



# Chapter 21: Web App Development with ASP.NET in C#, A Deeper Look

Internet & World Wide Web  
How to Program, 5/e

*Note:* This chapter is a copy of Chapter 27 of our book *Visual C# 2010 How to Program*. For that reason, we simply copied the PowerPoint slides for this chapter and *did not* re-number them



## OBJECTIVES

In this chapter you'll learn:

- To use the **Web Site Administration Tool** to modify web application configuration settings.
- To restrict access to pages to authenticated users.
- To create a uniform look-and-feel for a website using master pages.
- To use ASP.NET Ajax to improve the user interactivity of your web applications.



## **27.1** Introduction

## **27.2** Case Study: Password-Protected Books Database Application

- 27.2.1 Examining the ASP.NET Web Site Template
- 27.2.2 Test-Driving the Completed Application
- 27.2.3 Configuring the Website
- 27.2.4 Modifying the `Default.aspx` and `About.aspx` Pages
- 27.2.5 Creating a Content Page That Only Authenticated Users Can Access
- 27.2.6 Linking from the `Default.aspx` Page to the `Books.aspx` Page
- 27.2.7 Modifying the Master Page (`Site.master`)
- 27.2.8 Customizing the Password-Protected `Books.aspx` Page

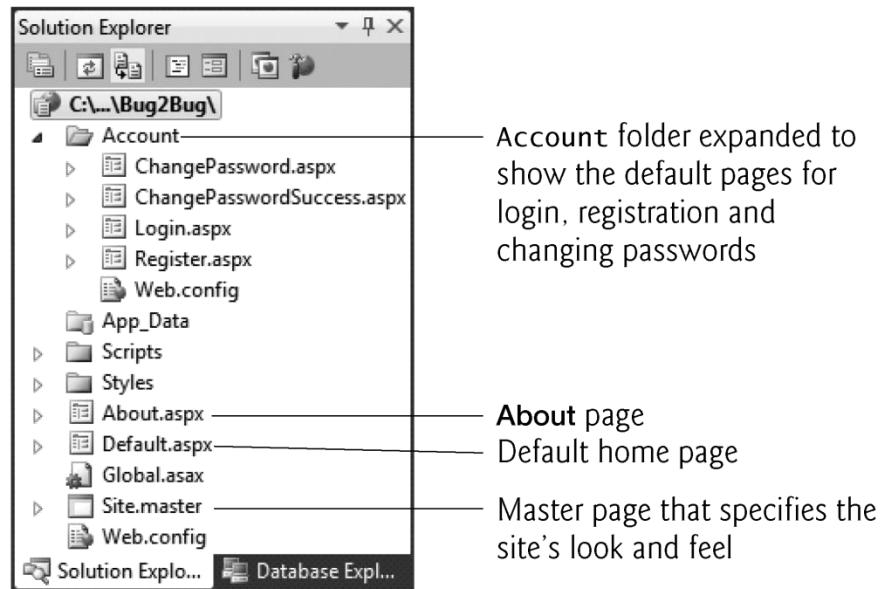
## **27.3** ASP.NET Ajax

- 27.3.1 Traditional Web Applications
- 27.3.2 Ajax Web Applications
- 27.3.3 Testing an ASP.NET Ajax Application
- 27.3.4 The ASP.NET Ajax Control Toolkit
- 27.3.5 Using Controls from the Ajax Control Toolkit

