CMER

Centre for Mobile Education and Research

Database Handling, Sessions, and AJAX

Post Back
ASP.NET Functionality

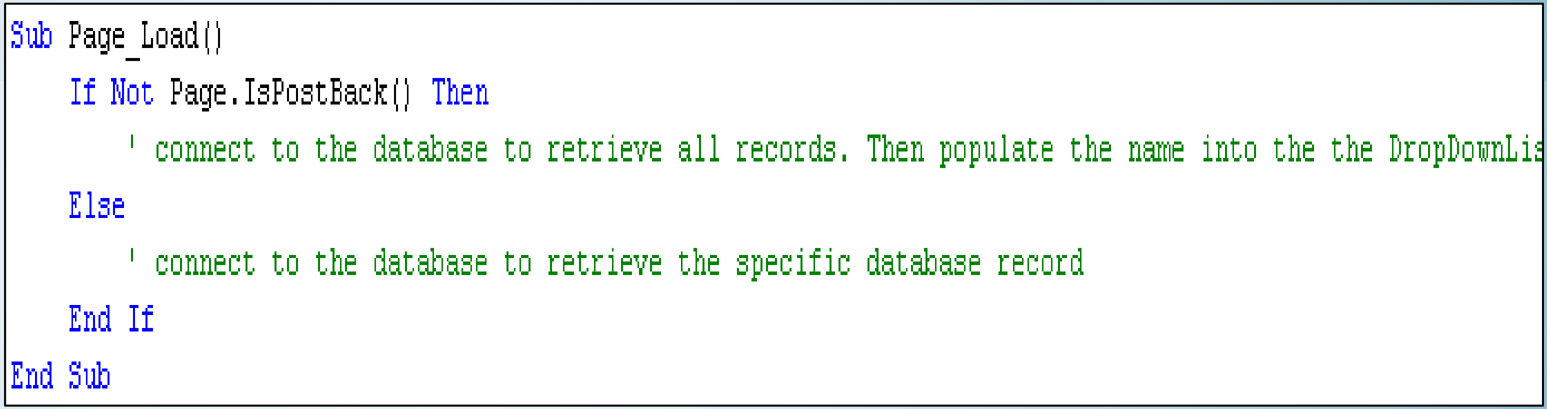
- The IsPostBack method in ASP.NET is similar to the BlackBerry .refresh method
  - IsPostBack determines whether a form is posted to the page or not
    - Everytime you invoke a server control that has a postback functionality, the page is refreshed

IsPostBack Example

- Recall the database example with the DropDownList
  - Assume that we would like to display customer information when a certain customer name is selected from the DropDownList
  - We need to postback the information from the DropDownList and then connect to the database to retrieve relevant record information

IsPostBack Example (continued)

- In the Page_Load event, create a conditional statement to determine whether the page is a post back or not



```
Sub Page_Load()
    If Not Page.IsPostBack() Then
        ' connect to the database to retrieve all records. Then populate the name into the the DropDownLis
    Else
        ' connect to the database to retrieve the specific database record
    End If
End Sub
```

IsPostBack Example (continued)

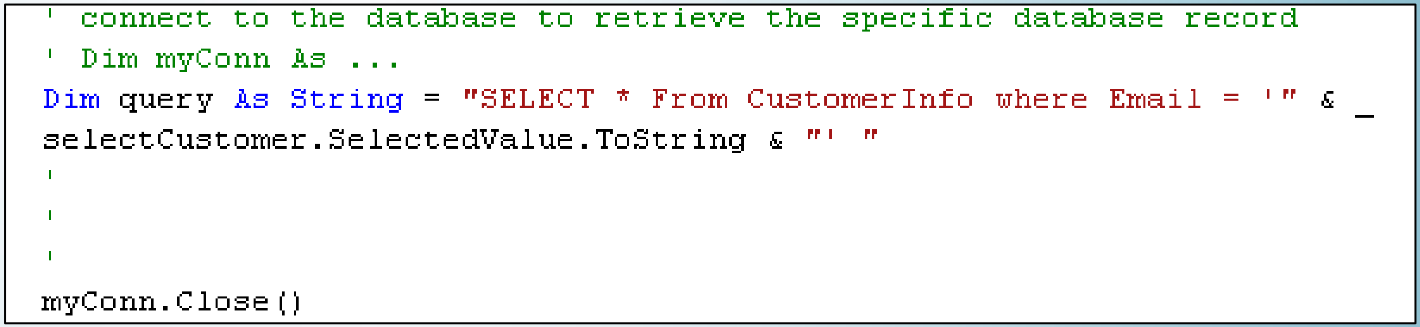
- To select a particular record, one can use the WHERE clause in SQL statements

- Example:

```
Select * From Customers where Email=‘jane_doe@doe_com’
```

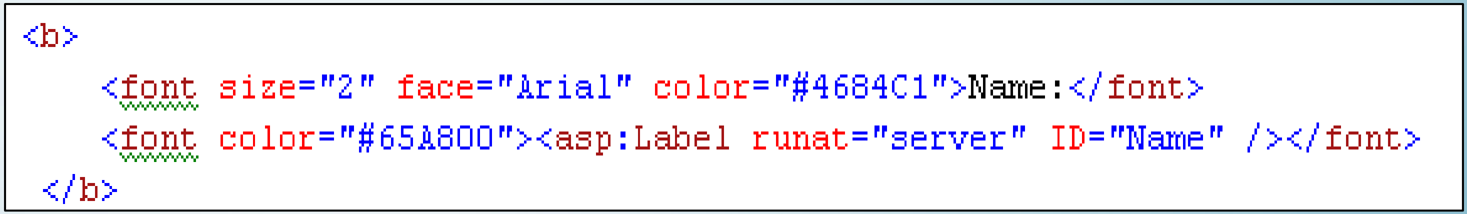
- For this example, we will select the value of the drop down menu item

IsPostBack Example (continued)



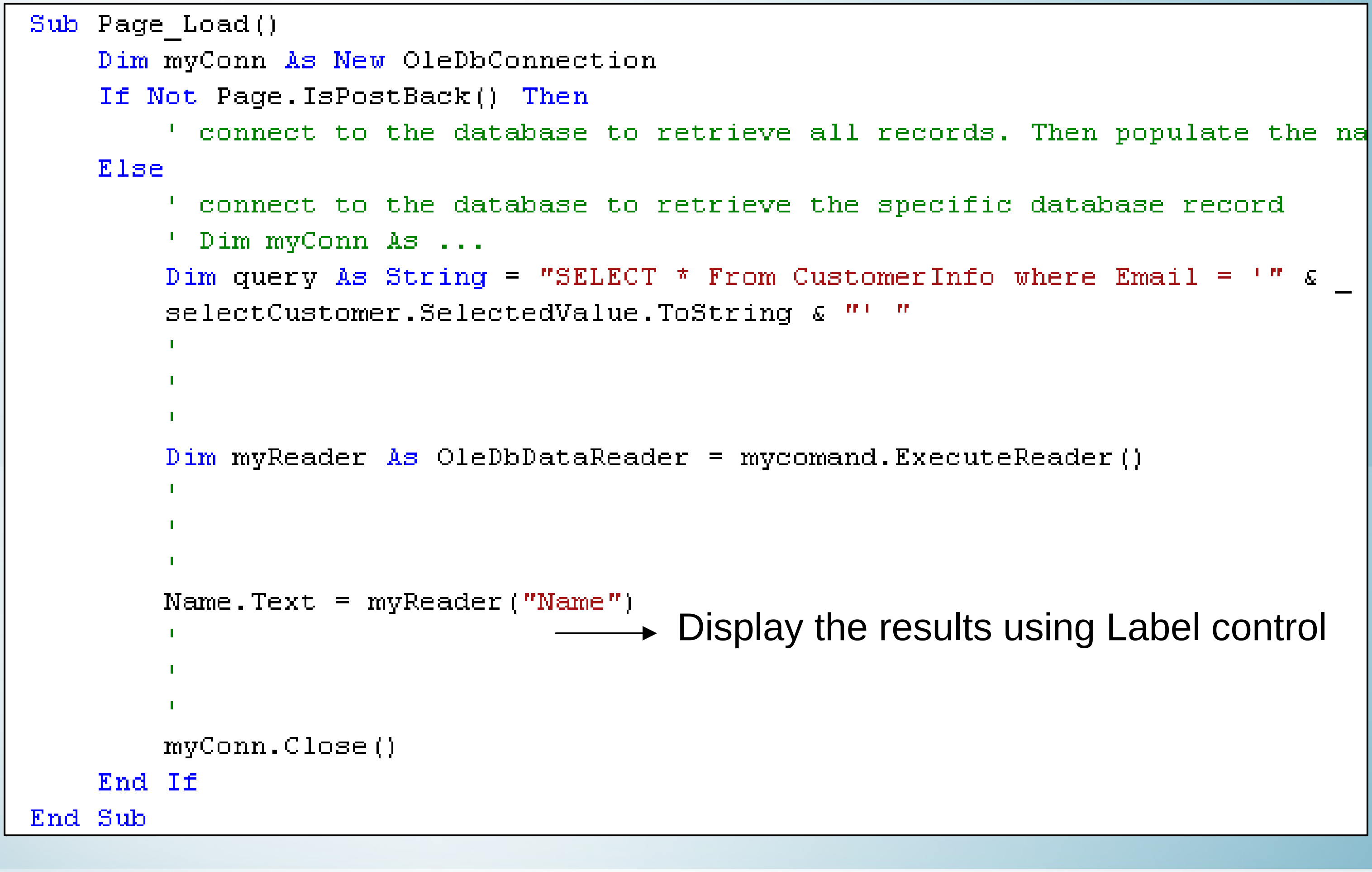
```
' connect to the database to retrieve the specific database record
' Dim myConn As ...
Dim query As String = "SELECT * From CustomerInfo where Email = '" & _
selectCustomer.SelectedValue.ToString & "' "
'
'
'
myConn.Close()
```

