

PHP Tutorial 4

Advanced Dynamic Site Features

Authentication

In the last tutorial, you created Web pages that could add, delete, and update records in the *National Exasperator* database. But you wouldn't want to allow just anyone who visits the Web site to access these pages, let alone delete ads from the site. So in this tutorial you're going to learn how to password protect these sensitive, mission critical Web pages.

The following steps assume that you've worked through the last three tutorials, and have all of the completed files ready to go. You'll build on them in the following steps.

Building a Login Page

The first step is to create a login page—a simple form with fields for a user name and password. After a successful login from this page, an administrator will be able to access the administration pages.

1. Open the file *login.php* in the *admin* folder of the site.
This page will contain the form for typing in an administrator's user name and password. You'll add the form next.
2. Click in the empty space directly below the headline "Administrator Login." Choose Insert→Form→Form.
Dreamweaver adds a red dashed line to the page, indicating the beginning and ending <form> tags.
3. In the Property Inspector, type *login* as the name of the form.
While this step isn't required, it's good to get into the habit of assigning your forms descriptive names. Next, you'll add a box for entering a user name.

Note: The next steps assume you have the Form Accessibility feature turned on (see page 367 in the printed book.) To make sure this is in fact the case, open the Preferences window by choosing Edit→Preferences (Dreamweaver→Preferences), select the Accessibility category, and make sure the "Form Objects" checkbox is turned on.

4. Choose Insert→Form→Text Field.

The Input Tag Accessibility Attributes window appears (Figure PHP Tutorial 4-1). You'll add a label that will appear next to the form field on the page.

Note: You can also use the Forms tab of the Insert bar to add forms and form objects to your page as described on page 363.

5. Type *Username:* in the Label box, select the “Wrap with label tag” and “Before form item” buttons, and then click OK.
Dreamweaver inserts a form field with a descriptive label.
6. In the Property Inspector, choose Paragraph from the Formatting menu.
This wraps a paragraph tag around the label and form field. Next, you'll provide a descriptive name for this form field.
7. Click the newly inserted form field, and in the Property inspector, name this field *username*.
Next, because a login requires both a user name and password, you need to also add a form field for capturing a password.