## **27.4** Wrap-Up



## 27.2.1 Examining the ASP.NET Web Site Template

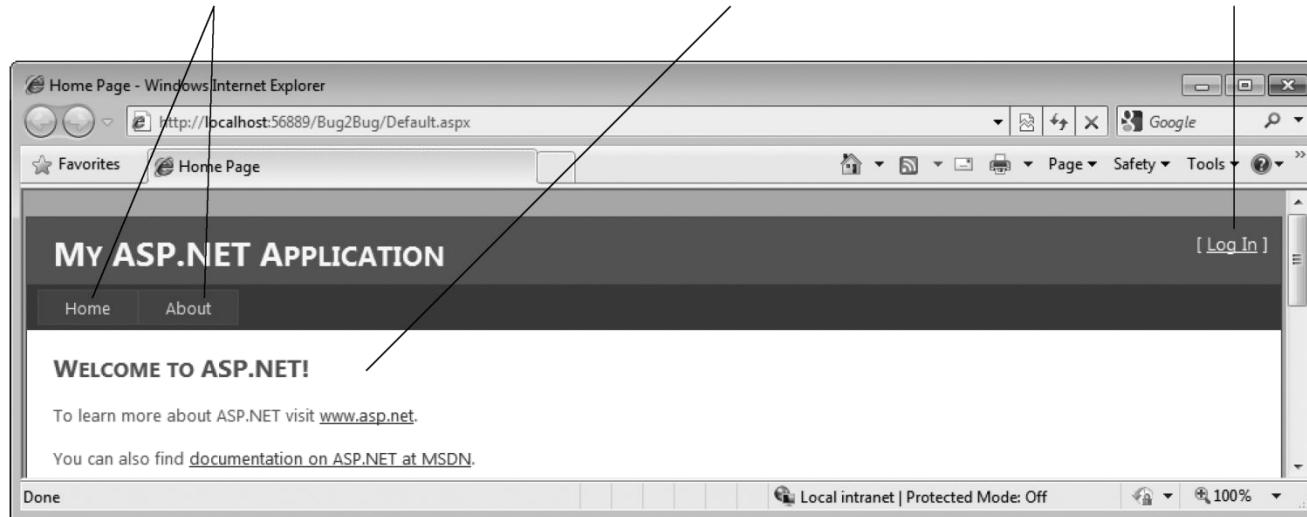


**Fig. 27.1** | The default ASP.NET Web Site in the Solution Explorer.

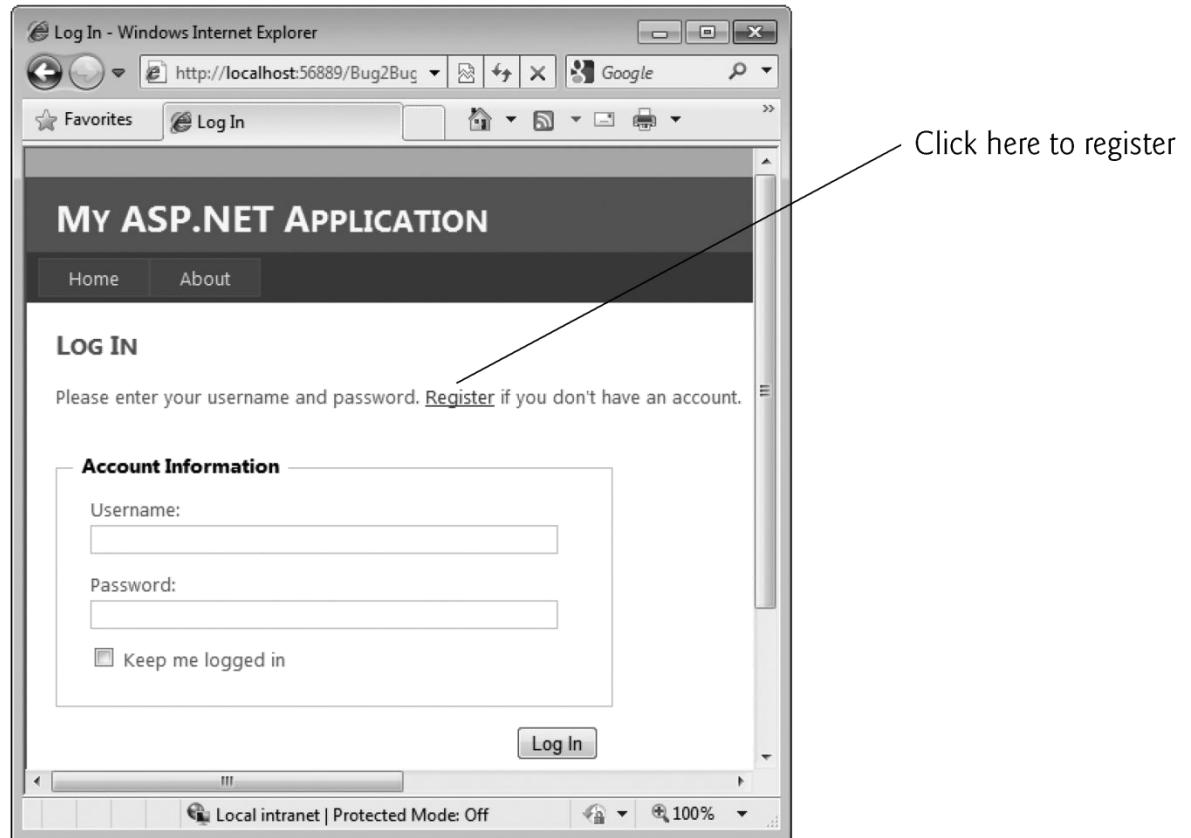
Navigation bar contains links to the **Home** and **About** pages

You can customize the content of each page and the look-and-feel of the website

Click this link to log into the website



**Fig. 27.2** | Default Home page of a website created with the ASP.NET Web Site template.



**Fig. 27.3** | Login page.



Register - Windows Internet Explorer

http://localhost: 4321 Google

Favorites Register

## My ASP.NET APPLICATION

Home About

### CREATE A NEW ACCOUNT

Use the form below to create a new account.

Passwords are required to be a minimum of 6 characters in length.

**Account Information**

User Name:

E-mail:

Password:

Confirm Password:

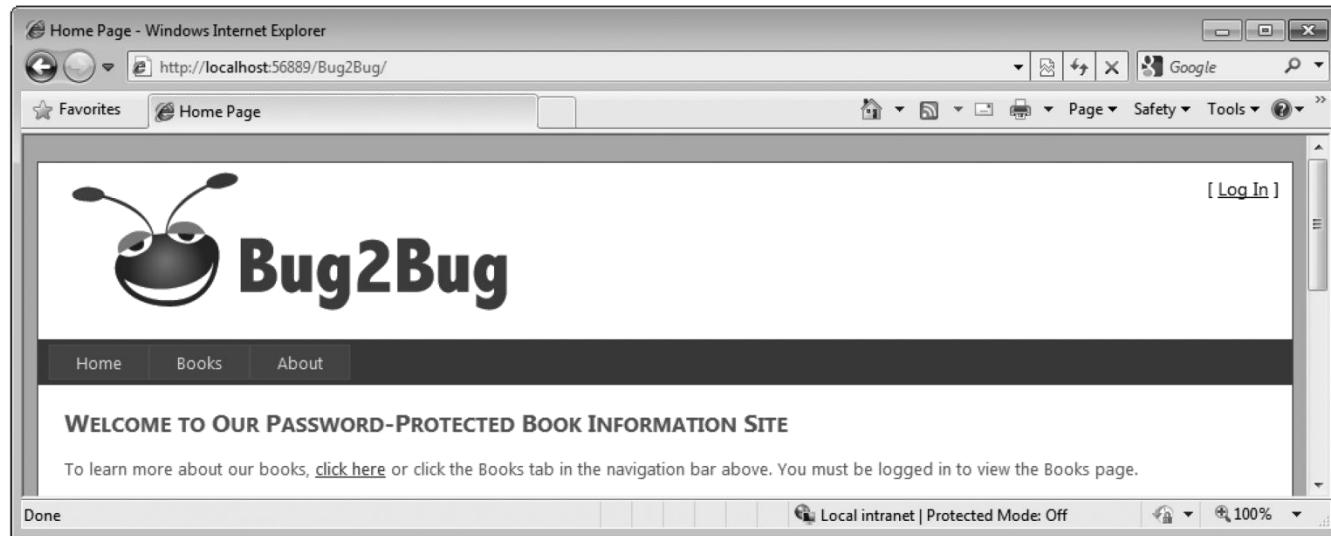
**Create User**

Local intranet | Protected Mode: Off 100%

**Fig. 27.4** | Register page.

## 27.2.2 Test-Driving the Completed Application





**Fig. 27.5** | Home page for the completed Bug2Bug website.



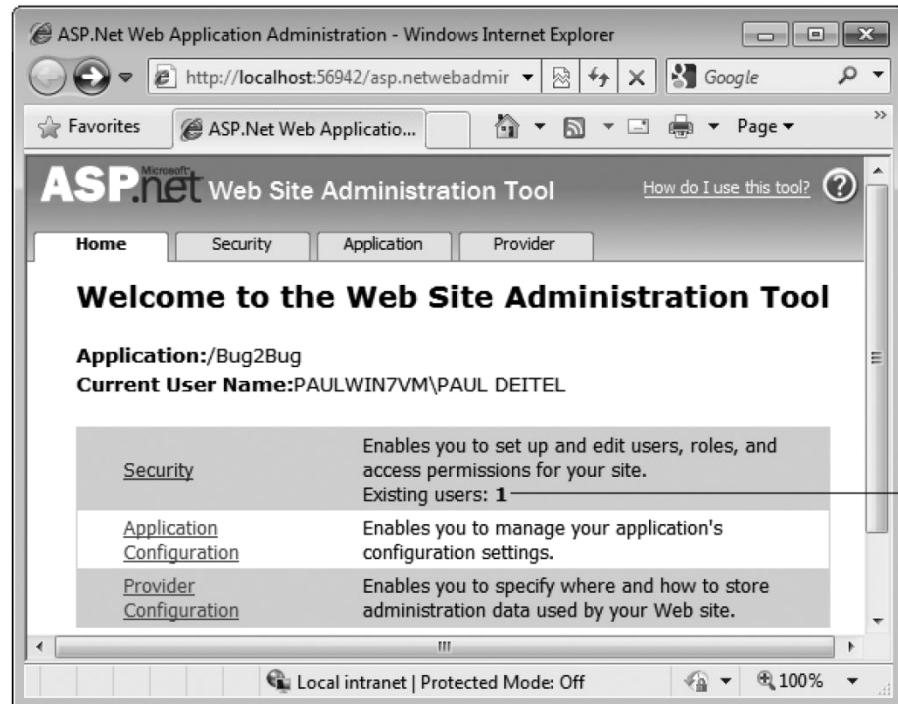
**Fig. 27.6** | Books.aspx displaying books by Harvey Deitel (by default).



**Fig. 27.7** | Books.aspx displaying books by Greg Ayer.



## 27.2.3 Configuring the Website



This will say 0 if you have not yet created an account to test the website

**Fig. 27.8** | Web Site Administration Tool for configuring a web application.

ASP.NET Web Application Administration - Windows Internet Explorer  
http://localhost:56942/asp.netwebadminfiles/secur

ASP.NET Web Site Administration Tool

Home Security Application Provider

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.

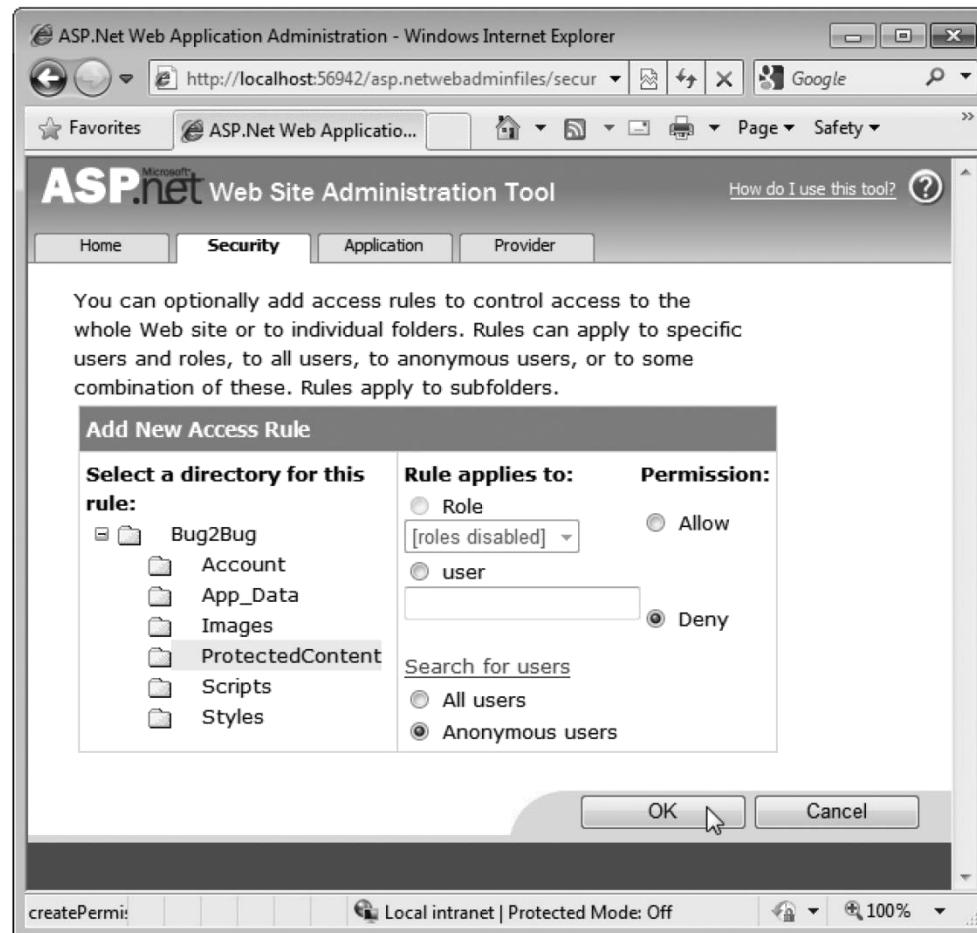
Click the links in the table to manage the settings for your application.

