

Bài 2: ASP.NET Server Controls

- ✓ **Server Controls**
- ✓ **HTML và ASP.NET Web Controls**
- ✓ **Validation Controls và Rich Controls**
- ✓ **List Controls và User Controls**

Server Controls

I. Server Controls

Điều khiển trên Server là công nghệ mới và là thành phần chủ yếu để xây dựng những trang Web với các Form tương tác. Chúng cho phép ta sử dụng mô hình lập trình hướng sự kiện trên server, tương tự như lập trình hướng sự kiện khi phát triển các chương trình ứng dụng trên Windows.

Kiến trúc sự kiện

- ASP.NET Web Form tương tự như ứng dụng trên Windows, tất cả phụ thuộc vào **sự kiện**.
- Ta tạo một form và đặt vào đó các điều khiển để tương tác với người sử dụng.
- Sự kiện sẽ phát sinh khi người sử dụng thao tác trên các điều khiển, ta xây dựng các trình xử lý cho các sự kiện đó.
- Các trình xử lý sự kiện sẽ cập nhật dữ liệu trên trang Web, trên các điều khiển hoặc thực thi các công việc theo yêu cầu.

Server Controls

Biên dịch

- Toàn bộ trang Web, bao gồm tất cả các thành phần của HTML, text, và các dữ liệu khác được biên dịch thành một lớp (**Page class**). Lớp dữ liệu này sẽ được thực thi và trả lại kết quả cho Client.
- Các dữ liệu tĩnh (text, HTML, client-side script ...) được trả lại cho Client qua Response khi lớp này thực thi, chúng ta không truy nhập được trên Server.
- Các điều khiển với thuộc tính **runat="server"** được khởi tạo thành các đối tượng thành viên trong lớp Page. Điều này có nghĩa là ta có thể viết chương trình sử dụng các đối tượng đó như truy nhập các thuộc tính, gọi các phương thức và xử lý các sự kiện trên Server.
- Mô hình này được thực hiện thông qua **kiến trúc postback**, các thành phần dữ liệu trong trang Web sẽ được gửi trả lại cho tập ASP.NET đó trên server khi người sử dụng tương tác với các điều khiển trên trang Web.

Các điều khiển trên server được biên dịch thành các đối tượng toàn cục trong lớp Page

Server Controls

Các nhóm điều khiển trên server

- **HTML Server Controls:** Tương ứng với các thành phần HTML.
- **ASP.NET Web Form Controls:** Bộ điều khiển được xây dựng bởi ASP.NET
 - **ASP.NET Validation Controls**
 - **ASP.NET List Controls:**
 - **ASP.NET Rich Controls:**
 - **ASP.NET Mobile Controls:**
 - **Các loại điều khiển khác như:** Login Controls, DataSource Controls, DataDisplay Controls, ... (theo các phiên bản ASP.NET)
- **User Controls:** Custom Controls và Composite Controls

Server Controls

Server controls và loại dữ liệu server

- Vấn đề đáng lưu ý là khi nào sử dụng các điều khiển trên Server và các thành phần HTML thông thường. Trên thực tế, ta có 3 tùy chọn có thể sử dụng đó là, dùng thành phần input thông thường của HTML, sử dụng điều khiển `HtmlInputText`, hoặc sử dụng điều khiển ASP Textbox.
- Server controls làm tăng tài nguyên sử dụng của trang Web so với sử dụng thành phần HTML thông thường. Trong một số trường hợp, ta có thể không cần phải sử dụng điều khiển trên server.
 - ✓ Thành phần đó chỉ dùng để thực thi chương trình client-side script
 - ✓ Liên kết dùng để mở URL mà không cần xử lý giá trị địa chỉ trên server
 - ✓ Khi các thuộc tính, phương thức và sự kiện của thành phần đó không sử dụng trên server

Ta vẫn có thể sử dụng tập hợp `Request.Form` và `Request.QueryString` để truy nhập thành phần HTML thông thường cũng như điều khiển trên server.

Server Controls

Sử dụng server controls

Design – Time: Trong phần giao diện (*.aspx)

```
<form runat="server">  
    <ServerControlClass id="ID" runat="server" attribute="value"/>  
    <ServerControlClass id="ID" runat="server" attribute="value"/>  
</form>
```

Run – Time: Trong chương trình (*.cs)

```
<script runat="server">  
    ServerControlClass ID = new ServerControlClass();  
    ID.attribute = value  
</script>
```

HTML Server Controls

2. HtmlServerControls

HtmlServerControls được định nghĩa trong Namespace **System.Web.UI.HtmlControls**.

Là các điều khiển server tương ứng với các thành phần HTML, cho phép chuyển một thẻ HTML thành điều khiển server.. Dạng thức:

<%@ Import Namespace = "System.Web.UI.HtmlControls" %>

Design – Time: Trong phần giao diện (*.aspx)

```
<form runat="server">  
    <HtmlTag id="ID" runat="server" attribute="value"/>  
    <HtmlTag id="ID" runat="server" attribute="value">  
        Value  
    </HtmlTag>  
</form>
```

Run – Time: Trong chương trình (*.cs)

```
<script runat="server">  
    HtmlControlClass ID = new HtmlControlClass();  
    ID.attribute = value;  
</script>
```

HTML Server Controls

2.1. Các lớp cơ sở

HtmlControl: Là lớp cơ sở gốc

Thuộc Tính

Attributes Tập hợp thuộc tính name/value của điều khiển. Except custom

ClientID Trả lại ID của điều khiển tạo ra bởi ASP.NET

Controls Trả lại đối tượng ControlCollection gồm tất cả các điều khiển con

Disabled Boolean. Thiết lập kích hoạt của điều khiển

EnableViewState Boolean. Thiết lập viewstate của điều khiển và các con

ID Thiết lập ID của control

Page Đối tượng Page chứa điều khiển

Parent Điều khiển mẹ, chứa điều khiển hiện hành

Style Tập hợp các thuộc tính CSS

TagName Tên thẻ của thành phần

Visible Boolean. Thiết lập sự hiển thị

Phương Thức

DataBind() Kết hợp dữ liệu với điều khiển và các con

FindControl() Tìm điều khiển được chứa trong điều khiển hiện hành

HasControls()

