

Comprehensive step-by-step instructions for installing Moodle 1.5.x, 1.6.x, or 1.7.x from scratch on Macintosh under Mac OS X 10.3.x or 10.4.x

by Robert Brenstein, rjb@robelko.com

created on 3 September 2005
last updated on 25 November 2006

These instructions assume basic familiarity with Macintosh operation and capability to use the Unix Terminal but do not require advanced knowledge of Unix behind the graphical user interface or having a proper webmaster or networking expertise.

These instructions are specifically for server running locally under your full control and using OS X client (that is the "normal" or desktop edition of OS X) not the OS X Server.

These instructions are for setting up a secure production server. If you want only to experiment with Moodle, you may prefer to get the Moodle4OSX package, which offers simple one-step installation.

Hardware requirements

Verify that your Macintosh hardware meets the requirements. Any G4, G5, or Pro Macintosh will serve Moodle well. Even the original G4 can support a Moodle site with a few thousand users under a moderate usage. However, the faster the computer, the better, and for active sites a G5 or Intel Mac is clearly preferred, particularly for newer versions of Moodle.

Moodle will truly benefit from having a lot of memory, so a computer with at least 1 GB of RAM is required and 2 GB recommended. For computer pool usage, that is many truly simultaneous users, 4 GB is likely required. When your Moodle site has lots of files to download, particularly large files, using a SCSI card with a fast hard drive might improve performance.

Having a non-shared fast ethernet connection to network is also important when performance is an issue. You may thus want to check with your network administrator that the hub port used by your Moodle server is not shared with other computers.

Mac OS X install

Reinstalling system software is strongly recommended if a used computer is becoming your Moodle server. Not needed if a factory new computer is used.

To reinstall, insert Mac OS X installation CD 1 and double-click the "Install Mac OS X" or "Install Mac OS X and bundled software" file. Follow the standard or custom installation as desired. It is recommended to use the option to erase the volume before installation.

Primary user

At the end of system install or when starting a brand new computer, you have to create the primary user. This will be the administrator account of your server, so don't give it a personal name but something more generic like system administrator or moodle administrator or server administrator.

Server address and host name

In order to give public access to your server, it has to be connected to a publicly accessible network and have an officially assigned IP address. Normally, it should also have a proper host name assigned to allow a more mnemonic URL (internet address). Host name has to be defined by the DNS administrator of your network and the matching data entered into the TCP/IP panel of the Network control panel.

System configuration

Open "System Preferences" from the Apple menu. Configure "Energy Saver" control panel to never go to sleep and never put hard drives to sleep. Turn off screen saver in "Desktop and Screen Saver" control panel. Adjust other settings as desired. You may want to disable automated checking for system updates if you learn about them on your personal computer or elsewhere.

You may also want to place Terminal, Console, Activity Monitor, and Network Utility on the dock for easy access. They are all hiding in the "Utilities" folder inside the "Applications" folder.

Xcode

You should also install Xcode so you can use CVS. While Moodle package installer appears somewhat simpler, CVS-based installation makes minor upgrades simpler, particularly when you do minor code changes to your Moodle installation, and thus makes the maintenance easier. You can skip installation of Xcode examples and documentation to save disk space. The installation creates "Developer" folder on the top level of the startup disk.

To install Xcode, insert the CD with Xcode Tools or System Installation CD 1. Look for the file "XcodeTools.mpgk" or similar. It is also possible to install the whole Xcode from Apple's web site. It is a hefty download, though.

System update

If you installed system from CD or even if you got a brand new computer, you should check now whether any system updates are available.

Restart (really optional but we do it for the good measure), launch "Disk Utility" and do "Repair Disk Permissions" on your system volume.

Repairing permissions is a somewhat controversial step, so do it at your judgement.

Open "Software Update..." from the Apple menu and do installations as needed and desired.

You should repeat this step, possibly even a few times, since new updates may show up after some updates are installed.

Restart, launch "Disk Utility" and do "Repair Disk Permissions" on your system volume again.

Utility applications

Install any other applications you may need to use. The suggestions here are not to be considered recommendations but examples of programs that are tested and functional. You are free to use whatever you desire and just substitute the program names in the instructions accordingly.

Having a programmer-oriented text editor is a good idea, for example TextWrangler, a free sibling of BBEdit. When installing TextWrangler or BBEdit make sure to install them as the Unix service as well.

You may also want to have a database editor, for example YourSQL or CocoaMySQL (both freeware) for editing MySQL directly on the server. YourSQL is fairly limited but it does all the basics and is quite simple to use. Either program is more convenient than using myPhpAdmin within Moodle or the editor provided by the MySQL group when working directly on the server.

You might consider to install all these into "Applications2" folder instead of "Applications" to keep them separately from Apple-provided software, but this is not really necessary. You may also want to add them to the dock to have easy access to them.

