OpenEyes - Installing OpenEyes on a Macintosh

Editors: G W Aylward
Version: 0.9:
Date issued: 15 June 2012

Target Audience

General Interest	✔
Heathcare managers	
Ophthalmologists	✔
Developers	✔

Amendment Record

Issue	Description	Author	Date
0.9	Draft	G W Aylward	15/06/2012

Table of Contents

Introduction

This document gives detailed instructions on how to create a standalone installation of OpenEyes on a Macintosh. It assumes that the following software is already installed. Instructions on how to install these is given in a companion document.[1]

Software	Function
Apache	Industry standard web server software
PHP	Widely used scripting language for dynamic web pages
MySQL	Backend relational database

Create a virtual domain

We recommend changing the default Apache configuration so that OpenEyes runs in it's own subdomain. This is a more supported configuration in terms of how the code is being managed, and it will mean not having to move the .htaccess file each time you pull new code. The following steps will create a subdomain called 'openeyes.localdomain'.

1. Edit the Apache configuration files

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

Scroll to the last page of the file, and uncomment the second of the following two lines;

```
# Virtual hosts
#Include conf/extra/httpd-vhosts.conf
```

Apache will then look at an additional configuration file when it loads. Now edit this file;

```
sudo edit /opt/local/apache2/conf/extra/httpd-vhosts.conf
```

remove the Virtual host example, and replace with two, the first of which should look like this;

```
# Localhost
<VirtualHost *:80>
    ServerName localhost
    <Directory "/Users/bill/Sites">
        Options Indexes FollowSymLinks
        AllowOverride None
```

```
            Order allow,deny
            Allow from all
        </Directory>
    </VirtualHost>
```

Ensure you replace "/Users/bill/Sites" with the path to the directory that you want to serve websites from. This section will enable other websites in your 'Sites' directory to continue as normal. Now add a second virtual host;

```
    # Local virtual server for OpenEyes
    <VirtualHost *:80>
        ServerName openeyes.localdomain
        DocumentRoot /Users/bill/Sites/openeyesdev/openeyes

        <Directory /Users/bill/Sites/openeyesdev/openeyes>
            Options FollowSymLinks
            AllowOverride All
            Order allow,deny
            Allow from all
        </Directory>

        RewriteEngine On
        RewriteRule "^(.*/)?\.git/" - [F,L]

        ErrorLog /var/log/apache2/openeyes-error.log
        LogLevel warn
        CustomLog /var/log/apache2/openeyes-access.log combined
    </VirtualHost>
```

2. Restart Apache

Force Apache to re-read the configuration files by re-starting it;

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

Install OpenEyes

1. Create a directory

Decide where you want to serve openeyes from, and create an appropriate directory. This must be the same path as specified in the httpd-vhosts.conf file.

```
cd ~/Sites
mkdir openeyesdev
```

2. Install Yii

Download and install version 1.1.8 of the Yii framework from here. Currently OpenEyes is not supporting versions of Yii later than 1.1.8.

Copy it into your working folder renaming it as yii

```
cp -Rfp ~/Downloads/yii-1.1.8.r3324 ~/Sites/openeyesdev/yii
```

3. Install OpenEyes

Change directory and use git clone to download the latest version of the OpenEyes code.

```
cd ~/Sites/openeyesdev
git clone https://github.com/openeyes/OpenEyes.git openeyes
```

By default, the development branch is downloaded, but you can check which branch you are in using the following commands;

```
cd openeyes
git branch
```

This should show ;

```
*development
```

4. Configure OpenEyes

Create copies of the index.php file and the configuration file;

```
cd ~/Sites/openeyesdev/openeyes
cp index.example.php index.php
```

```
cp protected/config/local/common.sample.php protected/config/local/common.php
```

Edit the new config file;

```
edit protected/config/local/common.php
```

Enter the details of your database, including username and password

```
'db' => array(
    'connectionString' => 'mysql:host=localhost;dbname=openeyes',
    'username' => 'username',
    'password' => 'password',
    ),
```

It is also suggested that you change the setting of the environment parameter from 'live' to 'dev' which will display debugging information in the browser;

```
    'ldap_dn' => 'CN=Users,dc=example,dc=com',
    'environment' => 'dev',
    //'audit_trail' => false,
```