Users	Roles	Access Rules
Existing users: 1 <a href="#">Create user</a> <a href="#">Manage users</a>  <a href="#">Select authentication type</a>	Roles are not enabled <a href="#">Enable roles</a> <a href="#">Create or Manage roles</a>	<a href="#">Create access rules</a> <a href="#">Manage access rules</a>

Local intranet | Protected Mode: Off

This will say 0 if you have not yet created an account to test the website

Fig. 27.9 | Security page of the Web Site Administration Tool.

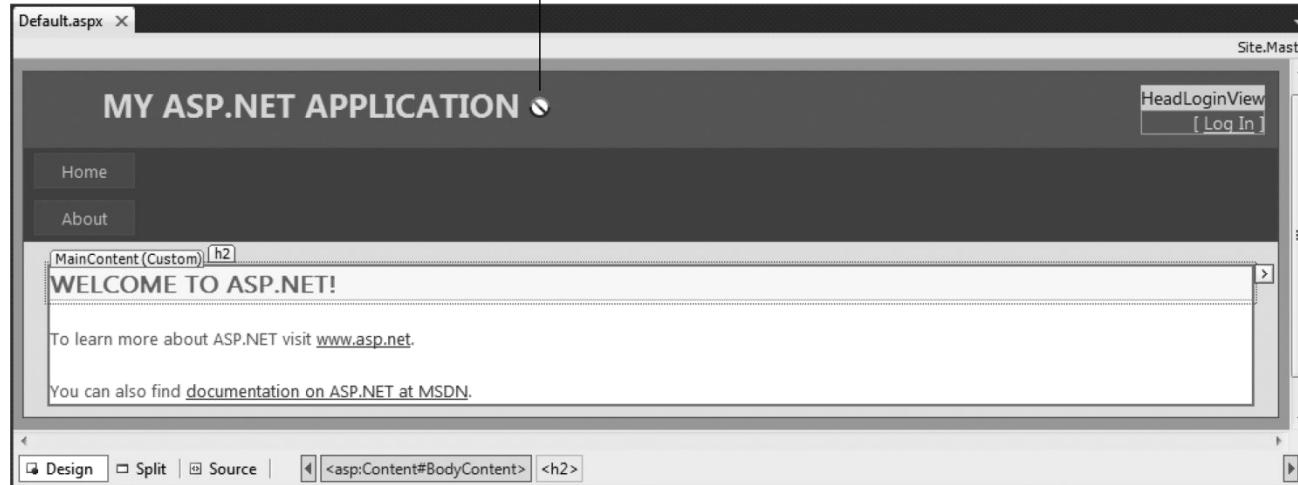


**Fig. 27.10** | Add New Access Rule page used to configure directory access.

## 27.2.4 Modifying the Default.aspx and About.aspx Pages



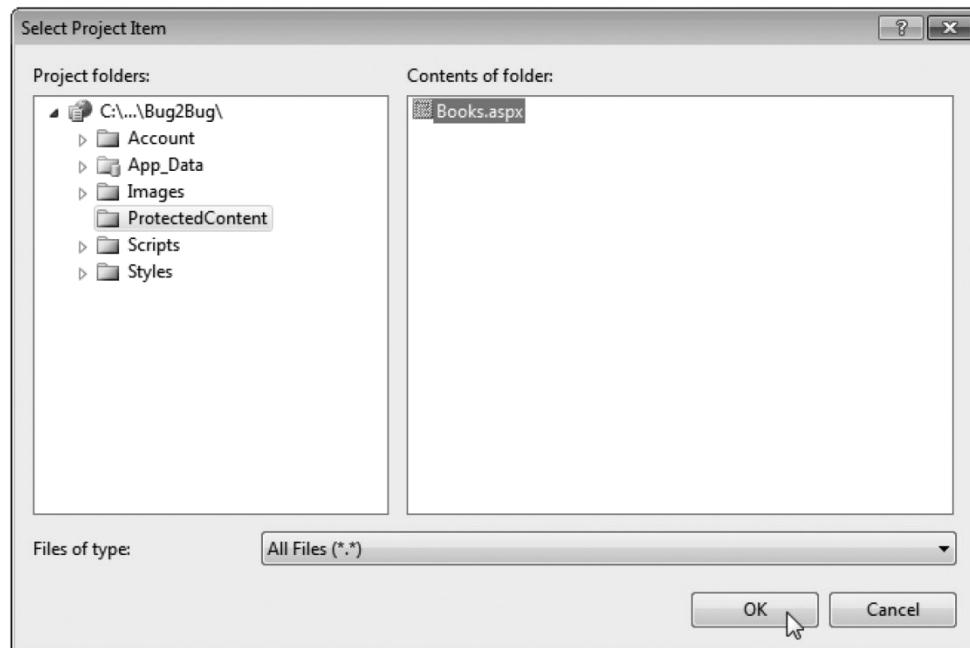
This cursor indicates a part of a content page that cannot be edited because it's inherited from a master page



**Fig. 27.11** | Default.aspx page in Design view.



## 27.2.6 Linking from the Default.aspx Page to the Books.aspx Page



**Fig. 27.12** | Selecting the Books.aspx page from the Select Project Item dialog.

## 27.2.7 Modifying the Master Page (Site.master)





◀ <html> <body> <form> <div.page> <div.header> <div.title> <h1> <asp:Image#Image1>