Backups

Backing strategies for a server are not so trivial since you don't want to backup live database files, for example, or impact users accessing your server remotely. A production server of any importance should have a RAID-based clone of its hard disk plus a proper incremental backup of all files to another (best external) medium every week or more often. You will typically produce a dump of your database and produce a zip or tar archive of your Moodle data files, and back these dump and archive files up using some backup software, like Retrospect (commercial), SuperDuper (shareware) or Carbon Copy Cloner (freeware).

Apache configuration

If you installed TextWrangler, launch "Terminal" and enter 'edit /private/etc/httpd/httpd.conf' or, better, launch TextWrangler and choose "Open File by Name..." from the File menu, then enter or paste /private/etc/httpd/httpd.conf into the filename field and click "open". This will open the Apache config file. You can, of course, use vi, vim or whatever else you like.

Locate the line with #ServerAdmin. Remove the hash to activate this command. Modify the webmaster's email address appropriately.

Locate the line with #ServerName. Remove the hash and replace the example name with your host name or IP address (if not using host name to access this server). Whatever you enter here has to match what is in the TCP control panel (if using IP address) or DNS settings (if using host name). It will also have to match Moodle's config.php. While it is tempting to simply use localhost or 127.0.0.1, it is better to enter the final address, so you don't have to change all the settings again when going public.

You should change 'Options Indexes FollowSymLinks MultiViews' in the <Directory "/Library/WebServer/Documents"> block to 'Options FollowSymLinks MultiViews' -- that is remove 'indexes' from the active options in order to suppress displaying directories.

If you prefer a single Apache log with extended information, put hash in front of 'CustomLog "/private/var/log/httpd/access_log" common' and remove it in front of 'CustomLog "/private/var/log/httpd/access_log" combined'

To accomodate changes in file structure (which we do shortly), change 'Alias /manual/ "/Library/WebServer/Documents/manual/"' to 'Alias /manual/ "/Library/WebServer/Documents/apache/manual/"'. Change accordingly <Directory "/Library/WebServer/Documents/manual"> to <Directory "/Library/WebServer/Documents/apache/manual"> in the same block.

To tune performance, you may also want to change Timeout from 300 to, for example, 120 and MaxKeepAliveRequests from 100 to for example, 85. The former reduces the delay for killing stale connections. The latter reduces the number of open connections to be less than the number of threads allowed by MySQL (normally 100).

Check Moodle forums for discussions on tuning Apache performance. For example, if you server has lots of traffic, you may want to increase StartServers. If your server is overloaded, you may lower MaxClients.

You may also want to disable Rendezvous (in Panther) or Bonjour (in Tiger) advertising. For example, if you don't want Rendezvous to advertise private sites, comment out (put hash in front of) 'RegisterUserSite customized-users'.

Save changes. You may be asked for admin password when doing so.

Note: if you make any Apache config changes after you started Apache, you need to shut it down and restart in order to make those changes in effect. You do it by turning Personal Web Sharing in the "Sharing" control panel under the "Services" tab off, and once it is reported off, then on again.

Apache activation

Open "System Preferences" and switch to the "Sharing" panel, then further to the "Services" tab. Activate Personal Web Sharing. You may also activate now FTP Access if desired or required. Switch to the "Firewall" tab and activate Firewall if it is not active. You may also want to set the computer name on top of that panel.

Open web browser on this or another computer and enter the host name or IP address of your Moodle server. You should see the default page of Apache with "Seeing this instead of web site you expected?" line in big text. If you don't, you need to figure out what gives. Seeing

this page verifies that web service works correctly.

Web server folders

Open your system volume. Open the "Library" folder and then the "WebServer" folder. Rename "Documents" to "apache" and create a new empty folder called "Documents". That folder will be the root of main web site.

Drop the "apache" folder into "Documents", so you can access Apache docs if needed. The alias we modified while editing Apache config allows you to simply enter `http://domain.or.ipaddress/manual/` to get to those docs.

PHP installation

Fetch Marc Liyanage's PHP installer from <http://www.entropy.ch/software/macosex/php/>. If you want to learn more about PHP, visit <http://www.php.net/>. You can fetch either PHP4 or PHP5. Whichever you get, make sure it is for Apache 1.3 (Apache 2.x is used by OS X Server). You can use either branch (that is either PHP4 or PHP5), just check the system requirements for the Moodle version you install to know which version in which branch you need to have (cf. http://docs.moodle.org/en/Installing_Moodle) and match it to the OS X version and the hardware (PPC or Intel) you use. Read the information provided by Marc Liyanage or check his forums if you have doubts. Check also Moodle forums for indications of problems with specific PHP releases. Links to just a tad older releases can be often found in the PHP forum on Entropy.

