

Advanced Programming Language (630501)

Fall 2011/2012 – Lectures Notes # 14

Introduction to ASP.NET

Outline of the Lecture

- Developing Mobile Web Forms.
- Using Multiple Forms in a Single Page.
- ASP.NET Controls.

Developing Mobile Web Forms

- ASP.NET pages ends with an **.aspx** extension.
- **Example**

Example 1 : Single Form ASP.NET Page

```
<%@ Page Inherits="System.Web.UI.MobileControls.MobilePage"
    Language="C#" %>
<%@ Register TagPrefix="Mobile"
    Namespace="System.Web.UI.MobileControls"
    Assembly="System.Web.Mobile" %>
<Mobile:Form id="FormOne" runat="server">
<Mobile:Label runat="server">Welcome to the Microsoft Mobile Internet
Toolkit!</Mobile:Label>
</Mobile:Form>
```

```
<%@ Page Inherits="System.Web.UI.MobileControls.MobilePage"
Language="C#" %>
<%@ Register TagPrefix="Mobile"
Namespace="System.Web.UI.MobileControls"
Assembly="System.Web.Mobile" %>
```

- The **@ Page** directive defines page-specific attributes used by the ASP.NET page parser and compiler.
- The **Inherits** attribute specifies that the page is inherited from the **"System.Web.UI.MobileControls.MobilePage"** class, which itself is inherited from the ASP.NET Page class.



- The **Language** attribute specifies the language to be used in the page. For our example, we have used C#.NET.
- The **@ Register** directive associates aliases with namespaces and class names. In the preceding Mobile Web form, we use the **tagprefix** of **Mobile** to associate with the **System.Web.UI.MobileControls** namespace.
- The **<Mobile:Form>** element. This element acts as a container to group controls together logically.

```
<Mobile:Form id="FormOne" runat="server">
```

In our case, we have a **<Mobile:Label>** control, which simply provides a label for text to be displayed.

```
<Mobile:Label runat="server">Welcome to the Microsoft Mobile Internet  
Toolkit!</Mobile:Label>
```

Using Multiple Forms in a Single Page

- In ASP.NET mobile pages, you can have multiple mobile forms in a Mobile Web page.

Example 2 : Multiple_forms.aspx

```
<%@ Page Inherits="System.Web.UI.MobileControls.MobilePage"  
Language="VB" %>  
<%@ Register TagPrefix="Mobile"  
Namespace="System.Web.UI.MobileControls"  
Assembly="System.Web.Mobile" %>  
<Mobile:Form id="FormOne" runat="server">  
<Mobile:Label runat="server">This is the first form</Mobile:Label>  
<Mobile:Link runat="server" navigateURL="#FormTwo">Goto Form  
Two</Mobile:Link>  
</Mobile:Form>  
<Mobile:Form id="FormTwo" runat="server">  
<Mobile:Label runat="server">This is the second form</Mobile:Label>  
</Mobile:Form>
```

- use the **<Mobile:Link>** control. The **navigateURL** attribute contains the ID of the form to link to. Note that the ID is preceded by a number character (#).



ASP.NET Controls

- ASP.NET has the following kinds of Controls

1. HTML Client Controls.

2. HTML Server Controls.

- The *System.Web.UI.HtmlControls* namespace contains classes that allow you to create HTML server controls on a Web Forms page. HTML server controls run on the server and map directly to standard HTML tags supported by most browsers. This allows you to programmatically control the HTML elements on a Web Forms page.

- An example of an HTML Server control is as follows:

```
<select id="partOfWorld" runat="server">
```

- Notice that HTML Server controls are similar to the normal HTML elements, except that they have the additional **runat** attribute.

- The **id** attribute acts as a unique identifier for the server controls.

3. ASP.NET Server Controls

- Besides the HTML server controls, ASP.NET provides a different set of server controls known as ASP.NET server controls.

- The following is an example of an ASP.NET server controls:

```
<asp:button runat="server" id="button1" onClick="Button1_clicked"  
text="Submit"/>
```

- This ASP.NET server control will render itself as an **<input>** element when viewed using a Web browser. ASP.NET server controls expose properties and events that you can set and service. For example, this ASP.NET server control defines the **onClick** event. When the button is clicked, the **Button1_clicked** method would be serviced (which is covered in the next section).