Boolean. Kiểm tra xem có các điều khiển con

Sự Kiện

DataBinding

Sự kiện xảy ra khi dữ liệu được kết hợp với điều khiển

HtmlContainerControl: lớp cơ sở thẻ chứa

Thuộc tính:

InnerHtml

Truy nhập nội dung HTML và văn bản giữa thẻ đóng và thẻ mở

InnerText

Truy nhập nội dung văn bản giữa thẻ đóng và thẻ mở

HTML Server Controls

2.2. Các lớp dữ liệu

Namespace `System.Web.UI.HtmlControls` bao gồm các lớp dữ liệu tương ứng của các HTML controls. Các lớp dữ liệu này được kế thừa từ lớp **HtmlControl** hoặc **HtmlContainerControl** tùy theo các thành phần HTML tạo ra là loại thẻ chứa thông tin hay thẻ rỗng.

- Lớp tạo thẻ có chứa thông tin: kế thừa từ lớp `HtmlContainerControl`
- Lớp tạo thẻ rỗng: kế thừa từ lớp `HtmlControl`

Tuy nhiên các loại điều khiển khác nhau hỗ trợ và cung cấp các thuộc tính cũng như sự kiện khác nhau. Ví dụ như khi thao tác với thẻ liên kết ta phải truy nhập được thuộc tính ***href*** hoặc bắt sự kiện khi người sử dụng click vào liên kết đó

HTML Server Controls

2.2. Các lớp dữ liệu

Các điều khiển bao gồm thuộc tính, phương thức và các hỗ trợ sự kiện khác nhau

HTML	Class	Thuộc tính	Sự Kiện
<a>	HtmlAnchor	Href, Target, Title, Name	OnserverClick
	HtmlImage	Align, Border, Height, Width, Src, Alt	None
<form>	HtmlForm	Name, Enctype, Method, Target	None
<button>	HtmlButton	CausesValidation	OnserverClick
<input type="button"> <input type="submit"> <input type="reset">	HtmlInput Button	Name, Type, Value, CausesValidation	OnserverClick
<input type="text"> <input type="pwd">	HtmlInput Text	MaxLength, Name, Size, Type, Value	OnServer Change
<input type="checkbox">	HtmlInput CheckBox	Checked, Name, Type, Value	OnServer Change
<input type="radio">	HtmlInput RadioButton	Checked, Name, Type, Value	OnServer Change
<input type="image">	HtmlInput Image	Align, Alt, Border, Name, Src, Type, Value, CausesValidation	OnServerClick
<input type="file">	HtmlInput File	Accept, MaxLength, Name, PostedFile, Size, Type, Value	None

HTML Server Controls

2.2. Các lớp dữ liệu

Các điều khiển bao gồm thuộc tính, phương thức và các hỗ trợ sự kiện khác nhau

<code><input type="hidden"></code>	HtmlInputHidden	Name, Type, Value	OnServer Change
<code><textarea></textarea></code>	HtmlTextArea	Cols, Rows, Name, Value	OnServer Change
<code><select></select></code>	HtmlSelect	Multiple, SelectedIndex, Size, Value, DataTextField, DataSource, DataValueField	OnServer Change
<code><table></table></code>	HtmlTable	Align, Border, BgColor, BorderColor, Width, Height Rows (collection)	None
<code><tr></tr></code>	HtmlTableRow	Valign,... Cells (collection)	None
<code><td></td></code>	HtmlTableCell	Colspan, RowSpan, Width, Height, NoWrap	None

Các thành phần khác mà chúng không có các lớp dữ liệu định nghĩa tương ứng thì Asp.Net tự động thay thế bởi một lớp dữ liệu **HtmlGenericControl**. là một lớp dữ liệu public kế thừa từ lớp **HtmlContainerControl**

HTML Server Controls

Ví dụ: HtmlServerControl

Tệp .aspx

```
<%@ Page Language="C#" CodeFile="Tệp.cs" Inherits="MyPage"%>
<html>
<head runat="server">
    <title>Basic ASP.NET Web Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <h1>Welcome to ASP.NET</h1>
        <p>Type your name and click the button.</p>
        <p><input type="" text ID="TextBox1" runat="server"/>
        <input type="button" ID="Button1" runat="server"
            Value="Click" OnServerClick="Button1_Click"/>

        </p>
        <p> <div ID="Label1" runat="server"/></p>
    </form>
</body>
</html>
```

Tệp .cs

```
using System;
using System.Web;
using System.Web.UI;
using System.Web.UI.HtmlControls;

public partial class MyPage : System.Web.UI.Page {
    void Button1_Click(object sender, System.EventArgs e) {
        Label1.Text = ("Welcome, " + TextBox1.Value);
    }
}
```

ASP.NET Web Server Controls

3. ASP.NET Web Server Controls

ASP.NET Web Server Controls được định nghĩa trong Namespace **System.Web.UI.WebControls**.

