



BUILDING PHP APPLICATIONS WITH MACROMEDIA DREAMWEAVER MX

By Harish Kamath

This article copyright [Melonfire](#) 2000–2002. All rights reserved.

Table of Contents

<u>A Marriage Of Minds</u>	1
<u>Hooking Up</u>	2
<u>Test Drive</u>	3
<u>Breaking Ground</u>	5
<u>Bringing In The Database</u>	7
<u>Appearances Are Everything</u>	10
<u>In And Out</u>	12
<u>I, Robot</u>	14
<u>Weaving The Web</u>	15
<u>Naming Names</u>	16

A Marriage Of Minds

You've already heard of Macromedia Flash, the de facto standard for high quality Web animation. But Flash is not the only ace up Macromedia's sleeve – they've got another winner in their hands with their HTML authoring tool, aptly named Dreamweaver. The latest release of this product, recast as Dreamweaver MX, does a lot more than just help you create HTML pages – it's now a full-featured rapid application development (RAD) tool for Web developers, and it can substantially reduce the time you spend coding ASP, JSP or PHP-based Web sites.

Most RAD tools include very limited support for PHP – which makes them unwelcome visitors on my computer, since most of my work involves working with PHP and MySQL-based sites. When I got my hands on Dreamweaver MX, though, I was impressed with its feature set and capabilities. Sure, it didn't include *everything* on my wish list...but it's clear that the guys at Macromedia have been thinking about the problem, and have made a serious effort to address PHP in their tool.

Over the next few pages, I'll be exploring the new PHP functionality in Dreamweaver MX, in the hope that you find it useful the next time you have a PHP project to code. Come on in, and let's get going!

Hooking Up

Our current development environment consists of a Windows/Linux network, with most developers developing code on Windows PCs and deploying to Linux-based servers. This kind of configuration is supported by Dreamweaver MX, as are other permutations and combinations – so matching the tool to your environment should not be a major issue.

The first step, obviously, is to get yourself a copy of the tool. Hook yourself up to a PC with a high-speed connection, log on to <http://www.macromedia.com/software/>, pick the appropriate version and platform, take Macromedia's user survey (hey, nothing comes free anymore) and get yourself a trial copy. This trial copy works for 30 days, which is more than enough time for you to take the tool through its paces and decide whether it's worth the price tag.

It's a 49-meg download, so don't try this if you're faint of heart or have a s-l-o-w connection – instead, order a copy of the Macromedia evaluation kit on CD-ROM from the Macromedia online store. Sure, it'll take a little longer to get to you, but it beats watching paint dry while the files download.

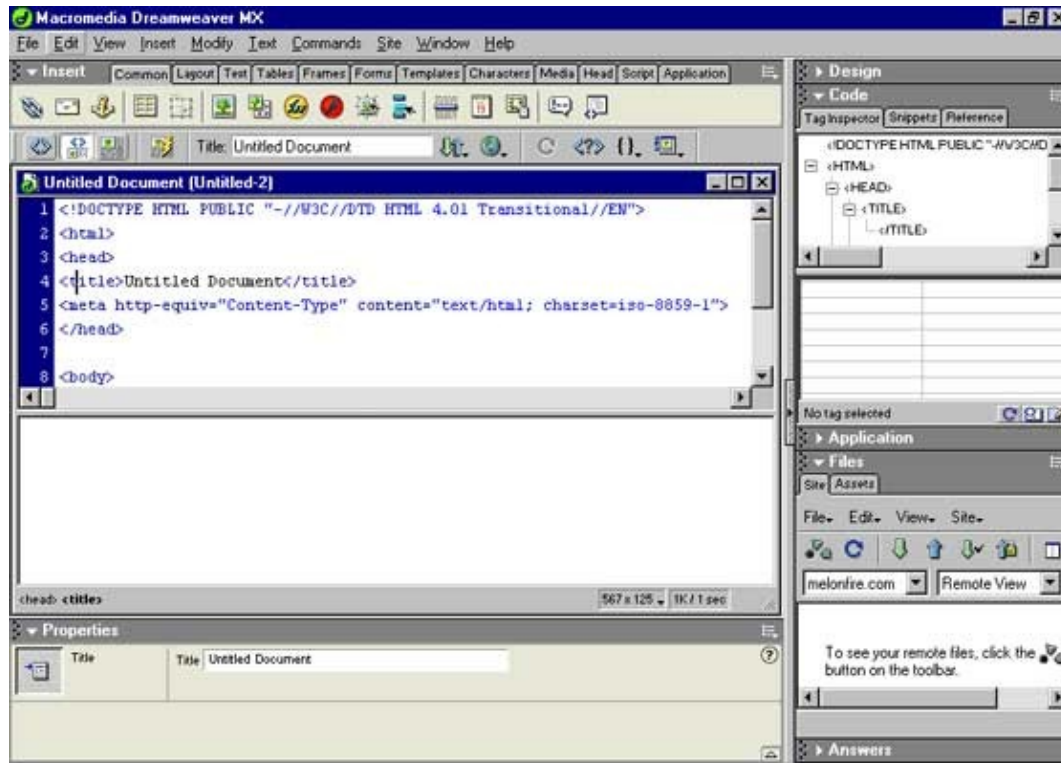
Installation is a breeze – click, accept the license terms (make sure you read them), click a few more times, and you're done!

