25

ASP.NET 2.0 and ASP.NET Ajax

Outline

25.1 Introduction

25.2 Creating and Running a Simple Web Form Example

- 25.2.1 Examining an ASPX File
- 25.2.2 Examining a Code-Behind File
- 25.2.3 Relationship Between an ASPX File and a Code-Behind File
- 25.2.4 How the Code in an ASP.NET Web Page Executes
- 25.2.5 Examining the XHTML Generated by an ASP.NET Application
- 25.2.6 Building an ASP.NET Web Application

25.3 Web Controls

- 25.3.1 Text and Graphics Controls
- 25.3.2 `AdRotator` Control
- 25.3.3 Validation Controls

Outline

Outline

25.1 Introduction

- ASP.NET 2.0 and Web Forms and Controls
  - Web application development with Microsoft’s ASP.NET 2.0 technology
  - Web Form files have the filename extension .aspx and contain the Web page’s GUI
  - Every ASPX file created in Visual Studio has a corresponding class written in a .NET language, such as Visual Basic
    - Contains event handlers, initialization code, utility methods and other supporting code
    - Called the code-behind file
    - Provides the ASPX file’s programmatic implementation

25.2.1 Examining an ASPX File

- **Examining an ASPX File**
  - **ASP.NET comments begin with <%-- and terminate with --%>**
  - `Page` **directive**
    - **Specifies the language of the code-behind file**
  - `AutoEventWireup` **attribute**
    - **Determines how Web Form events are handle**
  - `Inherits` **attribute**
    - **Specifies the class in the code-behind file from which this ASP.NET class inherits**
  - **ASP.NET markup is not case-sensitive, so using a different case is not problematic**

25.2.1 Examining an ASPX File

- runat attribute
  - Indicates that when a client requests this ASPX file:

    Process the head element and its nested elements on the server and generate the corresponding XHTML sent to the client

- asp: tag prefix in a control declaration
  - Indicates that it is an ASP.NET Web control
  - Each Web control maps to a corresponding XHTML element
  - When processing a Web control, ASP.NET generates XHTML markup that will be sent to the client to represent that control in a Web browser

- span element
  - Contains text that is displayed in a Web page

Outline

WebTime.aspx

```
<%-- Fig. 25.1: WebTime.aspx --%>
<%-- A page that displays the current time in a Label. --%>
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="WebTime.aspx.vb"
   Inherits="WebTime" EnableSessionState="False" %>

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

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
   <title>A Simple Web Form Example</title>
</head>
<body>
   <form id="form1" runat="server">
   <div>
      <h2>
      Current time on the Web server:</h2>
      <p>
         <asp:Label ID="timeLabel" runat="server" BackColor="Black"
            EnableViewState="False" Font-Size="XX-Large"
            ForeColor="Yellow"></asp:Label>
      </p>
   </div>
   </form>
</body>
</html>
```

ASP.NET comments

Page directive to specify information needed by ASP.NET to process this file

Document type declaration

Mark up of a label web control

25.2.2 Examining a Code-Behind File

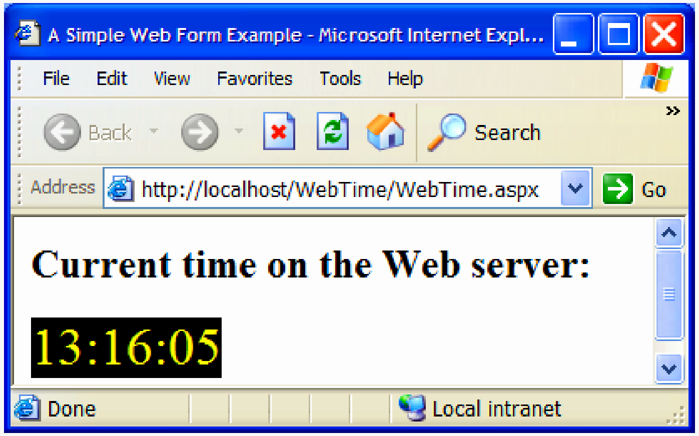
- **Examining a Code-Behind File**
  - `System.Web.UI`
    - **Contains classes and controls that assist in building Web-based applications**
    - **Class** `Page`
      - **Provides event handlers and objects necessary for creating Web-based applications**
      - **Method** `Page_Init`
        - **Handles the** `Init` **event**
        - **Indicates that a page is ready to be initialized**
    - **Class** `Control`
      - **The base class that provides common functionality for all Web controls**

WebTime.aspx.vb

```
' Fig. 25.2: WebTime.aspx.vb
' Code-behind file for a page that displays the current time.
Partial Class WebTime
   Inherits System.Web.UI.Page

   ' initializes the contents of the page
   Protected Sub Page_Init(ByVal sender As Object, _
      ByVal e As System.EventArgs) Handles Me.Init
      ' display the server's current time in timeLabel
      timeLabel.Text = DateTime.Now.ToString("hh:mm:ss")
   End Sub ' Page_Init
End Class ' WebTime
```

Retrieves the current time and formats it as hh:mm:ss



http://localhost/WebTime/WebTime.aspx

Current time on the Web server:

13:16:05

25.2.3 Relationship Between an ASPX File and a Code-Behind File

- **ASP.NET creates two `Partial` classes behind the scenes**
  - **A class defined in the code-behind file**
  - **A class based on the markup in the ASPX file**
    - **Defines the page's visual representation**
- **Web controls**
  - `System.Web.UI.WebControls`
  - **Derive from class `WebControl`**

25.2.4 How the Code in an ASP.NET Web Page Executes

- **When an instance of a page is created**
  - `PreInit` **event occurs, invoking method** `Page_PreInit`
  - `Init` **event invokes method** `Page_Init`
    - **Used to initalize objects and other aspects of the page**
  - `Load` **event occurs and the** `Page_Load` **event handler executes**
    - **Inherited from class** `Page`
  - **The page processes any events raised by the page's controls after** `Page_Load` **finishes executing**
  - `Unload` **event occurs when a Web Form object is ready for garbage collection**
    - **Event handler** `Page_Unload`
      - **Inherited from class** `Page`
      - **Contains any code that releases resources**

25.2.5 Examining the XHTML Generated by an ASP.NET Application

- XHTML Generated by an ASP.NET Application
  - XHTML forms can contain visual and nonvisual components
  - Attribute `method`
    - Specifies the method by which the Web browser submits the form to the server
  - Attribute `action`
    - Identifies the name and location of the resource that will be requested when this form is submitted
  - The `runat` attribute is removed when the form is processed on the server
    - The `method` and `action` attributes are added
    - The resulting XHTML form is sent to the client browser

Outline

WebTime.html

```
<!-- Fig. 25.3: WebTime.html -->
<!-- The XHTML generated when WebTime.aspx is loaded. -->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
   <title>A Simple Web Form Example</title>
</head>
<body>
   <form name="form1" method="post" action="WebTime.aspx" id="form1">
      <div>
         <input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value=
            "/wEPDwUJODExMDE5NzY5ZGSzVbs789nqEeoNueQCnCJQEUgykw==" />
      </div>

      <div>
         <h2>Current time on the Web server:</h2>
         <p>
            <span id="timeLabel" style="color:Yellow;
               background-color:Black;font-size:XX-Large;">13:51:12</span>
         </p>
      </div>
   </form>
</body>
</html>
```

Declaration for a non-visual component: hidden input

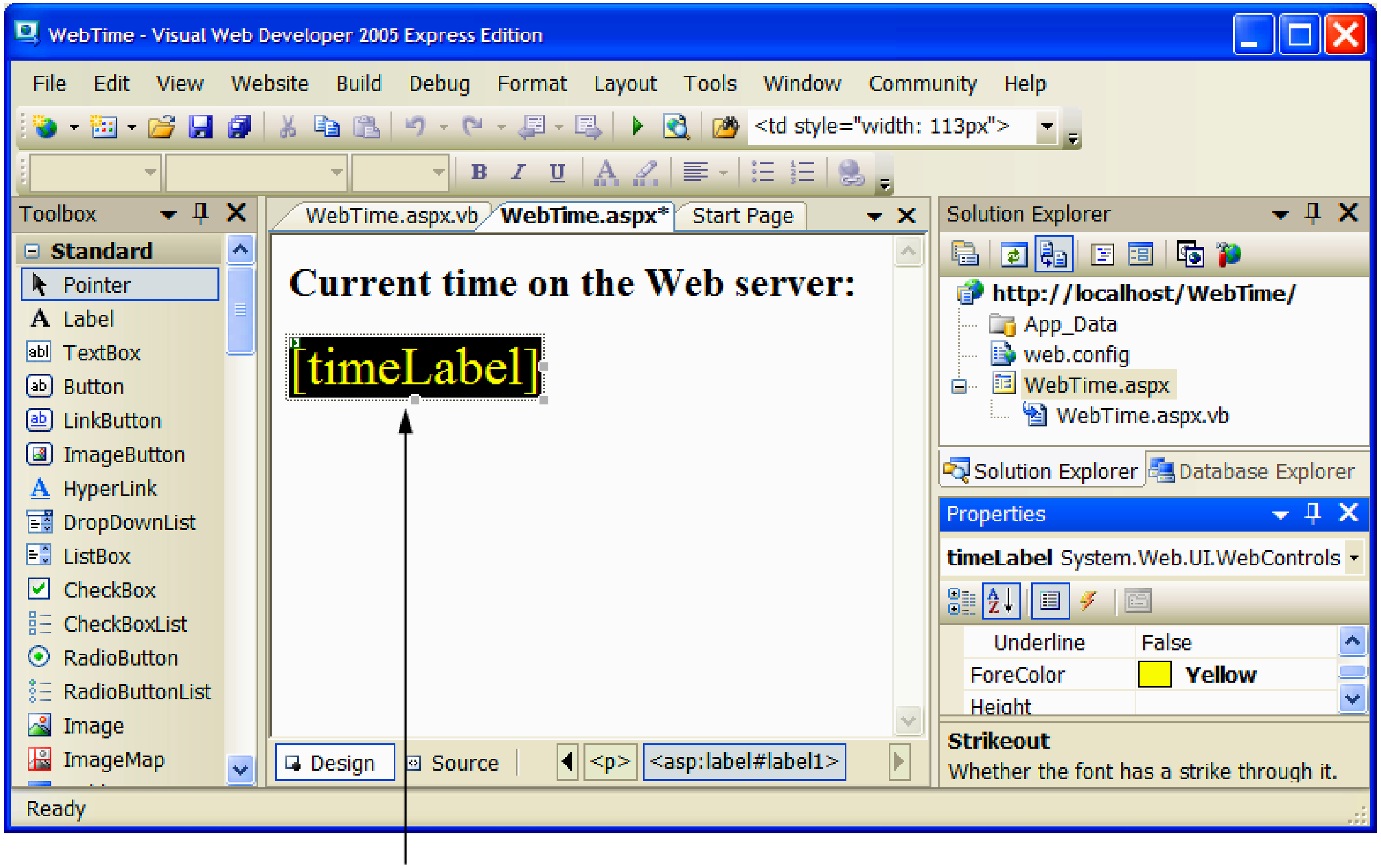
Represents the text in the label

25.2.6 Building an ASP.NET Web Application

- **Step 1: Creating the Website (Fig. 25.4)**
  - http://localhost (IIS's root directory) corresponds to C:\InetPub\wwwroot
- **Step 2: Examining the** Solution Explorer **of the Newly Created Project (Fig. 25.5)**
- **Step 3: Examining the** Toolbox **in Visual Web Developer (Fig. 25.6)**
- **Step 4: Examining the Web Forms Designer (Fig. 25.7-8)**
- **Step 5: Examining the Code-Behind File in the IDE (Fig. 25.9)**
- **Step 6: Renaming the ASPX File**
- **Step 7: Renaming the Class in the Code-Behind File and Updating the ASPX File**
- **Step 8: Changing the Title of the Page**

25.2.6 Building an ASP.NET Web Application (Cont.)

- **Step 9: Designing the Page (Fig. 25.10)**
  - Relative positioning
    - Positions of controls and other elements are relative to the Web Form’s upper-left corner
  - Absolute positioning
    - Controls are located exactly where they are dropped on the Web Form
  - Visual Web Developer is a WYSIWYG editor
    - “What You See Is What You Get”
- **Step 10: Adding Page Logic**
- **Step 11: Running the Program**
  - `Web.config` **stores configuration settings for an ASP.NET Web application**



Current time on the Web server:

[timeLabel]

timeLabel System.Web.UI.WebControls

Underline	False
ForeColor	🟨 Yellow
Height	

Strikeout
Whether the font has a strike through it.

Label

Fig. 25.11 | **`WebTime.aspx`** after adding a **`Label`** and setting its properties.

25.3.1 Text and Graphics Controls

- **Examining Web Controls on a Sample Registration Form**
  - `Image` **control**
    - **Inserts an image into a Web page**
      - `ImageUrl` **property specifies the file location of the image to display**
  - `TextBox` **control**
    - **Allows the you to obtain text from the user and display text to the user**
  - `DropDownList` **control**
    - **Provides a list of options to the user**
    - **Each item in the drop-down list is defined by a** `ListItem` **element**
  - `HyperLink` **control**
    - **Adds a hyperlink to a Web page**
      - `NavigateUrl` **property specifies the resource that is requested**
  - `RadioButtonList` **control**
    - **Provides a series of radio buttons for the user**
  - `Button` **control**
    - **Represents a button that triggers an action when clicked**

  - **Visual Web Developer displays smart tag menus for many ASP.NET controls to facilitate performing common tasks**

Outline

```
<%-- Fig. 25.13: WebControls.aspx --%>
<%-- Registration form that demonstrates Web controls. --%>
<%@ Page Language="VB" AutoEventWireup="false"
   CodeFile="WebControls.aspx.vb" Inherits="WebControls" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml11/DTD/xhtml11-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Web Controls Demonstration</title>
</head>
<body>
   <form id="form1" runat="server">
   <div>
      <h3>This is a sample registration form.</h3>
      <p>
         <em>Please fill in all fields and click Register.</em></p>
      <p>
         <asp:Image ID="userInformationImage" runat="server"
            EnableViewState="False" ImageUrl="~/Images/user.png" /> &nbsp;
         <span style="color: teal">
            Please fill out the fields below.</span>
      </p>
      <table id="TABLE1">
         <tr>
            <td style="width: 230px; height: 21px" valign="top">
               <asp:Image ID="firstNameImage" runat="server"
```

Set the color of a specific piece of text

WebControls.aspx

(1 of 5)

```
                    EnableViewState="False" ImageUrl="~/Images/fname.png" />
                 <asp:TextBox ID="firstNameTextBox" runat="server"
                    EnableViewState="False"></asp:TextBox>
              </td>
              <td style="width: 231px; height: 21px" valign="top">
                 <asp:Image ID="lastNameImage" runat="server"
                    EnableViewState="False" ImageUrl="~/Images/lname.png" />
                 <asp:TextBox ID="lastNameTextBox" runat="server"
                    EnableViewState="False"></asp:TextBox>
              </td>
           </tr>
           <tr>
              <td style="width: 230px" valign="top">
                 <asp:Image ID="emailImage" runat="server"
                    EnableViewState="False" ImageUrl="~/Images/ema
                 <asp:TextBox ID="emailTextBox" runat="server"
                    EnableViewState="False"></asp:TextBox>
              </td>
              <td style="width: 231px" valign="top">
                 <asp:Image ID="phoneImage" runat="server"
                    EnableViewState="False" ImageUrl="~/Images/phone.png" />
                 <asp:TextBox ID="phoneTextBox" runat="server"
                    EnableViewState="False"></asp:TextBox>
                 Must be in the form (555) 555-5555.
              </td>
           </tr>
        </table>
        <p>
```

Define a `TextBox` control used to collect the user’s first name

`WebControls.aspx`

(2 of 5)

```
         <asp:Image ID="publicationsImage" runat="server"
            EnableViewState="False"
            ImageUrl="~/Images/publications.png" /> &nbsp;
         <span style="color: teal">
            Which book would you like information about?</span>
      </p>
      <p>
         <asp:DropDownList ID="booksDropDownList" runat="server"
            EnableViewState="False">
            <asp:ListItem>Visual Basic 2005 How to Program 3e
                 </asp:ListItem>
            <asp:ListItem>Visual C# 2005 How to Program 2e
                 </asp:ListItem>
            <asp:ListItem>Java How to Program 6e</asp:ListItem>
            <asp:ListItem>C++ How to Program 5e</asp:ListItem>
            <asp:ListItem>XML How to Program 1e</asp:ListItem>
         </asp:DropDownList>
      </p>
      <p>
         <asp:HyperLink ID="booksHyperLink" runat="server"
            EnableViewState="False" NavigateUrl="http://www.deitel.com"
            Target="_blank">
            Click here to view more information about our books
         </asp:HyperLink>
      </p>
      <p>
         <asp:Image ID="osImage" runat="server" EnableViewState="False"
            ImageUrl="~/Images/os.png" /> &nbsp;
```

Defines a `DropDownList`

Each item in the drop-down list is defined by a `ListItem` element

Adds a hyperlink to the web page

WebControls.aspx

```
         <span style="color: teal">
            Which operating system are you using?</span>
      </p>
      <p>
         <asp:RadioButtonList ID="operatingSystemRadioButtonList"
            runat="server" EnableViewState="False">
            <asp:ListItem>Windows XP</asp:ListItem>
            <asp:ListItem>Windows 2000</asp:ListItem>
            <asp:ListItem>Windows NT</asp:ListItem>
            <asp:ListItem>Linux</asp:ListItem>
            <asp:ListItem>Other</asp:ListItem>
         </asp:RadioButtonList>
      </p>
      <p>
         <asp:Button ID="registerButton" runat="server"
            EnableViewState="False" Text="Register" />
      </p>
   </div>
   </form>
</body>
</html>
```

Defines a `RadioButtonList` control

Each item in the drop-down list is defined by a `ListItem` element

Defines a `Button` web control

25.3.2 AdRotator Control

- AdRotator control
  - Randomly selects an image to display
  - Generates a hyperlink to the Web page associated with that image
  - Browsers that do not support images display alternate text that is specified in the XML document
  - The browser loads the Web page associated with that image when a user clicks the image or substituted text

25.3.2 `AdRotator` Control (Cont.)