Read the installation instructions on Marc's web site. If you got the dmg file, double-click it if it didn't automount. Otherwise, just decompress the tar file with the installer. If you want to use BOMArchiveHelper instead of StuffIt, click on the downloaded tar file and select "Open With" from the File menu in Finder. It should list BOMArchiveHelper as the default application. Just select it.

Double-click the installer pkg or mpkg file and do the standard installation. You may be asked for admin password during installation. The program files will be installed into `/usr/local/php5/` or `/php4/` directory depending on the branch you are installing.

To test whether PHP works, use TextWrangler (or whatever editor you decided for) to create a text file named "phpinfo.php" and place it in the "Library/WebServer/Documents" folder. This file should contain a single line: `<?php phpinfo() ?>` with no return character. Open that file in your browser by entering `http://domain.or.ipaddress/phpinfo.php` as URL. You should see a table with information about the PHP module and status of its components. If you don't, you need to figure out what went wrong.

PHP setup in Apache

Should you want to edit PHP-related config file for Apache, open file `/usr/local/php5/httpd.conf.php` or `/usr/local/php5/entropy-php.conf`, depending on the PHP version, as you opened Apache config. However, there is normally nothing to change there.

Note: early versions of Marc's PHP installer inserted PHP activation code directly into the `httpd.conf` file. Newer versions install only an alias (file link) in the `/private/etc/httpd/users/` directory. All conf files in that directory are automatically included by newer versions of Apache. You may want to check `admin.conf` (a file which matches your short user name) there to see whether it does not overwrite something.

PHP configuration

Enter 'edit `/usr/local/php5/lib/php.ini`' or launch TextWrangler and choose "Open File by Name..." from the File menu, then paste `/usr/local/php5/lib/php.ini` into the filename field and click "open" (or use whatever other editing method you chose) to open the PHP configuration file.

Change `memory_limit` to 16M (default is 8 or 16, depending on PHP version). You can set it to 32M if your server has 2 GB or more of RAM.

Change `magic_quotes_gpc` from Off to On.

Change `post_max_size` and `upload_max_filesize` to 16M or 20M or even more (default is 8M and 2M, respectively). The set the limit to the size of uploaded files. You may set them to as high as 80M if you want to allow users uploading really large files (you can still set this limit lower in Moodle).

Check that the following settings are correct (they are normally default on Macintosh):

```
file_uploads = On
session.auto_start = 0
session.bug_compat_42 = 0
session.bug_compat_warn = 0
```

Save changes. Turn Apache off and on (see note in the Apache configuration) to activate the changes. Switch to browser and refresh the page with `phpinfo.php` output. Search, for example, for `magic_quotes_gpc` and see that it is now on.

MySQL installation

Go to <http://www.mysql.com/downloads/> in your browser and look for "Mac OS X (package format) downloads" section of the MySQL Community Server. Fetch the installer package for a standard version that matches your Mac OS X version and hardware platform (PowerPC or x86 = Intel) and meets the requirements of the Moodle version you will run (http://docs.moodle.org/en/Installing_Moodle). If you want to learn more about MySQL, visit <http://dev.mysql.com/>. Some troubleshooting tips are also at Marc Liyanage's site, <http://www.entropy.ch/software/macosex/mysql/>.

Double-click the dmg file if it did not automount. Double-click the installer pkg or mpkg file and do standard installation. The program files will be installed into `/usr/local/mysql-version/` directory ('version' means the actual numeric version). The installer also makes a symbolic link `/usr/local/mysql`, that points to the just installed MySQL version. If `/usr/local/mysql` exists, it is renamed to `/usr/local/mysql.bak.first`.

Install also MySQLStartupItem, a component that launches MySQL automatically when system boots, by double-clicking its installer package and following the instructions. The StartupItem needs to be installed only once. This means that there is no need to install it each time you upgrade MySQL later. The Startup Item for MySQL is installed as /Library/StartupItems/MySQLCOM.

Install MySQL prefs pane as well. It allows you to start and stop MySQL from "System Preferences". To do this, open the Library folder in your user folder (not the one on root level of your hard disk). If there is a folder called PreferencePanels, drop the MySQL.Prefpane file there. If the folder does not exist, create it first.

Open Terminal and type in the following commands to change the owner of data files from root to mysql. This is required because MySQL runs its process under user called mysql.

```
cd /usr/local/mysql
sudo chown -R mysql data/
```

Open System Preferences and select MySQL pane. Click the "Start MySQL Server" button. You will be asked for your admin password and after a moment, the server should start running. Make sure that the checkbox to automatically start MySQL at startup is checked before closing the System Preferences.