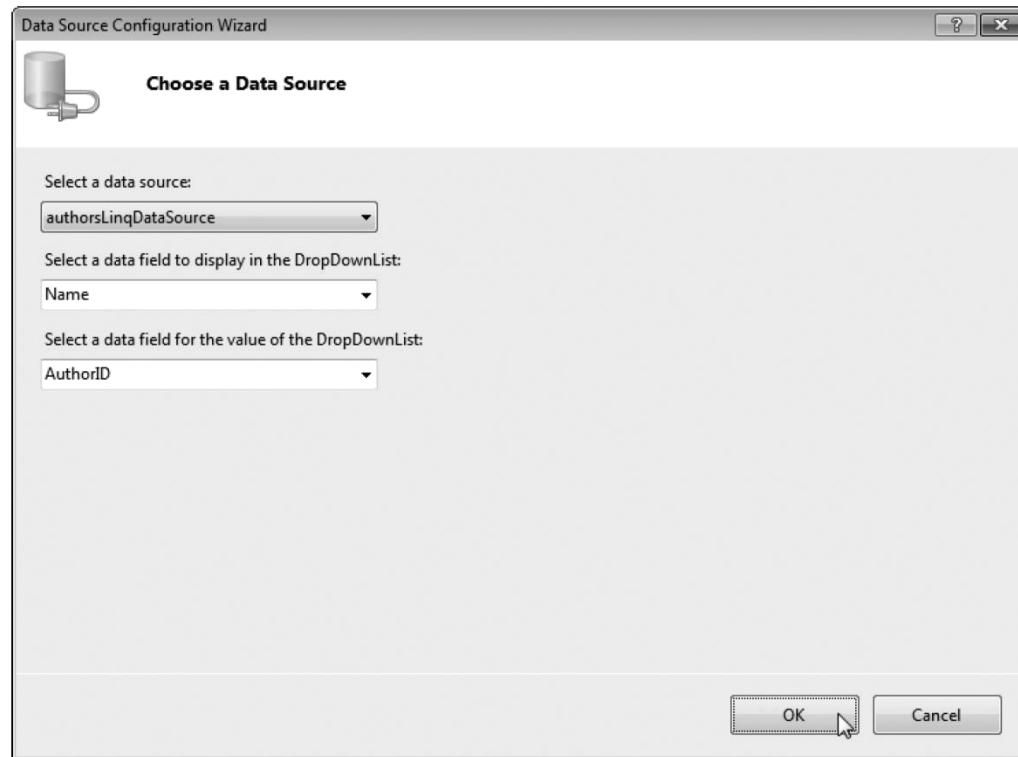
**Fig. 27.13** | Buttons for selecting parts of a page in Design view.

## 27.2.8 Customizing the Password-Protected Books.aspx Page

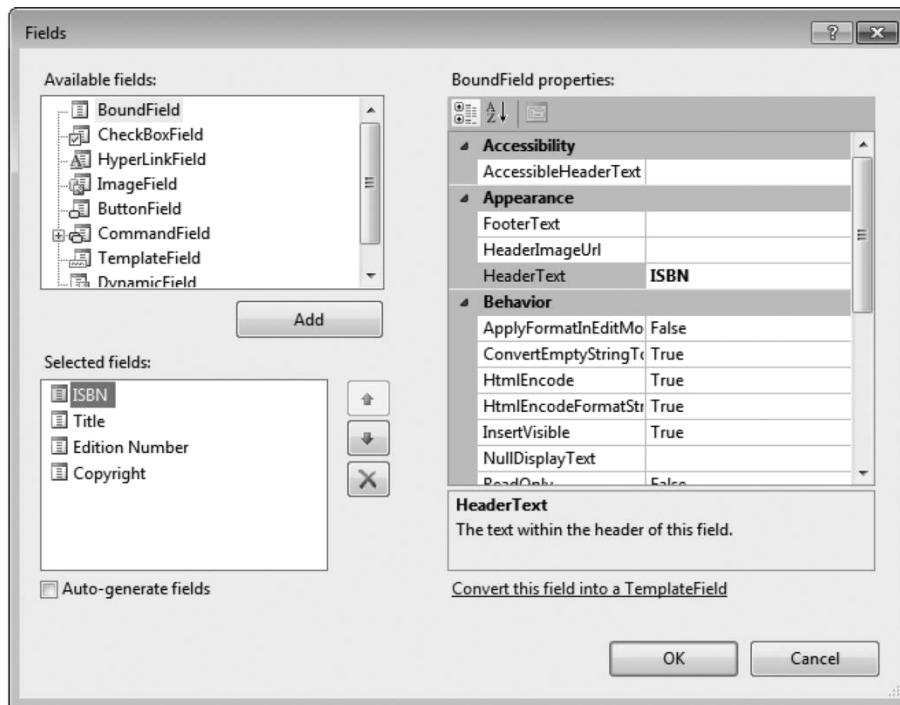




**Fig. 27.14** | Object Relational Designer for the Books database.



**Fig. 27.15** | Choosing a data source for a DropDownList.



**Fig. 27.16** | Creating GridView fields in the Fields dialog.



```
1 // Fig. 27.17: ProtectedContent_Books.aspx.cs
2 // Code-behind file for the password-protected Books page.
3 using System;
4 using System.Linq;
5 using System.Web.UI.WebControls;
6
7 public partial class ProtectedContent_Books : System.Web.UI.Page
8 {
9     // data context queried by data sources
10    BooksDataContext database = new BooksDataContext();
11
12    // specify the Select query that creates a combined first and last name
13    protected void authorsLinqDataSource_Selecting( object sender,
14        LinqDataSourceSelectEventArgs e )
15    {
16        e.Result =
17            from author in database.Authors
18            select new { Name = author.FirstName + " " + author.LastName,
19                        author.AuthorID };
20    } // end method authorsLinqDataSource_Selecting
21
```