- `AdRotator` control
  - Accesses an XML file to determine what advertisement image, hyperlink URL and alternate text to display and include in the page
    - Create an `XmlDataSource` control to connect to the XML file
  - `ImageUrl` Element
    - Specifies the path (location) of the advertisement’s image
  - `NavigateUrl` Element
    - Specifies the URL for the Web page that loads when a user clicks the advertisement
      - The URL cannot contain any whitespace
  - `AlternateText` Element
    - Contains text that displays in place of the image when the browser cannot locate or render the image
      - Also a `tooltip`
  - `Impressions` Element
    - Specifies how often a particular image appears, relative to the other images

Outline

FlagRotator.aspx

(1 of 2)

```
<%-- Fig. 25.15: FlagRotator.aspx --%>
<%-- A Web Form that displays flags using an AdRotator control. --%>
<%@ Page Language="VB" AutoEventWireup="false"
   CodeFile="FlagRotator.aspx.vb" Inherits="FlagRotator" %>

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

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Flag Rotator</title>
</head>
<body background="Images/background.png">
    <form id="form1" runat="server">
    <div>
       <h3>AdRotator Example</h3>
       <p>
          <asp:AdRotator ID="countryRotator" runat="server"
             DataSourceID="adXmlDataSource" />
          <asp:XmlDataSource ID="adXmlDataSource" runat="server"
             DataFile="~/App_Data/AdRotatorInformation.xml">
          </asp:XmlDataSource>
       </p>
    </div>
    </form>
</body>
</html>
```

Defines an AdRotator control

Defines an XmlDataSource control, which supplies the data to the AdRotator control

Outline

AdRotatorInformation.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<!-- Fig. 25.16: AdRotatorInformation.xml -->
<!-- XML file containing advertisement information. -->
<Advertisements>
   <Ad>
      <ImageUrl>Images/france.png</ImageUrl>
      <NavigateUrl>http://www.cia.gov/library/publications/
         the-world-factbook/geos/fr.html
      </NavigateUrl>
      <AlternateText>France Information</AlternateText>
      <Impressions>1</Impressions>
   </Ad>

   <Ad>
      <ImageUrl>Images/germany.png</ImageUrl>
      <NavigateUrl>http://www.cia.gov/library/publications/
         the-world-factbook/geos/gm.html
      </NavigateUrl>
      <AlternateText>Germany Information</AlternateText>
      <Impressions>1</Impressions>
   </Ad>

   <Ad>
      <ImageUrl>Images/italy.png</ImageUrl>
      <NavigateUrl>http://www.cia.gov/library/publications/
         the-world-factbook/geos/it.html
      </NavigateUrl>
      <AlternateText>Italy Information</AlternateText>
```

Each `Ad` element represents an advertisement that will be displayed by the `AdRotator` control

Specifies the path of the advertisement’s image

Specifies the URL for the Web page that loads when user clicks on the advertisement

Text that displays in place of the image when the browser cannot render the image

Specifies how often a particular image appears

25.3.3 Validation Controls

- `Validation` **Control**
  - **Determines whether the data in another Web control is in the proper format**
  - **Provide a mechanism for validating user input on the client**
  - **The validator is converted into ECMAScript when the XHTML for the page is created**
    - **ECMAScript is a scripting language that enhances the functionality and appearance of Web pages**
      - **For security reasons, validation is always performed on the server**

25.3.3 Validation Controls (Cont.)

- `ControlToValidate` **property**
  - **Indicates** `Validator` **verifies a control's contents**
- `ErrorMessage` **Property**
  - **Text displayed if the validation fails**
- `Display` **property**
  - **Determines if the validator is displayed only when validation fails**
- `RegularExpressionValidator`
  - **Matches a Web control's content against a regular expression**
    - **That expression is assigned to** `ValidationExpression` **property**
- `IsPostBack` **property**
  - **Determines whether the page is being loaded due to a postback**
    - **The current page reloads when the user submits the form**

- `EnableViewState` **attribute**
  - **Determines whether a Web control's state persists when a postback occurs**

Outline

Validation.aspx

(1 of 5)

```
<%-- Fig. 25.17: Validation.aspx --%>
<%-- Form that demonstrates using validators to validate user input. --%>
<%@ Page Language="VB" AutoEventWireup="false"
   CodeFile="Validation.aspx.vb" Inherits="Validation" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml11/DTD/xhtml11-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Demonstrating Validation Controls</title>
</head>
<body>
   <form id="form1" runat="server">
   <div>
      Please fill out the following form.<br /><em>All fields are
      required and must contain valid information.</em><br />
      <br />
      <table>
         <tr>
            <td style="width: 100px" valign="top">
               Name:</td>
            <td style="width: 450px" valign="top">
               <asp:TextBox ID="nameTextBox" runat="server">
               </asp:TextBox><br />
               <asp:RequiredFieldValidator
                  ID="nameInputValidator" runat="server"
                  ControlToValidate="nameTextBox" Display="Dynamic"
                  ErrorMessage="Please enter your name.">
               </asp:RequiredFieldValidator>
```

Defines a `RequiredFieldValidator` which confirms the `TextBox` is not empty

```
               </td>
            </tr>
            <tr>
               <td style="width: 100px" valign="top">E-mail address:</td>
               <td style="width: 450px" valign="top">
                  <asp:TextBox ID="emailTextBox" runat="server">
                  </asp:TextBox>
                  &nbsp;e.g., user@domain.com<br />
                  <asp:RequiredFieldValidator
                     ID="emailInputValidator" runat="server"
                     ControlToValidate="emailTextBox" Display="Dynamic"
                     ErrorMessage="Please enter your e-mail address.">
                  </asp:RequiredFieldValidator>
                  <asp:RegularExpressionValidator
                     ID="emailFormatValidator" runat="server"
                     ControlToValidate="emailTextBox" Display="Dynamic"
                     ErrorMessage=
                        "Please enter an e-mail address in a valid format."
                     ValidationExpression=
                        "\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*">
                  </asp:RegularExpressionValidator>
               </td>
            </tr>
            <tr>
               <td style="width: 100px; height: 21px" valign="top">
                  Phone number:</td>
               <td style="width: 450px; height: 21px" valign="top">
                  <asp:TextBox ID="phoneTextBox" runat="server">
                  </asp:TextBox>
```

Defines a `RequiredFieldValidator` which confirms the `TextBox` is not empty

Defines a `RegularExpressionValidator` control to determine whether data is entered in a valid format

```
               &nbsp;e.g., (555) 555-1234<br />
               <asp:RequiredFieldValidator
                  ID="phoneInputValidator" runat="server"
                  ControlToValidate="phoneTextBox" Display="Dynamic"
                  ErrorMessage="Please enter your phone number.">
               </asp:RequiredFieldValidator>
               <asp:RegularExpressionValidator
                  ID="phoneFormatValidator" runat="server"
                  ControlToValidate="phoneTextBox" Display="Dynamic"
                  ErrorMessage=
                     "Please enter a phone number in a valid format."
                  ValidationExpression=
                     "((\(\d{3}\) ?)|(\d{3}-))?\d{3}-\d{4}">
               </asp:RegularExpressionValidator>
            </td>
         </tr>
      </table>
      <br />
      <asp:Button ID="submitButton" runat="server" Text="Submit" /><br />
      <br />
      <asp:Label ID="outputLabel" runat="server"
         Text="Thank you for your submission." Visible="False">
      </asp:Label>
   </div>
   </form>
</body>
</html>
```

Defines a `RequiredFieldValidator` which confirms the `TextBox` is not empty

`Validation.aspx`

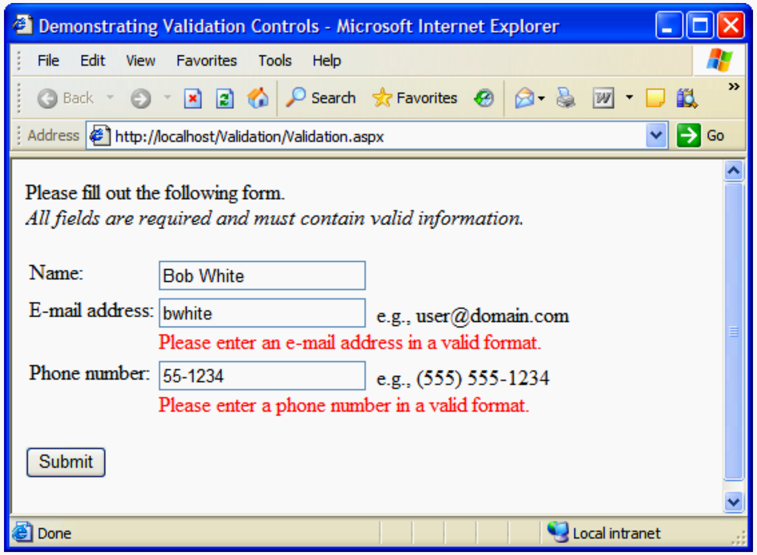
(3 of 5)

Defines a `RegularExpressionValidator` control to determine whether data is entered in a valid format

Validation.aspx

(5 of 5)

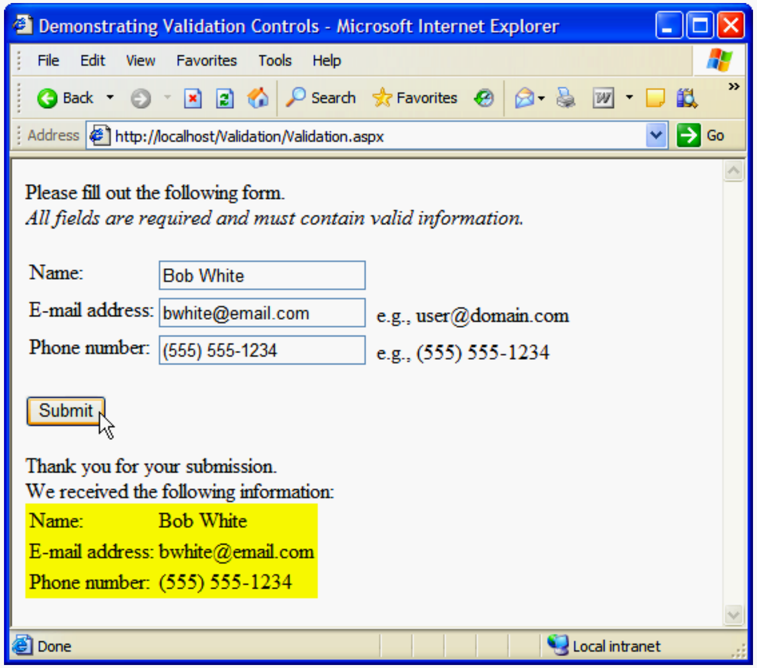
(c)


Demonstrating Validation Controls - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Validation/Validation.aspx Go
Submit
Done
Local intranet

Please fill out the following form.
All fields are required and must contain valid information.

Name: Bob White
E-mail address: bwhite e.g., user@domain.com
Please enter an e-mail address in a valid format.
Phone number: 55-1234 e.g., (555) 555-1234
Please enter a phone number in a valid format.

(d)


Demonstrating Validation Controls - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Validation/Validation.aspx Go
Submit
Done
Local intranet

Please fill out the following form.
All fields are required and must contain valid information.

Name: Bob White
E-mail address: bwhite@email.com e.g., user@domain.com
Phone number: (555) 555-1234 e.g., (555) 555-1234

Thank you for your submission.
We received the following information:

Name:	Bob White
E-mail address:	bwhite@email.com
Phone number:	(555) 555-1234

Outline

Validation.aspx .vb

```
' Fig. 25.18: Validation.aspx.vb
' Code-behind file for the form demonstrating validation controls.
Partial Class Validation
   Inherits System.Web.UI.Page

   ' Page_Load event handler executes when the page is loaded
   Protected Sub Page_Load(ByVal sender As Object, _
      ByVal e As System.EventArgs) Handles Me.Load
      ' if this is not the first time the page is loading
      ' (i.e., the user has already submitted form data)
      If IsPostBack Then
         ' retrieve the values submitted by the user
         Dim name As String = nameTextBox.Text
         Dim email As String = emailTextBox.Text
         Dim phone As String = phoneTextBox.Text

         ' create a table indicating the submitted values
         outputLabel.Text &= _
            "<br />We received the following information:" & _
            "<table style=""background-color: yellow"">" & _
            "<tr><td>Name: </td><td>" & name & "</td></tr>" & _
            "<tr><td>E-mail address: </td><td>" & email & "</td></tr>" & _
            "<tr><td>Phone number: </td><td>" & phone & "</td></tr>" & _
            "<table>"
         outputLabel.Visible = True ' display the output message
      End If
   End Sub ' Page_Load
End Class ' Validation
```

Determines whether the page is being loaded due to a postback

Retrieves the values of the `TextBox`es

25.4 Session Tracking

- **Session Tracking**
  - **Tracking individual clients**
  - **The request/response system on which the Web operates is facilitated by HTTP**
    - **HTTP is a stateless protocol**
      - **Does not support persistent connections**
        - **Web servers cannot maintain state information regarding particular clients**
  - **A session represents a unique client on a Web site**
    - **The client will still be recognized as the same user if the client leaves a site and returns later**
      - **Each client must identify itself to the server**
        - **Help the server distinguish among clients**

25.4.1 Cookies

- Cookies
  - Piece of data stored in a small text file on the user's computer
    - Maintains information about the client during and between browser sessions
  - Are sent and received as a collection of type `HttpCookieCollection`
  - Web Forms can examine the cookies it sent to the client during previous communications
  - The expiration date determines how long the cookie remains on the client's computer
    - The Web browser maintains the cookie for the duration of the browsing session if there is no expiration date
  - `Response` object's `Cookies` property
    - Used to write cookies to a client
    - Used to access cookies
  - Can be read by an application only if they were created in the domain in which the application is running

25.4.2 Session Tracking with `HttpSessionState`

- Class `HttpSessionState`
  - Provides session-tracking
  - Every Web Form includes an `HttpSessionState` object
    - Accessible through property `Session` of class `Page`
  - Can store name–value pairs using method `Add`
    - When adding an attribute that has the same name, the object associated with that attribute is replaced

Properties	Description
`Count`	Specifies the number of key–value pairs in the `Session` object.
`IsNewSession`	Indicates whether this is a new session (i.e., whether the session was created during loading of this page).
`IsReadOnly`	Indicates whether the `Session` object is read-only.
`Keys`	Returns a collection containing the `Session` object’s keys.
`SessionID`	Returns the session’s unique ID.
`Timeout`	Specifies the maximum number of minutes during which a session can be inactive (i.e., no requests are made) before the session expires. By default, this property is set to 20 minutes.

Fig. 25.27 | **HttpSessionState** properties.

```
<%-- Fig. 25.25: Options.aspx --%>
<%-- Allows client to select programming languages and access --%>
<%-- book recommendations. --%>
<%@ Page Language="VB" AutoEventWireup="false"
   CodeFile="Options.aspx.vb" Inherits="Options" %>

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

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Sessions</title>
</head>
<body>
   <form id="form1" runat="server">
   <div>
      <asp:Label ID="promptLabel" runat="server" Font-Bold="True"
         Font-Size="Large" Text="Select a programming language:">
      </asp:Label>
      <asp:RadioButtonList ID="languageList" runat="server">
         <asp:ListItem>Visual Basic 2005</asp:ListItem>
         <asp:ListItem>Visual C# 2005</asp:ListItem>
         <asp:ListItem>C</asp:ListItem>
         <asp:ListItem>C++</asp:ListItem>
         <asp:ListItem>Java</asp:ListItem>
```

Options.aspx

(1 of 3)

Outline

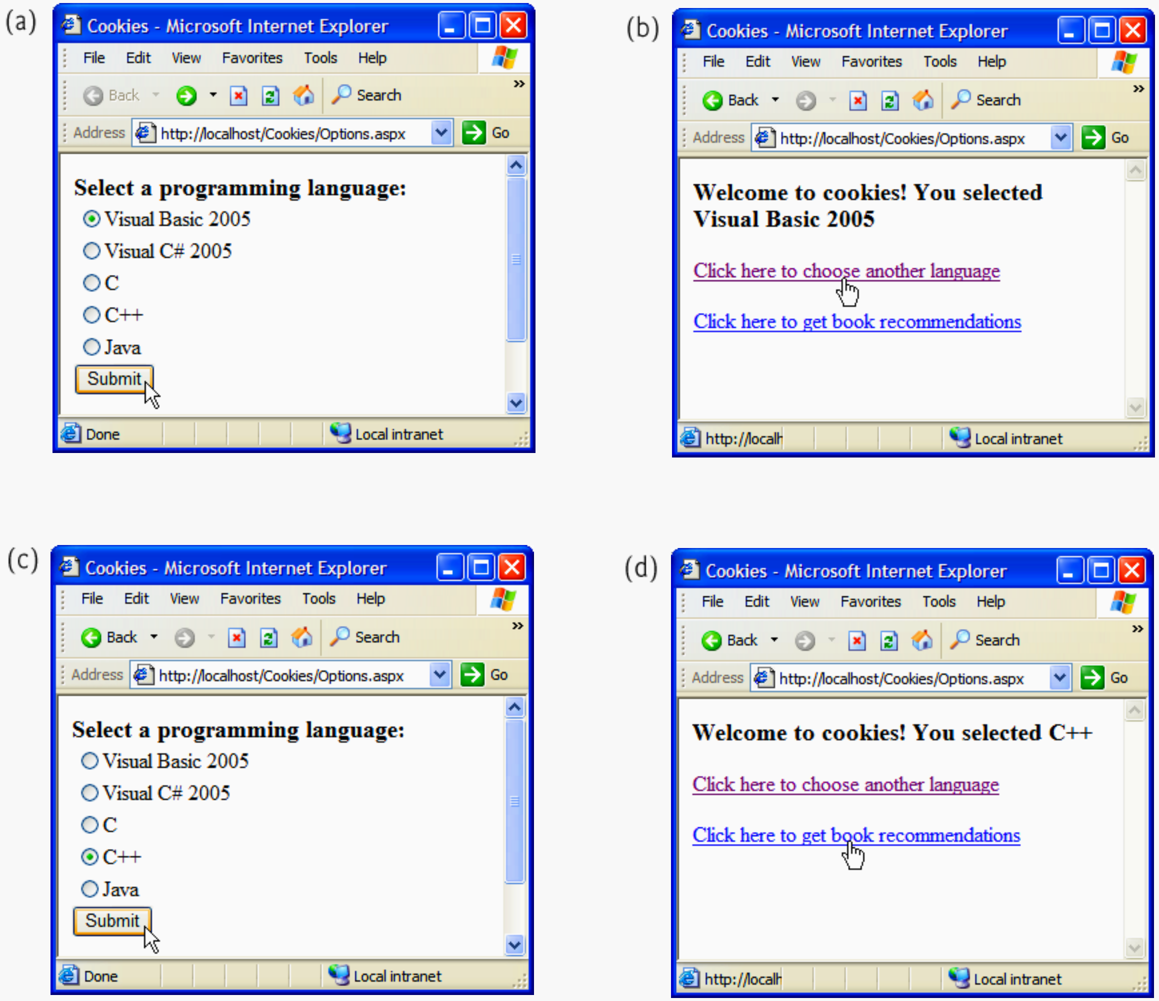
Options.aspx

(2 of 3)

```
      </asp:RadioButtonList>
      <asp:Button ID="submitButton" runat="server" Text="Submit" />
      <asp:Label ID="responseLabel" runat="server" Font-Bold="True"
         Font-Size="Large" Text="Welcome to sessions!" Visible="False">
      </asp:Label><br />
      <br />
      <asp:Label ID="idLabel" runat="server" Visible="False">
      </asp:Label><br />
      <br />
      <asp:Label ID="timeoutLabel" runat="server" Visible="False">
      </asp:Label><br />
      <br />
      <asp:HyperLink ID="languageLink" runat="server"
         NavigateUrl="~/Options.aspx" Visible="False">
         Click here to choose another language
      </asp:HyperLink><br />
      <br />
      <asp:HyperLink ID="recommendationsLink" runat="server"
         NavigateUrl="~/Recommendations.aspx" Visible="False">
         Click here to get book recommendations
      </asp:HyperLink>
   </div>
   </form>
</body>
</html>
```

Hidden labels to display tracking information

Outline


(a)
Cookies - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search
Address Go
Done
Local intranet
(b)
(c)
(d)
http://localh

(a)

Address: http://localhost/Cookies/Options.aspx

Select a programming language:

- ● Visual Basic 2005
- ○ Visual C# 2005
- ○ C
- ○ C++
- ○ Java

Submit

(b)

Address: http://localhost/Cookies/Options.aspx

Welcome to cookies! You selected Visual Basic 2005

Click here to choose another language

Click here to get book recommendations

(c)

Address: http://localhost/Cookies/Options.aspx

Select a programming language:

- ○ Visual Basic 2005
- ○ Visual C# 2005
- ○ C
- ● C++
- ○ Java

Submit

(d)

Address: http://localhost/Cookies/Options.aspx

Welcome to cookies! You selected C++

Click here to choose another language

Click here to get book recommendations

`Options.aspx`

(3 of 3)

```
' Fig. 25.26: Options.aspx.vb
' Processes user's selection of a programming language
' by displaying links and writing information in a Session object.
Partial Class Options
   Inherits System.Web.UI.Page
   ' stores values to represent books
   Private books As New System.Collections.Hashtable()

