Lab 4: ASP.NET 2.0 Membership, Login Controls, and Role Management

Forms authentication is a popular means of securing Internet applications. In ASP.NET's brand of forms authentication, you designate a login page for logging in unauthenticated users and use URL directives to specify which pages require logins and which ones do not. When an anonymous (that is, unauthenticated) user attempts to access a page that's only available to authenticated users, ASP.NET automatically redirects them to the login page.

ASP.NET 1.x simplified forms authentication by introducing declarative means for designating login pages and specifying authorization rules. ASP.NET 2.0 further simplifies forms authentication by providing the following facilities:

A membership service for managing users and user credentials
Login controls for logging in users, creating new users, and more
A role management service for enacting role-based security

In this lab, you'll use all three to build a secure administrative interface for the MyComics application. You'll begin by creating a login page. Then you'll secure the admin page you created in Lab 2 so that only authenticated users can access it, and add links to the master page for navigating to the admin page and logging in. Finally, you'll use role-based security to tighten permissions on the admin page so that only administrators can access it.

Estimated time to complete this lab: 60 minutes

Lab Setup

If you have not completed Lab 2 (ASP.NET 2.0 Data Access) previous to this lab, enable database caching as follows:

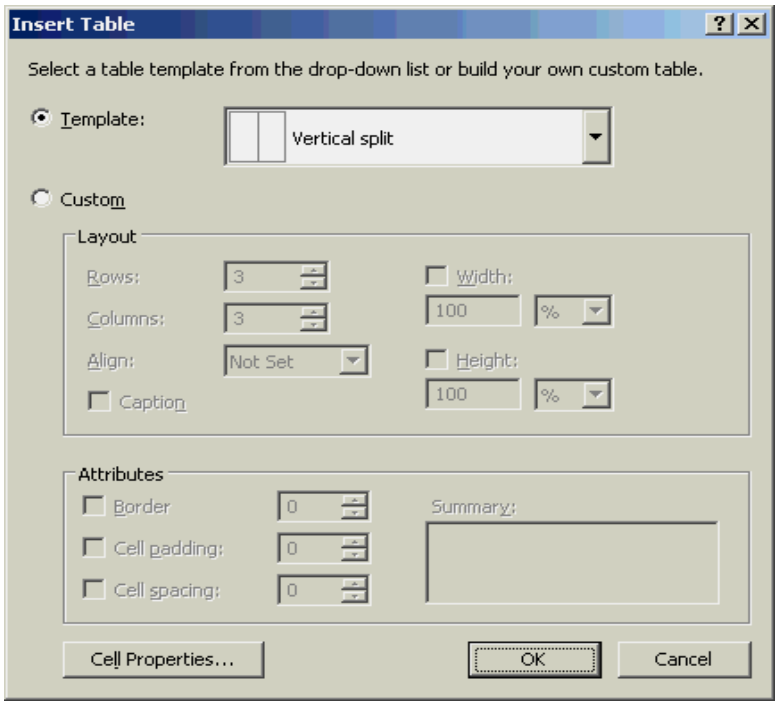
a. Open a Visual Studio command prompt window. You'll find it under "All Programs->Microsoft Visual Studio 2005 ->Visual Studio Tools->Visual Studio Command Prompt."

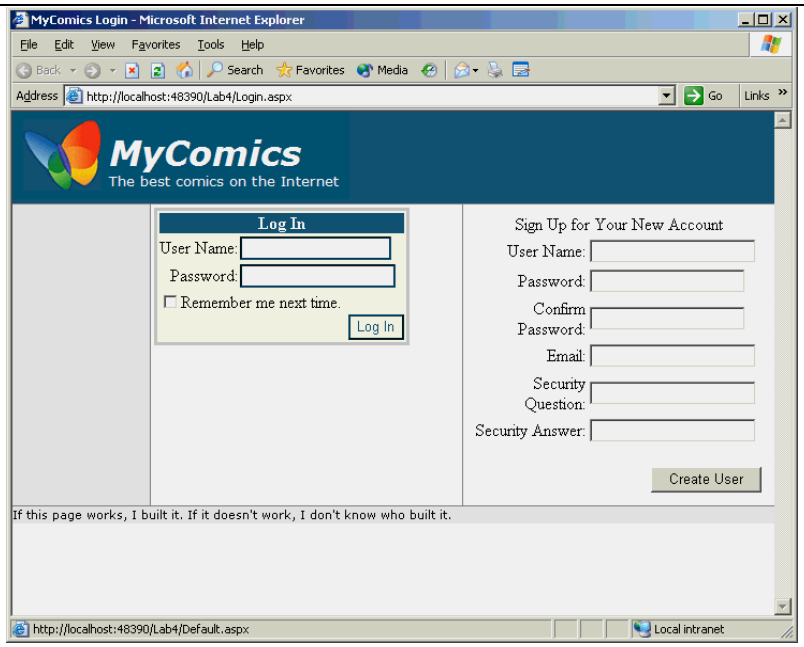
b. Navigate to the C:\MSLabs\ASP.NET\LabFiles\Database directory.

