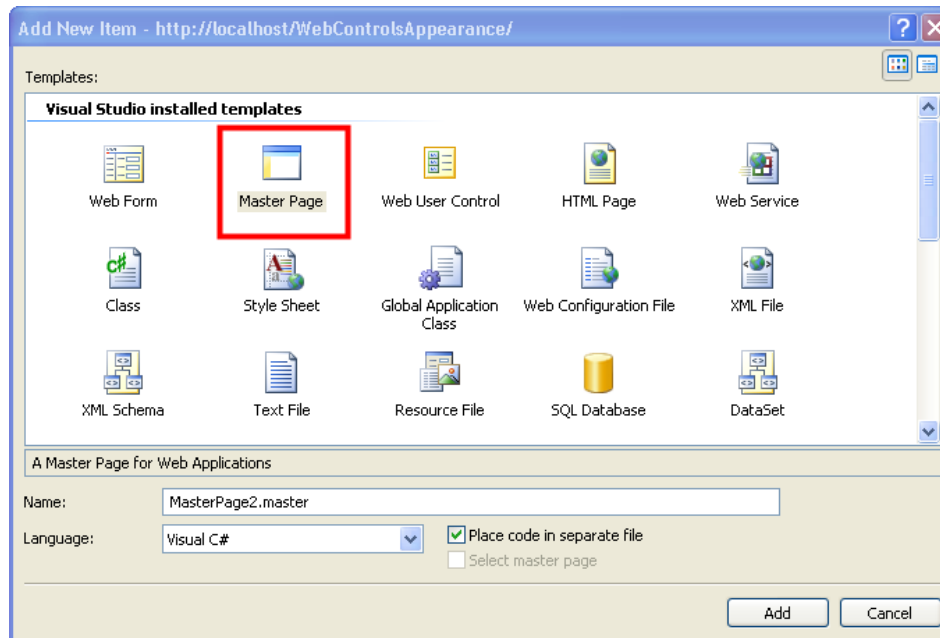
BookHome.aspx



Master Page

❖ Tạo Master page.

Website → Add New Item → Master page

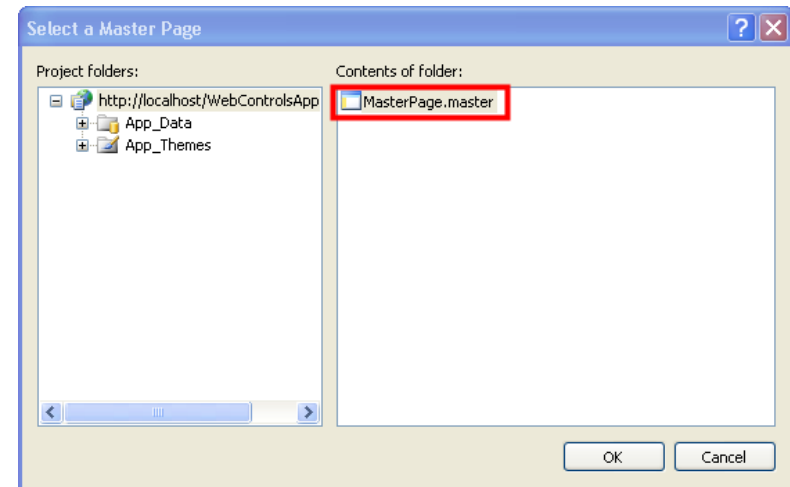
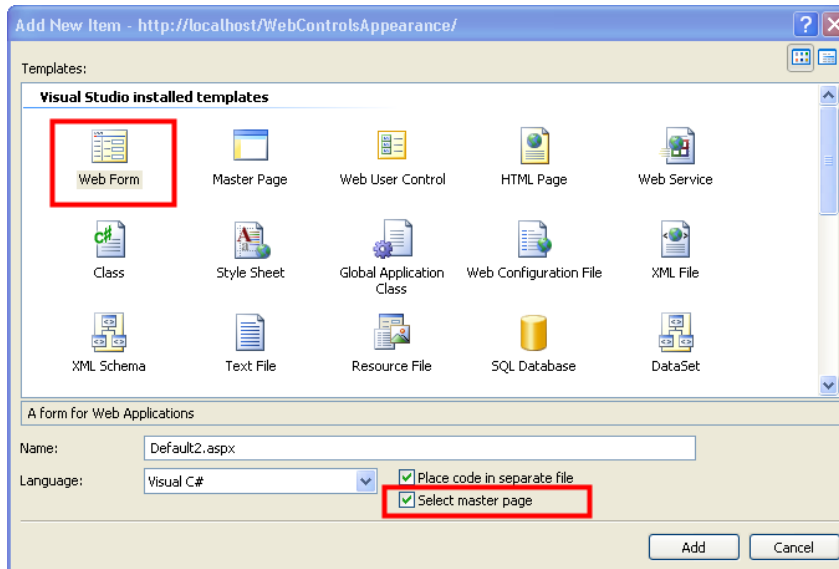