In case you're wondering what the MX stands for...hey, join the club, because it seems even Macromedia doesn't know. Here's what their FAQ (<http://www.macromedia.com/software/mx/faq/>) has to say on the topic: "The MX moniker is not an acronym and doesn't have a literal translation. MX designates products that are major new releases and part of the Macromedia MX product family." Ain't that interesting?

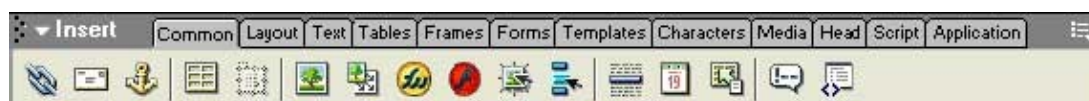
I'm assuming here that the rest of your development box is all set up, with a Web server (I use Apache, <http://httpd.apache.org/>), PHP (<http://www.php.net/>) and a database server (MySQL rocks, get it from <http://www.mysql.com/>). In case you don't have all these, get yourself set up, and then flip the page to take Dreamweaver MX for a test drive.

Test Drive

Dreamweaver MX's interface can be quite daunting to developers used to working in plain-vanilla text editors for PHP development. Here's a quick screen shot of the Dreamweaver work area:



A quick overview of the interface, before we get into the nitty-gritty of PHP development using Dreamweaver MX. At the top is the "Insert" panel that gives you single-click access to insert important HTML elements like images, tables, hyperlinks et al. Obviously, Macromedia's been careful to include support for their other products here too...it's very, very easy to add media like Fireworks images and Flash movies to an HTML document created in Dreamweaver.



The right side of the Dreamweaver workspace includes a number of roll-up panels, each performing a very specific function:

1. The "Design" panel allows the developer to add, edit and delete CSS and HTML styles, and add "behaviors" (ready-to-use JavaScript code snippets) to an HTML document.
2. The "Code" panel provides a handy HTML code inspector that allows you quick access to any section of your HTML document, and also provides you with reference material in case you forget the difference between a `<tr>` and a `<td>`.
3. The "Application" panel provides you with reusable objects that can be bound to your application, and used

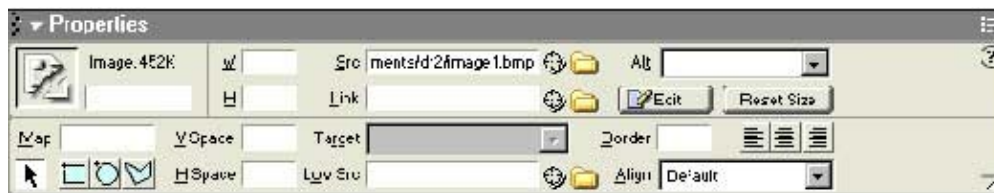
to reduce the time you spend on code development. I'll be talking about this in greater detail a little further down.