Bộ điều khiển được xây dựng bởi ASP.NET sử dụng một tập hợp các thuộc tính chuẩn giúp cho việc thiết kế và lập trình đơn giản hơn. Dạng thức:

<%@ Import Namespace = "System.Web.UI.WebControls" %>

Design – Time: Trong phần giao diện (*.aspx)

```
<form runat="server">  
    <asp:ServerControlClass id="ID" runat="server" attribute="value"/>  
    <asp:ServerControlClass id="ID" runat="server" attribute="value">  
        Text  
    </asp:ServerControlClass>  
</form>
```

Run – Time: Trong chương trình (*.cs)

```
<script runat="server">  
    ASPControlClass ID = new ASPControlClass();  
    ID.attribute = value;  
</script>
```

ASP.NET Web Server Controls

3. ASP.NET Web Server Controls

- Các Web Form controls được thiết kế với cấu trúc giao diện thống nhất. Khác với HTML control, tất cả các Web Form controls sử dụng chung Text cho thuộc tính “value”.
- Web Form Controls sử dụng cùng một tập các thuộc tính và chúng sẽ tự động tạo ra các thành phần tương ứng của HTML liên quan đến dữ liệu HTML. Ví dụ, chúng tự động tạo text label cho các CheckBox và Radio.
- ASP.NET Web Form Controls được thiết kế với 2 đặc tính so với các HTML controls là:
 - Tự động xây dựng các thành phần Form
 - Đơn giản hoá việc xây dựng, thiết kế các Web Form

ASP.NET Web Server Controls

3.1. Lớp cơ sở

WebControl: Là lớp cơ sở gốc

Thuộc Tính

Attributes Tập hợp thuộc tính name/value của điều khiển. Except custom

ClientID Trả lại ID của điều khiển tạo ra bởi ASP.NET

Controls Trả lại đối tượng ControlCollection gồm tất cả các điều khiển con

Disabled Boolean. Thiết lập kích hoạt của điều khiển

EnableViewState Boolean. Thiết lập viewstate của điều khiển và các con

ID Thiết lập ID của control

Page Đối tượng Page chứa điều khiển

Parent Điều khiển mẹ, chứa điều khiển hiện hành

Style Tập hợp các thuộc tính CSS

TagName Tên thẻ của thành phần

Visible Boolean. Thiết lập sự hiển thị

Phương Thức

DataBind() Kết hợp dữ liệu với điều khiển và các con

FindControl() Tìm điều khiển được chứa trong điều khiển hiện hành

HasControls() Boolean. Kiểm tra xem có các điều khiển con

Sự Kiện

DataBinding Sự kiện xảy ra khi dữ liệu được kết hợp với điều khiển

ASP.NET Web Server Controls

3.2. Các lớp dữ liệu

Điều khiển Web Form Controls tương đương với các thành phần form

<code><asp:Hyperlink></code>	<code><a>...</code>
<code><asp:LinkButton></code>	<code><a></code>
<code><asp:Image></code>	<code></code>
<code><asp:Panel></code>	<code><div>...</div></code>
<code><asp:Label></code>	<code>...</code>
<code><asp:Button></code>	<code><input type="submit"> <input type="button"></code>
<code><asp:TextBox></code>	<code><input type="text"> <input type="password"> <textarea></code>
<code><asp:CheckBox></code>	<code><input type="checkbox"></code>
<code><asp:RadioButton></code>	<code><input type="radio"></code>
<code><asp:ImageButton></code>	<code><input type="image"></code>
<code><asp:Table></code>	<code><table>...</table></code>
<code><asp:TableRow></code>	<code><tr>...</tr></code>
<code><asp:TableCell></code>	<code><td>...</td></code>

ASP.NET Web Controls còn bao gồm các bộ điều khiển khác nhau hỗ trợ thực hiện các tính năng khác nhau như List Controls, Login Controls, DataControls ...

ASP.NET Web Server Controls

3.2. Các lớp dữ liệu

Các lớp dữ liệu trong Web Form Controls sử dụng lúc run-time

Control	Properties	Event
HyperLink	ImageUrl, NavigateUrl, Target, Text	None
LinkButton	CommandArgument, CommandName, Text, CausesValidation	OnClick, onCommand
Image	AlternateText, ImageAlign, ImageUrl	None
Panel	BackImageUrl, HorizontalAlign, WrapText	None
Label	Text	None
Button	CommandArgument, CommandName, Text, CausesValidation	OnClick, onCommand
TextBox	AutoPostBack, Columns, MaxLength, ReadOnly, Rows, Text, TextMode, Wrap	OnTextChanged
CheckBox	AutoPostBack, Checked, Text, TextAlign	OnCheckChanged
RadioButton	AutoPostBack, Checked, GroupName, Text, TextAlign	OnCheckChanged
Image	ImageUrl, AlternateText	
ImageMap	HotSpot (Rectangle,CircleHotSpot)	OnClick, onCommand
ImageButton	CommandArgument, CommandName, CausesValidation	OnClick, onCommand
Table	BackImageUrl, CellPadding, CellSpacing, GridLines, HorizontalAlign, Rows	None
TableRow	Cells, HorizontalAlign, VerticalAlign	None
TableCell	ColumSpan, RowSpan, HorizontalAlign, VerticalAlign, Text, Wrap	None
Literal	Text	None
Placeholder	None	None
MultiView	ActiveViewIndex	ActiveViewChanged
View	Visible	Activate, Deactivate

ASP.NET Web Server Controls

3.3. Sử dụng thuộc tính Web Control

Khi sử dụng ASP.NET Web Form Control, ta định nghĩa chúng trong trang Web như các thành phần của HTML và thiết lập các thuộc tính lúc **design – time** tương tự như HtmlControl.

Điểm khác biệt giữa Web Form controls và HTML controls là thuộc tính của các điều khiển HTML chỉ là dữ liệu kiểu String nhưng thuộc tính của các điều khiển Web Form thường sử dụng dữ liệu kiểu khác nhau trong lúc **run – time**

- **Kiểu Enumeration** liệt kê các giá trị
- **Kiểu Unit** giá trị kích thước độ đo
- **Kiểu đối tượng Color** giá trị màu sắc

ASP.NET Web Server Controls

3.3. Sử dụng thuộc tính Web Control

Khi sử dụng ASP.NET Web Form Control, ta định nghĩa chúng trong trang Web như các thành phần của HTML và thiết lập các thuộc tính lúc **design – time** tương tự như HtmlControl.