   ' initializes the Hashtable of values to be stored in a Session
   Protected Sub Page_Init(ByVal sender As Object, _
      ByVal e As System.EventArgs) Handles Me.Init
      books.Add("Visual Basic 2005", "0-13-186900-0")
      books.Add("Visual C# 2005", "0-13-152523-9")
      books.Add("C", "0-13-142644-3")
      books.Add("C++", "0-13-185757-6")
      books.Add("Java", "0-13-148398-6")
   End Sub ' Page_Init

   ' if postback, hide form and display links to make additional
   ' selections or view recommendations
   Protected Sub Page_Load(ByVal sender As Object, _
      ByVal e As System.EventArgs) Handles Me.Load

```

Outline

Options.aspx.vb

(1 of 3)

```
      If IsPostBack Then
         ' user has submitted information, so display message
         ' and appropriate hyperlinks
         responseLabel.Visible = True
         idLabel.Visible = True
         timeoutLabel.Visible = True
         languageLink.Visible = True
         recommendationsLink.Visible = True

         ' hide other controls used to make language selection
         promptLabel.Visible = False
         languageList.Visible = False
         submitButton.Visible = False

         ' if the user made a selection, display it in responseLabel
         If languageList.SelectedItem IsNot Nothing Then
            responseLabel.Text &= " You selected " & _
               languageList.SelectedItem.Text.ToString()
         Else
            responseLabel.Text &= " You did not select a language."
         End If

         ' display session ID
         idLabel.Text = "Your unique session ID is: " & Session.SessionID

```

Returns the unique session ID

Options.aspx.vb

(2 of 3)

Outline

```
         ' display the timeout
         timeoutLabel.Text = "Timeout: " & Session.Timeout & " minutes."
      End If
   End Sub ' Page_Load

   ' record the user's selection in the Session
   Protected Sub submitButton_Click(ByVal sender As Object, _
      ByVal e As System.EventArgs) Handles submitButton.Click
      ' if the user made a selection
      If languageList.SelectedItem IsNot Nothing Then
         Dim language As String = languageList.SelectedItem.ToString()

         ' get ISBN number of book for the given language
         Dim ISBN As String = books(language).ToString()

         Session.Add(language, ISBN) ' add name/value pair to Session
      End If
   End Sub ' submitButton_Click
End Class ' Options
```

Returns the maximum amount of time that an `HttpSessionState` object can be inactive before it is discarded

`Options.aspx.vb`

(3 of 3)

Place the language and its corresponding ISBN number in the `HttpSessionState` object

Outline

Recommendations.aspx

(1 of 2)

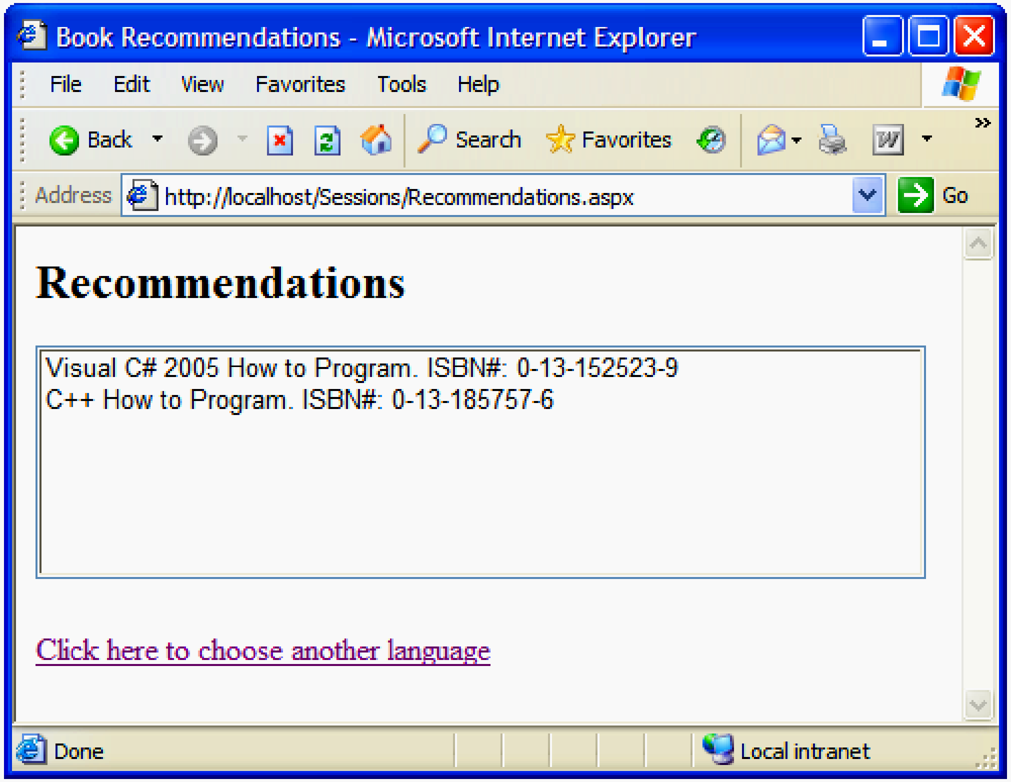
```
<%-- Fig. 25.28: Recommendations.aspx --%>
<%-- Displays book recommendations using a Session object. --%>
<%@ Page Language="VB" AutoEventWireup="false"
   CodeFile="Recommendations.aspx.vb" Inherits="Recommendations" %>

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

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Book Recommendations</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
       <asp:Label ID="recommendationsLabel" runat="server"
          Font-Bold="True" Font-Size="X-Large" Text="Recommendations">
       </asp:Label><br />
       <br />
       <asp:ListBox ID="booksListBox" runat="server" Height="125px"
          Width="450px"></asp:ListBox><br />
       <br />
       <asp:HyperLink ID="languageLink" runat="server"
          NavigateUrl="~/Options.aspx">
          Click here to choose another language
       </asp:HyperLink>&nbsp;</div>
    </form>
</body>
</html>
```

Used to present the recommendations to the user

Outline


Book Recommendations - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Sessions/Recommendations.aspx
Go
Done
Local intranet

Recommendations

Visual C# 2005 How to Program. ISBN#: 0-13-152523-9
C++ How to Program. ISBN#: 0-13-185757-6

Click here to choose another language

`Recommendations.aspx`

(2 of 2)

```
' Fig. 25.29: Recommendations.aspx.vb
' Creates book recommendations based on a Session object.
Partial Class Recommendations
   Inherits System.Web.UI.Page

   ' read Session items and populate ListBox with any book recommendations
   Protected Sub Page_Init(ByVal sender As Object, _
      ByVal e As System.EventArgs) Handles Me.Init
      ' determine whether Session contains any information
      If Session.Count <> 0 Then
         For i As Integer = 0 To Session.Count - 1
            ' get current key name from Session object
            Dim keyName As String = Session.Keys(i)

            ' use keyName to display one of Session's name-value pairs
            booksListBox.Items.Add(keyName & _
               " How to Program. ISBN#: " & _
               Session(keyName).ToString())
```

Provides the number of session items contained in a `Session` object

Recommendations .aspx.vb

(1 of 2)

Indexes the collection containing all the keys in the session to receive the current key

Concatenate and display the string that is the recommendation in the `ListBox`

```
         Next
      Else
         ' if there are no session items, no language was chosen, so
         ' display appropriate message and clear and hide booksListBox
         recommendationsLabel.Text = "No Recommendations"
         booksListBox.Items.Clear()
         booksListBox.Visible = False

         ' modify languageLink because no language was selected
         languageLink.Text = "Click here to choose a language"
      End If
   End Sub ' Page_Init
End Class ' Recommendations
```

Outline

Recommendations.aspx.vb

(2 of 2)

25.5.1 Building a Web Form That Displays Data from a Database

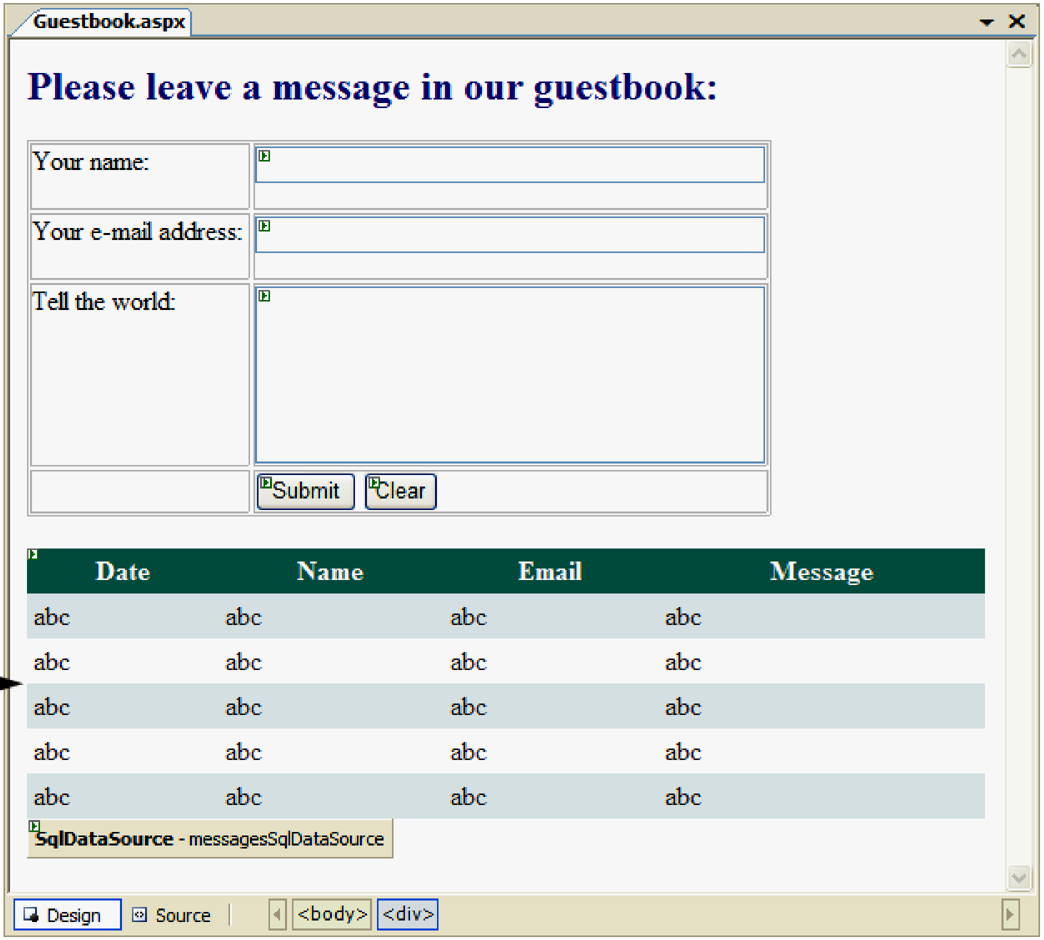
- **Step 1: Creating the Project**
- **Step 2: Creating the Form for User Input (Fig. 25.30)**
- **Step 3: Adding a `GridView` Control to the Web Form**
- **Step 4: Adding a Database to an ASP.NET Web Application**
  - **Database used by an ASP.NET Web site should be located in the project's `App_Data` folder**
- **Step 5: Binding the `GridView` to the Messages Table of the Guestbook Database (Fig. 25.31-34)**
- **Step 6: Modifying the Columns of the Data Source Displayed in the `GridView` (Fig. 25.35)**
- **Step 7: Modifying the Way the `SqlDataSource` Control Inserts Data (Fig. 25.36)**
  - **Each column is represented as a `BoundField`**

25.5.1 Building a Web Form That Displays Data from a Database (Cont.)

- `GridView`
  - `DataSourceID` **property**
    - **Identifies the data source that is used to fill the** `GridView` **with data at runtime**
  - `HeaderText` **property**
    - **Indicates the text that appears as the column header**
      - **By default, this is the name of the column in the data source**
  - **Each column is represented as a** `BoundField`

25.5.1 Building a Web Form That Displays Data from a Database (Cont.)

- `SqlDataSource`
  - `ConnectionString` **property**
    - **Indicates the connection that the `SqlDataSource` control interacts with**
    - **The value of this property uses an ASP.NET expression, delimited by <%$ and %>**
  - `SelectCommand`**,** `DeleteCommand`**,** `InsertCommand` **and** `UpdateCommand` **properties**
    - **Contain the `SELECT`, `DELETE`, `INSERT` and `UPDATE` SQL statements, respectively**
  - **Parameters that are set programmatically are defined by** `Parameter` **elements**
  - **Parameters that obtain their values from controls are defined by** `ControlParameter` **elements**



Please leave a message in our guestbook:

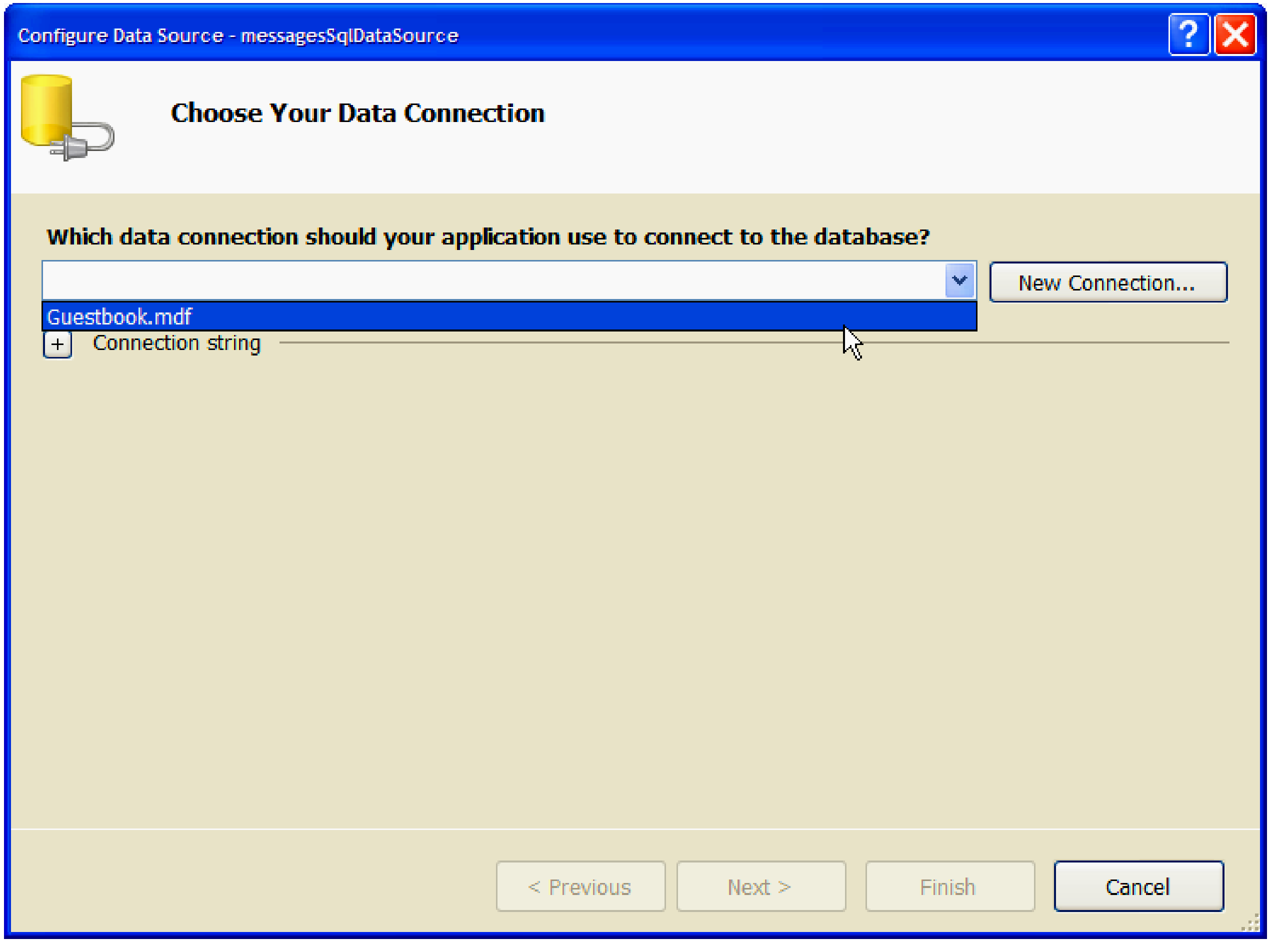
Your name:	
Your e-mail address:	
Tell the world:	
	Submit Clear

Date	Name	Email	Message
abc	abc	abc	abc
abc	abc	abc	abc
abc	abc	abc	abc
abc	abc	abc	abc
abc	abc	abc	abc

SqlDataSource - messagesSqlDataSource

GridView control

Fig. 25.30 | Guestbook application GUI in **Design** mode.



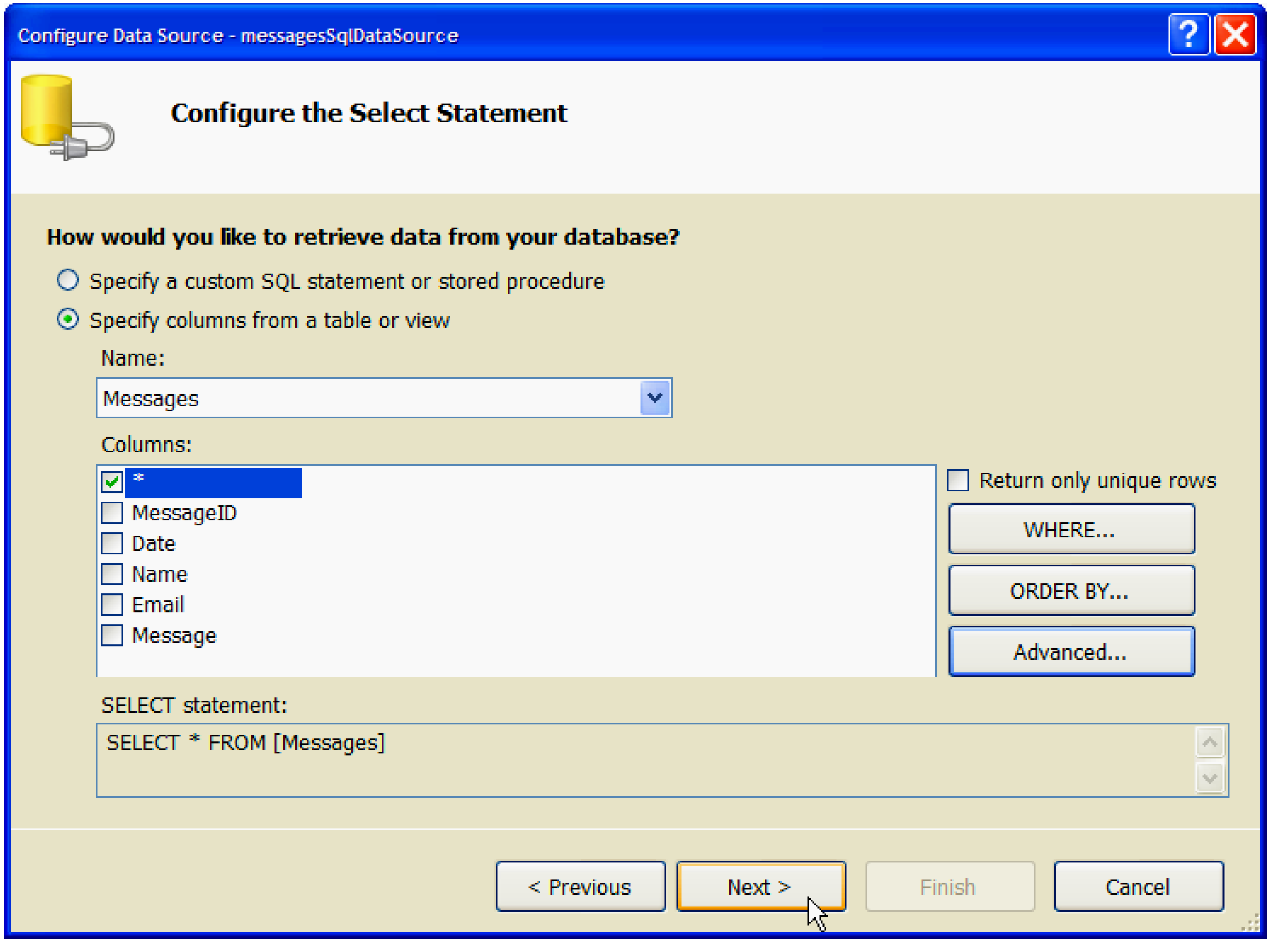
Configure Data Source - messagesSqlDataSource

Choose Your Data Connection

Which data connection should your application use to connect to the database?

Guestbook.mdf

Fig. 25.31 | **Configure Data Source** dialog in Visual Web Developer.



Configure Data Source - messagesSqlDataSource

Configure the Select Statement

How would you like to retrieve data from your database?

○ Specify a custom SQL statement or stored procedure

● Specify columns from a table or view

Name: Messages

Columns:

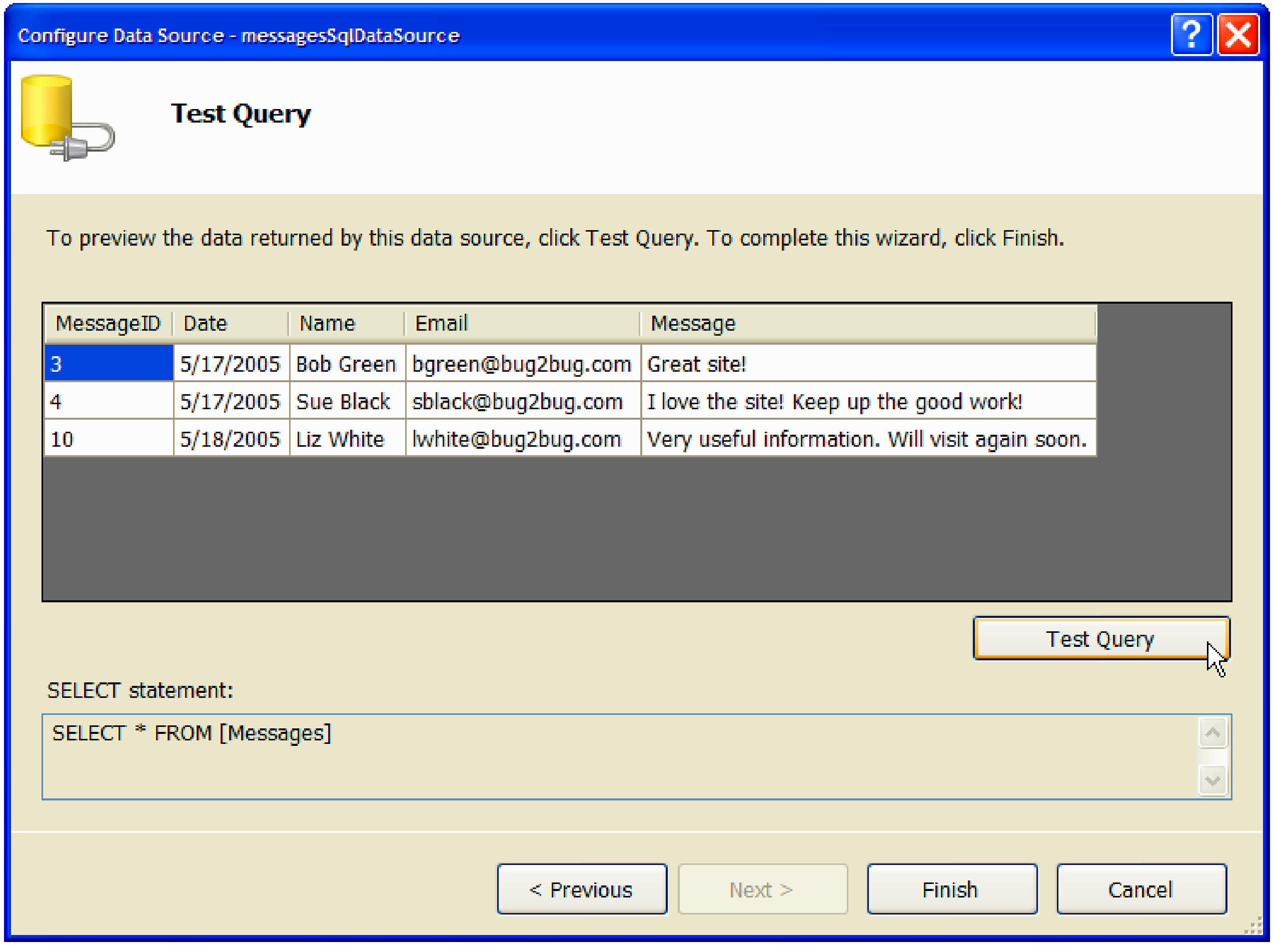
☑ *
□ MessageID
□ Date
□ Name
□ Email
□ Message

□ Return only unique rows

SELECT statement:

```
SELECT * FROM [Messages]
```

Fig. 25.32 | Configuring the **`SELECT`** statement used by the **`SqlDataSource`** to retrieve data.



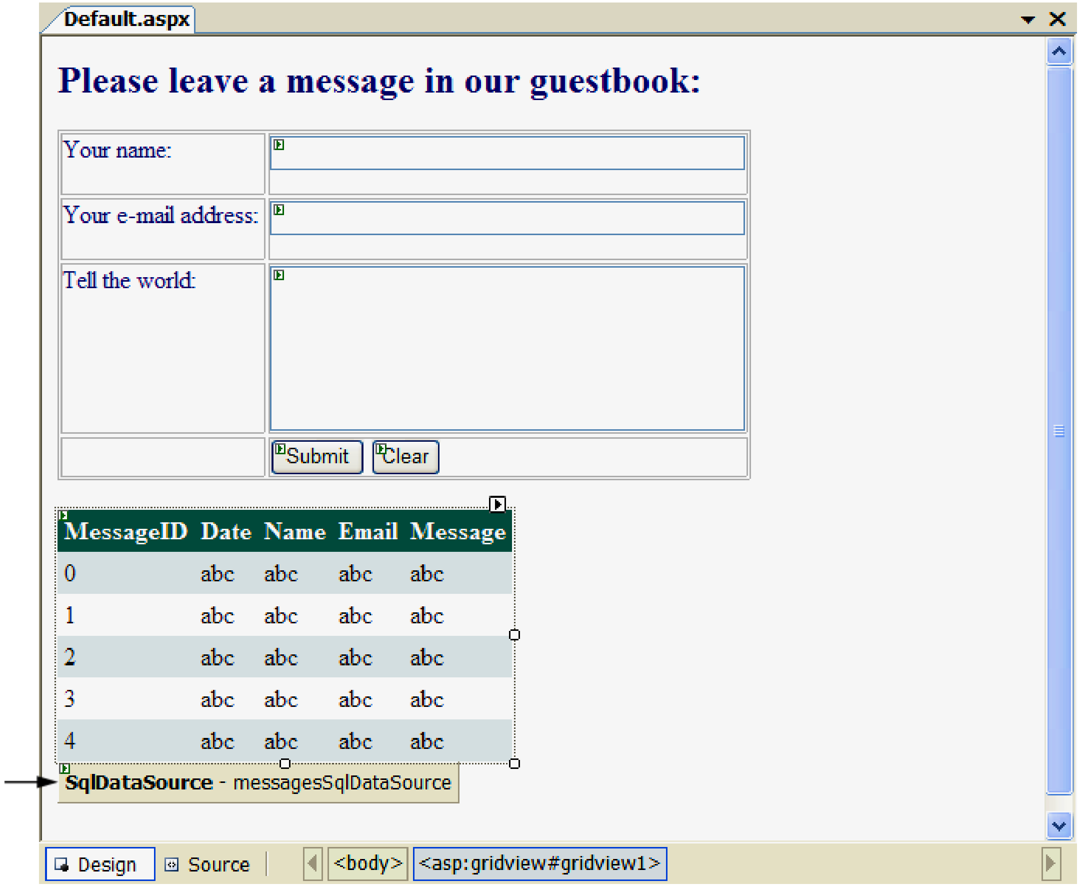
To preview the data returned by this data source, click Test Query. To complete this wizard, click Finish.

MessageID	Date	Name	Email	Message
3	5/17/2005	Bob Green	bgreen@bug2bug.com	Great site!
4	5/17/2005	Sue Black	sblack@bug2bug.com	I love the site! Keep up the good work!
10	5/18/2005	Liz White	lwhite@bug2bug.com	Very useful information. Will visit again soon.

SELECT statement:

```
SELECT * FROM [Messages]
```

Fig. 25.33 | Previewing the data retrieved by the `SqlDataSource`.



Please leave a message in our guestbook:

Your name:

Your e-mail address:

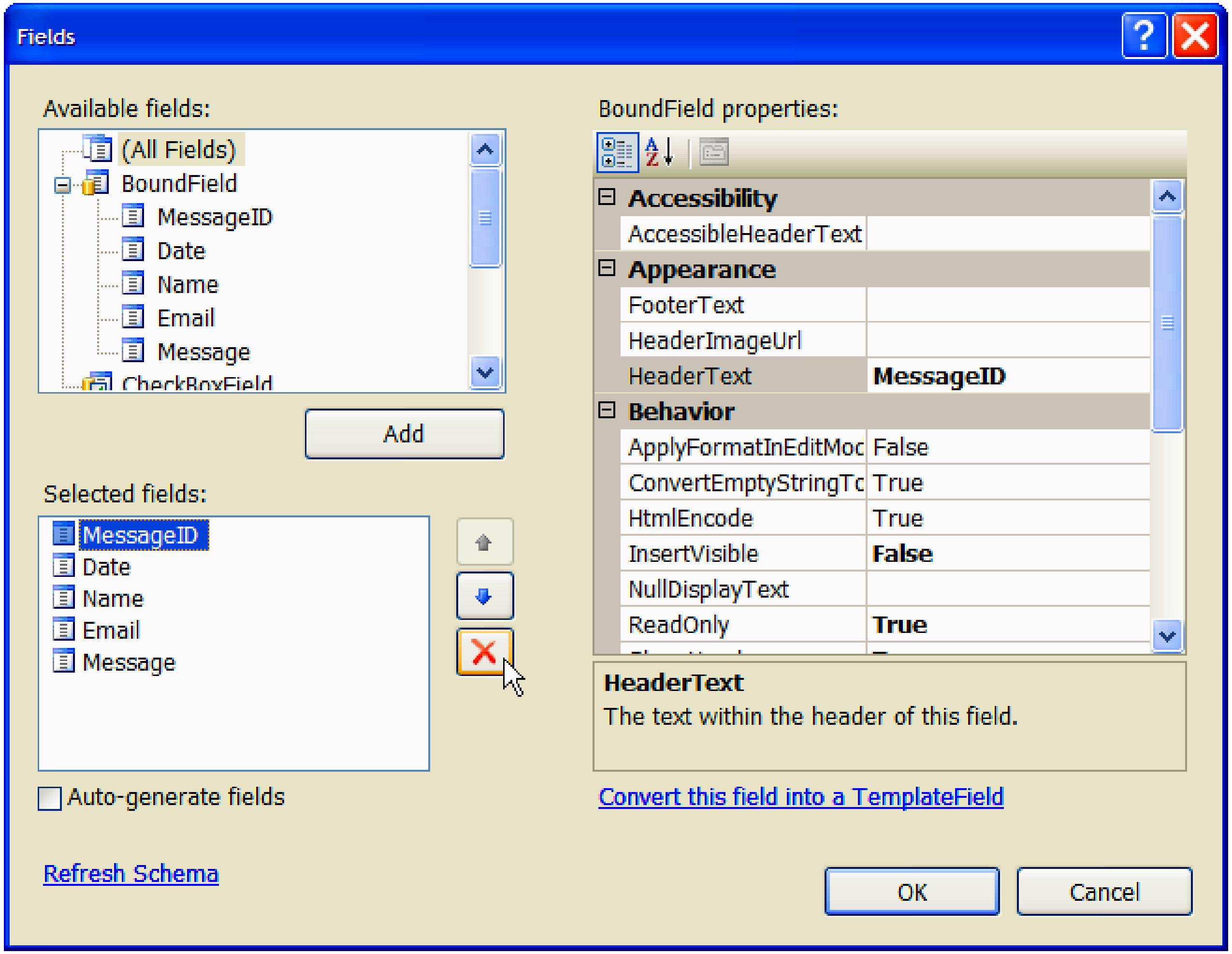
Tell the world:

MessageID	Date	Name	Email	Message
0	abc	abc	abc	abc
1	abc	abc	abc	abc
2	abc	abc	abc	abc
3	abc	abc	abc	abc
4	abc	abc	abc	abc

SqlDataSource - messagesSqlDataSource

SqlDataSource control

Fig. 25.34 | **Design** mode displaying `SqlDataSource` control for a `GridView`.



Fields

Available fields:

- (All Fields)
- BoundField
  - MessageID
  - Date
  - Name
  - Email
  - Message
- CheckBoxField

Selected fields:

- MessageID
- Date
- Name
- Email
- Message

□ Auto-generate fields

BoundField properties:

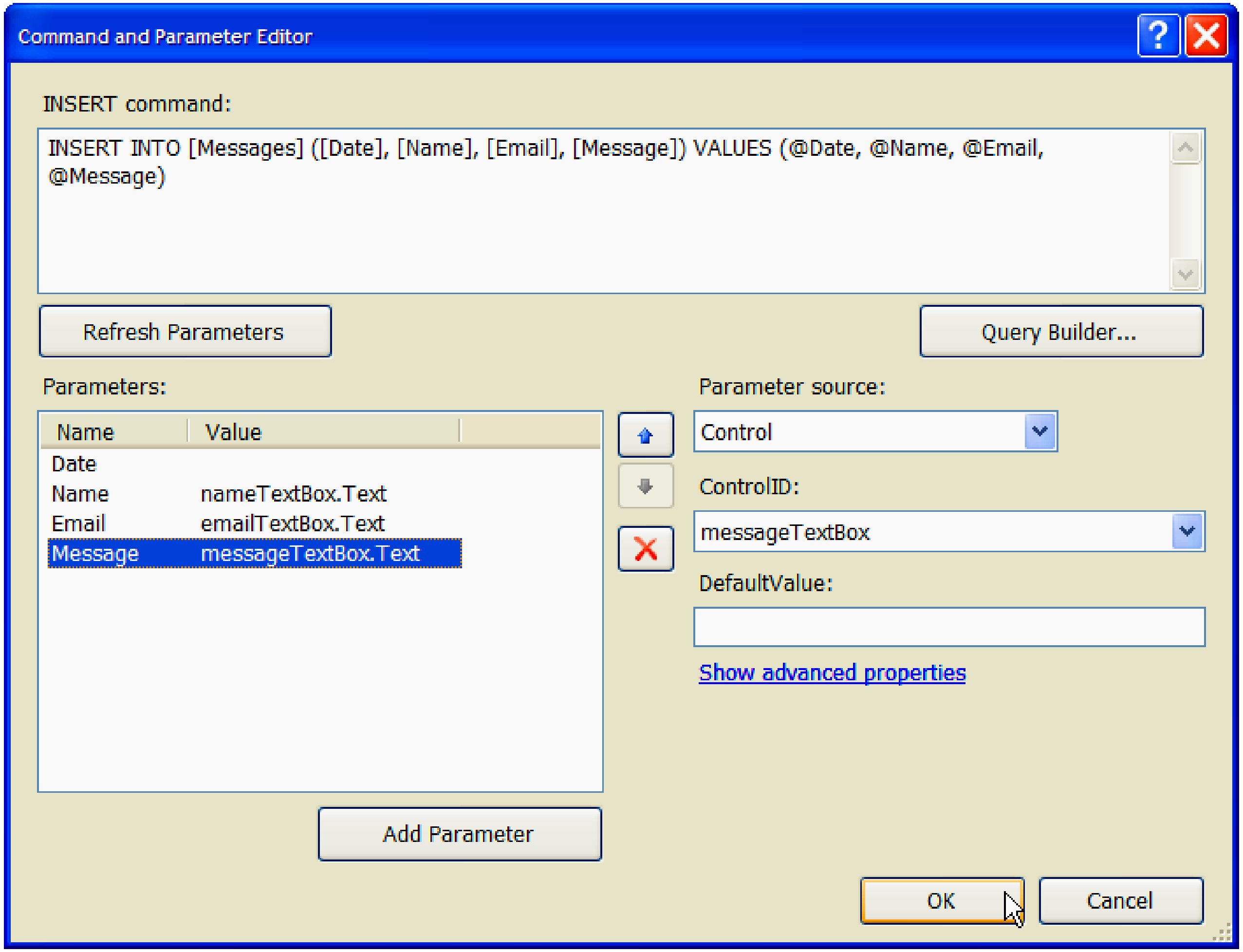
Accessibility	
AccessibleHeaderText	
Appearance	
FooterText	
HeaderImageUrl	
HeaderText	**MessageID**
Behavior	
ApplyFormatInEditMoc	False
ConvertEmptyStringTc	True
HtmlEncode	True
InsertVisible	**False**
NullDisplayText	
ReadOnly	**True**

HeaderText
The text within the header of this field.

Convert this field into a TemplateField

Refresh Schema

Fig. 25.35 | Removing the **`MessageID`** column from the **`GridView`**.



Command and Parameter Editor

INSERT command:

```
INSERT INTO [Messages] ([Date], [Name], [Email], [Message]) VALUES (@Date, @Name, @Email,
@Message)
```

Parameters:

Name	Value
Date	
Name	nameTextBox.Text
Email	emailTextBox.Text
Message	messageTextBox.Text

Parameter source: Control

ControlID: messageTextBox

DefaultValue:

Fig. 25.36 | Setting up **INSERT** parameters based on control values.

Outline

```
<%-- Fig. 25.37: Guestbook.aspx --%>
<%-- Guestbook Web application with a form for users to submit --%>
<%-- guestbook entries and a GridView to view existing entries. --%>
<%@ Page Language="VB" AutoEventWireup="false"
   CodeFile="Guestbook.aspx.vb" Inherits="Guestbook" %>

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

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Guestbook</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
       <h2>
          Please leave a message in our guestbook:</h2>
       <table>
          <tr>
             <td style="width: 130px; height: 21px" valign="top">
                Your name:<br />
             </td>
             <td style="width: 300px; height: 21px" valign="top">
                <asp:TextBox ID="nameTextBox" runat="server"
                   Width="300px"></asp:TextBox>
             </td>
          </tr>
```

Guestbook.aspx

(1 of 6)

Outline

Guestbook.aspx

(2 of 6)

```
            <tr>
               <td style="width: 130px" valign="top">
                  Your e-mail address:<br />
               </td>
               <td style="width: 300px" valign="top">
                  <asp:TextBox ID="emailTextBox" runat="server"
                     Width="300px"></asp:TextBox>
               </td>
            </tr>
            <tr>
               <td style="width: 130px" valign="top">
                  Tell the world:<br />
               </td>
               <td style="width: 300px" valign="top">
                  <asp:TextBox ID="messageTextBox" runat="server"
                     Height="100px" Rows="8" Width="300px">
                    </asp:TextBox>
               </td>
            </tr>
            <tr>
               <td style="width: 130px" valign="top">
               </td>
               <td style="width: 300px" valign="top">
                  <asp:Button ID="submitButton" runat="server"
                     Text="Submit" />
                  <asp:Button ID="clearButton" runat="server"
                     Text="Clear" />
               </td>
```