- Now we would like to display the results
  - For simplicity, we will use the Label ASP.NET control to display the results



```
<b>
    <font size="2" face="Arial" color="#4684C1">Name:</font>
    <font color="#65A800"><asp:Label runat="server" ID="Name" /></font>
 </b>
```

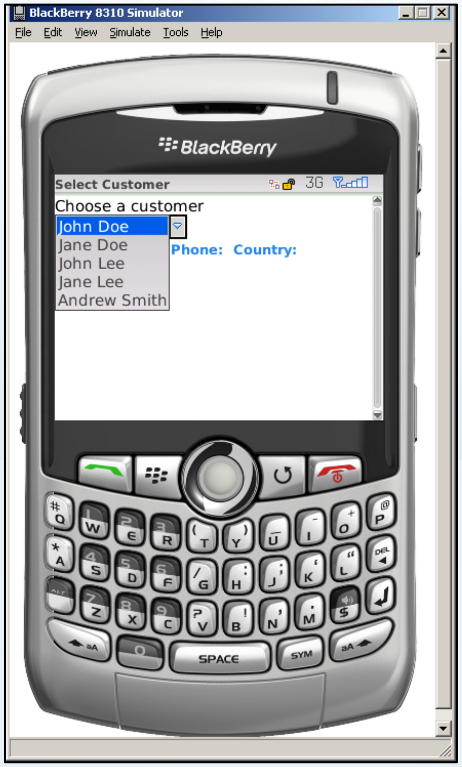
IsPostBack Example (continued)



```
Sub Page_Load()
    Dim myConn As New OleDbConnection
    If Not Page.IsPostBack() Then
        ' connect to the database to retrieve all records. Then populate the na
    Else
        ' connect to the database to retrieve the specific database record
        ' Dim myConn As ...
        Dim query As String = "SELECT * From CustomerInfo where Email = '" & _
        selectCustomer.SelectedValue.ToString & "' "
        '
        '
        '
        Dim myReader As OleDbDataReader = mycomand.ExecuteReader()
        '
        '
        '
        Name.Text = myReader("Name")
        '
        '
        '
        myConn.Close()
    End If
End Sub
```

Display the results using Label control

IsPostBack Example (continued)

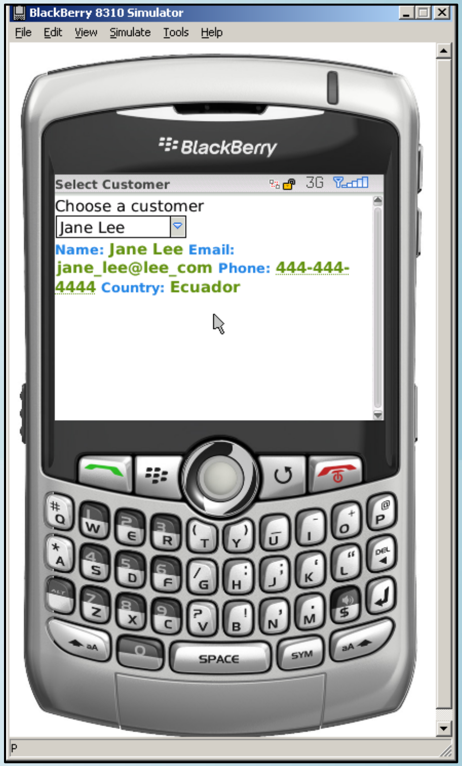


Select Customer

Choose a customer

John Doe
Jane Doe
John Lee
Jane Lee
Andrew Smith

Phone: Country:



Select Customer

Choose a customer

Jane Lee

Name: Jane Lee Email: jane_lee@lee_com Phone: 444-444-4444 Country: Ecuador

Exception Handling

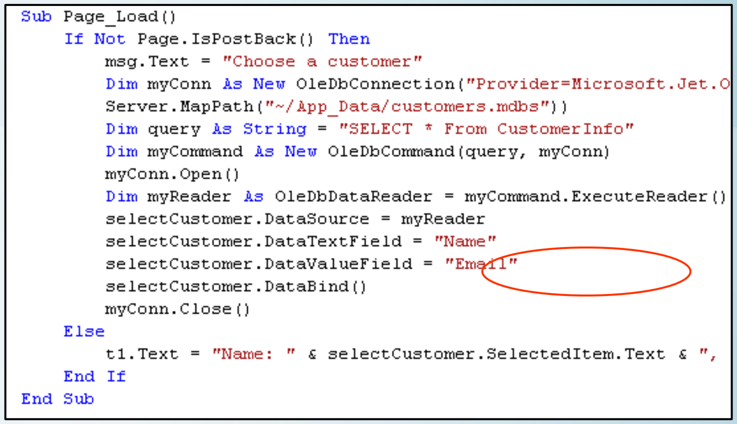
- Exceptions are anomalies that happen when running a program

- Exception handling is very useful when detecting and handling errors at runtime

- In cases a program fails to handle exceptions, the default behaviour of any programming environment is to terminate the program

Exception Handling (continued)

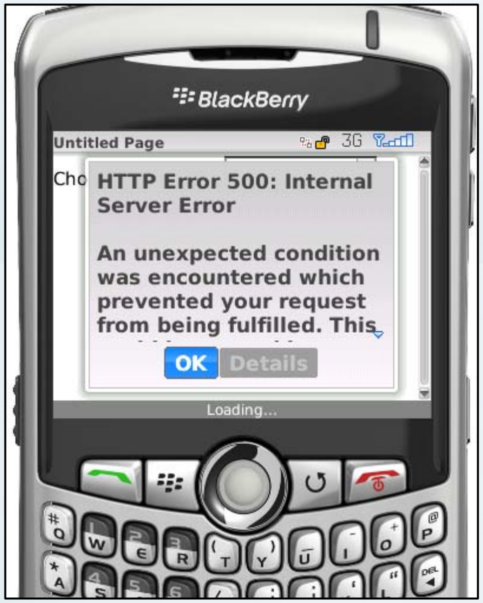
- The try, catch and finally are the three keyword that handle exceptions in ASP.NET
- For example, assume that from the previous example, the database filename was misspelled



```
Sub Page_Load()
    If Not Page.IsPostBack() Then
        msg.Text = "Choose a customer"
        Dim myConn As New OleDbConnection("Provider=Microsoft.Jet.O
        Server.MapPath("~/App_Data/customers.mdbs"))
        Dim query As String = "SELECT * From CustomerInfo"
        Dim myCommand As New OleDbCommand(query, myConn)
        myConn.Open()
        Dim myReader As OleDbDataReader = myCommand.ExecuteReader()
        selectCustomer.DataSource = myReader
        selectCustomer.DataTextField = "Name"
        selectCustomer.DataValueField = "Email"
        selectCustomer.DataBind()
        myConn.Close()
    Else
        t1.Text = "Name: " & selectCustomer.SelectedItem.Text & ",
    End If
End Sub
```

Exception Handling (continued)

- The following error will occur in BlackBerry device



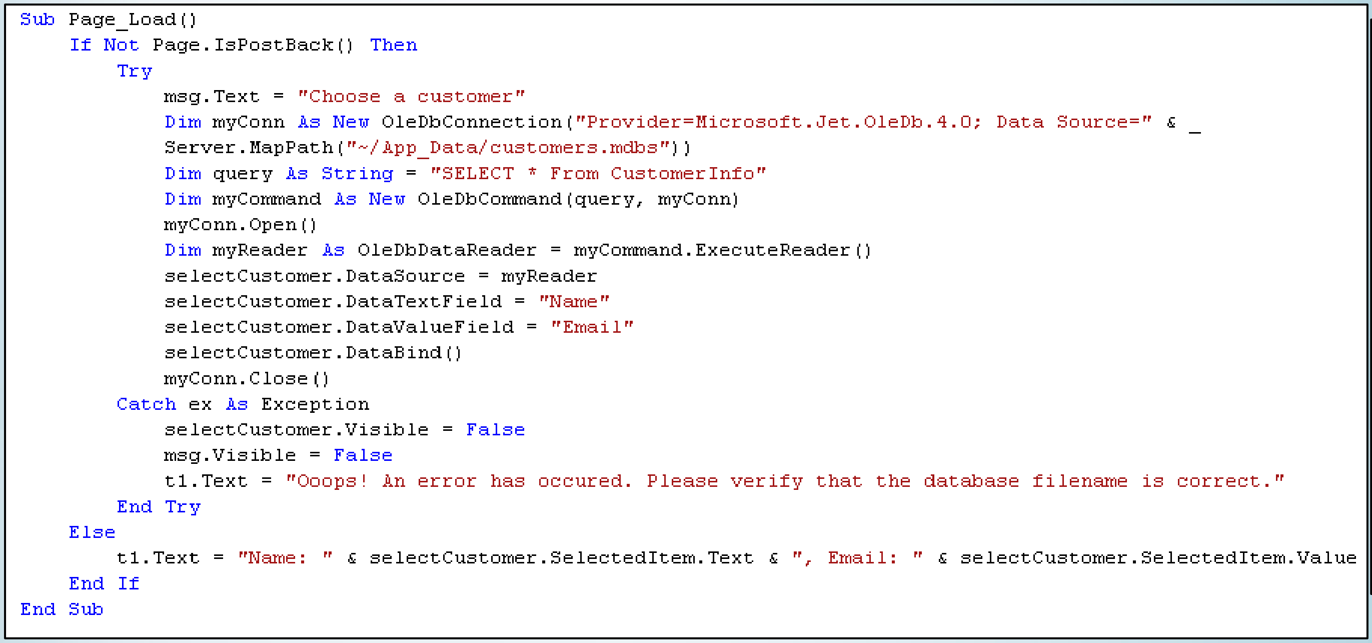
HTTP Error 500: Internal Server Error

An unexpected condition was encountered which prevented your request from being fulfilled. This