Điểm khác biệt giữa Web Form controls và HTML controls là thuộc tính của các điều khiển HTML chỉ là dữ liệu kiểu String nhưng thuộc tính của các điều khiển Web Form thường sử dụng dữ liệu kiểu khác nhau trong lúc **run – time**

- | | |
|---------------------------|---|
| - Kiểu Enumeration | Kiểu liệt kê các giá trị |
| - Kiểu Unit | Kiểu kích thước độ đo |
| - Kiểu Color | Kiểu màu sắc |
| - Kiểu Object | Các kiểu dữ liệu đối tượng thông thường |

ASP.NET Web Server Controls

3.3. Sử dụng thuộc tính Web Control

Enumeration

Hầu hết các enumeration đều cung cấp phương thức chuyển đổi kiểu dữ liệu của đối tượng **TypeDescriptor** được định nghĩa trong namespace **System.ComponentModel**.

Ví dụ: Với kiểu Enumeration ImageAlign (design-time)

```
<asp:DropDownList id="selAlign" runat="server">
    <asp:ListItem value="0">ImageAlign.NotSet</asp:ListItem>
    <asp:ListItem value="1">ImageAlign.Left</asp:ListItem>
    <asp:ListItem value="2">ImageAlign.Right</asp:ListItem>
    <asp:ListItem value="3">ImageAlign.BaseLine</asp:ListItem>
    <asp:ListItem value="4">ImageAlign.Top</asp:ListItem>
    <asp:ListItem value="5">ImageAlign.Middle</asp:ListItem>
    <asp:ListItem value="6">ImageAlign.Bottom</asp:ListItem>
    <asp:ListItem value="7">ImageAlign.AbsBottom</asp:ListItem>
    <asp:ListItem value="8">ImageAlign.AbsMiddle</asp:ListItem>
    <asp:ListItem value="9">ImageAlign.TextTop</asp:ListItem>
</select>
```

ASP.NET Web Server Controls

3.3. Sử dụng thuộc tính Web Control

Enumeration

Khi sử dụng run – time sẽ phải là

```
<script runat="server">  
    objImage.ImageAlign = ImageAlign.Right;  
</script>
```

Ta có thể sử dụng lớp TypeDescriptor để ép kiểu

```
<%@ Import Namespace="System.ComponentModel" %>  
    TypeDescriptor.GetConverter(GetType(HorizontalAlign)).  
        ConvertFromString("Left");
```

Hoặc

```
Value = CType(HorizontalAlign.Left, Integer);
```

ASP.NET Web Server Controls

3.3. Sử dụng thuộc tính Web Control

Unit

Một số thuộc tính của ASP Web Form controls sử dụng dữ liệu của đối tượng **Unit** khi thiết lập giá trị tại thời điểm run-time ta phải chuyển đổi giá trị dữ liệu

Ví dụ:

```
<asp:Image id="MyImage" Src="MyPic.gif" runat="server" Height="100px" Width="50%" />
```

Lớp dữ liệu định nghĩa đối tượng Unit là một bộ phận của namespace Web Form control nên được tự động nạp. Đối tượng Unit cung cấp **2 thuộc tính**

Type	Kiểu dữ liệu sử dụng để xác định độ đo. Thuộc vào một trong UnitType Enumeration (Cm, Mm, Em, Ex, Inch, Percentage, Pica, Pixel, Point)
Value	Giá trị số của kiểu dữ liệu

3 phương thức

Percentage	Tạo đối tượng Unit có kiểu Percentage sử dụng 32-bit signed integer
Pixel	Tạo đối tượng Unit có kiểu Pixel sử dụng 32-bit signed integer
Point	Tạo đối tượng Unit có kiểu Point sử dụng 32-bit signed integer

ASP.NET Web Server Controls

3.3. Sử dụng thuộc tính Web Control

Color

Sử dụng giá trị là các đối tượng. Ví dụ khi ta sử dụng các thuộc tính BackColor, ForeColor, BorderColor ... là những đối tượng của lớp **System.Drawing.Color** mà không phải là giá trị String. Khi sử dụng tại design-time ta có thể thiết lập trực tiếp màu chữ và điều khiển Web Form sẽ tự động chuyển đổi nhưng trong run – time ta phải xác định giá trị màu

Ví dụ:

```
<asp:Textbox id="MyTextBox" Text="Đây là textbox" runat="server"  
        BackColor="Red" ForeColor="White" />
```

System.Drawing cung cấp các thuộc tính tên màu cho tất cả các thành phần HTML và 3 phương thức để tạo các đối tượng Color

FromArgb	Tạo đối tượng Color 32-bit xác định các thành phần alpha, red, green, blue
FromKnownColor	Tạo đối tượng Color từ tên chuẩn của HTML
FromName	Tạo đối tượng Color từ tên chuẩn, 32-bit hoặc s

ASP.NET Web Server Controls

Ví dụ: ASP.NET Web Server Control

Tệp .aspx

```
<%@ Page Language="C#" CodeFile="Tệp.cs" Inherits="MyPage"%>
<html>
<head runat="server">
    <title>Basic ASP.NET Web Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <h1>Welcome to ASP.NET</h1>
        <p>Type your name and click the button.</p>
        <p><asp:TextBox ID="TextBox1" runat="server">
</asp:TextBox>
        <asp:Button ID="Button1" runat="server" Text="Click"
OnClick="Button1_Click"/>
        </p>
        <p><asp:Label ID="Label1"
runat="server"></asp:Label> </p>
    </form>
</body>
</html>
```

