

WampServer

How to Set up and Get Started with PHP MySQL Web Development

1. Introduction

AMP is an acronym for Apache (an HTTP Server), MySQL (a relational database) and PHP (a server-side programming platform). These are industrial-strength, open-source software that collectively can be used to develop, deploy and run web applications. Depending on the operating platform, we have:

- WAMP: Windows-Apache-MySQL-PHP
- LAMP: Linux-Apache-MySQL-PHP
- MAMP: Mac-Apache-MySQL-PHP

You can choose to install the components individually, or use a *bundled* software package. For example,

- Zend Server (@ <http://files.zend.com/help/Zend-Server-Community-Edition/zend-server-community-edition.htm>) (for Windows, Linux, and Mac OS).
- WampServer (@ <http://www.wampserver.com/en/>) (Windows only).
- Easy PHP (@ <http://www.easyphp.org/>).
- XAMPP (@ <http://www.apachefriends.org/en/xampp.html>) (for Linux, Windows, Mac OS).
- Glossword WAMP (@ <http://glossword.biz/glosswordwamp/>) (Windows only).

I shall assume that you are familiar with Apache, MySQL and PHP. Otherwise, read "[How to setup and get started with Apache HTTP Server](#)", "[How to setup and get started with MySQL](#)".

2. Setting Up WampServer

WampServer (@ <http://www.wampserver.com/en/>) bundles Apache, MySQL and PHP for Windows in a single package. It also includes tools such as PhpMyAdmin (a MySQL database administration tool), SQL Buddy (an alternative to PhpMyAdmin), XDebug (a PHP Debugging Tool), WebGrind (a PHP Profiling Tool).

2.1 Installing WampServer

To Install WampServer:

1. **Download:** Goto WampServer mother site at <http://www.wampserver.com/en> ⇒ Select "DOWNLOADS" ⇒ Choose the appropriate package for your platform (e.g., WampServer 2.2e (32 bits)) ⇒ Download the installation file (e.g., wampserver2.2e-php5.4.3-httpd2.2.22-mysql5.5.24-32b.exe).
2. **Install:** To install the WAMPServer, simply run the downloaded installation file. Choose your installation directory. I shall assume that WampServer is installed in "d:\WampServer", denoted as <WAMP_SERVER_HOME>, in this article.

2.2 Starting WampServer

To start the WampServer, choose "start WampServer" from the "Start" menu; or run "wampmanager.exe" from the WampServer installed directory. An icon will appear on the icon tray. "Green" icon indicates that all the services have started. "Red" indicates that all the services have stopped. You can put the server online (so that user can access the server over the Internet) by selecting the option "Put Online". Alternatively, you could put the server "Offline", such that it is accessible only within the localhost.

2.3 Verifying the Installation

By default, a directory called "www" was created under the WampServer installed directory, which serves as the root directory of the Apache Server. A welcome page called "index.php" was also created in the "www" directory.

To verify the installation:

1. Start the WampServer. Wait for the icon to appear green (indicates that all services have started). Otherwise, click the icon and select "Start All Services". For testing, put the server "offline" to disable external access (i.e., it can be accessed only in localhost).
2. Start a browser and issue URL `http://localhost`. This will in turn request for "index.php". Observe the output and compare with "index.php" source.

2.4 Writing a Hello-world PHP Script

Use a program editor to write a simple PHP script called "hello.php" and save under the "www" directory, as follows:

```
<html>
<head>
<title>First PHP Page</title>
</head>
<body>
<?php
echo '<h1>Hello, world!</h1>';
?>
</body>
</html>
```

To run this script, start a browser and issue URL `http://localhost/hello.php`. Check the output produced via "view source". Take note that PHP is a server-side technology (instead of client-side technology like JavaScript). The PHP statements `<?php ... ?>` are processed in the server, and the results returned to the client (browser).