4. The "Files" panel provides you with a convenient file manager to view, edit and manipulate files, on both a local or remote server.

5. The "Answers" panel provides quick access to Dreamweaver MX's help files and search engine.



Finally, the bottom of the work area hosts the Properties panel, which allows you to alter the properties of the element currently being edited. In order to see how this works, try adding an image from the "Insert " panel into your document, and notice how all relevant image properties appear at the bottom of the workspace, providing you with an easy way to make alterations to the element attributes.



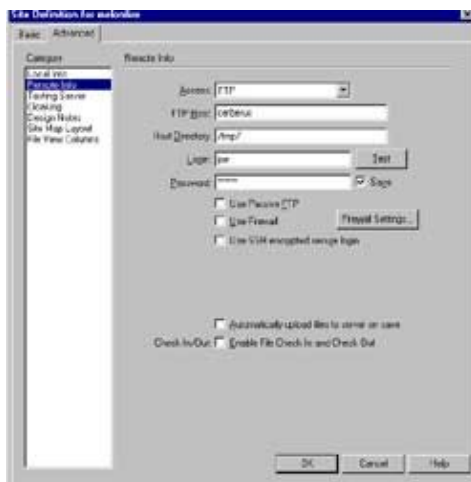
Breaking Ground

Dreamweaver MX makes things easier by allowing you to organize the components of a Web application – static as well as dynamic – into a single entity called a "site". In order to begin, create a new site via the Site → New Site menu and say hello to the wizard that appears.

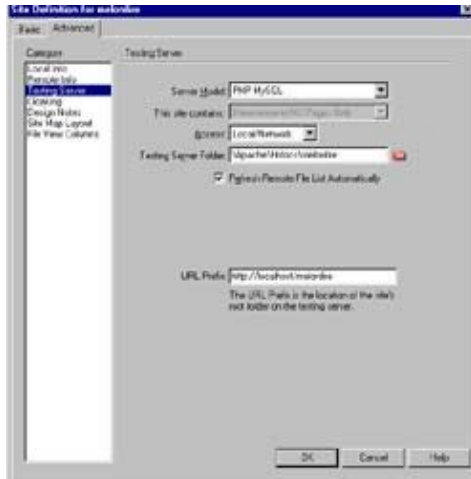
First up, you need to specify your working directory for the site – the name of the site, the root folder for the files and the default folder to look in for images. These folders may be on your local machine, or on a different machine – Dreamweaver refers to this collection of data as the "local site".



You can also optionally tell Dreamweaver about a "remote site" – this is usually the location where your Web server is running, and where your files will ultimately go for testing and deployment. Dreamweaver allows you to also specify the access mechanism for such a remote site – FTP, LAN, SourceSafe and so on.



Finally, if you're developing dynamic pages, you should also specify a "testing server" – this is a folder which can be used by Dreamweaver to process your scripts and dynamically generate documents, either from a database or elsewhere. In an Apache+PHP+MySQL environment, this folder would be the same as the root folder for the application on the Web server. You can also specify a URL prefix, which is used when you try to access the application via a Web browser.



Here's a list of the values I specified when creating a sample site in Dreamweaver:

```
Local info:
Site name: melonfire
Local root folder: C:\Apache\htdocs\melonfire\

Remote info:
Access: FTP
FTP Host: cerberus
Login: me
Password: **

Testing server:
Server model: PHP MySQL
Access: Local/Network
Testing server folder: C:\Apache\htdocs\melonfire\
URL prefix: http://localhost/melonfire/
```

In case you missed something, all this means is that I'll be doing all my development on my local machine (which has Apache, PHP and MySQL all talking to each other) in the folder "C:\Apache\htdocs\melonfire\", and will also be uploading copies of the files to a remote server named "cerberus" via FTP

Bringing In The Database

One of the coolest features of Dreamweaver MX is the ability to build a dynamic database-driven Web site without having to write a single line of PHP code on your own. Over the next few pages I'm going to demonstrate how, by having Dreamweaver do the hard work of writing PHP code that allows you to view, add and delete records from a MySQL table.