Tệp .cs

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

public partial class MyPage : System.Web.UI.Page {
    void Button1_Click(object sender, System.EventArgs e) {
        Label1.Text = ("Welcome, " + TextBox1.Text);
        Label1.BackColor = Color.FromName("Blue");
    }
}
```


Validation Controls

4. Validation Controls

- Không phải là các điều khiển dữ liệu mà là điều khiển dùng để kiểm tra dữ liệu trong các điều khiển khi trang Web được gửi cho server để đảm bảo tính an toàn.
- Nguyên tắc thực hiện của các Validator Controls là ta sẽ kết hợp chúng với các điều khiển nhập liệu cần kiểm tra. Các validator sẽ kiểm tra dữ liệu trong điều khiển và nếu thấy có lỗi thì sẽ hiển thị qua ValidatorSummary.
- Validators tự động kiểm tra trình duyệt và thiết bị đầu vào và sẽ tạo ra các hàm kiểm tra client-side tương ứng. Validator thực hiện việc kiểm tra dữ liệu trên cả client và server khi trang Web được gửi đi để tránh trường hợp phần kiểm tra client bị thay đổi. Chúng ta có thể tắt chức năng kiểm tra trên client của validator nếu không cần thiết.

Validation Controls

4. Validation Controls

Validators thường chỉ sử dụng trong design – time, bao gồm:

<ASP:RequireFieldValidator>	Kiểm tra dữ liệu vào của các điều khiển, phải là khác trống. Có thể sử dụng kết hợp với các validator khác
<ASP:RangeValidator>	Kiểm tra dữ liệu vào của điều khiển phải nằm trong phạm vi text hoặc số. Nếu dữ liệu control cần kiểm tra trống thì sẽ không thực hiện
<ASP:CompareValidator>	Kiểm tra dữ liệu vào với dữ liệu của controls khác hoặc với giá trị xác định. Kiểu dữ liệu cũng có thể kiểm tra. Nếu dữ liệu control cần kiểm tra trống thì sẽ không thực hiện
<ASP:RegularExpressionValidator>	Kiểm tra dữ liệu vào với một chuỗi RegularExpression. Nếu dữ liệu control cần kiểm tra trống thì sẽ không thực hiện
<ASP:CustomValidator>	Thực hiện các hàm kiểm tra do ta xây dựng (client-side và server-side)
<ASP:ValidatorSummary>	Hiển thị các thông báo lỗi của tất cả các validator controls

Validation Controls

4.1. Lớp cơ sở BaseValidator

Tất cả các Validator Control đều kế thừa từ lớp cơ sở BaseValidator (System.Web.UI.WebControls.BaseValidator). Các thuộc tính và phương thức của BaseValidator:

ControlToValidate	Thiết lập hoặc trả lại tên của điều khiển nhập liệu cần kiểm tra
EnableClientScript	Boolean, sử dụng chức năng kiểm tra trên Client
Enabled	Boolean, sử dụng validator hay không
ErrorMessage	Thiết lập text của thông báo lỗi hiển thị bởi ValidatorSummary
IsValid	Boolean, dữ liệu hợp lệ hay không
Validate()	Thực hiện kiểm tra dữ liệu và cập nhật thuộc tính IsValid

Validation Controls

4.2. Các lớp dữ liệu Validator

Các thuộc tính và sự kiện của các lớp Validator:

Control	Properties	Event
RequireFieldValidator	InitialValue	None
RangeValidator	MaximumValue, MinimumValue, Type	None
CompareValidator	ControlToCompare, Operator, Type, ValueToCompare	None
RegularExpressionValidator	ValidationExpression	None
CustomValidator	ClientValidationFunction	OnServerValidate
ValidatorSummary	DisplayMode, ShowHeaderText, ShowMessageBox, ShowSummary	None

Validation Controls

4.2. Các lớp dữ liệu Validator

Ví dụ:

A required field value:

```
<input type="text" id="txtRequired" runat="server" />
<asp:RequiredFieldValidator id="valRequired"
runat="server"
ControlToValidate="txtRequired"
ErrorMessage="* You must enter a value in the
textbox"
Display="dynamic"> * </asp:RequiredFieldValidator>
```

The same value again

```
<input type="text" id="txtCompare" runat="server" />
<asp:CompareValidator id="valCompare"
runat="server"
ControlToValidate="txtCompare"
ControlToCompare="txtRequired"
Operator="Equal"
ErrorMessage="* You must enter the same value in the
textbox"
Display="dynamic">* </asp:CompareValidator>
```

A Date after 3rd March 2006

```
<input type="text" id="txtCompareDate"
runat="server" />
<asp:CompareValidator id="valCompareDate"
runat="server"
ControlToValidate="txtCompareDate"
ValueToCompare="3/3/2006"
Operator="GreaterThan"
Type="Date"
ErrorMessage="* the Date must be later than 3rd
March 2006"
Display="dynamic"> * </asp:CompareValidator>
```

A number between 1 and 10

```
<input type="text" id="txtRange" runat="server" />
<asp:RangeValidator id="valRange" runat="server"
ControlToValidate="txtRange"
MaximumValue="10"
MinimumValue="1"
Type="Integer"
ErrorMessage="* the Number must between 1 and 10"
Display="dynamic"> * </asp:RangeValidator>
```

Validation Controls

4.2. Các lớp dữ liệu Validator

Ví dụ:

```
Match Expression "<b>.*@.*\..*</b>"
<input type="text" id="txtRegExpr" runat="server" />
<asp:RegularExpressionValidator id="valRegExpr"
runat="server"
ControlToValidate="txtRegExpr"
ValidationExpression=".*@.*\..*"
ErrorMessage="* Your entry does not match the
regular expression"
Display="dynamic"> *
</asp:RegularExpressionValidator>
```

