Installing PHP under Windows

Not a Linux fan? Looking for a hassle−free way of setting up a PHP environment on your Windows box? Read as Matthew guides us through the process of installing Apache, MySQL and PHP on a Windows based machine, otherwise known as WAMP.

This article is geared towards people who want to code and test their own, or other people's, php scripts on their Windows box. No advanced technical knowledge is assumed. This tutorial is unsuitable for advanced users.

Why Apache and MySQL? Because they are available for free, quite easy to use and reliable. Why windows? Because there are many installation guides written for the Linux user, but Windows generally gets overlooked. And getting it all to work can be a real daunting task.

What is Apache?

Apache is our web server of choice; it's free, reliable and well supported. In fact, as of October 2003 apache web servers had a market of over 67%, compared to their nearest rival Microsoft's ISS having under 25% (http://news.netcraft.com/) So its not just a few people who think Apache is a good thing! A web server basically serves web pages from the server to the client browser. For every page you see when connecting to the internet, somewhere there is a network server sending it to you. The pages are usually sent in html format, along with images and other media.

What is MySQL?

[From MySQL.com] *'MySQL is the world's most popular open source database, recognized for its speed and reliability.'* A Database is a place for you to store data, whether it is user details or article text. MySQL is highly configurable and easy to use.

What is PHP?

[From PHP.net] '*PHP is a widely−used general−purpose scripting language that is especially suited for Web development and can be embedded into HTML.*' It allows web pages to be dynamically created, that is they can be easily updated and changed quickly. I'm assuming if you want to use PHP you have some idea what it is, and what it can do, if not, there are too many possibilities to be summed up in a paragraph.

WAMP!

Yes, despite what many may say, Apache, MySQL and PHP (AMP) can be installed and work fine with Windows; this tutorial explains how. This tutorial is written for Windows XP, Apache 2.0.47, MySQL 4.0.15, and PHP 4.3.3; however, it should work fine with Win2K, and other versions of AMP. I would not recommend installing AMP on Win95/98/ME.

Ok, before we start remember to run an up−to−date virus checker on any files that you download from the internet etc. You must be logged on as a user with administrator privileges.

Download Appropriate Files

- Get the latest version of the Apache HTTP Server from http://httpd.apache.org/. Make sure you get the Win32 Binary (MSI Installer) version
- Get the latest version of MySQL, with installer, from www.mysql.com
- Get the latest version of PHP from www.php.net. Make sure you get the ZIP file with server API included

Install Apache Web Server

Run the Apache installer, and follow all on screen instructions. As you are setting this up for development and testing, the Network Domain and Server Name will both be ‘localhost’ (No quote marks). For the contact address, enter any email address you want; it’s not really important on a development/test server. Make sure Apache is installed as a service.

Test Apache Web Server

Open your web browser, and navigate to the address ’localhost’ (or http://localhost/). You should see a lovely page telling you Apache has been set up and is working correctly, but no pages have yet been added. If you’d like to at this stage, you may add any html pages you want. If you installed Apache with the default paths, go to the directory ‘C:\Program Files\Apache Group\Apache2\htdocs’; this is where any pages you want to make available are kept. Feel free to delete all the pages that start with index and add your own index.html file. index.html will be the first page you see when you go to http://localhost/, try it out, add your own! Congratulations your web server is up and running!

Next we will be installing MySQL, a free database program. This will be necessary for almost all php scripts beyond any very basic scripts. Create a temporary directory – on the desktop is a good place; we will be getting rid of it soon anyway. Extract the MySQL ZIP file to the temp directory and run the setup.exe. Follow all the onscreen instructions. Once the setup program has finished and installed MySQL, delete the temporary directory.

Click Start −> Run and type cmd in the box, click Ok. This will open a console window (or MS−Dos box). In the console window enter the following commands and check that you receive the correct response:

```
Enter: cd c:\mysql\bin
Enter: mysqld−max−nt −−install
Response: Service successfully installed.
Enter: net start MySql
Console should say: The MySql service is starting.
Console should say: The MySql service was started successfully.
```

Congratulations you have installed MySQL; however, before we are done here we are going to make MySQL more secure, by adding a password, so only you can change the databases.

At the console window type the following commands (replace new_password with your new password):

```
Enter: cd c:\mysql\bin
Enter: mysql −u root mysql
Response: Welcome to the MySQL monitor .....
Enter: UPDATE user SET Password=PASSWORD('new_password')
Enter: WHERE user='root';
Response: OK...
Enter: FLUSH PRIVILEGES;
Response: OK...
Enter: DELETE FROM user WHERE user='';
Response: OK...
Enter: DELETE FROM user WHERE Host='%';
Response: OK...
Enter: DELETE FROM user WHERE User='';
Response: OK...
Enter: DELETE FROM db WHERE Host='%';
Response: OK...
Enter: exit
Response:  Bye
Enter: exit
Response: Console window closes.
```

Create the directory C:\php and extract all files in the PHP ZIP file here. You will now need to copy some files to your Apache Directory (C:\Program Files\Apache Group\Apache2\).

