



Free and Open Source software to store the electronic medical record

Installation

Creating a LAMP server

Installation summary (for experienced LINUX users)		
Ubuntu	Linux operating system	currently using Ubuntu 11.04
SSH	network protocol for remote administration of Unix/Linux computers	<i>sudo apt-get install ssh</i>
Apache2	a network protocol/web server for remote administration of Unix computers (directs incoming web-pages to the correct service on the computer)	<i>sudo apt-get install apache2</i>
PHP-5	computer programming language	<i>sudo apt-get install php5 libapache2-mod-php5</i>
MySQL Server	database engine	<i>sudo apt-get install mysql-server</i>
phpMyAdmin	GUI portal for MySQL	<i>sudo apt-get install phpmyadmin</i>
gPHPedit	editor for PHP-5	<i>sudo apt-get install gphpedit</i>

Installing Apache, MySQL and PHP on Ubuntu

To produce a server LAMP installation on a PC Ubuntu system, including Apache 2, PHP 5, MySQL 5 and phpMyAdmin, it is necessary to use only one command to have a comprehensive and reliable system:

```
sudo apt-get install apache2-mpm-prefork mysql-server libapache2-mod-php5 php5-mysql phpmyadmin
```

In addition to the Ubuntu root password, you will be requested to supply a MySQL password for the root user of the MySQL database engine. Strong passwords will be needed on Internet servers. During the installation, do not forget to enter the type of server that phpMyAdmin must interface: this is 'Apache'. If you forget, you must patch the file `/etc/apache2/apache2.conf` to include the line: `Include/etc/phpmyadmin/apache.conf` and restart Apache2 with `/etc/init.d/apache2 restart`

After that the following software will be installed and ready for use:

- Web server Apache 2
- MYSQL database server
- PHP 5 with the Apache module and the gd library
- phpMyAdmin for an easy database management

It is also a good idea to install a PHP editor if you are going to be creating/editing PHP scripts. This can be done with:

```
sudo apt-get install gPHPedit
```

Setting up Apache 2

If you follow the instructions above, the system is fully operational. However, a small correction to the configuration has many benefits in a few simple steps. This is the DocumentRoot of the Apache server, which by default is located in `/var/www`, but it would be better to move it in your Home directory for several reasons, including the issue of file permissions, a better management of files and a more efficient procedure in the event of a crash or system backup. It doesn't require particular LINUX skills, although you should have some idea how to use the LINUX terminal. In your Home folder, create a directory called `www`, that will be your web root:

```
mkdir /home/user/www
```

Obviously, from now on, remember to replace `user` with your Home name.

In Apache2, there are separate configurations for each site that the web will go to, loaded in `/etc/apache2/sites-available`. By default, there is only one site available, called default, but you can load multiple configurations simultaneously, activating from time to time only those necessary.

In our case, we must create a new site and define the DocumentRoot (`/home/user/www/`) and then activate it for Apache2.

Creating the new site

1. Modify the Apache configuration file with a notepad, for example:

```
sudo gedit /etc/apache2/sites-available/...
```
2. Change the DocumentRoot parameter so it will point to the new location, eg. /home/user/www/.
3. Change the Directory instruction, replacing <Directory /var/www/> with <Directory /home/user/www/>
4. Save the file
5. Restart Apache2:

```
sudo /etc/init.d/apache2 restart
```

If you have forgotten to create the directory /home/user/www/, a warning message will appear. By opening a browser to <http://localhost/> you should be redirected to the content of ~/www/.

This is a basic configuration, best suited for local tests, not for Web server usage. To increase security and add some customization, you can modify the main Apache 2 configuration file, located in /etc/apache2/apache2.conf. Remember to create a backup of this file before you start changing it – it may never work again! You must also restart Apache2 for changes to take effect. For more information, visit the Apache site on the web.

For managing the configuration of PHP, php.ini is the configuration file and is located in /etc/php5/apache2.

Installing the MDS6 database

The empty database that is used to start using is provided on the distribution CD – it ends with '.sql'. Copy this file to your hard disk in some easy place to find (e.g. in the Desktop or the home folder where you refer to it as ~/xxxxxxxxx.sql).

Using the LINUX terminal, enter:

```
sudo mysql -h 127.0.0.1 -u root -p
```

You will enter to give the Ubuntu root password and the password you gave to the MySQL database during installation.

Once in the command mode for MySQL, enter for example:

```
source /home/user/Desktop/MDSFoss_Beta.sql
```

For 'user' you may need to substitute another word, your user name in Linux. The .sql file may also have another name.

Watch for error messages. If it is not clear what they mean, assistance is available on the hotline of NetCom Technologies (+94 65 364 2477).

Setting up PHP 5

If PHP was installed using apt-get or with the Synaptic Package Manager, it will be installed automatically and ready to run. If Apache2 is correctly installed (see above) it will recognise PHP scripts (either stand-alone or embedded in HTML) and pass them correctly on to the PHP processor. There are a large number of extensions to the PHP language that can be called easily by the 'include' command that provide specialized functionality. The production of PDF-files for the reports in MDSFoss is an example of this – they use functions from the class 'FPDF' published as free software by Olivier Plathey. PHP itself is also free software released under the GPL License.

The PHP programming language

This is the language that MDSFoss uses to communicate with the MySQL database. PHP is a general-purpose scripting language that was originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into HTML source documents and interpreted by a web server with a PHP processor module (in our case Apache2) that generates the web page document. The developers of MDSFoss use gPHPEdit as the editor for generating PHP scripts. As a general-purpose programming language, PHP code can also be processed by an interpreter application in command-line mode – this is not currently used in MDSFoss.

The PHP interpreter only executes PHP code found in an HTML screen within its delimiters. Anything outside its delimiters is not processed by PHP (although non-PHP text is still subject to control structures described within PHP code). The most common delimiters are <?php to open and ?> to close PHP sections. The purpose of these delimiters is to separate PHP code from non-PHP code, such as HTML.

Variables are prefixed with a dollar symbol and a type does not need to be specified in advance. Unlike function and class names, variable names are case sensitive. Both double-quoted (") and heredoc strings allow the ability to embed a variable's value into the string. PHP treats newlines as whitespace in the manner of a free-form language (except when inside string quotes), and statements are terminated by a semicolon. PHP has three types of comment syntax: /* */ marks block and in-line comments; // and # are used for one-line comments. The echo statement is one of several facilities PHP provides to output text (e.g. to a web browser).

In terms of keywords and language syntax, PHP is similar to most high level languages that follow the C-style syntax. IF conditions, FOR and WHILE loops and function returns are similar in syntax to languages such as C, C++, Java and Perl.