```
A Prime number over 100
<input type="text" id="txtCustom" runat="server" />
<asp:CompareValidator id="valComparePrime"
runat="server"
ControlToValidate="txtCustom"
ValueToCompare="100"
Operator="GreaterThan"
```

```
Type="Integer"
ErrorMessage="* the Prime number must be greater
than 100"
Display="dynamic"> * </asp:CompareValidator>
<asp:CustomValidator id="valCustom" runat="server"
ControlToValidate="txtCustom"
ClientValidationFunction="ClientValidate"
OnServerValidate="ServerValidate"
ErrorMessage="* it is not a Prime number"
Display="dynamic"> * </asp:CustomValidator>

<asp:ValidationSummary id="valSummary"
runat="server"
HeaderText="<b>The following error were
found:</b>"
ShowSummary="true" DisplayMode="List"
/>
DisplayMode: List, BulletList, SingleParagraph
```

Validation Controls

4.2. Các lớp dữ liệu Validator

Ví dụ:

```
<script language="JavaScript">
function ClientValidate(objSource, objArgs) {
    var blnValid = true;
    var intNumber = objArgs.Value;
    if (intNumber % 2 == 1) {
        var intDivisor = Math.floor(intNumber/3);
        if (intDivisor > 2) {
            for (var i=3; i<=intDivisor; i=i+2) {
                if (intNumber % intDivisor == 0) {
                    blnValue = false;
                    break;
                }
            }
        }
    }
    else
        blnValue = false;
}
```

```
    else
        blnValue = false;
    objArgs.IsValid = blnValue;
    return;
}
</script>
```

```
<script runat="server">
protected ServerValidate (object s, EventArgs e) {

    // Logic tương tự hàm Client

}
</script>
```

Rich Controls

5. Rich Controls

ASP.NET cung cấp các bộ điều khiển đóng gói hỗ trợ các chức năng trên web. Chúng hoạt động như những chương trình con với đầy đủ tính năng được thiết lập sẵn. Các loại điều khiển Rich Controls:

- **Ad Rotator:** điều khiển nội dung quảng cáo
- **Calendar:** Lịch làm việc
- **FileUpload:** Upload tệp tin
- **MultiView:** Đa chế độ hiển thị
- **Wizard:** Tương tự như Multiview với chức năng thay đổi

Rich Controls

5.1. Ad Rotator Control

Ad Rotator cho phép hiển thị các nội dung quảng cáo một cách ngẫu nhiên trên trang web. Nội dung cần hiển thị có thể chứa trong tệp XML hoặc CSDL

Cú pháp:

```
<asp:AdRotator id="ID" AdvertisementFile="XML/DBFile"  
KeywordFilter="keywords" Target="Name"  
OnAdCreated = "Method" runat="server" />
```

Thuộc tính:

- KeywordFilter: Từ khoá cho các nội dung quảng cáo
- Target: Tên cửa sổ, khung sẽ hiển thị quảng cáo
- OnAdCreated: Thực hiện khi sự kiện AdCreated phát sinh để tạo control

Rich Controls

5.1. Ad Rotator Control

Thuộc tính:

- AdvertisementFile: Tập XML hoặc DBFile có cấu trúc như sau

```
<? xml version="1.0" encoding="utf-8" ?>
```

```
<Advertisements>
```

```
<Ad>
```

```
    <ImageUrl> Bắt buộc </ImageUrl>
```

```
    <NavigateUrl>Link đến nội dung quảng cáo</NavigateUrl>
```

```
    <AlternateText>Text thay thế</AlternateText>
```

```
    <Impressions>Tần suất xuất hiện</Impressions>
```

```
    <Keyword>Phân loại keyword</Keyword>
```

```
    <Caption>Caption</Caption>
```

```
</Ad>
```

```
</Advertisement>
```

Rich Controls

MyAds.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<Advertisements>
  <Ad>
    <ImageUrl>logo1.png</ImageUrl>
    <NavigateUrl>
      http://www.abc.com
    </NavigateUrl>
    <AlternateText> QC1 </AlternateText>
    <Impressions>100</Impressions>
    <Keyword>banner</Keyword>
  </Ad>
  <Ad>
    <ImageUrl>logo2.png</ImageUrl>
    <NavigateUrl>
      http://www.123.com
    </NavigateUrl>
    <AlternateText>QC2 </AlternateText>
    <Impressions>80</Impressions>
    <Keyword>banner</Keyword>
  </Ad>
</Advertisements>
```

MyAd.aspx

```
<form runat="server">
  <h3>AdRotator Example</h3>
  <asp:AdRotator id="test1" runat="server"
    AdvertisementFile = "MyAds.xml"
    Borderwidth="1"
    Target="_newwindow"
    OnAdCreated="AdCreated_Event" />
  <br/>
  <asp:label id="Message" runat="server" />
</form>

<script language="C#" runat="server">
  void AdCreated_Event(Object sender,
    AdCreatedEventArgs e) {
    Message.Text=e.NavigateUrl;
  }
</script>
```

Rich Controls

5.2. Calendar Control

Tạo lịch làm việc, cung cấp chức năng lựa chọn ngày, thứ, tuần, năm và thiết lập giao diện lịch

Cú pháp:

```
<asp:Calendar id="ID" attribute= "value" runat="server" />
```

Sự kiện:

- SelectionChanged: Khi lựa chọn ngày, tuần, tháng
- DayRender: Khi các ô trong lịch được hiển thị
- VisibleMonthChanged: Thay đổi tháng