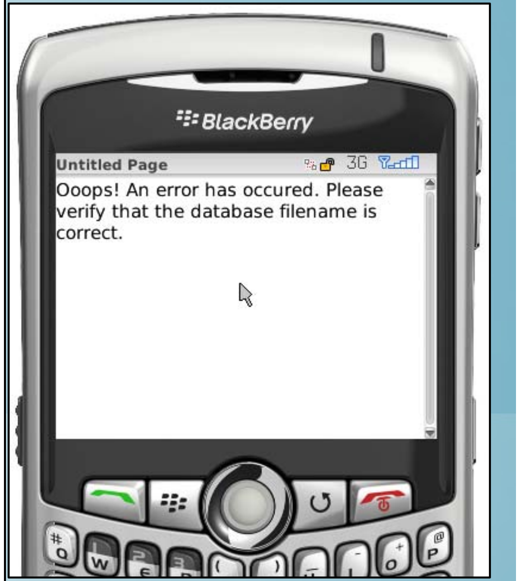
To handle this exception, the try/catch clause is used

```
Try
        …Connect to Database…
Catch
        … Catch any exceptions…
End Try
```

Exception Handling Database Example



```
Sub Page_Load()
    If Not Page.IsPostBack() Then
        Try
            msg.Text = "Choose a customer"
            Dim myConn As New OleDbConnection("Provider=Microsoft.Jet.OleDb.4.0; Data Source=" & _
            Server.MapPath("~/App_Data/customers.mdbs"))
            Dim query As String = "SELECT * From CustomerInfo"
            Dim myCommand As New OleDbCommand(query, myConn)
            myConn.Open()
            Dim myReader As OleDbDataReader = myCommand.ExecuteReader()
            selectCustomer.DataSource = myReader
            selectCustomer.DataTextField = "Name"
            selectCustomer.DataValueField = "Email"
            selectCustomer.DataBind()
            myConn.Close()
        Catch ex As Exception
            selectCustomer.Visible = False
            msg.Visible = False
            t1.Text = "Ooops! An error has occured. Please verify that the database filename is correct."
        End Try
    Else
        t1.Text = "Name: " & selectCustomer.SelectedItem.Text & ", Email: " & selectCustomer.SelectedItem.Value
    End If
End Sub
```



Ooops! An error has occured. Please verify that the database filename is correct.

Rendering the Correct Markup

- Due to the fact that different devices support different types of markup languages, it is important that a mobile Web application is capable of rendering the appropriate markup
- This requires the development of different versions of the mobile Web application to support a larger number of mobile devices

Rendering the Correct Markup (.NET Mobile Controls)

- .NET Mobile controls extend the .NET Framework and Visual Studio to build Mobile Web applications that can deliver markup to larger set of mobile devices
- One **does not** have to write different code for each device
- .NET Mobile Controls renders the appropriate markup (i.e. HTML 3.2, WML 1.1, cHTML, or XHTML)

Rendering the Correct Markup (.NET Mobile Controls)

- When retrieving records from databases, it is important to consider the orientation of the mobile devices
  - – i.e. do not output database information containing five columns in one screen (would not fit properly)
  - – .NET mobile controls can solve the problem (it takes care of the display issues based on what the device can support)

Rendering the Correct Markup (.NET Mobile Controls)

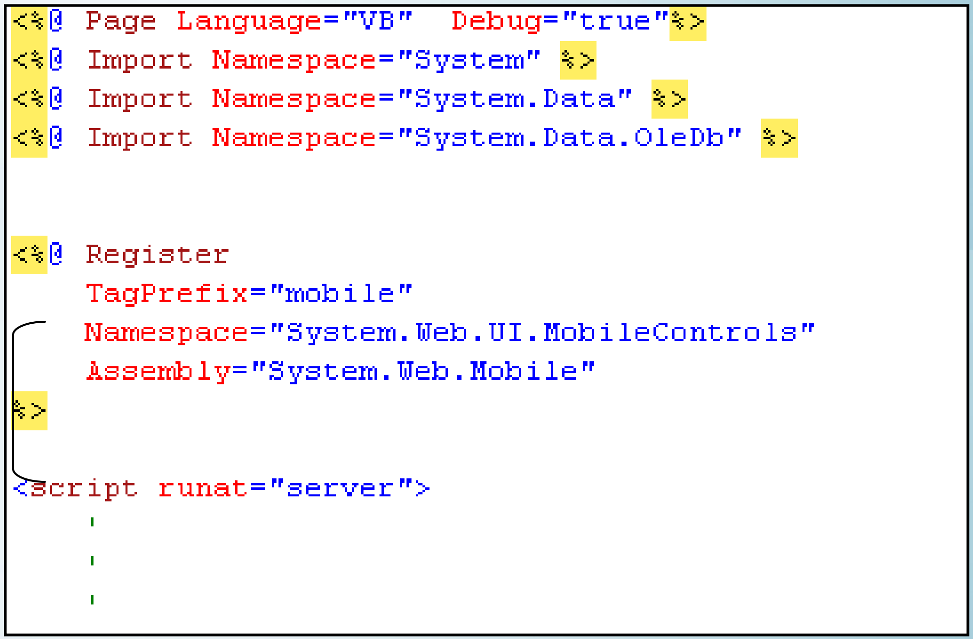
- For example, if you are developing a Web application, one can use the data binding control in ASP.NET such as GridView

- However, when developing mobile Web applications, GridView may not be appropriate

- Instead, one can use .NET Mobile Controls

.NET Mobile Controls

- One needs to register the page to indicate that it needs to support Mobile Controls



```
<%@ Page Language="VB"  Debug="true"%>
<%@ Import Namespace="System" %>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.OleDb" %>


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

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

.NET Mobile Controls (continued)

- The .NET Mobile is an extension in the .NET Framework
  - Formerly called Microsoft Mobile Internet Toolkit

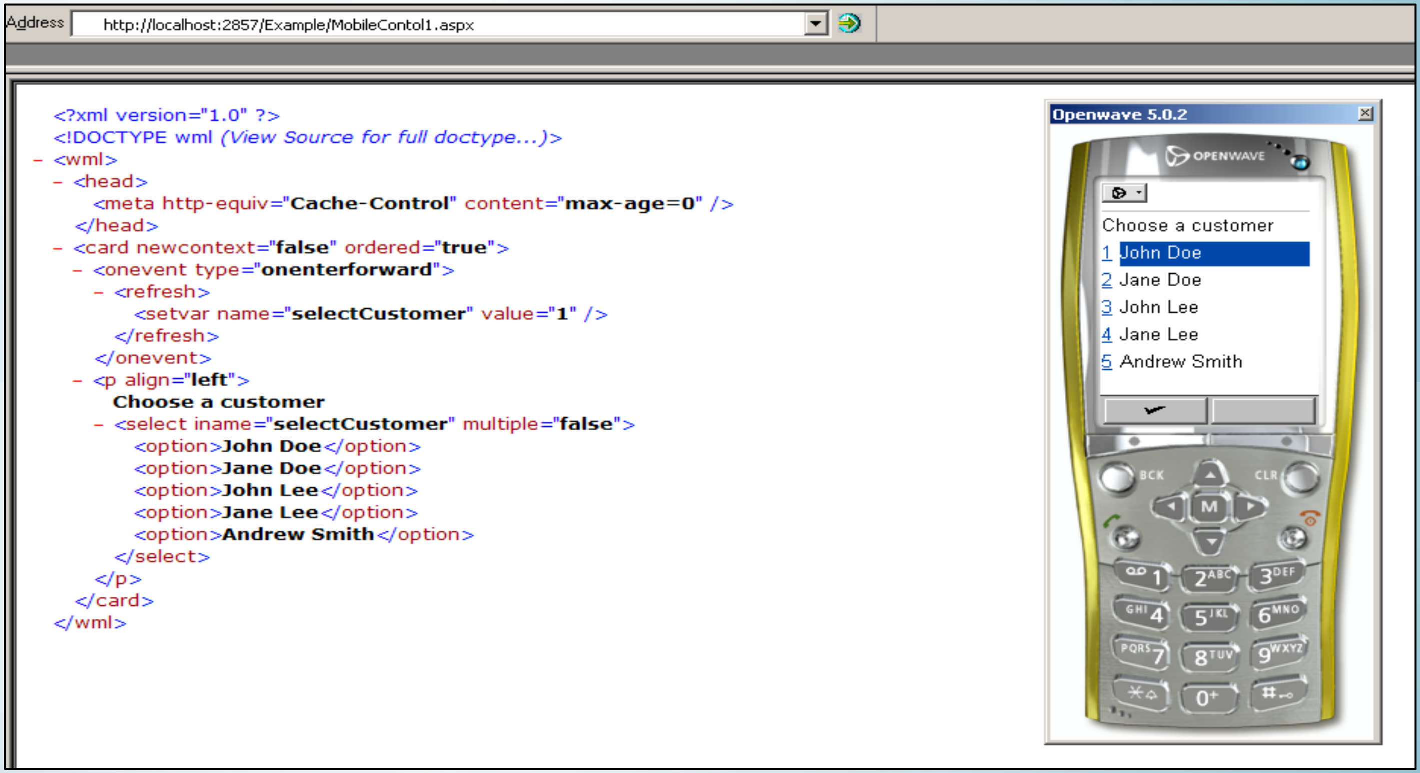
- The Mobile class
  System.Web.UI.MobileControls
  is the base class for all Mobile Web Forms

.NET Mobile Controls Example

- This example performs the same as the DropDownList example
  - The main difference is the use of Mobile Controls
  - Depending on the type of device and what markup it supports, mobile controls are able to output the response in a markup language that the device can understand

Running the .NET Mobile Controls Example using Openwave SDK 5.1

Content is rendered in WML 1.0 since this device only understands this markup language



Address http://localhost:2857/Example/MobileContol1.aspx

