Walkthrough: Creating a Web Site with Membership and User Login

.NET Framework 4 69 out of 91 rated this helpful

A typical requirement for Web sites is to allow only some users (authenticated users) to see certain pages. In that case, it is typical for the Web site to provide a way for users to log in and be authenticated, and to hide information from anonymous users (users who are not logged in).

> **Note**
>
> If you create a web application in Microsoft Visual Studio 2010 using a Web project template, the template includes pages that implement basic login functionality using login controls and ASP.NET membership. However, the template does not include all the functionality that is described in this walkthrough. If the template does not include the functionality you want, or if you prefer to create the membership system yourself, you can use the same ASP.NET controls that are in the template to create custom login and membership pages.
> This walkthrough shows you how to manually put together these ASP.NET controls and ASP.NET membership services to create an application that authenticates users and that hides information from anonymous users. For information about how to use the Web site project templates, see Walkthrough: Creating an ASP.NET Web Site with Basic User Login.

Tasks illustrated in this walkthrough include the following:

- Creating a Web site.
- Creating a home page that is accessible to all users.
- Creating a login page.
- Creating a members-only page.
- Configuring membership.
- Adding new users (members).
- Letting users change their password.
- Letting users reset a forgotten password.

Prerequisites

In order to complete this walkthrough, you will need the following:

- Visual Studio 2010 or Visual Web Developer 2010 Express installed on your computer.
- SQL Server Express installed locally on your computer.

- Access to an email server that can forward email messages. (The server does not have to be able to receive messages.) If you do not have access to an email server, you can run the procedures in this walkthrough, but you will not be able to use the password-recovery feature.

Creating the Web Site

To create custom user login pages, you will first create an empty Web site.

This walkthrough uses a Web site project. You could use a Web application project instead. For information about the difference between these Web project types, see Web Application Projects versus Web Site Projects.

To create a Web site

1. Start Visual Studio 2010 or Visual Web Developer 2010 Express.
2. On the **File** menu, click **New Web Site**. (If you do not see this option, click **New**, and then click **Web Site**.)

   The **New Web Site** dialog box is displayed.
3. Under **Installed Templates**, click **Visual Basic** or **Visual C#** and then select **ASP.NET Empty Web Site**.

   You are selecting an Empty Web Site template because you will add membership and login capability by hand rather than using the features built into the other templates.
4. In the **Web location** list box, select **File System** and enter the name of the folder where you want to keep the pages of the Web site. For example, enter the folder name **C:\Websites\Membership** and then click **OK**.

Visual Studio creates an empty Web site that contains only a Web.config file.

Creating the Home Page

You will first create the home page where all users of the Web site will start.

To create a home page

1. In **Solution Explorer**, right-click the name of the Web site project and then click **Add New Item**.

   The **Add New Item** dialog box is displayed.
2. If you do not see the **Solution Explorer** window, select the **View** menu and then click **Solution Explorer**.
3. Select **Web Form**, accept the default name (Default.aspx), and then click **Add**.
4. Switch to **Design** view and add text such as **My Home Page**.

5. In the **Formatting** toolbar, use the **Block Format** drop-down list to format the text as **Heading 1**.

Creating the Login Page

As part of your Web site, you must establish the user's identity (authenticate the user) so that the Web site can perform actions based on who the user is, such as showing or hiding information. To get the user's identity, you must have the user log in. Therefore, your Web site must include a login page.

To create a login page

1. In **Solution Explorer**, right-click the name of the Web site project and then click **Add New Item**.
2. Select **Web Form**, change the default name to **Login.aspx**, and then click **Add**.

Note
For this walkthrough, the page must be named Login.aspx. By default, ASP.NET authentication is configured to work with a page that has this name. (Although you will not do so in this walkthrough, you can change the default login page in the Web.config file.)

3. In the Login.aspx page, switch to **Design** view.
4. Add text such as **Login Page** to the page and again use **Block Format** to format the title as **Heading 1**.
5. Press ENTER to create a new paragraph.
6. From the **Login** group of the **Toolbox**, drag a Login control to the new paragraph.

   The Login control is a single control that will prompt the user for credentials and validate them. If you want, you can use the **Auto Format** link on the smart tag panel to apply formatting to the Login control.

Displaying Detailed Login Errors

The **Login** control includes validation to help users enter correct information. For example, if a user skips the password, a validator control displays an asterisk (*) next to the **Password** text box. To provide more detail about errors in the page, you will add a ValidationSummary control.

To display detailed login errors

1. From the **Validation** group of the **Toolbox**, drag a ValidationSummary control onto the Login.aspx page. Drop the ValidationSummary under the Login control.
2. In the **Properties** window for the ValidationSummary control, set the ValidationGroup property to **Login1**, which is the ID of the Login control that you added previously.