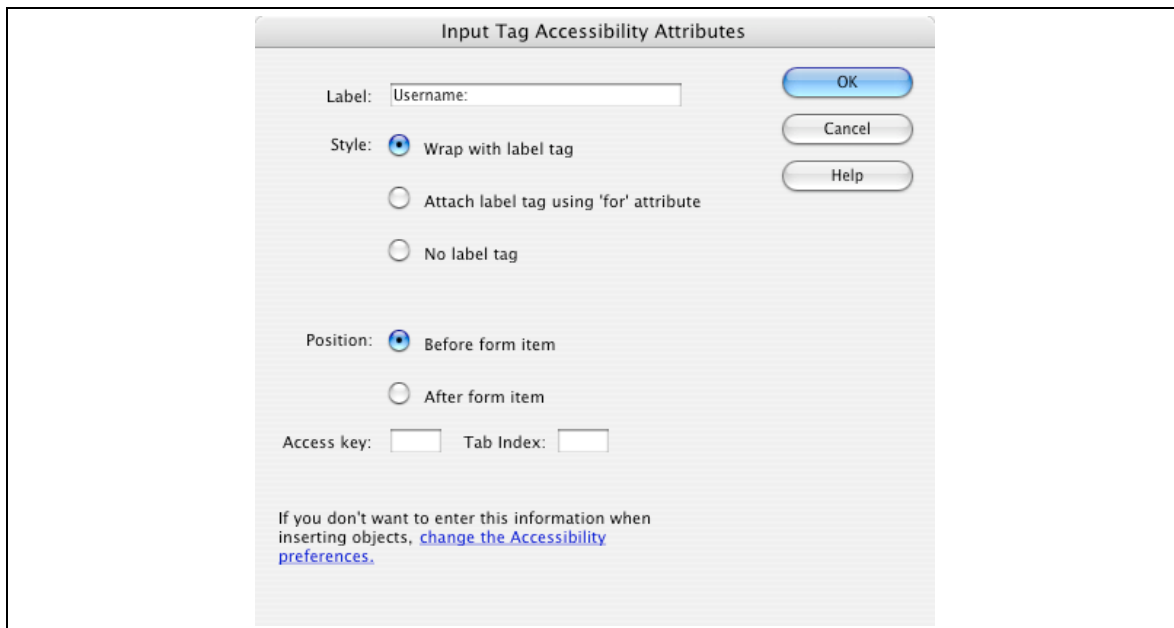


Figure PHP Tutorial 4-1. Dreamweaver's Accessibility features let you add helpful controls— including a descriptive label—to form elements.

8. Click to the right of the form field you just inserted and press the Enter (Return) key to add a new paragraph.
The routine for adding the next form field is the same.
9. Choose Insert→Form→Text Field. In the window that appears, type *Password:* in the label box and then click OK.
This inserts another form field and label.
10. Select the new form field, and in the Property inspector, name this field *password* and select the Password radio button.
By turning this form element into a password field, anything your visitors type in the field will be displayed like this **** or this ●●●, hiding the secret password from nosy passersby watching over their shoulder.

To complete the form, you'll add a Submit button.

11. In the document window, click to the right of the password field and press the Enter (Return) key. Choose Insert→Form→Button.

The Accessibility window appears yet again. In this case, however, you don't need to add a label, since text will appear directly on the button.

12. Click Cancel to close the Accessibility window and insert a Submit button.

The form is complete. Now it's time to let Dreamweaver do its magic.

13. Choose Window→Server Behaviors to open the Server Behaviors window.

Alternatively, you can use the keyboard shortcut: Ctrl+F9 (@cmd-F9).

14. Click the + button and choose User Authentication→Log In User.

The Log In User window appears (see Figure PHP Tutorial 4-2). The first three items should already be filled out: the name of the form, the name of the user name field, and the name of the password field. If you had more than one form on the page, or additional fields inside the one form, you'd have to tell Dreamweaver which form and which fields to use for collecting the login information.