Let's assume that I have a database table called "user", as shown below.

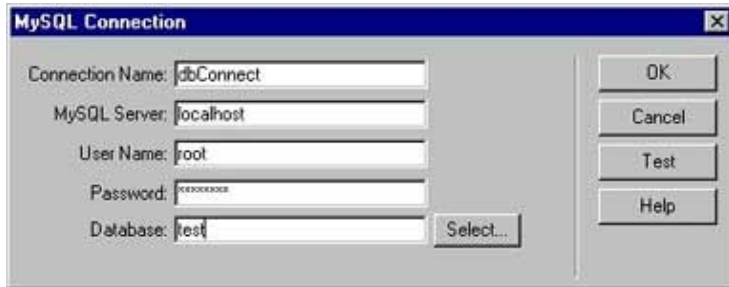
```
mysql> SELECT * FROM user;
+-----+-----+-----+
| id | username | password |
+-----+-----+-----+
| 3 | merrill | 2fda14e52bde1a87 |
| 4 | pooja | 0d53926314545f3c |
| 5 | john | 2ca0ede551581d29 |
| 6 | joe | 7b57f28428847751 |
| 7 | tom | 675bd1463e544441 |
| 8 | bill | 656d52cb5d0c13cb |
+-----+-----+-----+
6 rows in set (0.05 sec)
```

Now, the built-in features of Dreamweaver MX allow you to automate the processes of listing the records in the table, adding new records, modifying existing records, and deleting records – all with minimum hand coding.

The first thing to be handled is the database connection. Dreamweaver MX does the dirty work of opening, managing and closing the database connection – all you need to do is pop open the "Application" panel (on the right side of the workspace), click the "Database" tab, hit the button with a plus (+) symbol on it, and select "MySQL Connection" from the option list.



You should see the following dialog:

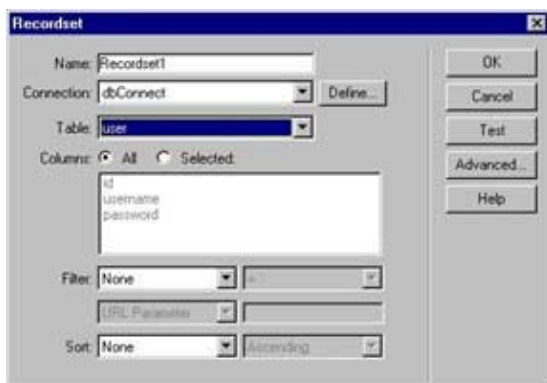


This box allows you to specify access information for the database server. For this example, create a connection called "dbConnect" and provide Dreamweaver with the username and password required to access the database server, together with the name of the database to be used.

Once the connection has been created, Dreamweaver will automatically add a set of scripts to your site directory; these will be used for all subsequent database interaction. If the connection is created successfully, you should be able to view all the tables (as well as views and stored procedures for compliant databases) in the "Database" tab of the "Application" panel.



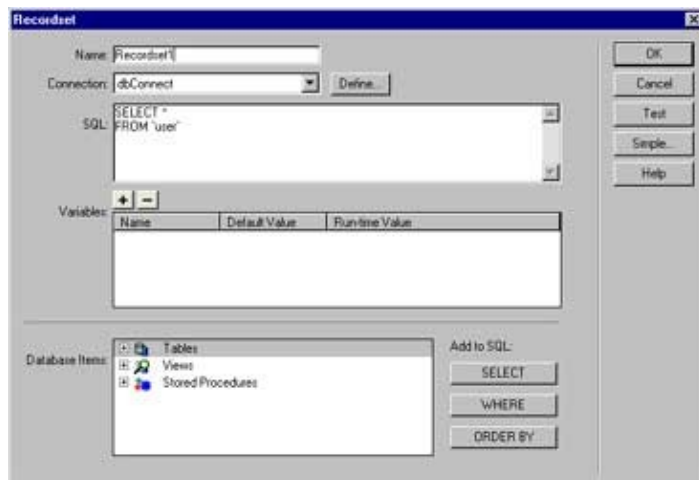
With the database connection created, the next step is to write a script that will list all the records in the table. Use the "Bindings" tab of the "Application" panel to create a "Recordset" object (click the button with the plus symbol again to get to this option) and you'll get a simple wizard-like interface which allows you to retrieve a set of records from the database.



