Record Store User authentication

User authentication was added to Dreamweaver in Dreamweaver MX 2004 on the PHP platform, but the user interface for user authentication may be slightly different than described in this article. To access the **user authentication server behaviors** in **MX 2004 and 8, select plus (+) > User Authentication in the Server Behaviors panel.**

If you are using Dreamweaver MX, you will need Macromedia MX Developer Resource Kit 2. Unfortunately Macromedia MX Developer Resource Kit 2 is no longer available at Adobe website, therefore you have two choices:

1. Download Dreamweaver CS4 trial version for 30 days
2. Download Dreamweaver 8 (for this project only- it will take 10-15 minutes to download 62MB) at http://www.maxtips.com/dmw8.zip

The first part of the Record Store tutorial taught you how to use templates to create a PHP site for a small store. You connected to a database and returned records. You created an administrative section where you could add or change CD information. These are the basic functions you would want in any application.

However there were several features missing from that beginning application. Without user authentication, a user could access your administrative pages change the information simply by knowing your administration URL. Also, we did not cover how to give users a **search function** to find CDs. Lastly, the database provided with the tutorial is small. Soon, you would outgrow a single page catalog, and customers would need to be able to have some way to find a particular album.

In this tutorial, you will add **user authentication** through a **login page** for the administrative site portions. Next, you will add a search function that will search for CDs in the database and return those records to the same page as the catalog. Finally, you will learn **how to plan the search functionality** and how to prevent search implementation mistakes.

In this tutorial, you will manually write the code to create efficient pages. This will give you a chance to learn how to use the built-in manual coding features in Dreamweaver. I will guide you through most of the code. If you do find any of the coding too difficult or tedious, you can simply paste the code from the tutorial into your pages and browse application as an end-user.

Creating a user authentication system

You will create a login page where the user enters his/her username and password, add a behavior to each page you wish to protect from unauthorized users, and create a login failure page that will notify a user that his or her login failed and who to contact.

Creating the login page

The login page asks the user to enter his/her user name and password and checks it against the database records to verify the user's access to certain sections. You are going to code a form and then apply the Log In User Server Behavior to the page.

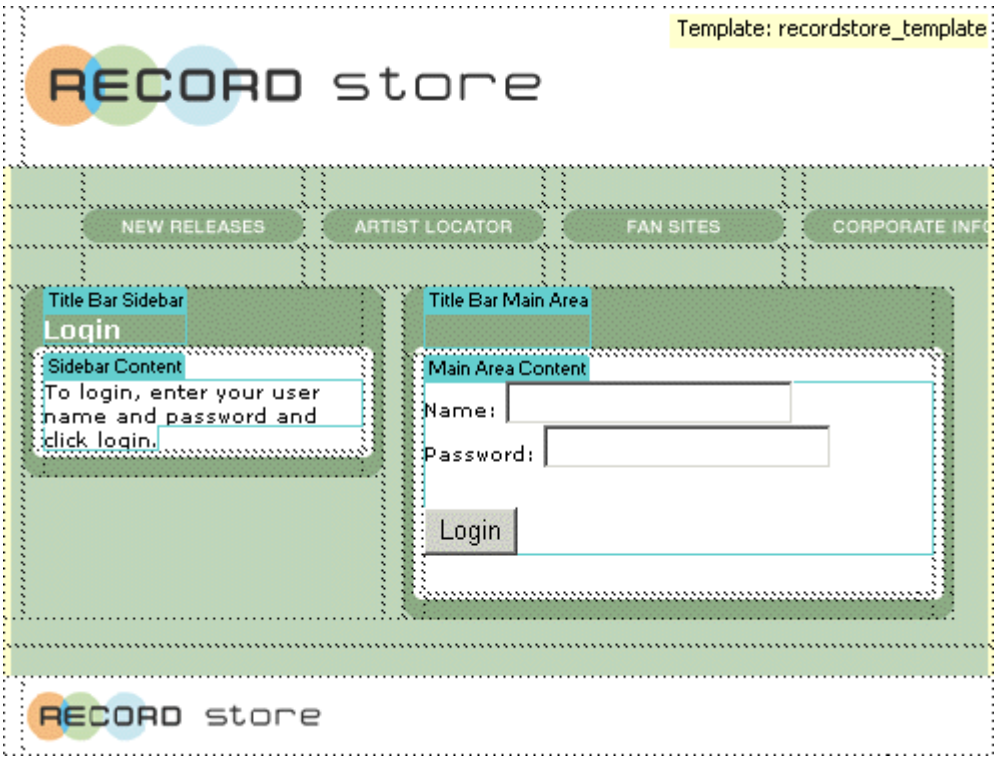
1. Select File > New to create a new page, select the dynamic page category, select PHP from the list of dynamic pages, and click OK.
2. Save the page as "**adminlogin.php**".
3. To ensure the page matches the other pages in your Record Store site, apply the recordstore_template from the Assets panel to the page. To do this, select the Assets tab in the Files panel group. Then, select the record_store template and click the Apply button at the bottom of the Assets panel. If prompted, select the document body region, and click OK to apply the template to the page.
4. On your login page, instruct your users on what to do on the login page. This may seem unnecessary, but always prevents user confusion. I recommend as a best practice that you offer more documentation than less to users. It helps web novices find their way through your website and it will help you maintain the page at a later time.