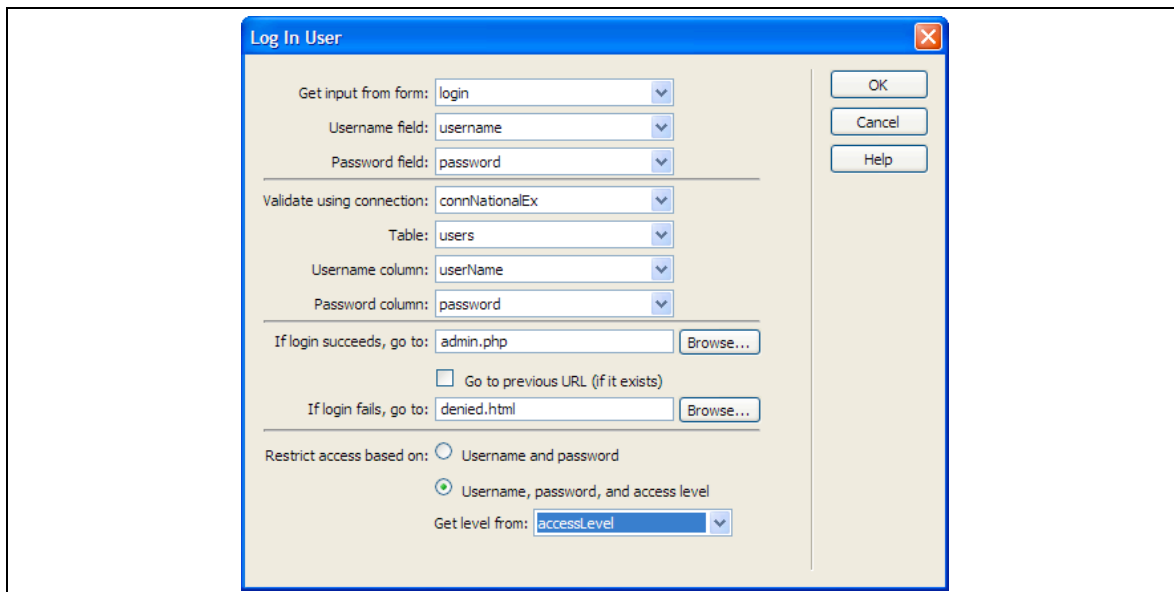


Figure PHP Tutorial 4-2. Before visitors can access password-protected pages, they first need to log into the site. Dreamweaver's Log In User server behavior makes adding this feature a snap.

15. From the "Validate using connection" pop-up menu, select "connNationalEx."

This indicates which database contains the login information. You also need to specify which table and columns contain the user name and password.

16. From the Table menu, select "Users". From the "Username column" pop-up menu, choose "userName." From the "Password column" pop-up menu, choose "userPassword."

You've just established the basic logic of the login behavior: Whatever a visitor types into the two form fields is compared with data stored inside the Users table in the *National Exasperator* database.

Next, you need to specify what happens when there's a match—when your visitor types a valid user name and password into the form—and what happens when the visitor types in an invalid user name or password.

17. Click the Browse button to the right of the “If login succeeds, go to” field. In the Select File window, navigate to and select the file *admin.php* inside the *admin* folder. Click OK.

You’ve just chosen the page that will appear if the login is successful; your visitor’s browser will display an administration page. (For the purposes of this tutorial, it doesn’t matter if the “Go to previous URL checkbox (see page 813 in book) is turned on.)

18. Click the Browse button to the right of the “If login fails, go to” field. In the Select File window, navigate to and select the file *denied.html* inside the *admin* folder. Click OK.

This, of course, is the “access denied” page that will appear when somebody types in an invalid user name or password.

Because this section of the site is for administrators only, you’ll add an additional layer of security by restricting administrative pages using an access level as well as a password and user name. In this way, you’ll also be able to have other password-protected pages—such as a special “paid subscribers” section—for registered visitors, without letting them access administrative areas of the site.

19. Select the “Username, password, and access level” button. From the “Get level from” pop-up menu, choose “userAccess.”

The “Log In User” window should now look like Figure PHP Tutorial 4-2. The database table includes a special field for defining the access privileges of each registered user. For example, each administrator record in the *users* table also includes the value *admin* in the userAccess field.

20. Click OK. Save this file. Press F12 to preview it in your Web browser.

You’ll now try out your newly created login page.

21. In your Web browser, type anything you want in the two fields; click Submit.

Unless you’ve just made an incredible guess, you just typed in a user name and password that doesn’t exist in the database. If the technology gods are smiling, an “Access Denied” page appears.

Now try it again:

22. Click the “Try logging in again” link to return to the login page. Type *bigcheese* in the Username field and *sesame* in the Password field; submit the form.

This time, you’re in: the browser takes you to the main administration page. Here you can jump to the pages you created earlier for adding, updating, and deleting classified ads.

Tip: “sesame” is simply an awful password. Don’t ever use it, or any word you can find in a dictionary, as a password. The reason? Web vandals often launch so-called “Dictionary attacks,” in which they run through different terms pulled from a dictionary until they find a match.

The login script works just fine—you end up at the right page when you type in a valid user name and password. However, none of those other pages are protected yet. You can go to any of them, even if you haven’t logged in. In the next part of this tutorial, you’ll lock down each admin page, so only logged-in administrators can access them.

Password-Protecting the Administration Pages

The password protection features offered by Dreamweaver require you to add a server behavior to each page you wish to protect.

1. Open the file *admin.php* in the *admin* folder.

This page is the main jumping-off point for adding, deleting, and updating products. It should only be accessible to administrators, so you’ll add password protection to it.