   (If you do not see the **Properties** window, from the **View** Menu, click **Properties Window**.)

3. Save the page and close it.

You can now test the login page.

You can now test the login page.

To test the login page

1. In **Solution Explorer**, right-click the Login.aspx page and select **View in Browser**.
2. Without entering anything into the login control, click **Log In**.

   An asterisk is displayed next to the **User Name** box and next to the **Password** box, because you did not enter values for these values. In addition, the ValidationSummary control displays error messages that provide details about the errors on the page.
3. Close the browser.

Displaying Information for Logged-In Users

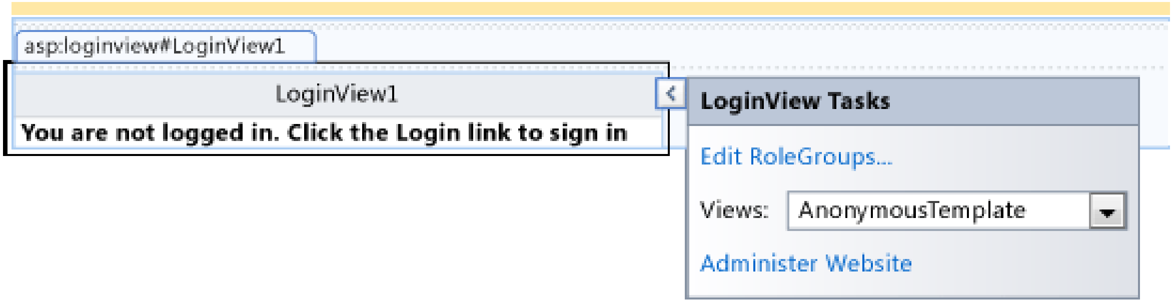
The next step is to modify the home page to customize the display depending on whether the user is logged in. Anonymous users will see a generic message that invites them to log in. Logged-in users will see a message that welcomes them by their user name.

To customize the display for logged-in users

1. Switch to or open the Default.aspx page and switch to **Design** view.
2. From the **Login** group of the **Toolbox**, drag a LoginView control onto the page.

   The LoginView control is displayed with its **AnonymousTemplate** template open. This template allows you to define the content that users will see before they log in.
3. Click the edit area of the LoginView control to activate editing.
4. In the edit area of the LoginView control's **AnonymousTemplate** template, enter **You are not logged in. Click the Login link to sign in.**.

   The following illustration shows the LoginView control edit area that contains the text.

   You are not logged in. Click the Login link to sign in

   Views: AnonymousTemplate

   It might be easier to copy the text and paste it into the edit area or add the text as markup in **Source** view than entering the text directly into the edit area. The following example shows how the text appears in markup.

```
<asp:LoginView ID="LoginView1" runat="server" >
      <AnonymousTemplate>
          You are not logged in. Click the Login link to sign in.
       </AnonymousTemplate>
</asp:LoginView>
```

5. In **Design** view, select the LoginView control, and then on the **LoginView Tasks** panel, in the **Views** list, click **LoggedInTemplate**. If you do not see the **LoginView Tasks** panel, right-click the heading of the LoginView control and then click **Show Smart Tag**.

   The **LoggedInTemplate** defines the content that will be displayed to users who have logged in.

6. Click the edit area of the LoginView control to activate editing and then enter the following text: **You are logged in. Welcome,**

7. Place the cursor in the edit area of the **LoggedInTemplate** after the text you just added. From the **Login** group of the **Toolbox**, double-click the LoginName control. The **LoginName** control displays the name of the user who is logged in.

8. From the **Login** group of the **Toolbox**, drag a LoginStatus control into the page. The LoginStatus control displays a **Login** link when the user is not logged in. When the user is logged in, the control displays a **Logout** hyperlink.

9. Save the page.

Creating the Members-Only Page

A typical task in many Web sites is to allow only logged-in users to view certain pages. Therefore, the next task for this walkthrough is to create a folder for members-only pages, add a page to it, and then create a rule that limits access to pages in the folder.

To add a new folder to the Web site

1. In **Solution Explorer**, right-click the name of the Web site and then click **New Folder**.

2. Name the folder **MemberPages**.

3. In **Solution Explorer**, right-click the MemberPages folder and then click **Add New Item**.

4. Add a new Web Forms page named Members.aspx.

5. Click **Add**.

   Note

   Make sure that you create the page in the MemberPages folder.

6. Switch to **Design** view and add text to the page, such as **Welcome to the members-only page.** The exact text is not important, as long as you will be able to recognize this page when you see it in the browser. Use the **Block Format** drop-down list to format the text as **Heading 1**.