```
         </tr>
      </table>
      <br />
      <asp:GridView ID="messagesGridView" runat="server"
         AutoGenerateColumns="False" CellPadding="4"
         DataKeyNames="MessageID" DataSourceID="messagesSqlDataSource
         ForeColor="#333333" GridLines="None" Width="600px">
         <FooterStyle BackColor="#1C5E55" Font-Bold="True"
            ForeColor="White" />
         <Columns>
            <asp:BoundField DataField="Date" HeaderText="Date"
               SortExpression="Date" />
            <asp:BoundField DataField="Name" HeaderText="Name"
               SortExpression="Name" />
            <asp:BoundField DataField="Email" HeaderText="Email"
               SortExpression="Email" />
            <asp:BoundField DataField="Message" HeaderText="Message"
               SortExpression="Message" />
         </Columns>
         <RowStyle BackColor="#E3EAEB" />
         <EditRowStyle BackColor="#7C6F57" />
         <SelectedRowStyle BackColor="#C5BBAF" Font-Bold="True"
            ForeColor="#333333" />
         <PagerStyle BackColor="#666666" ForeColor="White"
            HorizontalAlign="Center" />
         <HeaderStyle BackColor="#1C5E55" Font-Bold="True"
            ForeColor="White" />
         <AlternatingRowStyle BackColor="White" />
      </asp:GridView>
```

Declare the `GridView` control

Identifies the data source that is used to fill the `GridView` with data at runtime

Properties that set various aspects of the `GridView`’s appearance and behavior

Define the columns that appear in the `GridView`

Each `BoundField` represents a column

```
<asp:SqlDataSource ID="messagesSqlDataSource" runat="server"
   ConnectionString="<%$ ConnectionStrings:ConnectionString %>"
   DeleteCommand="DELETE FROM [Messages] WHERE [MessageID] =
      @MessageID" InsertCommand="INSERT INTO [Messages]
      ([Date], [Name], [Email], [Message])
      VALUES (@Date, @Name, @Email, @Message)"
   SelectCommand="SELECT * FROM [Messages]" UpdateCommand=
      "UPDATE [Messages] SET [Date] = @Date, [Name] = @Name,
      [Email] = @Email, [Message] = @Message
      WHERE [MessageID] = @MessageID">
   <DeleteParameters>
      <asp:Parameter Name="MessageID" Type="Int32" />
   </DeleteParameters>
   <UpdateParameters>
      <asp:Parameter Name="Date" Type="String" />
      <asp:Parameter Name="Name" Type="String" />
      <asp:Parameter Name="Email" Type="String" />
      <asp:Parameter Name="Message" Type="String" />
      <asp:Parameter Name="MessageID" Type="Int32" />
   </UpdateParameters>
   <InsertParameters>
      <asp:Parameter Name="Date" Type="String" />
      <asp:ControlParameter ControlID="nameTextBox" Name="Name"
         PropertyName="Text" Type="String" />
      <asp:ControlParameter ControlID="emailTextBox" Name="Email"
         PropertyName="Text" Type="String" />
      <asp:ControlParameter ControlID="messageTextBox"
         Name="Message" PropertyName="Text" Type="String" />
   </InsertParameters>
```

Define a `SqlDataSource` control

Indicates the connection through which the `SqlDataSource` control interacts which the database

Parameters that are set programmatically

Indicates the control from which the parameter gets its value

Specifies the property that contains the actual value to be used as the parameter value

Parameters that obtain their values from controls

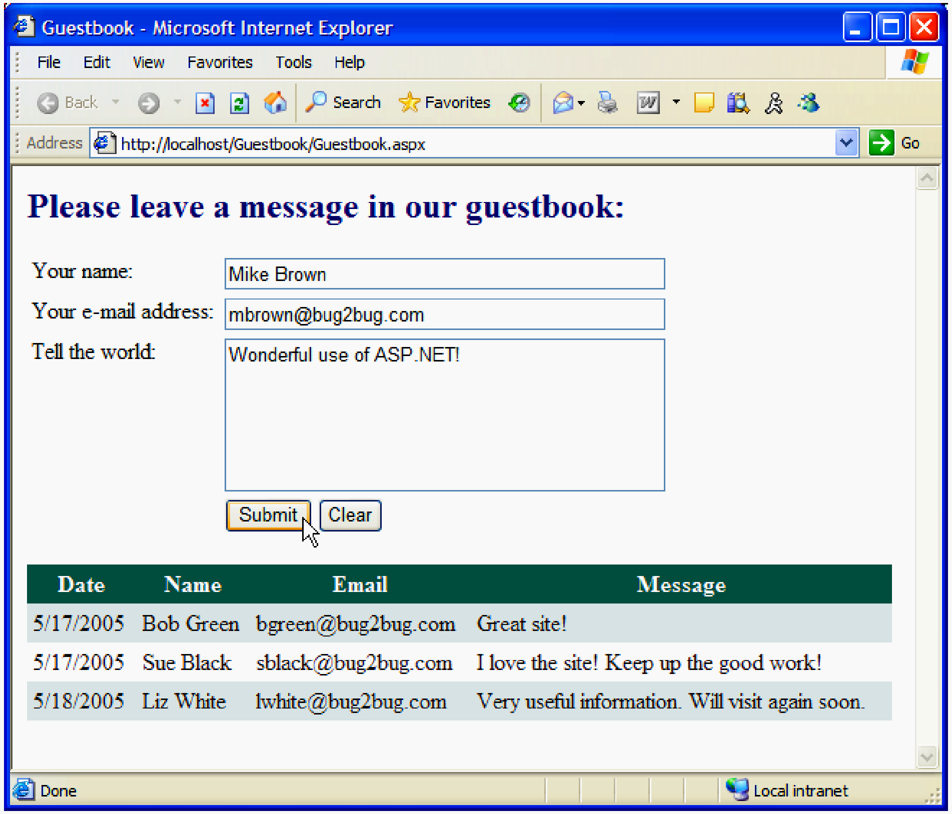
Outline

```
      </asp:SqlDataSource>
   </div>
   </form>
</body>
</html>
```

Guestbook.aspx

(5 of 6)

(a)


Guestbook - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Guestbook/Guestbook.aspx
Go
Submit
Clear
Done
Local intranet

Please leave a message in our guestbook:

Your name: Mike Brown

Your e-mail address: mbrown@bug2bug.com

Tell the world: Wonderful use of ASP.NET!

Date	Name	Email	Message
5/17/2005	Bob Green	bgreen@bug2bug.com	Great site!
5/17/2005	Sue Black	sblack@bug2bug.com	I love the site! Keep up the good work!
5/18/2005	Liz White	lwhite@bug2bug.com	Very useful information. Will visit again soon.

25.5.2 Modifying the Code-Behind File for the Guestbook Application

- SqlDataSource
  - InsertParameters collection
    - Contains an item corresponding to each parameter in the SqlDataSource's INSERT command
  - Insert method
    - Executes the control's INSERT command against the database

- GridView method DataBind
  - Refreshes the information displayed in the GridView

```
' Fig. 25.38: Guestbook.aspx.vb
' Code-behind file that defines event handlers for the guestbook.
Partial Class Guestbook
   Inherits System.Web.UI.Page

   ' Submit Button adds a new guestbook entry to the database,
   ' clears the form and displays the updated list of guestbook entries
   Protected Sub submitButton_Click(ByVal sender As Object, _
      ByVal e As System.EventArgs) Handles submitButton.Click
      ' create a date parameter to store the current date
      Dim currentDate As New System.Web.UI.WebControls.Parameter( _
         "Date", TypeCode.String, DateTime.Now.ToShortDateString())

      ' set the @Date parameter to the date parameter
      messagesSqlDataSource.InsertParameters.RemoveAt(0)
      messagesSqlDataSource.InsertParameters.Add(currentDate)

      ' execute an INSERT SQL statement to add a new row to the
      ' Messages table in the Guestbook database that contains the
      ' current date and the user's name, e-mail address and message
      messagesSqlDataSource.Insert()
```

Guestbook.aspx .vb

Assign a `String` representation of the current data to a new object of type `Parameter`

Remove the `Date` item in the `InsertParameters` collection

Add our `Date` parameter

Executes the `INSERT` command against the database: add a row to the `Messages` table

```
      ' clear the TextBoxes
      nameTextBox.Text = ""
      emailTextBox.Text = ""
      messageTextBox.Text = ""

      ' update the GridView with the new database table contents
      messagesGridView.DataBind()
   End Sub ' submitButton_Click

   ' Clear Button clears the Web Form's TextBoxes
   Protected Sub clearButton_Click(ByVal sender As Object, _
      ByVal e As System.EventArgs) Handles clearButton.Click
      nameTextBox.Text = ""
      emailTextBox.Text = ""
      messageTextBox.Text = ""
   End Sub ' clearButton_Click
End Class ' Guestbook
```

Outline

Guestbook.aspx .vb

(2 of 2)

25.6.1 Examining the Completed Secure Books Database Application

- **Form authentication**
  - **Protect a page so that only users known to the site can access it**
- **Example of visual inheritance:**
  - **Master page defines common GUI elements that are inherited by each page in a set of content pages**

25.8.2 Creating the Secure Books Database Application

- **Step 1: Creating the Web Site**
- **Step 2: Setting Up the Web Site's Folders**
- **Step 3: Configuring the Application's Security Settings (Fig. 25.46-50)**
  - Access rule grants or denies access to a particular Web application directory for a specific user
- **Step 4: Examining the Auto-Generated `Web.config` Files**
  - `authorization` **element indicates who is authorized to access a folder over the Web**
  - `mode` **attribute species that we want to use forms authentication**
  - `deny` **element specifies the users to whom we wish to deny access**
- **Step 5: Creating a Master Page (Fig. 25.51-53)**
  - `Master` **directive specifies that this file defines a master page**
  - `script` **element stores the code that is usually in the code behind file**
  - `ContentPlaceHolder` **control serves as a placeholder for content that will be defined**

25.6.2 Creating the Secure Books Database Application (Cont.)

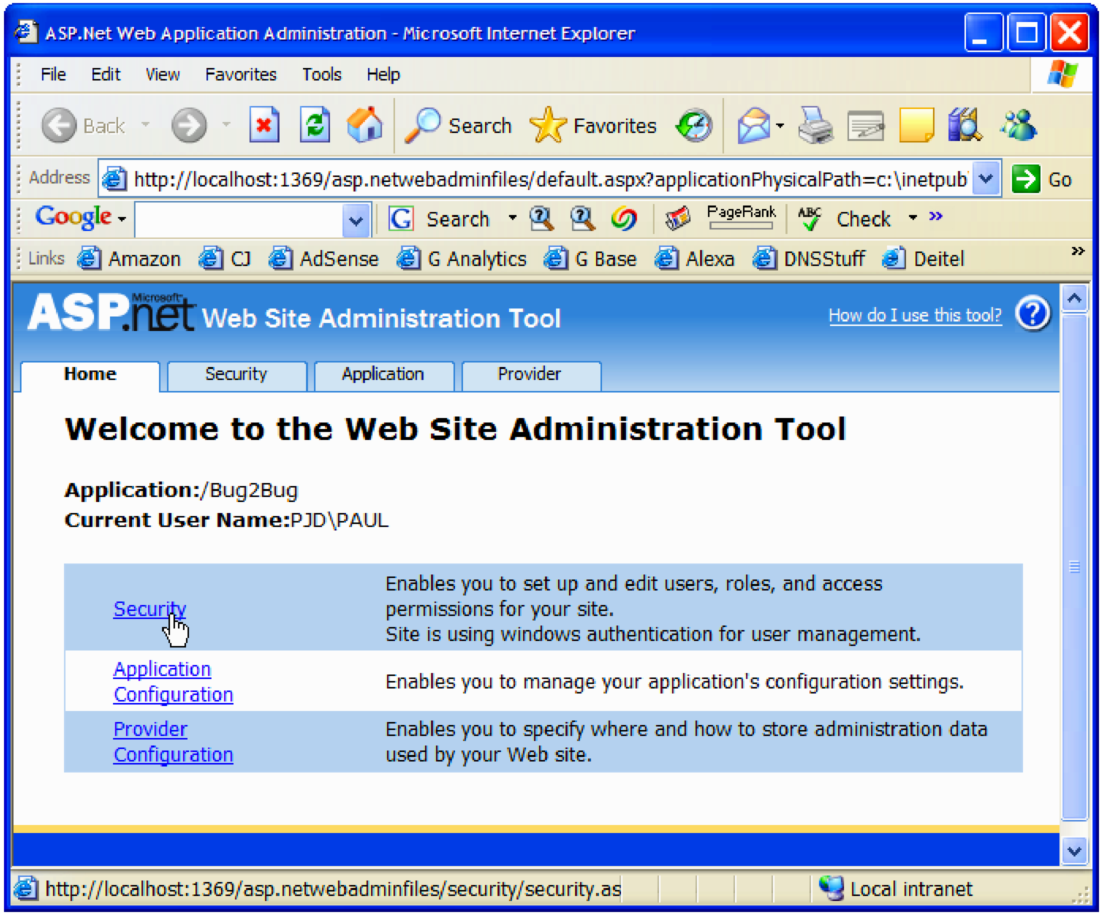
- **Step 6: Creating a Content Page (Fig. 25.54-55)**
  - `Content` **control will place page-specific content that will replace the master page's** `ContentPlaceHolder`
- **Step 7: Adding a** `CreateUserWizard` **Control to a Content Page (Fig. 25.56)**
- **Step 8: Creating a Login Page (Fig. 25.57)**
  - **ASP.NET writes an encrypted cookie containing data about an authenticated user**
  - **Authenticated user is redirected to the page specified by the** `Login` **control's** `DestinationPageUrl` **property**
- **Step 9: Creating a Content Page That Only Authenticated Users Can Access**
- **Step 10: Customizing the Secure Page**
  - `LoginName` **control displays the current authenticated user name**

25.6.2 Creating the Secure Books Database Application (Cont.)

- **Step 11: Connecting the `CreateUserWizard` and `Login` Controls to the Secure Page**
- **Step 12: Generating a `DataSet` Based on the `Books.mdf` Database**
  - `ObjectDataSource` **encapsulates an object that provides access to a data source**
- **Step 13: Creating and Configuring an `AuthorsTableAdapter` (Fig. 25.58)**
- **Step 14: Creating and Configuring a `TitlesTableAdapter` (Fig. 25.59-60)**

25.6.2 Creating the Secure Books Database Application (Cont.)

- **Step 15: Adding a `DropDownList` Containing Authors' First and Last Names (Fig. 25.61-63)**
  - **A business object is an object that accesses data through another object**
  - `DropDownList`**'s** `AutoPostBack` **property indicates that a postback occurs each time the user selects an item**
- **Step 16: Creating a `GridView` to Display the Selected Author's Books (Fig. 25.64-65)**
  - `Enable Sorting` **causes the column headings in the** `GridView` **to turn into hyperlinks that allow users to sort the data**
  - `Enable Paging` **causes the** `GridView` **to split across multiple pages**
  - `PageSize` **property determines the number of entries per page**
- **Step 17: Examining the Markup in `Books.aspx` (Fig. 25.66)**



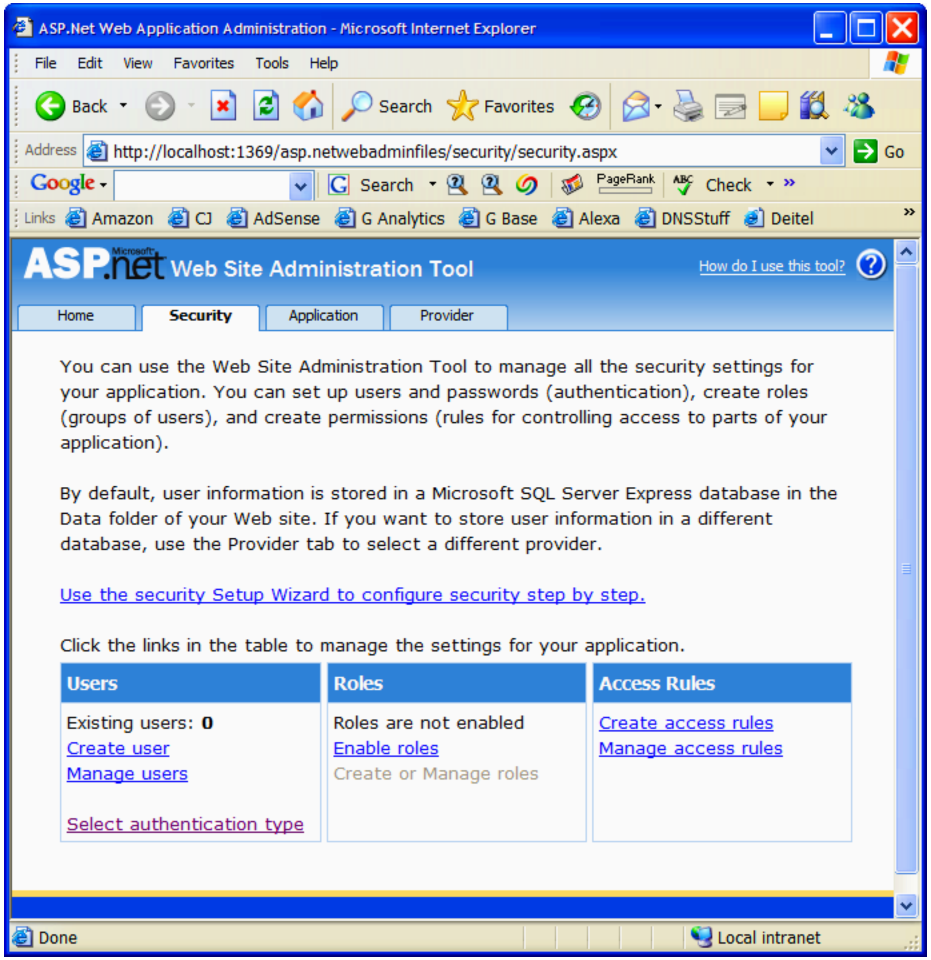
ASP.Net Web Application Administration - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Address
http://localhost:1369/asp.netwebadminfiles/default.aspx?applicationPhysicalPath=c:\inetpub
Go
Google
Search
PageRank
Check
Links
Amazon
CJ
AdSense
G Analytics
G Base
Alexa
DNSStuff
Deitel
ASP.net Web Site Administration Tool
How do I use this tool?
Home
Security
Application
Provider
http://localhost:1369/asp.netwebadminfiles/security/security.as
Local intranet

Welcome to the Web Site Administration Tool

Application:/Bug2Bug
Current User Name:PJD\PAUL

Security	Enables you to set up and edit users, roles, and access permissions for your site. Site is using windows authentication for user management.
Application Configuration	Enables you to manage your application's configuration settings.
Provider Configuration	Enables you to specify where and how to store administration data used by your Web site.

Fig. 25.46 | **Web Site Administration Tool** for configuring a web application.


ASP.Net Web Application Administration - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:1369/asp.netwebadminfiles/security/security.aspx Go
Google Search PageRank Check
Links Amazon CJ AdSense G Analytics G Base Alexa DNSStuff Deitel
ASP.net Web Site Administration Tool
How do I use this tool?
Home Security Application Provider
Done
Local intranet

You can use the Web Site Administration Tool to manage all the security settings for your application. You can set up users and passwords (authentication), create roles (groups of users), and create permissions (rules for controlling access to parts of your application).

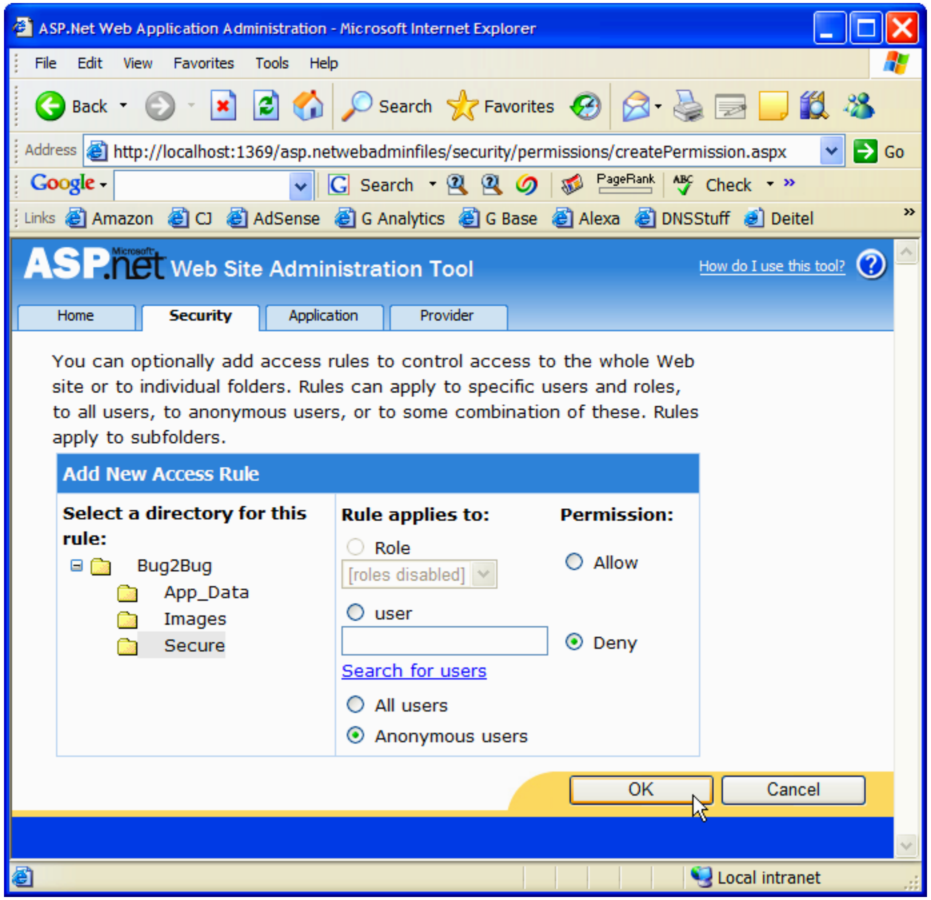
By default, user information is stored in a Microsoft SQL Server Express database in the Data folder of your Web site. If you want to store user information in a different database, use the Provider tab to select a different provider.

Use the security Setup Wizard to configure security step by step.

Click the links in the table to manage the settings for your application.

Users	Roles	Access Rules
Existing users: **0** Create user Manage users Select authentication type	Roles are not enabled Enable roles Create or Manage roles	Create access rules Manage access rules

Fig. 25.49 | Main page of the **Web Site Administration Tool** after enabling forms authentication.



ASP.Net Web Application Administration - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:1369/asp.netwebadminfiles/security/permissions/createPermission.aspx Go
Google Search PageRank Check
Links Amazon CJ AdSense G Analytics G Base Alexa DNSStuff Deitel
ASP.net Web Site Administration Tool
How do I use this tool?
Home
Security
Application
Provider

You can optionally add access rules to control access to the whole Web site or to individual folders. Rules can apply to specific users and roles, to all users, to anonymous users, or to some combination of these. Rules apply to subfolders.

Add New Access Rule

Select a directory for this rule:

- Bug2Bug
  - App_Data
  - Images
  - Secure

Rule applies to:

○ Role [roles disabled]

○ user

Search for users

○ All users

● Anonymous users

Permission:

○ Allow

● Deny

Fig. 25.50 | **Add New Access Rule** page used to configure directory access.

Outline

Bug2bug.master

(1 of 2)