c. Type CacheSetup.

Exercise 1
Add a login page

In this exercise, you'll create a login page that includes a Login control for logging in existing users and a CreateUserWizard control for registering new users.

Tasks	Detailed Steps
1. Open the Web site	**a.** Start Microsoft Visual Studio and use the "File->Open Web Site" command to open the C:\MSLabs\ASP.NET\Starter\<Language>\Lab4 site.
2. Add a page to the Web site	**a.** Right-click C:\..\Lab4 in Solution Explorer and use the "Add New Item" command to add a page named Login.aspx. Choose "Web Form" as the template type and check the "Place code in separate file" and "Select master page" boxes. Before clicking the Add button, be sure C# or Visual Basic is selected in the Language box. **b.** When asked to pick a master page, select Site.master. **c.** Change Title="Untitled Page" to Title="MyComics Login" in Login.aspx's @ Page directive. **d.** Click the Design button to switch to Design view. **e.** Place the cursor in the Content control, and then use the "Layout->Insert Table" command to add a "vertical split" table:  **f.** Right-click the left table cell and select "Insert->Cell to the Right." **g.** Set the middle cell's Width to 1 and its BgColor to Gray. **h.** Set the Width of the leftmost cell to 50%. Also set its VAlign property to "top." **i.** Set the VAlign property of the rightmost table cell to "top." You're now ready to add controls to the page.
3. Add a Login control and a CreateUserWizard control	**a.** Drag a Login control from the Toolbox and drop it into the leftmost table cell. **b.** Drag a CreateUserWizard control from the Toolbox and drop it into the rightmost table cell. **c.** Press Ctrl+F5 to launch Login.aspx. Here's what you should see:



NOTE:

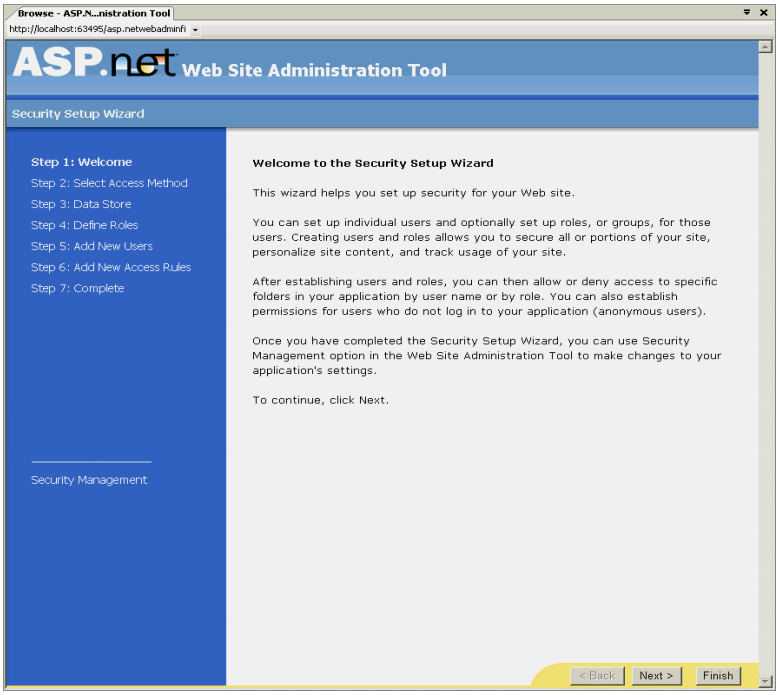
If you want to polish the page's appearance, use the "Auto Format" command in both the "Login Tasks" and "CreateUserWizard Tasks" menus.