**Fig. 27.17** | Code-behind file for the password-protected Books page. (Part 1 of 2.)



```
22 // specify the Select query that gets the specified author's books
23 protected void titlesLinqDataSource_Selecting( object sender,
24     LinqDataSourceSelectEventArgs e )
25 {
26     e.Result =
27         from book in database.AuthorISBNs
28         where book.AuthorID ==
29             Convert.ToInt32( authorsDropDownList.SelectedValue )
30         select book.Title;
31 } // end method titlesLinqDataSource_Selecting
32
33 // refresh the GridView when a different author is selected
34 protected void authorsDropDownList_SelectedIndexChanged(
35     object sender, EventArgs e )
36 {
37     titlesGridView.DataBind(); // update the GridView
38 } // end method authorsDropDownList_SelectedIndexChanged
39 } // end class ProtectedContent_Books
```

**Fig. 27.17** | Code-behind file for the password-protected Books page. (Part 2 of 2.)



ProtectedContent/Books.aspx × ..../Site.master

**Bug2Bug**

Home Books About

MainContent (Custom)

Author: Databound ▾

LinqDataSource - authorsLinqDataSource  
asp:gridview#titlesGridView

ISBN	Title	Edition Number	Copyright
Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound

1 2

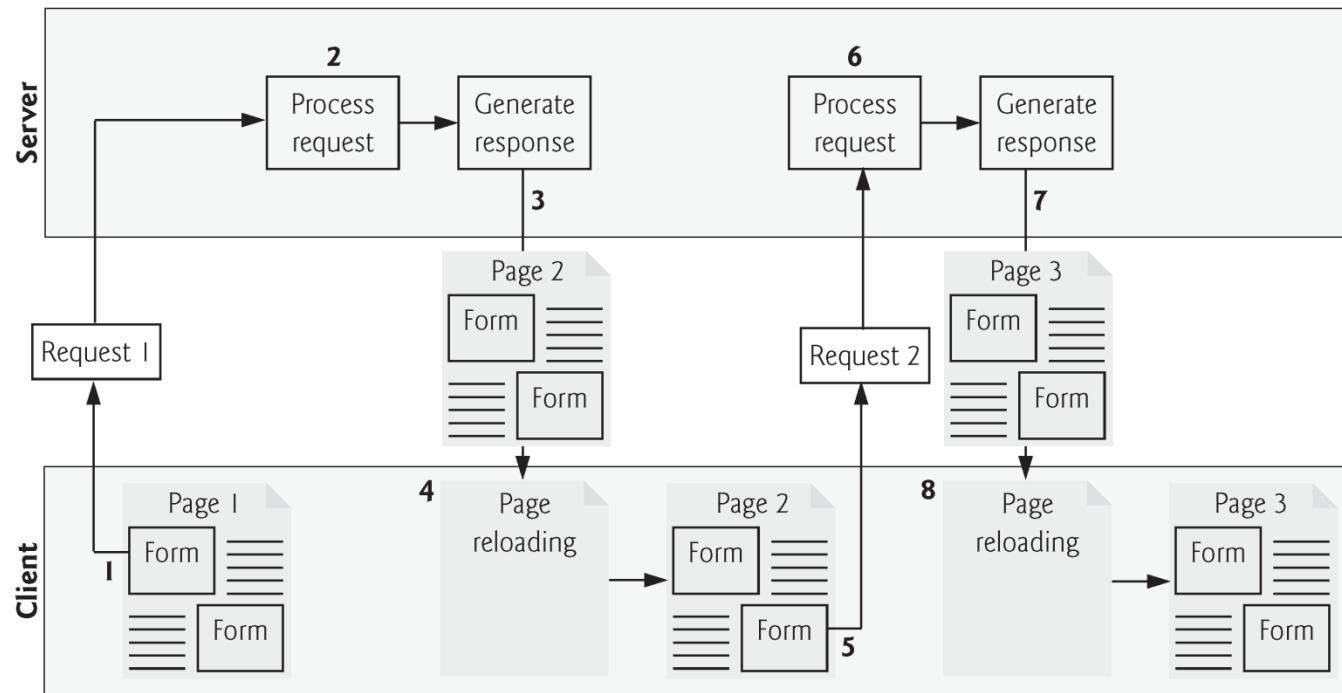
LinqDataSource - titlesLinqDataSource

Design Split Source <asp:Content#Content2> <p> <asp:GridView#titlesGridView> >

**Fig. 27.18** | Completed Books.aspx page in Design mode.

## 27.3.1 Traditional Web Applications

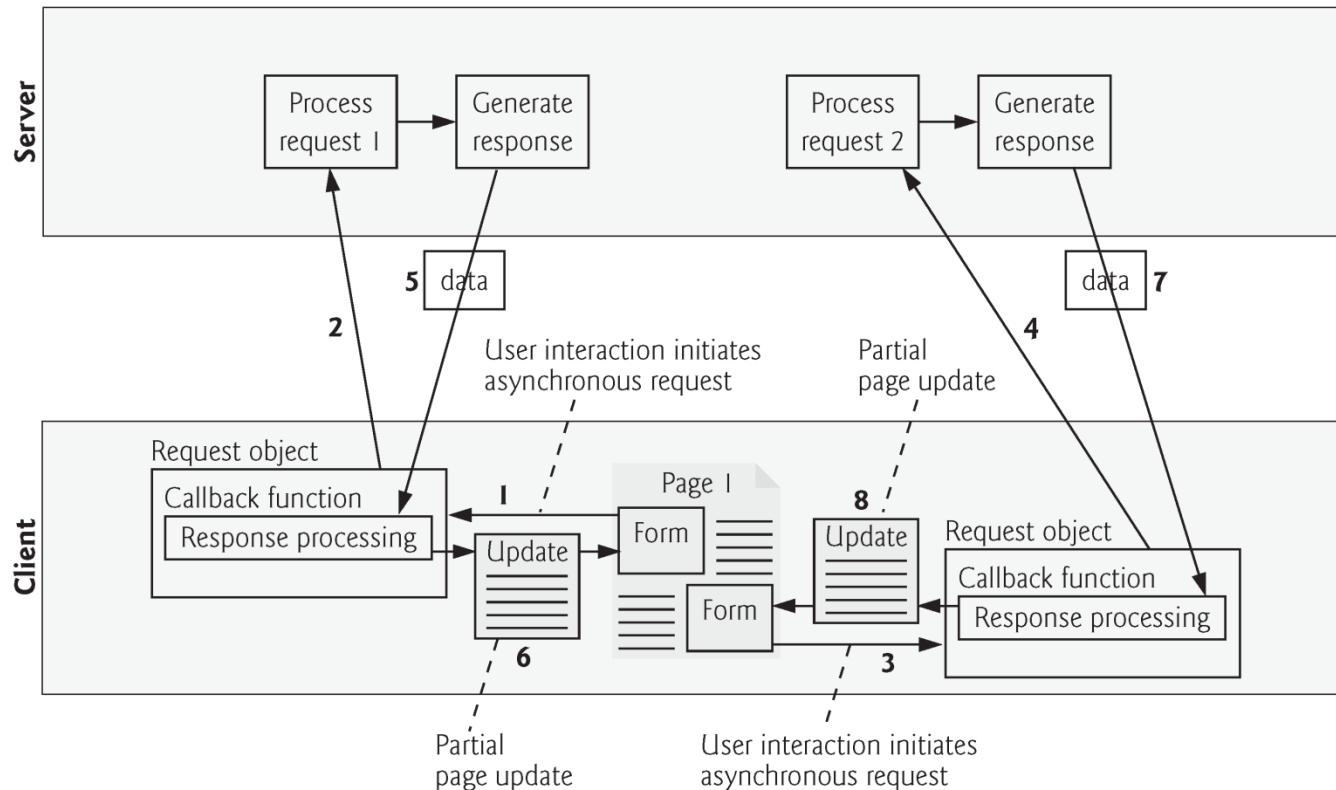