In order to start the server if you haven't installed the Preference pane, open the Terminal and execute:

```
sudo ./bin/mysqld_safe &
```

Note: most of commands we execute in Terminal will be preceded by sudo so they are executed as the root user. The sudo command will require your entering the admin password to ensure that you really have privileges to run it. If you repeat sudo a few times in a short time span, only the first sudo should ask for password.

Configuring (securing) MySQL

We will let Moodle access the database as the root user (note that MySQL's root user is not the same as OS X's root user). We will disable remote access to Moodle databases (meaning that only programs running on the server itself, that is localhost, can gain access), so there is no need to create an extra user assuming that our server is in a secure location.

By default MySQL's root user has no password, so we need to set it:

```
sudo bin/mysqladmin -u root password new_password
```

Replace "*new_password*" above with the actual password you wish to have, of course.

In newer versions of MySQL, a special security script is provided, which you should run. Execute in Terminal:

```
ls -al bin/
```

to see the list of files in the bin folder to check whether mysql_secure_installation is there. If you have it, you can skip setting the password directly and set it through this script instead.

```
sudo bin/mysql_secure_installation
```

The script sets the root password, removes anonymous users, disallows remote root login, and removes test database and access to it. If you have already set the password directly, the script gives you an option to skip that step.

If you don't have the mysql_secure_installation script, you should secure the same items manually using the mysql utility program (cf. section on creating Moodle database).

```
DELETE FROM mysql.user WHERE User="";
DELETE FROM mysql.user WHERE User='root' AND Host!='localhost';
DROP DATABASE test;
DELETE FROM mysql.db WHERE Db='test' OR Db='test\_%';
FLUSH PRIVILEGES;
```

Note: if you get an error "line 33: mysql: command not found ... Failed!" while executing the "mysql_secure_installation" script, you need to edit the script file and change mysql to bin/mysql in line 33. You can edit the script file using TextWrangler by opening /usr/local/mysql/bin/mysql_secure_installation.

Moodle maintenance and folder and database naming convention

This section is not a true installation step; however, you need to consider future maintenance to make some decisions affecting further installation.

The installation instructions on moodle.org suggest to keep the Moodle program files in a folder called "moodle" and its data files in a folder called "moodledata". They also suggest to name the database "moodle". This is all fine except that if you plan to stay with Moodle for longer, you need to consider future updates and upgrades. Going from Moodle 1.6.2 to 1.6.3, for example, is an update. Going from Moodle 1.6.3 to 1.7.1, for example, is an upgrade. These terms are, however, often used interchangeably.

Minor updates can be made in place since they usually involve only minor changes to the Moodle program files. However, major upgrades normally involve changes in the database structure besides heavy changes in the program. It is a good idea (and usually required) to do major upgrade while keeping the working version of Moodle. This means we need to accommodate two Moodle versions running in parallel. One way to accomplish this is to amend the major version number to version-specific files.

For example, if installing Moodle 1.5.x, we create:

```
/library/webserver/documents/moodle/ -- folder for the Moodle 1.5 program
```

/library/webserver/moodledata15/ -- folder for Moodle 1.5 data

and the corresponding database name in MySQL is: moodle15

Then, when Moodle 1.6.x is installed while Moodle 1.5 is still running as the production site, we create

library/webserver/documents/moodle16/ -- folder for the Moodle 1.6 program

/library/webserver/moodledata16/ -- folder for Moodle 1.6 data

and the corresponding database name in MySQL is: moodle16

When we are ready to actually switch to the new version, we just need to

```
rename folder /library/webserver/documents/moodle/ to /library/webserver/documents/moodle15/
```

```
rename folder /library/webserver/documents/moodle16/ to /library/webserver/documents/moodle/
```

If there is no need to run Moodle 1.6 in parallel to Moodle 1.5 in preparation for the switchover, you can do both these steps at once. However, any site that does any customization of Moodle will likely need time to transfer these changes to the new version and/or experiment with the new features the new version brings.