```
<%-- Fig. 25.53: Bug2bug.master --%>
<%-- Master page that defines common features of all pages in the --%>
<%-- secure book database application. --%>
<%@ Master Language="VB" %>

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

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Bug2Bug</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
       <table border="0" cellpadding="0" cellspacing="0"
          style="width: 100%; height: 100%">
          <tr>
             <td height="130" style="width: 887px">
                <asp:Image ID="headerImage" runat="server"
                   ImageUrl="~/Images/bug2bug.png" />
             </td>
          </tr>
```

Specify the language is VB

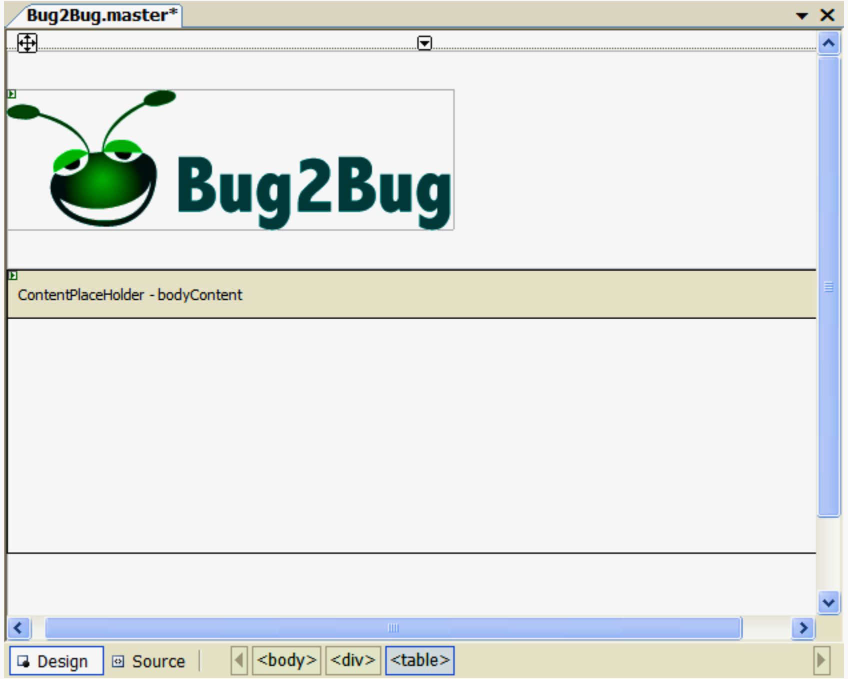
Display logo picture

```
          <tr>
             <td style="width: 887px" valign="top">
                <asp:contentplaceholder id="bodyContent" runat="server">
                </asp:contentplaceholder>
             </td>
          </tr>
       </table>
       &nbsp;
    </div>
    </form>
</body>
</html>
```

Define a `ContentPlaceHolder` control

Bug2bug.master

(2 of 2)



```
<%-- Fig. 25.56: CreateNewUser.aspx --%>
<%-- Content page using a CreateUserWizard control to register
<%@ Page Language="VB" MasterPageFile="~/Bug2Bug.master"
   Title="Create a New User" %>
<asp:Content ID="Content1" ContentPlaceHolderID="bodyContent"
   Runat="Server">
   <asp:CreateUserWizard ID="CreateUserWizard1" runat="server"
      BackColor="#F7F6F3" BorderColor="#E6E2D8" BorderStyle="Solid"
      BorderWidth="1px" Font-Names="Verdana" Font-Size="0.8em"
      ContinueDestinationPageUrl="~/Secure/Books.aspx">
      <WizardSteps>
         <asp:CreateUserWizardStep runat="server">
         </asp:CreateUserWizardStep>
         <asp:CompleteWizardStep runat="server">
         </asp:CompleteWizardStep>
      </WizardSteps>
      <SideBarStyle BackColor="#5D7B9D" BorderWidth="0px"
         Font-Size="0.9em" VerticalAlign="Top" />
      <TitleTextStyle BackColor="#5D7B9D" Font-Bold="True"
         ForeColor="White" />
```

.aspx

Specify language, master page, and title

Replaces the master page's `ContentPlaceHolder`

Contains several properties that specify formatting styles for the control

Encapsulate the details of creating a user and issuing a confirmation message

Elements that define additional styles used to format specific parts of the control

```
      <SideBarButtonStyle BorderWidth="0px" Font-Names="Verdana"
         ForeColor="White" />
      <NavigationButtonStyle BackColor="#FFFBFF" BorderColor="#CCCCCC"
         BorderStyle="Solid" BorderWidth="1px" Font-Names="Verdana"
         ForeColor="#284775" />
      <HeaderStyle BackColor="#5D7B9D" BorderStyle="Solid"
         Font-Bold="True" Font-Size="0.9em"
         ForeColor="White" HorizontalAlign="Center" />
      <CreateUserButtonStyle BackColor="#FFFBFF" BorderColor="#CCCCCC"
        BorderStyle="Solid" BorderWidth="1px" Font-Names="Verdana"
        ForeColor="#284775" />
      <ContinueButtonStyle BackColor="#FFFBFF" BorderColor="#CCCCCC"
         BorderStyle="Solid" BorderWidth="1px" Font-Names="Verdana"
         ForeColor="#284775" />
      <StepStyle BorderWidth="0px" />
   </asp:CreateUserWizard>
</asp:Content>
```

CreateNewUser .aspx

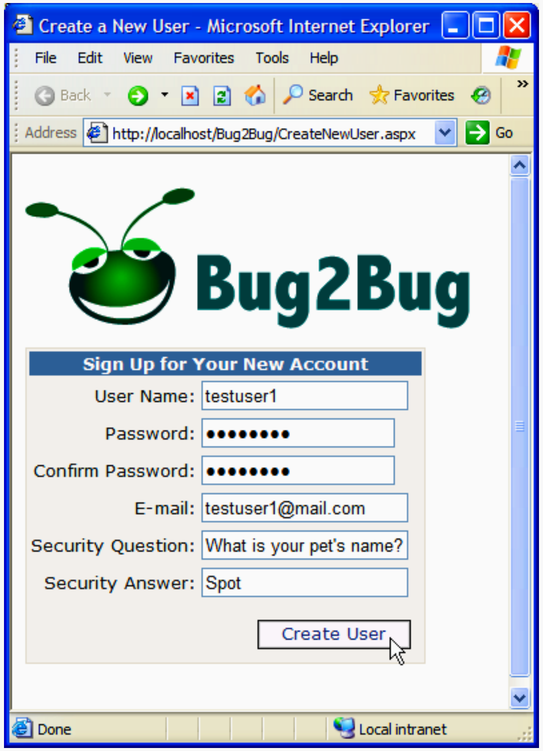
(2 of 3)

Outline

`CreateNewUser.aspx`

(3 of 3)

(a)



Create a New User - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Bug2Bug/CreateNewUser.aspx Go
Bug2Bug
Create User
Done
Local intranet

Sign Up for Your New Account

User Name: testuser1

Password: ●●●●●●●●

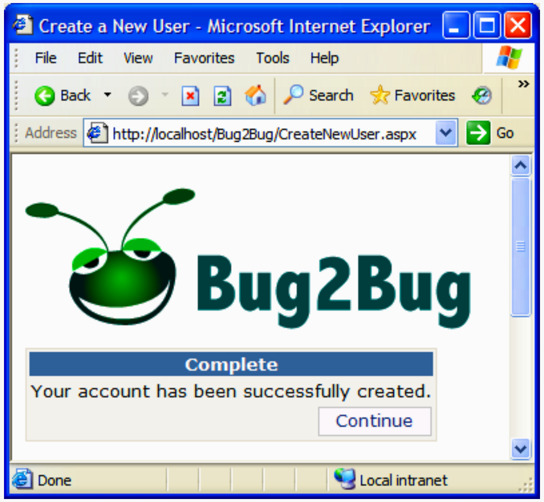
Confirm Password: ●●●●●●●●

E-mail: testuser1@mail.com

Security Question: What is your pet's name?

Security Answer: Spot

(b)

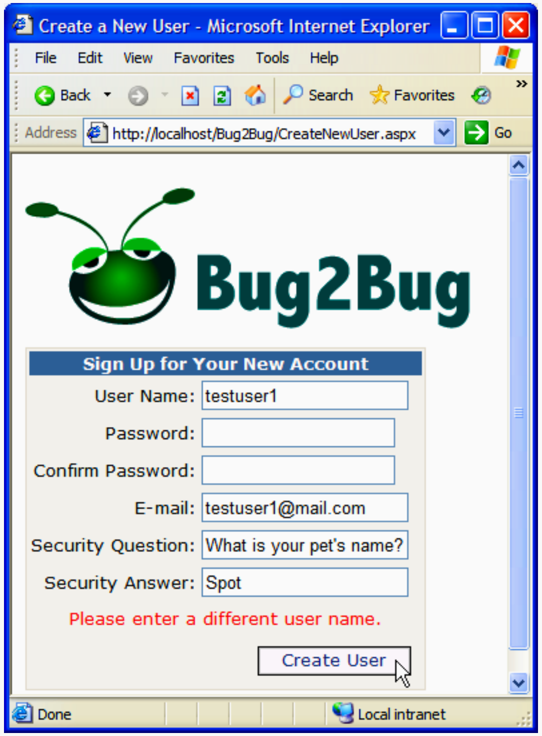


Create a New User - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Bug2Bug/CreateNewUser.aspx Go
Bug2Bug
Continue
Done
Local intranet

Complete

Your account has been successfully created.

(c)



Create a New User - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Bug2Bug/CreateNewUser.aspx Go
Bug2Bug
Create User
Done
Local intranet

Sign Up for Your New Account

User Name: testuser1

Password:

Confirm Password:

E-mail: testuser1@mail.com

Security Question: What is your pet's name?

Security Answer: Spot

Please enter a different user name.

```
<%-- Fig. 25.57: Login.aspx --%>
<%-- Content page using a Login control that authenticates users. --%>
<%@ Page Language="VB" MasterPageFile="~/Bug2Bug.master" Title="Login" %>
<asp:Content ID="Content1" ContentPlaceHolderID="bodyContent"
   Runat="Server">
   <asp:Login ID="Login1" runat="server" BackColor="#F7F6F3"
      BorderColor="#E6E2D8" BorderPadding="4" BorderStyle="Solid"
      BorderWidth="1px" CreateUserText="Click here to create a new user"
      CreateUserUrl="~/CreateNewUser.aspx" DisplayRememberMe="False"
      Font-Names="Verdana" Font-Size="0.8em" ForeColor="#333333"
      DestinationPageUrl="~/Secure/Books.aspx">
      <TitleTextStyle BackColor="#5D7B9D" Font-Bold="True"
         Font-Size="0.9em" ForeColor="White" />
      <InstructionTextStyle Font-Italic="True" ForeColor="Black" />
      <TextBoxStyle Font-Size="0.8em" />
      <LoginButtonStyle BackColor="#FFFBFF" BorderColor="#CCCCCC"
         BorderStyle="Solid" BorderWidth="1px" Font-Names="Verdana"
         Font-Size="0.8em" ForeColor="#284775" />
   </asp:Login>
</asp:Content>
```

Specify language, master page, and title

Replaces the master page's `ContentPlaceHolder`

Create a `Login` control

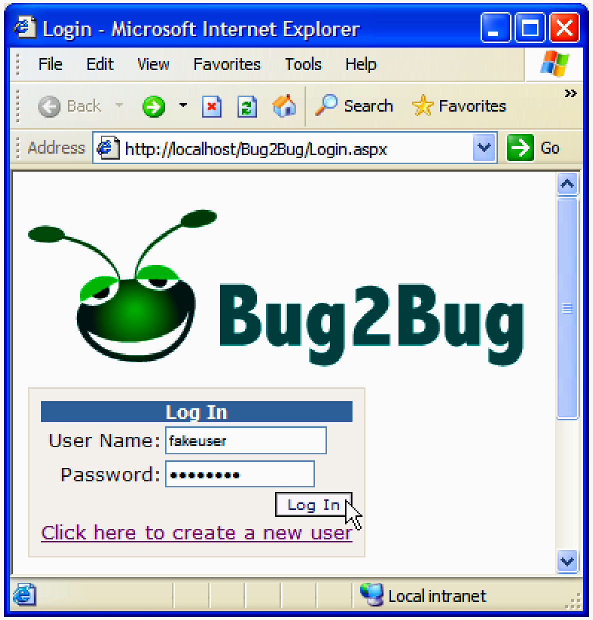
The elements define various formatting style applied to parts of the control

Outline

`Login.aspx`

(1 of 2)

(a)


Login - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Bug2Bug
Log In
Local intranet

http://localhost/Bug2Bug/Login.aspx

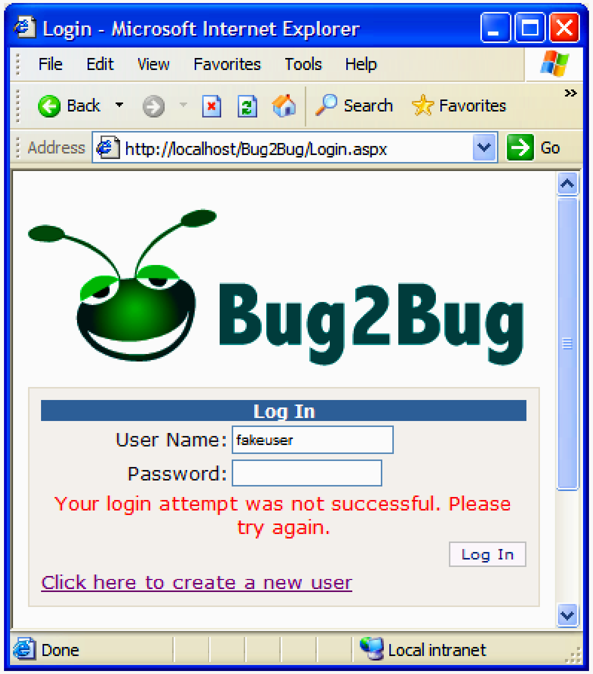
Log In

User Name: fakeuser

Password: ••••••••

Click here to create a new user

(b)


Login - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Bug2Bug
Log In
Done
Local intranet

http://localhost/Bug2Bug/Login.aspx

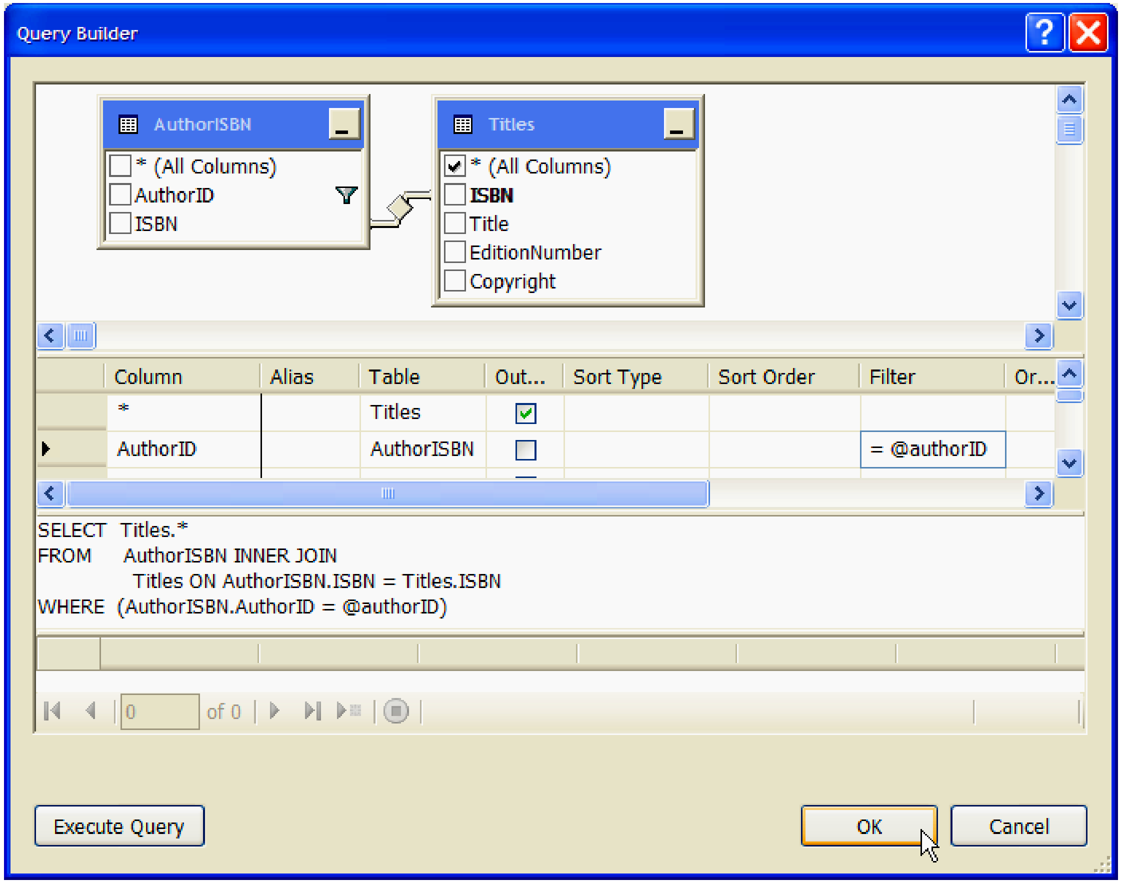
Log In

User Name: fakeuser

Password:

Your login attempt was not successful. Please try again.

Click here to create a new user



AuthorISBN

- □ * (All Columns)
- □ AuthorID
- □ ISBN

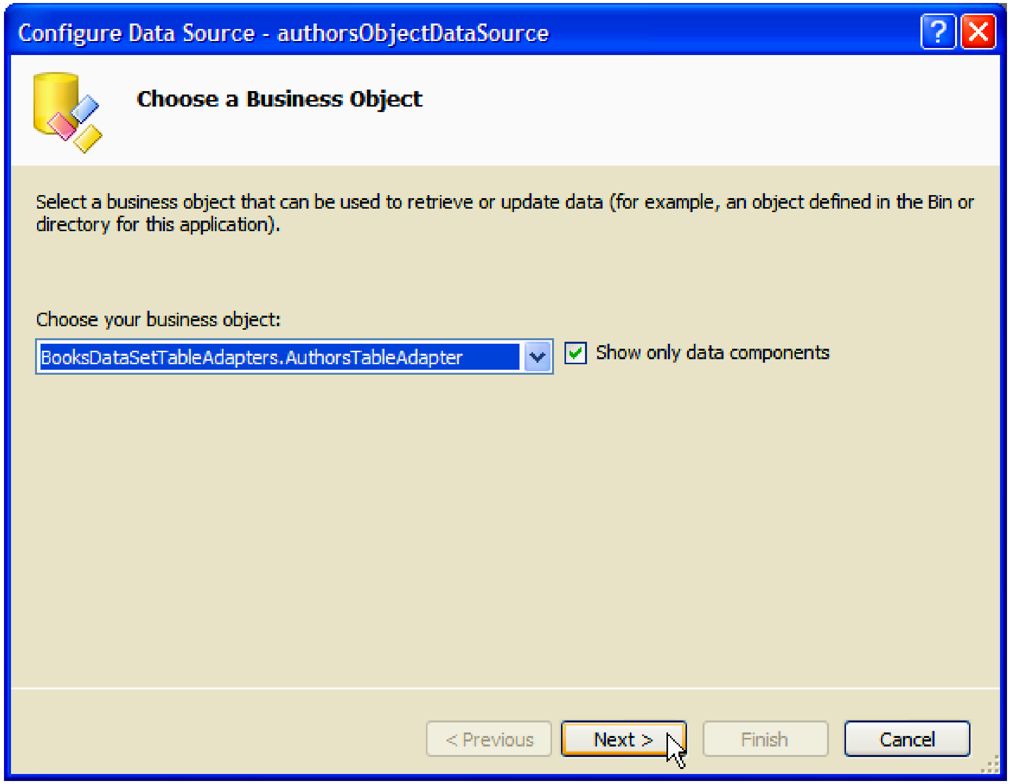
Titles

- ☑ * (All Columns)
- □ **ISBN**
- □ Title
- □ EditionNumber
- □ Copyright

Column	Alias	Table	Out...	Sort Type	Sort Order	Filter	Or...
*		Titles	☑				
AuthorID		AuthorISBN	□			= @authorID	