```
  <?xml version="1.0" ?>
  <!DOCTYPE wml (View Source for full doctype...)>
- <wml>
  - <head>
      <meta http-equiv="Cache-Control" content="max-age=0" />
    </head>
  - <card newcontext="false" ordered="true">
    - <onevent type="onenterforward">
      - <refresh>
          <setvar name="selectCustomer" value="1" />
        </refresh>
      </onevent>
    - <p align="left">
        Choose a customer
      - <select iname="selectCustomer" multiple="false">
          <option>John Doe</option>
          <option>Jane Doe</option>
          <option>John Lee</option>
          <option>Jane Lee</option>
          <option>Andrew Smith</option>
        </select>
      </p>
    </card>
  </wml>
```

Openwave 5.0.2

Choose a customer

1 John Doe

2 Jane Doe

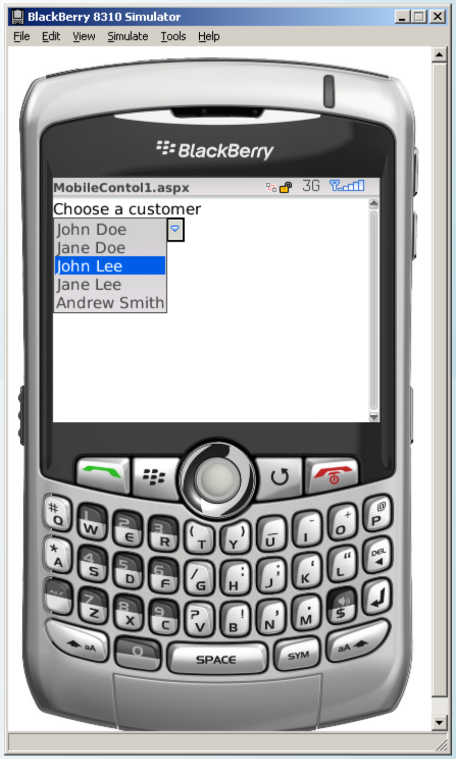
3 John Lee

4 Jane Lee

5 Andrew Smith

Running the .NET Mobile Controls Example using BlackBerry 8310

- Output is in HTML



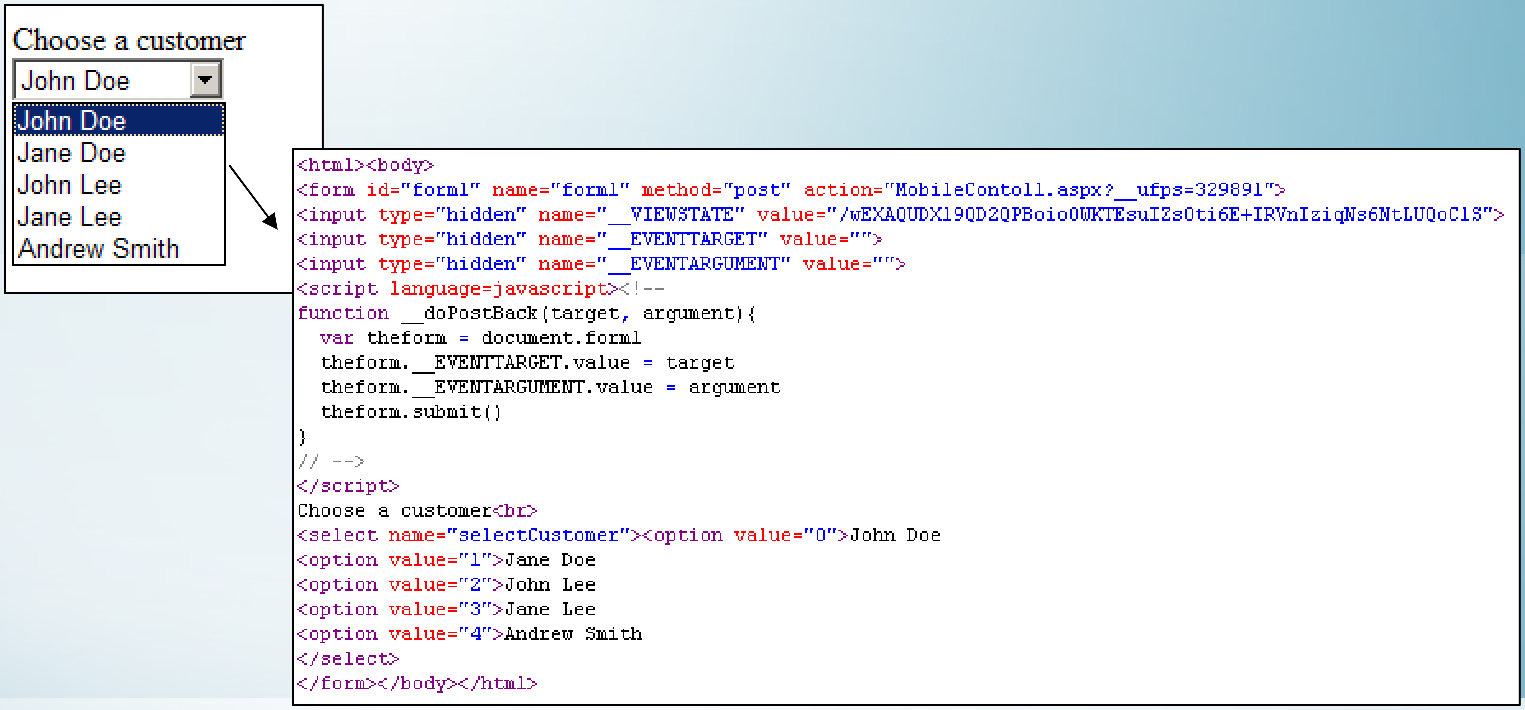
MobileContol1.aspx

Choose a customer

John Doe
Jane Doe
John Lee
Jane Lee
Andrew Smith

Running the .NET Mobile Controls Example using Internet Explorer

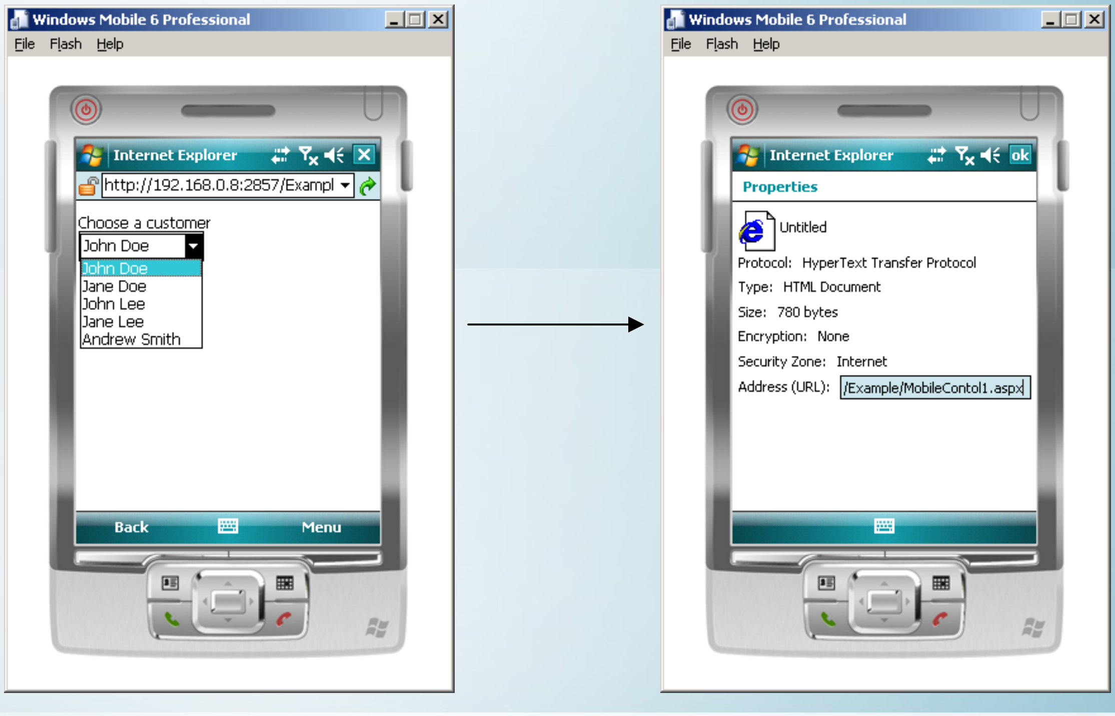
- Output in HTML