Copy the following files:

```
C:\php\php4ts.dll
C:\php\sapi\php4apache2.dll
C:\php\php.ini-recommended
```

Rename ‘php.ini−recommended’ to ‘***php.ini***’ and open it in notepad.

Search for 'doc_root' until you find the line:

```
doc_root =
```

Change this line to:

```
doc_root = C:\Program Files\Apache Group\Apache2\htdocs
```

Open the file httpd.conf in notepad. Httpd.conf can be found in: C:\Program Files\Apache Group\Apache2\conf\; add the following lines to the end of the file:

```
   LoadModule php4_module php4apache2.dll
   AddType application/x-httpd-php .php
```

Congratulations you have just installed PHP on your computer! But don’t start the part yet, test to make sure it works!

Test PHP

To test that PHP has been correctly installed click ‘***Start > All Programs > Apache > Configuration > Test Configuration***’. A console window should show briefly then disappear, if this happens then everything is ok, it if stays, take note of any errors and makes sure you haven’t missed any of the instructions above.

Restart Apache

For all the changes to take effect, and Apache to recognize PHP, Apache must be restarted. To do this click ‘***Start > All Programs > Apache > Control > Restart***’.

Now to see whether Apache recognises PHP, open notepad and save a file called "test.php" (putting the quotes around the name insures that notepad saves it as test.php, and not test.php.txt. Notepad sometimes tries to be too helpful!).

In this file type the following:

```
<?php
 echo 'PHP is working.<br>';
 echo phpinfo();
?>
```

Save this file in your doc root (C:\Program Files\Apache Group\Apache2\htdocs).

Open your web browser and navigate to http://localhost/test.php you should see the line 'PHP is working' and a whole load of info about the current PHP configuration.

Testing Apache, MySQL and PHP Together

Call up a console window (Start > Run > cmd > OK), change to the mysql bin directory (cd c:\mysql\bin) start mysql, logged in as the root user (mysql −u root −p <enter> password)

We are now going to create a (simple) database:

```
CREATE DATABASE simple;
USE simple;
CREATE TABLE simple_table (
   id INT AUTO_INCREMENT,
   text MEDIUMTEXT,
   PRIMARY KEY (id)
);

Response: OK …

Enter: GRANT ALL
Enter: ON simple.*
Enter: TO testuser@localhost
Enter: IDENTIFIED BY 'testpassword';
Response: OK …
```

Create a new file called "test_insert_mysql.php" and enter the following code into it:

```
<?php
 // Connect to the database
 $dbhost = 'localhost';
 $dbusername = 'testuser';
 $dbpasswd = 'testpassword';
 $database_name = 'simple';
 $connection = mysql_connect("$dbhost","$dbusername","$dbpasswd")
  or die ('Couldn\'t connect to server.');
 $db = mysql_select_db("$database_name", $connection)
  or die('Couldn\'t select database.');

 // Generate SQL code to store data on database.
 $insert_sql = 'INSERT INTO simple_table (text) VALUES (\'test text, 1,2,3\')';

 // Execute SQL code.
 mysql_query( $insert_sql )
  or die ( 'It Didn\’t Work: ' . mysql_error() );

 // Tell User we are done.
 echo 'Code Inserted';
?>
```

Create a new file called "test_select_mysql.php" and enter the following code into it:

```
<?php
 // Connect to the database
 $dbhost = 'localhost';
 $dbusername = 'testuser';
 $dbpasswd = 'testpassword';
 $database_name = 'simple';
 $connection = mysql_connect("$dbhost","$dbusername","$dbpasswd")
  or die ('Couldn\'t connect to server.');
 $db = mysql_select_db("$database_name", $connection)
```

```
  or die('Couldn\'t select database.');

 // Generate code to retrieve data from database.
 $select_sql = 'SELECT text FROM simple_table';

 // Retrieve code from database.
 $result = mysql_query( $select_sql )
  or die ( 'It Didn\’t Work: ' . mysql_error() );

 // Display results to user.
 while ( $row = mysql_fetch_object ( $result ) )
 {
  echo $row–>text . ‘<br>’;
 }
?>
```

Browse to http://localhost/test_insert_sql.php, then http://localhost/test_select_sql.php. Every time you view test_insert_sql.php, it adds a line to the database; viewable from test_select_sql.php. Now, even if you reset you computer the data is still stored in the database. This example may not be the most thrilling, but hopefully it gives you a small idea of what is possible with WAMP.

If the above scripts all work as planned, then your WAMP environment is set up correctly! Tune in to the next PHP Tutorial for some actual coding!