You can now add a hyperlink to the home page that sends users to the members-only page. In a real application, you would probably put the members-only page link in the logged-in template of the LoginView control. That way, visitors to your site would not see the link until they are logged in. However, for this walkthrough, you will make the link available to all users so that you can see the effect of trying to view a members-only page without first logging in.

To link to the members-only page

1. Open or switch to open the Default.aspx page.
2. From the **Standard** group of the **Toolbox**, drag a HyperLink control onto the page.
3. In the **Properties** window for the HyperLink control, do the following:
    a. Set the **Text** property to **Members-only page**.
    b. Set the **NavigateUrl** property to **~/MemberPages/Members.aspx**. This points to the page that you created previously.
4. Save the page.

You can now test the hyperlink you just added to the home page.

To test the link

1. In **Solution Explorer**, right-click the Default.aspx page and then click **View in Browser**.
2. Click **Members-only page**.

   The members-only page is displayed, because it is not protected yet. (Moreover, by default, ASP.NET uses Windows authentication, so it considers you logged in under your Windows credentials.)

3. Close the browser.

Configuring Membership

The next step is to configure ASP.NET membership and set up users. To do this, you can use the Web Site Administration Tool, which provides a wizard-like interface for making configuration settings. When you complete the configuration, a SQL Server database named ASPNETDB.MDF is created in the App_Data folder of the project. The database contains membership information for this Web site.

For this walkthrough, you will define a single user.

To create a membership user

1. On the **Website** menu, click **ASP.NET Configuration**.
2. Select the **Security** tab, click the **Use the security Setup Wizard to configure security step by step** link, and then click **Next**.

   The wizard displays a page where you can select the authentication method that your Web site will use.

3. Select the **From the Internet** option.

   This option specifies that your Web site will use ASP.NET forms authentication, which relies on the ASP.NET membership system. When you use forms authentication, users log in to the Web site by using the login page that you created earlier in this walkthrough. (The **From a local area network** option configures the Web site to use Windows authentication, which is practical if your site is accessed only by people who are on a corporate network. As noted earlier, this is the default setting for ASP.NET membership.)

4. Click **Next**.

   The wizard indicates that the application is configured to use advanced provider settings. By default, membership information is stored in a Microsoft SQL Server database file in the App_Data folder of your Web site.

5. Click **Next**.

6. Clear the **Enable roles for this Web site** check box, and then click **Next**.

   The wizard displays a page where you can create new users.

7. Enter information that defines a user of your Web site. Use the following values as guidelines. (You can use any values that you want, but be sure to note your entries. You will use these entries later in the walkthrough.)

   - **User Name** Your name (without spaces), or a sample name.
   - **Password** A password. A strong password is required (one that includes uppercase and lowercase letters, punctuation, and that is at least eight characters long).
   - **E-mail** Your personal email address. If you have to reset your password, this email address will be used to send you a new password. Therefore, you need to enter a valid email address.
   - **Security Question** and **Security Answer** Enter a question and answer that can be used later if you need to reset your password.

8. Select the **Active User** check box.

9. Click **Create User**.

   The wizard displays a confirmation page.

   > **Note**
   >
   > Leave the Web Site Administration tool open.

Setting Up Access Rules for the Members Page

Earlier in the walkthrough, you created a folder named MemberPages and added a members-only page. In this part of the walkthrough, you will create a rule that makes sure that only logged-in users can access pages in that folder.

To set up access rules

1. In the security wizard in the Web Site Administration Tool, click **Next**.

   The wizard displays the **Add New Access Rules** page.

2. In the **Add New Access Rule** box, expand the node for your Web site.

3. Select **MemberPages**, the folder that you created earlier.

3. Select **MemberPages**, the folder that you created earlier.

4. Under **Rule applies to**, select **Anonymous Users**.

5. Under **Permission**, select **Deny**.

   The rule you are creating denies access to anonymous users — that is, users who have not logged in.

6. Click **Add This Rule**.

   The new rule is displayed in the grid. When users request a page from the MemberPages folder, the rules are checked to determine whether the user is allowed access to the page.

7. Click **Finish**.

You are now done with the wizard. The wizard closes and you are returned to the **Security** tab of the Web Site Administration Tool.

Testing the Members-Only Page

You can test the members-only page by accessing it as an anonymous user or as a logged-in user.

To test the members-only page

1. In Visual Studio, switch to the Default.aspx page and press CTRL+F5 to run the Web site.

2. When the Default.aspx page is displayed in the browser, do not log in. Instead, click the **Members-only page** link.

   The Login.aspx page is displayed, because access to the member's page is denied for anonymous users.

3. On the login page, enter the user name and password that you used earlier to create the user to log in.

   When you log in, the site redirects you to the members-only page, because you are no longer an anonymous user.

4. Close the browser window.

Adding New Users

Earlier in the walkthrough, you created a user by using the Web Site Administration Tool. The Web Site Administration Tool is useful if you are working with a small, defined list of users, such as creating users for a small team. However, in many Web sites, users can register themselves. To support this feature, ASP.NET includes the CreateUserWizard control that performs the same task you performed earlier by using the Web Site Administration Tool.

Creating a Registration Page

In this part of the walkthrough, you will add a functionality that allows users to register on your Web site. You will start by creating a registration page.

To create a registration page

To create a registration page

1. In **Solution Explorer**, right-click the name of the Web site project and then click **Add New Item**.

2. Add a new **Web Form** named **Register.aspx** and click **Add**.

   > **Note**
   >
   > Make sure that you create the page in the root of the Web site, not in the MemberPages folder.

3. In the Register.aspx page, switch to **Design** view and enter text such as **Register** into the page. In the **Formatting** toolbar, use the **Block Format** drop-down list to format the text as **Heading 1**.

4. From the **Login** group of the **Toolbox**, drag a CreateUserWizard control onto the page.

5. In the **Properties** window for the CreateUserWizard control, set the ContinueDestinationPageUrl property to **~/Default.aspx**.

   This configures the control so that when users click **Continue** after they create a user, they are returned to the home page.

6. From the **Standard** group of the **Toolbox**, drag a HyperLink control onto the page.

7. In the **Properties** window for the HyperLink control, do the following:

   - Set the **Text** property to **Home**.

   - Set the NavigateUrl property to **~/Default.aspx**.

8. Save and close the page.

You will now add a link that displays the registration page to both the home page and the Login page. For this walkthrough, assume that you want to display the registration link only to users who are not logged in.

To create a registration link on the home page

1. Switch to or open the Default.aspx page.

2. Right-click the LoginView control that you added previously, and then click **Show Smart Tag**.

3. In the **LoginView Tasks** panel, select **AnonymousTemplate** from the **Views** list in order to activate editing in the anonymous template.

4. From the **Standard** group of the **Toolbox**, drag a HyperLink control into the anonymous template. It might be easier to put the cursor in the editing area and double-click the HyperLink control in the **Toolbox**.

5. In the **Properties** window for the HyperLink control, do the following:

   a. Set the **Text** property to **Register**.

   b. Set the **NavigateUrl** property to **Register.aspx**.

6. Switch to or open the Login.aspx page.

7. From the **Standard** group of the **Toolbox**, drag a HyperLinkcontrol into the page.

8. In the **Properties** window for the HyperLinkcontrol, do the following:

   a. Set the **Text** property to **Register**.

   b. Set the **NavigateUrl** property to Register.aspx.

   You can now test the registration process.

To test registration

1. Switch to the Default.aspx page and press CTRL+F5 to run the Web site and display the Default.aspx page.

   Because you are not logged in, the **Register** link is displayed.

2. Click the **Register** link. The registration page is displayed.

3. In the boxes, enter a new user name, a strong password, an email address, and a security question and answer. (All the pieces of information are required.)

4. Click **Create User**. A confirmation message is displayed.

5. Click **Continue**.

   You are returned to the home page as a logged-in user. Note that the **Login** link has changed to **Logout** and that the information displayed in the Login control is from the **LoggedInTemplate** property, not from the **AnonymousTemplate** property.

6. Click the **Logout** link.The page changes to display the information for anonymous users.

7. Click the **Login** link.

8. Enter the credentials for the user you just created.You are logged in as the new user.

9. Close the browser window.

Letting Users Change Their Password

Users sometimes want to change their passwords, and it is often impractical to perform this task by hand. You can use another ASP.NET control to let users change passwords on their own. To change a password, users must know their existing password.

You will add a page where logged-in users can change their password.

To create a password-change page

1. In **Solution Explorer**, right-click the MemberPages folder, click **Add New Item**, add a new **Web Form**, name it **ChangePassword.aspx** and then click **Add**.

   Make sure that you create the page in the MemberPages folder. You are putting the page in the MemberPages folder because only logged-in users can change their passwords.

2. In the ChangePassword.aspx page, switch to **Design** view and enter text such as **Change Password** and format it using **Heading 1**.

3. From the **Login** group of the **Toolbox**, drag a ChangePassword control onto the page.

4. In the **Properties** window for the ChangePassword control, set the **ContinueDestinationPageUrl** property to **~/Default.aspx**.

   This configures the control so that when users click **Continue** after they change a password, they are redirected to the home page.

