T: 0191 222 5000
E: enquiries@netskills.ac.uk
www.netskills.ac.uk

Installing PHP & MySQL

CSC3504 Assignment 4

Installing PHP

Now you have the **L**(Linux) and **A**(Apache) parts of your LAMP stack operational, the next stage is to add the **P**(PHP).

PHP uses the same compilation and install method as Apache to create two things. Firstly it creates the PHP executables (i.e. "PHP itself"). You also get `mod_php` the PHP module that Apache uses to access and run PHP scripts. This (`mod_php`) is created as a dynamic (or shared) module which will exist as a separate file and is loaded into Apache every time it starts.

You will find the PHP documentation handy for this as well:

http://www.php.net/manual/en/

You are installing on "Apache 2.x on Unix systems"....

Prepare Apache

If you haven't already done so, make sure that you have added in support for DSO modules by recompiling Apache e.g.

```
./config.nice --enable-so
```

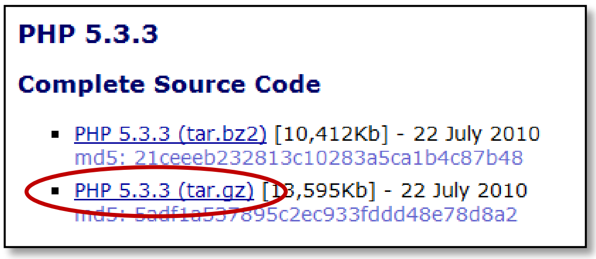
Grab the source code

You will need to download the PHP source to your VM using one of the two methods you used during your initial Apache build.

You get the source from:

```
http://www.php.net/downloads.php
```

You need the `tar.gz` download of the current version (now `5.3.8`)



PHP 5.3.3

Complete Source Code

- PHP 5.3.3 (tar.bz2) [10,412Kb] - 22 July 2010
  md5: 21ceeeb232813c10283a5ca1b4c87b48
- PHP 5.3.3 (tar.gz) [13,595Kb] - 22 July 2010
  md5: 5adf1a537895c2ec933fddd48e78d8a2

Figure 1. PHP download

Unpack the source code to your `build` directory (again refer to your Apache instructions for some reminders of how to do this).

© JISC Netskills 2011

Configure your source tree and install

Use the configure script that came with your PHP source to start the installation process (i.e. switch to your PHP source directory and use `./configure` in there)

Make sure to include the following configure options:

- The path to the Apache binary for the **Ap**ache E**x**ten**s**ion Tool (`apxs`)
- The path information to explicitly install PHP to `/usr/local/php`

HINT: You can view the possible configure options by running:

```
./configure --help
```

Once you have successfully configured then you can `make` and `make install` PHP.

Check basic server configuration

Apache

Make sure `httpd.conf` includes instructions to:

- Load the `libphp5.so` file (i.e. `mod_php`)
- Use `mod_php` to handle requests for files ending in `.php`

PHP

Make sure you also create a `php.ini` file in `/usr/local/php/lib`

- Do this by taking a copy of `php.ini-production` from the source directory (renaming it as `php.ini` of course!)

Get PHP running

The goal of this part of this assignment is to deliver a PHP script via the web as follows:

1. Create a file called `index.php` in `/usr/local/www/php`
   (i.e. create the directory too!)
2. Your `index.php` file must return the results of `phpinfo()` showing that:
   - PHP errors will be displayed
   - The time zone has been correctly set to Europe/London
   - Short opening tags are permitted in PHP scripts

   You will need to edit your php.ini file to achieve these
3. Your `index.php` file must be automatically returned as the directory index i.e. it can be accessed directly at:

   `http://vm-eliot-NNN.ncl.ac.uk/php/`

Adding MySQL

The **M** in LAMP… MySQL… is already installed on your VM.

In order to add support for it to the stack you will need to correctly build PHP to work with it.

You shouldn't need anything from the MySQL documentation for these tasks, but for reference it is at:

http://dev.mysql.com/doc/

You are using version 5.0, which you will find at:

http://dev.mysql.com/doc/refman/5.0/en/

There is also a section devoted to working with PHP:

http://dev.mysql.com/usingmysql/php/

Check MySQL

First, check the required bits of MySQL are on your VM by asking **yum** (the package manager for CentOS) to list all the installed packages containing MySQL by typing:

```
yum list installed mysql*
```

You should see something like:

```
[a1234@vm-eliot-000 ~]# yum list installed *mysql*
Installed Packages
mysql.i386                 5.0.77-4.el5_5.4        installed
mysql.x86_64               5.0.77-4.el5_5.4        installed
mysql-devel.i386           5.0.77-4.el5_5.4        installed
mysql-devel.x86_64         5.0.77-4.el5_5.4        installed
mysql-server.x86_64        5.0.77-4.el5_5.4        installed
```

The version numbers may be slightly different (higher) on your install.