Once you've filled in the requisite information – note that you can also sort and filter the results – hit the "Test" button to see if everything works as it should. If it does, save your settings and you'll now see a new item in the "Binding" tab of the "Application" panel, referencing your newly created Recordset.



You can customize the SQL query created via the "Advanced..." button in this wizard.



Appearances Are Everything

Now, go back to the editor and decide how you want to display the data returned in the Recordset. I like to keep things simple, so I merely created a bulleted list of users,

```
<ul>
<li></li>
</ul>
```

and then dragged columns from the Recordset object in the "Application" panel to the appropriate place on the page where I'd like the data displayed (Dreamweaver wrote the code for me).

```
<ul>
<li><?php echo $row_Recordset1['username']; ?></li>
</ul>
```

And when you preview it in the browser, you should see something like this:

```
• merrill
```

Cool, huh? I managed all that using simple point-and-click actions, with no requirement to know anything about PHP's MySQL API or remember complex function syntax.

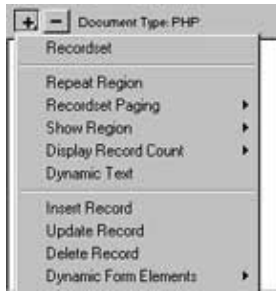
Of course, the example's still somewhat incomplete – all it does is display the first record. What we really need is the entire recordset returned by the SQL query. In order to meet this and other requirements, Dreamweaver includes a variety of different "server behaviors", which can be used to add extra functionality to your application.

In order to see how this works, select the first item of your newly-created list in the editor,

```
• {Recordset1.username}
```

pop open the "Server Behavior" tab of the "Application" panel, and select the "Repeat Region" behavior from the drop-down option list, and tell it to loop for as many times as there are records.





Dreamweaver will automatically insert PHP code that iterates over the returned Recordset object and print the entire set of records returned. Here's the code,

```
<ul>
<?php do { ?>
<li><?php echo $row_Recordset1['username']; ?></li>
<?php } while ($row_Recordset1 =
mysql_fetch_assoc($Recordset1)); ?>
</ul>
```

and here's the output:

- merrill
- pooja
- john
- joe
- tom
- bill

Dreamweaver comes with a whole bunch of other, similar behaviors to meet most common requirements – you can just as easily (for example) display the total number of records found, implement paging with next and previous links, or dynamically show sections of a page based on the current position of the recordset pointer.



In And Out

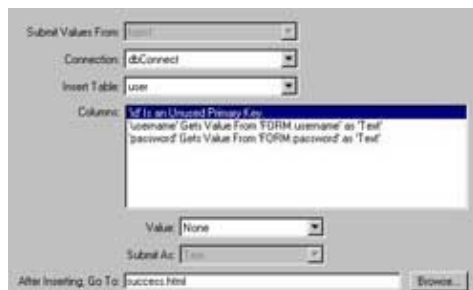
OK, so that took care of displaying the records – now how about inserting and deleting records?

First, use the Dreamweaver editor to create an HTML form that maps to the columns of the "user" table.



A screenshot of an HTML form within a dashed red border. It contains three input fields: "Username" (text), "Password" (password), and a "Submit" button.

Once the form has been prepared, apply the "Insert Record" server behavior, and you'll see a dialog box like this:



This dialog box makes it very easy to map form fields to the table in which data is to be inserted. Once you select the appropriate database connection and table, the fields of the table are displayed; these fields can be mapped to specific fields in the form via the "Value" drop-down list. Dreamweaver takes care of writing the SQL query needed to insert the data entered by the user in each field into the appropriate column in the table.

After you've finished mapping form fields to table columns, remember to specify the URL to which the browser should be redirected once the data has been successfully inserted into the database.