Rich Controls

MyCalendar.aspx

```
<form runat="server">
<asp:Calendar id="Calendar1"
    OnSelectionChanged="Date_Selected"
    SelectionMode="DayWeekMonth"
    Font-Name="Verdana"
    Font-Size="12px"
    NextPrevFormat="ShortMonth"
    SelectWeekText="week"
    SelectMonthText="month"
    runat="server">

    <TodayDayStyle Font-Bold="True"/>
    <DayHeaderStyle Font-Bold="True"/>
    <OtherMonthDayStyle ForeColor="gray"/>
    <TitleStyle BackColor="#3366ff" ForeColor="white"
        Font-Bold="True"/>
    <SelectedDayStyle BackColor="#ffcc66" Font-
        Bold="True"/>
    <NextPrevStyle ForeColor="white" Font-
        Size="10px"/>
    <SelectorStyle BackColor="#99ccff"
        ForeColor="navy" Font-Size="9px"/>
</asp:Calendar>
</form>

<script runat="server">
    protected void Date_Selected (object sender,
        EventArgs e)
    {
        Label1.Text ="Todays date is: "+
            Calendar1.TodaysDate.ToShortDateString();
        Label2.Text = "You date of birth is: " +
            Calendar1.SelectedDate.
                ToShortDateString();
    }
</script>
```

Rich Controls

5.2. Calendar Control

<div><div>≤</div><div>December 2016</div><div>≥</div></div>						
Su	Mo	Tu	We	Th	Fr	Sa
<u>27</u>	<u>28</u>	<u>29</u>	<u>30</u>	<u>1</u>	<u>2</u>	<u>3</u>
<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>
<u>11</u>	<u>12</u>	<u>13</u>	<u>14</u>	<u>15</u>	<u>16</u>	<u>17</u>
<u>18</u>	<u>19</u>	<u>20</u>	<u>21</u>	<u>22</u>	<u>23</u>	<u>24</u>
<u>25</u>	<u>26</u>	<u>27</u>	<u>28</u>	<u>29</u>	<u>30</u>	<u>31</u>
<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>
Today's date is: 12/15/2016						
Your date of birth is: 12/8/2016						

Rich Controls

5.3. MultiView Control

Cho phép hiển thị thông tin trên trang theo dạng tab, ta có thể tạo nhiều sub form. MultiView được tạo bởi một tập hợp View Controls, trong mỗi View ta có thể đặt nhiều điều khiển khác nhau. Tại mỗi thời điểm có 1 View được kích hoạt hiển thị gọi là active view

Cú pháp:

```
<asp:MultiView id="ID" attribute= "value" runat="server" />
```

Thuộc tính:

- **ActiveViewIndex**: Chỉ số view kích hoạt hiển thị
- **Views**: Tập hợp các View Controls

Về tìm hiểu thêm: FileUpload Control và Wizard Control

Rich Controls

5.3. MultiView Control

MultiView1	
View1	
Step 1 - Product Details Product ID <input type="text"/> Product Name <input type="text"/> Price/Unit <input type="text"/> <input type="button" value="Next >>"/>	
View2	
Step 2 - Order Details Order ID <input type="text"/> Quantity <input type="text"/> <input type="button" value=" << Previous"/> <input type="button" value="Next >>"/>	
View3	
Step 3 - Summary Product Details Product ID : [lblProductID] Product Name : [lblProductName] Price/Unit : [lblPrice] Order Details Order ID : [lblOrderID] Quantity : [lblQuantity] <input type="button" value=" <<Previous"/> <input type="button" value="Submit >>"/>	

List Controls

6. List Controls

List Control sử dụng để tạo các điều khiển với dữ liệu dạng tập hợp với nhiều tùy chọn cho phép người dùng lựa chọn. ASP.NET cung cấp các loại điều khiển danh sách gồm:

- **DropDownList:** Tạo thẻ <select> với 1 lựa chọn
- **ListBox:** Tạo thẻ <select> với nhiều lựa chọn
- **CheckBoxList:** Tập hợp checkbox dạng bảng hoặc danh sách
- **RadioButtonList:** Tập hợp radio dạng bảng hoặc danh sách
- **BulletList:** Tập hợp danh mục không có thứ tự

Chú ý: ListItem được sử dụng cho các điều khiển danh sách

List Controls

6.1. Lớp cơ sở ListControl

ListControl: Là lớp cơ sở gốc kế thừa từ lớp WebControl

Thuộc Tính:

AutoPostBack

Tự động gửi cho server khi user thay đổi mục chọn danh sách

**DataMember
DataSource**

Tên bảng dữ liệu trong Dữ liệu nguồn dùng để nạp dữ liệu cho các mục danh sách

DataTextField

Tên trường của dữ liệu nguồn dùng để tạo ra text của mục danh sách

DataTextFormatString

Thiết lập chuỗi định dạng của dữ liệu sẽ được hiển thị. Ví dụ {0:C} để hiển thị tiền tệ

DataValueField

Tên trường của dữ liệu nguồn dùng để tạo ra value của mục danh sách

Items

Trả lại tập hợp các mục trong list control

SelectedIndex

Chỉ số của mục chọn trong danh sách. Nếu danh sách thuộc loại chọn nhiều thành phần thì dùng Selected của từng mục
Trả lại đối tượng là mục chọn.

SelectedItem

Sự kiện:

OnSelectedIndexChanged

Sự kiện xảy ra trên server khi có thay đổi mục chọn và trang Web được gửi trả lại cho server

List Controls

6.2. Các lớp dữ liệu

Control	Properties	Methods
DropDownList	Items	None
ListBox	Rows, SelectionMode	None
ListItem	Attributes, Selected, Text, Value	FromString
CheckBoxList	CellPadding, CellSpacing, RepeatColumns, TextAlign RepeatDirection (Horizontal, Vertical) RepeatLayout (Table, Flow, Un/OrderedList)	None
RadioButtonList	CellPadding, CellSpacing, RepeatColumns, RepeatDirection, RepeatLayout, TextAlign	None
BulletList	BulletStyle (NotSet, Circle, Square, Numbered, ...) DisplayMode (HyperLink, LinkButton, Text)	None

