SIMIAN systems

Setting up a Sitellite development environment on Mac OS X

Sitellite Content Management System

Introduction

Mac OS X is a great platform for web application development, and now with tools like VMWare Fusion and Parallels for running Windows applications inside your Mac, you can even do your Internet Explorer testing as effortlessly as testing the native web browsers. This is a huge plus for web developers on the Mac.

Plus, Mac OS X is Unix-based, so you gain the whole power of the command line under the hood (Applications > Utilities > Terminal.app) as well as the benefit of being on a very similar underlying platform as those you're likely to be deploying on (e.g., Linux, Mac or any Unix-based host). All of this makes the Mac a very compelling development platform for web applications.

This document will outline the process of setting up a complete Sitellite development environment on your Mac OS X desktop.

Apache and PHP

Mac OS X actually comes pre-installed with Apache and PHP, so all you need to do is enable them. To do this, go to your System Preferences then to Sharing under Internet & Network. From here, make sure Web Sharing is on. That's all there is to it, and your Mac will now start Apache and PHP for you automatically when you boot up.

For OS X 10.4 and earlier, you have to manually enable PHP as well by copying `/etc/php.ini.default` to `/etc/php.ini` and uncommenting these two lines in `/etc/httpd/httpd.conf`:

```
LoadModule php4_module libexec/httpd/libphp4.so
AddModule mod_php4.c
```

Make sure to restart Apache for this change to take effect. You can restart Apache in the Terminal with the following command:

```
sudo apachectl restart
```

Installing the MySQL database server

You can find MySQL for download on the MySQL website here:

```
http://www.mysql.com/
```

Go to the Developer Zone to find the Community Edition of the MySQL server and find the download link for the latest stable version of the

MySQL Server. At the time of writing, the latest stable version is 5.0.
Make sure to choose the "Mac OS X (package format)" download, which
will provide an installer to get up and running faster.

Once you have saved this to your downloads, double-click the down-
loaded .dmg file to mount it then double-click the installer .pkg file on the
mounted volume. The default options should mostly work fine, with the
following recommended changes:

1. Uncheck the "Enable Strict Mode", which may help prevent some
    SQL errors from the use of non-standard MySQL syntax.

2. I recommend leaving the character set to Latin1 even on multilingual
    sites. Sitellite's multilingual capabilities do not depend on the
    database for encoding/decoding, simply for storage, and this prevents
    several potential issues related to key lengths in MySQL.

When you have finished the installer, MySQL will be running in the
background of your computer.

Next, double-click the MySQLStartupItem.pkg file which will install the
MySQL startup item so that MySQL runs automatically when you first boot
up.

And finally, double-click the MySQL.prefPane file as well, which will install
and load the MySQL preference pane into your System Preferences.
This is where you can start and stop the MySQL server from.

If you are using OS X 10.4 or have issues connecting to MySQL in PHP,
edit your `/etc/php.ini` file and edit the following setting in the `[MySQL]`
block:

```
mysql.default_socket = /tmp/mysql.sock
```

Make sure to restart Apache for this change to take effect.

Editing your hosts file

By default, Apache runs a single website on your desktop, which is fine
for developing one website, but if you will be developing multiple websites
or new versions of the same website, it is often helpful to be able to run
multiple virtual sites from the same machine. To get set up to do this,
we'll first need to edit the operating system's hosts file.

The location of the hosts file on Mac OS X is in the
`/etc` folder, accessible through the Terminal. To edit this file, enter the
following commands into Terminal:

```
sudo bash
(enter your administrator password)
vi /etc/hosts
```

Alternately, if you are using BBEdit/Text Wrangler or TextMate for text editing you can type one of these instead of 'vi':

```
bbedit /etc/hosts
mate /etc/hosts
```

The format of the hosts file is very simple. Any line that begins with a # character is ignored by Windows and the other lines have the following format:

```
127.0.0.1     localhost
```

The first part is the IP address followed by the host name. Each entry should be kept on its own line. To add a new entry for your first development site, add a new line as follows:

```
127.0.0.1     www.mydevsite.lo
```

The 127.0.0.1 IP address is your computer's personal number for itself. It is always 127.0.0.1, even though the IP address of your machine as seen by other machines is different.

The name "www.mydevsite.lo" is a made-up domain name. It can be any name you wish, however I recommend using the ".lo" top-level name to signify that the site is on your own "local" machine and not a real site on the web. Any name at this extension is safe because ".lo" is not a real top-level domain.

After you've created several sites on your desktop, your hosts file will look something like this:

```
127.0.0.1     localhost
127.0.0.1     www.personalsite.lo
127.0.0.1     www.companysite.lo
127.0.0.1     www.clienta.lo
127.0.0.1     www.clientb.lo
127.0.0.1     www.clientc.lo
```

One thing to note, you shouldn't erase the original "localhost" entry as it may be expected by other programs on your computer.