The above assumes that we keep the production version of the program with the same name in order to avoid changes in the internet address. This is not truly required. As an alternative, we can have Moodle 1.5 in a folder called moodle15 and use Apache alias feature to redirect users from moodle to moodle15. For example,

```
Alias /moodle/ "/Library/WebServer/Documents/moodle15x/"
Alias /moodle "/Library/WebServer/Documents/moodle15x/"
```

```
<Directory "/Library/WebServer/Documents/moodle15x/">
    Options FollowSymlinks MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

In this case, switching Moodle production versions requires only editing the aliases in the httpd.conf file and restarting Apache once the new version is fully functional. Of course, corresponding changes to the config.php file are still needed.

The details of handling updates and upgrades in either scenario are provided in a separate document.

So, you need to decide whatever naming scheme suits your needs. In the following instructions, we will simply refer to the "moodle" folder and the "moodledata" folder as well as the "moodle" database as generic terms, so you will need to substitute these with whatever you actually decided to use. To make clearer, they will be shown in italics where used as generic names.

Note: if your web site is fully operated by Moodle, that is Moodle is your homepage, it may be tempting to install Moodle into the root folder of your web site rather than into a subfolder. As you can guess by now, it is not really a good idea. It is better to install Moodle in a subfolder and use Apache redirection to make that folder behave as the root of your web site.

Deciding which Moodle version to install

Typically, you would get the "Latest Stable Branch" which is the newest build of the current official release. In Moodle jargon, a Stable Release is the set of files as they were on the day a given version was officially released. There may be a major release, like 1.5, or its update, like 1.5.1. The Latest Stable Branch includes whatever patches and changes were made since then, which is indicated by a plus sign after the release number, like 1.5.1+. At some point in time, this becomes the next stable release, 1.5.2, and the work continues with 1.5.2+ leading to 1.5.3. And so on.

Tip: if you are ever not sure which version of Moodle you are using, open the file version.php in the Moodle folder if you have direct access to Moodle files. When looking at a running Moodle in a browser, you get the same info when pointing mouse to the small Moodle logo at the bottom of the administration index page.

Note: The internal Moodle version number is given as *yyyymmddbb* which is *yyyy*=year, *mm*=month, *dd*=day of the branch release date, and *bb*=build, a sequentially increasing build number since that date. The *yyyymmdd* stays constant for a given branch.

Deciding on the installation mode of Moodle

You can install Moodle software by downloading it as a ZIP archive (installation pack) or fetching it from CVS (code versioning system) repository at SourceForge. The former is quite straightforward, whereas the latter requires using Terminal, although it is not really more complicated.

CVS-based installation makes minor updates simpler, particularly when you do minor code changes to your Moodle installation. When using installation pack, in order to install an update, you typically fetch a newer pack, rename the old Moodle folder, and then place the newer "moodle" folder in the "Documents" folder. You need then to transfer the config file. If you modified any PHP code, you need to check whether your changes are not overlapping changes in the distribution files and transfer the changes to new files (if files were not modified, old files can be copied over). In case of CVS-based installation, the cvs command does the merging of new files with old ones. Of course, in case of conflicts (overlaps of changes), you need to decide what to do. With lots of changes, CVS does make this process much simpler.

Well, it is your call which mode you want to use.

Note: while it is possible to switch to using installation pack at any time, it is not possible to use CVS for updates if the installation was done using installation pack. When installing, CVS places some information it needs for updating in each folder it creates. This information is not there when installation pack is used.

Arranging mail service for Moodle

In order to send emails, Moodle need to have access to a mail server, actually mail-sending or SMTP server as it is technically referred to.

OS X comes with a SMTP server program, which can be activated to provide this service locally. However, most institutions installing Moodle already have some mail service, so it is usually simpler to use that service than operating your own mail service, besides leaving extra CPU cycles for Moodle. Running a mail server is not difficult but still requires some work since mail server must be setup properly if you don't want to create a heaven for spammers.

If you really want or have to run your own mail service, you need to get Postfix running. You can activate it and control using the WitiSMTP (freeware) or Postfix Enabler (shareware).

The step-by-step instructions for manually activating postfix are on, for example

<http://www.macsoxhints.com/article.php?story=20031025022626398>
<http://www.david-reitter.com/software/osxpostfix.html>.

If you have an older OS X with Sendmail, you may want to install Postfix instead of using the provided Sendmail. For instructions see, for example

<http://www.stepwise.com/Articles/Workbench/eart.index.html>.

Some institutions also firewall inside mail servers (that is block those mail servers from sending emails directly out), so you may need to get Moodle server recognized as a trusted mail sender. This is normally required if a mail gateway is run in your organization. Since Moodle will be sending lots of emails which may be mistaken for spamming, you want to make sure it is properly recognized. Talk to the Postmaster about this if there is one.

Regardless where your mail server is, it is a good idea to set up a generic email account for Moodle administration, like `moodle@your.domain` or `moodle-admin@your.domain`, so the basic contact address is not tied to a specific person. Mail coming to this address should be redirected by the mail server to whomever administers Moodle. The mail exploder can even send those emails to more than one person.

If your mail server requires verification to send mail out, this account will be a must to have and will be your ticket to configuring the Moodle site below.

If you can setup an autoresponder for an email address, you may also want to create an email account `noreply@your.domain` or `moodle-noreply@your.domain` and make it autoreply with a message similar to the following:

"noreply@your.domain" is not a valid email address. You have sent this email most likely attempting to respond to a forum post in Moodle. To respond to a forum post, you need to visit that forum and use the "reply" link below the post.

It is just invariable that some users will try to reply to forum posts without noticing that the poster has their email address hidden, and this automated response will notify them that their email couldn't reach the recipient.

Installing Moodle software using the installation pack

Fetch the desired Moodle installation pack in the zip format from <http://download.moodle.org/> (the "Download Moodle" link at moodle.org). Safari should automatically unzip it when downloading and place the downloaded "moodle" folder on the desktop.

Open the hard disk, then the "Library" folder, then the "WebServer" folder, and drop the "moodle" folder from desktop into the "Documents" folder there.

If you decided to call the Moodle folder something else than "moodle," rename it accordingly.

Installing Moodle software using CVS

Launch Terminal and execute:

```
cd /library/webserver/documents/  
cvs -d:pserver:anonymous@moodle.cvs.sourceforge.net:/cvsroot/moodle login
```

When you are asked for password, just press the return key.

If you get an error, it is likely that SourceForge has some sort of hiccup. It does happen occasionally that SourceForge is not accessible or malfunctions. If that happens, wait a few minutes and try again. If it still fails, then try again even later or visit <http://sourceforge.net> to see whether there are any announcement of service interruption.

Make sure that there is no folder named "moodle" in the "Documents" folder or you get an error.

```
cvs -z3 -d:pserver:anonymous@moodle.cvs.sourceforge.net:/cvsroot/moodle co -r MOODLE_RELEASE_TAG moodle
```

where `MOODLE_RELEASE_TAG` is the tag of the version you want to fetch. This tag is given in the second column on <http://download.moodle.org/>. For example, to install the latest stable branch of version 1.6, you would execute:

```
cvs -z3 -d:pserver:anonymous@moodle.cvs.sourceforge.net:/cvsroot/moodle co -r MOODLE_16_STABLE moodle
```

Now you will see CVS downloading all the files from CVS repository. This can take a while to finish, depending on how busy SourceForge is and how fast the network connection works.

If you decided to call the Moodle folder something else than "moodle," rename it accordingly after downloading is finished.

Installing phpMyAdmin

Moodle is nicely modular and allows to add activity modules, filters, extensions, question types, and other plugins. You will probably add these later. Now, let just fetch phpMyAdmin. It will enable you to do some tweaks directly in the database when working remotely. Locally at the server, working in YourSQL or alike will usually be preferable.

Go to moodle.org in your web browser, then to "Modules and plugins from the Main Menu. Search the database for "myadmin" and switch to the "view single" tab. Download the ZIP archive using the "download" link. Unzip phpMyAdmin and move the resulting folder into the "admin" folder inside the "moodle" folder. If the phpMyAdmin's folder is not called "mysql" rename it so.

Creating Moodle files folder

Go to the "WebServer" folder and create the moodledata folder, naming it accordingly. This folder will contain all course and user as well as some administrative files for Moodle. We place it parallel to the web root folder (that is in the "Documents" folder), so it is not possible to get to it through web browsers.

Note: if you have a second hard drive, particularly a fast SCSI drive, place this folder on that drive for improved performance.

Setting access privileges

Apache under OS X runs processes under user "www" so we need to adjust some ownership information. We also need to make sure that Moodle has full access to all files. To ensure that, execute in Terminal:

```
sudo chown -R admin:www /Library/WebServer/Documents/moodle/  
sudo chown -R admin:www /Library/WebServer/moodledata/  
sudo chmod -R 777 /Library/WebServer/Documents/moodle/  
sudo chmod -R 777 /Library/WebServer/moodledata/
```

Keeping web robots at bay

Moodle has a built-in robot protection, but we can also make things a tad simpler for it. To keep all well-behaving robots away from indexing the moodle site, create a file named "robots.txt" in the "Documents" folder with the following:

```
User-agent: *  
Disallow: /
```

This will block the entire site for all robots. If you want to exclude only the Moodle folder, then set the robots file to:

```
User-agent: *  
Disallow: /moodle/
```

If you want to allow, for example, Google but exclude others, enter

```
User-agent: Googlebot  
Disallow:
```

```
User-agent: *  
Disallow: /
```

Creating Moodle database

Open the Terminal program and execute:

```
cd /usr/local/mysql  
sudo bin/mysql -u root -p
```

You probably need to enter two passwords, first for sudo and then for mysql. A 'mysql>' as prompt will indicate that you are in the MySQL utility program.

If installing Moodle 1.5.x, enter:

```
create database moodle_database_name;
```

If installing Moodle 1.6.x or 1.7.x, enter:

```
create database moodle_database_name default character set utf8 collate utf8_unicode_ci;
```

Now exit the MySQL utility:

```
quit;
```

Once back in Terminal, you need to make sure the changes are active:

```
sudo bin/mysqladmin -p reload
```

Note: Standard Moodle installation instructions include executing the grant command after creating the database:

```
mysql> grant select,insert,update,delete,create,create temporary tables,drop,index,alter on moodle_database_name.* to moodleuser@localhost identified by 'user_password';
mysql> flush privileges;
```

We don't need this because root by default has

```
GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' IDENTIFIED BY PASSWORD '*xxx' WITH GRANT OPTION
```

Installing Moodle service

If you have your browser running, quit it. This ensures that any old Moodle sessions are closed and don't interfere with installation.

Launch your web browser (it can be on the Moodle server or another computer) and connect to your Moodle by going to

`http://hostname.or.ipaddress/moodle/`

If your URL is correct, the Moodle installation script should greet you. What follows varies somewhat from version to version, but the general pattern is basically the same.

At the first screen, the Moodle installation agent is asking you to choose the language to use during the installation.

Installer checks then the computer. You can continue if there are no problems, otherwise, you need to address them.

Next you need to enter the web address (your Moodle's full URL), Moodle directory (the filepath to the Moodle software folder), and data directory (the filepath to the Moodle files folder).

Regarding web address: As URL, you can enter either IP address or, better, fully qualified domain name; ie. `http://my.domain.edu/moodle`. No slash at the end. You can also enter `http://localhost/moodle` but you will need to change that later to allow access from other computers.

Note: if you are using Apache aliasing/redirection, you should enter here what users see.

Regarding Moodle directory: The standard filepath to Moodle folder is `/Library/WebServer/Documents/moodle`. You need to edit it if you called your "moodle" folder something else.

Regarding data directory: The data folder path defaults to `/Library/WebServer/Documents/moodledata` but we want to remove the "Documents/" part since we placed the folder above web root. If you made a custom name for this folder, you need to further adjust the path accordingly. When you proceed, Moodle installer verifies that folders exist and complain if not found. Note that it will also claim that the folder is not found if it has no write access to it.

The next installation step deals with database access. MySQL selection should stay since this is what we installed. Host server can stay as "localhost" since we defined only localhost-based access for our admin. Database is the name of the database, that is 'moodle' if you did not set it otherwise. User is root and password is what you set earlier. Table prefix can stay at default "mdl_". We need to change it only when operating multiple Moodle services within a single database, which is generally not a good idea.

The installer will now attempt to connect to MySQL and report problems if it fails. Otherwise, it will continue and create the config.php file in the moodle folder. Should it fail, it will offer you to download the file or copy its content and install manually. However, if the problem is with access privileges, you can correct them and click the refresh (reload) button in your browser.

Moodle 1.6 and 1.7 will report fetching the language you chose for installation before creating the config.php file.

Next comes the copyright notice. Not much to it.

Now you will get a page reporting creation of database tables. Lots of SQL gibberish. You just have to make sure that there is "success" reported for each table. Moodle 1.7 tells you that at the top and bottom.

Next page will tell you the exact version of Moodle to be installed. The number in parentheses is in `yyymmddb` format (year-month-day-build). Release notes for this version should follow.

Now we come to the global settings. Each option has an explanation, so just read through them and set accordingly. Don't fret too much if you are not sure. You can change them all in the admin interface once the service is operational. The mail service info should match what you arranged in the section dealing with mail service above.

Note: configuring global settings at this point is not part of 1.7.x installation. You configure them through the Site Administration block after basic installation.

When you click "Save Changes", you get a confirmation page and the installation will automatically (no need to click) continue with creation of database tables for individual activity modules. Again, you just need to verify that each one was created successfully.

Next are tables created for the backup functionality. Database is then updated for blocks stuff. Next various components are set up. Again, verify that all was successful. Next come authorization and paypal.

Now come main page settings. Fill the fields as desired. Again, you can change them all later using the admin interface.

Note: Moodle 1.7.x installation presents the main page settings after defining the primary admin user.

Now you set the primary admin user. You could set it as a real person, particularly if there is a single administrator, but it is usually better to make it a generic user, Moodle Admin, for example, and let it use a generic email address (cf. section on mail service). Make sure to enter some password since there is no password set. Note that admin user will be using the internal authentication, so it will work with the given password regardless which authentication mode you choose for your site. Yes, you can change admin settings later as well.

To save admin settings click "Update profile". Once the changes are saved, the installer will continue to your Moodle homepage.

The basic installation is done.

Setting site authentication

By default, Moodle is set to mail-based authentication. This means that new users can add themselves as your Moodle users. If you want to use another authentication mode to control access, you should set it up right away, before anyone gets a chance to sign on. If you need more time to arrange it, set the authentication to manual accounts only to prevent undesired users or put your Moodle into maintenance mode.

To set authentication, click the Users item in the Administration (Site Administration in 1.7) block and then follow to Authentication. Use the popup menu on top to select the desired authentication mode. Provide the necessary configuration data and save the changes to make it active.

The authentication page allows you to set also whether guest login button is showing, provide link to an alternate login page, and set editing locks on fields in user profile. Additional options may be shown for specific authentication modes.

Note that after saving changes, you will return to the same page. This happens because changing the authentication method may bring new fields to fill or options to set.

Note: changing the authentication method in Moodle 1.7.x reloads the page right away, so the additional fields or options are immediately present, but saving the configuration returns you to the same page nevertheless.

Further customizing Moodle through web interface

To customize your site, click the "Configuration" in the Admin block on the home page to be taken to the configuration panel.

Note that "Administration" panel is not the same as "Configuration." Administration is a collection of all administration areas, including Configuration, Users, etc. To see the Admin page now, click the "Administration" in the breadcrumb (in case you don't know what breadcrumbs refers to: it is the line of links on top that tells you where you are in structure and which allows you to hop quickly higher in hierarchy). Similarly, if you click "Users" in the Admin block on the home page, you end up on the "Users" panel not the main "Administration" panel. If you want to get to the "Administration" panel directly from the home page, you need to click the "admin..." link at the very bottom of the admin block.

You can also set/check settings for individual modules. Go to Configuration or Administration and click "Modules". You can then show/hide modules and change settings, if available for a given module.

Note: the web-based administration in Moodle 1.7 is totally revamped. All administration areas are listed in the Site Administration block and selecting an area, displays a list of subareas. The actual admin panels are shown in the content area as before.

Do not forget to save changes on each page you modified the settings.

Further customizing Moodle through config.php

Open your moodle folder and get file information on the config.php file. This file was created by Moodle, so most likely only user www has access to it. In order to be able to edit it manually, set the owner to admin with read/write privileges while leaving www as group. Make sure www can still read and write.

You can configure additional Moodle options by directly setting the \$CFG variable in that file. A number of optional flags are described in the config-dist.php file in the root folder of Moodle. Some additional options can be found in the forums.

Installing additional languages (Moodle 1.6 and 1.7 only)

By default, Moodle comes with English language pack only. If you want to have additional languages installed:

In Moodle 1.6: go to the Administration page, click the Language link, and finally click the Language Import Utility link.
In Moodle 1.7: click the Language link in the Admin block, then click Language Packs.

Select each language you want installed and click the installation button.

In Moodle 1.6, you should also set the site's default language on that page. In Moodle 1.7, click Language Settings to do it.

Make sure to visit the language page once a while to update the installed standard language packs.

Setting cron service

Now it is time to set up cron to perform periodic functions like sending mails, cleaning, backups, etc.

First, check whether all is ok with cron process by running it in the browser. Simply enter

