Advanced Programming Language (630501)
Fall 2011/2012 – Lectures Notes # 15

Introduction to ASP.NET
Mobile Web Forms Controls

Outline of the Lecture

- The Code Components
- ASP.NET Architecture
- Mobile Web Forms Controls

The Code Components

Example 15.1

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

<script language="c#" runat=server>
protected void Page_Load(Object sender, EventArgs e)
{
   Form1.Alignment = Alignment.Center;
   Label1.Text = "Jordan";
   Label2.Text = "Iraq";
}
</script>

<mobile:Form id="Form1" runat="server">
  <mobile:Label id="Label1" runat="server" />
  <mobile:Label id="Label2" runat="server" />
</mobile:Form>
```

Example 15.2

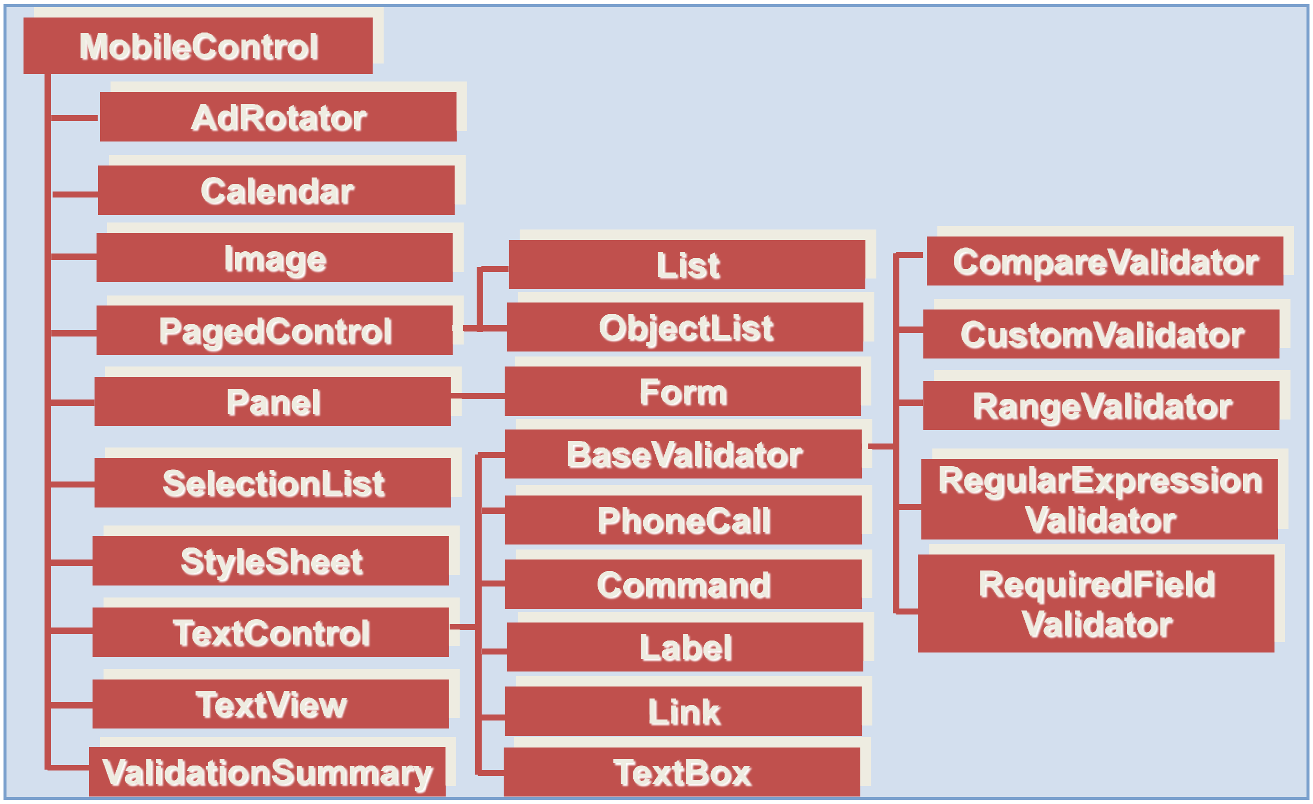
```
<%@ Page Inherits=
   "System.Web.UI.MobileControls.MobilePage" Language="C#" %>
```

```
<%@ Register TagPrefix="mobile"
    Namespace="System.Web.UI.MobileControls"
    Assembly="System.Web.Mobile" %>

<script language="c#" runat="server">
protected void Submit_Click(Object sender, EventArgs e)
{
   WelcomeMessage.Text = "Welcome to my app, " + NameEdit.Text;
   ActiveForm = Form2;
}
</script>

<mobile:Form runat="server">
  <mobile:Label runat="server">Enter your name</mobile:Label>
  <mobile:TextBox runat="server" id="NameEdit" />
  <mobile:Command runat="server" id="Submit"
                  OnClick="Submit_Click"> OK
  </mobile:Command>
</mobile:Form>
<mobile:Form runat="server" id="Form2">
  <mobile:Label runat="server" id="WelcomeMessage" />
</mobile:Form>
```

ASP.NET Architecture

- The Web client first interacts with Internet Information Server (IIS). If the Web client is accessing HTML pages, IIS will communicate with the underlying operating system to fetch the HTML pages. If the Web client is accessing an ASP.NET application, the ASP.NET application will first be compiled to produce a .NET runtime class.
- The .NET runtime class is then compiled and invoked to produce HTML to be sent to the client side.
- One important difference between ASP.NET and ASP is that ASP.NET applications are parsed and compiled once and then cached, so that subsequent requests do not go through the same time-consuming steps. This creates a positive impact on the performance of ASP.NET applications.

Mobile Web Forms Controls



Mobile Web Forms Controls kinds

1. *Container Controls*
2. *Control Transfer Controls: Command Control*
3. *Text Display Controls*
4. *Data-Bound Controls*
5. *Validation Controls*
6. *Specific Controls*

Container Controls

- ***Form Control***
  - Multiple forms per page
  - Container for other controls
  - Only one active form at a time
  - By default – first form active when page accessed
    - Set via **`ActiveForm`**

- ***Panel Controls***
  - Provides grouping for multiple controls.
  - Single control for displaying, hiding, enabling, or disabling a set of controls.
  - Applying styles to panel that are inherited by child controls.

```
<mobile:Panel id="Panel1" runat="server">

    <mobile:Label id="Label1" runat="server">Text…
    </mobile:Label>

</mobile:Panel>
```

Control Transfer Controls: Command Control

- Like ***Button*** control in ASP.NET
- Way to invoke ASP.NET ***event handlers*** from UI elements
- ***SoftkeyLabel*** property
  - Specify text for ***Softkey*** on supporting mobile phones

```
<mobile:Command id="cmdSelectProduct" runat="server“

softkeylabel=“Next”>Select Product</mobile:Command>
```

Text Display Controls

- ***Label*** control
  - Small amount of content – read-only text
- ***TextBox*** control
  - Single-line input text boxes
- ***TextView*** control
  - Large fields of text
  - Small set of markup: bold, italics, page break, paragraph, anchor tag
  - Support pagination