```
<html><body>
<form id="form1" name="form1" method="post" action="MobileContol1.aspx?__ufps=329891">
<input type="hidden" name="__VIEWSTATE" value="/wEXAQUDX19QD2QPBoioOWKTEsuIZsOti6E+IRVnIziqNs6NtLUQoClS">
<input type="hidden" name="__EVENTTARGET" value="">
<input type="hidden" name="__EVENTARGUMENT" value="">
<script language=javascript><!--
function __doPostBack(target, argument){
  var theform = document.form1
  theform.__EVENTTARGET.value = target
  theform.__EVENTARGUMENT.value = argument
  theform.submit()
}
// -->
</script>
Choose a customer<br>
<select name="selectCustomer"><option value="0">John Doe
<option value="1">Jane Doe
<option value="2">John Lee
<option value="3">Jane Lee
<option value="4">Andrew Smith
</select>
</form></body></html>
```

Running the .NET Mobile Controls Example using Microsoft Mobile 6

- Output in HTML



http://192.168.0.8:2857/Exampl

Choose a customer

John Doe

John Doe
Jane Doe
John Lee
Jane Lee
Andrew Smith

Properties

Untitled

Protocol: HyperText Transfer Protocol

Type: HTML Document

Size: 780 bytes

Encryption: None

Security Zone: Internet

Address (URL): /Example/MobileContol1.aspx

ObjectList Mobile Control

- When dealing with databases, displaying records in a professional format is essential (i.e. tabular format)
- The ObjectList mobile control displays the results in tabular format and customizes that output based on the device capabilities

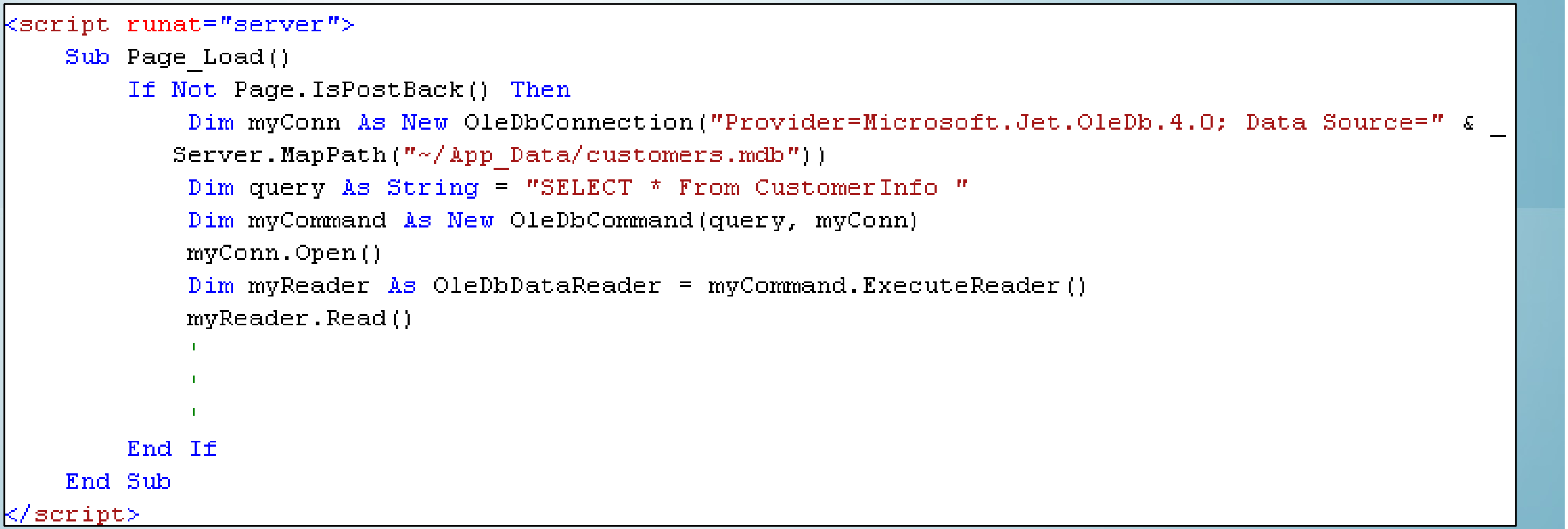
ObjectList Mobile Control Example

- In this example, we would like to display the content of the CustomerInfo table when the ASP.NET page loads.

- We also would like to limit the display to particular fields, then the user can click on the customer’s to view more details (i.e. telephone number, country, etc.)

ObjectList Mobile Control Example (continued)

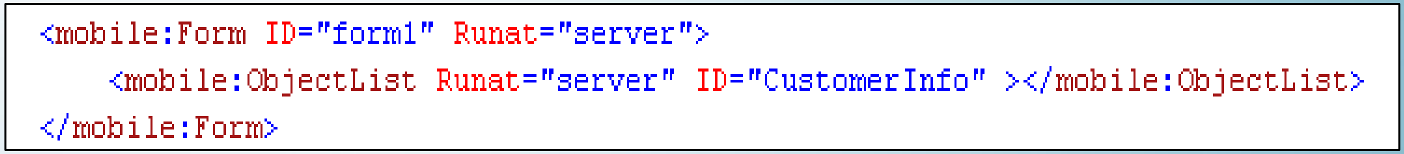
- We begin by reading all the content in the CustomerInfo table and store the records into the data reader object



```
<script runat="server">
    Sub Page_Load()
        If Not Page.IsPostBack() Then
            Dim myConn As New OleDbConnection("Provider=Microsoft.Jet.OleDb.4.0; Data Source=" & _
           Server.MapPath("~/App_Data/customers.mdb"))
            Dim query As String = "SELECT * From CustomerInfo "
            Dim myCommand As New OleDbCommand(query, myConn)
            myConn.Open()
            Dim myReader As OleDbDataReader = myCommand.ExecuteReader()
            myReader.Read()
            '
            '
            '
        End If
    End Sub
</script>
```

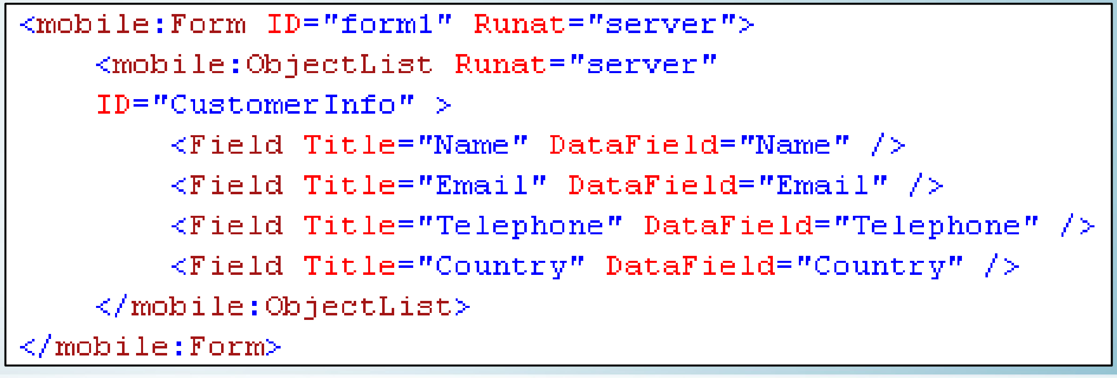
ObjectList Mobile Control Example (continued)

- We then create the necessary mobile control using <mobile:ObjectList> directive



```
<mobile:Form ID="form1" Runat="server">
    <mobile:ObjectList Runat="server" ID="CustomerInfo" ></mobile:ObjectList>
</mobile:Form>
```

- Then we add the necessary database fields that we wish to display