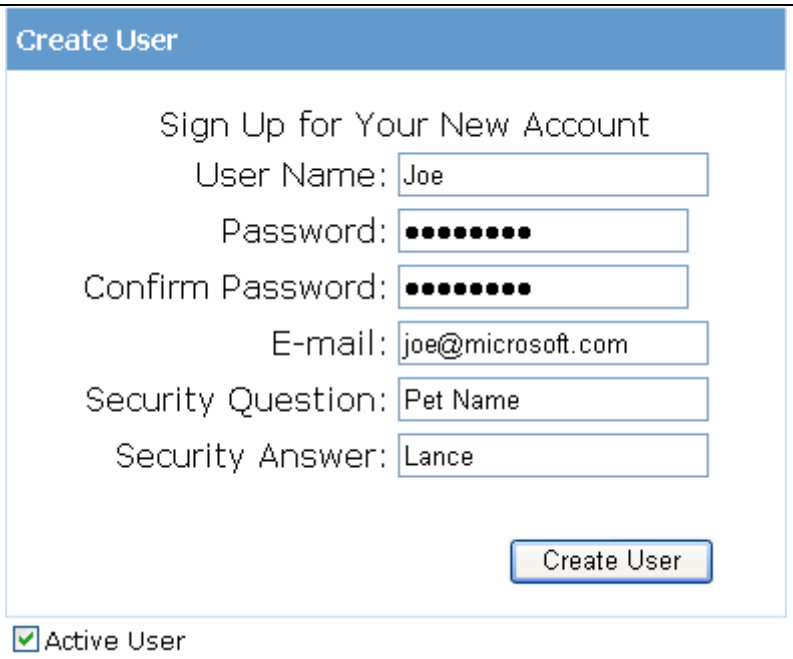
d. Close the browser and return to Visual Studio.

Exercise 2

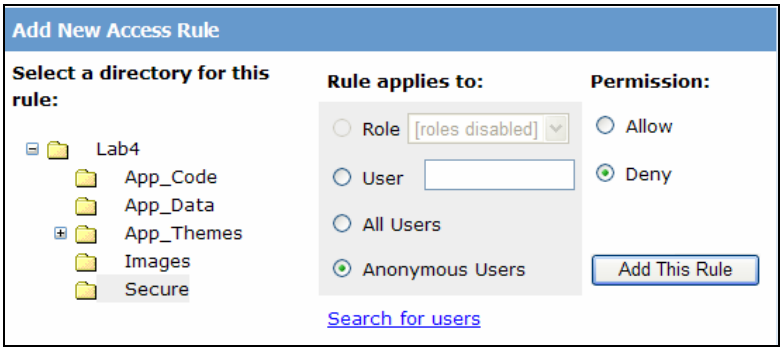
Apply authentication and authorization settings

In this exercise, you'll turn on forms authentication and configure the admin page so that it's accessible only to authenticated users. You'll also configure the application to use ASP.NET's SQL Server membership provider and create a database for the provider to use.

Tasks	Detailed Steps
1. Move the admin page	**a.** Right-click C:\..\Lab4 in Solution Explorer and and select **"New Folder."** Name the folder "Secure". **b.** Move Admin.aspx and Admin.aspx.cs<vb> to the Secure folder.
2. Run the Security Setup Wizard	**3.** Use Visual Studio's "Website->ASP.NET Configuration" command to display the Web Site Administration Tool. **a.** In the Web Site Administration Tool, click the Security tab at the top of the page or the Security link in the body of the page. **b.** Click "Use the security Setup Wizard to configure security step by step" to start the Security Setup Wizard.  **c.** In step 1 of the Security Setup Wizard ("Welcome"), click Next. **d.** In step 2 of the Security Setup Wizard ("Select Access Method"), select "From the internet" and click Next. **e.** In step 3 ("Data Store"), click Next. **f.** In step 4 ("Define Roles"), click Next. You'll enable role management later. **g.** In step 5 ("Add New Users"), fill in the form to create a user. (Remember the user name and password you enter here because you'll need them later on. Use Joe and P@ssw0rd. Also notice that the Web Site Administration Tool enforces strong passwords here.) Then click the "Create User" button followed by the Next button.



h. In step 6 ("Add New Access Rules"), select the Secure folder in the tree view under "Select a directory for this rule" as shown below. Then select "Anonymous users" and "Deny" and click "Add This Rule" to add a rule to Web.config denying unauthenticated users access to the Secure directory. Then click Next.



i. In step 7 ("Complete"), click Finish.

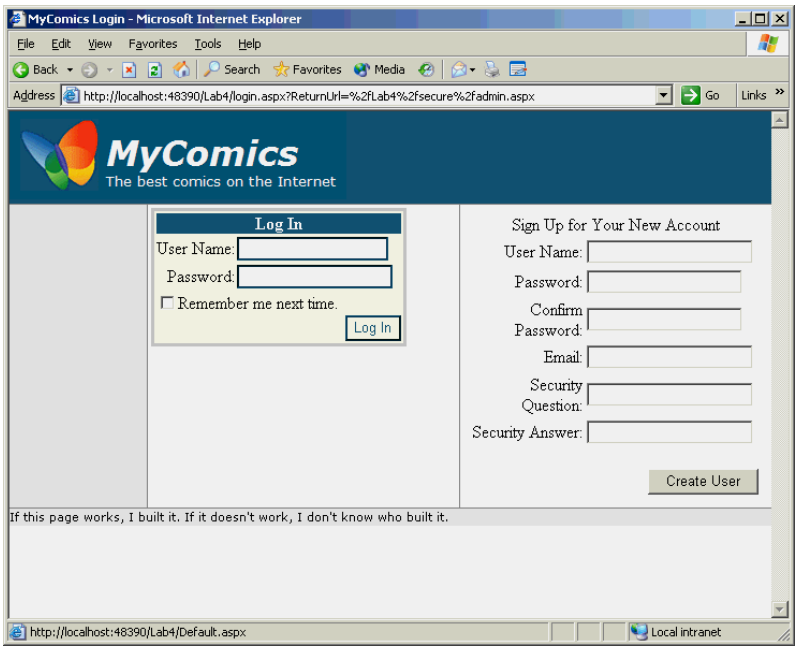
j. Swich to Visual Studio 2005 and open Web.config and try to identify the changes made by the Security Setup Wizard.

k. Even though it doesn't appear in the Solution Explorer window, a new Web.config file was added to the Web site by the Security Setup Wizard. That file is in the Secure folder, and it contains authorization rules for the Secure directory. Make the new Web.config file visible by right-clicking the Secure folder in Solution Explorer and using the "Add Existing Item" command to select the Web.config file in the application's Secure folder.

l. Open Secure\Web.config and inspect its <authorization> element. How does this element relate back to the authorization rule you created in the Security Setup Wizard that denies anonymous users access to resources in the Secure folder?

m. Even though it doesn't appear in the Solution Explorer window, a new ASPNETDB.mdf database file was added to the Web site. Make the ASPNETDB database visible by right-clicking the App _Data folder in Solution Explorer and using the "Add Existing Item" command to select the ASPNETDB.MDB file in the application's App_Data folder.

By creating a user through the Web Admin you created a SQL Server Express database contained within your application. This database contains the

	security settings for this application. **n.** Right click on ASPNETDB.MDF and select Open. **o.** You will transferred to the Server Explorer window and there will be a connection for the ASPNETDB database. **p.** Drill into the tables of ASPNETDB and review. See if you can find the user you added.
4. Test the security settings	**a.** Select Default.aspx in the Solution Explorer window and press Ctrl+F5 to launch it. Verify that Default.aspx (not the login page) appears in your browser. **b.** Change "Default.aspx" in the browser's address bar to "Secure/Admin.aspx." Verify that Login.aspx appears, as shown below.  **c.** Log in using the user name and password that you created in the Security Setup Wizard. Verify that Admin.aspx appears in the browser. **d.** Click the browser's Back button twice to return to Default.aspx. **e.** Change "Default.aspx" in the browser's address bar to "Secure/Admin.aspx" again. Verify that Admin.aspx appears without requiring another login. **f.** Close the browser and return to Visual Studio.

Exercise 3

Add navigation links to the master page

In this exercise, you'll add links to the master page for navigating to Admin.aspx and for logging in and out. The former link will be a conventional HyperLink control, the latter a LoginStatus control. You'll also add a public method to the master page enabling the login page to hide these links.