```
http://domain.or.ipaddress/moodle/admin/cron.php
```

as the URL and you should see cron's output. Note that some cron functions, namely cleanup, are done only every 4th or 6th run, so feel free to try running it several times to see the differences.

If all is fine, open Terminal and enter

```
crontab -e
```

This will normally open cron control file using the vi editor. If the file does not exist, it will be created (and a message to that effect will briefly flash). To add a line, type i (lowercase-i) to start insert mode. Then type or paste

```
*/* * * * * curl -q -s -o /dev/null http://domain.or.ipaddress/moodle/admin/cron.php
```

and press the escape key to exit the insert mode. Save the file by typing :wq and return. If you mess up, you can exit without saving by typing :q! and return (after exiting insert mode first).

The URL for the command should look familiar, so requires no explanation. The */5 means that we will run every 5 minutes and the other asterisks mean we will run round the clock. You can change this if you wish. For example, to run cron 10, 30, and 50 minutes after each hour, enter '10,30,50' instead of */5'. To run it every 10 mins, change it to */10' (value before slash are minutes of the hour and value after slash is the step value). The -o option redirects the output to null device, that is a digital black hole. If you want to see the output, delete the '-o /dev/null' part or (to save yourself retyping when reactivating this option) replace o with 0 (zero) to make curl ignore it. The output will be then sent as email to you (you that is the admin user) on the local system.

If you want to see the current crontab settings, execute

```
crontab -l
```

(that is lowercase L not uppercase i) in the Terminal.

If you feel uncomfortable using Terminal and vi, you can create/edit crontab file using a utility like CronniX.

Wrap-up

This completes the basic installation.

You can throw away the phpinfo.php file. Moodle offers a link on the admin page which displays the same info.

Launch "Disk Utility" and do "Repair Disk Permissions" on your system volume. The comment high above about this being controversial still stands, so don't feel you must do it.

If you have DiskWarrior (3 or newer) from Alsoft, you might want to boot it up and optimize/repair disk directories.

You may want to make a backup now.

Happy Moodling :-)