Check the local build environment

Next check that your build environment knows that you are using a 64-bit version of MySQL by typing:

```
env
```

Somewhere in the output (the O/S environment for the user you are running as) – probably at the top, you should find the line:

```
LDFLAGS=-L/usr/lib64/mysql
```

This has been pre-set on your VMs so you don't have to remember to add it in every time. It will also write itself into the **config.nice** files made by PHP too ☺

© JISC Netskills 2011

Recompile PHP to add MySQL support

You will need to recompile PHP to build the libraries it needs to communicate with MySQL.

This will require you to add the following options to your existing configuration

```
--with-mysql=/usr/lib64/mysql/mysql_config
```

and

```
--with-mysqli=mysqlnd
```

[*HINT* remember PHP also creates a `config.nice` file, so you should be able to tack these new options onto it ☺]

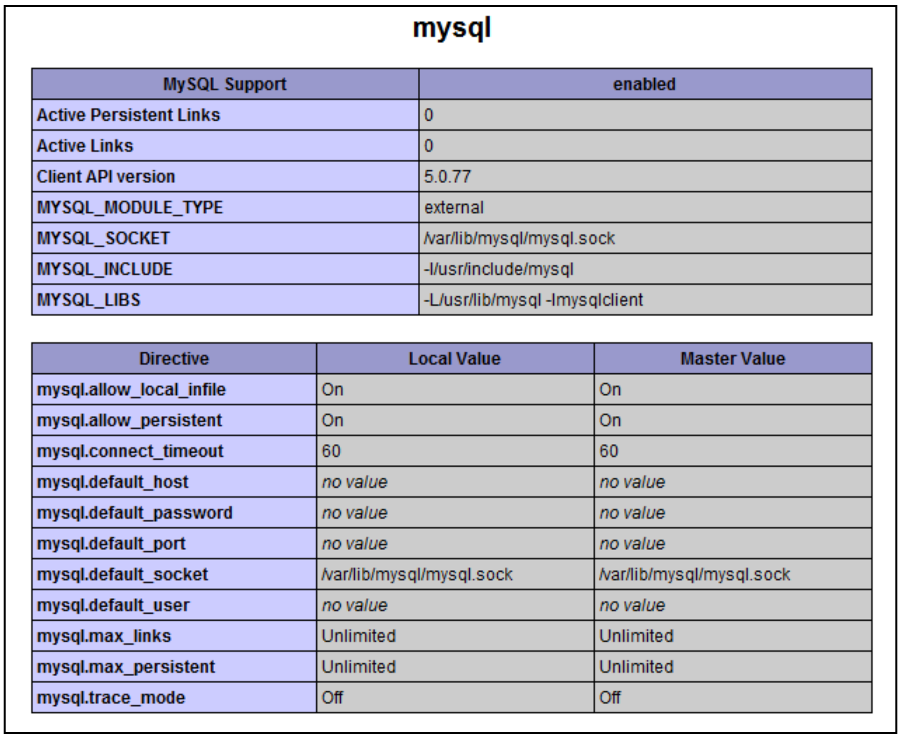
These extra options tell the compiler where the tools are to build the correct MySQL libraries (similar to the flag to tell it where the Apache extension tool is to build `mod_php`).

Remember you will need to `make` and `sudo make install` to complete the build.

Finish off with a restart of Apache to load the newly updated PHP module into the server configuration.

Check your installation

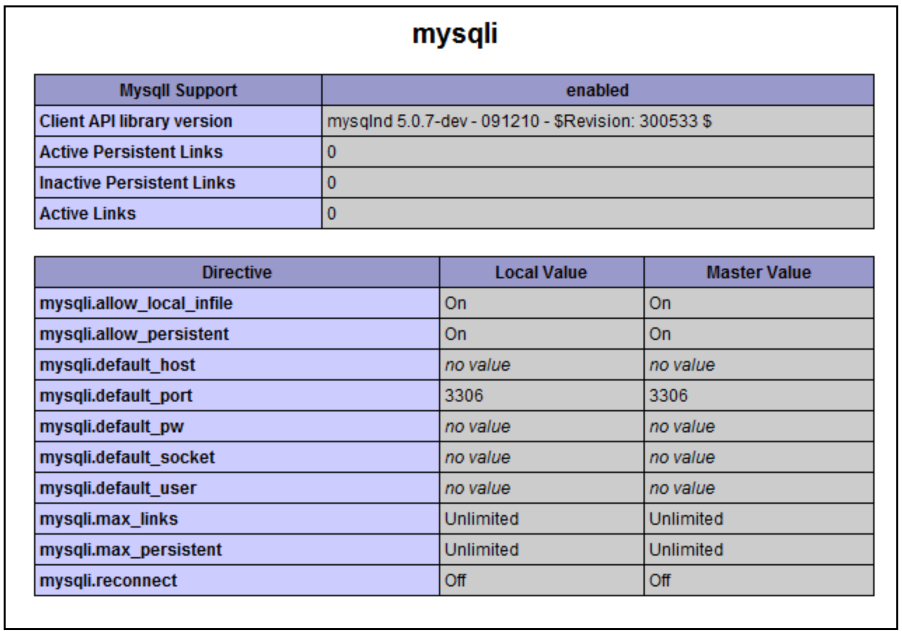
If you visit your `phpinfo()` script you should see that the connection and driver information for MySQL and its connectors are now complete – a bit like this (again the version numbers you have may be slightly different)….