```
<mobile:Form ID="form1" Runat="server">
    <mobile:ObjectList Runat="server"
    ID="CustomerInfo" >
        <Field Title="Name" DataField="Name" />
        <Field Title="Email" DataField="Email" />
        <Field Title="Telephone" DataField="Telephone" />
        <Field Title="Country" DataField="Country" />
    </mobile:ObjectList>
</mobile:Form>
```

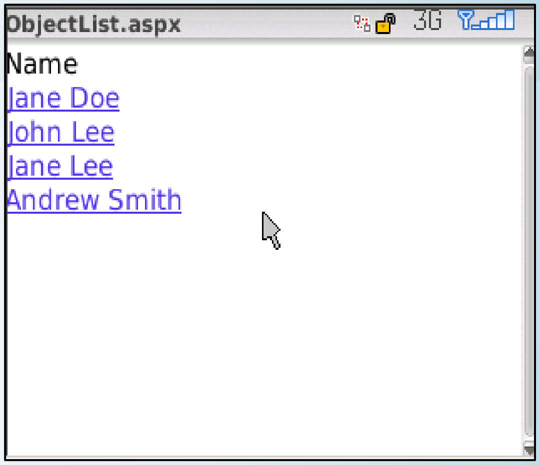
ObjectList Mobile Control Example (continued)

- Since we would to specify the fields to be shown on the ObjectList, we need to add the property: AutoGenerateFields="False"

- We can also control which fields can be visible when the list view is shown in tabular format by adding the property: TableFields="Name;Email"

ObjectList Mobile Control Output in BlackBerry 8310

- Since the screen size is limited, the ObjectList mobile control would customize the display to fit on the screen
  - It will only display one field; in this example it would select the first listed field, or Name



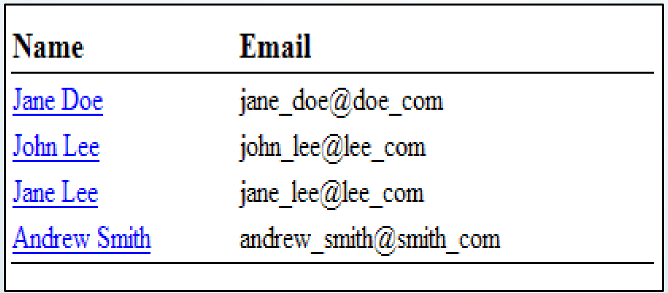
Name
Jane Doe
John Lee
Jane Lee
Andrew Smith

ObjectList Mobile Control Output in BlackBerry 8310

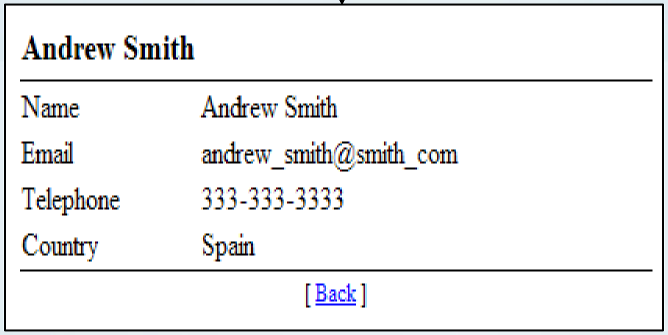
- Although the TableFields="Name;Email" was set to display the two columns: Name and Email, using the BlackBerry 8310, the control only rendered one field
- Browsing the same file using Internet Explorer, we would expect the ObjectList to display both columns since the display size is not an issue as in the BlackBerry device

ObjectList Example: Output in IE versus BlackBerry

Output in HTML



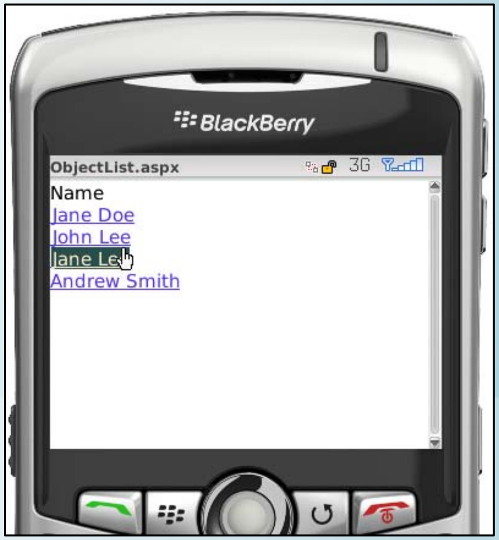
Name	Email
Jane Doe	jane_doe@doe_com
John Lee	john_lee@lee_com
Jane Lee	jane_lee@lee_com
Andrew Smith	andrew_smith@smith_com



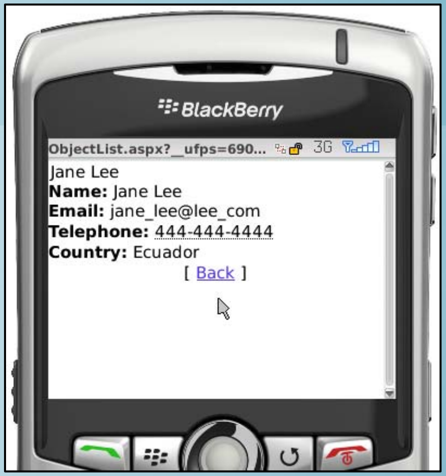
Andrew Smith

Name	Andrew Smith
Email	andrew_smith@smith_com
Telephone	333-333-3333
Country	Spain

[Back]



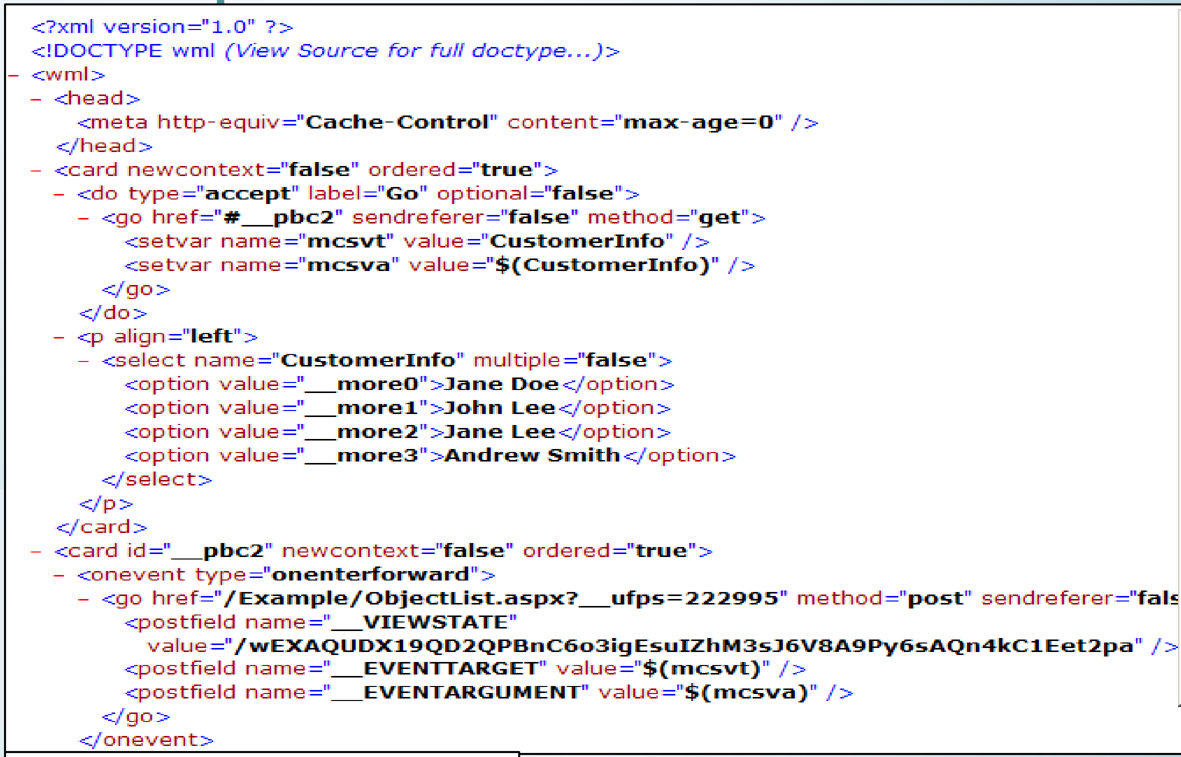
Name
Jane Doe
John Lee
Jane Lee
Andrew Smith



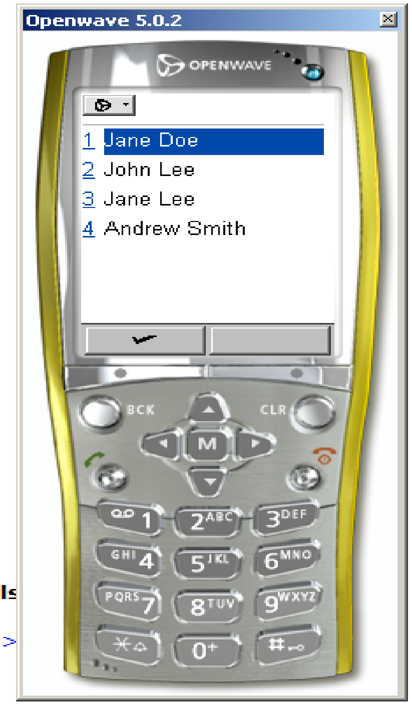
Jane Lee
Name: Jane Lee
Email: jane_lee@lee_com
Telephone: 444-444-4444
Country: Ecuador
[Back]

ObjectList Example: Output in OpenWave 5.1 (1/2)

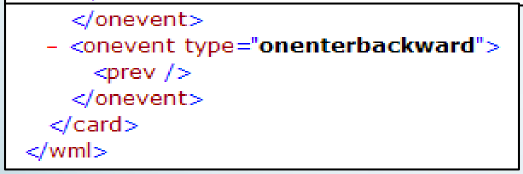
Output in WML