2.5 WampServer Directory Structure

WampServer is organized in the following directory structure:

- **bin**: contains binaries for Apache, MySQL, and PHP, in their respectively sub-directory. For each component, you can install multiple versions and select one of them in operation.
- **apps**: contains server-side tools such as PhpMyAdmin, SQL Buddy, and WebGrind.
- **tools**: contains client-side tool such as xdc (XDebug Client).
- **www**: is the apache server's root directory.
- **logs**: contains apache access and error logs; and mysql log files.
- **alias**: contains the the apache's alias configuration for PhpMyadmin, SQL Buddy and WebGrind.
- **scripts**:
- **lang**:
- **tmp**:

2.6 Apache Configuration

The Apache's configuration file "http.conf" is located at "`<WAMPSEVER_HOME>\bin\apache\Apache2.2.xx\conf`". Take note of the following configuration directives.

- The *core* directives are:

- `ServerRoot "<WAMPSEVER_HOME>/bin/apache/apache2.2.xx"`
- `Listen 80`
- `ServerName localhost:80`

```
DocumentRoot "<WAMPSEVER_HOME>/www/"
```

The `ServerRoot` specifies the apache's installed directory. `Listen` specifies the TCP port number for Apache web server, in this case, default of TCP port 80. The `DocumentRoot` specifies the root directory of the web server.

- The directives related to *directory authorisation* are:

- #
- # Each directory to which Apache has access can be configured with respect
- # to which services and features are allowed and/or disabled in that
- # directory (and its subdirectories).
- #
- # First, we configure the "default" to be a very restrictive set of
- # features.
- #
- <Directory />
- Options FollowSymLinks
- AllowOverride None
- Order deny,allow
- Deny from all
- </Directory>
- #
- # This should be changed to whatever you set DocumentRoot to.
- #
- <Directory "<WAMPSEVER_HOME>/www/">
- Options Indexes FollowSymLinks
- AllowOverride all
- # onlineoffline tag - don't remove
- Order Deny,Allow
- Deny from all
- Allow from 127.0.0.1

</Directory>

[TODO] Explanation

- The directives related to logging are:

- ErrorLog "<WAMPSEVER_HOME>/logs/apache_error.log"

CustomLog "<WAMPSEVER_HOME>/logs/access.log" common

2.7 PHP Configuration

The PHP's configuration file "php.ini" is located at "<WAMPSEVER_HOME>\bin\apache\Apache2.2.xx\bin".

[TODO] Explanation

2.8 MySQL Configuration

The MySQL's configuration file "my.ini" is located at "<WAMPSEVER_HOME>\bin\mysql\mysql5.5.xx", i.e., the MySQL installed directory. Take note of the following settings:

- MySQL Server
- [wampmysqld]
- port = 3306
- socket = /tmp/mysql.sock
- key_buffer = 16M
- max_allowed_packet = 1M
- table_cache = 64

- `sort_buffer_size = 512K`
- `net_buffer_length = 8K`
- `read_buffer_size = 256K`
- `read_rnd_buffer_size = 512K`
- `myisam_sort_buffer_size = 8M`
- `basedir=<WAMP_SERVER_HOME>/bin/mysql/mysql5.5.xx`
- `log-error=<WAMP_SERVER_HOME>/logs/mysql.log`

`datadir=<WAMP_SERVER_HOME>/bin/mysql/mysql5.5.xx/data`

The port specifies the TCP port number for the MySQL server (defaulted to 3306). The basedir specifies the MySQL installed directory. The datadir specifies the databases directory. The log-errorspecifies the MySQL error log file.

- MySQL Server Daemon (mysqld.exe)

- [mysqld]

`port=3306`

- Interactive Client (mysql)

- [client]

- #password = your_password

`port = 3306`

2.9 Customizing MySQL

The default MySQL installation provides a superuser called "root" without a password. It also provides an anonymous user with no password. You need to set a password for root, remove the anonymous user, and create a new user for operational use.

There are many ways to administer MySQL:

1. PhpMyAdmin
2. SQL Buddy
3. MySQL Interactive client ("mysql.exe")

SQL Buddy

To set password for superuser "root":

1. Login to SQL Buddy via URL `http://localhost/sqlbuddy`, with user "root" and empty password.
2. Select "Users".
3. Check the row of "User" of "root", "Host" of "localhost" and select "Edit".
4. Enter the new password in "Change password" box.
5. Repeat the above for `root@127.0.0.1` (IP address for localhost) and `root@::1` (IPv6 localhost address).