mysql

MySQL Support	enabled
Active Persistent Links	0
Active Links	0
Client API version	5.0.77
MYSQL_MODULE_TYPE	external
MYSQL_SOCKET	/var/lib/mysql/mysql.sock
MYSQL_INCLUDE	-I/usr/include/mysql
MYSQL_LIBS	-L/usr/lib/mysql -lmysqlclient

Directive	Local Value	Master Value
mysql.allow_local_infile	On	On
mysql.allow_persistent	On	On
mysql.connect_timeout	60	60
mysql.default_host	*no value*	*no value*
mysql.default_password	*no value*	*no value*
mysql.default_port	*no value*	*no value*
mysql.default_socket	/var/lib/mysql/mysql.sock	/var/lib/mysql/mysql.sock
mysql.default_user	*no value*	*no value*
mysql.max_links	Unlimited	Unlimited
mysql.max_persistent	Unlimited	Unlimited
mysql.trace_mode	Off	Off



mysqli

Mysqli Support	enabled
Client API library version	mysqlnd 5.0.7-dev - 091210 - $Revision: 300533 $
Active Persistent Links	0
Inactive Persistent Links	0
Active Links	0

Directive	Local Value	Master Value
mysqli.allow_local_infile	On	On
mysqli.allow_persistent	On	On
mysqli.default_host	*no value*	*no value*
mysqli.default_port	3306	3306
mysqli.default_pw	*no value*	*no value*
mysqli.default_socket	*no value*	*no value*
mysqli.default_user	*no value*	*no value*
mysqli.max_links	Unlimited	Unlimited
mysqli.max_persistent	Unlimited	Unlimited
mysqli.reconnect	Off	Off

Figure 2. MySQL configuration viewed via `phpinfo()`

Establish a connection using PHP

You can now use PHP to make MySQL connections. For this part of the assignment the goal is to create a simple PHP script that will run on **your** web server, make a connection to a MySQL server running on a **different** VM, query a database on that server and return a web page containing the results.

Therefore, your job is to create a simple PHP script that will:

- Access the MySQL server running on `vm-eliot-000.ncl.ac.uk`
- Connect using the username `vm-eliot-NNN*` and password `csc3504`
- Use the database called `csc3504` and query the table `vm_eliot_NNN*`
- Extract the unique 6 character string (in the field `VM_STRING`) stored against your VM number (in the field `VM_NAME`) and return it as the following (exact) HTML snippet*

  `<p>vm-eliot-NNN uses reference: XXXXXX</p>`
- Deliver a web page** containing the snippet at:

  `http://vm-eliot-NNN.ncl.ac.uk/php/dbtest/`

*Replace NNN with *your* VM number and XXXXXX with the unique 6 character string you will get from the database.

Notice the database table and field names use underscores (_) not dashes (-)

**The rest of the page can be constructed however you like as long as it is proper HTML and it works!

© JISC Netskills 2011

Submitting Your Work

Although you may have been submitting files as you go along, the *only* URLs and files that will be marked will be those included in the *last submission you make before the deadline*.

This means that you will need to submit the unique files and URLs for all the components together.

Below you will find a checklist for the whole assessment and the filenames and URLs you will need to include.

Please pay close attention as *some of your files may need to be renamed in order to submit successfully*.

Also not you only need to submit one copy of each file for the final submission. For example the final version of your `httpd.conf` should contain all the changes required for all of the sub-components.

URLs to be auto-marked

These URLs should be submitted to the **URL** submission area for this assignment:

```
http://vm-eliot-NNN.ncl.ac.uk/php/

http://v-eliot-NN.ncl.ac.uk/php/dbtest/
```

A script will access these URLs, check they are delivering the required content, functionality or error message as per the assignment details and store a copy of the output.

You should ensure that they all deliver the correct content when accessed ...and that your server is on and running!

Files to submit

Submit the files required for marking in *a single zip archive* to the **FILES** submission area for this assignment.

Name your zip file:

```
a123456789-vm-eliot-NNN.zip
```

Where **a123456789** is your student number and **NNN** is your VM number

File Checklist

Your file...	Renamed as...	✗/✓
`php.ini`	`vm-eliot-NNN-php.ini.txt`	
`httpd.conf`	`vm-eliot-NNN-httpd.conf.txt`	
`index.php` from `/dbtest/`	`vm-eliot-NNN-db-index.php`	