Once you're done, save the PHP script and access it via your browser. You should see the HTML form you created:

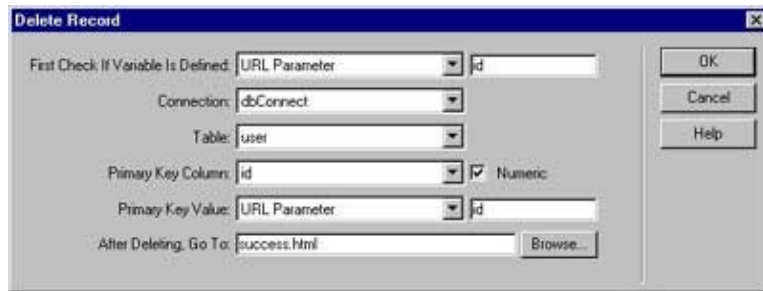


A screenshot of the HTML form from the previous image. The "Username" field is now filled with the text "jimmy". The "Password" field is empty and masked with dots. The "Submit" button is still present.

Enter some values and submit the form. The code generated by Dreamweaver MX will insert the values into the "user" table and then redirect the browser to the URL selected...all without the developer writing a single line of code!

It's just as simple to delete records – create a new PHP script and use the "Delete Record" server behavior. You'll see the following dialog box, which asks you to specify the variables that the script should use when

identifying which record to delete.



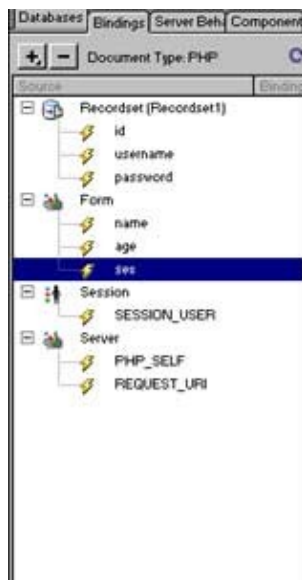
As you can see, in the example above, I've told Dreamweaver to look for the "id" variable in the URL string, and to delete the record corresponding to that ID in the "user" table. Save the script (I'll assume it's called "delete.php"), and access it via your browser (remember to add the "id" variable–value pair to the URL string, like thus: <http://your.server/delete.php?id=15>), and the record corresponding to the ID provided should get deleted from the database.

Obviously, it's pretty easy to now link the three scripts above with each other, so as to build a primitive, yet fully–functional, user administration module (in case you're wondering, yes, Dreamweaver also comes with an "Update Record" behavior – I'll leave that to you to experiment with.)

I, Robot

In addition to recordset bindings, Dreamweaver also makes it easy to create bindings to other variables – you can register variable names in the "Bindings" tab of the "Application" panel, and then simply drag and drop those variable names from the panel into the code editor whenever you need to manipulate them; Dreamweaver automatically writes the PHP code to echo the current value of the variable to the standard output.

As an example, consider the following example, in which I've added three form variables, one session variable and one server variable to the Dreamweaver "Bindings" tab.



I can now access any of these variables in my PHP script simply by dragging them from the "Bindings" tab into the editor window – Dreamweaver automatically writes PHP code to echo() the current value of the variable to the screen.

If you're using a large number of common variables in different scripts within your applications – session variables are a good example – this drag-and-drop feature can save you a fair amount of repetitive typing. And that's always a Good Thing. Now if only Dreamweaver could automatically track variables as I created them and add them to the "Bindings" tab automatically, without me having to manually register them...

Weaving The Web

The thing worth noting about the examples on the previous pages is that I was able to execute them very rapidly, with no actual hand-coding of PHP function calls – Dreamweaver did most of the heavy lifting, with me just clicking my way through menus and dialog boxes. If you're new to PHP, or just looking for a RAD tool to quickly develop basic PHP applications, Dreamweaver MX, with its built-in behaviors, could be just the ticket to get you up and running in as little time as possible.