2. Make sure the Server Behaviors window is open (Window→Server Behaviors). Click the + button and choose User Authentication→Restrict Access To Page.

(Alternatively, you can use the User Authentication menu in the Insert bar, as pictured in Figure 23-1 on page 810.)

The Restrict Access To Page window appears (see Figure PHP Tutorial 4-3). Since you want to limit access to administrators only, make sure the page is restricted to those with the proper access level.

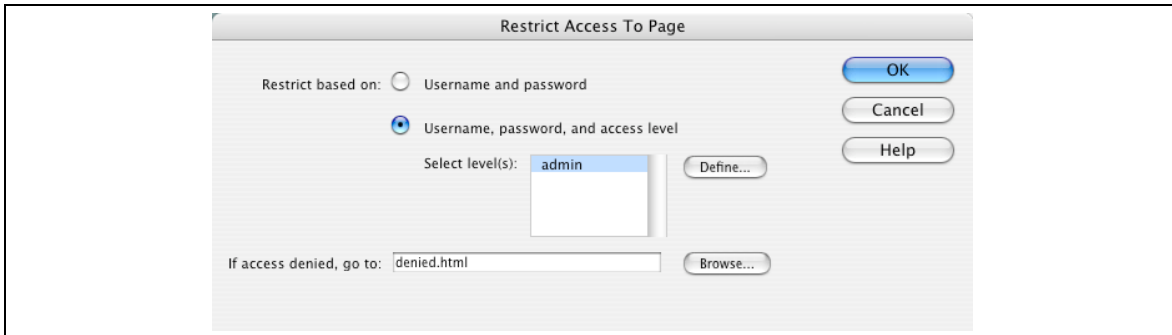


Figure PHP Tutorial 4-3. Administrative pages can be reserved for those with the access level “admin,” while regular subscribers of the National Exasperator would have access to pages intended for subscribers.

3. Select the “Username, password, and access level” radio button.
You want to specify which type of user has access to this page, but first you must tell Dreamweaver what the different levels *are*.
4. Click Define to open the Define Access Levels window. In the Name field, type *admin*. Next, click OK to close the window.
The word *admin* appears in the “Select levels” box. If you had other areas of the site with different access privileges, such as a subscriber area of the site only paying subscribers could access, you could continue to add levels by repeating this step.
5. Click Browse; select the *denied.html* file in the *admin* folder and then click OK. Click OK again to close the Restrict Access To Page window.
To finish this page, you’ll add a “Log out” link.
6. Select Logout (the last link in the left-hand navigation bar on the page).
You’ll turn this text into a “Log out” link.
7. On the Server Behaviors panel, click the + button and then select User Authentication→Log Out User.
Again, this option is also available from the Insert bar as pictured in Figure 23-1 on page 810. In any case, the Log Out User window appears (see Figure PHP Tutorial 4-4). The first radio button should already be selected. The text *Selection: “Logout”* appears in the menu.
These are the proper settings; you’re simply adding the logout script to the words you selected on the page.
Next, you’ll tell Dreamweaver which page to go to after the visitor logs out.

[tutorial_logout.tif], drops

Figure PHP Tutorial 4-4. This server behavior lets you offer visitors the polite option of logging out from your site. It destroys the session variables that track the login status of a visitor.

- Click the Browse button; navigate to and select the file *index.php* in the root folder of the site, and click OK.

When people log out, they'll simply end up at the main products page. Since they're no longer logged in as administrators, they won't be able to access any of the administrative pages without logging back in. The Log Out User window should now look like Figure PHP Tutorial 4-4.

- Click OK to close the Restrict Access to Page window.

Now it's time to test the result.

- Choose File→Save; press F12 to preview the page in your browser.

One of two things will happen: Either you'll end up on the Access Denied page, or you'll see the *National Exasperator's* administration page.