5. Save and close the page.

Creating a Password-Change Link on the Home Page

You can now add a link to the home page that displays the ChangePassword.aspx page. You will make the link available only to users who are logged in.

To create a password-change link on the home page

1. Switch to or open the Default.aspx page.

2. Right-click the LoginView control and then click **Show Smart Tag**.

3. In the **LoginView Tasks** pane, in the **Views** list, click **LoggedInTemplate**.

   This switches the LoginView control to edit mode for the content that will be displayed to users who are logged in.

4. From the **Standard** group of the **Toolbox**, drag a HyperLink control into the editing region.

5. In the **Properties** window for the HyperLink control, do the following:

   a. Set the **Text** property to **Change password**.

   b. Set the **NavigateUrl** property to **~/MemberPages/ChangePassword.aspx**.

Testing the Password-Change Page

You can now test the password-change process.

To test the password-change page

1. Press CTRL+F5 to run the Web site.

2. In the Default.aspx, page, click the **Login** link and log in as one of the users you have created.

   When you are finished, you are returned to the home page as a logged-in user.

3. Click the **Change password** link.

4. In the password-change page, enter the old password and a new password, and then click **Change Password**.

5. Click **Continue**.

6. On the home page, click **Logout**.
7. Click **Login**.
8. Log in with the new password.
9. Close the browser.

Letting Users Reset Their Password

When users forget their password, the Web site can enable them to recover or reset their password. The password can be recovered (that is, sent to the user) if the password is not hashed.

When a password is hashed, the membership system does not store the actual password. Instead, the system processes the password using a one-way algorithm (a hashing algorithm) that produces a unique value for the password, and then stores this hash value. This algorithm can be repeated to test a user's password at login, but cannot be reversed to produce the actual password. This increases the security of the membership database, because getting access to the database does not mean that passwords are exposed.

By default, the membership provider stores the password as a hash value. Therefore, the password cannot be recovered. Instead, if a user has forgotten a password, the Web site must generate a new password and send it to the user in email. For your Web site to send email messages your computer must have access to a Simple Mail Transport Protocol (SMTP) server.

In this procedure, you will add a password-reset page to your Web site and configure the Web site to send users a new password by email. To reset a password, a user must provide a user name and must answer the security question that was provided when the user registered. The new password is sent to the email address that the user provided when registering.

To create the password-reset page

1. In **Solution Explorer**, right-click the name of the Web site project and then click **Add New Item**.
2. Add a new **Web Form** page, name it **PasswordRecovery.aspx** and click **Add**.

   > **Note**
   >
   > Make sure that you create the page in the root of the Web site, not in the MemberPages folder.

3. In the PasswordRecovery.aspx page, switch to **Design** view and enter text such as **Forgot Password** into the page and format it as **Heading 1**.
4. From the **Login** group of the **Toolbox**, drag a PasswordRecovery control onto the page.
5. Open or switch to the Login.aspx page and switch to **Design** view.
6. From the **Standard** group of the **Toolbox**, drag a HyperLink control onto the page.
7. Set the **Text** property to **Forgot password** and the **NavigateUrl** property to **~/PasswordRecovery.aspx**.

Configuring the Web site to Use an SMTP Server

Next, you will configure your Web site to use an SMTP server. To configure the server correctly, you must have the setup information. If the server requires authentication, you will need the user name and password. For information about how to obtain this setup information, contact the system administrator. After you have determined how to access the SMTP server, you must configure your Web site to route email messages to that server. You can do so in the Web Site Administration Tool or by making an entry in your Web site's Web.config file, which contains a series of settings that determine how your application runs. The following procedure shows how to use the ASP.NET Web Site Administration Tool to perform this task.

To configure the Web site to use an SMTP server

1. On the **Website** menu, click **ASP.NET Configuration**.

2. In the Web Site Administration Tool, click the **Application** tab.

3. Under **SMTP Settings**, click **Configure SMTP email settings**. The tool displays a page where you can configure email.

4. Enter the information that the page prompts you for.

5. Click **Save**, and in the confirmation page, click **OK**.

   The Web Site Administration Tool creates a Web.config file that has the settings that you have made in the **mailSettings** section.

6. Close the browser window in which the Web Site Administration tool is displayed.

7. Open the Web.config file.

8. Under the **system.net** element and under the **mailSettings** element, verify your **smtp** and **host** settings.

   Note

   If your SMTP server requires a secure connection, you must set the **enableSsl** attribute to **true**. You do this in the **network** section of the **smtp** element in the Web.config file.