**Fig. 27.19** | Traditional web application reloading the page for every user interaction.



## 27.3.2 Ajax Web Applications

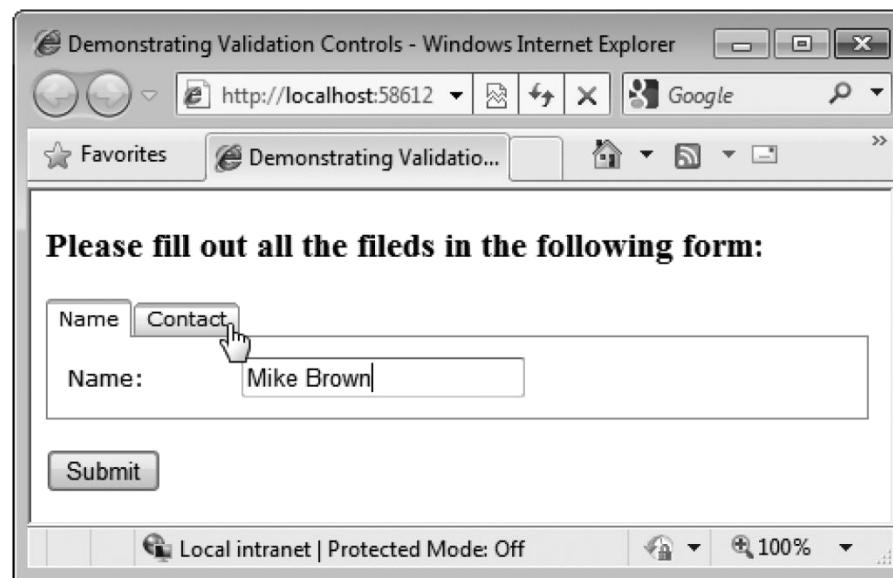


**Fig. 27.20** | Ajax-enabled web application interacting with the server asynchronously.

## 27.3.3 Testing an ASP.NET Ajax Application



a) Entering a name on the **Name** tab then clicking the **Contact** tab



Please fill out all the fields in the following form:

Name  Contact

Name: Mike Brown

Submit

Local intranet | Protected Mode: Off 100%

**Fig. 27.21** | Validation application enhanced by ASP.NET Ajax. (Part 1 of 3.)

b) Entering an e-mail address in an incorrect format and pressing the *Tab* key to move to the next input field causes a callout to appear informing the user to enter an e-mail address in a valid format

**Fig. 27.21** | Validation application enhanced by ASP.NET Ajax. (Part 2 of 3.)

c) After filling out the form properly and clicking the **Submit** button, the submitted data is displayed at the bottom of the page with a partial page update

The screenshot shows a Windows Internet Explorer window with the title "Demonstrating Validation Controls - Windows Internet Explorer". The address bar shows the URL "http://localhost:58612". The page content is as follows:

**Please fill out all the fields in the following form:**

Name  Contact

E-mail:  e.g., email@domain.com

Phone:  e.g., (555) 555-1234

Thank you for your submission  
We received the following information:  
Name: Mike Brown  
E-mail: mbrown@deitel.com  
Phone: (555) 555-1234