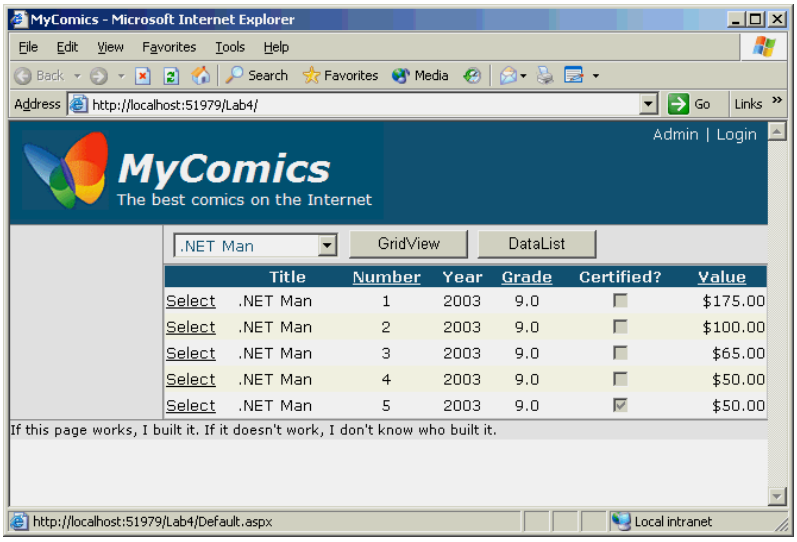
Tasks	Detailed Steps

1. Add links to the master page	**a.** Open Site.master in the designer and switch to Source view.
	b. Find the table cell containing the HyperLink control that points to ~/Images/MyComics.gif. Currently the cell's content is defined like this:
	`<asp:HyperLink ID="HyperLink1" ImageUrl="~/Images/MyComics.gif" NavigateUrl="~/Default.aspx" Runat="server">Hyperlink</asp:HyperLink>`
	c. Edit the contents of the cell to look like this: (text can be found in C:\ MSLabs\ASP.NET\LabFiles\lab4.txt)
	`<table width="100%" cellpadding="0" cellspacing="0">`
	`<tr>`
	`<td>`
	`<asp:HyperLink ID="HyperLink1" ImageUrl="~/Images/MyComics.gif" NavigateUrl="~/Default.aspx" Runat="server" />`
	`</td>`
	`<td align="right" valign="top">`
	`<asp:Panel ID="LinksPanel" Runat="server">`
	`<asp:HyperLink ID="AdminLink" Text="Admin" NavigateUrl="~/Secure/Admin.aspx" Runat="server" ForeColor="White" Font-Names="Verdana" Font-Size="10pt" EnableTheming="False" Style="text-decoration: none" />`
	`<span style="font-size: 10pt; color: white; font-family: Verdana">|</span>`
	`<asp:LoginStatus ID="LoginStatusControl" Runat="server" ForeColor="White" Font-Names="Verdana" Font-Size="10pt" EnableTheming="False" Style="text-decoration: none" />`
	`<span style="font-size: 10pt; color: white; font-family: Verdana">`
	` `
	`</span>`
	`</asp:Panel>`
	`</td>`
	`</tr>`
	`</table>`
	
	What you're doing is embedding a 1-row, 2-cell HTML table inside an existing cell—the one containing the MyComics banner—and placing the existing HyperLink in the left-hand cell while adding a HyperLink and a LoginStatus control to the right-hand cell. Putting the new controls in their own cell enables you to position them in the banner's

upper-right corner. You're also encapsulating the controls (and the text that separates them) in a Panel control so that later on you can show and hide them by toggling the Panel's Visible property.

This modification could have been made in Design view, but it's arguably easier to do it in Source View. Visual Studio is completely supportive of hand-written HTML, and for good reason!

d. Run Default.aspx in your browser and verify that links entitled "Admin" and "Login" appear in the page's upper-right corner, as pictured below. Also verify that clicking either link takes you to the login page.



e. Close the browser and return to Visual Studio.

2. Hide the links on the login page

a. Open Site.master.<cs or vb> and add the following method to the Site class:

C#

```
public void HideBannerLinks ()
{
LinksPanel.Visible = false;
}
```

VB

```
Public Sub HideBannerLinks()
       LinksPanel.Visible = False
End Sub
```

b. Open Login.aspx in Design view and double-click the body of the page to add a Page_Load method to Login.aspx.<cs. Or vb>

c. Add the following statement to the body of the Page_Load method:

C#

```
((Site) Master).HideBannerLinks ();
```

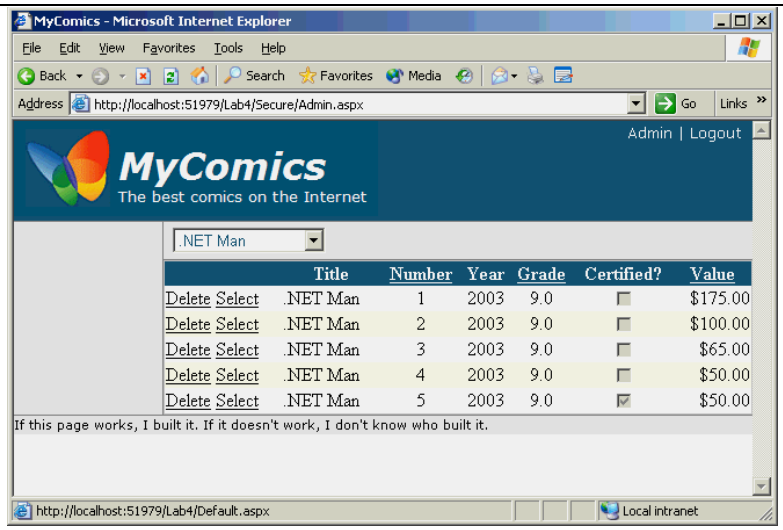
VB

```
CType(Master, Site).HideBannerLinks()
```

d. Run Default.aspx in your browser.

e. Click the Admin link in the upper-right corner. When the login page appears, verify that the "Admin" and "Login" links do NOT appear.

f. Log in using the user name and password you registered earlier. Verify that Admin.aspx appears, and that the link in the upper-right corner reads "Logout" instead of "Login," as shown below.



3. Change the logout action.

a. In the Admin page, click the Logout button in the upper-right corner. What happens?

b. Close the browser and return to Visual Studio.

c. Rather than return to the login page after clicking "Logout," you might prefer to return to the home page (Default.aspx). To that end, open Site.master in Design view and select the LoginStatus control.

d. Set the LoginStatus control's LogoutAction property to "Redirect."

e. Set the LoginStatus control's LogoutPageUrl property to "~/Default.aspx."

f. Launch Default.aspx again and navigate to Admin.aspx.

g. Click "Logout" and verify that you return to Default.aspx. Also verify that the LoginStatus control reads "Login."

h. Log in and go back to Admin.aspx.

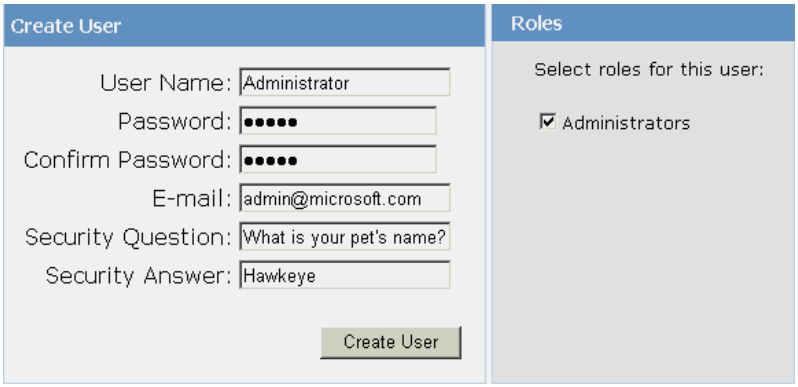
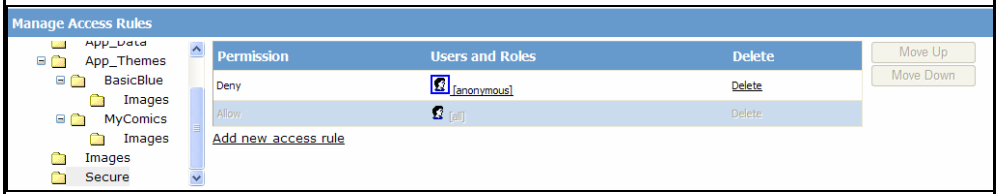
i. Click the MyComics logo in the upper-left corner of the page. Verify that you go back to Default.aspx and that the LoginStatus control reads "Logout" rather than "Login."

j. Close the browser and return to Visual Studio.