5. Set permissions

Create the runtime directory;

```
mkdir ~/Sites/openeyesdev/openeyes/protected/runtime
```

Set appropriate permissions to allow Apache to access the framework and files

```
cd ~/Sites/openeyesdev/
chmod 775 openeyes/assets openeyes/cache openeyes/protected/cache openeyes/protected/runtime
```

Add the Apache _user to the staff group to allow it to read the files in the framework and the installation;

```
dseditgroup -o edit -u bill -p -a _www -t user staff
dseditgroup -o checkmember -u _www staff
```

If you intend to use the Gii module generator, you will need to allow _www to write to the modules subdirectory;

```
cd ~/Sites/openeyesdev/openeyes/protected
chmod 775 modules
```

Setup the Database

1. Create the database and set its permissions

Use a mysql client to create the database, and ensure it has the appropriate permissions (same username and password as in your config.php file.

```
mysql -u root -p
CREATE DATABASE openeyes;
GRANT ALL PRIVILEGES ON openeyes.* TO 'username'@'localhost'
IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
EXIT
```

2. Install sample data

Install the sample data module and import the sample data set.

```
cd ~/Sites/openeyesdev/openeyes/protected/modules
git clone https://github.com/openeyes/Sample.git sample
bzcat ~/Sites/openeyesdev/openeyes/protected/modules/sample/
dump/openeyes_0.10.0_sample.sql.bz2 |mysql -u root -p -D
openeyes
```

The last command will prompt you for your mysql root password.

3. Run the migrations to set up the database schema

```
cd ~/Sites/openeyesdev/openeyes/protected
./yiic migrate
```

Install Optional Modules

OpenEyes will not do much without any event types, which (apart from booking) are loaded as optional modules. The following steps are required to install the operation note module and EyeDraw. Installation of subsequent modules will be similar, and will be covered by separate installation manuals.

1. Download the module

This event type allows recording of operation notes, and also depends on the EyeDraw module being loaded. These two modules are installed as follows.

```
cd ~/Sites/openeyesdev/openeyes/protected/modules
git clone https://github.com/openeyes/EyeDraw.git eyedraw
git clone https://github.com/openeyes/OphTrOperationnote.git OphTrOperationnote
```

2. Add to the module array in the config file

Yii needs to know about which modules to load so add this to the configuration file.

```
edit ~/Sites/openeyesdev/openeyes/protected/config/local/common.php
```

The array should look like this after editing

```
'modules' => array(
        'OphTrOperationnote',
    ),
```

Note that only the event type modules need to be added here, EyeDraw provides widgets that the event type modules can use but isn't an event type module itself and so doesn't need to be included in this array.