   The SMTP settings in the Web.config file will resemble the following example.

```
<system.net>
  <mailSettings>
    <smtp from="joe@contoso.com">
        <network host="<server>" password="<password>"
         userName="joe@contoso.com" enableSsl="true" />
    </smtp>
  </mailSettings>
</system.net>
```

   Security Note

   To protect configuration information like the user name and password, you can have ASP.NET encrypt the portion of the Web.config file where the SMTP information is stored. For more information, see Encrypting Configuration Information Using Protected Configuration. In addition, by default ASP.NET sends the **smtp** server credentials in clear text and can be intercepted by programs that record network activity. For a production site, you should use the SSL (secure sockets layer) to encrypt sensitive information that is exchanged with the server.

Testing Password Reset

You can now test the process of resetting your password. You will then use the new password to log in.

To test the password-reset page

1. Press CTRL+F5 to run the Web site.
2. Click **Login**.
3. In the Login page, click the **Forgot password** link. Enter your user name and then click **Submit**.
4. Enter the answer to the security question and then click **Submit**.
5. Wait for a few minutes and then check your email.
6. Use the new password to log in.

Next Steps

This walkthrough has illustrated a simple but complete scenario for creating an application that prompts users for credentials, displays information to logged-in users, limits access to pages, and lets users reset a forgotten password.

You can create more sophisticated pages and applications by using the techniques and controls illustrated in the walkthrough. For example, you might want to do the following:

- Create additional users and define roles (groups) such as managers, sales, or members and assign users to different roles. For details, see Walkthrough: Managing Web Site Users with Roles.
- Change the appearance of login controls. The Login, PasswordRecovery, and CreateUserWizard controls all support templates, which lets you configure and auto-format the text and buttons they contain.
- Combine membership with profile properties, which lets you store user-specific settings for each user. For details, see Walkthrough: Maintaining Web Site User Information with Profile Properties.
- Add login controls to ASP.NET master pages. Master pages let you define a page layout that you can use for all the pages in your application. For details, see Walkthrough: Creating and Using ASP.NET Master Pages in Visual Web Developer.

See Also

Concepts

Introduction to Membership

Community Additions

I get "Unable to connect to SQL Server database" Error msg

while going to all this steps for making membership login...in configuring membership i m geting an error showing that "unable to connect to SQL Server Database"

plz help me...

sumit1212
8/12/2013

WebForms UnobtrusiveValidationMode requires a ScriptResourceMapping for 'jquery' error in VS2012

If you're using Visual Studio 2012, you may get an error when testing the login page:-

WebForms UnobtrusiveValidationMode requires a ScriptResourceMapping for 'jquery'

You need to add the following to web.config:-

```
  ...
  <appSettings>
      <add key="ValidationSettings:UnobtrusiveValidationMode" value="None" />
   </appSettings>
</configuration>
```

Details can be found in this article: http://www.codeproject.com/Articles/465613/WebForms-UnobtrusiveValidationMode-requires-a

Sneaky Fruit
3/18/2013

No details on aspnet_regsql command-line utility

Excellent tutorial! One big problem though... In section 'Configuration Membership', when creating a membership user, the ASP.NET Configuration tells me "Could not establish a connection to the database. If you have not yet created the SQL Server database, exit the Web Site Administration tool, use the aspnet_regsql command-line utility to create and configure the database, and then return to this tool to set the provider."

No details are given as to where to find the aspnet_regsql.exe and no details are given about what to place into the web.config file to connect to the DB (i.e. connectionStrings).

For those of you with this problem, check out the second and sixth post in the following forum topic: http://forums.asp.net/t/1199478.aspx/1

paddyirishman05
12/18/2012

i

k

bobble14988
7/27/2012

Thanks for the interesting and useful tutorial!

Many thanks for the tutorial, it has been very useful for me in learning the basics of ASP.NET!

Also, many many thanks to J.W.Clark for his tips on altering the machine.config file to point to your SQL Server instance instead of SQLEXPRESS. I was getting the 'Unable to connect to SQL Database' problem when using the ASP.NET config tool: amending the config files as per these instructions has fixed the issue (I had to do it in the .net v4 folder but it is the same change).

Matt FTW
6/25/2012

Visual Studio 2010 ASP.NET Web Administration Tool

To find the ASP.NET Web Administration Tool. You must go to the **Menu**, then "**Project**" -> "**ASP.NET Configuration**"

ran2k9
3/31/2012

Password recovery and SMTP Failing

Hi,
I tried that walkthrough, and found it very interesting until I had to use the recovery password feature which uses the SMTP mail service.
That part of the application uses the ASP.NET Configuration command in the Website menu. One is told to configure the SMTP e-mail settings.
There is not much information about this in the walkthrough. I would appreciate some help about this problem.
I use VS2010, the .NET Framework 4, and Windows 7.
For the SMTP settings, I used my computer name as the server name. Is that ok?
The port is 25 by default. Is that correct?