```
  <?xml version="1.0" ?>
  <!DOCTYPE wml (View Source for full doctype...)>
- <wml>
  - <head>
      <meta http-equiv="Cache-Control" content="max-age=0" />
    </head>
  - <card newcontext="false" ordered="true">
    - <do type="accept" label="Go" optional="false">
      - <go href="#__pbc2" sendreferer="false" method="get">
          <setvar name="mcsvt" value="CustomerInfo" />
          <setvar name="mcsva" value="$(CustomerInfo)" />
        </go>
      </do>
    - <p align="left">
      - <select name="CustomerInfo" multiple="false">
          <option value="__more0">Jane Doe</option>
          <option value="__more1">John Lee</option>
          <option value="__more2">Jane Lee</option>
          <option value="__more3">Andrew Smith</option>
        </select>
      </p>
    </card>
  - <card id="__pbc2" newcontext="false" ordered="true">
    - <onevent type="onenterforward">
      - <go href="/Example/ObjectList.aspx?__ufps=222995" method="post" sendreferer="fals
          <postfield name="__VIEWSTATE"
            value="/wEXAQUDX19QD2QPBnC6o3igEsuIZhM3sJ6V8A9Py6sAQn4kC1Eet2pa" />
          <postfield name="__EVENTTARGET" value="$(mcsvt)" />
          <postfield name="__EVENTARGUMENT" value="$(mcsva)" />
        </go>
      </onevent>
```



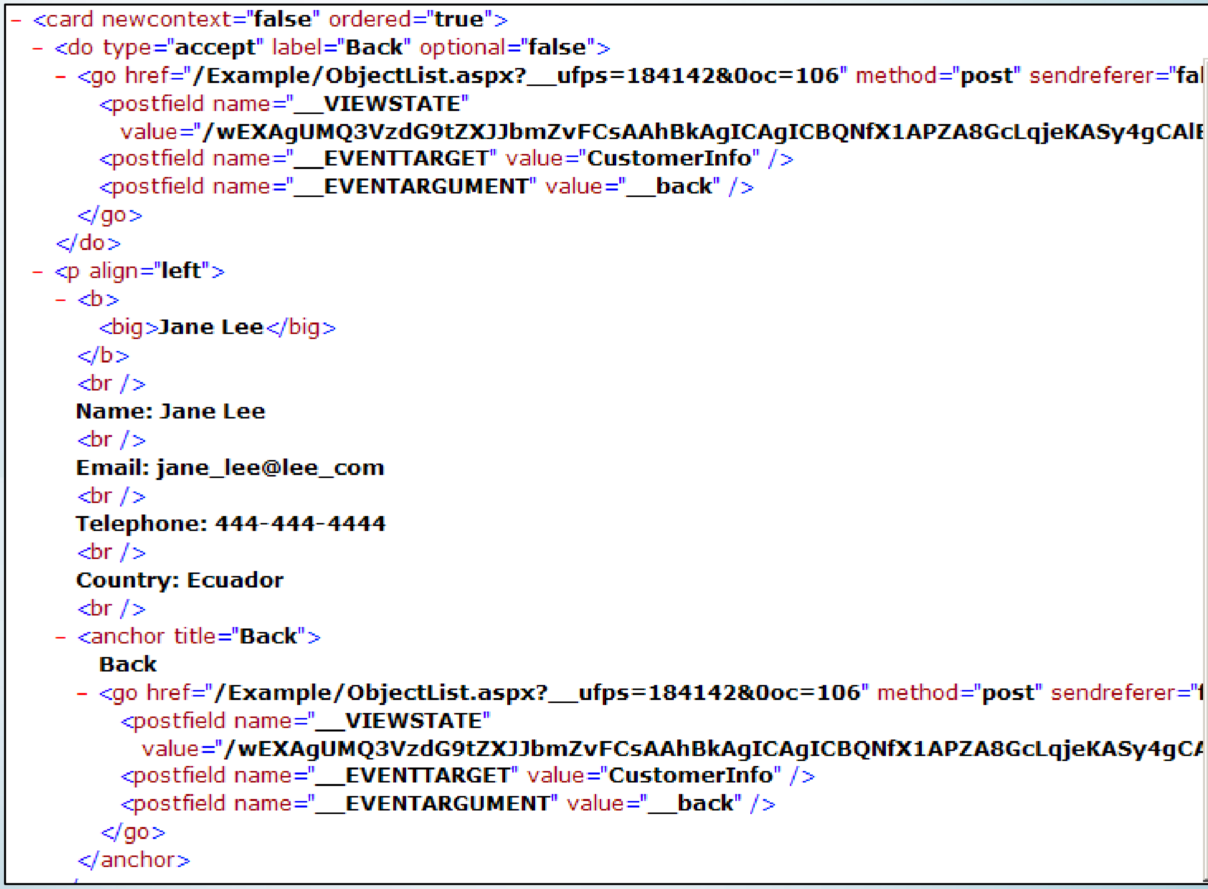
1 Jane Doe
2 John Lee
3 Jane Lee
4 Andrew Smith



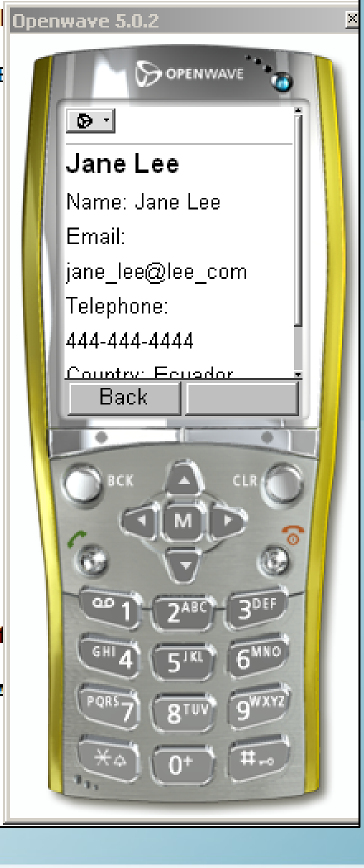
```
      </onevent>
    - <onevent type="onenterbackward">
        <prev />
      </onevent>
    </card>
  </wml>
```

ObjectList Example: Output in OpenWave 5.1 (2/2)

Selecting Jane Doe



```
- <card newcontext="false" ordered="true">
  - <do type="accept" label="Back" optional="false">
    - <go href="/Example/ObjectList.aspx?__ufps=184142&0oc=106" method="post" sendreferer="fal
        <postfield name="__VIEWSTATE"
          value="/wEXAgUMQ3VzdG9tZXJJbmZvFCsAAhBkAgICAgICBQNfX1APZA8GcLqjeKASy4gCAlE
        <postfield name="__EVENTTARGET" value="CustomerInfo" />
        <postfield name="__EVENTARGUMENT" value="__back" />
      </go>
    </do>
  - <p align="left">
    - <b>
        <big>Jane Lee</big>
      </b>
      <br />
      Name: Jane Lee
      <br />
      Email: jane_lee@lee_com
      <br />
      Telephone: 444-444-4444
      <br />
      Country: Ecuador
      <br />
    - <anchor title="Back">
        Back
      - <go href="/Example/ObjectList.aspx?__ufps=184142&0oc=106" method="post" sendreferer="f
          <postfield name="__VIEWSTATE"
            value="/wEXAgUMQ3VzdG9tZXJJbmZvFCsAAhBkAgICAgICBQNfX1APZA8GcLqjeKASy4gCA
          <postfield name="__EVENTTARGET" value="CustomerInfo" />
          <postfield name="__EVENTARGUMENT" value="__back" />
        </go>
      </anchor>
```



Openwave 5.0.2

Jane Lee

Name: Jane Lee

Email:

jane_lee@lee_com

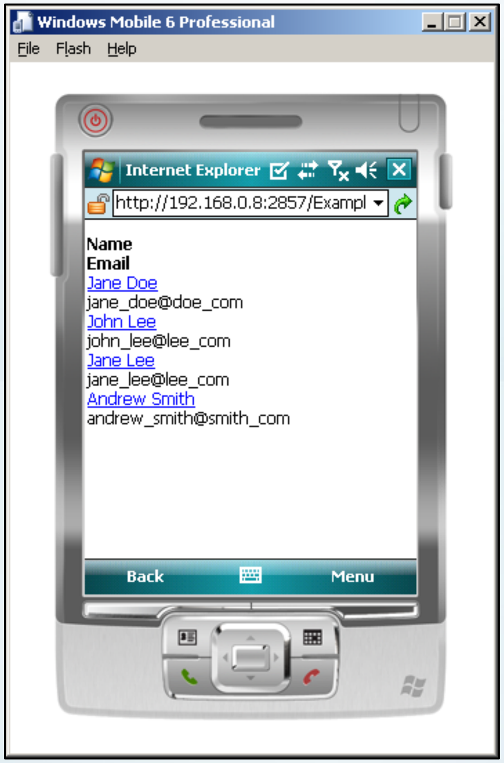
Telephone:

444-444-4444

Country: Ecuador

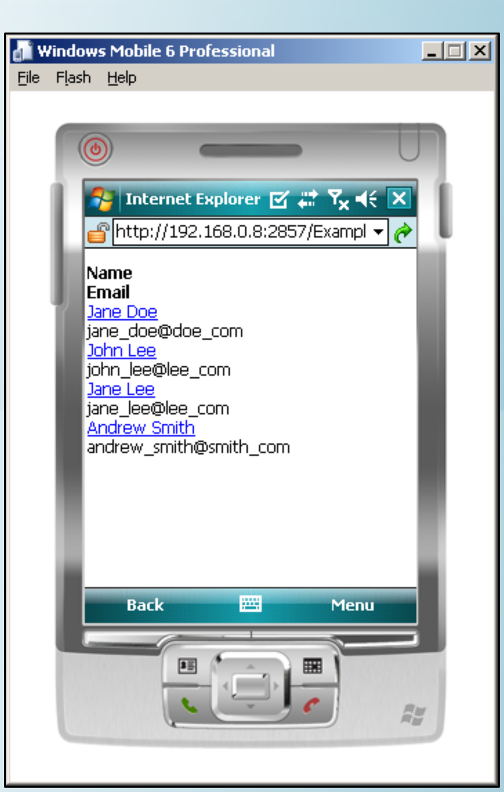
Back

ObjectList Example: Output in Windows Mobile 6.0