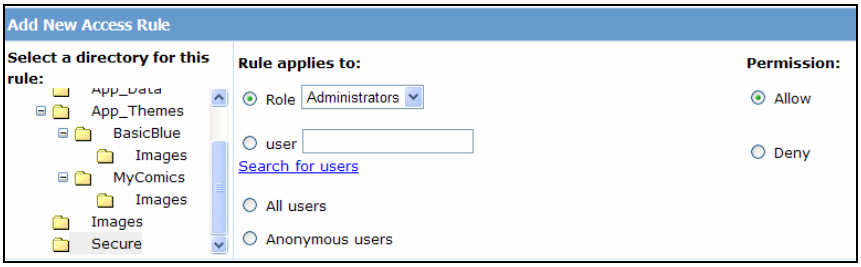
Exercise 4

Enable role-based security

In this exercise, you'll enable ASP.NET's role manager and configure it to use the SQL Server provider. You'll also create an account named Administrator and a group named Administrators and assign the Administrator account to the Administrators group. Then you'll modify the site's security settings so that only Administrators can view Admin.aspx.

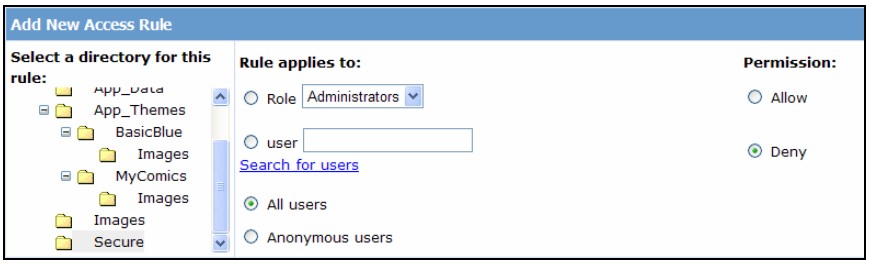
Tasks	Detailed Steps
1. Enable the ASP.NET role manager	**a.** Use Visual Studio's "Website->ASP.NET Configuration" command to launch the Web Site Administration Tool. **b.** Go to the Web Site Administration Tool's Security page. **c.** Click "Enable roles." **d.** Open Web.config. Can you spot the change that was just made?
2. Create an Administrators group	**e.** Click "Create or Manage roles" on the Web Site Administration Tool's Security page. **f.** Type "Administrators" and click the "Add Role" button. **g.** Click the Back button in the lower right corner of the page to return to the main Security page.
3. Create an Administrator account	**a.** Click "Create user" on the Web Site Administration Tool's Security page. **b.** Fill in the form as shown below. Be sure to check the Administrators box so the new user—Administrator—will be added to the Administrators group. Use the password of "P@ssw0rd".  **c.** Click the "Create User" button. **d.** Click the Back button to return to the main Security page.
4. Make Admin.aspx off limits to nonadministrators	**a.** Click "Manage access rules" on the Web Site Administration Tool's Security page. **b.** Select the Secure folder as shown below. Then click the Delete button at right to delete the rule that denies anonymous users access to the Secure folder. Answer Yes when prompted for confirmation.  **c.** Click "Add new access rule." **d.** Fill in the Add New Access Rule form as shown below to add a rule allowing administrators access to the Secure folder. Be SURE to select the Secure folder in

the tree view at left before clicking OK.



e. Click "Add new access rule" again.

f. Fill in the Add New Access Rule form as shown below to add a rule denying all users access to the Secure folder. Once more, be SURE to select the Secure folder in the tree view at left before clicking OK.



g. Open the Web.config file in the Secure folder and inspect the <authorization> element. What changes did the Web Site Administration Tool make?

h. Run Default.aspx and click the Admin link in the upper-right corner.

i. Log in using the user name and password that you registered in Exercise 3. What happens?

j. Try logging in again, but this time log in as Administrator. Verify that Admin.aspx appears.

Summary

Here's a recap of what you learned in this lab:

- How to build a login page around Login and CreateUserWizard controls
- How to create an ASP.NET database for storing membership data, role data, and more
- How to use the Web Site Administration Tool to select membership and role providers
- How to use the Web Site Administration Tool to enable forms authentication and apply basic security settings
- How to programmatically converse with a master page from a content page
- How to use LoginStatus controls to display links for logging in and out
- How to change the logout action of a LoginStatus control
- How to enable the ASP.NET role management service
- How to create roles and assign users to roles
- How to create role-based access rules for your Web site

Take a moment to review the application in its current form. As you do, here are some questions to ponder:

- *Did securing the admin page require a significant amount of code?*

- *If you wanted to modify the master page to show login names for authenticated users, how would you go about it? What control would you use to display login names?*

- *If you decided to move the ASP.NET database to a remote server, how would you go about it? Would any code changes be required?*