   Replace the "**Sidebar Content**" with the login instructions. It is also a best practice to include a mail link to a system administrator so that you can easily contact somebody when they need to get a login.

   Change the **Title Bar Sidebar Content** as you wish or delete it.

5. Delete the "**Title Bar Main Area**" text and change the "Title Bar Sidebar" text to "**Login**."
6. **Delete** the "Main Area Content" text.
7. In the Main Area Content, insert a form from the **Forms tab** in the **Insert bar**.
8. In the form you created, insert **two text fields** and a **button**.
9. To define what these text boxes are, we need to click before each text box and type some text to label them.
10. Click before the 1st text box and type: Name
11. Click before the 2nd text box and type: Password
12. You can check the diagram in the next page.
13. Click the **Name** text field. Go to the **Property Inspector Panel**, change the name to "**username**".
14. Click the **Password** text field and change its name to "**password**". For the password field, change the field type to password. Changing the field type to password ensures that no other users can see the password while the user types it; instead, **asterisks** appear as the user types his/her password.

15. Click **Submit** button and change the content(label) to "**Login**" in the Property inspector. Your page will look something like this.



You are now ready to add the **Log In User server behavior** from the set of User Authentication server behaviors.

1. In the Application panel group, select Server Behaviors, then select **User Authentication**, and finally select **Log In User**.
2. In the dialog box for each behavior specify the form that will pass each behavior the login information. Since there is only one form in this case, it will be the default. Dreamweaver will also grab the username and the password fields automatically.
3. Select your connection: conn1
4. For the table, select "**userauthentication**", the table with the usernames and passwords.

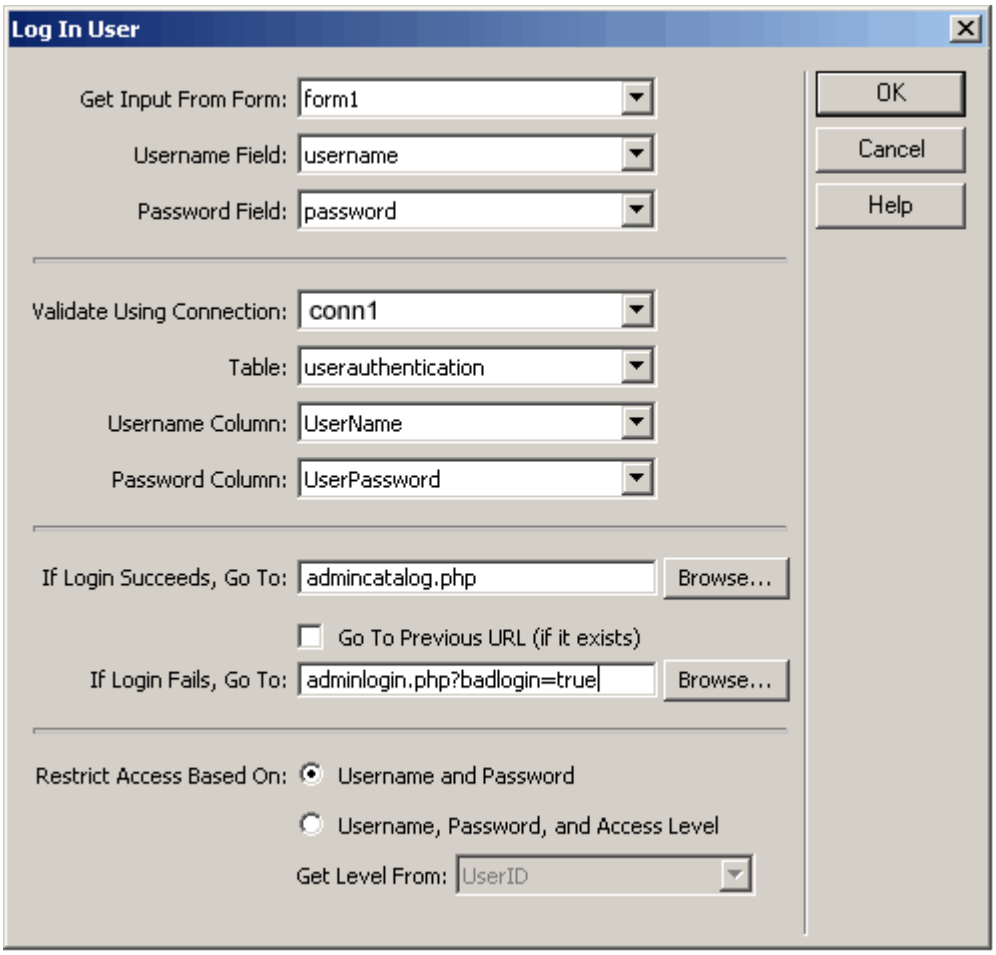
   Select "**UserName**" or the name of the column in the **UserAuthentication** table for the Username Column and "**UserPassword**" for the Password Column.