```
http://192.168.0.8:2857/Exampl
```

```
Name
Email
Jane Doe
jane_doe@doe_com
John Lee
john_lee@lee_com
Jane Lee
jane_lee@lee_com
Andrew Smith
andrew_smith@smith_com
```



http://192.168.0.8:2857/Exampl

Name
Email
Jane Doe
jane_doe@doe_com
John Lee
john_lee@lee_com
Jane Lee
jane_lee@lee_com
Andrew Smith
andrew_smith@smith_com

ObjectList Example Summary

- As shown in the examples, executing the same ASP.NET code does not necessarily mean that it will display seminally on all mobile devices
- One unique advantage of the ObjectList mobile control is that it renders the content to be displayed on each device according to its orientation and capabilities

Session Handling

- Since HTTP is stateless, keeping track of users becomes very difficult
- Session objects overcomes this limitation
- Time spent at a particular Web site is called a session
  - Once a user exits, the session is abandoned

Session Variables

- You can store information into each session

- Declaring a session variable is valid until the valid is abandoned

- Syntax in ASP.NET: Session(“var_name”)

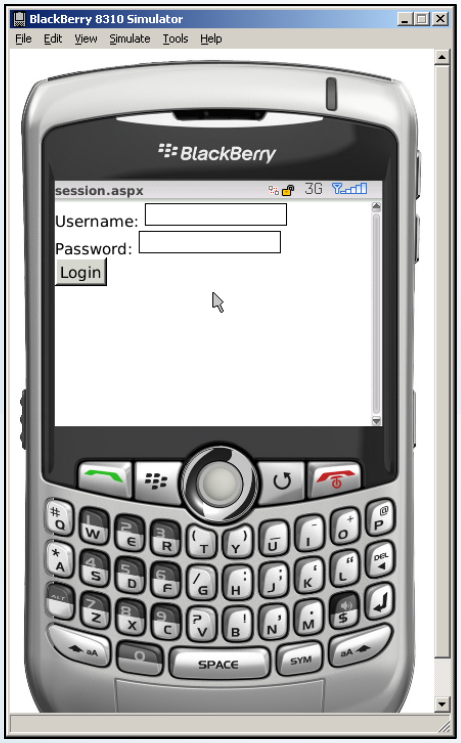
- You can control the duration of the session using the session timeout property

Session Example Using BlackBerry 8310

- In this example, a user logs In through session.aspx which will check a username and password
- If the login is successful, it initiates a session variable called “Username” and “Logged”
- The session timeout is set to 1 minute

Session Example Using BlackBerry 8310 (continued)

Login panel



session.aspx

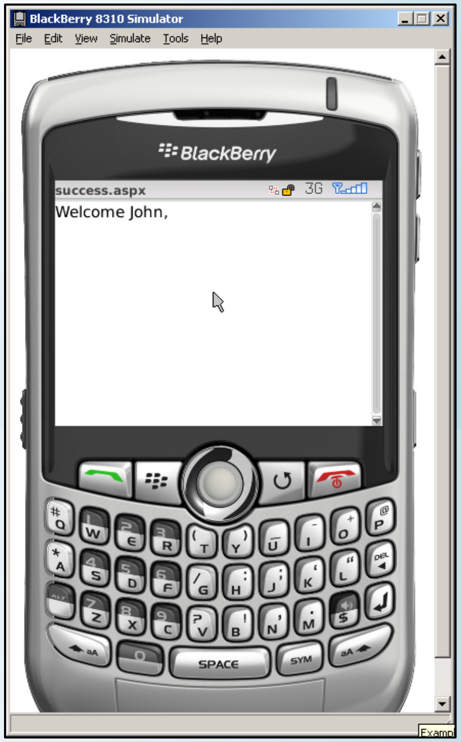
Username:

Password:

Login

Login.aspx

After successful login

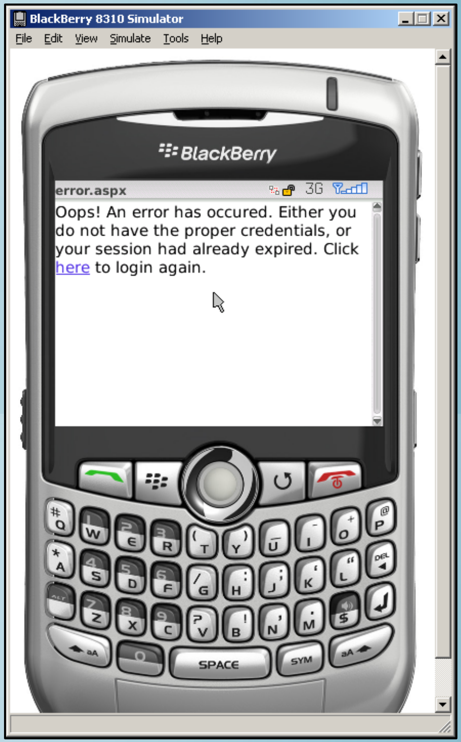


success.aspx

Welcome John,

Success.aspx

After session expires



error.aspx

Oops! An error has occured. Either you do not have the proper credentials, or your session had already expired. Click here to login again.

Error.aspx

AJAX and BlackBerry Devices

- Stands for Asynchronous JavaScript And XML (AJAX)
- Used to create rich, interactive Web sites
- AJAX can be used to create interactive mobile Web applications

AJAX and BlackBerry Devices

- Not all BlackBerry devices support the AJAX technology
  - Upgrades are available that can provide AJAX support for older models
- Models beyond the BlackBerry Bold 9000 support AJAX
- Simulators in the Visual Studio Plug-in do not support AJAX
  - You must externally run simulators for the BlackBerry 9000 model or beyond

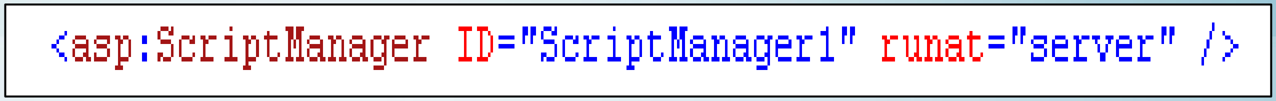
AJAX and ASP.NET

- ASP.NET supports the AJAX technology
- In AJAX, data is retrieved using the XMLHttpRequest object or remote scripting in some browsers
- AJAX-enabled Web applications can retrieve data from the server asynchronously in the background without interfering with the existing UI or display

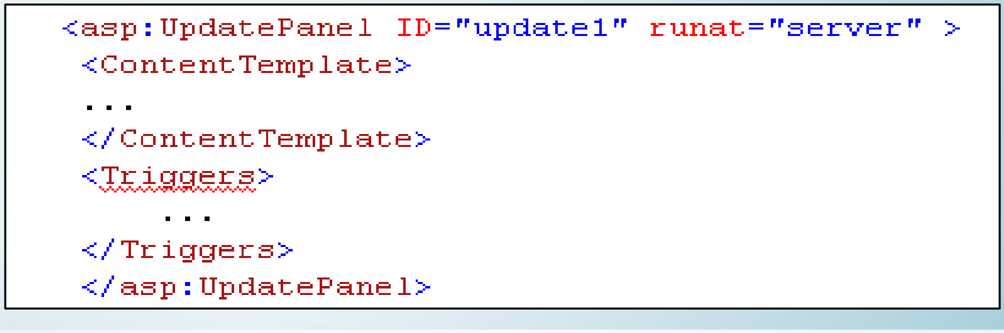
AJAX in ASP.NET Example

- The <asp:scriptManager > is necessary when building an AJAX-enabled Web page

- The UpdatePanel is used to control the section of a Web page to which the AJAX technology is applied



```
<asp:ScriptManager ID="ScriptManager1" runat="server" />
```



```
<asp:UpdatePanel ID="update1" runat="server" >
 <ContentTemplate>
 ...
 </ContentTemplate>
 <Triggers>
     ...
 </Triggers>
 </asp:UpdatePanel>
```

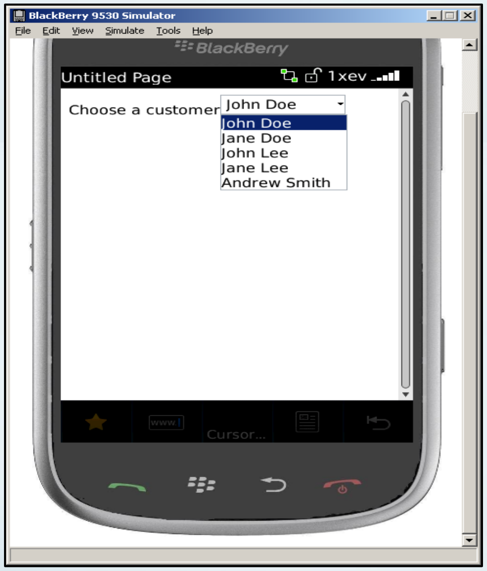
AJAX in ASP.NET Example (continued)

- The <Triggers> command instructs the ASP.NET to determine which server control “triggers” the event that causes the UpdatePanel to be updated

- The <ContentTemplate> contains the server controls that need to be updated

AJAX in ASP.NET Example (continued)

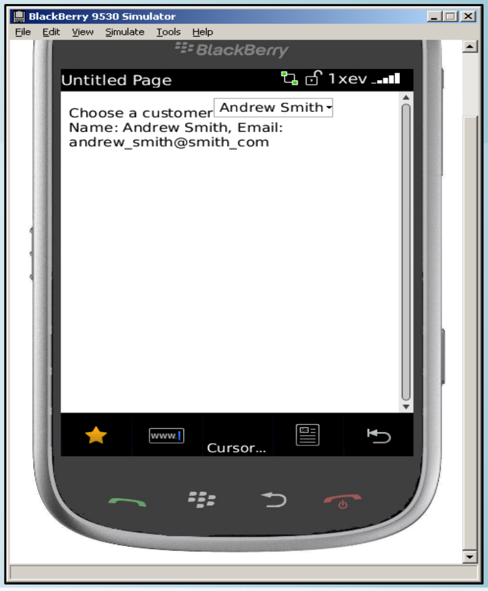
This example populates the customer names into a drop down menu. Once an item is selected, the Name and Email are written to the screen asynchronously in the background



Untitled Page

Choose a customer John Doe

John Doe
Jane Doe
John Lee
Jane Lee
Andrew Smith



Untitled Page

Choose a customer Andrew Smith

Name: Andrew Smith, Email: andrew_smith@smith_com

AJAX and Mobile Devices

- When building a mobile Web application, it is important to take into consideration the bandwidth limitations
- Because AJAX uses XMLHttpRequest, it may not be good idea to create mobile Web applications that extensively use AJAX
  - Limited bandwidth
  - Limited memory