Master Page

```
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs"
    Inherits="MasterPage" %>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:contentplaceholder id="ContentPlaceHolder1" runat="server">
            </asp:contentplaceholder>
        </div>
    </form>
</body>
</html>
```

Master Page

❖ Tạo Content page



Master Page


❖ Content page

```
<%@ Page Language="C#" MasterPageFile="~/MasterPage.master"
    AutoEventWireup="true"
    CodeFile="Default2.aspx.cs"
    Inherits="Default2"
%>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
    Runat="Server">
</asp:Content>
```

❖ Master page

```
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs"
    Inherits="MasterPage" %>

....
<asp:contentplaceholder id="ContentPlaceHolder1" runat="server">
</asp:contentplaceholder>
....
```



Truy xuất nội dung trong Master Page

❖ Trong một số trường hợp, ta cần truy xuất và thay đổi một số nội dung trong Master page từ Content Page.

❖ Cách 1:

Sử dụng phương thức `FindControl` từ đối tượng Master

Ví dụ:

```
HyperLink ad = (HyperLink)Master.FindControl("controlInMasterPage");  
if (ad != null)  
{  
    ad.ImageUrl = "images/Logo.gif";  
    ad.NavigateUrl = "http://www.interneturl.com";  
}
```

Truy xuất nội dung trong Master Page

❖ Cách 2:

Đóng gói dữ liệu trong Master page thành các thuộc tính có thể truy xuất (Properties).

Content page sẽ truy xuất dữ liệu của Master page thông qua các Properties này.

Truy xuất nội dung trong Master Page

❖ Ví dụ: khai báo 2 properties trong master page

```
public partial class ProgrammedContentMaster : System.Web.UI.MasterPage
{
    public string AdImageUrl
    {
        get { return imgbtnAd.ImageUrl; }
        set { imgbtnAd.ImageUrl = value; }
    }
    public string AdNavigateUrl
    {
        get { return imgbtnAd.NavigateUrl; }
        set { imgbtnAd.NavigateUrl = value; }
    }
}
```

❖ Truy xuất từ Content Page

```
ProgrammedContentMaster pcm = (ProgrammedContentMaster)Master;
pcm.AdImageUrl = "~/Images/something.gif";
pcm.AdNavigateUrl = "http://www.somewhereelse.com";
```

Truy xuất nội dung trong Master Page

❖ Cách 3: chỉ định kiểu cụ thể của Master Page trong content page

```
<%@ Page Language="C#" MasterPageFile="~/MasterPage.master" .... %>
```

```
<%@ MasterType VirtualPath="~/MasterPage.master" %>
```

hoặc

```
<%@ MasterType TypeName="MyMasterPageClassName" %>
```

❖ Theo cách này, việc truy xuất đến Master page từ content page trong code behind không cần thực hiện ép kiểu

```
this.Master.AdImageUrl = "~/Images/something.gif";
```

```
this.Master.AdNavigateUrl = "http://www.somewhereelse.com";mm42
```





Thank You !

www.themegallery.com