```
SELECT  Titles.*
FROM     AuthorISBN INNER JOIN
          Titles ON AuthorISBN.ISBN = Titles.ISBN
WHERE  (AuthorISBN.AuthorID = @authorID)
```

Fig. 25.59 | Query Builder for designing a query that selects books written by a particular author.



Configure Data Source - authorsObjectDataSource

Choose a Business Object

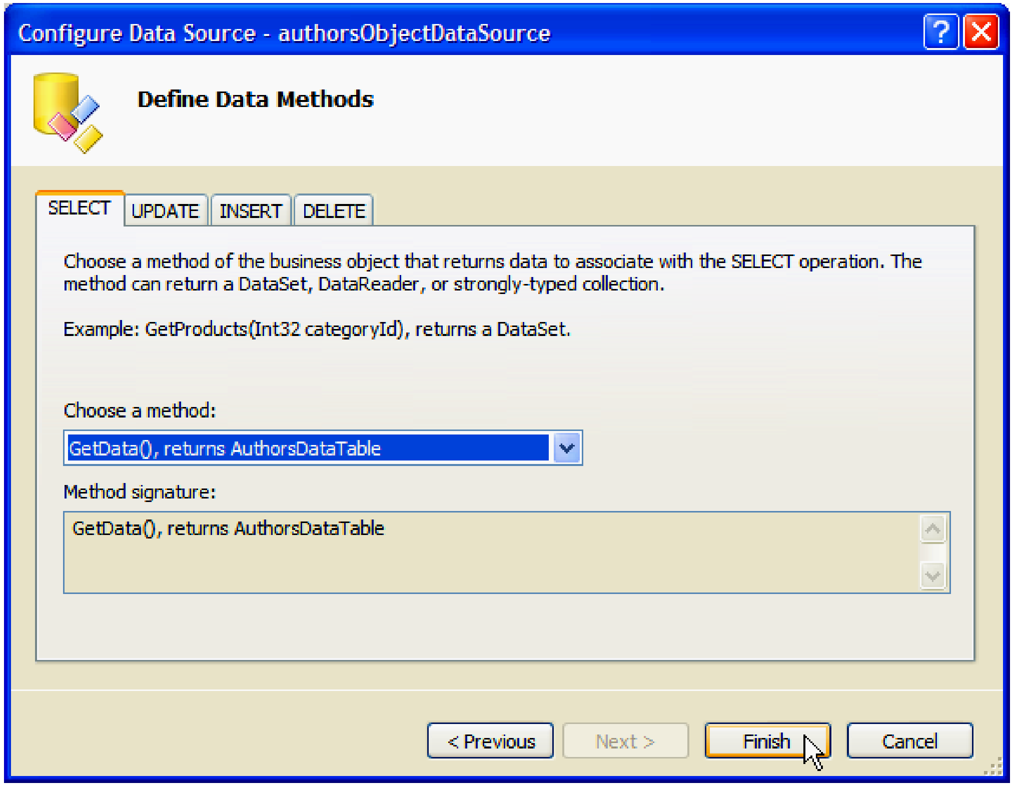
Select a business object that can be used to retrieve or update data (for example, an object defined in the Bin or directory for this application).

Choose your business object:

BooksDataSetTableAdapters.AuthorsTableAdapter

☑ Show only data components

Fig. 25.61 | Choosing a business object for an `ObjectDataSource`.



Define Data Methods

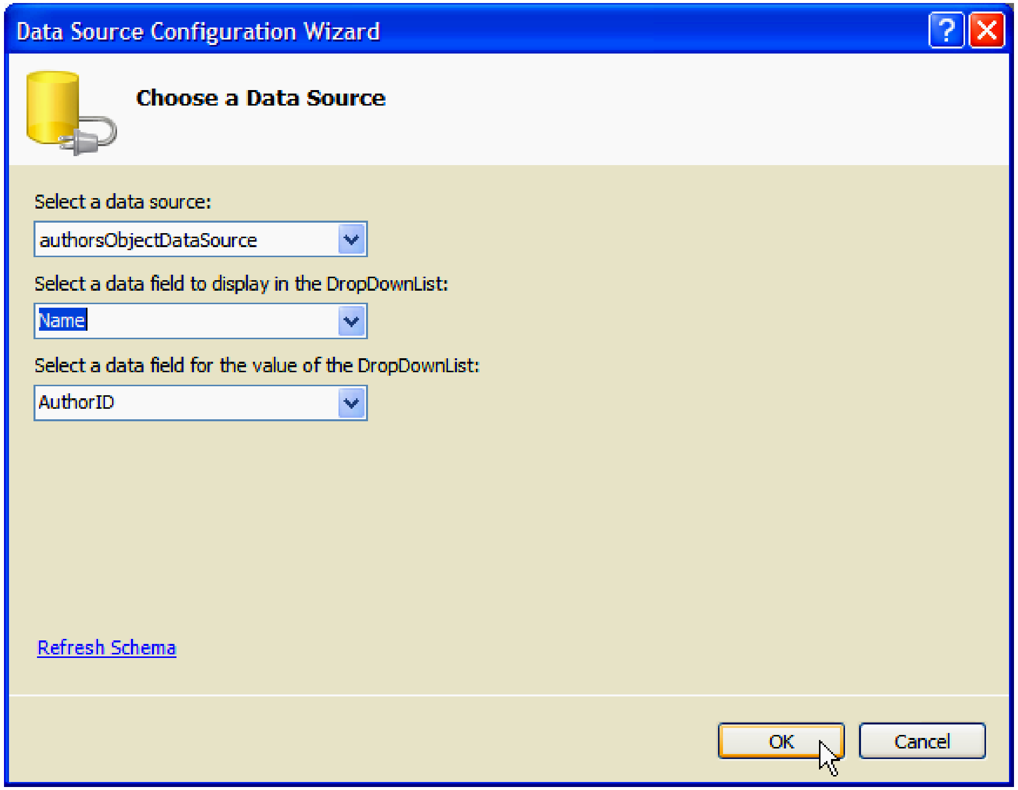
Choose a method of the business object that returns data to associate with the SELECT operation. The method can return a DataSet, DataReader, or strongly-typed collection.

Example: GetProducts(Int32 categoryId), returns a DataSet.

Choose a method: GetData(), returns AuthorsDataTable

Method signature: GetData(), returns AuthorsDataTable

Fig. 25.62 | Choosing a data method of a business object for use with an `ObjectDataSource`.

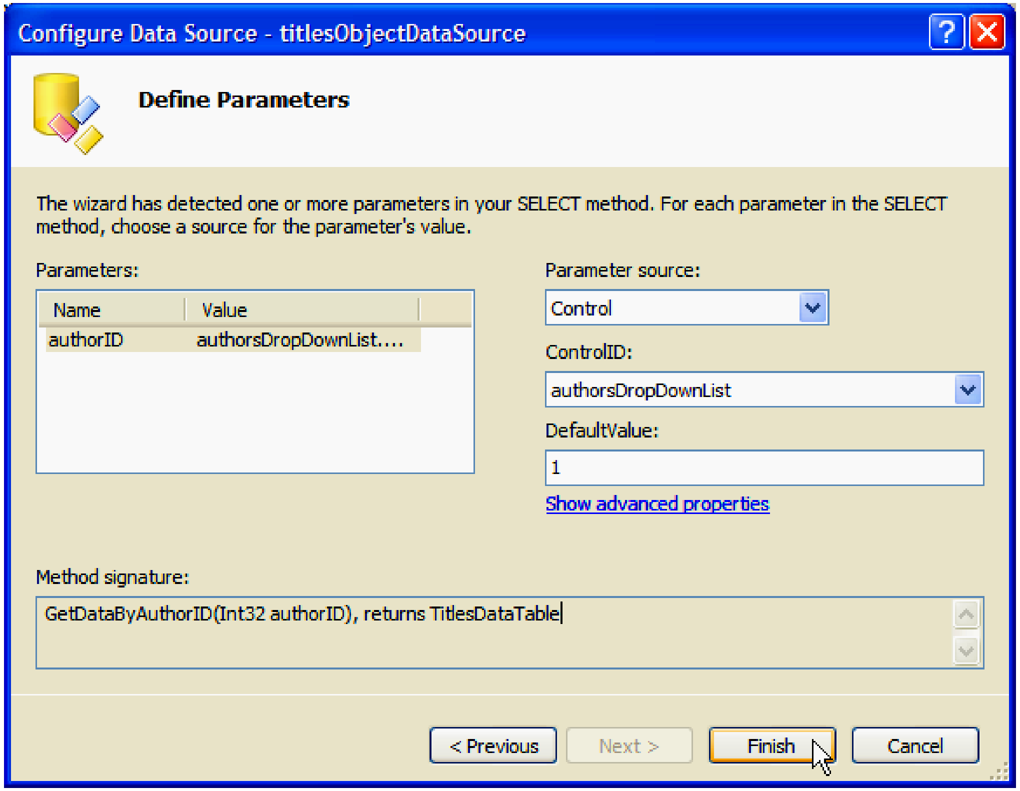


Select a data source:
authorsObjectDataSource

Select a data field to display in the DropDownList:
Name

Select a data field for the value of the DropDownList:
AuthorID

Fig. 25.63 | Choosing a data source for a **DropDownList**.



Configure Data Source - titlesObjectDataSource

Define Parameters

The wizard has detected one or more parameters in your SELECT method. For each parameter in the SELECT method, choose a source for the parameter's value.

Parameters:

Name	Value
authorID	authorsDropDownList....

Parameter source: Control

ControlID: authorsDropDownList

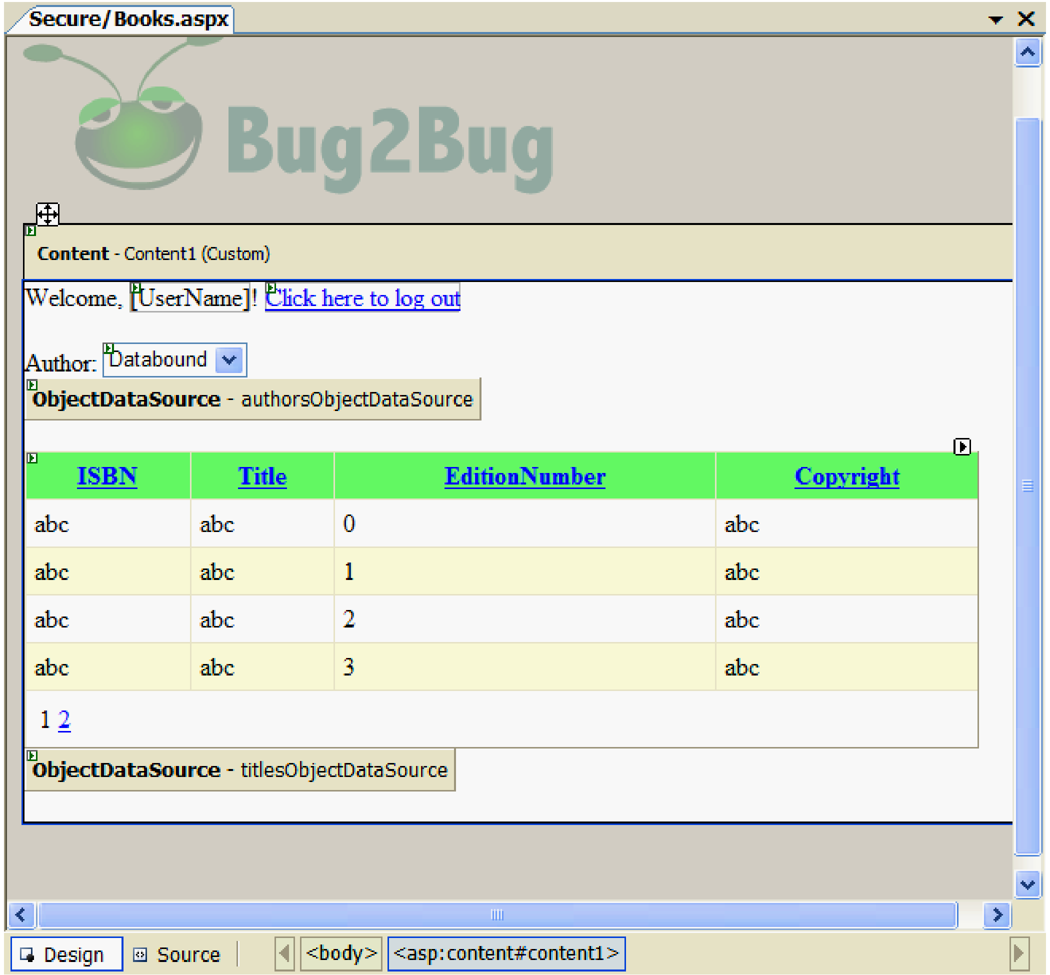
DefaultValue: 1

Show advanced properties

Method signature:

```
GetDataByAuthorID(Int32 authorID), returns TitlesDataTable
```

Fig. 25.64 | Choosing the data source for a parameter in a business object’s data method.



Content - Content1 (Custom)

Welcome, [UserName]! Click here to log out

Author: Databound

ObjectDataSource - authorsObjectDataSource

ISBN	Title	EditionNumber	Copyright
abc	abc	0	abc
abc	abc	1	abc
abc	abc	2	abc
abc	abc	3	abc

1 2

ObjectDataSource - titlesObjectDataSource

Fig. 25.65 | Completed **`Books.aspx`** in **Design** mode.