If you quit your Web browser after the previous section of this tutorial, or never logged in to begin with, the Restrict Access To Page server behavior is working: It doesn't recognize you as a registered administrator and denies you access to this page. Click the "Try logging in again" link to go to the login page. Type *bigcheese* in the Username field and *sesame* in the Password field, and submit the form. You're now logged in and are taken to the admin page.

However, if you logged in following the instructions from the previous section in this tutorial, and you haven't quit your Web browser in the meantime, you're still logged in. In this case, the Restrict Access To Page server behavior is again doing its job. You're allowed onto this admin page, because you *are* a registered administrator.

- Click Log Out.

The site logs you out and takes you to the main products page. To make sure you really are logged out, you'll open the administration page again.

- Return to Dreamweaver and the *admin.php* page. Press F12 (option-F12 on Mac) to preview the page again.

You're immediately redirected to the Access Denied page. You're not logged in, so you can't see the administration page. Hooray! The page is successfully protected from snoops.

Note: Some browsers will "cache," or store, the previously viewed administration page, so you might not actually see the Access Denied page. In this case, reload the page by clicking your browser's refresh button.

Of course, the most vulnerable pages (the update, delete, and add product pages) are still accessible to everyone. You need to lock down those pages as well.

- Open the page *add_product.php* page and repeat steps 2-5 in the instructions above.

Repeat this step for all other dynamic pages (*delete_product.php* and *update_product.php*) in the *admin* folder, with the exception of *login.php*. (After all, *that page* should be visible to those who haven't yet logged in.)

If you want, you can also add a Log Out link to each of these pages by repeating steps 6-9.

Now all of the administrative pages in the site are password-protected. Only authorized administrators who log into the site can add, edit, or delete records from the database.

Displaying a Portion of a Page to Logged-In Users

Even though unauthorized users can't access any of the pages that can change the database, they can still see the links you added to the Product Details page in the last chapter—"Edit this Information" and "Delete this Product." Nothing particularly earth-shattering will happen—unauthorized users will just end up at the

Access Denied page—but even that’s not very elegant. Wouldn’t it be tidier if those links didn’t even *show up* except to people logged in as administrators?

You’ll set that up next.

1. Open the *product_details.php* page.

You’ll be doing a little painless programming in the Code view at this point.

2. In the document window, click inside the text “Edit this Information”; in the tag selector, click the `<p>`.

You’ve just selected the paragraph containing the two links. This paragraph should appear only to administrators.

3. Choose View→Code.

The document window switches into Code view (see **Error! Reference source not found.**).

4. Click at the beginning of the selection, just before the opening `<p>`.

The insertion point is now at the start of the paragraph. You’ll add some programming code here.

5. **Type** `<?php if (isset($_SESSION['MM_UserGroup']) && $_SESSION['MM_UserGroup']=='admin') {?>`.

The opening `<?php` tells the application server that some PHP code is coming. In other words, this isn’t HTML—it’s a program that the application server must process. The IF part of this code indicates what’s called a *conditional statement*. In this case, it means, “*if* this person is logged in with an access level of ‘admin,’ *then* the paragraph will appear on the page.”

To determine if the visitor is logged in with the proper access level, the code sneaks a peak at a session variable called `MM_UserGroup`. As mentioned on page 814, when someone logs into the site, the server behavior creates a session variable called `MM_UserGroup`. This variable follows the visitor around the site and contains a word that indicates what level of access he has. The programming you just added first checks to see if the session variable exists (that’s the *isset* weirdness) and then verifies that the session variable is “admin” which indicated that the visitor is logged on as an administrator,. If all of that is true the paragraph letting the visitor access the edit and delete links appears.

6. Click to the right of the closing `</p>` tag (just after “Delete this product”) and type `<?php } ?>`

This code concludes the conditional statement. In other words, all of the HTML between the first line of code you added in the previous step and this final `<?php } ?>` will appear *only* if the user is logged in as an administrator.

The code should now look like **Error! Reference source not found.**. You need to do one last thing on this page.

7. Stay in Code view, and scroll to the top of the page. Place your cursor after the very beginning of the file, before the very first line which begins with `<?php`.