Local intranet | Protected Mode: Off | 100%

**Fig. 27.21** | Validation application enhanced by ASP.NET Ajax. (Part 3 of 3.)

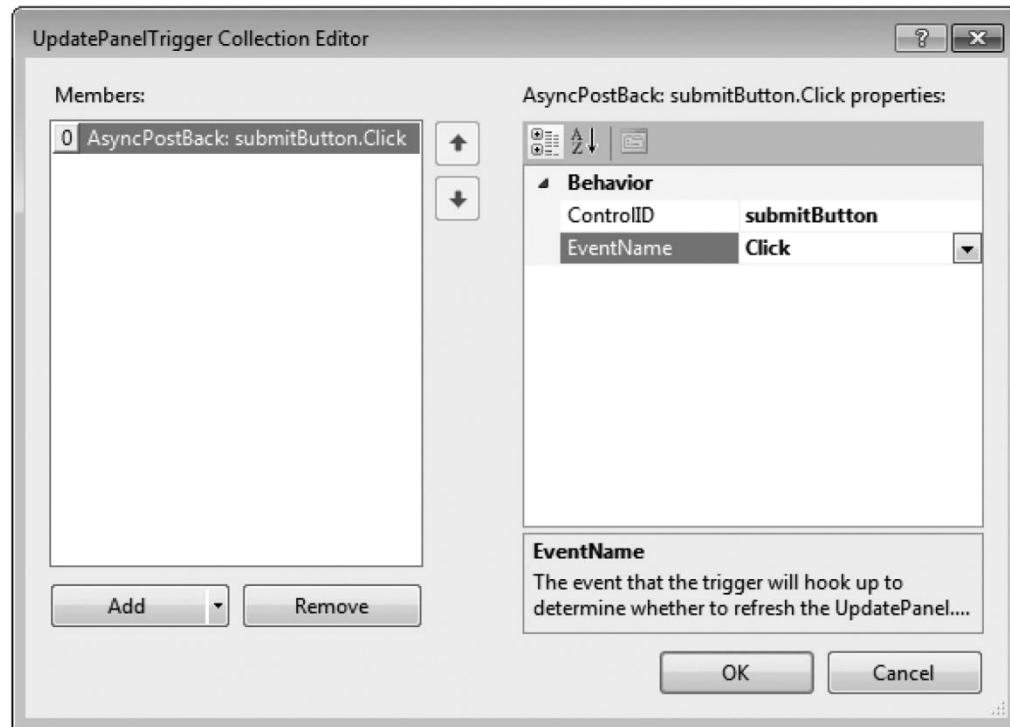


## 27.3.5 Using Controls from the Ajax Control Toolkit

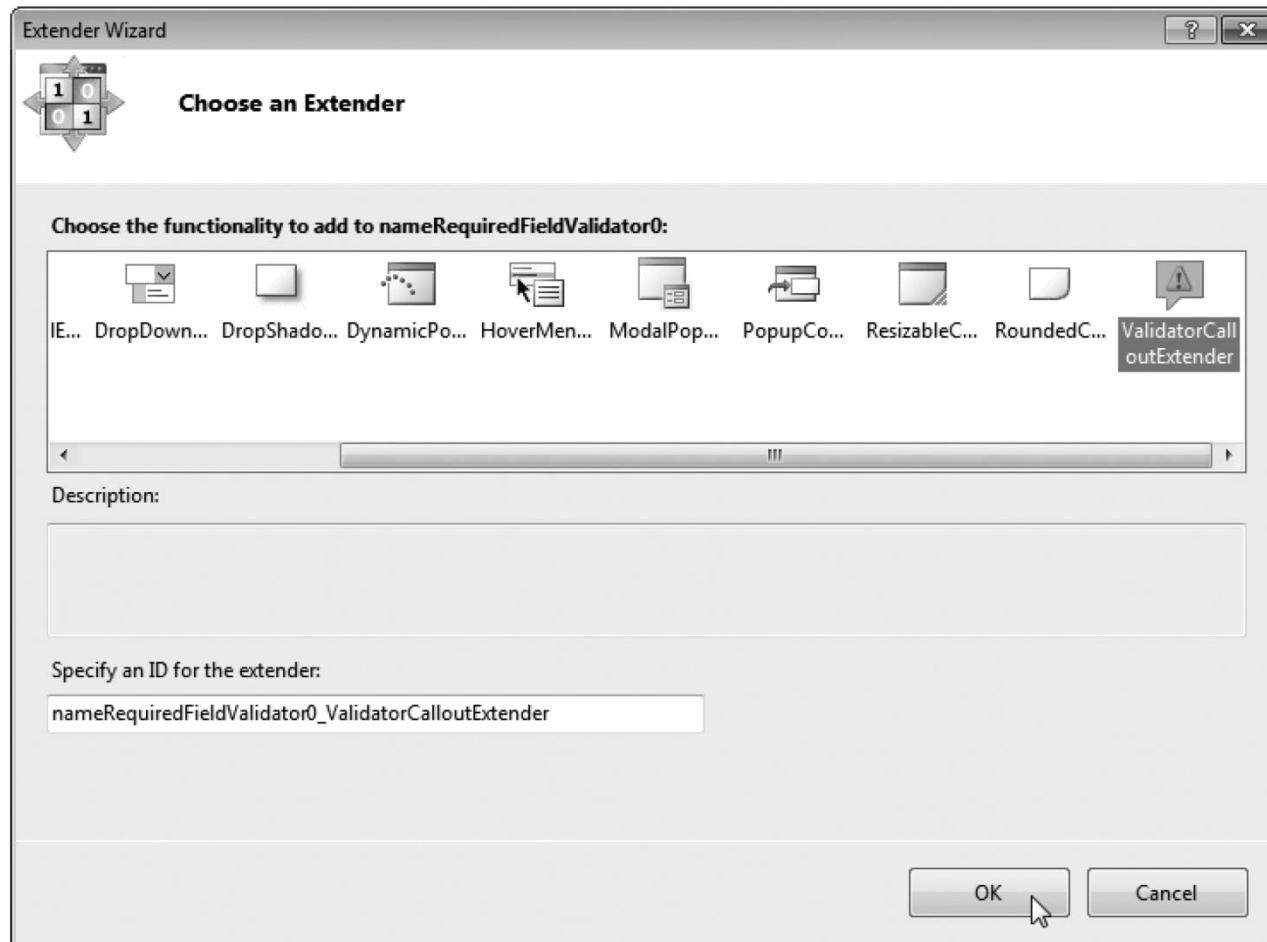


## Common Programming Error 27.1

Putting more than one `ScriptManager` and/or `ToolkitScriptManager` control on a Web Form causes the application to throw an `InvalidOperationException` when the page is initialized.



**Fig. 27.22** | Creating a trigger for an UpdatePanel1.



**Fig. 27.23** | Creating a control extender using the Extender