The port is 25 by default. Is that correct?
In the From:, I entered my email address.
For the authentication, I tried two of the three options, None, and NTLM (Windows authentication).
No matter what I try, I always get the same error message:
ERROR:
Exception Details: System.Net.Sockets.SocketException: Aucune
connexion n'a pu être établie car l'ordinateur cible l'a expressément
refusée [2001:0:4137:9e76:188d:a6f:3f57:fdf5]:25
That means: No connection could be established because the destination
computer rejected it.

I wander what is wrong. I would like to know where to get some information about that problem.
I know that other people have encountered the same problem and never found any solution. I would like to change that.

Another question: how do I know that somone has answered my post?
Thanks in advance.

Note: I tried the forums but it is hard to find the correct one to post to and do not expect any better answer from those.
Jacs

JacSophie
2/6/2012

Minor error in the tutorial

In step no 5 of "Letting user reset their password", the file to be changed to is NOT login.aspx but Default.aspx.

In all the preceding text, the file has always been referred to as "Default.aspx" except in this instance.

Uttam

Dear Uttam,
I think that you are wrong. It is the Login.aspx page that has to host the link to the ChangePassword page.
Sorry, but it works for me.
Jac

JacSophie
2/6/2012

compilier error message

I have tried this tutorial twice and can only get as far as testing the link to the Member-only page and unless i`m repeating my mistake I still get the same error
"Compilier error message: CS1061:`ASP.default_aspx` does not contain a defination for 'LoginView1_changed' and no extension method 'LoginView1_ViewChanged' accepting a first argument of type 'ASP.default_aspx' could be found(are you missing a directive or an assembly reference?) "
as far as I can see I have done everything verbatim and sqlce is up and running. What am I missing ?

Cheiftain
1/28/2012

Password Recovery and SSL SMTP failing

As far as I can tell the problem has not been fixed since ASP.NET 2.0 - Password recovery fails to establish a SSL connection with SMTP servers.

To get around this I installed code from this blog and mails are sent:

http://blogs.msdn.com/b/vikas/archive/2008/04/29/bug-asp-net-2-0-passwordrecovery-web-control-cannot-send-emails-to-ssl-enabled-smtp-servers.aspx

MaxDaniels
12/29/2011

I get "Unable to connect to SQL Server database" Error msg

When I got to the Web Administration Tool, I get this error message

"There is a problem with your selected data store. This can be caused by an invalid server name or credentials, or by insufficient permission. It can also be caused by the role manager feature not being enabled. Click the button below to be redirected to a page where you can choose a new data store.

The following message may help in diagnosing the problem: Unable to connect to SQL Server database.

"

I wonder what can cause this. PLEASE NOTE: On other pages apart from login.aspx, my application can interact with the SQL database properly, displaying data from the database as well as modifying data with no problem. Its only when I get to this Web Administration Tool that I get this error. I use Visual Studio 2010 and SQL Server Enterprise. PLEASE HELP ME, how do I get out this and get the members thing to work?

Thanks in advance,

Frank.

savedlema
11/24/2011

Login not authenticated

I followed the exact tutorial, and seems like i am stuck at the login part.

from default page, I clicked on to the member's page and it prompted for user/password. But even when I enter it, it is rejecting with "Your login attempt was not successful. Please try again".

I am pretty sure about the user/password. I am new to asp and sql..

I am runnning sql server 2008 and visual 2008. Please help me.

Thanks

Debian

Debian_asp
11/9/2011

Provide code for Oracle database

Please provide sample code that can be used with a Oracle database.

kms123
9/13/2011

Login Issues

I understand the everything you have written but, as a new user, I have a problem can anyone assist (no complex technical jargon please :) I have created a site from Scratch with user login features etc and I have created various pages that can only be accessed by certain users. I have assigned all of the roles and access permissions, however, I don't know how to redirect my users to the group of pages that they have permission to view. I haven't used the Asp.NET Web Site template, simply built the website following the walkthrough. My site is for my school. Each student has an account and can only access the pages for their grade. So I have folders for each grade and the pages accessible to only those in that grade are stored in this folder. All students firstly Log in, then they are directed to the Homepage. I have a link on the homepage called 'Go to your Account' can anyone please advise me on how to set the NavigationURL so that each student is sent to the correct page??????? Thank you in advance.....(I have a headache!!)

Leigh_01
7/6/2011

Recovery password error send email

Hi,

I've got error when test recovery password feature like below :

The SMTP server requires a secure connection or the client was not authenticated. The server response was: 5.5.1 Authentication Required. Learn more at
can you give me a suggest for this error

thanks
kamal

MuhammadKamal
4/22/2011

Query about "CreateUserWizard" control

when we use CreateUserWizard control in our website then where is the authenticated user record is stored and how can we view it as a whole ??