That said, while Dreamweaver MX represents a definite evolution in Macromedia's support for PHP developers, the program is certainly not perfect. Among the things that would help to make the PHP coding experience even more pleasant in Dreamweaver MX: automatic syntax checking, automatic code completion, automatic variable registration and binding, more built-in behaviors, greater PHP toolbar functionality, and support for related plug-in applications (such as the Zend Optimizer). Maybe next time...

More information about Dreamweaver MX can be obtained at the following links:

The Macromedia Designer and Developer Center (PHP section), at

<http://www.macromedia.com/desdev/topics/php.html>

The Getting Started With Dreamweaver MX tutorial, at

<http://www.dmxzone.com/ShowDetail.asp?NewsId=3707>

The official Web site for "Dreamweaver MX: PHP Web Development" at

<http://www.dmxzone.com/ShowDetail.asp?NewsId=3340>, a book that teaches Web professionals how to use Dreamweaver MX to produce PHP-backed Web sites.

Extending Dreamweaver MX and PHP: The PHP Login Suite for Dreamweaver MX, at

<http://www.dreamweavermxsupport.com/extensions/phploginsuite/>, which contains 16 Server Behaviors and 2 objects designed to allow you to quickly and easily create a MySQL-backed authentication system for a Web site.

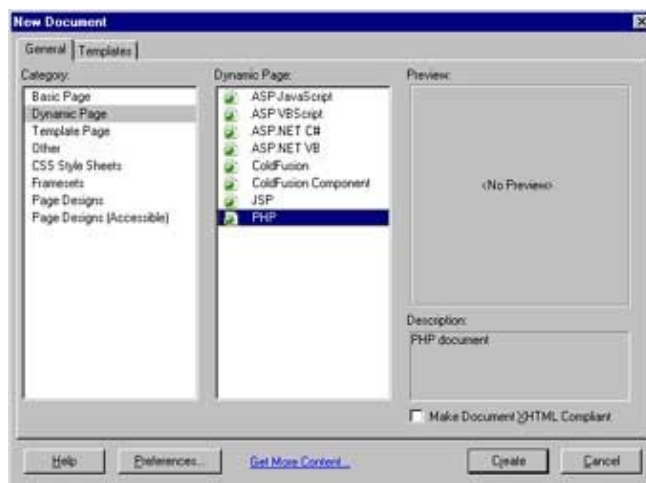
Until next time...be good!

Note: All examples in this article have been tested on Windows (8 with Macromedia Dreamweaver MX Trial Edition. Examples are illustrative only, and are not meant for a production environment. Melonfire provides no warranties or support for the source code described in this article. YMMV!

Naming Names

Once your site has been set up, it's time to get started with a simple example – a page that display PHP environment settings via the `phpinfo()` function.



Dreamweaver provides you with an easy way to get started with building a PHP script – pop open the File → New menu, and select "Dynamic Page" and "PHP" from the pre-defined templates.



You can now add the function call to `phpinfo()` in this page – switch to the code view using the View → Code menu option, and type the following code into your document:

```
<?php phpinfo(); ?>
```

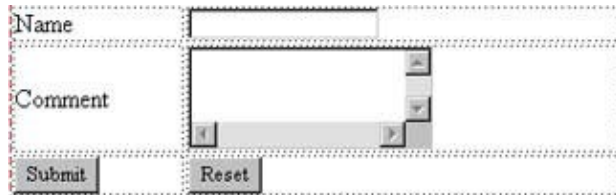
Save the file, and have Dreamweaver preview it for you via the toolbar icon (you can also hit the F12 key) and a new browser window should pop up with the output of the script.

PHP Version 4.2.3		
System	Windows 2003 4.1.0	
Build Date	Sep 6 2002 11:38:41	
Server API	Apache	
Virtual Directory Support	enabled	
Configuration File (php.ini) Path	C:\WINDOWS\php.ini	
Debug Build	no	
Thread Safety	enabled	
<small>The program is distributed under the Zend Scripting Language License. Zend Engine v1.2.0, Copyright (c) 1999-2002 Zend Technologies</small>		

If you're using PHP, you're going to be doing a lot of form processing – let's see how you can speed up the process in Dreamweaver. First, create a simple, properly-formatted HTML form using a combination of the "Tables" and "Forms" tabs on the "Insert" panel – this form should contain two text input areas, one named "name" and the other named "comment".