```
<%-- Fig. 25.66: Books.aspx --%>
<%-- Displays information from the Books database. --%>
<%@ Page Language="VB" MasterPageFile="~/Bug2Bug.master"
   Title="Book Information" %>
<asp:Content ID="Content1" ContentPlaceHolderID="bodyContent"
   Runat="Server">
   Welcome,
   <asp:LoginName ID="LoginName1" runat="server" />!
   <asp:LoginStatus ID="LoginStatus1" runat="server"
      LogoutAction="RedirectToLoginPage"
      LogoutText="Click here to log out" />
   <br />
   <br />
   Author:
   <asp:DropDownList ID="authorsDropDownList" runat="server"
      AutoPostBack="True" DataSourceID="authorsObjectDataSource"
      DataTextField="Name" DataValueField="AuthorID">
   </asp:DropDownList>
   <asp:ObjectDataSource ID="authorsObjectDataSource"
      runat="server" OldValuesParameterFormatString="original_{0}"
      SelectMethod="GetData"
      TypeName="BooksDataSetTableAdapters.AuthorsTableAdapter">
   </asp:ObjectDataSource>
   <br />
   <br />
```

Create a `LoginName` control; displays the authenticated user’s name when the page is requested and viewed in a browser

(1 of 4)

Create a `LoginStatus` control; redirect user to login page after logging out

Indicates that changing the selected item in the list causes a postback to occur

Specifies that this `ObjectDataSource` accesses the `Books` database by calling `GetData` of `BooksDataSet`’s `AuthorsTableAdapter`

Outline

```
   <asp:GridView ID="titlesGridView" runat="server" AllowPaging="True"
      AllowSorting="True" AutoGenerateColumns="False" CellPadding="5"
      DataKeyNames="ISBN" DataSourceID="titlesObjectDataSource"
      PageSize="4" Width="600px">
      <Columns>
         <asp:BoundField DataField="ISBN" HeaderText="ISBN"
            ReadOnly="True" SortExpression="ISBN" />
         <asp:BoundField DataField="Title" HeaderText="Title"
            SortExpression="Title" />
         <asp:BoundField DataField="EditionNumber"
            HeaderText="EditionNumber" SortExpression="EditionNumber" />
         <asp:BoundField DataField="Copyright" HeaderText="Copyright"
            SortExpression="Copyright" />
      </Columns>
      <HeaderStyle BackColor="LightGreen" />
      <AlternatingRowStyle BackColor="LightYellow" />
   </asp:GridView>
   <asp:ObjectDataSource ID="titlesObjectDataSource" runat="server"
      OldValuesParameterFormatString="original_{0}"
      SelectMethod="GetDataByAuthorID"
      TypeName="BooksDataSetTableAdapters.TitlesTableAdapter">
      <SelectParameters>
         <asp:ControlParameter ControlID="authorsDropDownList"
            DefaultValue="1" Name="authorID"
            PropertyName="SelectedValue" Type="Int32" />
      </SelectParameters>
   </asp:ObjectDataSource>
</asp:Content>
```

Create the `GridView` that displays information about the books written by the selected author

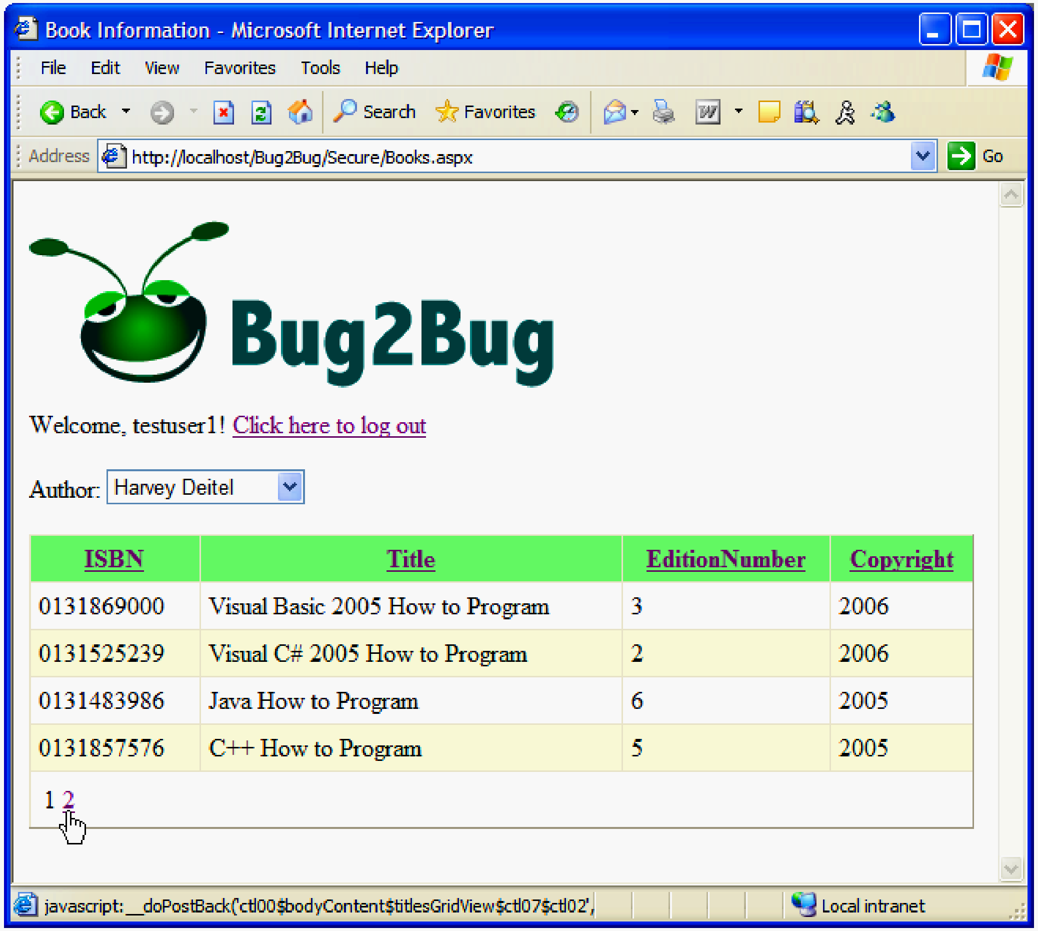
Define the `ObjectDataSource` used to fill the `GridView` with data

Specifies that the value of method `GetDataByAuthorID`'s parameter comes from the `SelectedValue` property of `authorsDropDownList`

Books.aspx

(3 of 4)

(a)


Book Information - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Address
http://localhost/Bug2Bug/Secure/Books.aspx
Go
Bug2Bug
Local intranet

Welcome, testuser1! Click here to log out

Author: Harvey Deitel

ISBN	Title	EditionNumber	Copyright
0131869000	Visual Basic 2005 How to Program	3	2006
0131525239	Visual C# 2005 How to Program	2	2006
0131483986	Java How to Program	6	2005
0131857576	C++ How to Program	5	2005

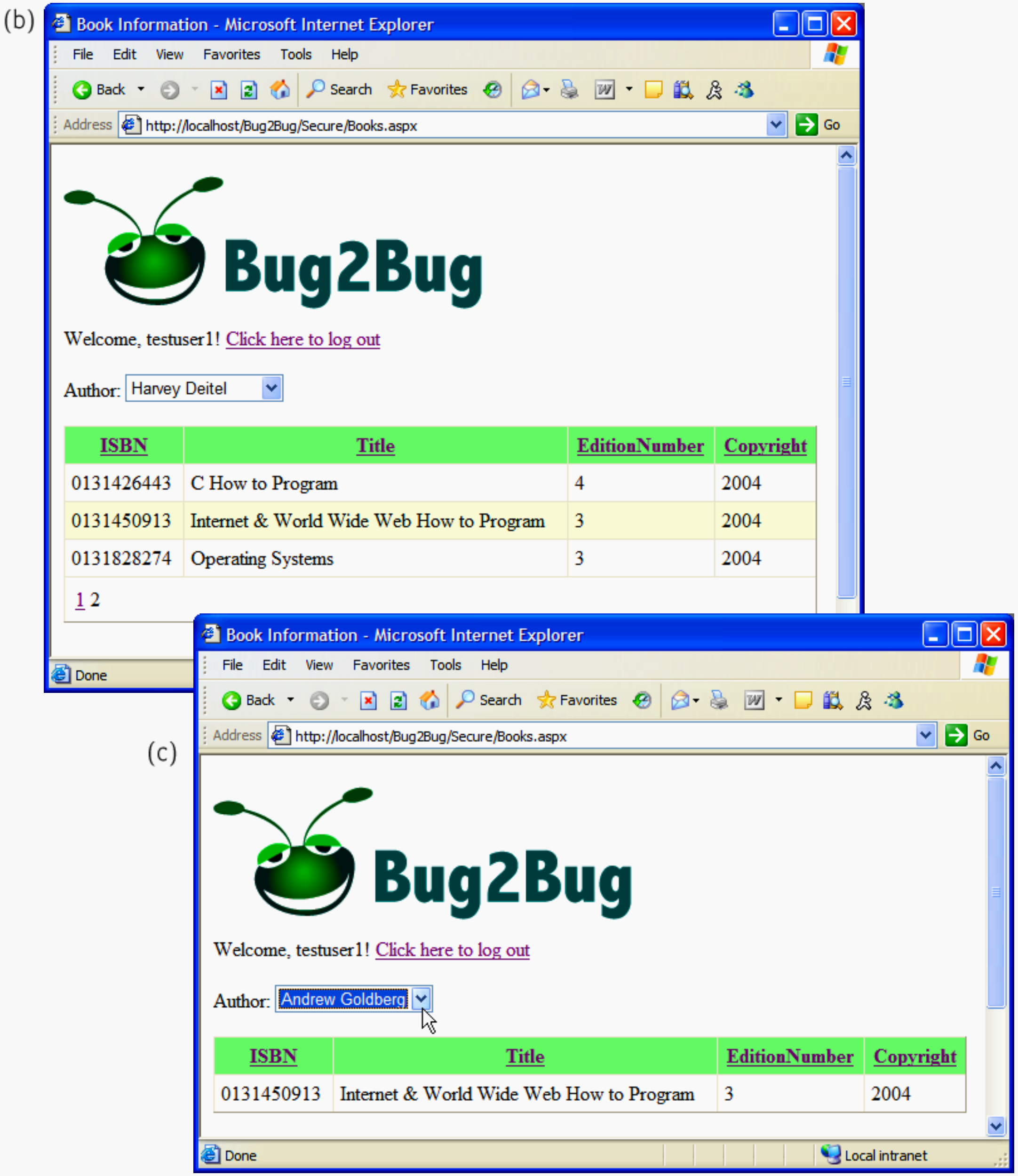
1 2

javascript:__doPostBack('ctl00$bodyContent$titlesGridView$ctl07$ctl02',

Outline

`Books.aspx`

(4 of 4)


(b)
Book Information - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go
Bug2Bug
Done
(c)
Book Information - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go
Bug2Bug
Done
Local intranet

(b)

http://localhost/Bug2Bug/Secure/Books.aspx

Welcome, testuser1! Click here to log out

Author: Harvey Deitel

ISBN	Title	EditionNumber	Copyright
0131426443	C How to Program	4	2004
0131450913	Internet & World Wide Web How to Program	3	2004
0131828274	Operating Systems	3	2004

1 2

(c)

http://localhost/Bug2Bug/Secure/Books.aspx

Welcome, testuser1! Click here to log out

Author: Andrew Goldberg

ISBN	Title	EditionNumber	Copyright
0131450913	Internet & World Wide Web How to Program	3	2004

25.7 ASP.NET Ajax

- **Quickly and easily add Ajax functionality to existing ASP.NET web applications**
- **www.asp.net/ajax/downloads**
- **The Ajax Extensions package implements basic Ajax functionality**
- **ASP.NET Ajax Control Toolkit contains rich, Ajax-enabled GUI controls**

25.7 ASP.NET Ajax (cont.)

- Add the ASP.NET Ajax Control Toolkit controls to the **Toolbox** in Visual Web Developer (or Visual Studio)
  - Right click the **Toolbox** and choose **Add Tab**
  - Type **Ajax Toolkit** in the new tab
  - Right click the tab and select **Choose Items**
  - Navigate to the folder in which you extracted the Ajax Control Toolkit and select **AjaxControlToolkit.dll** from the **SampleWebSite\Bin** folder
  - A list of available Ajax controls will appear under the **Ajax Toolkit** tab when you are in **Design** mode

25.7 ASP.NET Ajax (cont.)

- **Application was not initially set up to support Ajax functionality, so we must first modify the `web.config` file**
  - **Select File > New Website… to display the New Website dialog**
  - **Create an empty ASP.NET Ajax-Enabled Website**
  - **Open the `web.config` file in this new application and copy its contents**
  - **Open the Validation application and replace the contents of its `web.config` file with the contents of the `web.config` file you just copied**
  - **The new `web.config` file adds the `system.web.extensions`, `httpHandlers` and `httpModules` sections, which specify the settings for running scripts that enable Ajax functionality**
  - To learn more about the details of these web.config modifications, visit

www.asp.net/ajax/documentation/live/configuringASPNETAJAX.aspx

```
<%-- Fig. 25.67: Validation.aspx --%>
<%-- Validation application enhanced by ASP.NET Ajax. --%>
<%@ Page Language="VB" AutoEventWireup="false"
   CodeFile="Validation.aspx.vb" Inherits="Validation" %>
<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit"
   TagPrefix="ajax" %>

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

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Demonstrating Validation Controls</title>
</head>
<body>
   <form id="form1" runat="server">
   <div>
      <ajax:ToolkitScriptManager ID="ToolkitScriptManager1"
         runat="server">
      </ajax:ToolkitScriptManager>
      &nbsp;Please fill out the following form.<br /><em>All fields are
      required and must contain valid information.</em><br /><br />
      <table>
         <tr>
            <td style="width: 100px" valign="top">
               Name:</td>
```

Validation.aspx

(1 of 5)

Make Ajax toolkit available to the application

Bundles the scripts for the toolkits controls

Outline

Validation.aspx

(2 of 5)

```
               <td style="width: 450px" valign="top">
                  <asp:TextBox ID="nameTextBox" runat="server">
                  </asp:TextBox>
                  <br />
                  <asp:RequiredFieldValidator
                     ID="nameInputValidator" runat="server"
                     ControlToValidate="nameTextBox" Display="None"
                     ErrorMessage="Please enter your name.">
                  </asp:RequiredFieldValidator>
                  <ajax:ValidatorCalloutExtender ID="nameInputCallout"
                     runat="server" TargetControlID="nameInputValidator"/>
               </td>
            </tr>
            <tr>
               <td style="width: 100px" valign="top">E-mail ad
               <td style="width: 450px" valign="top">
                  <asp:TextBox ID="emailTextBox" runat="server">
                  </asp:TextBox>
                  &nbsp;e.g., user@domain.com<br />
                  <asp:RequiredFieldValidator
                     ID="emailInputValidator" runat="server"
                     ControlToValidate="emailTextBox" Display="None"
                     ErrorMessage="Please enter your e-mail address.">
                  </asp:RequiredFieldValidator>
                  <ajax:ValidatorCalloutExtender ID="emailInputCallout"
                     runat="server" TargetControlID="emailInputValidator"/>
```

Each of these displays error messages in yellow callouts next to the input field when necessary

Outline

```
      <asp:RegularExpressionValidator
         ID="emailFormatValidator" runat="server"
         ControlToValidate="emailTextBox" Display="None"
         ErrorMessage=
            "Please enter an e-mail address in a valid format."
         ValidationExpression=
            "\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*">
      </asp:RegularExpressionValidator>
      <ajax:ValidatorCalloutExtender ID="emailFormatCallout"
         runat="server"
         TargetControlID="emailFormatValidator"/>
   </td>
</tr>
<tr>
   <td style="width: 100px; height: 21px" valign="top">
      Phone number:</td>
   <td style="width: 450px; height: 21px" valign="top">
      <asp:TextBox ID="phoneTextBox" runat="server">
      </asp:TextBox>
       &nbsp;e.g., (555) 555-1234<br />
      <asp:RequiredFieldValidator
         ID="phoneInputValidator" runat="server"
         ControlToValidate="phoneTextBox" Display="None"
         ErrorMessage="Please enter your phone number.">
      </asp:RequiredFieldValidator>
      <ajax:ValidatorCalloutExtender ID="phoneInputCallout"
         runat="server" TargetControlID="phoneInputValidator"/>
```

Validation.aspx

(3 of 5)

```
               <asp:RegularExpressionValidator
                  ID="phoneFormatValidator" runat="server"
                  ControlToValidate="phoneTextBox" Display="None"
                  ErrorMessage=
                     "Please enter a phone number in a valid format."
                  ValidationExpression=
                     "((\(\d{3}\) ?)|(\d{3}-))?\d{3}-\d{4}">
               </asp:RegularExpressionValidator>
               <ajax:ValidatorCalloutExtender ID="PhoneFormatCallout"
                  runat="server"
                  TargetControlID="phoneFormatValidator"/>
            </td>
         </tr>
      </table>
      <asp:UpdatePanel ID="UpdatePanel1" runat="server">
         <ContentTemplate>
            <asp:Button ID="submitButton" runat="server" Text="Submit" />
            <br /><br /><br />&nbsp;
            <asp:Label ID="outputLabel" runat="server"
               Text="Thank you for your submission." Visible="False">
            </asp:Label>
         </ContentTemplate>
      </asp:UpdatePanel>
   </div>
   </form>
</body>
</html>
```

Eliminates full page refreshes by enabling partial page refreshes

Validation.aspx

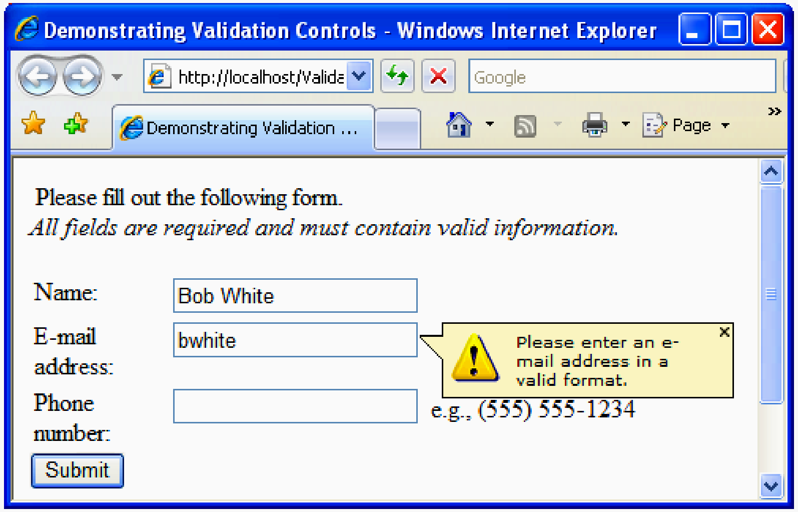
(4 of 5)

Outline

Validation.aspx

(5 of 5)

a) The user enters an e-mail address in an incorrect format and presses *Tab* to move to the next input field. A callout appears informing the user to enter an e-mail address in a valid format.


Demonstrating Validation Controls - Windows Internet Explorer
http://localhost/Valida
Google
Demonstrating Validation ...
Page

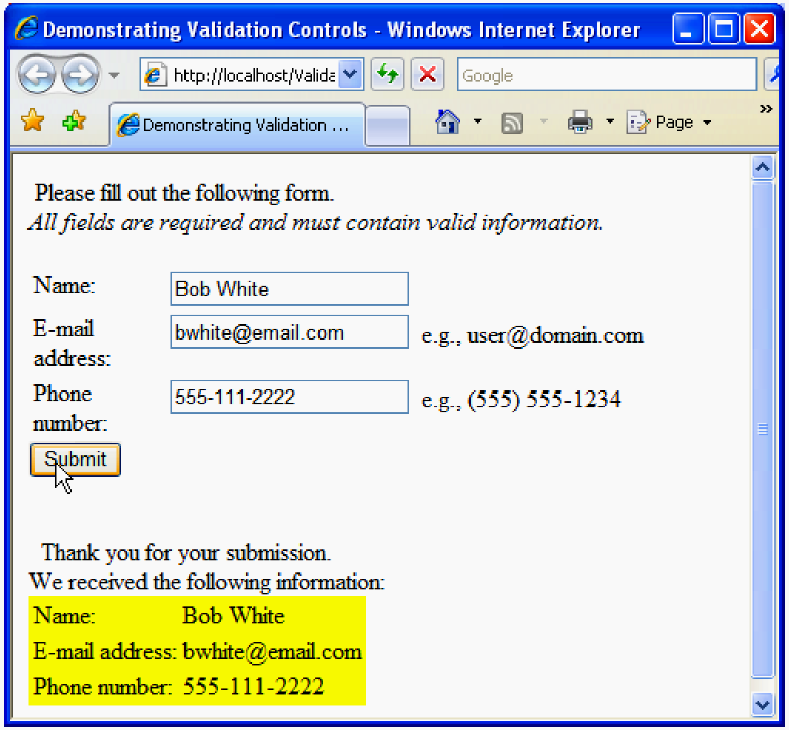
Please fill out the following form.

All fields are required and must contain valid information.

Name:	Bob White	
E-mail address:	bwhite	Please enter an e-mail address in a valid format.
Phone number:		e.g., (555) 555-1234

Submit

b) After the user fills out the form properly and clicks the **Submit** button, the submitted data is displayed at the bottom of the page with a partial page update.


Demonstrating Validation Controls - Windows Internet Explorer
http://localhost/Valida
Google
Demonstrating Validation ...
Page

Please fill out the following form.

All fields are required and must contain valid information.

Name:	Bob White	
E-mail address:	bwhite@email.com	e.g., user@domain.com
Phone number:	555-111-2222	e.g., (555) 555-1234

Submit

Thank you for your submission.
We received the following information:

Name:	Bob White
E-mail address:	bwhite@email.com
Phone number:	555-111-2222

Common Programming Error 25.1

Putting more than one instance of the `ScriptManager` control on a Web Form causes the application to throw an `InvalidOperationException` when the page is initialized.