To remove the anonymous user:

1. Login to SQL Buddy with superuser "root".
2. Select "Users".
3. Check the row of anonymous user (empty user name), and select "Delete".

You should not use superuser "root" for normal operations. "root" should be used for creating new user only. You could create a new user for normal operation. To create a new user called "wampuser":

1. Login to SQL Buddy with user "root".
2. Select "Users".
3. In the "ADD NEW USER", enter "localhost" in "host" box, "wampuser" in "name" box, and "xxxx" in "password" box. Select the appropriate authorization option (do not check the "Grant Option"). Click "Submit" button.

PhpMyAdmin

[TODO]

MySQL Interactive Client (mysql)

Read "[How to setup MySQL](#)".

3. Eclipse PDT (PHP Developer Tool)

A good IDE with a graphic debugger is critical for program development.

Eclipse PDT (PHP Developer Tool) is an IDE for PHP program development. The main advantage is it can debug PHP script with XDebug or Zend Debugger. Read "[Eclipse PDT \(PHP Developer Tool\)](#)".

4. PhpMyAdmin and SQL Buddy

PhpMyAdmin (<http://www.phpmyadmin.net>) is an open source web-base tool intended to handle the administration of MySQL. SQL Buddy (<http://www.sqlbuddy.com>) is an alternative to PhpMyAdmin.

4.1 PhpMyAdmin

PhpMyAdmin is bundled in WampServer, installed under <WAMP_SERVER_HOME>\apps\phpmyadmin3.x.xx.

The configuration file is "config.inc.php", located at the PhpMyAdmin installed directory. The default configuration requires you to hardcode the MySQL user and password inside the configuration file:

```
$cfg['Servers'][$i]['auth_type'] = 'config'
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['password'] = 'your_root_password';
$cfg['Servers'][$i]['AllowNoPassword'] = true;
```

I recommend that you change the configuration to prompt user to login with password by changing the auth_type from "config" to "cookie".

```
// $cfg['Servers'][$i]['auth_type'] = 'config'
$cfg['Servers'][$i]['auth_type'] = 'cookie';
// $cfg['Servers'][$i]['user'] = 'root';
// $cfg['Servers'][$i]['password'] = 'your_root_password';
// $cfg['Servers'][$i]['AllowNoPassword'] = true;
```

To launch PhpMyAdmin, start a browser and issue URL <http://localhost/phpmyadmin>.

4.2 SQL Buddy

SQL Buddy is an alternative to PhpMyAdmin, which is also bundled in WampServer, under directory <WAMP_SERVER_HOME>\apps\sqlbuddy1.3.x. SQL Buddy could be easier to use than PhpMyAdmin.

To use SQL Buddy, start a browser, and issue URL <http://localhost/sqlbuddy>.

5. Setting up SSL Support for Apache Server

Step 1: Create a Certificate for the Web Server

The first step to set up SSL/TLS is to create a certificate for your web server. To do so, you need OpenSSL, which is an open-source software available at <http://www.openssl.org> (source distributions only). For windows user, you can download the OpenSSL binary from <http://www.slproweb.com/products/Win32OpenSSL.html> (you also need "Visual C++ 2008 Redistributables". The download link is given in the same page). Download and install OpenSSL. (WampServer also bundled a copy of OpenSSL in d:\WampServer\bin\php\php5.3.5\extras\openssl but is hard to use.)

Create a self-signed certificate for the server: Use OpenSSL to create a self-signed certificate for server by issuing the following command from a command shell. First of all, a public-private key pair needs to be generated. The private key is saved in the server in a secure file "server.key". The public key is store in a certificate "server.crt" to be transferred to the user.

```
> openssl req -new -x509 -days 36500 -sha1 -newkey rsa:1024 -nodes -keyout server.key -out server.crt  
-subj /C=SG/O=Company/CN=localhost
```

- -x509 requests a x509 certificate to be generated.
- -days 36500 sets the expiration period for the certificate. The default is 30 days. I set to 100 years.
- -sha1 specifies that SHA1 encryption should be used.
- rsa:1024 sets the key as 1024 bit RSA.
- -nodes specifies no passphrase is to be used for the private key file.
- -keyout and -out specify where to store the private key (server.key) and certificate (server.crt).
- -subj sets the country code (/C), company name (/O), and the web site address (/CN). If you leave these out, you'll be prompted for them. The CN (Common Name) must be the same as the address of your web site, otherwise the certificate won't match and users will receive a warning when connecting.
- Refer to http://www.modssl.org/docs/2.2/ssl_reference.html for more information about OpenSSL command syntax.

To view the content of a certificate (which contains the public key of the server), issue the following openssl command:

```
> openssl x509 -in server.crt -noout -text
```

Step 2: Configuring Apache HTTP Server

First of all, move the private key file (server.key) and certificate (server.crt) to the Apache's configuration directory (<WAMP_SERVER_HOME>/bin/apache/apache2.2.xx/conf).

To configure apache for SSL/TLS support for WampServer, simply uncomment the following lines in apache's configuration "httpd.conf" (under <WAMP_SERVER_HOME>/bin/apache/apache2.2.xx/conf):

```
LoadModule ssl_module modules/mod_ssl.so  
Include conf/extra/httpd-ssl.conf
```

The LoadModule loads the SSL module and the Include directive includes more configuration options for SSL/TLS support. Tailor "conf/extra/httpd-ssl.conf":

```
Listen 443  
NameVirtualHost *:443  
  
<VirtualHost *:443>  
DocumentRoot "<WAMP_SERVER_HOME>/wwwssl"  
ServerName localhost:443  
ErrorLog "<WAMP_SERVER_HOME>/bin/apache/Apache2.2.17/logs/error.log"  
TransferLog "<WAMP_SERVER_HOME>/bin/apache/Apache2.2.17/logs/access.log"  
SSLEngine on
```

```
SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
SSLCertificateFile "<WAMPSEVER_HOME>/bin/apache/Apache2.2.17/conf/server.crt"
SSLCertificateKeyFile "<WAMPSEVER_HOME>/bin/apache/Apache2.2.17/conf/server.key"

<Directory "<WAMPSEVER_HOME>/wwwssl">
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
# SSLRequireSSL
</Directory>

</VirtualHost>
```

Verifying SSL/TLS Installation

To verify Apache with SSL, start the Apache Server. Start a browser and issue `https://localhost`.

Because the server certificate is self-signed and not signed by a trusted CA (Certificate Authority), browser issues a warning.

For firefox: Select "I understand the risk" ⇒ "Add Exception" ⇒ "Confirm Security Exception".

What if...

In case of error in the installation:

- Check the Apache and SSL log.
- Try connecting to the Apache server via OpenSSL as follows:

```
> openssl s_client -connect localhost:443
```

If the connection succeeds then an HTTP command such as `"GET /"` to retrieve a web page.

More on Certificate

You may be able to get a free certificate from CAcert.org.

You can attached a passphrase (i.e., password) to the private key file. However, to start Apache, you need to either hardcode the passphrase in the apache's configuration file (same security exposure as no passphrase) or provide the passphrase during the start-up dialog (can't automate the apache start-up!).

To generate a certificate for signing by CA:

1. Generate a public-private key pair and a certificate request:

```
2. > openssl req -new -sha1 -newkey rsa:1024 -nodes
```

```
3. -keyout server.key -out www.example.com.csr
```

```
-subj /O=Company/OU=Department/CN=www.example.com
```

we didn't use the `-x509` switch. The command will therefore generate a public-private key pair and certificate request (in a `.csr` file), but not a certificate (`.crt` file).

4. Send that certificate request file `"www.example.com.csr"` to the CA (with your payment).
5. Rename the received certificate to `server.crt` and verify its contents:

```
> openssl verify -CAfile /path/to/trusted_ca.crt -purpose sslserver server.crt
```

Check that the certificate corresponds to your private key:

```
> openssl x509 -noout -modulus -in server.pem | openssl sha1
```

```
> openssl rsa -noout -modulus -in server.key | openssl sha1
```