Next, you’ll add a little code that lets the page access the session variables.

8. Type `<?php session_start(); ?>` and press return to move the already existing code after this onto the next line. The first two lines in code view should now look like this:

```
<?php session_start(); ?>
<?php require_once('Connections/connNationalEx.php'); ?>
```

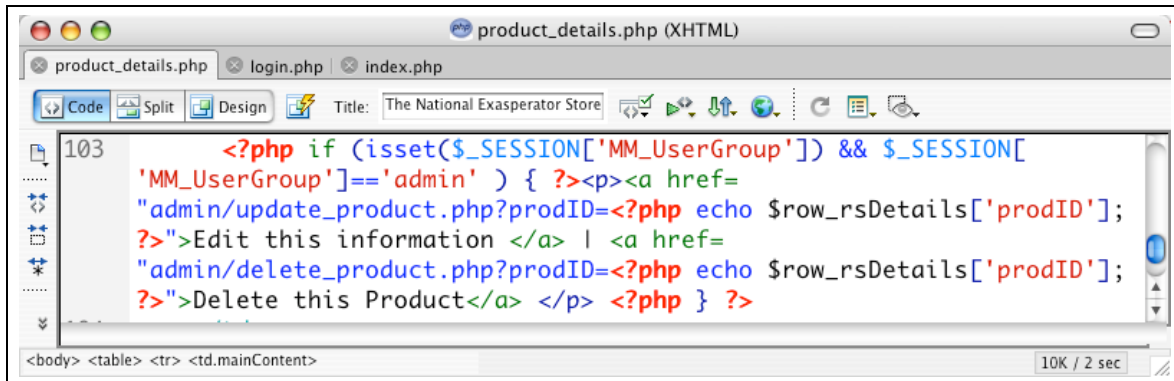
The code you just added, makes PHP turn on its magical session-handling powers. Now this page will be able to “see” an session variables set for the current visitor—such as whether the visitor is logged in and if he or she is an “admin” user.

9. Choose File→Save; press F12 (Option F12) to preview the page.

Because you logged out earlier, the links should now be invisible. To see them, you must log in and return to the product details page.

10. Go back to Dreamweaver, and open the *login.php* page. Press F12 (Option-F12) to preview the page. Type *bigcheese* in the Username field and *sesame* in the Password field. Click Submit.

You're now logged in and taken to the main administration page. If you return to a product details page, the links will miraculously return.



The screenshot shows a web browser window titled "product_details.php (XHTML)". The address bar shows "product_details.php", "login.php", and "index.php". The browser toolbar includes "Code", "Split", "Design", and "Title: The National Exasperator Store". The main content area displays PHP code for a conditional display of links based on user session data. The code is as follows:

```
103      <?php if (isset($_SESSION['MM_UserGroup']) && $_SESSION[
      'MM_UserGroup']=='admin' ) { ?><p><a href=
      "admin/update_product.php?prodID=?php echo $row_rsDetails['prodID'];
      ?>">Edit this information </a> | <a href=
      "admin/delete_product.php?prodID=?php echo $row_rsDetails['prodID'];
      ?>">Delete this Product</a> </p> <?php } ?>
```

The status bar at the bottom shows "<body> <table> <tr> <td.mainContent>" and "10K / 2 sec".

Figure PHP Tutorial 4-5. Here's the finished code that hides the "Delete this Product" paragraph from unregistered users.

11. Click the Store button in the top navigation bar on the page to go to the product listings page. Click the name of any product to see its details.

Voila! The links are back. You can freely edit or delete any product in the database. If you return to the administration page and click the Log Out button, you won't see these links until you log back in.

You could also use this trick to only show the "Log out" link if the visitor is logged in. With very little programming experience, you can use Dreamweaver's Server Behaviors in combination (and perhaps bring in server behaviors from extension developers) to build sophisticated database-driven Web sites.

Now go forth and electrify your sites!