User Controls

7. User Controls

- User Controls là thành phần tách biệt được đóng gói để tái sử dụng trong các trang Web khác nhau. User Controls được sử dụng giống như các điều khiển server
- Ưu điểm của UC là chúng có thể được Cache trong bộ nhớ trình duyệt bởi OutputCache Directive. Khi đó toàn bộ trang Web được reload sau mỗi Request còn UC được giữ nguyên
- UC hoạt động tương tự như một trang Web nhưng chúng được kế thừa từ lớp **System.Web.UI.UserControl** không hoạt động qua Request/Response mà phải đăng ký để sử dụng trong các trang aspx
- User Controls và Custom Controls có thể được tạo mới từ đầu hoặc có thể được chuyển đổi từ các trang ASP.NET có sẵn thành các điều khiển

User Controls

7.1. Tạo mới UC

UC là điều khiển tương tự như các trang ASP.NET, có thể chứa các thành phần Client Script, HTML, mã ASP.Net và các loại điều khiển server nhưng được xây dựng với các điểm khác biệt sau:

- UC được tạo ra với tệp **.ascx** và cần phải đăng ký sử dụng trong các trang Web với chỉ thị
- **<%@ Register TagPrefix = “...” TagName=“...” Scr=“.ascx” %>**
- Một điều khiển UC không có các thành phần thẻ <HTML>, <BODY>, và <FORM>
- User control không được phép gọi từ Web Request mà phải sử dụng trong các trang Web khác

User Controls

7.1. Tạo mới UC

Ví dụ:

header.ascx

```
<h1>Hello</h1>
```

```
<asp:TextBox id="txtText1" runat="server" />
```

Page.aspx

```
<%@ Register TagPrefix="TTT" Tagname="Hello" Src="header.ascx" %>
```

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

```
    <TTT:Hello runat="server"/>
```

```
</form>
```

User Controls

7.1. Tạo mới UC

Ví dụ:

header.ascx

```
<h1>Hello</h1>
```

```
<asp:TextBox id="txtText1" runat="server" />
```

Page.aspx

```
<%@ Register TagPrefix="TTT" Tagname="Hello" Src="header.ascx" %>
```

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

```
    <TTT:Hello runat="server"/>
```

```
</form>
```

User Controls

7.2. Tạo từ Page

Chuyển đổi một trang Web thành điều khiển là phương pháp khác đơn giản hơn. Khi xây dựng các ứng dụng Web có nhiều phần chương trình hoặc giao diện được sử dụng trong nhiều trang khác nhau cần phải test nghiệp vụ logic

- Tạo tệp **.aspx** để test các chức năng
- Đổi tên tệp thành **.ascx** và thay chỉ thị **<%@ Page %>** bởi chỉ thị **<%@ Control %>**. Bỏ các thành phần thẻ **<HTML><BODY><FORM>**
- Đăng ký để sử dụng trong các trang Web
<%@ Register TagPrefix = “...” TagName=“...” Scr=“.ascx” %>

List Controls

Control.aspx

```
<%@ Page Language="C#"
    AutoEventWireup="true"
    CodeFile="Control.aspx.cs"
    Inherits="VaryByControl" %>
<%@ OutputCache Duration="30"
    VaryByControl="lstMode" %>
<form runat="server">
<asp:DropDownList id="lstMode"
    runat="server" Width="187px">
    <asp:ListItem>Large</asp:ListItem>
    <asp:ListItem>Small</asp:ListItem>
    <asp:ListItem>Medium</asp:ListItem>
</asp:DropDownList>&nbsp;<br />
Control được tạo lúc:<br />
<asp:label id="TimeMsg" runat="server" />
</form>
```

Control.aspx.cs

```
using System;
using System.Web;
```

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

```
public partial class VaryByControl : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        switch (lstMode.SelectedIndex)
        {
            case 0:
                TimeMsg.Font.Size = FontUnit.Large;
                break;
            case 1:
                TimeMsg.Font.Size = FontUnit.Small;
                break;
            case 2:
                TimeMsg.Font.Size = FontUnit.Medium;
                break;
        }
        TimeMsg.Text = DateTime.Now.ToString("F");
    }
}
```

List Controls

Control.ascx

```
<%@ Control Language="C#"
    AutoEventWireup="true"
    CodeFile="Control.ascx.cs"
    Inherits="VaryByControl" %>
<%@ OutputCache Duration="30"
    VaryByControl="lstMode" %>

<asp:DropDownList id="lstMode"
    runat="server" Width="187px">
    <asp:ListItem>Large</asp:ListItem>
    <asp:ListItem>Small</asp:ListItem>
    <asp:ListItem>Medium</asp:ListItem>
</asp:DropDownList>&nbsp;<br />
Control được tạo lúc:<br />
<asp:label id="TimeMsg" runat="server" />
```

Control.aspx.cs

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

```
public partial class VaryByControl : System.Web.UI.User
Control
{
    protected void Page_Load(object sender, EventArgs e
    )
    {
        switch (lstMode.SelectedIndex)
        {
            case 0:
                TimeMsg.Font.Size = FontUnit.Large;
                break;
            case 1:
                TimeMsg.Font.Size = FontUnit.Small;
                break;
            case 2:
                TimeMsg.Font.Size = FontUnit.Medium;
                break;
        }
        TimeMsg.Text = DateTime.Now.ToString("F");
    }
}
```

User Controls

7.2. Tạo từ Page

Page.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" %>
<%@ Register Src="Control.ascx" TagName="VaryByControl" TagPrefix="uc1" %>
<html>
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <uc1:VaryByControl ID="VaryByControl1" runat="server" />
        </div>
    </form>
</body>
</html>
```

Chú ý: Control có thể được tạo với thuộc tính, sự kiện và có thể được đăng ký trong Assembly hoặc Web.config