5. Finally, set the If Login Succeeds, Go To field to "**admincatalog.php**". This is the page the user goes to if he/she logs in successfully. Set the If Login Fails, Go To field to "**adminlogin.php**". This is the login page the user is already on.

   Typically, one problem is that if the user's login fails, the user sees the login page again without any explanation of why the login failed. To fix this, add some information to this page, as described in the next section. First, set a

parameter in the URL of the **adminlogin.php** page.
Add ?badlogin=true to the end of adminlogin.php as you see below:

Your dialog should look like this.



6. **Save the page and upload it to the server.**

Creating the failure page

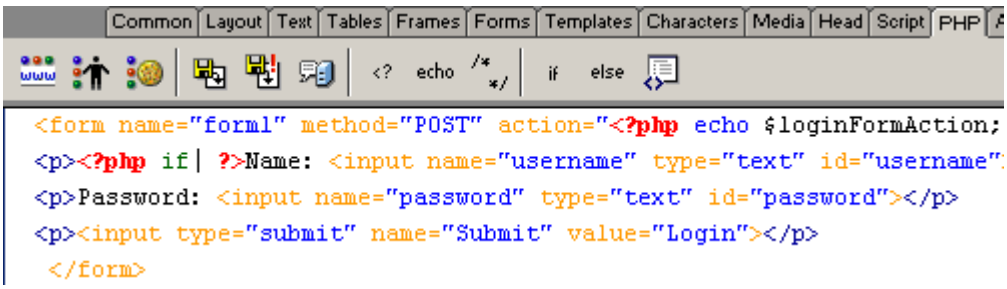
The failure page appears when a user's login does not work. You will build the failure page within the login page using an **IF statement** in PHP. If the URL that calls the login page has a failure parameter attached to it, the login page will display information about the login failure.

If the URL that calls the login page does *not* have a failure parameter attached to it (for instance, first time the user accesses the page there will be no URL parameter), the user will see the normal login form without the additional information.

This means that when the user goes to the login page without having tried to login, the user sees the normal login page. However, if the user goes to the login page after a login failure, he/she will sees the same login page, but with more information so that he/she can try to log on again or contact a system administrator.

1. Click the **adminlogin form**. Move your cursor to the space before the first label for the first text field, and press shift-return to insert a non-breaking space.
2. Click the **PHP tab** in the **insert bar** and click the **IF icon** to insert an `if` conditional into the page. When press the IF icon, the screen will split so you can view the code and design views at the same time.

3. Click the code view button to simply see only the code view.
4. The if statement appears with the cursor waiting for your input. See the figure below.



5. Now you will learn a little bit about PHP language. You will set up a condition so that when the form returns with a badlogin, the page checks for the parameter and displays additional text about the login failure for the user.

6. Ensure your cursor is inside the opening **<?php>** tag, as shown in the figure above and type the code below:

   ```
   (isset($HTTP_GET_VARS["badlogin"])) {
   ```

   Your if statement should now look like this:

   ```
   <?php if (isset($HTTP_GET_VARS["badlogin"])) { ?>
   ```

   This code starts the PHP processor and checks the **HTTP_GET_VARS** which are the parameters attached to the end of the URL that may call the login page. If the "badlogin" parameter exists, the `IF` condition is true, and the code inside the "{" symbol executes. Next, add the end of the `IF` statement. After the end of the "?>", add "<?php } ?>" . Your code should now look like this:

   ```
   <?php if (isset($HTTP_GET_VARS["badlogin"])) { ?><?php } ?>
   ```

7. Add a message between the PHP tags that instructs your user on what to do if his/her login fails. This message may include an administrator's e-mail address or link for the user to contact someone and get a proper login.

   For instance, I added the following between my PHP tags:

   ```
   <p>< font color="red">You have entered a bad password or
   username. Please try again or contact your administrator for
   more details.</font></p>
   ```

   So the whole code block looks like this:

```
<?php if (isset($HTTP_GET_VARS["badlogin"]))
{ ?>
<font color="red">You have entered a bad
password or username. Please contact your
administrator for more details.</font></p>
<?php } ?>
```

8. **Save your page and upload.**

Now your page will show a small message in red font if the user tries to login with an invalid login name or password. If the user has a valid password, the system will send the user to the **admincatalog.php** page.

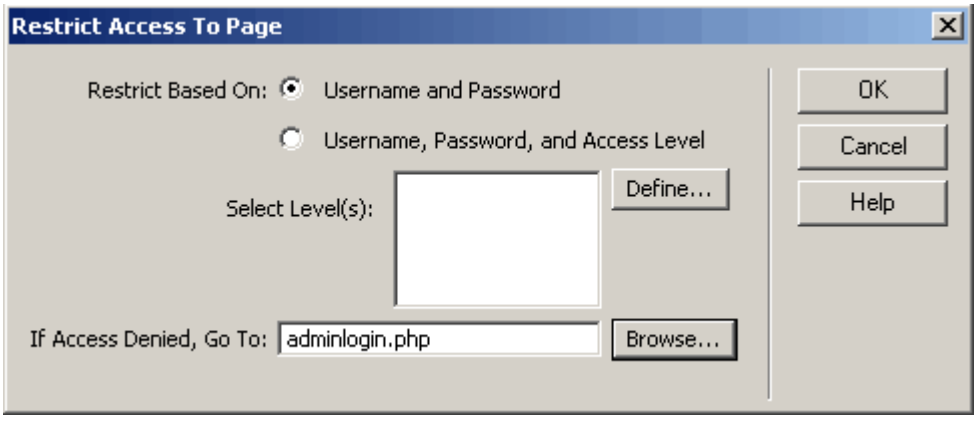
Adding the user authentication scripts to your admin pages

Now that you have built the login page, you need to apply the Restrict Access to Page server behavior to each of the pages that you wish to protect. When a user attempts to access a page with the Restrict Access to Page server behavior, your application will redirect them to the failure page so that they can enter a username/password or contact an administrator. If the user logs in successfully on the adminlogin page, the application will route the user to the admincatalog page. Therefore, you will add the first **Restrict Access to Page** server behavior to the **admincatalog page.**

1. Open the **admincatalog.php** page.
2. Click anywhere in the page and from the **Server Behavior panel**. Insert a Restrict Access to Page server behavior from the **User Authentication server** behavior category.
3. In the **Restrict Access To Page** dialog, restrict access based on the username and password only. You can make specific site sections available to specific groups of people, but we are not using this aspect of the Restrict Access To Page server behavior in this tutorial.

   If the application denies access to a user, you want the application to route the user to the login page so that they can login to view the page properly. So, in the "If Access Denied, Go To:" field, enter "**adminlogin.php**".

   The Restrict Access to Page dialog should look like the following figure:



4. Click OK. Save your file and upload.

Check your website:

Go to:

http://www.maxtips.com/dw/3/adminlogin.php

In the above address, make sure you use your number instead of 3 (account number)

Tip:

You may want to add the Restrict Access to Page behavior to the other admin pages in your site. You may also want to add a Logout User behavior to the protected pages so that users can end their sessions manually if they leave the site. If they don't close the session, but leave their browser open, someone else using the same machine can use the back button to access a protected page. Unless you programmatically set a time limit, most sessions time out automatically after 20 minutes.