

Setting up PHP/MySQL on Your Mac

www.derwanderer.net

After I had my interest primed by upcoming work on PHP/Websites, WordPress plug-ins and the like, I realized it was necessary to investigate setting up my Mac to develop and debug PHP/MySQL sites locally. The steps were surprisingly fairly easy though there were a couple of aggravating gotchas. Hopefully, this document will save others time and frustration.

I was specifically interested in setting up a local installation of WordPress to try my hand at building simple CMS systems and developing custom plug-ins to augment CMS functionality. So, as a bonus, I'll explain how I got this to work.

The Three Products You Need:

You need three products working in concert to make all of this happen: PHP, MySQL, and the Apache web server.

MySQL:

You can download the latest version of MySQL here:

<http://dev.mysql.com/downloads/mysql/5.0.html>

Scroll down to the bottom for the Mac package. The installation is straightforward. Be sure to install all the packages and the preferences pane so that you can easily start MySQL Server visually and automatically at startup.

Logging into the service, creating databases, managing users is a topic itself, one that I can't cover here. However, the following lists the most common terminal commands that I use. Yes, for now, that's all I know how to do: geeky terminal commands:

- Go to folder: `cd /usr/local/mysql`
- Start up terminal: `sudo ./bin/mysqld_safe &`
- Confirm it's running: `./bin/mysqlshow -u root`
- See what all you can do: `./bin/mysqladmin -u root -help`
- Use the client to do stuff: `./bin/mysql -u root`
- Shut it down when done: `./bin/mysqladmin -u root shutdown`

For all of the above commands that use "-u root" as the user, you will also need to add "-p" if you've specified a password for the root user. When you do, you'll be prompted for a password.

The place that you'll do most of your work is in the client. Here's a screenshot of the client in action:

```
Terminal — mysql — 80x50
theron-welchs-computer:/usr/local/mysql Theron$ ./bin/mysqlshow -u root -p
Enter password:
+-----+
|   Databases   |
+-----+
| information_schema |
| mysql           |
| test           |
| testWP         |
+-----+
theron-welchs-computer:/usr/local/mysql Theron$ ./bin/mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.0.77 MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> █
```

OK, why not? Here are a couple of commands that you can use from the “mysql>” prompt inside the client:

Create a database:

```
CREATE DATABASE testWP
```

Create a user with access to the testWP database:

```
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER,INDEX ON
testWP.* TO 'yomamma@localhost' IDENTIFIED BY 'password';
```

PHP:

This was a little tricky. I found the most recommended link for installing PHP on the Mac, but the download from the most prominent link didn't install. The installer for the first link would just shut down immediately. The second one, labeled “insecure” was the one that worked. And, it was the one who's version matched exactly what I saw referenced on other pages. Well, this was a local installation so I didn't worry. Here's the link:

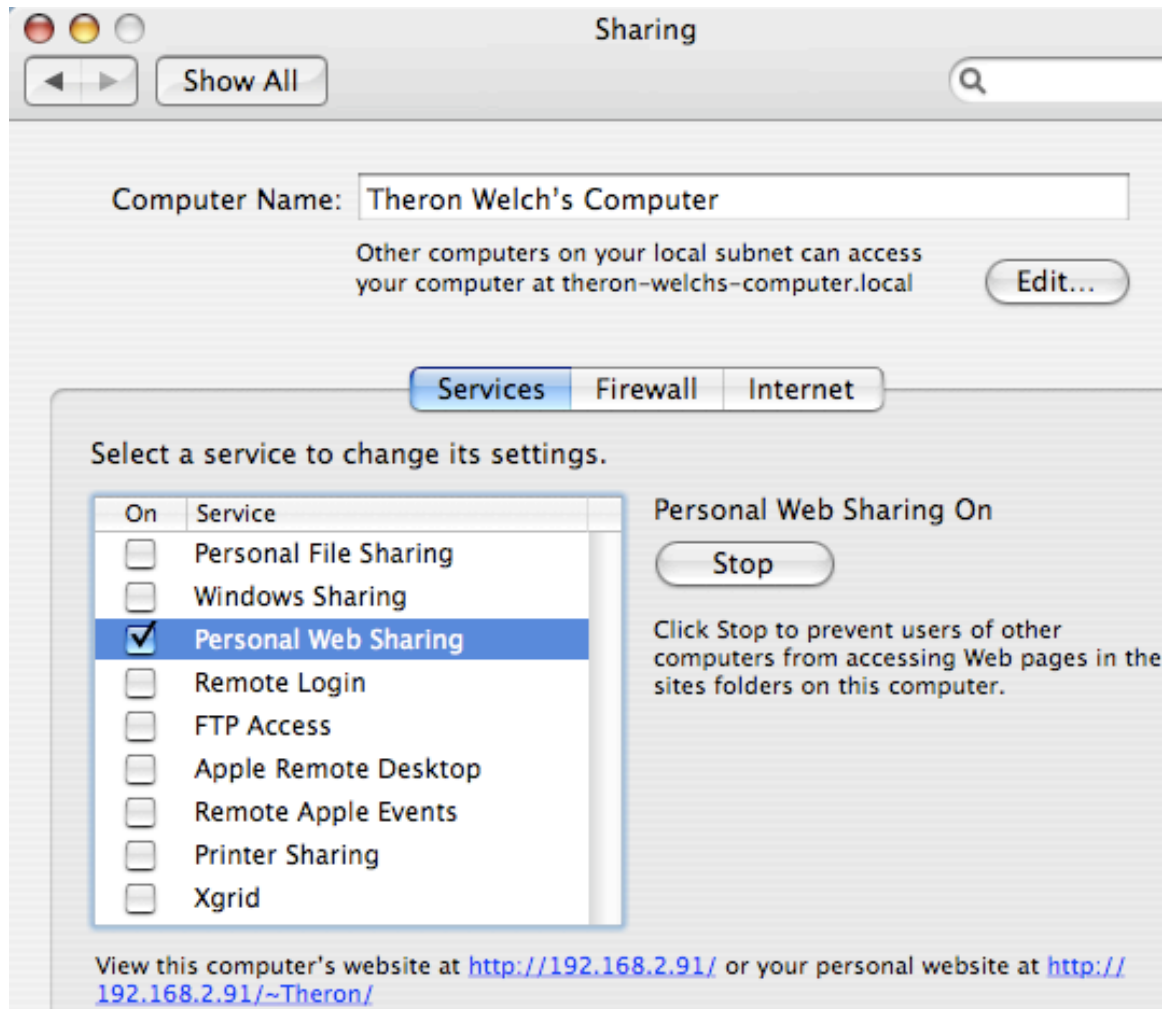
<http://www.entropy.ch/software/macosx/php/#install>

The installation was quite straightforward and there no little additional work needed.

Apache:

Lucky for us Mac users, Apache is already installed. The trick is figuring out how to make it start. The very first time I tested everything, my test worked because somehow the Apache server was already running. However, subsequent attempts failed because the server wasn't running. Uh, why not? I tried running it from the

command line, but it failed. When I looked up the specific message (I don't recall what the exact text was), it indicated that this particular error occurred when a different user from the one who installed it attempted to run it. At this point, I was quite confused at how to fix it. Well, on Windows, you start these sorts of services from the "Services" control panel applet. And, what's the equivalent of this on the Mac? The "Preferences Pane". Sure enough, I was able to start Apache here:



Testing the Three Products:

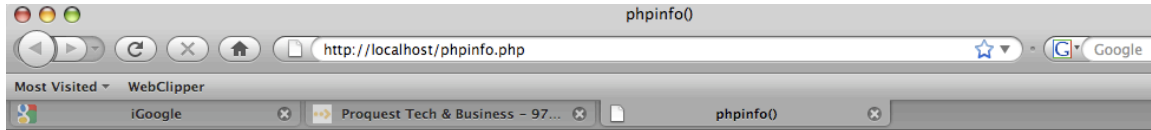
You can test that everything is working by creating a super simple PHP web page called "phpinfo.php" and saving it into your /Library/Web Server/Documents folder. Here's all the text you need in the file:


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

Navigate to it in your web browser with this address:

<http://localhost/phpinfo.php>

Here's what it looks like:



PHP Version 5.2.4 	
www.entropy.ch Release 1 (Universal Binary)	
System	Darwin theron-weichs-computer.local 8.10.1 Darwin Kernel Version 8.10.1: Wed May 23 16:33:00 PDT 2007; root:xnu-792.22.5~1/RELEASE_I386 I386
Build Date	Aug 31 2007 23:38:11
Configure Command	./configure '--prefix=/usr/local/php5' '--with-apxs' '--with-config-file-scan-dir=/usr/local/php5/php.d' '--with-iconv' '--with-openssl=/usr' '--with-zlib=/usr' '--with-gd' '--with-zlib-dir=/usr' '--with-ldap' '--with-xmlrpc' '--with-iconv-dir=/usr' '--with-snmp=/usr' '--enable-exif' '--enable-wddx' '--enable-soap' '--enable-sqlite-utf8' '--enable-ftp' '--enable-sockets' '--enable-dbx' '--enable-dbase' '--enable-mbstring' '--enable-calendar' '--enable-bcmath' '--with-bz2=/usr' '--enable-fastcgi' '--enable-cgi' '--enable-memory-limit' '--enable-zip' '--enable-pcntl' '--enable-shmop' '--enable-syssem' '--enable-sysvshm' '--enable-sysvmsg' '--with-curl=shared,/usr/local/php5' '--with-mysql=shared,/usr/local/php5' '--with-mysqli=shared,/usr/local/php5/bin/mysql_config' '--with-pdo-mysql=shared,/usr/local/php5' '--with-libxml-dir=shared,/usr/local/php5' '--with-xsl=shared,/usr/local/php5' '--with-pdflib=shared,/usr/local/php5' '--with-ldap=shared,/usr/local/php5' '--with-kerberos=/usr' '--with-ldap-ssl=/usr' '--with-jpeg-dir=/usr/local/php5' '--with-png-dir=/usr/local/php5' '--enable-gd-native-ttf' '--with-freetype-dir=/usr/local/php5' '--with-iodbc=shared,/usr' '--with-pgsql=shared,/usr/local/php5' '--with-pdo-pgsql=shared,/usr/local/php5' '--with-t1lib=/usr/local/php5' '--with-gettext=shared,/usr/local/php5' '--with-ming=shared,/usr/local/php5' '--with-mcrypt=shared,/usr/local/php5' '--with-mhash=shared,/usr/local/php5' '--with-mssql=shared,/usr/local/php5' '--with-fbsql=shared,/Users/liyanage/svn/entropy/universalbuild/src/FBDeveloperLibraries/Library/FrontBase' '--with-json=shared' '--enable-memcache' '--enable-openbase_module'
Server API	Apache
Virtual Directory Support	disabled
Configuration File (php.ini)	/usr/local/php5/lib

Finished:

Congratulations. If you're able to see the PHP Info in your web browser, everything is ready to go.

Bonus: Installing WordPress Locally

WordPress is famous for its "5 Minute Install" and, once I had the above three products working, I can confirm that it didn't take me much more than five minutes to install WordPress. The most difficult part was creating the database, but the exact commands that you need to create a database and a user are listed above.

So, go to www.wordpress.org and download the .zip file. Follow the instructions for the "5-Minute Install". Drop the .zip file's "wordpress" folder into the same folder as your recently created 'phpinfo.php' file. When you rename and edit the wp-config.php file, use the graphic below to set yourself up with correct database and login information.

Notice how the username that you specified MySQL is slightly different inside of the WordPress configuration file:

```
wp-config.php

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'testWP');

/** MySQL database username */
define('DB_USER', 'yomamma');

/** MySQL database password */
define('DB_PASSWORD', 'password');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
```

The final step in the WordPress instructions is to navigate to the “wp-admin.php” file. Here’s the full address, remember: <http://localhost/wordpress/wp-admin.php>

That’s basically it! Everything is working together now: Apache is serving up the web page, PHP is reading the code and connecting to your functioning MySQL database, and the WordPress code is rendering that lovely admin page:

