OpenEyes - Setting up a development environment on a Macintosh

Editors: G W Aylward
Version: 1.1:
Date issued: 9 May 2012

Target Audience

General Interest	
Heathcare managers	
Ophthalmologists	
Developers	✔

Amendment Record

Issue	Description	Author	Date
0.9	Draft	G W Aylward	29/12/2009
1.0	First release version	G W Aylward	16/06/2010
1.1	Updated for Lion	G W Aylward	09/05/2012

Table of Contents

Introduction

In 2001 Apple introduced a new operation system (OSX) based on UNIX. As a result, modern Macintosh systems represent an ideal development environment for OpenEyes, since most of the open source software employed in OpenEyes was developed and runs well on a UNIX operating system. This document provides a step by step guide to setting up a Macintosh computer as a development environment. When installing UNIX software, the use of MacPorts is recommended, as it greatly simplifies the steps required to install and update UNIX software. Familiarity with the use of the command line is essential for carrying out this task. The following table summarises the software required, and the function of each component.

Software	Function
Apache	Industry standard web server software
PHP	Widely used scripting language for dynamic web pages
MySQL	Backend relational database
OpenLdap	Implementation of LDAP server
Eclipse	Widely used open source integrated development environment

Mac Setup

The default formatting for an OSX disk is not case sensitive. To avoid compatibility problems with filenames to be transferred to or from a UNIX server, it is advisable to create a case sensitive partition to serve files from. The following commands will resize a partition on the fly as a case-sensitive volume, and replace the ‘Sites’ folder with a symbolic link;

```
diskutil list
```

Select the main OSX disk (in this case slice 2 (System))

This command will allocate 50 GB to the case sensitive partition, leaving 200.1 GB remaining

```
diskutil resizeVolume disk0s2 200.1G jhfsx Sites 50G
```

Put a sym link in your home folder using the following command;

```
 ln -s /Volumes/Sites Sites
```

Software Installation

Install Developer Tools

Install Xcode from the Apple web site or App store, then install the command line tools (from the Downloads section of the Xcode preferences). If you get compile errors during the next steps, you may also need to run the following command;

```
sudo xcode-select -switch /Applications/Xcode.app/Contents/Developer/
```

Installing MacPorts

Download the latest version of the MacPorts package installer from the MacPorts website. At the time of writing the required file was “MacPorts-2.0.4-10.7-Lion.dmg”. Double click on the downloaded file to open a disk image automatically, and run the installer software which will proceed with the installation (accept all defaults). This installs a skeleton structure in a separate directory (/opt/local), allowing you to install any UNIX software without interfering with items supplied with the operating system, which tend to be installed in /usr/local.

Once installation is complete, launch the Terminal application, as all subsequent steps will be run from the command line.

The next step is to update all the ports in the tree by running the following command;

```
sudo /opt/local/bin/port -v selfupdate
```

Installing the Apache web server

A version of Apache is provided with the standard installation. However this version is not updated using the OSX software update system, so it is recommended to install the latest version using MacPorts. This can be achieved as follows;

1. Turn off the Apple provided version

Turn off Apple’s "Personal Web Sharing" in the System Preferences so that the default Apache server is not running.

2. Install Apache

Return to the Terminal, and run the following commands to install the latest version of Apache;

```
cd /opt/local
sudo port install apache2
```

This will take some time (about 30 minutes), so go and make a cup of tea.

3. Configuring Apache

When installation is complete, you should edit the Apache configuration file by running the following command;

```
sudo edit apache2/conf/httpd.conf
```

FInd the following line,

```
DocumentRoot "/opt/local/apache2/htdocs"
```

and change it to the path to your 'Sites' folder, for example (replace 'bill' with your user name);

```
DocumentRoot "/Users/bill/Sites"
```

Repeat for the following line,

```
<Directory "/opt/local/apache2/htdocs">
```

Changing it to;

```
<Directory "/Users/bill/Sites">
```

Find the following line;

```
#ServerName www.example.com:80
```

and uncomment it, replacing the domain with localhost as follows;

```
ServerName localhost:80
```

Save the edited file, entering your password if requested by the editor software.

4. Running Apache

The following command will launch Apache, and will also make use of launchd to ensure it runs at boot time.

```
sudo launchctl load -w /Library/LaunchDaemons/
org.macports.apache2.plist
```

5. Testing Apache

To test that everything is working, use a text editor to save the following lines in a text file called index.html placed in your Sites sub-directory (eg: /Users/bill/Sites/index.html)

```
<html><body><h1>Apache is running successfully</h1></body></html>
```

Pointing your browser to http://localhost/ should display this file. Any subsequent edits of the Apache httpd.conf file should be followed by running the following command to force Apache to reload the configuation

```
sudo /opt/local/apache2/bin/apachectl -k restart
```

Installing MySQL

1. Install software

Change your directory back to /opt/local by typing at the command line

```
cd /opt/local
```

Install the latest version of MySQL server by typing the following;

```
sudo port install mysql55-server
```

2. Configure MySQL

Set up the main database;

```
sudo /opt/local/lib/mysql55/bin/mysql_install_db --user=mysql
sudo /opt/local/lib/mysql55/bin/mysqladmin -u root password
'password'
```

Rename link to mysql client to make it more familiar

```
 sudo ln -s /opt/local/lib/mysql55/bin/mysql mysql
```

3. Running MySQL

Launch MySQL server and ensure it auto starts when rebooting

```
sudo launchctl load -w /Library/LaunchDaemons/
org.macports.mysql55-server.plist
```

4. Configuring MySQL

Copy a template configuration file to my.cnf

```
sudo cp -Rfp /opt/local/share/mysql5/mysql/my-small.cnf /opt/
local/etc/mysql5/my.cnf
```

Edit the file to contain the contents below section.

```
sudo edit /opt/local/etc/mysql5/my.cnf
```

```
# The following options will be passed to all MySQL clients
```

```
[client]
port = 3306
socket = /opt/local/var/run/mysql55/mysqld.sock
default-character-set=utf8

# MacPorts default options
[mysqld]
port = 3306
socket = /opt/local/var/run/mysql55/mysqld.sock

# OpenEyes options
init_connect='SET collation_connection = utf8_general_ci'
init_connect='SET NAMES utf8'
character-set-server = utf8
collation-server = utf8_general_ci
innodb_file_per_table = 1
lower_case_table_names = 1
default_storage_engine = INNODB

# Skip-networking so multiple mysql server ports can be loaded
# without each competing for port 3306.
# One can always put the servers on different ports and
# remove or comment this option.
skip-networking

# Enable LOAD INFILE
local-infile=1
```

NB If you have problems with an openeyes migration with SQL insert faiilures, try commenting out the sql-mode line.

Re-start MySQL by stopping and restarting as in step 3. To stop MySQL use;

```
sudo launchctl unload -w /Library/LaunchDaemons/
org.macports.mysql55-server.plist
```

5. Set the root password

```
/opt/local/lib/mysql55/bin/mysqladmin -u root password 'new-password'
```

Installing PHP

1. Install software

Change your directory back to /opt/local by typing at the command line;

```
cd /opt/local
```

Install the latest version of PHP, including necessary extensions, by typing the following;

```
sudo port install php5 +apache2
sudo port install php5-mysql
sudo port install php5-sockets
sudo port install php5-gd
sudo port install php5-mbstring
```

The following extensions are also required to fully support Yii;

```
sudo port install php5-memcache
sudo port install php5-apc
sudo port install php5-mcrypt
sudo port install php5-soap
```

2. Let Apache know that PHP is available

The following commands register PHP with Apache and alters the httpd.conf file accordingly

```
cd /opt/local/apache2/modules
sudo /opt/local/apache2/bin/apxs -a -e -n "php5" libphp5.so
```

3. Edit the Apache configuration file using the following command

```
sudo edit /opt/local/apache2/conf/httpd.conf
```

Add the following lines to the end of the file

```
#
# Include PHP configurations
#
Include conf/extra/mod_php.conf
```

Restart Apache to read the new httpd.conf file

```
sudo /opt/local/apache2/bin/apachectl -k restart
```

4. Create a custom php.ini file

Copy the supplied development ini file to the active location;

```
sudo cp /opt/local/etc/php5/php.ini-development /opt/local/etc/
php5/php.ini
```

5. Configure PHP

Edit the PHP.ini file by typing;

```
sudo edit /opt/local/etc/php5/php.ini
```

Find the 'Module Settings' section and uncomment the date.timezone line, making it equal to GMT;

```
date.timezone = GMT
```

Find the line beginning 'pdo_mysql.default_socket=' and edit it to equal the default MySQL socket;

```
pdo_mysql.default_socket=/opt/local/var/run/mysql5/mysqld.sock
```

Similarly for the line beginning 'mysql.default_socket =';

```
mysql.default_socket = /opt/local/var/run/mysql5/mysqld.sock
```

And for 'mysqli.default_socket =';

```
mysqli.default_socket = /opt/local/var/run/mysql5/mysqld.sock
```

Also, find and uncomment the 'error_log' line and add a location for an error log;

```
error_log = /opt/local/var/log/php_errors.log
```

For this to work, you will need to create the file and change ownership to www;

```
cd /opt/local/var/log
sudo touch php_errors.log
sudo chown www php_errors.log
```

Restart Apache;

```
sudo /opt/local/apache2/bin/apachectl -k restart
```

6. Testing PHP

To test that everything is working, use a text editor to save the following lines in a text file called phpinfo.php placed in your Sites sub-directory (eg: /Users/bill/Sites/phpinfo.php)

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

Point your browser to http://localhost/phpinfo.php and a very long page of information about your php installation should be displayed.

Eclipse IDE

Eclipse is a fully equipped integrated development environment including advanced features. and is open Source,

1. Install Eclipse

Go to http://www.eclipse.org/downloads and download the version of eclipse for PHP developers most suited to your system (eg Mac Cocoa 64bit). Double click on the downloaded tar file to decompress it, and a folder called ‘eclipse will appear. Drag this to your Applications folder, and run it from there

Set up a folder for workspaces in a different location to the ‘Sites folder’, and make this your default workspace location. Create new projects there, but import and create files in a folder of the same name in the Sites folder, so that they can be served by Apache.

I order to get UK formatted dates, go to System Preferences, load ‘Language and Text’, and click the ‘Edit list’ button. Scroll to find ‘British English’ and check the box. The next time you launch Eclipse, it should use UK date. In about Eclipse, Installation details, Configuration, there should now be a line like;

```
user.country=GB
```

Debugger

You have two choices of debugger to use with Eclipse, and the instructions for both now follow. The author uses the second of these options.

Xdebug

Return to the terminal and type;

```
cd /opt/local
sudo port install php5-xdebug
```

Edit php ini tile so that php knows about xdebug

```
sudo edit /opt/local/etc/php5/php.ini
```

Add the following lines to the end and save the file

```
;;;;;;;;;;;;;;;;;;;
; Xdebug        ;
;;;;;;;;;;;;;;;;;;;
zend_extension=/opt/local/lib/php/extensions/no-debug-non-zts-20090626/xdebug.so
xdebug.remote_enable=1
xdebug.remote_host="localhost"
xdebug.remote_port=9000
xdebug.remote_handler="dbgp"
xdebug.profiler_enable = Off
xdebug.default_enable = On
```

Restart Apache;

```
sudo /opt/local/apache2/bin/apachectl -k restart
```

Zend Debugger

Go to http://www.zend.com/en/products/studio/downloads and download the latest version of the ZendDebugger (eg Studio Web Debugger, Mac OSX 64 bit version, depending on your system). Double click on the downloaded file which will unpack a directory. Within it, find the version of ZendDebugger.so compatible with version 5.3 of PHP. Copy this to the PHP extensions directory using the following command in the terminal;

```
sudo cp /Users/bill/Downloads/ZendDebugger-5.2.26-darwin9.5-
x86_64/5_3_x_comp/ZendDebugger.so /opt/local/lib/php/extensions/
no-debug-non-zts-20090626/
```

Edit the PHP.ini file by typing;

```
sudo edit /opt/local/etc/php5/php.ini
```

Include the following lines at the end of the file

```
;;;;;;;;;;;;;;;;;;;
; Zend Debugger   ;
;;;;;;;;;;;;;;;;;;;
zend_extension="/opt/local/lib/php/extensions/no-debug-non-zts-20090626/ZendDebug
ger.so"
zend_debugger.allow_hosts=127.0.0.1
zend_debugger.expose_remotely=always
```

Save the file, and restart Apache;

```
sudo /opt/local/apache2/bin/apachectl -k restart
```

Configure Eclipse

Launch Zend Studio, and open the preferences. Change the following

PHP>PHO Interpreter>PHP Version: PHP 5.3