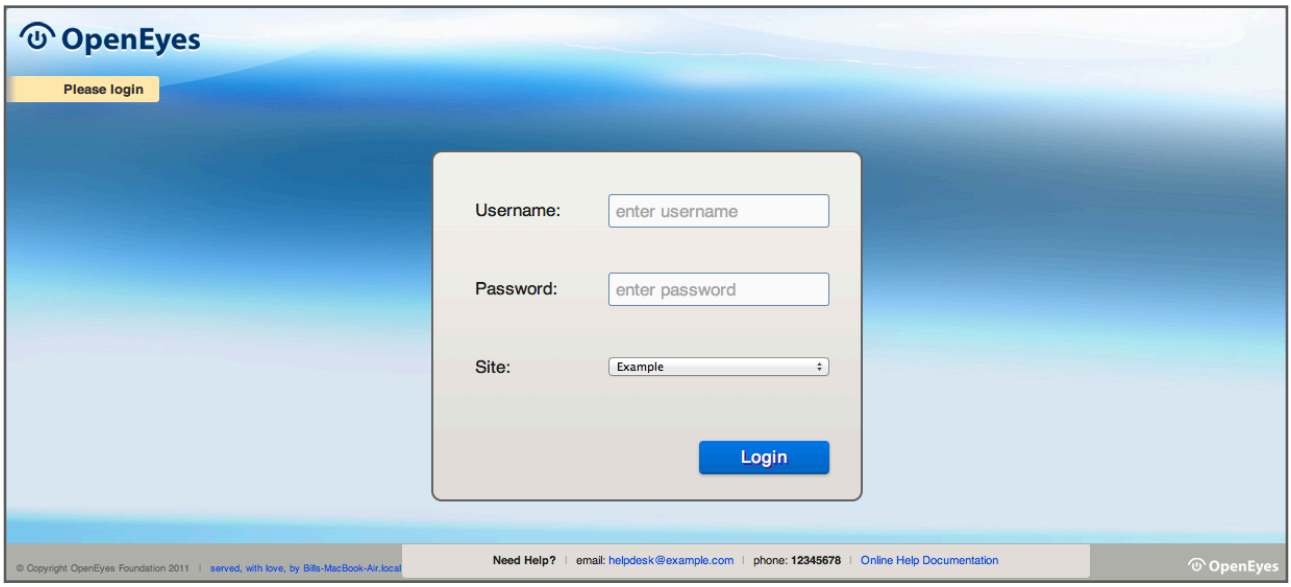
3. Run module specific migrations

The following commands will run any database migrations required for this module

```
cd ~/Sites/openeyesdev/openeyes/protected
./yiic migrate --migrationPath=application.modules.OphTrOperationnote.migrations
```

Test your setup

Point your browser to here and you should see the following login screen;



Login using the username and password ‘admin’ and ‘admin’. There are a number of test patients with numbers from 1000001 and above.

References

1. OpenEyes Mac Development Setup - link

Appendix 1

Configuration settings

In the common.php file there is a 'params' section which has various configuration parameters for the OpenEyes application. You should set these according to your requirements. This is a full list of the available options and what they do:

Parameter	Description
use_pas	Whether to use a PAS system defined in the db_pas clause (discussed in the previous step). This should be set to false if you do not wish to use a PAS or true if you do.
pseudonymise_patient_details	f set to true this will randomise any patient data coming back from a PAS system. This is useful if you need to run OpenEyes with a PAS database connection in an environment which is not completely secure. All patient data is randomised before being stored in the openeyes database so any access to the OpenEyes installation will not cause real patient data to be leaked.
ab_testing	This flag disables some of the authentication mechanisms in order to perform load testing with the Apache benchmark utility 'ab'. This should be set to false unless you specifically need to perform load testing. When enabled, most authentication mechanisms in the application are disabled so it should never be enabled when the system contains real patient data.
auth_source	This defines how users will authenticate to OpenEyes, possible values are: BASIC - users are authenticated against the local openeyes database only using SHA1 password hashes. LDAP - users are authenticated against an LDAP server, such as an Active Directory domain controller. If auth_source is set to 'LDAP', the following additional options are available to configure the LDAP connection: (ldap_server, ldap_port, ldap_admin_dn, ldap_password, ldap_dn) These define the LDAP connection according to the LDAP standard. If you need help configuring these parameters you should contact the administrator of your LDAP server or Active Directory domain controller.

Parameter	Description
environment	This defines whether the OpenEyes instance is a live instance or a development instance and how errors are handled. Possible values are: dev - Application errors are displayed in full in the browser with as much detail as possible, stack traces, which line of code caused the fault, etc. live - Application errors are trapped and logged to the application log (at protected/runtime/application.log) and a static error page is displayed to the user. The static error pages for both 404 and 500 errors are in protected/views/. The pages display a helpdesk phone number and email address which are defined by the configuration parameters helpdesk_phone and helpdesk_email respectively.
audit_trail	If set to true a log of all MySQL operations will be stored in the database table tbl_audit_trail. This is useful for debugging data issues but adds a lot of overhead so for most instances it is recommended to leave this set to false.
watermark	This allows a watermark banner to be displayed on all pages of the OpenEyes application. The banner is shown as white text on a red bar at both the top and bottom of the page. If this is desired, you simply set the watermark to the text you want to be displayed. This option is useful to display messages to users of the application regarding system problems, planned maintenance etc.
watermark_admin	The same as watermark but is only displayed to users with the admin role.
watermark_description	Similar to watermark but is shown as an additional pink banner underneath the top watermark banner, and only on the login page.
google_analytics_account	Allows a Google Analytics account to be used to collect visitor statistics from OpenEyes. If you want to use Google Analytics you should set this to the identifier of the site as it's configured in your Analytics account. This is usually an alphanumeric code beginning with 'GA'.
local_users	Defines a list of users to be authenticated locally when LDAP authentication is enabled. The default users 'admin' and 'username' are typically in this list.