Anubhav Tiwari
2/26/2011

2/26/2011

It is possible to do this using a different version of SQL Server.

Portions of this were authored by a
Microsoft MVP. Unfortunately, I did not take notes on his name.

The Web Site Administration Tool is also a web project which by default uses the “LocalSqlServer” as its connection string. It is independent of the connection string defined in your developing web project. “LocalSqlServer” by default uses SQL Server Express User instance connection string so that it always fails on the machine which doesn’t installed SQL Server 2005 Express Edition.

Hence, please overwrite the connection string of “LocalSqlServer” in machine.config. For example:

```
<connectionStrings>
```

Take out this default line which was installed by Visual Studio:

```
<add
name="LocalSqlServer" connectionString="data
source=.\SQLEXPRESS;Integrated
Security=SSPI;AttachDBFilename=|DataDirectory|aspnetdb.mdf;User Instance=false"
providerName="System.Data.SqlClient" />
```

Replace it with this line to match the instance of SQL Server you want to use:

```
<add name="LocalSqlServer"
connectionString="Data Source=serverName;Integrated
Security=True;Database=aspnetdb"
providerName="System.Data.SqlClient" />
```

</connectionStrings>"

In the replacement connection
string, the "Database=aspnetdb" references the ASP default schema
which is installed via aspnet_regsql.exe utility using default settings. The
utility in its default installation is found at C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\aspnet_regsql.exe.

For a default installation of SQL
Server 2008 R2, machine.config is located at
C:\Windows\Microsoft.NET\Framework\v2.0.50727\CONFIG\machine.config.

J.W. Clark
12/12/2010

View Users

Is there a way to add a form on your project that will display a form filled with a list of all registered users? Not sure where this information is stored or how to access it. Would use this for instance if an admin user wanted to see a list of all registered users for his site.

Dude88242
11/25/2010

Is there a way to do this with SQL Server and not SQL Server Express?

One of the requirements for this Walkthrough is to use SQL Server Express. I already have SQL Server 2008 R2 installed. Can I use this instead?

I think you can. you need to use aspnet_regsql.exe application in .net folder.
-suraj

Suraj Shrestha
11/24/2010

Andyes

nice tutorial

JtoheLzzo
11/23/2010

Good Tutorial, gives you the basics

not bad

JtoheLzzo
11/23/2010

Work Around for the “Failed to generate a user instance of SQL Server” Issue

This documentation is being modified to address the issue.

However, you can resolve the issue by following these steps:

1. Open **Internet Information Services** (IIS) Manager.

2. Under **Connections** Select **Application Pools**.

3. In the **Application pools** dialog box select **DefaultAppPool**.

4. Under **Actions**, click **Advanced Settings**.

5. Under **Process Model**, Click **Identity Field** to open the Application Pool Identity dialog box.

6. Select **Built-in Account** and then Select **LocalSystem**.

Do above steps for ASP.NET v4.0 App Pool also. Mine worked after I did this.

Dickran
8/10/2010

Typical Microsoft BS

The tutorial crashes as soon as a valid password is entered with :

"

SqlException (0x80131904): Login failed for user 'DEV2\ASPNET'.]
System.Data.SqlClient.SqlInternalConnection.OnError(

"

not further explanation .

No explanation of why you have to use the garbage sql server that is included with with visual studio for security .

No explanation of what the sql is that crashes the app.

Time wasters thank you !!!

annoyed_muther
7/13/2010

Really nice membership tutorial

The tutorial worked great on my local computer, but when I uploaded it to my webserver and I get an error 26 when I try to access SQL. I cannot find the SQL connections settings.

tfl - 29 06 10] Hi - and thanks for your post. You should post questions like this to the MSDN Forums at http://forums.microsoft.com/msdn or the MSDN Newsgroups at http://www.microsoft.com/communities/newsgroups/en-us/. You are much more likely get a quicker response using the forums than through the Community Content. For specific help about:

Visual Studio : http://groups.google.com/groups/dir?sel=usenet%3Dmicrosoft.public.vstudio%2C&
SQL Server   : http://groups.google.com/groups/dir?sel=usenet%3Dmicrosoft.public.sqlserver%2C&
.NET Framework : http://groups.google.com/groups/dir?sel=usenet%3Dmicrosoft.public.dotnet.framework
PowerShell   : http://groups.google.com/group/microsoft.public.windows.powershell/topics?pli=1
All Public   : http://groups.google.com/groups/dir?sel=usenet%3Dmicrosoft.public%2C&

Thomas Lee
6/29/2010