Add a couple of submit and reset button and the form is ready for action.



Here's the code:

```

<html>
<head>
<title>Untitled Document</title>
</head>

<body>
<form action="this.php" method="post">
<table width="75%" border="0">
<tr>
<td>Name</td>
<td><input name="name" type="text"></td>
</tr>

```

```

<tr>
<td>Comment</td>
<td><textarea name="comment" cols="" rows=""></textarea></td>
</tr>
<tr>
<td><input name="submit" type="submit" value="Submit"></td>
<td><input name="Reset" type="reset" value="Reset"></td>
</tr>
</table>
</form>

</body>
</html>

```

Now comes the interesting part – integrating PHP business logic into this example. Click the "PHP" tab that should have appeared on your "Insert" panel, and take a look at the functions it provides:



Pretty basic: you can insert PHP code to access POST, GET, cookie and session variables, include() and require() files, and attach programming constructs like "if" and "else" statements.

Now, you probably already know how PHP form processing typically works. A form processing script contains two logical blocks – one displays the empty form and the other processes the user input submitted. Therefore, the first thing to do is to insert an "if" conditional statement at the top of the code – just click on the "If" button and Dreamweaver will do it for you:

```

<?php if ?>
// form code

```

Next, you need to check if the form has been submitted or not, via the "submit" form variable. Use the "Form Variables" button, and write some code around what Dreamweaver inserts for you:

```

<?php if (!$HTTP_POST_VARS['submit']) { ?>
// form code

```

Once you've wrapped your empty form in the "if" branch of the conditional, you need to add form processing logic to the "else" branch. Use the "else" button on the toolbar, and let Dreamweaver insert it for you.

```

<?php } else { ?>

```

This "else" branch of the script is supposed to merely display the data entered by the user. You can do this via the echo() command, also accessible via a button on the PHP toolbar.

```
<?php echo $HTTP_POST_VARS['name']; ?>
```

Here's the final script:

```
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
</head>

<body>
<?php if (!$HTTP_POST_VARS['submit']) { ?>
<form action="this.php" method="post">
<table width="75%" border="0">
<tr>
<td>Name</td>
<td><input name="name" type="text"></td>
</tr>
<tr>
<td>Comment</td>
<td><textarea name="comment" cols="" rows=""></textarea></td>
</tr>
<tr>
<td><input name="submit" type="submit" value="Submit"></td>
<td><input name="Reset" type="reset" value="Reset"></td>
</tr>
</table>
</form>
<?php } else { ?>
<table width="75%" border="1" cellspacing="5" cellpadding="5">
<tr>
<td width="39%">Name</td>
<td width="61%"><?php echo $HTTP_POST_VARS['name']; ?></td>
</tr>
<tr>
<td>Comments</td>
<td><?php echo $HTTP_POST_VARS['comment']; ?></td>
</tr>
</table>
<?php } ?>
</body>
</html>
```

Now, you can easily test this code using the "Preview" button, as discussed earlier. Here's the output:

Name	<input type="text" value="Joe"/>
Comment	<input type="text" value="This rocks!"/>
<input type="button" value="Submit"/>	<input type="button" value="Reset"/>

Name	John Doe
Comments	This rocks!

It should be noted that Dreamweaver's attempt to make PHP scripting a point-and-click exercise, as demonstrated via the example above, isn't really an unqualified success – the functionality provided on the PHP toolbar is extremely limited, and most developers would find it more efficient to type in PHP statements themselves, rather than clicking their way through the buttons on the PHP tab. Don't be disappointed, though – although the program fails at this task, it excels when it comes to building database-driven dynamic pages (more on this on the next page).

Once you're happy with the way your script is working, you can transfer the file to your remote server (named "cerberus" in this case). Pop open the "Files" panel on the right side of the workspace, select the "local view", and use the arrow buttons to transfer files from your local site to the remote site.



You can also perform transfers in the other direction, by changing to the "remote view" and either putting or getting files to/from the server.