Configuring Apache

The first step is to create a new folder inside your Sites folder for your first virtual host (a.k.a. website in Apache-speak). This is where you will create all of your websites on your Mac. To keep the names consistent,

let's call the new folder "mydevsite.lo". So the complete path to your website will now be `/Users/you/Sites/mydevsite.lo` where "you" is your OSX username.

The Apache configuration folder is either in `/etc/httpd` or `/etc/apache2` depending on your version of Mac OS X. To edit your Apache configuration to add the new "www.mydevsite.lo" domain to your computer, enter the following into Terminal:

```
sudo bash
(enter your administrator password)
vi /etc/httpd/users/username.conf
```

Note that "username" should be replaced with your actual Mac OS X username so that the file you're editing refers to your own personal Apache configuration file. And of course you can also edit with BBEdit or TextMate instead of 'vi' as well.

In this file, add the following:

```
<Directory "/Users/username/Sites">
    AllowOverride All
</Directory>


NameVirtualHost www.mydevsite.lo


<VirtualHost www.mydevsite.lo>
    DocumentRoot "/Users/username/mydevsite.lo"
    ServerName www.mydevsite.lo
</VirtualHost>
```

Anywhere you see "username" replace that with your real username.

For future websites, you can skip the first part and only include everything from "NameVirtualHost" on. You will also need to change the website name and document root values for each site to point to the new domain and folder that you create. Using this technique, you can install as many websites as you want for development on your Mac desktop machine. Very handy!

Before Apache will notice the changes, you will need to restart it first. This can also be done through Terminal with the following command:

```
sudo apachectl restart
(enter your administrator password)
```

After you restart Apache you should be able to load a web browser and go to your new website (http://www.mydevsite.lo/) and see the text "Index of /". If you are seeing that, then Apache is now ready to run your new website.

Creating a MySQL database

To create a new database for your website, open Terminal and enter the following command to connect to MySQL:

```
mysql -p -u root
```

By default, the password is blank which is okay since it's just a development machine and not the live site, however feel free to read through the MySQL documentation on setting a root MySQL password.

Once you've logged into MySQL, you should be presented with a "mysql>" prompt. Here you can type SQL commands directly into the MySQL database server.

To create a new database for your website, run the following commands:

```
create database mydevsite;
grant all on mydevsite.* to mydevsite@localhost identified
by 'PASS';
flush privileges;
quit;
```

This will create a new database named "mydevsite" with a username "mydevsite" who can connect to it and a password "PASS". Feel free to change the password, but make sure to remember it for the Sitellite installation, which is next.

Installing Sitellite on your virtual host

To download Sitellite, go to the Sitellite community website here:

```
http://www.sitellite.org/
```

Note: For professional edition users, you will have been sent a separate download link.

There are two options for installing Sitellite on Unix-based systems, so I'll just cover the Quick Installation method here since they're both documented on the above site already. In Terminal again, enter the following commands:

```
cd ~/Sites/mydevsite.lo
wget -q -O - http://www.sitellite.org/go-sitellite | sh
```

This will actually download and unzip the latest stable version of Sitellite into your website folder. When the command has finished running, open

your browser and go to the following address to launch the web-based installer to complete the installation process:

```
http://www.mydevsite.lo/install/
```

Follow the steps of the Sitellite installer, entering the database info for the database created earlier and choosing a default password for your Sitellite "admin" user. When the installer finishes, it will take you to the Sitellite login screen where you can log in for the first time to your new Sitellite installation!

Installing additional add-ons

You can find all of the Sitellite add-ons available for download from the following website:

```
http://www.sitelliteforge.com/
```

For instructions on installing a particular add-on, look in its "install" folder for instructions. They are typically in a file named "INSTALL" and typically follow these steps:

1. Unzip the add-on into Sitellite's "inc/app" folder

2. Rename the folder to just its base name (e.g., "sitesearch" instead of "sitesearch-2.0.0-stable")

3. Open Sitellite and copy and paste any SQL in the add-on's "install/install-mysql.sql" file into the SQL Shell in Sitellite's DB Manager

4. Copy any .php files in the add-on's "install" folder into Sitellite's "inc/app/cms/conf/collections" folder

You should then be able to find the new add-on in one of the following three places:

1. By going to Control Panel > Content you might see new content types provided by the add-on

2. By going to Control Panel > Tools you might see an administrative interface for the add-on

3. By going to "http://www.mydevsite.lo/sitesearch-app" where "sitesearch" is the name of the new add-on, any public interface to the add-on will display

4. In the Xed wysiwyg editor, you might find new options for the add-on in the Box Chooser dialog window

Troubleshooting and support

For troubleshooting information, please refer to the Troubleshooting documentation on the Sitellite community website, or write to the community for support in the Sitellite community forum, both at http://www.sitellite.org/